DEEP LEARNING METHOD FOR PARTIAL DIFFERENTIAL EQUATIONS AND OPTIMAL PROBLEMS

by

Mo Zhou

Department of Mathematics
Duke University

Date: _____

Approved:

_____
Jianfeng Lu, Supervisor

_____
Jonathan Christopher Mattingly

_____
James H. Nolen

_____
Xiuyuan Cheng

Dissertation submitted in partial fulfillment of the
requirements for the degree of Doctor of Philosophy
in the Department of Mathematics
in the Graduate School of
Duke University

2023

ABSTRACT

DEEP LEARNING METHOD FOR PARTIAL DIFFERENTIAL EQUATIONS AND OPTIMAL PROBLEMS

by

Mo Zhou

Department of Mathematics
Duke University

Date: _____

Approved:

_____
Jianfeng Lu, Supervisor

_____
Jonathan Christopher Mattingly

_____
James H. Nolen

_____
Xiuyuan Cheng

An abstract of a dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Mathematics in the Graduate School of Duke University

2023

Abstract

Scientific computing problems in high dimensions are difficult to solve with traditional methods due to the curse of dimensionality. The recently fast developing machine learning techniques provide us a promising way to resolve this problem, elevating the field of scientific computing to new heights. This thesis collects my works on machine learning to solve traditional scientific computing problems during my Ph.D. studies, which include partial differential equation (PDE) problems and optimal control problems. The numerical algorithms in the works demonstrate significant advantage over traditional methods. Moreover, the theoretical analysis of the algorithms enhances our understanding of machine learning, providing guarantees that enable us to avoid treating it as a black box.

Acknowledgements

I would like to express my deepest gratitude to my advisor Prof. Jianfeng Lu, who has provided instructions in all aspects witnessed my growth. I'm also extremely grateful to Jiequn Han, who gave me great support in many research projects. I am also grateful to other research collaborators, including Manas Rachh and Carlos Borges. Many thanks to the committee for my preliminary exam and thesis defense, including Jonathan Mattingly, James Nolen, and Xiuyuan Cheng. Thanks should also go to the professors who wrote recommendation letters for me, including Jianfeng Lu, Jiequn Han, James Nolen, Manas Rachh, and Shira Viel. I would like to extend my sincere thanks to the staffs in the department, including Julia Gruhot, Andrew Schretter, Yunliang Yu, Laurie Triggiano, Delores Austin, etc., who keep providing us a comfortable research environment. I'd like to acknowledge my family and friends, who have supported me during my Ph.D. study.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

In recent years, machine learning has emerged as a powerful tool for scientific computing problem, especially in high dimensions. This thesis concludes most of the works I did during my Ph.D. studies on the combination of traditional scientific computing problems and machine learning techniques [HLZ20, ZHL21, ZL22, ZL23]. They include deep learning algorithms to solve PDE problems and optimal control problems, encompassing both numerical successes and theoretical analysis. Each of the work involves a great effort from me. A brief introduction of each work is given in each of the following sections, and the details for each work is presented in the following chapters. Some future directions of research are stated after introducing each of these works.

1.1 Solving high-dimensional eigenvalue problems using deep neural networks

In 2018, Han et al. proposed the deep BSDE method to solve high dimensional PDEs using artificial neural network [HJE18]. This method reformulates PDEs using backward stochastic differential equations and transform it into a shooting problem. The same idea could also be applied to solve the eigenvalue problem, which is very important in scientific computing, especially in quantum mechanics.

We modify the idea of deep BSDE and propose a new method to solve eigenvalue problems for linear and semilinear second order differential operators in high dimensions based on deep neural networks. The eigenvalue problem is reformulated as a fixed point problem of the semigroup flow induced by the operator, whose solution

can be represented by Feynman-Kac formula in terms of forward-backward stochastic differential equations. The method shares a similar spirit with diffusion Monte Carlo but augments a direct approximation to the eigenfunction through neural-network ansatz. The criterion of fixed point provides a natural loss function to search for parameters via optimization. Our approach is able to provide accurate eigenvalue and eigenfunction approximations in several numerical examples, including Fokker-Planck operator and the linear and nonlinear Schrödinger operators in high dimensions.

1.2 Actor-critic method for HJB equations

The DeepBSDE method could also be generalized to solve fully nonlinear PDEs, such as the Hamilton–Jacobi–Bellman (HJB) equation. In another work, we propose a novel numerical method for high dimensional HJB type elliptic PDEs. The HJB PDEs, reformulated as optimal control problems, are tackled by the actor-critic framework inspired by reinforcement learning, based on neural network parametrization of the value and control functions. Within the actor-critic framework, we employ a policy gradient approach to improve the control, while for the value function, we derive a variance reduced least-squares temporal difference method using stochastic calculus. To numerically discretize the stochastic control problem, we employ an adaptive step size scheme to improve the accuracy near the domain boundary. Numerical examples up to 20 spatial dimensions including the linear quadratic regulators, the stochastic Van der Pol oscillators, the diffusive Eikonal equations, and fully nonlinear elliptic PDEs derived from a regulator problem are presented to validate the effectiveness of our proposed method.

1.3 Single timescale actor-critic method for the linear quadratic regulator

The numerical successes of the deep learning methods for PDE problems motivate us to study the theoretical foundations. We aim to establish theoretical guarantee of machine learning algorithms. The first problem we consider is the linear quadratic regulator (LQR) in reinforcement learning, which is relatively easy compare with the optimal control problem due to its clear and simple structure.

We propose a single timescale actor-critic algorithm to solve the LQR problem. A least squares temporal difference (LSTD) method is applied to the critic and a natural policy gradient method is used for the actor. We give a proof of convergence with sample complexity $\mathcal{O}(\varepsilon^{-1}\log(\varepsilon^{-1})^2)$, which significantly outperform existing methods. The method in the proof is applicable to general single timescale bilevel optimization problems. We also numerically validate our theoretical results on the convergence. The technique for the convergence proof is also applicable to optimal control problems in continuous time, which is shown later.

1.4 Policy gradient for optimal control

Before proving the convergence guarantee for the single time-scale actor-critic method for the optimal control problem, we realize that the convergence for the actor only policy gradient method is already a hard task due to non-convexity. Therefore, we propose a innovative local optimal method to show this convergence. The dynamic can be viewed as the limit of a two time-scale actor-critic method as the critic becomes infinitely faster than the actor.

We consider the stochastic optimal control problem in continuous time and a policy gradient method to solve it. In particular, we study the gradient flow for

the control, viewed as a continuous time limit of the policy gradient. We prove the global convergence of the gradient flow and establish a convergence rate under some regularity assumptions. The main novelty in the analysis is the notion of local optimal control function, which is introduced to compare the local optimality of the iterate.

1.5 Actor-critic method for optimal control

After showing the convergence of the policy gradient method, we are fully prepared to prove the convergence of the single time-scale actor-critic method. The idea is to combine the methods in the previous three works.

We propose an single time-scale actor-critic framework to solve the stochastic optimal control problem with continuous time. We construct a policy gradient flow in the actor for policy improvement. We use the deep BSDE method to construct a variance reduced least square temporal difference method in the critic for policy evaluation. We show that the algorithm possesses a global convergence property under mild assumptions.

Chapter 2

Solving High-Dimensional Eigenvalue Problems Using Deep Neural Networks: A Diffusion Monte Carlo Like Approach

2.1 Introduction

Many fundamental problems in scientific computing can be reduced to the computation of eigenvalues and eigenfunctions of an operator. One primary example is the electronic structure calculations, namely, computing the leading eigenvalue and eigenfunction of the Schrödinger operator. If the dimension of the state variable is low, one can use classical approaches, such as the finite difference method or spectral method, to discretize the operator and to solve the eigenvalue problem. However, these conventional, deterministic approaches suffer from the so-called curse of dimensionality, when the underlying dimension becomes high, since the number of degrees of freedom grows exponentially as the dimension increases.

For high-dimensional problems, commonly arising from quantum mechanics, statistical mechanics, and finance applications, stochastic methods become more attractive and in many situations the only viable option. In the context of quantum mechanics, two widely used approaches for high-dimensional eigenvalue problems are the variational Monte Carlo (VMC) and diffusion Monte Carlo (DMC) methods [McM65, CCK77, BSS81, ZCG97, FMNR01, NTDR09]. These two approaches deal with the high dimensionality via different strategies. VMC relies on leveraging chemical knowledge to propose an ansatz of the eigenfunction (wavefunction in the context of quantum mechanics) with parameters to be optimized under the varia-

tional formulation of the eigenvalue problem. The Monte Carlo approach is used to approximate the gradient of the energy with respect to the parameters at each optimization iteration step. On the other hand, DMC represents the density of the eigenfunction with a collection of particles that follow the imaginary time evolution given by the Schrödinger operator, via a Feynman-Kac representation of the semi-group. It can be understood as a generalization of the classical power method from finite-dimensional matrices to infinite-dimensional operators. In electronic structure calculations, DMC usually can give more accurate eigenvalues compared with VMC, which relies on the quality of the proposed ansatz, while the particle representation of DMC often falls short of providing other information of the eigenfunction, such as its derivatives, unlike VMC.

As discussed above, one key to solving high-dimensional eigenvalue problems is the choice of function approximation to the targeted eigenfunction, ranging from the grid-based basis, spectral basis, to nonlinear parametrizations used in VMC, and to particle representations in DMC. Given the recent compelling success of neural networks in representing high-dimensional functions with remarkable accuracy and efficiency in various computational disciplines, it is fairly attempting to introduce neural networks to solve high-dimensional eigenvalue problems. This idea has been recently investigated under the variational formulation by [EY18, PPA$^+$19, HZE19, PSMF20, HSN20, CMC20]. Particularly [HZE19, PSMF20, HSN20, CMC20] has shown the exciting potential of solving the many-electron Schrödinger equation with neural networks within the framework of VMC. On the other hand, how to apply neural networks in the formalism of DMC has not been explored in the literature, which leaves a natural open direction to investigate.

In this paper, we propose a new algorithm to solve high-dimensional eigenvalue problem for the second-order differential operators, in a similar spirit of DMC while

based on the neural network parametrization of the eigenfunction. The eigenvalue problem is reformulated as a parabolic equation, whose solution can be represented by (nonlinear) Feynman-Kac formula in terms of forward-backward stochastic differential equations. Then we leverage the recently proposed deep BSDE method [EHJ17, HJE18] to seek optimal eigenpairs. Specifically, two deep neural networks are constructed to represent the eigenfunction and its scaled gradient. Then the neural network is propagated according to the semigroup generated by the operator. The loss function is defined as the difference between the neural networks before and after the propagation. Compared to conventional DMC, the proposed algorithm provides a direct approximation to the target eigenfunction, which overcomes the shortcoming in providing the gradient information. Moreover, since the BSDE formulation is valid for nonlinear operators, our approach can be extended to high-dimensional nonlinear eigenvalue problems, as validated in our numerical examples.

2.2 Numerical methods

2.2.1 The method for a linear operator

We consider the eigenvalue problem

$$\mathcal{L}\psi = \lambda\psi, \tag{2.1}$$

on $\Omega = [0, 2\pi]^d$ with periodic boundary condition where $\mathcal{L}$ is a linear operator of the form

$$\mathcal{L}\psi(x) = -\frac{1}{2}\operatorname{Tr}\left(\sigma\sigma^\top \operatorname{Hess}(\psi)(x)\right) - b(x)\cdot\nabla\psi(x) + f(x)\psi(x). \tag{2.2}$$

σ is a $d\times d$ constant invertible matrix such that $\sigma\sigma^\top$ is positive definite, $\nabla\psi$ denotes the gradient of ψ, $b(x)$ is a d-dimensional vector field and $\operatorname{Hess}(\psi)$ denotes the Hessian matrix of ψ.

To solve this eigenvalue problem, we augment a time variable and consider the following backward parabolic partial differential equation (PDE) in the time interval $[0, T]$:

$$\begin{cases} \partial_t u(t,x) - \mathcal{L}u(t,x) + \lambda u(t,x) = 0 & \text{in } [0,T] \times \Omega, \\ u(T,x) = \Psi(x) & \text{on } \Omega. \end{cases} \tag{2.3}$$

This is essentially a continuous time analog of the power iteration for matrix eigenvalue problem. Let us denote the solution of (2.3) as $u(T-t,\cdot) = \mathcal{P}_t^\lambda \Psi$ (note that the backward propagator $\{\mathcal{P}_t^\lambda\}_{t \le T}$ forms a semigroup, i.e., $\mathcal{P}_{t_1}^\lambda \circ \mathcal{P}_{t_2}^\lambda = \mathcal{P}_{t_1+t_2}^\lambda$). According to the spectral theory of the elliptic operator, if Ψ is a stationary solution of (2.3), i.e., $\mathcal{P}_T^\lambda \Psi = \Psi$, then (λ, Ψ) must be an eigenpair of $\mathcal{L}$. Therefore, we can minimize the "loss function" $\|\mathcal{P}_T^\lambda \Psi - \Psi\|^2$ with respect to (λ, Ψ) to solve the eigenvalue problem. While this is a non-convex optimization problem, we expect local convergence to a valid eigenpair with appropriate initialization. Note that, unlike power iteration in which λ is determined by the eigenvector, in our algorithm λ is treated as a variational parameter and optimized jointly with the eigenfunction, as in the DMC method.

The reformulation above turns the eigenvalue problem into solving a parabolic PDE in high dimensions. For the latter, we can leverage the recently developed deep BSDE method [EHJ17, HJE18, HL20] (which is why the parabolic PDE (2.3) is written backward in time). Let X_t solve the stochastic differential equation (SDE)

$$\mathrm{d}X_t = \sigma \, \mathrm{d}W_t, \tag{2.4}$$

or in the integral form

$$X_t = X_0 + \int_0^t \sigma \, \mathrm{d}W_s, \tag{2.5}$$

where W_t is a d-dimensional Brownian motion, and X_0 is sampled from some initial distribution ν. Then according to the Itô's formula, the solution to (2.3), $u(t,x)$

satisfies

$$u(t, X_t) = u(0, X_0) + \int_0^t (f(X_s)u(s, X_s) - \lambda u(s, X_s) - b(X_s)\nabla u(s, X_s))\,\mathrm{d}s + \int_0^t \sigma^\top \nabla u(s, X_s)\,\mathrm{d}W_s. \tag{2.6}$$

Note that simulating the two SDEs (2.5) and (2.6) is relatively simple even in high dimensions, while directly solving the PDE (2.3) is intractable. We remark that it is possible to add a drift term $b(X_t)\,\mathrm{d}t$ to the SDE (2.4) and modify (2.6) accordingly (see the discussion below).

Of course, a priori in (2.6) for both $u(s, \cdot)$ and $\nabla u(s, \cdot)$ are unknown, while we know that if we set $u(s, \cdot) = \Psi(\cdot)$, the eigenfunction we look for, and $\nabla u(s, \cdot) = \nabla\Psi(\cdot)$, the solution $u(t, \cdot)$ remains $\Psi(\cdot)$ for all t. The idea of our method is then to use two neural networks, $\mathfrak{N}_\Psi$ and $\mathfrak{N}_{\sigma^\top\nabla\Psi}$ as ansatz for the eigenfunction Ψ and its scaled gradient $\sigma^\top\nabla\Psi$, respectively. Assigning $u(0, X_0) = \mathfrak{N}_\Psi(X_0)$ and $\sigma^\top\nabla u(s, X_s) = \mathfrak{N}_{\sigma^\top\nabla\Psi}(X_s)$ in (2.6), the discrepancy for the propagated solution, i.e.,

$$\mathbb{E}_{X_0\sim\nu}\Big[\eta_1\big|\mathfrak{N}_\Psi(X_T) - u(T, X_T)\big|^2 + \eta_2\big|\mathfrak{N}_{\sigma^\top\nabla\Psi}(X_T) - \sigma^\top\nabla\mathfrak{N}_\Psi(X_T)\big|^2\Big] \tag{2.7}$$

then indicates the accuracy of the approximation. Here, we use $|\cdot|$ to denote the absolute value of a number or the Euclidean norm of a vector according to the context. Note that the second term above penalizes the discrepancy between the approximation of Ψ and its gradient, where η_1, η_2 are two weight hyperparameters. Therefore, using the above discrepancy as a loss function to optimize the triple $(\lambda, \mathfrak{N}_\Psi, \mathfrak{N}_{\sigma^\top\nabla\Psi})$ gives us a scheme to solve the eigenvalue problem. The above procedure can be directly extended to semilinear case where f depends on Ψ and $\nabla\Psi$, as we will discuss in Section 2.2.3.

To employ the above framework in practice, we numerically discretize the SDEs (2.5) and (2.6) using Euler–Maruyama method with a given partition of interval $[0, T]$

: $0 = t_0 < t_1 < \cdots < t_N = T$:

$$\mathcal{X}_0 = X_0, \qquad \mathcal{X}_{t_{n+1}} = \mathcal{X}_{t_n} + \sigma \Delta W_n, \tag{2.8}$$

and

$$\begin{aligned} \mathcal{U}_0 = \mathfrak{N}_\Psi(\mathcal{X}_0), \;\; \mathcal{U}_{t_{n+1}} = \mathcal{U}_{t_n} + \big(f(\mathcal{X}_{t_n})\mathcal{U}_{t_n} - \lambda \mathcal{U}_{t_n} - (b\sigma^{-1^\top} \mathfrak{N}_{\sigma^\top \nabla \Psi})(\mathcal{X}_{t_n})\big)\, \Delta t_n \\ + \mathfrak{N}_{\sigma^\top \nabla \Psi}(\mathcal{X}_{t_n}) \Delta W_n, \end{aligned} \tag{2.9}$$

for $n = 0, 1, \ldots, N-1$. Here $\Delta t_n = t_{n+1} - t_n$, $\Delta W_n = W_{t_{n+1}} - W_{t_n}$, and we use X_t and $\mathcal{X}_t/\mathcal{U}_t$ to represent the continuous and discretized stochastic process, respectively. The noise terms ΔW_n have the same realization in (2.8) and (2.9), as in the forward-backward SDEs (2.5) and (2.6).

The loss function (2.7) then corresponds to the discrete counterpart:

$$\mathbb{E}_{X_0 \sim \nu}\Big[\eta_1 \big|\mathfrak{N}_\Psi(\mathcal{X}_T) - \mathcal{U}_T\big|^2 + \eta_2 \big|\mathfrak{N}_{\sigma^\top \nabla \Psi}(\mathcal{X}_T) - \sigma^\top \nabla \mathfrak{N}_\Psi(\mathcal{X}_T)\big|^2\Big], \tag{2.10}$$

where $\nabla \mathfrak{N}_\Psi$ is the gradient of neural network $\mathfrak{N}_\Psi$ with respect to its input. In practice, the expectation in (2.10) is further approximated by Monte Carlo sampling, which is similar to the empirical loss often used in the supervised learning context. For a given batch size K, we sample K points $\{\mathcal{X}_0^k\}_{k=1}^K$ of the initial state from the distribution ν at each training step and estimate the gradient of the loss with respect to the trainable parameters using the empirical Monte Carlo average of (2.10):

$$\frac{1}{K} \sum_{k=1}^{K} \Big[\eta_1 \big|\mathfrak{N}_\Psi(\mathcal{X}_T^k) - \mathcal{U}_T^k\big|^2 + \eta_2 \big|\mathfrak{N}_{\sigma^\top \nabla \Psi}(\mathcal{X}_T^k) - \sigma^\top \nabla \mathfrak{N}_\Psi(\mathcal{X}_T^k)\big|^2\Big]. \tag{2.11}$$

We remark that the definition of the dynamic (2.5) is not unique and implicitly affects the detailed computation of the loss function (2.10) and (2.11). Specifically, in (2.5), the diffusion term σ is determined by the operator (2.2) while the choice of initial distribution and the drift term has some flexibility. If the drift in (2.5) changes, one can change the associated drift in (2.6) according to Itô's formula and define the loss

again as (2.10) and (2.11). In this work, we choose the form of (2.5) without the drift and ν being the uniform distribution on Ω to ensure that the whole region is reasonably sampled for the optimization of eigenfunction. Some importance sampling can also be used if some prior knowledge of the eigenfunction is available, which we will not go into further details in this work.

At a high level, our algorithm is in a similar vein as the power method for solving the eigenvalue problem in linear algebra. Both algorithms seek for a solution that is stationary under the propagation. However, one distinction is that our algorithm may also be used for general eigenvalues depending on the initialization of λ and Ψ. Using matrix notation, this is similar to using the objective function $\|(A-\lambda)v\|^2$ to find non-dominant eigenpair λ and v for the matrix A. The actual performance of solving for non-dominant eigenvalue depends on the initialization and spectral gap between eigenvalues, as will be illustrated by numerical results in Section 3. On the other hand, we find in the numerical experiments that if λ is initialized small enough, it will always converge to the first eigenvalue.

In practice, we use fully-connected feed-forward neural networks for the approximation of Ψ and $\sigma^\top \nabla \Psi$, respectively. To ensure periodicity of the neural network outputs, the input vector $x = (x_1, \ldots, x_d)$ is first mapped into a fixed trigonometric basis $\{\sin(jx_i), \cos(jx_i)\}_{i=1,j=1}^{d,\ M}$ of order M. Then the vector consisting of all basis components are fed into fully-connected neural networks with some hidden layers, each with several nodes. See Figure 2.1 for illustration of the involved network structure. We use ReLU as the activation function and optimize the parameters with the Adam optimizer [KB15].

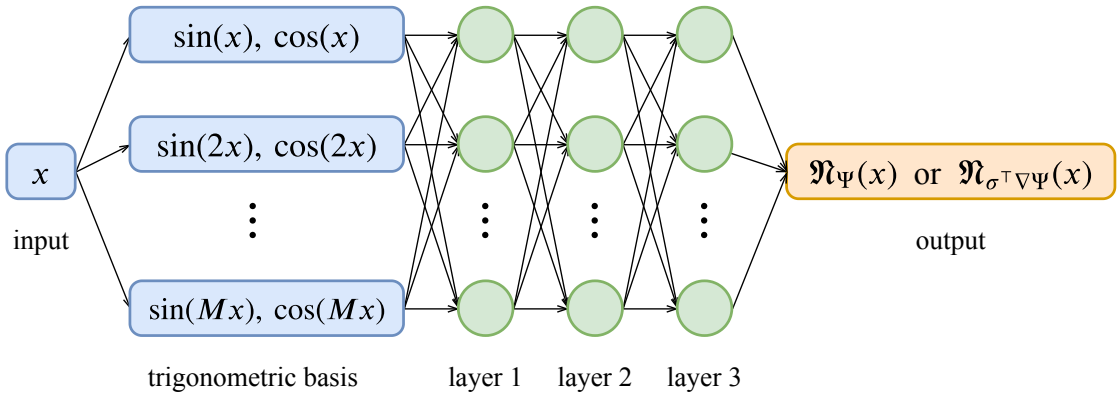

Figure 2.1: Illustration of the neural network.

2.2.2 Normalization

The above loss has one caveat though, as the trivial solution ($\mathfrak{N}_\Psi = 0, \mathfrak{N}_{\sigma^\top\nabla\Psi} = 0$) is a global minimizer. Therefore, normalization is required to exclude such a trivial case. We seek for eigenfunctions Ψ such that $\int_\Omega \Psi^2 = |\Omega|$, i.e., $\frac{1}{|\Omega|}\|\Psi\|_{L^2}^2 = 1$. [1] To proceed, we define the normalization constant

$$Z_\Psi = \operatorname{sign}\Big(\int_\Omega \mathfrak{N}_\Psi(x)\,\mathrm{d}x\Big)\Big(\frac{1}{|\Omega|}\int_\Omega \mathfrak{N}_\Psi(x)^2\,\mathrm{d}x\Big)^{1/2}. \tag{2.12}$$

Thus dividing $\mathfrak{N}_\Psi$ by Z_Ψ, we enforce that the parametrized function ensures the normalization condition. Note that the first factor on the right hand side of (2.12) is introduced to fix the global sign ambiguity of the eigenfunction.

In computation, given the parameters of $\mathfrak{N}_\Psi$, we do not have direct access to Z_Ψ. Instead, at the ℓ-th step of training, we use our batch of K data samples to

[1] The reason we set $\frac{1}{|\Omega|}\|\Psi\|_{L^2}^2 = 1$ instead of $\|\Psi\|_{L^2}^2 = 1$ is because we want to consider the problem in high dimensions in the domain $\Omega = [0, 2\pi]^d$. Consider the trivial case when $\mathcal{L} = -\Delta$, whose smallest eigenvalue is $\lambda = 0$ and any constant function is a corresponding eigenfunction. If $\|\Psi\|_{L^2}^2 = 1$, the constant function becomes $\Psi = (\frac{1}{2\pi})^{d/2}$, which vanishes as $d \to \infty$; instead the normalization $\|\Psi\|_{L^2}^2 = |\Omega|$ keeps the pointwise-value of Ψ as order 1, which benefits the training process.

approximate Z_Ψ via

$$\hat{Z}_\Psi^\ell = \operatorname{sign}\Big(\sum_{k=1}^K \mathfrak{N}_\Psi(\mathcal{X}_0^{k,\ell})\Big)\left(\frac{1}{K}\sum_{k=1}^K \mathfrak{N}_\Psi(\mathcal{X}_0^{k,\ell})^2\right)^{1/2}, \tag{2.13}$$

where the superscripts in $\mathcal{X}_0^{k,\ell}$ serve as the index of batch (k) and index of the training step (ℓ). The above is a Monte Carlo estimation of (2.12) if $\mathcal{X}_0$ is sampled from the uniform distribution (which we assume in this work). Due to the normalization procedure, $\hat{Z}_\Psi^\ell$ will enter into the loss function, and thus the stochastic gradient based on the empirical average over the batch becomes biased (since $\mathbb{E}(A/B) \neq \mathbb{E}A/\mathbb{E}B$ in general). To reduce the bias and make the training more stable, we introduce an exponential moving average scheme to the normalization constant in order to reduce the dependence of the loss on the estimated normalization constant of the current batch. In our implementation, we use

$$Z_\Psi^\ell = \gamma_\ell Z_\Psi^{\ell-1} + (1-\gamma_\ell)\hat{Z}_\Psi^\ell. \tag{2.14}$$

Here $\gamma_\ell \in (0,1)$ is the moving average coefficient for decay. It is observed that small γ_ℓ at the beginning makes training efficient, and later on its value is increased such that the gradient is less biased.

Given the introduced normalization factor, the neural network approximation $\mathcal{U}_0$ in the updating scheme (2.9) is replaced by (we suppress the training step index in $\mathcal{X}$)

$$\mathcal{U}_0^k = \frac{\mathfrak{N}_\Psi(\mathcal{X}_0^k)}{Z_\Psi^\ell}, \tag{2.15}$$

and we would hope to reduce the discrepancy between the solution of (2.9) at time T and the normalized neural network approximation $\mathfrak{N}_\Psi(\mathcal{X}_T^k)/Z_\Psi^\ell$ through training. The associated batch approximation of loss function used for the computation of

stochastic gradient is as follows

$$\frac{1}{K}\sum_{k=1}^{K}\left(\eta_1\left|\frac{\mathfrak{N}_\Psi(\mathcal{X}_T^k)}{Z_\Psi^\ell}-\mathcal{U}_T^k\right|^2+\eta_2\left|\mathfrak{N}_{\sigma^\top\nabla\Psi}(\mathcal{X}_T^k)-\frac{\sigma^\top\nabla\mathfrak{N}_\Psi(\mathcal{X}_T^k)}{Z_\Psi^\ell}\right|^2\right)+\eta_3(Z_0-Z_\Psi^\ell)^+. \tag{2.16}$$

In the last term above, Z_0 is a hyperparameter and η_3 is the associated weight. This term is introduce to prevent Z_Ψ^ℓ being too small; otherwise the normalization would become unstable. In each training step, we calculate the gradient of (2.16) with respect to all the parameters to be optimized, including the eigenvalue λ and parameters in the neural network ansatz $\mathfrak{N}_\Psi$ and $\mathfrak{N}_{\sigma^\top\nabla\Psi}$. Note that in (2.16) we do not normalize $\mathfrak{N}_{\sigma^\top\nabla\Psi}$ since its scale has been determined implicitly. When K is reasonably large and if we neglect the discretization error of simulating the SDEs, the empirical sum in (2.16) can be interpreted as a Monte Carlo approximation to the loss (ignoring the sign ambiguity)

$$\mathbb{E}_{X_0\sim\nu}\left[\eta_1\left|\frac{\mathfrak{N}_\Psi(X_T)}{\|\mathfrak{N}_\Psi\|_2/|\Omega|^{\frac{1}{2}}}-u(T,X_T)\right|^2+\eta_2\left|\mathfrak{N}_{\sigma^\top\nabla\Psi}(X_T)-\frac{\sigma^\top\nabla\mathfrak{N}_\Psi(X_T)}{\|\mathfrak{N}_\Psi\|_2/|\Omega|^{\frac{1}{2}}}\right|^2\right], \tag{2.17}$$

where $u(T,X_T)$ is defined as (2.6) except that $u(0,X_0)=|\Omega|^{\frac{1}{2}}\mathfrak{N}_\Psi(X_0)/\|\mathfrak{N}_\Psi\|_2$. We remark that the normalization procedure introduced here shares a similar spirit with Batch Normalization [IS15], which is widely used in the training of neural networks.

We summarize our algorithm as pseudocode in Algorithm 1.

2.2.3 The method for a semilinear operator

Our algorithm can be generalized to solve eigenvalue problems for semilinear operator

$$\mathcal{L}\psi(x)=-\frac{1}{2}\operatorname{Tr}\left(\sigma\sigma^\top\operatorname{Hess}(\psi)(x)\right)-b(x)\cdot\nabla\psi(x)+f(x,\psi(x),\sigma^\top\nabla\psi(x)). \tag{2.18}$$

The method for semilinear problems is almost the same to previous sections, except for a few modifications. The SDE for X_t is the same as (2.5) while equation (2.6)

Algorithm 1 Neural network based eigensolver

Input: operator $\mathcal{L}$, terminal time T, number of time intervals N, loss weights η_1, η_2, η_3, Z_0, neural network structures, number of iterations, learning rate, batch size K, moving average coefficient γ_ℓ in (2.14)

Output: eigenvalue λ, eigenfunction $\mathfrak{N}_\Psi$ and rescaled gradient $\mathfrak{N}_{\sigma^\top \nabla \Psi}$

initialization: eigenvalue λ, $\mathfrak{N}_\Psi$, $\mathfrak{N}_{\sigma^\top \nabla \Psi}$ and normalization factor Z_Ψ^0

for $\ell = 1$ **to** the number of iterations **do**

sample K points of $\mathcal{X}_0$ and sample K Wiener processes W_t

compute $\mathcal{X}_{t_n}$ via (2.8)

compute the normalization factor Z_Ψ^ℓ via (2.13) and (2.14)

normalize and propagate via (2.15) and (2.9)

compute the gradient of loss (2.16) with respect to the trainable parameters

update the trainable parameters by the Adam method

end for

that the solution of the PDE (2.3) satisfies becomes

$$\begin{aligned} u(t, X_t) =& u(0, X_0) + \int_0^t \sigma^\top \nabla u(s, X_s) \, \mathrm{d}W_s \\ &+ \int_0^t \left(f(X_s, u(s, X_s), \sigma^\top \nabla u(s, X_s)) - b(X_s) \cdot \nabla u(s, X_s) - \lambda u(s, X_s) \right) \mathrm{d}s. \end{aligned} \tag{2.19}$$

The discretization of X_t, equation (2.8), remains unchanged while equation (2.9) needs modification according to (2.19):

$$\begin{aligned} \widetilde{\mathcal{U}}_{t_{n+1}} =& \mathcal{U}_{t_n} + \left(f(\mathcal{X}_{t_n}, \mathcal{U}_{t_n}, \mathfrak{N}_{\sigma^\top \nabla \Psi}) - \lambda \mathcal{U}_{t_n} - (b \sigma^{-\top} \mathfrak{N}_{\sigma^\top \nabla \Psi})(\mathcal{X}_{t_n}) \right) \Delta t_n \\ &+ \mathfrak{N}_{\sigma^\top \nabla \Psi}(\mathcal{X}_{t_n}) \Delta W_n, \\ \mathcal{U}_{t_{n+1}} =& \mathrm{Clip}(\widetilde{\mathcal{U}}_{t_{n+1}}, P, Q), \end{aligned} \tag{2.20}$$

where (for $P < Q$) Clip is a clipping function given by

$$\mathrm{Clip}(u, P, Q) = \begin{cases} P, & \text{if } u < P; \\ u, & \text{if } P \le u \le Q; \\ Q, & \text{if } u > Q. \end{cases} \tag{2.21}$$

Here we introduce the clipping function ($-P = Q > 0$) to prevent numerical instability caused by the nonlinearity of f in (2.19) (for instance, the cubic term in the nonlinear Schrödinger equation (2.25) below), especially at the early stage of training. It checks $\mathcal{U}_{t_n}$ and replaces those whose absolute values are larger than Q with $\text{sign}(\mathcal{U}_{t_n})Q$, where $Q > 0$ is an upper bound of the absolute value of the true normalized eigenfunction. Given the modified forward dynamics (2.20), the loss function for the semilinear operators are defined the same as (2.16), and the training algorithm is the same too.

2.3 Numerical results

In this section, we report the performance of the proposed eigensolver in three examples: the Fokker-Planck equation, the linear Schrödinger equation, and the nonlinear Schrödinger equation. The domain Ω is always $[0, 2\pi]^d$ with periodic boundary condition. In each example we consider the dimension $d = 5$ and $d = 10$. We also solve the second eigenpair of the linear Schrödinger equation with $d = 10$ to illustrate that our algorithm is able to get non-dominant eigenpairs. All The hyperparameters are given in 2.5. We examine the errors of the prescribed eigenvalue, the associated eigenfunction, and the gradient of the eigenfunction. The errors for eigenfunctions and gradients of eigenfunctions are computed in the L^2 and L^∞ sense, approximated through a set of validation points. Given a set of validation points $\{\mathcal{X}^k\}_{k=1}^K$, we use the quantity

$$\text{err}_\Psi := \left[\frac{1}{K} \sum_{k=1}^{K} \left(\frac{\mathfrak{N}_\Psi(\mathcal{X}^k)}{Z_\Psi^\ell} - \frac{\Psi(\mathcal{X}^k)}{\left(\frac{1}{K}\sum_{m=1}^{K} \Psi(\mathcal{X}^m)^2\right)^{\frac{1}{2}}} \right)^2 \right]^{\frac{1}{2}} \tag{2.22}$$

and

$$\mathrm{err}_{\Psi}^{\infty} \coloneqq \max_{k} \left| \frac{\mathfrak{N}_{\Psi}(\mathcal{X}^k)}{Z_{\Psi}^{\ell}} - \frac{\Psi(\mathcal{X}^k)}{\left(\frac{1}{K}\sum_{m=1}^{K} \Psi(\mathcal{X}^m)^2\right)^{\frac{1}{2}}} \right| \tag{2.23}$$

to measure the error for eigenfunction where Z_{Ψ}^{ℓ} is computed via equation (2.14), with a known reference eigenfunction Ψ. We use

$$\begin{aligned} \mathrm{err}_{\sigma^\top \nabla \Psi} \coloneqq \Bigg[\frac{1}{Kd} \sum_{k=1}^{K} \Bigg| & \frac{\mathfrak{N}_{\sigma^\top \nabla \Psi}(\mathcal{X}^k)}{\left(\frac{1}{Kd}\sum_{m=1}^{K} |\mathfrak{N}_{\sigma^\top \nabla \Psi}(\mathcal{X}^m)|^2\right)^{\frac{1}{2}}} \\ & - \frac{\sigma^\top \nabla \Psi(\mathcal{X}^k)}{\left(\frac{1}{Kd}\sum_{m=1}^{K} |\sigma^\top \nabla \Psi(\mathcal{X}^m)|^2\right)^{\frac{1}{2}}} \Bigg|^2 \Bigg]^{\frac{1}{2}} \end{aligned} \tag{2.24}$$

to quantify the error for the gradient approximation. We record and plot the error every 100 steps in the training process, with a smoothed moving average of window size 10. The final error reported is based on the average of the last 1000 steps. Besides the errors above, we also visualize and compare the density of the true eigenfunction and its neural network approximation (since it is hard to visualize the high-dimensional eigenfunction directly). The density of a function Ψ is defined as the probability density function of $\Psi(X)$ where X is a uniformly distributed random variable on Ω. In practice, the density is approximated by Monte Carlo sampling. As shown below, in all three examples, we find that the eigenpairs (with gradients) are solved accurately and the associated densities match well.

2.3.1 Fokker-Planck equation

In this subsection we consider the linear Fokker-Planck operator

$$\mathcal{L}\psi = -\Delta\psi - \nabla \cdot (\psi \nabla V) = -\Delta\psi - \nabla V \cdot \nabla\psi - \Delta V \psi,$$

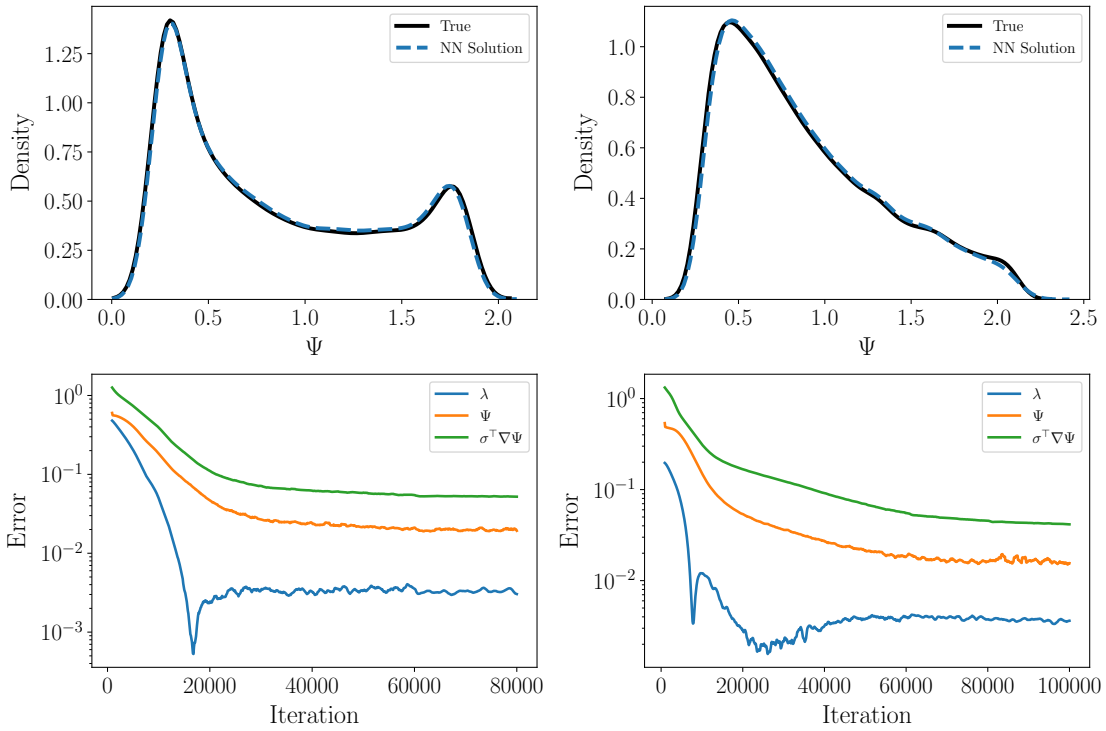

Figure 2.2: Density and error curves for Fokker-Planck equation.

where $V(x)$ is a potential function. The smallest eigenvalue of $\mathcal{L}$ is $\lambda_1 = 0$ and the corresponding eigenfunction is $\Psi(x) = e^{-V(x)}$, which can be used to compute the error. We consider an example $V(x) = \sin\big(\sum_{i=1}^{d} c_i \cos(x_i)\big)$, where x_i is the i-th coordinate of x, and c_i takes values in $[0.1, 1]$. The function V is periodic by construction. Figure 2.2 shows the density and error curves for the Fokker-Planck equation in $d = 5$ and $d = 10$. For $d = 5$, the final errors of the eigenvalue, eigenfunction (in L^2 and L^∞) and the scaled gradient are 3.08e-3, 2.91e-2, 1.25e-1 and 4.91e-2. For $d = 10$, the final errors are 3.58e-3, 1.62e-2, 1.08e-1 and 4.10e-2.

2.3.2 Linear Schrödinger equation

In this subsection we consider the linear Schrödinger operator

$$\mathcal{L}\psi = -\Delta\psi + V\psi,$$

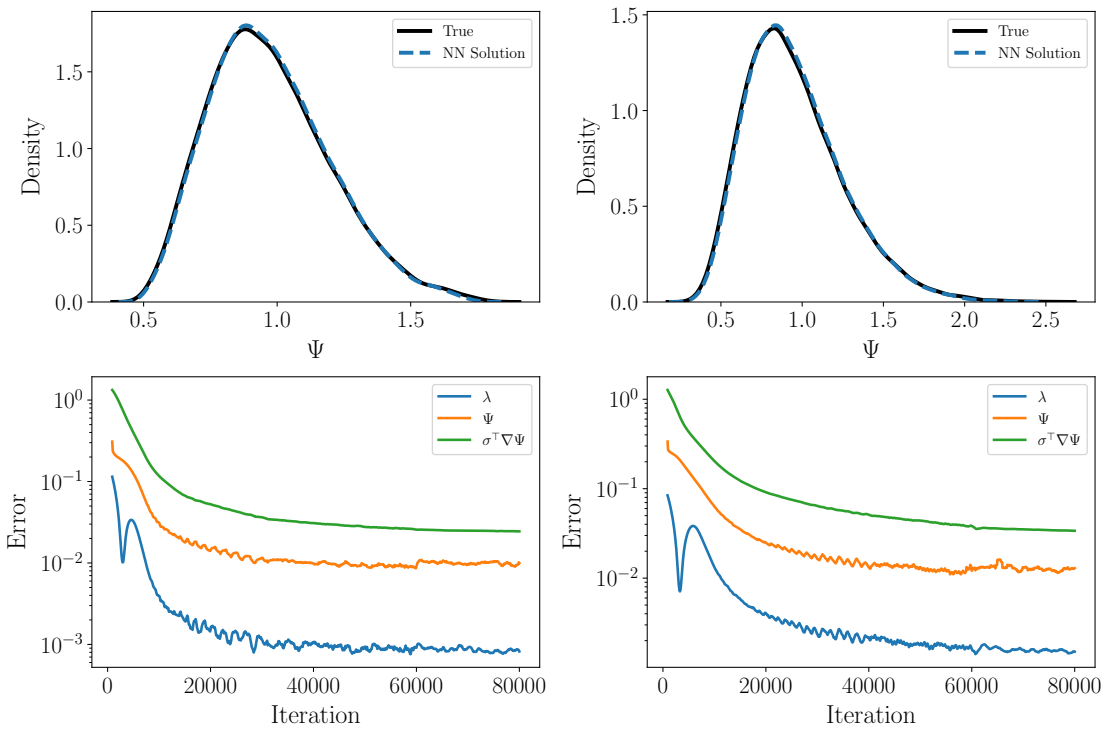

Figure 2.3: Density and error curves for linear Schrödinger equation.

where $V(x)$ is a potential function. Here we choose $V(x) = \sum_{i=1}^{d} c_i \cos(x_i)$, in which c_i takes values in $[0, 0.2]$. With potential function being such a form, the problem is essentially decoupled. Therefore we are able to compute the eigenvalues and eigenfunctions in each dimension through the spectral method and obtain the final first eigenpair with high accuracy for comparison. The computation details are provided in Section 2.4. Figure 2.3 shows the density and error curves for the Schrödinger equation in $d = 5$ and $d = 10$. For $d = 5$, the final errors of the eigenvalue, eigenfunction (in L^2 and L^∞) and the scaled gradient are 8.84e-4, 7.87e-3, 4.78e-2 and 2.30e-2. For $d = 10$, the final errors are 1.40e-3, 1.19e-2, 7.48e-2 and 3.14e-2.

2.3.3 Nonlinear Schrödinger equation

We finally consider a nonlinear Schrödinger operator with a cubic term, arising from the Gross-Pitaevskii equation [Gro61, Pit61] for the single-particle wavefunction in a Bose-Einstein condensate:

$$\mathcal{L}\psi = -\Delta\psi + \epsilon\psi^3 + V\psi. \tag{2.25}$$

Here we assume $\epsilon = 1$ and consider a specific external potential

$$V(x) = -\frac{1}{c^2}\exp\left(\frac{2}{d}\sum_{i=1}^{d}\cos x_i\right) + \sum_{i=1}^{d}\left(\frac{\sin^2 x_i}{d^2} - \frac{\cos x_i}{d}\right) - 3, \tag{2.26}$$

such that $\lambda = -3, \Psi(x) = \exp(\frac{1}{d}\sum_{j=1}^{d}\cos(x_j))/c$ is an eigenpair of the operator (2.25). Here c is a positive constant such that $\int_\Omega \Psi^2(x)\mathrm{d}x = |\Omega|$. In this example we used $P = -5$ and $Q = 5$ in the clip function (2.21). Figure 2.4 shows the density and error curves for the nonlinear Schrödinger equation in $d = 5$ and $d = 10$. For $d = 5$, the final errors of the eigenvalue, eigenfunction in L^2 and L^∞ and the scaled gradient are 1.53e-3, 8.07e-3, 6.58e-2 and 4.36e-2. For $d = 10$, the final errors are 2.6e-4, 4.60e-3, 3.83e-2 and 3.55e-2.

2.3.4 An example for the second eigenpair

In this subsection we reconsider the Schrödinger operator

$$\mathcal{L}\psi = -\Delta\psi + V\psi,$$

with the additional goal of finding the second eigenpairs. The potential function is double-well in each dimension: $V(x) = \sum_{i=1}^{d} A_i\cos(2x_i)$. The reference solutions are solved by the same numerical method as in Section 2.3.2.

We first consider the cases when the eigenvalues are well-separated. Assuming $d = 10$, $A_1 = 1.5$ and $A_i = 0.2$ for $2 \le i \le 10$, the associated eigen-gaps are $\lambda_2 - \lambda_1 =$

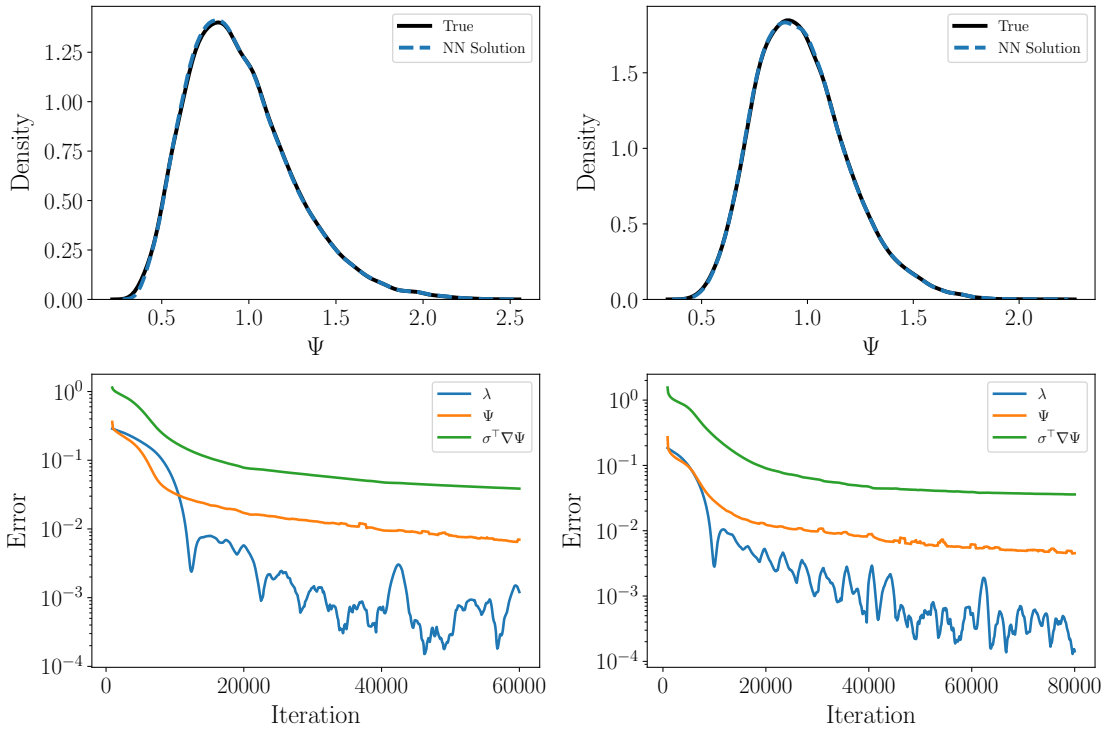

Figure 2.4: Density and error curves for nonlinear Schrödinger equation.

4.52e-1 and $\lambda_3 - \lambda_2 = 4.52\text{e-}1$. If we follow the training procedure described previously, we are able to find the first eigenpair with errors of 2.34e-3, 3.95e-3, 2.37e-2 and 1.21e-2. The left column in Figure 2.5 shows the density and error curves.

On the other hand, our method is also able to find the second eigenpair given some mild prior estimate of the eigenvalue. Suppose that we have an approximation $\overline{\lambda}$ of the true second eigenvalue λ_2. We can firstly fix this approximated eigenvalue and train the neural networks only with some steps. With such a pre-training procedure, we expect that the neural networks would reach a reasonable approximation of the second eigenfunction and its scaled gradient. Then, we train both the eigenvalue and the neural networks for the eigenfunction until convergence. The right column in Figure 2.5 shows the density and error curves of the second eigenpair, obtained by following the described procedure. In this example, $\overline{\lambda} = \lambda_2 + 0.1$, and the final errors

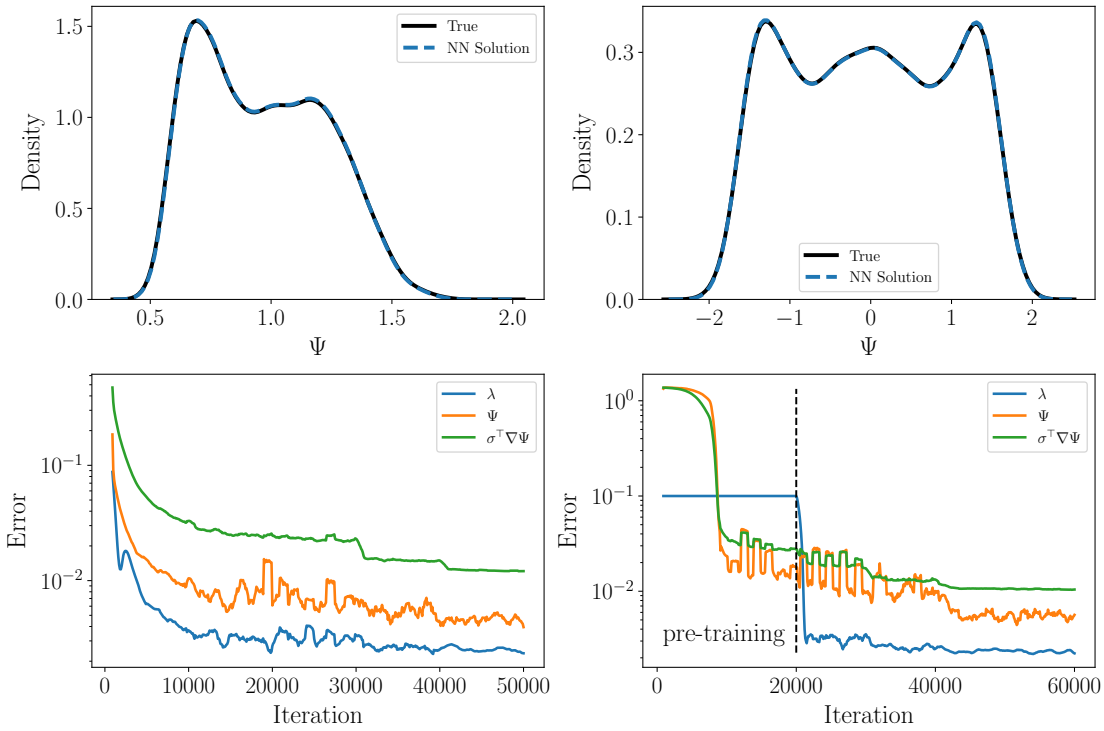

Figure 2.5: Density and error curves for the second eigenfunction.

are 2.23e-3, 5.65e-3, 5.08e-2 and 1.04e-2.

We remark that the eigenvalue problem becomes more challenging when it is nearly degenerate, i.e., the first and second eigenvalues are close to each other. This is a common phenomenon for various numerical methods, and our algorithm is no exception. Suppose we consider a one-dimensional problem with potential $V(x) = 5\cos(2x)$. The first and second eigenvalues are -2.153 and -2.076 respectively, with a small gap 8.7e-2. If we train the model for the first eigenpair like before, the obtained eigenfunction is plotted on the left of Figure 2.6. If we solve the second eigenpair with the additional pre-training procedure introduced above, we still cannot get the second eigenfunction approximately, even with a fixed true second eigenvalue. The results can be improved if we furthermore have certain approximation to the second eigenfunction. For instance, if we initialize the neural networks

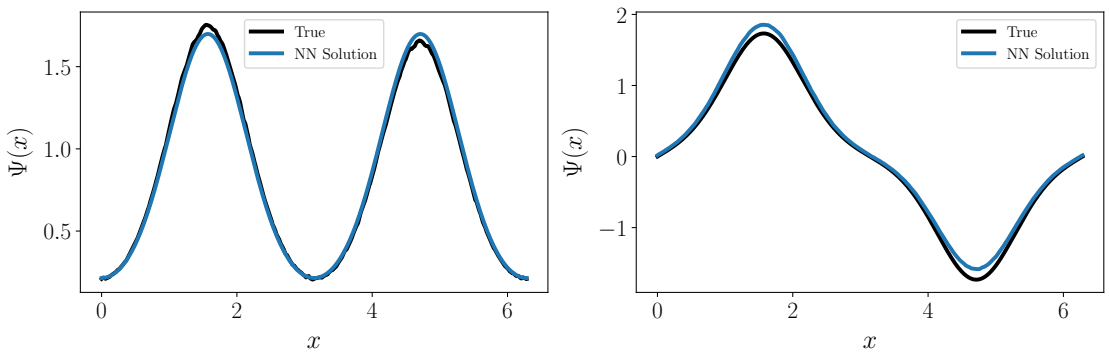

Figure 2.6: Plot of the eigenfunctions.

for the eigenfunction as $\psi_{init} = \psi_2 + \epsilon\psi_1$ where ψ_1 and ψ_2 are the first and second eigenfunctions respectively and ϵ denotes the degree of perturbation. We take $\epsilon = 0.3$ in our numerical experiment, and we remark that the direction of perturbation is not necessarily in ψ_1. Such initialization can be achieved through a supervised learning procedure. With this initialization, the final solution to the second eigenfunction is plotted on the right of Figure 2.6.

According to the numerical results of both the well-separated and degenerate problems, we find that better initialization of eigenvalue or eigenfunctions can usually improve results. For the degenerate problem, even in the one-dimensional case, both the first and second eigenpairs are difficult to solve accurately since the final solution may involve a mix of two eigenfunctions.

2.4 Spectrum method for linear Schrödinger equation

Suppose that the potential function in the linear Schrödinger operator $\mathcal{L} = -\Delta + V$ is decoupled with the form $V(x) = \sum_{j=1}^{d} c_j \cos(x_j)$, then we can solve the corresponding eigenvalue problem in a decoupled way. Specifically, assume we can solve the one-

dimensional eigenvalue problem

$$-\Psi_j''(x) + c_j \cos(x)\Psi_j(x) = \lambda_j \Psi_j(x), \quad x \in [0, 2\pi]. \tag{2.27}$$

Then one can easily verify that $\lambda = \sum_{j=1}^{d} \lambda_j$ and

$$\Psi(x) = \prod_{j=1}^{d} \Psi_j(x_j) \tag{2.28}$$

together define an eigenpair of the original high-dimensional Schrödinger operator.

To solve (2.27), we can employ the classical spectrum method. For a fixed $N \in \mathbb{N}$, assume that

$$\Psi_j(x) = \sum_{m=-N}^{N} a_m^j e^{imx}, \tag{2.29}$$

then

$$\Psi_j'(x) = \sum_{m=-N}^{N} m a_m^j e^{imx} i. \tag{2.30}$$

Let $\varphi_n(x) = e^{inx}$ $(n = -N, \cdots, N)$ be the test functions. By (2.27) and periodicity, we have

$$\int_0^{2\pi} (\Psi_j'(x)\varphi_n'(x) + c_j \cos(x)\Psi_j(x)\varphi_n(x))\mathrm{d}x = \lambda_j \int_0^{2\pi} \Psi_j(x)\varphi_n(x)\mathrm{d}x. \tag{2.31}$$

Since $\int_0^{2\pi} e^{imx} e^{inx}\mathrm{d}x = 2\pi\delta_{m+n}$ and

$$\int_0^{2\pi} \cos(x) e^{imx} e^{inx}\mathrm{d}x = \frac{1}{2}\int_0^{2\pi} (e^{i(m+n+1)x} + e^{i(m+n-1)x})\mathrm{d}x = \pi(\delta_{m+n+1} + \delta_{m+n-1}),$$

the left- and right-hand sides of (2.31) become

$$\begin{aligned}
&\int_0^{2\pi} (\Psi_j'(x)\varphi_n'(x) + c_j\cos(x)\Psi_j(x)\varphi_n(x))\mathrm{d}x \\
&= \sum_{m=-N}^{N} a_m^j \int_0^{2\pi} (-mne^{imx}e^{inx} + c_j\cos(x)e^{imx}e^{inx})\mathrm{d}x \\
&= \sum_{m=-N}^{N} a_m^j \pi(-2mn\delta_{m+n} + c_j\delta_{m+n+1} + c_j\delta_{m+n-1}) \\
&= \pi(2n^2 a_{-n} + c_j a_{-n-1} + c_j a_{-n+1}),
\end{aligned} \tag{2.32}$$

(assuming $a_m = 0$ for $|m| > N$) and

$$\int_0^{2\pi} \Psi_j(x)\varphi_n(x)\mathrm{d}x = 2\pi a_{-n}, \tag{2.33}$$

respectively. Therefore, equation (2.31) can be rewritten in the matrix form:

$$\pi \begin{bmatrix} 2N^2 & c_j & & & & & & \\ c_j & \ddots & \ddots & & & & & \\ & \ddots & 2 & \ddots & & & & \\ & & \ddots & 0 & \ddots & & & \\ & & & \ddots & 2 & \ddots & & \\ & & & & \ddots & \ddots & c_j & \\ & & & & & c_j & 2N^2 \end{bmatrix} \begin{bmatrix} a_{-N}^j \\ \vdots \\ \vdots \\ \vdots \\ a_N^j \end{bmatrix} = 2\pi\lambda_j \begin{bmatrix} a_{-N}^j \\ \vdots \\ \vdots \\ \vdots \\ a_N^j \end{bmatrix}. \tag{2.34}$$

This is a standard eigenvalue problem in numerical algebra. Suppose $\tilde{\lambda}_j$ and

$$(a_{-N}, \ldots, a_N)$$

is the eigenvalue and associated eigenvector of the matrix in (2.34), then $\lambda_j = \tilde{\lambda}_j/2$ is the approximated eigenvalue in (2.27), and (2.29) provides the associated eigenfunction, which is equivalent to a real eigenfunction up to a complex constant.

2.5 Hyperparameters in the numerical examples

We first report hyperpameters commonly used in all three numerical examples and then list in Tables 2.1-2.3 those specific to the examples. In all three examples, the order of the trigonometric basis $M = 5$, the constant $Z_0 = 2$ for regularizing the normalization factor, the weight parameters in the loss $\eta_1 = 1000, \eta_2 = 20, \eta_3 = 100$. In the following tables, the structures of the neural networks are represented by vectors, whose elements denote the number of nodes within each layer. The learning rate and moving average decay γ_ℓ are both piecewise constant, whose values and boundaries are given separately. For example, in 5-dimensional Fokker-Planck problem, the learning rate is 1×10^{-4} for the first 30000 steps, 5×10^{-5} from the 30001-st to the 60000-th step and 1e-5 after the 60000-th step. The moving average decay γ_ℓ is defined similarly, with the same boundaries.

We remark that the choice of the terminal time T is a trade-off between discretization errors and training errors. For a fixed number of time steps N, a large T will result in large discretization errors. On the other hand, if a small T is used, the discrepancy between $\mathcal{P}_T^\lambda \Psi$ and Ψ is less significant when a wrong Ψ is used (think about the extreme case that $T = 0$, any Ψ would give 0 loss), which causes difficulty in the optimization of the parameters.

We also remark that the choice of the size of the neural networks is a trade-off between the approximation accuracy and computational cost. Three is chosen as a moderate depth while the widths are chosen to guarantee enough approximation capability. We choose ReLU as the activate function to save the computation cost of backpropagation in the calculation of derivatives, without a sacrifice of accuracy. The learning rates are chosen to be non-increasing piecewise constant according to common practice.

Table 2.1: Parameters for Fokker-Planck.

Parameters	$d = 5$	$d = 10$
terminal time T	0.2	0.2
number of time intervals N	80	120
structure of neural networks	[80, 80, 80]	[300, 300, 300]
number of iterations	80000	100000
learning rate	$[1 \times 10^{-4}, 5 \times 10^{-5}, 1 \times 10^{-5}]$	$[5 \times 10^{-5}, 2 \times 10^{-5}, 1 \times 10^{-5}]$
moving average decay γ_ℓ	[0.2, 0.5, 0.9]	[0.2, 0.5, 0.9]
piecewise constant boundaries	[30000,60000]	[60000, 80000]
batch size K	1024	1024

Table 2.2: Parameters for linear Schrödinger.

Parameters	$d = 5$	$d = 10$
terminal time T	0.3	0.3
number of time intervals N	80	120
structure of neural networks	[80, 80, 80]	[300, 300, 300]
number of iterations	80000	80000
learning rate	$[1 \times 10^{-4}, 5 \times 10^{-5}, 1 \times 10^{-5}]$	$[5 \times 10^{-5}, 5 \times 10^{-5}, 1 \times 10^{-5}]$
moving average decay γ_ℓ	[0.2, 0.5, 0.9]	[0.2, 0.5, 0.9]
piecewise constant boundaries	[30000, 60000]	[40000, 60000]
batch size K	1024	1024

Table 2.3: Parameters for nonlinear Schrödinger.

Parameters	$d = 5$	$d = 10$
terminal time T	0.2	0.3
number of time intervals N	120	200
structure of neural networks	[80, 80, 80]	[300, 300, 300]
number of iterations	60000	80000
learning rate	$[5 \times 10^{-5}, 2 \times 10^{-5}, 1 \times 10^{-5}]$	$[5 \times 10^{-5}, 2 \times 10^{-5}, 1 \times 10^{-5}]$
moving average decay γ_ℓ	[0.2, 0.9, 0.99]	[0.1, 0.9, 0.99]
piecewise constant boundaries	[20000, 40000]	[40000, 60000]
batch size K	2048	2048

2.6 Conclusion and future works

In this paper, we propose a new method to solve eigenvalue problems in high dimensions using neural networks. Our method is able to compute both eigenvalues and corresponding eigenfunctions (with gradients) with high accuracy.

There are several natural directions for future work. First, to apply our methodol-

Table 2.4: Parameters for well-separated linear Schrödinger.

state	first eigenpair	second eigenpair
terminal time T	0.2	0.2
number of time intervals N	320	320
structure of neural networks	[200, 200, 200]	[200, 200, 200]
number of iterations	50000	50000
learning rate	$[5 \times 10^{-4}, 1 \times 10^{-4}, 1 \times 10^{-5}]$	$[5 \times 10^{-4}, 1 \times 10^{-4}, 1 \times 10^{-5}]$
moving average decay γ_ℓ	[0.1, 0.2, 0.9]	[0.1, 0.9, 0.99]
piecewise constant boundaries	[30000, 40000]	[30000, 40000]
batch size K	2048	2048

Table 2.5: Parameters for degenerate linear Schrödinger.

state	first eigenpair	second eigenpair
terminal time T	0.2	0.2
number of time intervals N	80	80
structure of neural networks	[40, 40]	[40, 40]
number of iterations	6000	6000
learning rate	$[5 \times 10^{-4}, 1 \times 10^{-4}, 1 \times 10^{-5}]$	$[5 \times 10^{-4}, 1 \times 10^{-4}, 1 \times 10^{-5}]$
moving average decay γ_ℓ	[0.1, 0.2, 0.9]	[0.1, 0.2, 0.9]
piecewise constant boundaries	[2000, 4000]	[2000, 4000]
batch size K	512	512

ogy to quantum many-body systems, we need to respect the permutation symmetry in our ansatz for the wavefunctions. Previous works [HZE19, CMC20, HSN20, PSMF20] have proposed various flexible neural-network ansatz to incorporate in the symmetry, which can be combined with our approach. Moreover, in DMC, importance sampling techniques are often essential to improve the accuracy. In our context, this means to choose a better underlying diffusion process guided by a trial wavefunction depending on the problem. Last, the scalability of the method has to be tested on larger systems beyond the toy numerical examples in this work.

On the theoretical aspects, the understanding of the stability and convergence of the proposed method is a fascinating future direction. While the general analysis might be quite difficult given the highly nonlinear approximation induced by the neural networks and also the complicated optimization strategy, some perturbative analysis, especially in the linearized regime, might be possible. We will leave these

to future works.

Chapter 3

Actor-Critic Method for High Dimensional Static Hamilton–Jacobi–Bellman Partial Differential Equations Based on Neural Networks

3.1 Introduction

The Hamilton-Jacobi-Bellman (HJB) equation is an important family of partial differential equations (PDEs), given its connection with optimal control problems that lead to a wide range of applications. The unknown in the HJB equation can be viewed as the total expected value function for optimal control problems. The equation can be derived from the dynamic programming principle pioneered by Bellman [Bel66], which gives a necessary and sufficient condition of the optimality. Theoretical results for the existence and uniqueness of the HJB equations are well established; see, e.g., [YZ99]. From the viewpoint of stochastic control, the relationship between the viscosity solution of the HJB equations and the backward stochastic differential equations (BSDEs) is introduced in [CIL92, PP90, Pen91, PP92, Par98].

The wide applications of HJB equations call for efficient numerical algorithms. Various numerical approaches have been developed in the literature, including the monotone approximation scheme [BJ02, FL07], the finite volume method [WJT03, RW06], and the Galerkin method [BSW97, BSW98]. In [OS88], non-oscillatory schemes are developed to solve the HJB equations exploring the connection with hyperbolic conservation laws. The HJB equations related to reachability problems

are studied in [MT03, MBT05, Lyg04]. A general survey for classical methods to solve the optimal control problem numerically can be found, e.g., in [Rao09]. While these conventional approaches have been quite successful, they fall short for solving HJB equations in high dimensions due to the curse of dimensionality [Bel66]: the computational cost goes up exponentially with the dimensionality. Many works attempt to mitigate this fundamental difficulty by leveraging dimension reduction techniques such as proper orthogonal decomposition, sparse grid, pseudospectral collocation, and tensor decomposition (see e.g., [KVX04, KW17, KK18, DKK21, OSS19]). The performance of these algorithms heavily depends on how well the low dimensional representation matches the solutions, and is typically problem dependent and thus with limited applicability.

To better address the challenge of high dimensionality, a promising direction is to consider the artificial neural network as a more flexible and efficient function approximation tool. This topic has received a considerable amount of attention and been a rapidly developing field in recent years. Several numerical approaches for high dimensional PDEs based on neural network parametrization have been proposed; see e.g., the reviews [WHJ21, BHJK20] and references therein.

For HJB type equations and related optimal control problems, the most tightly connected approach to our work is the deep BSDE method [HJE18, EHJ17], which reformulates parabolic PDEs as control problems using BSDEs, and uses deep neural network parametrization for the solution and control to solve this problem. Theoretical results for convergence of this method are studied in [HL20]. The deep BSDE method and its variants have been applied to solve HJB type equations, stochastic control problems, and differential games (see e.g., [HJE18, EHJ17, CWNMW19, HL17, PWET19, PWG21, NR21, KSS20b, JPPZ20, HH20, HHL20]). Numerical algorithms for solving high dimensional deterministic and stochastic control problems

based on other forms combined with deep learning approximation have also been investigated in [EH16, NZGK21, BCJ19, HH21].

While some methods mentioned above have been successful in solving PDEs in high dimensions, there have been two issues that remain to be addressed. On the one hand, most of these works concern parabolic PDEs of finite time horizon (often of order one), while only a few works investigate the static elliptic HJB equations corresponding to control problems with infinite time horizon. On the other hand, most existing works consider equations where the optimal controls are explicitly known given the value function or without controls, while there are many important HJB type equations for which the optimal control is cast through an optimization problem and hence implicit. Recently, an algorithm for a high dimensional finite-time horizon stochastic control problem with implicit optimal control is considered in [JPPZ22], based on the deep BSDE formulation associated with the stochastic maximum principle. In this paper, we take a different approach and focus on solving the static elliptic type HJB equation with implicit control, in which the above two challenges are compounded.

Our proposed numerical method is heavily inspired by the literature on reinforcement learning (RL) [SB18], which is of course closely related to control problems. Our motivation for borrowing techniques from RL is due to the impressive revolution and great success in recent years in deep RL by utilizing neural network parametrization [MKS$^+$13, SHM$^+$16, DCH$^+$16]. In the RL context, the control problem is usually formulated as a Markov decision process (MDP) on discrete time and state space. If the model is given, finding the optimal policy can be viewed as solving a discrete HJB equation. It is then natural to ask whether algorithms developed in the RL context can be generalized to the context of solving high dimensional HJB equations.

In this paper, we reformulate the HJB type fully nonlinear elliptic PDEs into

stochastic control problems and leverage the actor-critic framework in conjunction with a neural network approximation to solve the equations. The actor-critic methods are a class of algorithms in RL [SB18]. These algorithms iteratively evaluate and improve the current policies (i.e., controls) until final convergence. The critic refers to the value function of a given policy. The process of estimating the critic is called policy evaluation. The most common algorithms for policy evaluation are temporal difference (TD) methods [KT00, BSGL09, VL10], or their variants, such as the TD(λ) [DWS12] and the least-squares TD (LSTD) [PS08, MSBS10, Boy99]. The actor refers to the policy function, and we need to make policy improvement based on a given value function. In this case, the most popular method is policy gradient [KT00, BA06, AB07, DWS12, WBH$^+$16] and their variants, such as natural policy gradient [PS08, BSGL09]. In this work, we propose a variance reduced version of the LSTD method for policy evaluation derived using stochastic calculus. We also adapt the policy gradient method for policy improvement to the continuous-time stochastic control problem.

3.2 Theoretical background for actor-critic

3.2.1 Control formulation of elliptic equations

Consider the following fully nonlinear elliptic PDE

$$\inf_{u\in U}\left[\frac{1}{2}\operatorname{Tr}\left(\sigma\sigma^\top \operatorname{Hess}(V)\right)(x,u)+b(x,u)^\top\nabla V(x)+f(x,u)\right]-\gamma V(x)=0 \quad \text{in } \Omega, \tag{3.1}$$

with boundary condition $V(x) = g(x)$ on $\partial\Omega$. Here the state space Ω is an open, connected set in $\mathbb{R}^d$ with piecewise smooth boundary, and the control space U is a convex closed domain in $\mathbb{R}^{d_u}$. We assume that $V(x) \in C^2(\overline{\Omega})$, $f(x,u) \in C(\overline{\Omega} \times U)$, $b(x,u) \in C(\overline{\Omega} \times U; \mathbb{R}^d)$, $\sigma(x,u) \in C(\overline{\Omega} \times U; \mathbb{R}^{d\times d_w})$ with $\sigma(x,u)\sigma^\top(x,u)$ being

uniformly elliptic and bounded, and $\gamma \geq 0$ is a constant. Here and in the following, we use ∇ and Hess to denote the gradient and Hessian operators.

As a starting point of our approach, we reformulate the above elliptic equation as an optimal control problem. Let $(\widetilde{\Omega}, \mathcal{F}, \{\mathcal{F}_t\}_{t\geq 0}, \mathbf{P})$ be a filtered probability space. Consider the following stochastic differential equation (SDE)

$$\mathrm{d}X_t = b(X_t, u_t)\,\mathrm{d}t + \sigma(X_t, u_t)\,\mathrm{d}W_t \tag{3.2}$$

with initial condition $X_0 = x \in \Omega$, where $u_t \in U \subset \mathbb{R}^{d_u}$ is an $\mathcal{F}_t$-adapted control field and W_t is a d_w-dimensional $\mathcal{F}_t$-standard Brownian motion. As we solve the equation in the domain Ω, we define a stopping time

$$\tau = \inf\{t : X_t \notin \Omega\}. \tag{3.3}$$

It is a standard result that $\tau < \infty$ a.s.; see, for example, [Kle05].

We then consider an optimal control problem to minimize the following cost functional

$$J^u(x) = \mathbb{E}\Big[\int_0^\tau f(X_s, u_s)e^{-\gamma s}\,\mathrm{d}s + e^{-\gamma\tau}g(X_\tau) \mid X_0^u = x\Big]. \tag{3.4}$$

In this cost functional, f can be interpreted as running cost, g is the terminal cost when the SDE hits the boundary $\partial\Omega$, and γ is the discount rate.

The control u is chosen over the set of stochastic processes that have values in U and are adapted to the filtration $\mathcal{F}_t$. Define

$$V(x) = \inf_u J^u(x) \tag{3.5}$$

as the optimal value function (i.e., optimal cost-to-go function). According to standard results in stochastic control theory [YZ99], V satisfies the time-independent HJB equation

$$\inf_u \big\{\mathcal{L}^u V(x,u) + f(x,u) - \gamma V(x)\big\} = 0 \tag{3.6}$$

in Ω with boundary condition $V(x) = g(x)$ on $\partial\Omega$, where

$$\mathcal{L}^u V(x) = \frac{1}{2} \operatorname{Tr}\left(\sigma\sigma^\top \operatorname{Hess}(V)\right)(x,u) + b(x,u)^\top \nabla V(x)$$

is the generator of the controlled SDE (3.2). Note that the HJB equation (3.6) coincides with the original PDE (3.1) and, hence, we can solve the PDE (3.1) by solving the optimal control problem to obtain the optimal value function.

3.2.2 Actor-critic method in stochastic optimal control problem

Our approach for solving the optimal control problem is based on the actor-critic framework. In such methods, one solves for both the value function and control field. The control (i.e., policy in the RL terminology) is known as the *actor*, while the value function corresponding to the control is known as the *critic* since it is used to evaluate the optimality of the control. Accordingly, the actor-critic algorithms consist of two parts: policy evaluation for the critic and policy improvement for the actor. While many approaches have been developed under the actor-critic framework [KT00, AB07, PS08, BSGL09, VL10, DWS12, WBH$^+$16], we will focus on simple and perhaps the most popular algorithms: TD learning for the value function given a policy and policy gradient for improving the control.

TD for discrete Markov decision processes

To better convey the idea, let us first briefly recall the algorithms for the discrete-time MDP with finite state and action space; more details can be found in e.g., [SB18]. The MDP starts with some initial state S_0 in the state space $\mathcal{S}$, possibly sampled according to a distribution. At time $t \in \mathbb{N}$, given the current state S_t, the agent picks an action A_t in the action set $\mathcal{A}$ according to a policy. We assume that the policy is

deterministic, i.e., the policy is a map π from the state space $\mathcal{S}$ to the action space $\mathcal{A}$:

$$A_t = \pi(S_t). \tag{3.7}$$

After the action A_t is chosen, the system state will transit to S_{t+1}, according to a probability transition function

$$\mathbb{P}(S_{t+1} = s' \mid S_t = s, A_t = a) = p(s' \mid s, a). \tag{3.8}$$

The action also incurs a cost R_{t+1}, which we assume to be given by a deterministic function of the previous state S_t, action A_t, and the current state S_{t+1}:

$$R_{t+1} = f(S_t, A_t, S_{t+1}). \tag{3.9}$$

The goal of the MDP problem is to choose the best policy to minimize the expected total discounted cost

$$\mathbb{E}_{S_0 \sim \mu, \pi} \Big[\sum_{t=1}^{\infty} \beta^{t-1} R_t \Big], \tag{3.10}$$

where $\beta \in (0, 1)$ is a discount factor, μ is the distribution of the initial state S_0, and we have used $\mathbb{E}_\pi$ to indicate the dependence on the transition dynamics on the choice of the policy π.

To solve the MDP problem, it is convenient to introduce the (state) value function w.r.t. a policy π as the expected cost starting at states s under that policy:

$$V^\pi(s) = \mathbb{E}_\pi \Big[\sum_{t=1}^{\infty} \beta^{t-1} R_t \mid S_0 = s \Big]. \tag{3.11}$$

By the dynamic programming principle [Bel66], for any given policy π, the value function satisfies

$$V^\pi(s) = \mathbb{E}_\pi \Big[\sum_{t=1}^{n} \beta^{t-1} R_t + \beta^n V^\pi(S_n) \mid S_0 = s \Big] \tag{3.12}$$

for any $n \geq 1$. In order to minimize the total cost (3.10), we search for an optimal policy π^* that satisfies for all π,

$$V^{\pi^*}(s) \leq V^{\pi}(s) \qquad \forall\, s \in \mathcal{S}. \tag{3.13}$$

Specifically, by the optimality principle, we have

$$V^{\pi^*}(s) = \min_{\{a_t\} \subset \mathcal{A}} \mathbb{E}\left[\sum_{t=1}^{n} \beta^{t-1} R_t + \beta^n V^{\pi^*}(S_n) \mid S_0 = s, A_t = a_t, t = 0, 1, \cdots, n-1\right]. \tag{3.14}$$

Note that while in (3.12) and (3.14) the right-hand side starts at time 0, we can start at any time and run the process for n steps due to stationarity.

Let us make a couple of remarks for the setup of the discrete MDP used here. First, in RL, reward is usually used instead of cost and, hence, one maximizes the total reward instead of minimizing the cost; evidently, the two viewpoints are equivalent up to a change of sign. We use cost, which is more in line with the control literature and also our problem in the continuous setting. Second, the cost R_t is not necessarily a deterministic function as in (3.9), but may follow some probability distribution together with the next state:

$$\mathbb{P}(S_{t+1} = s', R_{t+1} = r \mid S_t = s, A_t = a) = p(s', r \mid s, a). \tag{3.15}$$

Moreover, the policy can also be probabilistic rather than deterministic as assumed in (3.7). We choose the simplified setting for the cost and policy to make it consistent with our continuous optimal control setting. Finally, we use an MDP without stopping time and thus without terminal cost for simplicity. The adaptation to our PDE setup will be discussed below in Sections 3.2.2 and 3.2.2.

TD learning is a class of algorithms that evaluate a given control (i.e., policy) by updating the value function, combining a Monte Carlo estimate of the running cost

over a time period and the dynamic programming principle for the future cost-to-go. For a policy π to be evaluated, with a given trajectory $\{S_t, t \geq 0\}$, we update the value function at each t by

$$\widehat{V}^{\pi}(S_t) \longleftarrow \widehat{V}^{\pi}(S_t) + \alpha \left(\sum_{k=1}^{n} \beta^{k-1} R_{t+k} + \beta^n \widehat{V}^{\pi}(S_{t+n}) - \widehat{V}^{\pi}(S_t) \right), \tag{3.16}$$

where α is the learning rate and $\widehat{V}^{\pi}$ on the right-hand side is the current estimate of the value function. In (3.16), we only update the value function at the state S_t and the value at other states remain unchanged. In practice, this update of the value function is usually done for multiple trajectories.

In the above updating rule, we have used the n-step TD $\mathrm{TD}_n^{\pi}(S_t)$, defined as

$$\mathrm{TD}_n^{\pi}(S_t) = \sum_{k=1}^{n} \beta^{k-1} R_{t+k} + \beta^n \widehat{V}^{\pi}(S_{t+n}) - \widehat{V}^{\pi}(S_t), \tag{3.17}$$

which depends on the trajectory of length $n+1$ (t to $t+n$) from the starting state S_t. TD_n^{π} can be understood as an indicator of the inconsistency between the current estimate of the value function with a sampled value using n-steps of the MDP, since according to (3.12), $\mathbb{E}_{\pi}\mathrm{TD}_n^{\pi}$ vanishes if $\widehat{V}^{\pi}$ agrees with the true value function V^{π}. Hence, TD learning can be viewed as a stochastic fixed point iteration for the value function.

When function approximation is used for the value function, in particular nonlinear approximations such as neural networks, an alternative approach, the LSTD is often used to overcome potential divergence problems of TD learning [PS08, Boy99]. Instead of the stochastic fixed point updating formula as (3.16), in the LSTD method, the parameters are optimized to minimize the squares of the TD error as a loss function. More specifically, if the value function is parametrized as $V^{\pi}(\cdot; \theta_V)$, we solve

for θ_V by

$$\min_{\theta_V} \mathbb{E}_{S_0 \sim \mu, \pi} \Big[\Big(\sum_{t=1}^{n} \beta^{t-1} R_t + \beta^n V^\pi(S_n; \theta_V) - V^\pi(S_0; \theta_V) \Big)^2 \mid S_0 \Big], \tag{3.18}$$

where μ is some initial distribution for the state S_0. In practice, (3.18) is often solved using the stochastic gradient descent method. Such method has been proved successful in e.g., [MKS+13, DCH+16].

Policy gradient for discrete MDP

Policy gradient is a class of methods to learn parametrized policies through gradient based algorithms. Assume we consider a class of (deterministic) policy parametrized as

$$A_t(S_t) = \pi(S_t; \theta_\pi), \tag{3.19}$$

where θ_π denotes a collection of parameters and $\pi(\cdot; \theta)$ is some chosen nonlinear parametrization.

To find an optimal policy, we aim to minimize the objective function (cf. (3.14))

$$J(\theta_\pi) = \mathbb{E}_{S_0 \sim \mu, \pi(\cdot; \theta_\pi)} \big[\sum_{k=1}^{n} \beta^{k-1} R_k + \beta^n \widehat{V}^\pi(S_n) \big] \tag{3.20}$$

w.r.t. the collective parameter θ_π. Note that (3.20) explicitly takes into account the cost of the first n steps, while using an (approximate) value function $\widehat{V}^\pi$ for the future cost after n steps, coming from e.g., the TD learning algorithm. The objective function (3.20) can be optimized using stochastic gradient method. Using a stochastic estimate of the gradient $\widehat{\nabla J} \approx \nabla_{\theta_\pi} J$, so that the parameter is updated as

$$\theta_\pi \leftarrow \theta_\pi - \alpha \widehat{\nabla J} \tag{3.21}$$

with suitable learning rate α. Note that in principle, one needs to differentiate all terms involved in (3.20) w.r.t. θ_π; however, in practice, in the actor-critic framework,

one typically leaves out the derivative of $\widehat{V}^\pi$ w.r.t. π, as it is impractical to compute since $\widehat{V}^\pi$ is obtained using e.g., TD learning. Nevertheless, we would still need to differentiate $\widehat{V}^\pi(S_n)$ w.r.t. S_n, as the state S_n is affected by the choice of the policy, thus

$$\frac{\partial \widehat{V}^\pi(S_n)}{\partial \theta_\pi} \doteq \frac{\partial \widehat{V}^\pi}{\partial S_n}\frac{\partial S_n}{\partial \theta_\pi},$$

where $\doteq$ indicates that the (functional) derivative $\frac{\delta \widehat{V}^\pi}{\delta \pi}$ is omitted. Dropping this term is often applied in actor-critic algorithms. Some justifications can be found in [DWS12]: Under certain conditions, the approximated gradient is still in the direction of improving the performance and the set of critical points of the objective function coincides with the set of zero approximated gradients.

Since we consider deterministic policies, the policy gradient approach discussed above is in the same spirit as the *deterministic policy gradient algorithm* proposed in [SLH+14]. One difference is that we use the state value function $V(s)$ while [SLH+14] uses the state-action value function (Q-function) $Q(s,a)$. Another difference is that the objective function used in [SLH+14] is based on the stationary distribution of a state-action pair given the policy, while we roll out a trajectory for the cost function (combined with using an estimated value function for future cost), which is more suitable to an actor-critic framework. Our approach is also easier to generalize to the continuous setting, which will be discussed in Section 3.2.2.

TD for continuous optimal control problems

We now introduce how to adapt the above algorithmic ideas to the continuous setting.

Given a control function $u(x) \in C(\overline{\Omega})$ (which corresponds to π in the discrete setting), the corresponding value function is given by

$$V^u(x) = \mathbb{E}_u\Big[\int_0^\tau f(X_s, u(X_s))e^{-\gamma s}\,\mathrm{d}s + e^{-\gamma\tau}g(X_\tau) \mid X_0 = x\Big]. \tag{3.22}$$

Here $\mathbb{E}_u$ indicates expectation w.r.t. the trajectory (with a fixed policy u). This is just the cost functional in (3.4) with a specific control policy. In the continuous setting, the dynamical programming principle indicates that the value function V^u satisfies the PDE (see e.g., [YZ99])

$$\frac{1}{2}\operatorname{Tr}\left(\sigma\sigma^\top \operatorname{Hess}(V^u)\right)(x,u(x)) + b(x,u(x))^\top \nabla V^u(x) + f(x,u(x)) - \gamma V^u(x) = 0 \,\text{in}\, \Omega \tag{3.23}$$

with boundary condition $V^u(x) = g(x)$ on $\partial\Omega$.

To better convey the idea, we first consider a fixed time interval $[0,T]$ with $T > 0$ and neglect the stopping time and also the domain boundary. Necessary modifications regarding the stopping time and boundary will be explained below. Applying Itô's formula to $e^{-\gamma t}V^u(X_t)$, we get

$$\begin{aligned} e^{-\gamma T}V^u(X_T) = V^u(X_0) + \int_0^T e^{-\gamma s}\Big[\frac{1}{2}\operatorname{Tr}\big(\sigma\sigma^\top \operatorname{Hess}(V^u)\big)(X_s,u(X_s)) \\ + b(X_s,u(X_s))^\top \nabla V^u(X_s) - \gamma V^u(X_s)\Big]\,\mathrm{d}s \\ + \int_0^T e^{-\gamma s}\nabla V^u(X_s)^\top \sigma(X_s,u(X_s))\,\mathrm{d}W_s. \end{aligned} \tag{3.24}$$

Combined with the PDE (3.23), (3.24) gives

$$\begin{aligned} V^u(X_0) = \int_0^T e^{-\gamma s} f(X_s,u(X_s))\,\mathrm{d}s \\ - \int_0^T e^{-\gamma s}\nabla V^u(X_s)^\top \sigma(X_s,u(X_s))\,\mathrm{d}W_s + e^{-\gamma T}V^u(X_T). \end{aligned} \tag{3.25}$$

The term $\int_0^T e^{-\gamma s}\nabla V^u(X_s)^\top \sigma(X_s,u(X_s))\,\mathrm{d}W_s$ is a martingale w.r.t. T because ∇V and σ are bounded according to our assumptions [Kle05]. Therefore, taking the

expectation, we arrive at

$$V^u(X_0) = \mathbb{E}_u \left[\int_0^T e^{-\gamma s} f(X_s, u(X_s)) \, \mathrm{d}s + e^{-\gamma T} V^u(X_T) \mid X_0 \right]. \tag{3.26}$$

We observe that this is the analog of (3.12) in the continuous setting, where the unit time discount $e^{-\gamma}$ is the analog of the discount factor β in the discrete setting. Compared with the discrete time setting, besides (3.26), we have in addition (3.25) before taking expectation, thanks to Itô's lemma. Exploiting the two identities, analogously to the discrete case, we define two versions of TD in the continuous setting as

$$\begin{aligned} \mathrm{TD}_1^u = \int_0^T e^{-\gamma s} f(X_s, u(X_s)) \, \mathrm{d}s - \int_0^T e^{-\gamma s} \nabla V(X_s)^\top \sigma(X_s, u(X_s)) \, \mathrm{d}W_s \\ + e^{-\gamma T} V(X_T) - V(X_0), \end{aligned} \tag{3.27}$$

$$\mathrm{TD}_2^u = \int_0^T e^{-\gamma s} f(X_s, u(X_s)) \, \mathrm{d}s + e^{-\gamma T} V(X_T) - V(X_0). \tag{3.28}$$

Note that both TD_1 and TD_2 depend on the trajectory X_t; in particular, they should be viewed as random variables, while we suppress such dependence in the notation. From (3.25) and (3.26), if V is the exact value function corresponding to the control u, we have

$$\mathrm{TD}_1^u = 0, \quad \mathbb{P}\text{-a.s.} \tag{3.29}$$

$$\mathbb{E}_u \, \mathrm{TD}_2^u = 0. \tag{3.30}$$

Note in particular that TD_1^u vanishes without taking the expectation for the exact value function while $\mathrm{Var}(\mathrm{TD}_2^u) = \mathbb{E}_u[\int_0^T e^{-2\gamma s} |\nabla V(X_s)^\top \sigma(X_s, u(X_s))|^2 \, \mathrm{d}s] > 0$ if $\nabla V^\top(x)\sigma(x,u) \not\equiv 0$. Moreover, as the difference between TD_1 and TD_2 is given by a martingale term, for any approximate value function, we have

$$\mathbb{E}_u \mathrm{TD}_1^u = \mathbb{E}_u \mathrm{TD}_2^u.$$

Now let us introduce two loss functionals for the critic in the spirit of LSTD:

$$L_1(V) = \mathbb{E}_{X_0\sim\mu,u}\left(\mathrm{TD}_1^u\right)^2 \tag{3.31}$$
$$= \mathbb{E}_{X_0\sim\mu,u}\Big[\Big(\int_0^{T\wedge\tau} e^{-\gamma s} f(X_s, u(X_s))\,\mathrm{d}s$$
$$- \int_0^{T\wedge\tau} e^{-\gamma s}\nabla V(X_s)^\top \sigma(X_s, u(X_s))\,\mathrm{d}W_s + e^{-\gamma(T\wedge\tau)}V(X_{T\wedge\tau}) - V(X_0)\Big)^2\Big],$$

$$L_2(V) = \mathbb{E}_{X_0\sim\mu,u}\left(\mathrm{TD}_2^u\right)^2 \tag{3.32}$$
$$= \mathbb{E}_{X_0\sim\mu,u}\Big[\Big(\int_0^{T\wedge\tau} e^{-\gamma s} f(X_s, u(X_s))\,\mathrm{d}s + e^{-\gamma(T\wedge\tau)}V(X_{T\wedge\tau}) - V(X_0)\Big)^2\Big],$$

where μ is some initial distribution for X_0 and we have also taken into account the stopping time τ when the process hits the domain boundary. Here the two losses are viewed as functionals of the value function V, the finite-dimensional function approximation will be discussed in the next section.

The stochastic gradient method is used to minimize the loss function in LSTD to find the best approximation of the value function. Written in terms of functional variations, this amounts to approximating

$$\mathbb{E}_{X_0\sim\mu,u}\frac{\delta\big(\mathrm{TD}_1^u\big)^2}{\delta V} \approx \frac{\delta\big(\mathrm{TD}_1^u\big)^2}{\delta V}\big(X_t\big) = 2\mathrm{TD}_1^u\big(X_t\big)\frac{\delta \mathrm{TD}_1^u}{\delta V}\big(X_t\big), \tag{3.33}$$

$$\mathbb{E}_{X_0\sim\mu,u}\frac{\delta\big(\mathrm{TD}_2^u\big)^2}{\delta V} \approx \frac{\delta\big(\mathrm{TD}_2^u\big)^2}{\delta V}\big(X_t\big) = 2\mathrm{TD}_2^u\big(X_t\big)\frac{\delta \mathrm{TD}_2^u}{\delta V}\big(X_t\big), \tag{3.34}$$

where we evaluate the right-hand side term on a single realization of the trajectory to the ease the notation. In our numerical implementation, we use multiple trajectories to further improve the computation efficiency. As we remark above, since (3.25) holds true without taking the expectation, the right-hand side of (3.33) thus vanishes for the exact value function for any realization of X_t, in particular, the variance of the stochastic gradient is 0. In comparison, while the stochastic estimate of (3.34) has the expectation 0 for the exact value function, for each trajectory, the right-hand

side is not 0. This means that the stochastic gradient estimate (3.34) has a larger variance than the estimate (3.33). Let us remark that the vanishing variance property of (3.33) is similar to quantum Monte Carlo [FMNR01], for which the variance of the local energy estimate also vanishes at the ground state.

In the following, to distinguish the two loss functions, we call the method based on L_1 (3.31) the variance reduced LSTD (VR-LSTD), while that corresponding to L_2 (3.32) is named the LSTD. We will demonstrate in our numerical experiments that VR-LSTD gives better results than LSTD.

Policy gradient for continuous optimal control problems

For the actor part, we use policy gradient to improve the policy. According to the dynamical programming principle [YZ99], for the optimal value function V, we have

$$V(X_0) = \inf_u \mathbb{E}_u \Big[\int_0^T f(X_s, u(X_s)) e^{-\gamma s} \, \mathrm{d}s + e^{-\gamma T} V(X_T) \mid X_0 \Big], \tag{3.35}$$

where u is minimized over the set of admissible controls. In other words, the control u should minimize the functional on the right-hand side. Therefore, we can use the following loss function for the actor, for which we also incorporate the stopping time:

$$J(u) = \mathbb{E}_{X_0 \sim \mu, u} \Big[\int_0^{T \wedge \tau} f(X_s, u(X_s)) e^{-\gamma s} \, \mathrm{d}s + \widehat{V}(X_{T \wedge \tau}) e^{-\gamma (T \wedge \tau)} \Big], \tag{3.36}$$

where $\widehat{V}$ is the current estimate of the value function (via TD learning in the critic part). Observe that this loss function is a continuous analog of (3.20).

In the numerical algorithm, the control, as a high dimensional function, will be parametrized as a neural network $u(\cdot; \theta_u)$, where θ_u denotes collectively the parameters. The parameters are optimized using a stochastic approximation to gradients of $J(u)$. Similarly to our discussion of policy gradient for the discrete case in Section 3.2.2, when differentiating the loss function (3.36) w.r.t. the parameters of the

control θ_u, several terms would contribute to the derivative, including the control $u(\cdot)$ itself, the SDE trajectory $X_\cdot$, the stopping time τ, and also the estimated value function $\widehat{V}(\cdot)$. Similarly to the discrete case, we will drop the functional derivative of $\widehat{V}$ w.r.t. u, i.e., the derivative $\frac{\delta \widehat{V}}{\delta u}\frac{\partial u}{\partial \theta_u}$, since the dependence of $\widehat{V}$ on u is through the algorithm for the critic, e.g., the TD learning, which is impractical to track. Furthermore, if $\hat{V}$ is the optimal value function, treating it as a fixed function and optimizing u in (3.36) gives the optimal policy function. Therefore, we approximate the functional derivative as

$$\frac{\delta J}{\delta u} \doteq \mathbb{E}_{X_0\sim\mu,u}\Bigg[\int_0^{T\wedge\tau} \frac{\delta f(X_s,u(X_s))}{\delta u} e^{-\gamma s}\,\mathrm{d}s + \mathbb{1}_{\{\tau<T\}} f(X_\tau,u(X_\tau))e^{-\gamma\tau}\frac{\delta\tau}{\delta u} + \nabla\widehat{V}(X_{T\wedge\tau})e^{-\gamma(T\wedge\tau)}\frac{\delta X_s}{\delta u}\bigg|_{s=T\wedge\tau} + \mathbb{1}_{\{\tau<T\}}(\mathcal{L}^u-\gamma)\widehat{V}(X_\tau)e^{-\gamma\tau}\frac{\delta\tau}{\delta u}\Bigg], \quad (3.37)$$

where $\doteq$ indicates that we leave out the contribution from the functional derivative of $\widehat{V}$ w.r.t. u and we have

$$\frac{\delta f(X_s,u(X_s))}{\delta u} = \frac{\partial f}{\partial x}\frac{\delta X_s}{\delta u} + \frac{\partial f}{\partial u}\left(\mathrm{Id} + \nabla u(X_s)\frac{\delta X_s}{\delta u}\right). \quad (3.38)$$

To obtain the formula, we have used Itô's lemma to rewrite

$$\mathbb{E}_{X_0\sim\mu,u}[\widehat{V}(X_{T\wedge\tau})e^{-\gamma(T\wedge\tau)}] = \mathbb{E}_{X_0\sim\mu,u}\Big[\widehat{V}(X_0) + \int_0^{T\wedge\tau}(\mathcal{L}^u-\gamma)\widehat{V}(X_s)\,\mathrm{d}s\Big], \quad (3.39)$$

and taken the derivative of the right-hand side w.r.t. u.

3.3 Numerical algorithm

In this section, we present our numerical algorithm for solving high dimensional HJB type elliptic PDEs based on the actor-critic framework discussed in the previous section.

3.3.1 Function approximation

In order to numerically deal with the high dimensional functions V and u, we use two neural networks to parametrize the value function $V(\cdot\,;\theta_V)$ and the control $u(\cdot\,;\theta_u)$, the parameters of which are denoted collectively by θ_V and θ_u, respectively. We apply the structure of the residual neural network [HZRS16] in pursuit of better optimization performance. A neural network $\phi(x;\theta)$ with l hidden layers is represented by

$$\phi(x;\theta) = F_l \circ \sigma_l \circ F_{l-1} \circ \sigma_{l-1} \circ \cdots \circ F_1 \circ \sigma_1 \circ F_0(x), \tag{3.40}$$

where F_i are linear transforms with dimensions depending on the width of hidden layers and the dimensions of inputs and outputs, and σ_i are elementwise activate functions with skip connection: $\sigma_i(x) = x + \mathrm{ReLU}(x)$.

Moreover, note that the VR-LSTD loss function L_1 (3.31) requires the gradient of the value function. Since we are using a neural network parametrization $V = V(\cdot\,;\theta_V)$, a direct approach is to use autodifferentiation of $V(x;\theta_V)$ w.r.t. x to calculate the gradient. We find that a better approach in practice is to use another neural network to represent ∇V, which is consistent with the observations in [HJE18, HLZ20]. Thus, for VR-LSTD, the gradient of the value function is represented by a separate neural network $G(\cdot;\theta_G)$ with collective parameters θ_G.

To summarize, w.r.t. the collective parameters, the loss functions for the critic

corresponding to (3.31) and (3.32) are

$$L_1(\theta_V, \theta_G) = \mathbb{E}_{X_0\sim\mu,u}\Bigg[\Big(\int_0^{T\wedge\tau} e^{-\gamma s} f(X_s, u(X_s))\,\mathrm{d}s \tag{3.41}$$
$$- \int_0^{T\wedge\tau} e^{-\gamma s} G(X_s;\theta_G)^\top \sigma(X_s, u(X_s))\,\mathrm{d}W_s$$
$$+ e^{-\gamma(T\wedge\tau)} V(X_{T\wedge\tau};\theta_V) - V(X_0;\theta_V)\Big)^2\Bigg],$$

$$L_2(\theta_V) = \mathbb{E}_{X_0\sim\mu,u}\Bigg[\Big(\int_0^{T\wedge\tau} e^{-\gamma s} f(X_s, u(X_s))\,\mathrm{d}s + e^{-\gamma(T\wedge\tau)} V(X_{T\wedge\tau};\theta_V) \tag{3.42}$$
$$- V(X_0;\theta_V)\Big)^2\Bigg].$$

We remark that there is no need to add penalty terms in L_1 to ensure the consistency between $V(x,\theta_V)$ and $G(x;\theta_G)$, because if we replace $V^u(\cdot)$ by $V(\cdot,\theta_V)$ in (3.24) and plug it in (3.41), we have

$$L_1(\theta_V, \theta_G) = \mathbb{E}_{X_0\sim\mu,u}\Bigg[\Big(\int_0^{T\wedge\tau} e^{-\gamma s}[(\mathcal{L}^u V - \gamma V)(X_s;\theta_V) + f(X_s, u(X_s))]\,\mathrm{d}s \tag{3.43}$$
$$- \int_0^{T\wedge\tau} e^{-\gamma s}\left(\nabla_x V(X_s;\theta_V) - G(X_s;\theta_G)\right)^\top \sigma(X_s, u(X_s))\,\mathrm{d}W_s\Big)^2\Bigg]$$
$$= \mathbb{E}_{X_0\sim\mu,u}\Big[\Big(\int_0^{T\wedge\tau} e^{-\gamma s}[(\mathcal{L}^u V - \gamma V)(X_s;\theta_V) + f(X_s, u(X_s))]\,\mathrm{d}s\Big)^2\Big] \tag{3.44}$$
$$+ \mathbb{E}_{X_0\sim\mu,u}\Big[\int_0^{T\wedge\tau} e^{-2\gamma s}\left|\sigma^\top(X_s, u(X_s))\left(\nabla_x V(X_s;\theta_V) - G(X_s;\theta_G)\right)\right|^2\,\mathrm{d}s\Big], \tag{3.45}$$

where $\mathcal{L}^u$ is the generator of the SDE and we have used Itô's isometry in the second step. Note that the first (3.44) and second (3.45) terms in (3.43) simultaneously enforce V to be the value function and its gradient to be consistent with G.

For a neural network parametrization of V, it is not easy to directly impose the Dirichlet boundary condition $V = g$ on $\partial\Omega$ in the parametrization. Thus, instead, we

add a penalty term to the loss functions (3.41) or (3.42) for the critic to help enforce the boundary condition

$$\eta\, \mathbb{E}_{X\sim \mathrm{Unif}(\partial\Omega)} \left[(V(X;\theta_V) - g(X))^2 \right], \tag{3.46}$$

where η is a penalty hyperparameter and $\mathrm{Unif}(\partial\Omega)$ denotes the uniform distribution on $\partial\Omega$.

3.3.2 Discretization of SDEs and stochastic integrals

In the implementation, we need to simulate numerically, based on a discretization of the diffusion process with approximating stopping time and exit point. The solution to the PDE problem crucially depends on the boundary condition, and thus in control formulation, the exit time and position of the SDE at the boundary. Several schemes have been developed in the literature to deal with the stopping time and exit point of the SDEs in related scenarios. Perhaps the most natural idea is to stop at the last step of the numerical SDE before exiting the domain, which has been tested in the context of using neural networks for solving PDEs in [KSS20b]. The error of such boundary treatment has been analyzed in [Gob00]. Moreover, several schemes have been proposed to improve the accuracy around the boundary. In [HNS20], the authors approximate the exit position by the intersection of the domain boundary and the line segment between the consecutive two steps before and after exiting the domain. It has also been considered to reduce step size when the discretized trajectory approaches the boundary [BP03]. Some bias reduction schemes with the bubble-wrap or max-sampling exit condition are proposed in [MZC+21]. After studying and testing several approaches for numerical discretization in our algorithms, we present two choices of discretization and give some remarks on the other schemes.

Let us start with a naïve approach. We can discretize the SDE (3.2) by the Euler–Maruyama scheme with a given partition of interval $[0, T]$: $0 = t_0 < t_1 < \cdots < t_N =$

T, where a constant step size $\Delta t = \frac{T}{N}$ is used, so $t_n = n\Delta t$. The SDE is discretized as

$$\mathcal{X}_0 = X_0, \qquad \mathcal{X}_{t_{n+1}} = \mathcal{X}_{t_n} + b(\mathcal{X}_{t_n}, u_n)\Delta t + \sigma(\mathcal{X}_{t_n}, u_n)\xi_n\sqrt{\Delta t}, \tag{3.47}$$

where $u_n = u(\mathcal{X}_{t_n}; \theta_u)$ and $\xi_n \sim N(0, I_{d_w})$ follows the standard normal distribution. Here, we use $\mathcal{X}_{t_n}$ to denote the discretized stochastic process, to distinguish from X_t, the continuous process. Given a numerical trajectory $\mathcal{X}_{t_n}, n = 0, \ldots, N$, we define

$$\bar{n} = \max\big\{n \in \{0, \ldots, N\} \mid \mathcal{X}_{t_i} \in \Omega,\ i = 0, 1, \cdots, n\big\}. \tag{3.48}$$

Thus, if $\bar{n} < N$, $\mathcal{X}_{t_{\bar{n}+1}}$ exits the domain as $\mathcal{X}_{t_{\bar{n}+1}} \notin \Omega$, while if $\bar{n} = N$ the trajectory $\mathcal{X}_{t_n}$ remains in the domain for $n = 0, 1, \ldots, N$.

Perhaps the most direct and intuitive approach for the boundary treatment is to view $t = \bar{n}\Delta t$ as the stopping time, even though $\mathcal{X}_{t_{\bar{n}}}$ is still inside Ω. This scheme will be referred to as the "naïve scheme" in the following. The stochastic integrations in (3.31), (3.32), and (3.36) are correspondingly approximated by

$$\begin{aligned}
&\int_0^{T\wedge\tau} e^{-\gamma s} f(X_s, u_s)\,\mathrm{d}s \approx \sum_{n=0}^{\bar{n}-1} e^{-\gamma n\Delta t} f(\mathcal{X}_{t_n}, u_n)\Delta t,\\
&\int_0^{T\wedge\tau} e^{-\gamma s}\nabla V(X_s)^\top \sigma(X_s, u_s)\,\mathrm{d}W_s \approx \sum_{n=0}^{\bar{n}-1} e^{-\gamma n\Delta t} G(\mathcal{X}_{t_n}; \theta_G)^\top \sigma(\mathcal{X}_{t_n}, u_n)\xi_n\sqrt{\Delta t},
\end{aligned} \tag{3.49}$$

where $\xi_n\sqrt{\Delta t}$ is the same realization of Brownian increments as in (3.47). We remark that this discretization scheme is similar to the one used in [KSS20b] for solving degenerate semilinear elliptic equations, in particular, both algorithms approximate the stopping time by $\bar{n}\Delta t$. However, we aim to solve the value function in the whole domain, while the method developed in [KSS20b] only aims at the value at a specific point; thus the overall framework of the algorithm is quite different.

After discretization, the loss functions (3.31), (3.32), and (3.36) are further ap-

proximated by Monte Carlo samples: for each iteration, we draw K independent sample trajectories (K is known as the batch size) by drawing initial point X_0 from the distribution μ and independent increments of the Brownian motion. At each iteration, we also draw K independent Monte Carlo samples uniformly from the boundary to approximate the expectation in (3.46). To update the parameters of the neural networks, we employ the Adam optimizer [KB15].

To apply the policy gradient method to the loss functional (3.36), we need to differentiate the discretized functional w.r.t. the control, similar to the functional derivative setting considered above in (3.37). While the first and third terms in (3.37), which involve derivatives of J through its dependence on u and the trajectory, can be easily dealt with on the discretized level using autodifferentiation, the second and fourth terms in (3.37) become tricky to deal with on the discrete level, since the stopping time is approximated by $\bar{n}\Delta t$, which is discrete so $\delta\bar{n}/\delta u$ is not really well defined. In our implementation, such terms are omitted in the policy gradient w.r.t. u; we leave a better numerical treatment of such terms to future works.

The pseudocode for our actor-critic method for solving high dimensional PDEs is summarized in Algorithm 1.

3.3.3 The adaptive step size scheme

It turns out in our numerical experiments that while the above naïve scheme is able to get reasonably accurate value functions, the approximation to control results in large errors, especially near the boundary (see Section 3.4 for more details). To improve the accuracy near the boundary, we adaptively shrink the step size when the trajectory approaches the boundary $\partial\Omega$, instead of using the uniform time step size as in the naïve scheme. More specifically, we use the following scheme at the boundary, which is motivated by the integration scheme used in [BP03] for the Feynman–Kac

Algorithm 2 Neural network based actor-critic solver for fully nonlinear PDEs

Input: A fully nonlinear PDE (3.1), terminal time T, number of time intervals N, loss weights η, neural network structures, number of iterations, learning rate, batch size K, the choice of TD

Output: Value function $V(\cdot\,;\theta_V)$, its gradient $G(x;\theta_G)$ if we choose VR-LSTD, and the control $u(\cdot\,;\theta_u)$

initialization: θ_{value} ($\theta_{\text{value}} = (\theta_V, \theta_G)$ for VR-LSTD and $\theta_{\text{value}} = \theta_V$ for LSTD) and θ_u

for $\ell = 1$ **to** the number of iterations **do**

Sample K independent trajectories $\mathcal{X}_{t_n}^k, k = 1, 2, \cdots, K$ {critic steps}

Estimate the gradient of the chosen critic loss ((3.41) + (3.46) or (3.42) + (3.46)) w.r.t. θ_{value} using the K trajectories and K boundary points

Update parameters θ_{value} using the Adam optimizer

Sample K independent trajectories $\mathcal{X}_{t_n}^k, k = 1, 2, \cdots, K$ {actor steps}

Estimate the gradient of the actor loss (3.36) w.r.t. θ_u using the K trajectories

Update parameters θ_u using the Adam optimizer

end for

representation of boundary value problems of the Poisson equation.

The idea is to reduce the time step size adaptively when $\mathcal{X}_t$ is close to the boundary, and thus to improve the accuracy of the trajectory. We consider the Euler–Maruyama scheme with varying step size given by

$$\mathcal{X}_{t_{n+1}} = \mathcal{X}_{t_n} + b(\mathcal{X}_{t_n}, u_n)h(\mathcal{X}_{t_n}) + \sigma(\mathcal{X}_{t_n}, u_n)\sqrt{h(\mathcal{X}_{t_n})}\,\xi_n, \tag{3.50}$$

where the step size $h(\mathcal{X}_{t_n})$ depends on the current position of the trajectory. For the choice of step size, we define a subset near the boundary of Ω as

$$\Gamma = \left\{x \in \overline{\Omega} \mid \operatorname{dist}(x, \partial\Omega) \le \varsigma\sqrt{3d\Delta t}\right\}, \tag{3.51}$$

where $\varsigma = \sup_{x\in\Omega, u\in U}\|\sigma(x,u)\|$ is the supremum of the operator norm of σ. The adaptive choice of the step size is specified as follows:

1. When $\mathcal{X}_{t_n} \in \Omega\backslash\Gamma$, it would be considered in the "interior" of Ω, as it is very unlikely that after one time step with step size Δt that the trajectory will exit the domain. Thus, we will use the basic constant step size $h(\mathcal{X}_{t_n}) = \Delta t$.

2. When $\mathcal{X}_{t_n} \in \Gamma$, we decrease the step size according to the distance of the trajectory to the boundary, with the minimum step size set as $\frac{1}{10^4}\Delta t$:

$$h(\mathcal{X}_{t_n}) = \max\Big\{\frac{1}{3d\varsigma^2}\operatorname{dist}(\mathcal{X}_{t_n}, \partial\Omega)^2, \frac{1}{10^4}\Delta t\Big\}.$$

This reduced step size, together with the width of Γ defined in (3.51), are decided such that the probability that $\mathcal{X}_{t_n} \in \Omega\backslash\Gamma$ goes out of Ω in the next step is small. Note that when the step size is small, the diffusion dominates the drift term, and thus it suffices to incorporate the diffusion part in the choice. Note that we have used the supremum of $\|\sigma\|$ for simplicity, one could also choose the criteria more locally if σ varies a lot across the domain. The minimum step size is set to balance the accuracy and computational cost as, otherwise, the scheme might spend an unnecessarily long time resolving the trajectory near the domain boundary.

In summary, we choose the adaptive step size $h = h(\mathcal{X}_{t_n})$ as

$$h(\mathcal{X}_{t_n}) = \begin{cases} \Delta t, & \mathcal{X}_{t_n} \in \Omega\backslash\Gamma, \\ \max\{\frac{1}{3d\varsigma^2}\operatorname{dist}(\mathcal{X}_{t_n}, \partial\Omega)^2, \frac{1}{10^4}\Delta t\}, & \mathcal{X}_{t_n} \in \Gamma. \end{cases} \tag{3.52}$$

It should be noted that, as a result of the adaptive step size, different trajectories may have different discretized time steps. The integrals are similarly discretized as in (3.49) with step size changed to $h(\mathcal{X}_{t_n})$:

$$\begin{aligned} &\int_0^{T\wedge\tau} e^{-\gamma s} f(X_s, u)\,\mathrm{d}s \approx \sum_{n=0}^{\bar{n}-1} e^{-\gamma\sum_{k=0}^{n-1} h(\mathcal{X}_{t_k})} f(\mathcal{X}_{t_n}, u_n) h(\mathcal{X}_{t_n}); \\ &\int_0^{T\wedge\tau} e^{-\gamma s} \nabla V(X_s)^\top \sigma(X_s)\,\mathrm{d}W_s \\ &\qquad \approx \sum_{n=0}^{\bar{n}-1} e^{-\gamma\sum_{k=0}^{n-1} h(\mathcal{X}_{t_k})} G(\mathcal{X}_{t_n}; \theta_G)^\top \sigma(\mathcal{X}_{t_n}, u_n)\xi_n\sqrt{h(\mathcal{X}_{t_n})}. \end{aligned} \tag{3.53}$$

For the policy gradient, similar to our numerical treatment in the case of naïve scheme, we use autodifferentiation generated by the computational graph in practice, instead of directly numerically approximating the functional derivative defined in (3.37). One reason is that the adaptive step size scheme further complicates the dependence of the trajectory and exit time on the control, compared with the naïve scheme and, hence, makes the direct numerical discretization of (3.37) even more difficult. In practice, the result from using autodifferentiation for the policy gradient seems to be quite accurate, as will be further discussed in the next section.

Remark. *In addition to the adaptive step size, in our numerical experiments, we have also tested the bounded sample of Brownian increments proposed in [BP03] to further avoid the potentially large error of the trajectory near the boundary due to tail events of the normal sample. We do not find, however, a significant difference in the result between using bounded samples versus the usual normal samples for Brownian increments. Therefore, we will stick to the normal samples for simplicity.*

Remark. *Moreover, besides adaptively shrinking the step size near the boundary, we have tested two approaches using constant step size, but try to improve the estimate of the exit time and exit point of the naïve scheme instead. They do not yield satisfactory numerical results, so we will only briefly sketch the ideas without going into details or presenting numerical results.*

One scheme is adapted from [HNS20], which tries to determine the exit point on $\partial\Omega$ more accurately. In this scheme, the exit position $\mathcal{X}_\tau$ on $\partial\Omega$ is numerically approximated by the intersection of $\partial\Omega$ and the line segment between $\mathcal{X}_{t_{\bar{n}}}$ and $\mathcal{X}_{t_{\bar{n}+1}}$; the stopping time is correspondingly adjusted. The numerical result from the scheme is still not accurate enough for the control near the boundary.

The linear interpolation above gives an error of order $\sqrt{\Delta t}$ due to the diffusion term; we can further improve the accuracy using a method proposed in [Gob00]. In-

stead of linear interpolation, we seek for a coefficient $\rho \in (0, 1]$ *such that* $\mathcal{X}_\tau$*, defined by*

$$\mathcal{X}_\tau = \mathcal{X}_{t_{\bar{n}}} + b(\mathcal{X}_{t_{\bar{n}}}, u_{\bar{n}})\rho\Delta t + \sigma(\mathcal{X}_{t_{\bar{n}}}, u_{\bar{n}})\sqrt{\rho\Delta t}\,\xi_{\bar{n}},$$

is on $\partial\Omega$*. In practice, we observe that numerically solving the coefficient* ρ *makes the training unstable.*

3.4 Numerical examples

In this section, we present the numerical results for the proposed method. We test on several examples: the linear quadratic regulator (LQR) problems, the stochastic Van der Pol oscillator problems, the diffusive Eikonal equations, and fully nonlinear elliptic PDEs derived from a regulator problem. To test the performance of our algorithm, we do not assume knowledge of the true solution or the explicit formula for the control given the value function. The considered dimensions in all four examples are as large as 20. The algorithm is implemented in Python with the deep learning library TensorFlow 2.0 [ABC+16]. In all the examples, the weight parameter η associated with the boundary condition (cf. (3.46)) is set to 1 and the terminal time is $T = 0.2$. The numbers of time intervals are $N = 50$ for problems in 4 dimensions (4d) and 5d, and $N = 100$ in 10d and 20d. As for the architecture of the neural networks, the width of the hidden layers is set to 200 in all problems, while the numbers of hidden layers are 2 for problems in 4d and 5d, and 3 in 10d and 20d. During the training, we use piecewise constant learning rates of 1×10^{-3}, 1×10^{-4}, and 1×10^{-5} consecutively in order to achieve high accuracy. The numbers of steps with learning rate 1×10^{-3} are 20000 for problems in 4d, 5d, and 10d, and 30000 in 20d. The numbers of steps with learning rate 1×10^{-4} and 1×10^{-5} are both 10000 in the four examples. The batch sizes are $K = 1024$ for problems in 4d and 5d, and $K = 2048$ in 10d and 20d. The parameters in the numerical examples are determined empirically. In order to

illustrate the effect of some parameters such as T and the basic step size Δt, we also compare the results with different parameters in the first example.

During the training, we sample a validation set $\{X^k\}_{k=1}^K$ uniformly in Ω, independent of the training, to evaluate the errors of the value function and the control. Note that the validation size K is the same as the batch size. We find that such sizes are enough to estimate the error accurately with a small variance. The relative L^2 errors are computed by

$$\text{err}_V^2 = \sum_{k=1}^{K} (V(X^k) - V(X^k; \theta_V))^2 / \sum_{k=1}^{K} V(X^k)^2 \tag{3.54}$$

and

$$\text{err}_u^2 = \sum_{k=1}^{K} |u(X^k) - u(X^k; \theta_u)|^2 / \sum_{k=1}^{K} |u(X^k)|^2, \tag{3.55}$$

where $V(\cdot)$ and $u(\cdot)$ are the true value and control functions, respectively (we will choose test examples such that these true solutions are known). In addition to the errors above, we also visualize the density of the true value function and compare that with its neural network approximation, considering the difficulty of visualizing functions in high dimensions directly. Here, the density of a function V is defined as the probability density function of $V(X)$, where X is uniformly distributed in Ω. In our numerical experiments, the density is estimated by Monte Carlo sampling.

Our numerical results indicate that in all the examples, the value functions are approximated accurately, and the associated densities match well with that of the true solution. Furthermore, the numerical results show that, for the critic, the VR-LSTD performs better than LSTD, as expected. The adaptive step size scheme also significantly improves the accuracy, in particular, for the control. The details can be found in the following subsections.

3.4.1 LQR

In this subsection we consider the PDE arising from the LQR problem, given by

$$\Delta V(x) + \inf_{u\in\mathbb{R}^d}\big(\beta u^\top \nabla V(x) + p|x|^2 + q|u|^2 - 2kd\big) - \gamma V(x) = 0 \quad \text{in } B_R \subset \mathbb{R}^d \tag{3.56}$$

with boundary condition $V(x) = kR^2$ on ∂B_R, where $B_R = \{x \in \mathbb{R}^d : |x| < R\}$. Here p, q, β, k are positive constants such that

$$k = \frac{\sqrt{q^2\gamma^2 + 4pq\beta^2} - \gamma q}{2\beta^2}. \tag{3.57}$$

This is the HJB equation corresponding to the controlled stochastic process

$$\mathrm{d}X_t = \beta u\,\mathrm{d}t + \sqrt{2}\,\mathrm{d}W_t \tag{3.58}$$

with cost functional

$$J^u(x) = \mathbb{E}\Big[\int_0^\tau (p|X_s|^2 + q|u(X_s)|^2 - 2kd)e^{-\gamma s}\,\mathrm{d}s + e^{-\gamma\tau}kR^2\Big], \tag{3.59}$$

where τ is the exit time of the domain B_R. The PDE has the exact solution as a quadratic function, $V(x) = k|x|^2$, and the optimal control is also explicitly given as $u^*(x) = \frac{-\beta}{2q}\nabla V(x) = \frac{-k\beta}{q}x$.

We choose the model parameters $p = q = R = \beta = \gamma = 1$ and $k = (\sqrt{5}-1)/2$. The numerical results for our two versions of TDs with different discretization schemes in 5d are shown in Table 3.1. The results of VR-LSTD have smaller errors due to the smaller asymptotic variance, as we discussed above. Moreover, the adaptive step size scheme is able to compute a more accurate control function, compared with the naïve scheme. One possible reason is that the adaptive step size scheme samples more points $\mathcal{X}_{t_n}$ near the boundary, which helps to improve the accuracy of the control function near the boundary. To further illustrate the idea, let us compare the

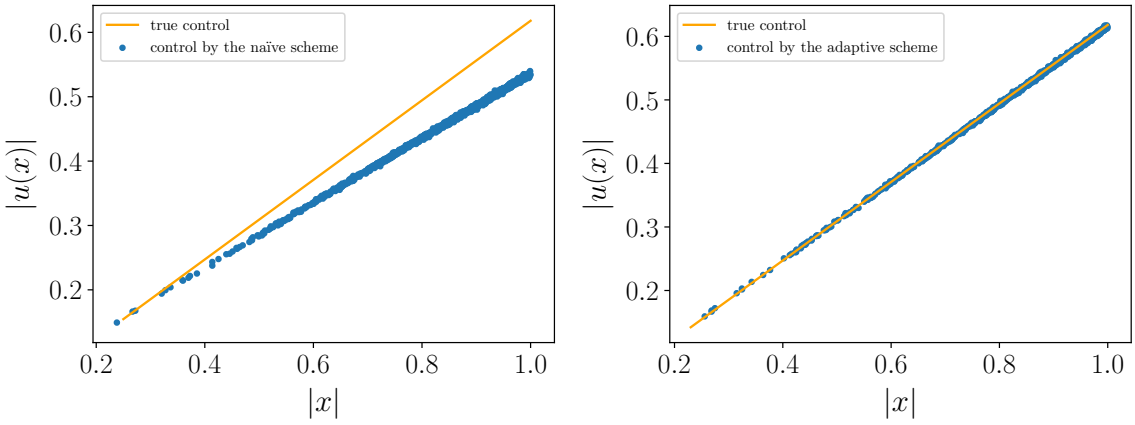

Figure 3.1: Comparison of schemes.

results for the naïve scheme and the adaptive step size scheme in 5d, both with critics optimized by VR-LSTD. The plot of the norm of the control $|u(x)|$ w.r.t. the norm of the variable $|x|$ is shown in Figure 3.1. The error of the control for the naïve scheme is significantly larger near the boundary. The adaptive step size scheme achieves a uniform accuracy of the control in the whole domain.

Therefore, for the rest of the numerical experiments, we will stick to the adaptive step size scheme and VR-LSTD loss function for the critic. Figure 3.2 shows the density and error curves for the LQR problem when $d = 5, 10, 20$. The sharp drop of the errors at steps 20000 and 30000 is due to the reduced learning rates at those steps. The final errors of the value functions and controls are 1.02×10^{-2} and 9.19×10^{-3} in 5d; 1.40×10^{-2} and 1.95×10^{-2} in 10d; 1.96×10^{-2} and 4.78×10^{-2} in 20d.

Considering that T and the basic step size Δt are two hyperparameters in our algorithm, we test different choices of their values in the 5d LQR example and provide the errors in Table 3.2. The results show that our algorithm is not sensitive to the choice of T and Δt.

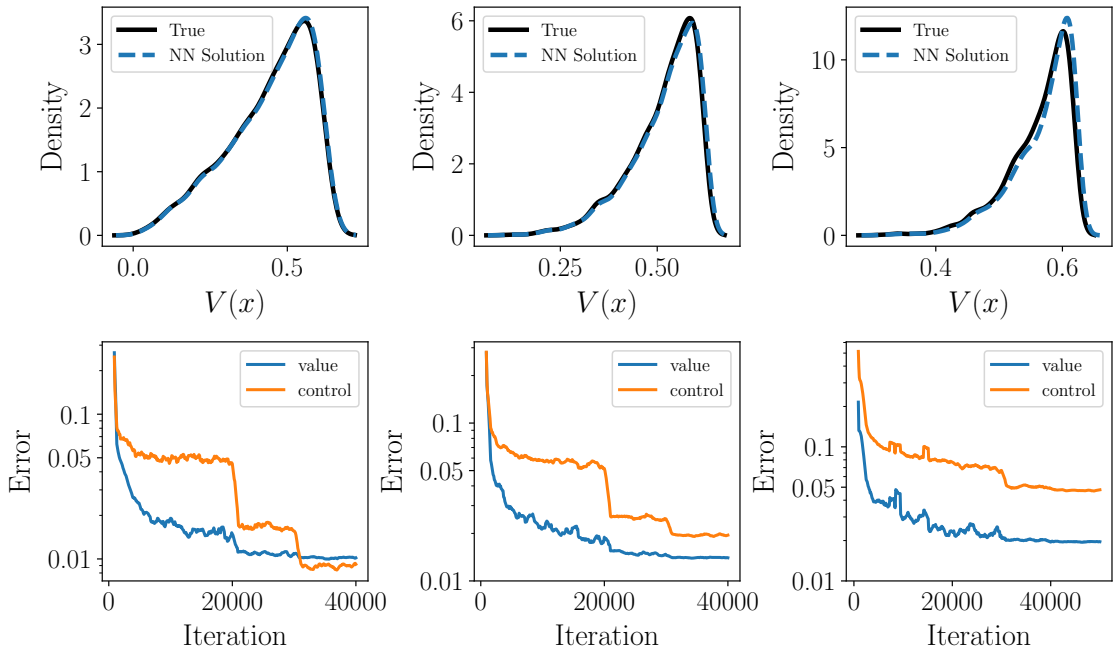

Figure 3.2: Density and error curves for LQR.

Table 3.1: Errors for different discretization schemes.

discretization	TD variant	error of value function	error of control
adaptive step size	VR-LSTD	1.02×10^{-2}	9.19×10^{-3}
adaptive step size	LSTD	1.58×10^{-1}	1.17×10^{-1}
naïve	VR-LSTD	1.29×10^{-2}	1.24×10^{-1}
naïve	LSTD	1.41×10^{-1}	8.55×10^{-2}

Table 3.2: Errors of value and control functions.

T		$T = 0.04$	$T = 0.1$	$T = 0.2$	$T = 0.4$	$T = 0.8$
50 intervals	value	8.40×10^{-3}	9.46×10^{-3}	1.04×10^{-2}	1.03×10^{-2}	9.97×10^{-3}
	control	1.15×10^{-2}	9.87×10^{-3}	1.01×10^{-2}	8.99×10^{-3}	8.33×10^{-3}
step size 0.004	value	3.17×10^{-2}	1.40×10^{-2}	9.96×10^{-3}	9.16×10^{-3}	8.76×10^{-3}
	control	3.51×10^{-2}	1.19×10^{-2}	9.39×10^{-3}	1.03×10^{-2}	1.07×10^{-2}

3.4.2 Stochastic Van der Pol oscillator

The Van der Pol oscillator is a popular example in the study of dynamical systems because of its chaotic behavior. The stochastic Van der Pol oscillator has been studied in [XGZ+11], in which some internal or external noise is considered. In

this subsection, we consider the generalized stochastic Van der Pol oscillator in high dimensional cases and solve the PDE

$$\Delta V(x) + \inf_{u\in\mathbb{R}^{d/2}}\big[b(x,u)^\top \nabla V(x) + f(x,u)\big] - \gamma V(x) = 0 \quad \text{in } B_R \subset \mathbb{R}^d, \tag{3.60}$$

where $d = 2n$ is even. The boundary condition is given by (with convention $x_0 = x_n$ and $x_{2n+1} = x_{n+1}$)

$$g(x) = a\sum_{i=1}^{2n}(x_i)^2 - \epsilon\Big(\sum_{i=1}^{n} x_{i-1}x_i + \sum_{i=n+1}^{2n} x_i x_{i+1}\Big). \tag{3.61}$$

Here a and ϵ are positive constants. The drift field is given by

$$b_i(x,u) = \begin{cases} x_{i+n} & (1 \le i \le n), \\ (1-x_{i-n}^2)x_i - x_{i-n} + u_{i-n} & (n+1 \le i \le 2n). \end{cases} \tag{3.62}$$

We choose the running cost as

$$\begin{aligned} f(x,u) = q|u|^2 + \gamma[&\sum_{i=1}^{n}(ax_i^2 - \epsilon x_i x_{i-1}) + \sum_{i=n+1}^{2n}(ax_i^2 - \epsilon x_i x_{i+1})] \\ &+ \frac{1}{4q}[(2ax_{n+1} - \epsilon x_{2n} - \epsilon x_{n+2})^2 + \sum_{i=n+2}^{2n}(2ax_i - \epsilon x_{i-1} - \epsilon x_{i+1})^2] - 4na \\ &- 2a\sum_{i=1}^{n} x_{n+i}x_i + \epsilon\sum_{i=1}^{n} x_{n+i}x_{i-1} + \epsilon\sum_{i=1}^{n-1} x_{n+i}x_{i+1} + \epsilon x_{2n}x_1 \\ &- (x_{n+1} - x_1 - x_1^2 x_{n+1})(2ax_{n+1} - \epsilon x_{2n} - \epsilon x_{n+2}) \\ &- \sum_{i=2}^{n}(x_{i+n} - x_i - x_i^2 x_{i+n})(2ax_{i+n} - \epsilon x_{i+n-1} - \epsilon x_{i+n+1}), \end{aligned} \tag{3.63}$$

so that the true value function has an explicit formula:

$$V(x) = a\sum_{i=1}^{2n}(x_i)^2 - \epsilon\Big(\sum_{i=1}^{n} x_{i-1}x_i + \sum_{i=n+1}^{2n} x_i x_{i+1}\Big). \tag{3.64}$$

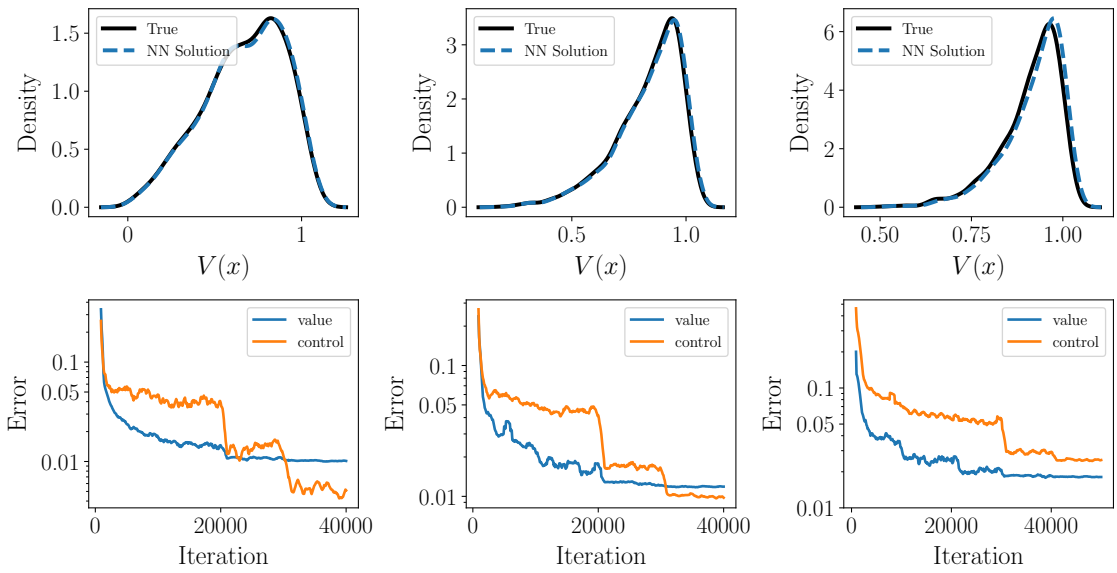

Figure 3.3: Density and error curves for Van der Pol.

The corresponding optimal control is given by $u_1^*(x) = -\frac{1}{2q}\partial_{n+1}V(x) = 2ax_{n+1} - \epsilon x_{2n} - \epsilon x_{n+2}$ and $u_i^*(x) = -\frac{1}{2q}\partial_{i+n}V(x) = 2ax_{i+n} - \epsilon x_{i+n-1} - \epsilon x_{i+n+1}$ for $i = 2, 3, \cdots, n$.

The PDE can be reformulated as a stochastic control problem with the controlled SDE given by

$$\mathrm{d}X_t = b(X_t, u)\,\mathrm{d}t + \sqrt{2}\,\mathrm{d}W_t \tag{3.65}$$

with objective function

$$J^u(x) = \mathbb{E}\left[\int_0^\tau f(X_s, u)e^{-\gamma s}\,\mathrm{d}s + e^{-\gamma\tau}g(X_\tau)\right]. \tag{3.66}$$

In the numerical experiments, we take $a = q = R = \gamma = 1$ and $\epsilon = 0.1$. Figure 3.3 shows the density and error curves when $d = 4, 10, 20$. The algorithm learns reasonably nice shapes of the value functions. The final errors of the value functions and controls are 1.01×10^{-2} and 5.12×10^{-3} in 4d; 1.19×10^{-2} and 9.77×10^{-3} in 10d; 1.81×10^{-2} and 2.50×10^{-2} in 20d.

3.4.3 Diffusive Eikonal equation

The Eikonal equation corresponds to the shortest-path problems with a given metric. In our experiments, we add a small diffusion term to regularize the equation (otherwise, the solution has kinks, which creates difficulty for the neural networks to approximate well in high dimensions). The diffusive Eikonal equation is given by

$$\begin{cases} \epsilon \Delta V(x) + \inf_{u \in B_1} \big(c(x) u^\top \nabla V(x) \big) + 1 = 0 \quad \text{in } B_R, \\ V(x) = a_3 - a_2 \quad \text{on } \partial B_R, \end{cases} \tag{3.67}$$

where

$$c(x) = \frac{3(d+1)a_3}{2da_2(2a_2 - 3a_3|x|)} > 0 \tag{3.68}$$

is a real valued function. Here a_2 and a_3 are positive constants such that $2a_2 - 3a_3 R > 0$ and $\epsilon = 1/(2da_2)$. We choose the form of c so that the true solution of the PDE is explicitly given by

$$V(x) = a_3|x|^3 - a_2|x|^2 \tag{3.69}$$

and the optimal control is $u^*(x) = x/|x|$. In the numerical test, we take $a_2 = 1.2$, $a_3 = 0.2$, and $R = 1$.

Unlike the previous two examples, the constraint on the control in this example poses a new challenge to the numerical algorithm. In order to ensure that the control u is in the unit ball, we construct a specific structure of the neural network for the control. Instead of outputting the control directly, the neural network gives a $d+1$ dimensional vector $(u_{\text{len}}, u_{\text{dir}}) \in \mathbb{R}^{d+1}$. The control is represented by

$$u = \frac{u_{\text{dir}}}{\delta + \text{ReLU}(u_{\text{len}}) + |u_{\text{dir}}|}, \tag{3.70}$$

where $\text{ReLU}(x) = \max(0, x)$ and $\delta = 10^{-15}$. This δ is to ensure that the denominator in (3.70) is not 0 to prevent numerical singularity. Figure 3.4 shows the density

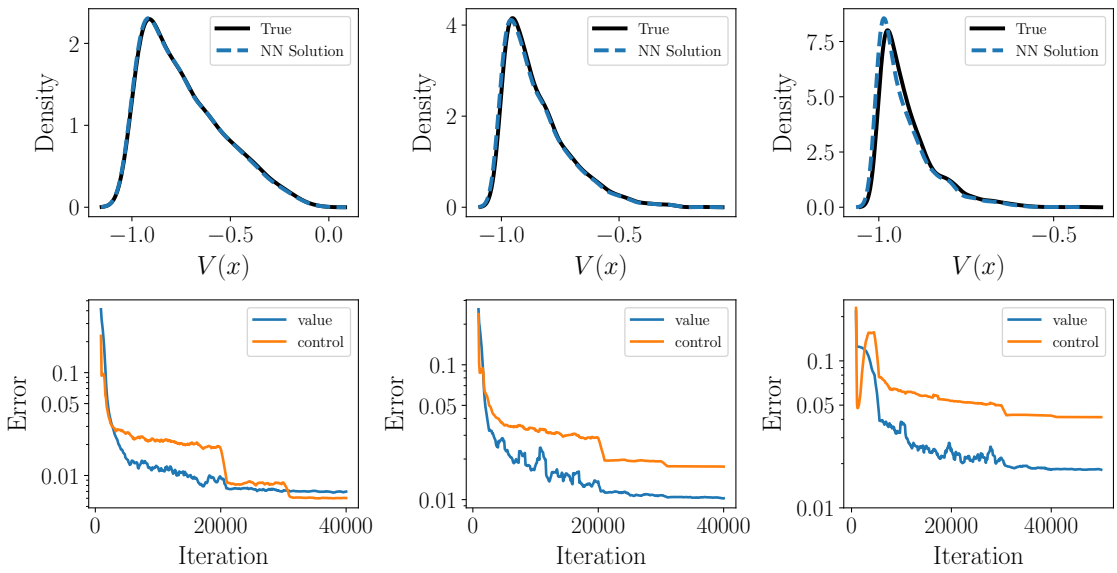

Figure 3.4: Density and error curves for Eikonal.

and error curves for the Eikonal equation when $d = 5, 10, 20$. We also tried the straightforward parametrization of the control function as before, with an additional penalty term $\eta' \mathbb{E}_{X \sim \mathrm{Unif}(\Omega)}[\mathrm{ReLU}(|u(X)| - 1)]$ in the loss for the actor. However, the numerical performances indicate that implementing the constraints of control directly like (3.70) is better than the penalty method. The final errors of the value functions and controls are 6.97×10^{-3} and 6.03×10^{-3} in 5d; 1.02×10^{-2} and 1.76×10^{-2} in 10d; 1.82×10^{-2} and 4.14×10^{-2} in 20d.

3.4.4 LQR with a nonconstant diffusion coefficient

In this subsection, we consider a variant of the LQR in which the diffusion coefficient σ is a function of both x and u. Consider the HJB equation

$$\inf_{u \in \mathbb{R}^d} \Big[\sum_{i=1}^{d} \big(\partial_i^2 V(x)(1 + \epsilon x_i u_i)^2 + \beta \partial_i V(x) u_i \big) + q|u|^2 + \widetilde{f}(x) \Big] - \gamma V(x) = 0 \quad \text{in } B_R \subset \mathbb{R}^d, \tag{3.71}$$

where

$$\widetilde{f}(x) = \gamma k|x|^2 + \sum_{i=1}^{d} \frac{k^2(\beta+2\epsilon)^2 x_i^2}{q + 2k\epsilon^2 x_i^2} - 2kd. \tag{3.72}$$

In contrast to the previous three examples, this is a fully nonlinear PDE. The corresponding SDE is

$$\mathrm{d}X_t = \beta u_t \,\mathrm{d}t + \sigma(X_t, u_t)\,\mathrm{d}W_t, \tag{3.73}$$

where $\sigma(x,u)$ is a diagonal matrix with i-th diagonal element $\sqrt{2}(1+\epsilon x_i u_i)$, $i = 1, \cdots, d$. The running cost is $f(x,u) = q|u|^2 + \widetilde{f}(x)$. The true value function is $V(x) = k|x|^2$ and the optimal control is

$$u_i^*(x) = -\frac{\beta\partial_i V(x) + 2\epsilon x_i \partial_i^2 V(x)}{2q + 2\epsilon^2 x_i^2 \partial_i^2 V(x)} = -\frac{(\beta+2\epsilon)x_i}{q/k + 2\epsilon^2 x_i^2}. \tag{3.74}$$

Note that this example coincides with the first example when $\epsilon = 0$. In the numerical experiments, we set the parameters $q = R = \beta = \gamma = 1, k = (\sqrt{5}-1)/2$ the same as the first example and $\epsilon = -1$. The final errors of the value functions and controls are 9.98×10^{-3} and 1.63×10^{-2} in 5d; 1.50×10^{-2} and 4.95×10^{-2} in 10d; 1.96×10^{-2} and 5.25×10^{-2} in 20d. This example showcases that our algorithm is able to solve fully nonlinear elliptic PDEs in high dimensions accurately.

3.5 Conclusion and future directions

In this paper, we propose and study numerical methods for high dimensional static HJB equations based on neural network parametrization and the actor-critic framework. There are several promising directions for future research. First, the scalability of the methods shall be further tested by problems of higher dimensions. Second, it would be interesting to extend our methods to other types of boundary conditions like natural boundary conditions or broader types of equations, such as the porous

medium equation. In both cases, the corresponding control formulation is not so clear. Third, one might explore the better numerical treatment of discretization of the functional derivative (3.37), rather than relying on autodifferentiation. Finally, as an outstanding challenge in the field of deep learning, theoretical analysis for convergence and error analysis of the proposed numerical methods would be of great interest.

Chapter 4

Single Timescale Actor-Critic Method to Solve the Linear Quadratic Regulator with Convergence Guarantees

4.1 Introduction

Reinforcement learning (RL) is a semi-supervised learning model that learns to take actions and interact with the environment in order to maximize the expected reward [SB18]. It has a wide range of applications, including robotics [KBP13], traditional games [SHM$^+$16], and traffic light control [Wie00]. RL is closely related to the optimal control problem [Ber19], where one usually minimizes the expected cost instead of maximizing the reward. Among all the control problems, the LQR [AM07] is the cleanest setup to analyze theoretically and has many applications [Has19, ESJB10]. Many research has been devoted to LQR. Early research mostly focused on model-based methods, such as deriving the explicit solution of the LQR with known dynamics. This research showed that the optimal control is a linear function of the state and the coefficient can be obtained by solving the Riccati equation [AM07]. Recent research focuses more on the model-free setting in the context of RL, where the algorithm does not know the dynamic and has only observations of states and rewards [TR18, MZSJ21].

The actor-critic method [KT00] is a class of algorithms that solve the RL or optimal control problems through alternately updating the actor and the critic. In this framework, we solve for both the control and the value function, which is the expected cost w.r.t. the initial state (and action). The control is known as the actor,

so in the actor update, we improve the control in order to minimize the cost; i.e., policy improvement. The value function is known as the critic. Hence, in the critic update, we evaluate a fixed control through computing the value function; i.e., policy evaluation.

On a broader scale, the actor-critic method belongs to the bilevel optimization problem [SMD17, Bar13], as it is an optimization problem (higher-level problem) whose constraint is another optimization problem (lower-level problem). In the actor-critic method, the higher-level problem is to minimize the cost (the actor) and the lower-level problem is to let the critic be equal to value function corresponding to the control, which is equivalent to minimizing the expected squared Bellman residual [BB96]. The major difficulty of a bilevel optimization problem is that when the lower-level problem is not solved exactly, the error could propagate to the higher-level problem and accumulate in the algorithm. One approach to overcome this problem is the two timescale method [KT00, WZXG20, ZDR21], where the update of lower-level problem is in a time scale that is much faster than the higher-level one. This method suffers from high computational costs because of the lower-level optimization. Another method is to modify the update direction to improve accuracy [Kak01], which also introduces extra cost. In order to reduce the cost, we seek an efficient single timescale method to solve LQR.

4.1.1 Our contributions

In this paper, we consider a single timescale actor-critic algorithm to solve the LQR problem. We apply an LSTD method [BB96] for the critic and a natural policy gradient method [Kak01] for the actor. For the critic, we derive an explicit expression for the gradient and design a sample method with the desired accuracy. For the actor, we apply a natural policy gradient method borrowed from [FGKM18]. We

give a proof of convergence with sample complexity $\mathcal{O}(\varepsilon^{-1}\log(\varepsilon^{-1})^2)$ to achieve an ε-optimal solution. To the best of our knowledge, our work is the first single timescale actor-critic method to solve the LQR problem with provable guarantees.

Our work not only solves the specific LQR problem but also advances the study of convergence for single timescale bilevel optimization. In our proof of convergence, we construct a Lyapunov function that involves both the critic error and the actor loss. We show that there is a contraction of the Lyapunov function in the algorithm. If we consider the actor and the critic separately, the critic error becomes an issue when we want to show an improvement of the actor and vice versa. Therefore, the higher and lower level problems have to be analyzed simultaneously for a single timescale algorithm.

4.1.2 Related works

Let us compare our work with related ones in the literature. Perhaps the most closely related work to ours is by [FYW20]. They consider a single timescale actor-critic method to solve the optimal control problem with discrete state and action spaces, while we solve the LQR problem with continuous state and action spaces. They add an entropy regularization in the loss function and achieve a sample complexity of $\mathcal{O}(\varepsilon^{-2})$ with linear parameterization.

For two timescale approaches, [YCHW19] study a two timescale actor-critic algorithm to solve the LQR problem in continuous space. They also use a natural policy gradient method for the actor [FGKM18]. For the critic, they reformulate policy evaluation into a minimax optimization problem using Fenchel's duality. Several critic steps are performed between two actor steps and their final sample complexity is $\mathcal{O}(\varepsilon^{-5})$. [ZDR21] study a bilevel optimization problem that is applied to a two timescale actor-critic algorithm on LQR. They obtain a complexity of $\mathcal{O}(\varepsilon^{-3/2})$.

They have assumed strong convexity of the higher-level loss function (actor) while our analysis does not require such assumptions.

Besides model-free approaches, another way to solve the LQR problem is to first learn the model through the system identification approach and then solve the model-based LQR. For example, [DMM+20] use a least square system identification approach to learn the model parameter and then solve the LQR, with sample complexity $\mathcal{O}(\varepsilon^{-2})$.

As can be seen from the above discussions, our single timescale algorithm achieves a lower sample complexity $\mathcal{O}(\varepsilon^{-1}\log(\varepsilon^{-1})^2)$, which is an improvement over previously proposed algorithms.

For the general bilevel optimization problem, we refer the reader to [CSXY22], where the authors summarize the existing bilevel algorithms and propose a STABLE method with $\mathcal{O}(\varepsilon^{-1})$ sample complexity under strong convexity assumption.

4.2 Theoretical background

First, we clarify some notations. We use $\|\cdot\|$ to denote the operator norm of a matrix and $\|\cdot\|_F$ to denote the Frobenius norm of a matrix. When we write $M \geq c$ where M is a symmetric matrix and c is a number, we mean $M - cI$ is positive semi-definite. Similarly, $M > c$ means $M - cI$ is positive definite.

We consider a discrete-time Markov process $\{x_s\}$ on a filtered probability space $(\Omega, \mathcal{F}, \{\mathcal{F}_s\}, \mathbb{P})$:

$$x_{s+1} = Ax_s + Bu_s + \xi_s,$$

where $x_s \in \mathbb{R}^d$ is an adapted state process, $u_s \in \mathbb{R}^k$ is the adapted control process, $A \in \mathbb{R}^{d\times d}$ and $B \in \mathbb{R}^{d\times k}$ are two fixed matrices. $\xi_s \sim N(0, D_\xi)$ is independent noise. The initial state $x_0 \sim \rho_0$, with some initial distribution ρ_0.

The goal is to minimize the infinite horizon cost functional

$$J(\{u_s\}) = \lim_{S\to\infty} \mathbb{E}\left[\frac{1}{S}\sum_{s=0}^{S-1} c(x_s, u_s)\right], \tag{4.1}$$

where $c(x,u) = x^\top Q x + u^\top R u$ is the one-step cost, with $Q \in \mathbb{R}^{d\times d}$ and $R \in \mathbb{R}^{k\times k}$ being positive definite. Theoretical results guarantee that the optimal control u^* is linear in x: $u_s^* = -K^* x_s$. If the model is known, we can obtain the optimal control parameter by $K^* = (R + B^\top P^* B)^{-1} B^\top P^* A$ where P^* is the solution to the Riccati equation [AM07]

$$P^* = Q + A^\top P^* A - A^\top P^* B(R + B^\top P^* B)^{-1} B^\top P^* A. \tag{4.2}$$

In this work, we consider the model-free setting (i.e., the algorithm does not have access to A, B, D_ξ, Q, R). We will use a stochastic policy parametrized as

$$u_s \sim \pi_K := N(-Kx_s, \sigma^2 I_k) \tag{4.3}$$

to encourage exploration, where $\sigma > 0$ is a fixed constant. Here, we use π_K to denote the distribution while we will not distinguish in notation a probability distribution with its density. We remark that adding exploration does not change the optimal K^* because the optimal policy parameters with or without exploration satisfy the same Riccati equation while adding exploration would help the convergence of the algorithm. Under this policy, the cost functional (4.1) is also denoted by $J(K)$ and the state trajectory can be rewritten as

$$x_{s+1} = Ax_s + B(-Kx_s + \sigma\omega_s) + \xi_s =: (A - BK)x_s + \epsilon_s$$

where $\omega_s \sim N(0, I_k)$ and $\epsilon_s \sim N(0, D_\epsilon)$ with $D_\epsilon = D_\xi + \sigma^2 BB^\top$ being positive definite. Let $\rho(\cdot)$ denote the spectral radius of a matrix. When $\rho(A - BK) < 1$, the state process has a stationary distribution $N(0, D_K)$, where $D_K \in \mathbb{R}^{d\times d}$ satisfies the

Lyapunov equation

$$D_K = D_\epsilon + (A - BK)D_K(A - BK)^\top. \tag{4.4}$$

In order to understand (4.4), let us assume that $x \sim N(0, D_K)$ follows the stationary distribution. Then, $x' = (A - BK)x + \epsilon \sim N(0, (A - BK)D_K(A - BK)^\top + D_\epsilon)$ also follows the stationary distribution, which leads to (4.4). D_K can also be expressed in terms of a series: since $\rho(A - BK) < 1$, we can recursively plug in the definition of D_K into the right hand side of (4.4) and obtain

$$D_K = \sum_{s=0}^{\infty} (A - BK)^s D_\epsilon ((A - BK)^\top)^s. \tag{4.5}$$

From here on, the notation $\mathbb{E}_K$ means the expectation with x (or x_0) $\sim N(0, D_K)$ if not specified and u (or u_s) $\sim \pi_K$. The state-action value function (Q function) and the state value function with respect to a control $\{u_s\}$ are defined by

$$\begin{aligned} Q(x, u) &= \sum_{s=0}^{\infty} \left(\mathbb{E}\left[c(x_s, u_s) \mid x_0 = x, u_0 = u\right] - J(\{u_s\})\right) \\ V(x) &= \sum_{s=0}^{\infty} \left(\mathbb{E}\left[c(x_s, u_s) \mid x_0 = x\right] - J(\{u_s\})\right) = \mathbb{E}_u\left[Q(x, u)\right] \end{aligned} \tag{4.6}$$

respectively. $V(x)$ is the expected extra cost if we start at $x_0 = x$ and follow a given policy. $Q(x, u)$ is the expected extra cost if we start at $x_0 = x$, take the first action $u_0 = u$, and then follow a given policy. These two functions are crucial in reinforcement learning. If the policy π_K follows (4.3), then the two functions in (4.6) are denoted by $Q_K(x, u)$ and $V_K(x)$ respectively. By definition, for any x and u, it satisfies the Bellman equation:

$$Q_K(x, u) = c(x, u) - J(K) + \mathbb{E}_K\big[Q_K(x', u') \mid x, u\big], \tag{4.7}$$

where (x', u') is the next state-action pair starting from (x, u).

We define P_K as the solution to the following matrix valued equation

$$P_K = (Q + K^\top R K) + (A - BK)^\top P_K (A - BK). \tag{4.8}$$

P_K can be interpreted as the second order adjoint state, and $P_K x_t$ is the shadow price for the system (see for example [YZ99]). We have the following two properties to illustrate the importance of P_K. The proofs are deferred to later sections.

Proposition 1. *Let the policy π_K be defined by (4.3) with $\rho(A - BK) < 1$. Then the cost function and its gradient w.r.t. K have the following explicit expressions:*

$$J(K) = \operatorname{Tr}(D_\epsilon P_K) + \sigma^2 \operatorname{Tr}(R), \tag{4.9}$$

$$\nabla_K J(K) = 2\left[(R + B^\top P_K B)K - B^\top P_K A\right] D_K. \tag{4.10}$$

Remark. *In the LQR problem, we usually assume that D_K is positive definite and hence invertible. Therefore, the critical point for $J(K)$ (i.e., when $\nabla_K J(K) = 0$) satisfies $K = (R + B^\top P_K B)^{-1} B^\top P_K A$. If we substitute this into (4.8), we recover the Riccati equation (4.2).*

Proposition 2. *Let the policy π_K be defined by (4.3) with $\rho(A - BK) < 1$. Then the value functions have the following explicit expressions:*

$$V_K(x) = x^\top P_K x - \operatorname{Tr}(D_K P_K),$$

$$Q_K(x,u) = \begin{bmatrix} x^\top & u^\top \end{bmatrix} \begin{bmatrix} Q + A^\top P_K A & A^\top P_K B \\ B^\top P_K A & R + B^\top P_K B \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix} - \sigma^2 \operatorname{Tr}(R + P_K B B^\top) - \operatorname{Tr}(D_K P_K). \tag{4.11}$$

If we concatenate x and u in the dynamic equation, the process can be written as

$$\begin{bmatrix} x_{s+1} \\ u_{s+1} \end{bmatrix} = \begin{bmatrix} A & B \\ -KA & -KB \end{bmatrix} \begin{bmatrix} x_s \\ u_s \end{bmatrix} + \begin{bmatrix} \xi_s \\ -K\xi_s + \sigma\omega_s \end{bmatrix}.$$

We simplify the expression by introducing some new notations: $z_s = [x_s^\top, u_s^\top]^\top$, thus $z_{s+1} = Ez_s + \widetilde{\epsilon}_s$, where

$$E = \begin{bmatrix} A & B \\ -KA & -KB \end{bmatrix}, \text{ and } \widetilde{\epsilon}_s \sim N(0, \Sigma_\epsilon) := N\left(0, \begin{bmatrix} D_\xi & -D_\xi K^\top \\ -KD_\xi & KD_\xi K^\top + \sigma^2 I_k \end{bmatrix}\right). \tag{4.12}$$

The ergodicity of the dynamics is guaranteed if $\rho(A - BK) = \rho(E) < 1$, where the identity $\rho(A - BK) = \rho(E)$ can be verified from

$$\rho(E) = \rho\left(\begin{bmatrix} I_d \\ -K \end{bmatrix} \begin{bmatrix} A & B \end{bmatrix}\right) = \rho\left(\begin{bmatrix} A & B \end{bmatrix} \begin{bmatrix} I_d \\ -K \end{bmatrix}\right) = \rho(A - BK).$$

The stationary distribution for z is given by

$$z \sim N(0, \Sigma_K) := N\left(0, \begin{bmatrix} D_K & -D_K K^\top \\ -KD_K & KD_K K^\top + \sigma^2 I_k \end{bmatrix}\right) \tag{4.13}$$

and we have $\Sigma_K = \Sigma_\epsilon + E\Sigma_K E^\top$.

4.3 The actor-critic algorithm

In this section, we present our specific design of the algorithm under the actor-critic framework. We apply an LSTD method for the policy evaluation (critic), with a detailed description for sampling the gradient of the loss function. We also use a natural policy gradient method for the policy improvement (actor). We will use $\mathcal{G}_t$ to denote the filtration generated by the training process. We use $\mathcal{O}(a)$ to denote a quantity that is is bounded by a constant times a, where this constant only depends on the problem setting (A, B, D_ϵ, Q, R, σ) and does not depend on the target accuracy or training trajectory. The dependence of the constants on the dimensions is explained in the proof of our theorem.

4.3.1 Policy evaluation for the critic

In this subsection, we describe the policy evaluation algorithm for a fixed policy π_K. We parametrize the state-action value function by Q_K^θ with θ as a parameter and subscript K indicating that it depends on the given policy π_K. We define the Bellman residual w.r.t. the critic parameter θ as

$$\mathsf{BR}_\theta(x,u) = c(x,u) - J(K) + \mathbb{E}_K\left[Q_K^\theta(x',u')|x,u\right] - Q_K^\theta(x,u).$$

Recall the exact Q function is given by (4.11), accordingly, we define a feature matrix

$$\phi(x,u) = \begin{bmatrix} x \\ u \end{bmatrix}\begin{bmatrix} x^\top & u^\top \end{bmatrix} \in \mathbb{R}^{(d+k)\times(d+k)} \tag{4.14}$$

and parametrize the Q function as

$$Q_K^\theta(x,u) = \mathrm{Tr}(\phi(x,u)\theta) - \theta', \tag{4.15}$$

where $\theta \in \mathbb{R}^{(d+k)\times(d+k)}$ and $\theta' \in \mathbb{R}$. Here, we denote

$$\theta = \begin{bmatrix} \theta^{11} & \theta^{12} \\ \theta^{21} & \theta^{22} \end{bmatrix},$$

which intends to approximate

$$\theta_K = \begin{bmatrix} Q + A^\top P_K A & A^\top P_K B \\ B^\top P_K A & R + B^\top P_K B \end{bmatrix}. \tag{4.16}$$

The scalar parameter θ' is to approximate $\sigma^2\,\mathrm{Tr}(R+P_KBB^\top)+\mathrm{Tr}(D_KP_K)$. Recall the Bellman equation (4.7), with parametrization (4.15), the Bellman residual is written as

$$\begin{aligned}\mathrm{BR}_\theta(x,u) &= c(x,u) - J(K) + \langle \mathbb{E}_K\left[\phi(x',u')|x,u\right] - \phi(x,u), \theta\rangle \\ &=: c(x,u) - J(K) + \langle \psi(x,u), \theta\rangle,\end{aligned}$$

where $\langle \cdot, \cdot \rangle$ is the trace inner product and we have defined

$$\psi(x,u) := \mathbb{E}_K \left[\phi(x',u') | x,u \right] - \phi(x,u)$$

for convenience. It is clear by definition that $\mathbb{E}_K[\psi(x,u)] = 0$ (recall that x follows the stationary distribution $N(0, D_K)$). The loss function for critic is then defined as the expectation of squared Bellman residual:

$$L_K(\theta) = \frac{1}{2}\mathbb{E}_K \left[\mathrm{BR}_\theta(x,u)^2 \right] = \frac{1}{2}\mathbb{E}_K \left[\left(c(x,u) - J(K) + \langle \psi(x,u), \theta \rangle \right)^2 \right]. \tag{4.17}$$

We will find that θ' does not affect the training, so only θ will be considered as the critic parameter from now on. According to the Bellman equation (4.7), the unique minimizer of (4.17) is the true parameter for the Q function w.r.t. π_K. By direct computation, the gradient (as a matrix) and Hessian (as a tensor) of the loss function w.r.t. θ are

$$\begin{aligned} \nabla L_K(\theta) &= \mathbb{E}_K \left[\left(c(x,u) - J(K) + \langle \psi(x,u), \theta \rangle \right) \psi(x,u) \right] \\ &= \mathbb{E}_K \left[\left(c(x,u) + \langle \psi(x,u), \theta \rangle \right) \psi(x,u) \right] \end{aligned} \tag{4.18}$$

and

$$\nabla^2 L_K(\theta) = \mathbb{E}_K \left[\psi(x,u) \otimes \psi(x,u) \right],$$

where $\otimes$ denotes the tensor product. The loss function L_K is strongly convex in θ, as will be shown later.

To minimize the loss (4.17), we use stochastic gradient descent method. Thus, we need an accurate sample estimate of $\nabla L_K(\theta)$ for given K and θ. For simplicity of notation, we denote

$$f(x,u) := \left(c(x,u) + \langle \psi(x,u), \theta \rangle \right) \psi(x,u) = c(x,u)\psi(x,u) + \left(\psi(x,u) \otimes \psi(x,u) \right) \cdot \theta \tag{4.19}$$

so that $\nabla L_K(\theta) = \mathbb{E}_K[f(x,u)]$. Note that $f(x,u)$ depends on θ and K, while we suppress that in the notation. We decompose the sampling into two steps: the first step is to obtain x, u that approximately follows the stationary distribution $N(0, \Sigma_K)$ and the second one is to sample $f(x,u)$ for given x, u.

For the first step, we use the Markov chain Monte Carlo (MCMC) method [GRS95]. Let N_0 and N be two integers that will be determined according to the error tolerance. Starting at $x_0 = 0$, we sample N independent trajectories of length $N_0 + 1$ according to the policy π_K. So, we obtain N samples $\{(x_{N_0}^{(i)}, u_{N_0}^{(i)})\}_{i=1}^N$ that follow the distribution of (x_{N_0}, u_{N_0}). For each pair $(x_{N_0}^{(i)}, u_{N_0}^{(i)})$, we generate N_1 unbiased sample for $\psi(x_{N_0}^{(i)}, u_{N_0}^{(i)})$, given by

$$\widehat{\psi}_j^{(i)} = \phi(x^{(i,j)}, u^{(i,j)}) - \phi(x_{N_0}^{(i)}, u_{N_0}^{(i)}) \qquad j = 1, 2, \cdots, N_1$$

where $x^{(i,j)}, u^{(i,j)}$ are sampled independently and follow the next step distribution conditioned on $(x_{N_0}^{(i)}, u_{N_0}^{(i)})$. Here, $N_1 = \mathcal{O}(1)$ is another predefined hyperparameter. We denote the mean by $\bar{\psi}^{(i)} = \frac{1}{N_1}\sum_{j=1}^{N_1} \widehat{\psi}_j^{(i)}$. Therefore, we can obtain an unbiased sample for $f(x_{N_0}^{(i)}, u_{N_0}^{(i)})$ by

$$\begin{aligned}\widehat{f}(x_{N_0}^{(i)}, u_{N_0}^{(i)}) =& \frac{1}{N_1}\sum_{j=1}^{N_1} c(x_{N_0}^{(i)}, u_{N_0}^{(i)})\widehat{\psi}_j^{(i)} \\ &+ \left[\frac{1}{N_1}\sum_{j=1}^{N_1} \widehat{\psi}_j^{(i)} \otimes \widehat{\psi}_j^{(i)} - \frac{1}{N_1 - 1}\sum_{j=1}^{N_1} (\widehat{\psi}_j^{(i)} - \bar{\psi}^{(i)}) \otimes (\widehat{\psi}_j^{(i)} - \bar{\psi}^{(i)})\right] \cdot \theta.\end{aligned} \tag{4.20}$$

Note that the first and second terms in the square bracket are unbiased samples for $\mathbb{E}[\widehat{\psi}_j^{(i)} \otimes \widehat{\psi}_j^{(i)}]$ and $\mathrm{Cov}(\widehat{\psi}_j^{(i)})$ respectively, which implies that the square bracket is an unbiased sample for $\psi(x_{N_0}^{(i)}, u_{N_0}^{(i)}) \otimes \psi(x_{N_0}^{(i)}, u_{N_0}^{(i)})$. Finally, the sample of gradient

$\nabla L_K(\theta)$ is given by

$$\widehat{\nabla L}_K(\theta) = \frac{1}{N}\sum_{i=1}^{N} \widehat{f}(x_{N_0}^{(i)}, u_{N_0}^{(i)}). \tag{4.21}$$

The one-step sample complexity is $\mathcal{O}(N_0 N_1 N)$. We remark that our LSTD is similar to a TD(0) algorithm, except that we have N trajectories and we omit $J(K)$ in (4.18). Denote $L_{K_t}(\theta)$ by $L_t(\theta)$ for simplicity. We also denote θ_t the critic parameter at step t. The gradient sample at step t (in matrix form) is denoted by $\widehat{\nabla L_t}(\theta_t)$ and the critic update is given by

$$\theta_{t+1} = \theta_t - \alpha_t \widehat{\nabla L_t}(\theta_t),$$

where α_t is the step size for the critic.

4.3.2 Policy improvement for the actor

For the actor algorithm, we borrow the idea from [FGKM18] which considered a policy gradient algorithm for the LQR problem. A similar approach is also studied by [YCHW19, ZDR21].

Motivated by the form of the gradient (4.10), we define

$$G_K := (R + B^\top P_K B)K - B^\top P_K A, \tag{4.22}$$

so that $\nabla_K J(K) = 2G_K D_K$. Therefore, a vanilla policy gradient algorithm looks like

$$K_{t+1} = K_t - \beta_t G_{K_t} D_{K_t},$$

where G_{K_t} and D_{K_t} may be replaced by some estimates and β_t is the step size for the actor.

Instead of the vanilla policy gradient, we would consider the commonly used variant known as the natural policy gradient method [Kak01]. The natural policy gradient uses the inverse Fisher information matrix to precondition the gradient so

that the gradient is taken w.r.t. the metric induced by the Hessian of the loss function [PS08]. This method has been studied in e.g., [Kak01, PS08, BSGL09, LZBY20]. The Fisher information matrix at each state x is given by

$$F_x(K) = \mathbb{E}_{u\sim\pi_K}\left[\nabla_K \log(\pi_K(u|x)) \otimes \nabla_K \log(\pi_K(u|x))\right], \tag{4.23}$$

which is a tensor in $\mathbb{R}^{k\times d}\otimes\mathbb{R}^{k\times d}$ as $K \in \mathbb{R}^{k\times d}$ is a matrix. Then, the (average) Fisher information matrix is defined as

$$F(K) = \mathbb{E}_{x\sim N(0,D_K)}\left[F_x(K)\right] = \mathbb{E}_K\left[\nabla_K \log(\pi_K(u|x)) \otimes \nabla_K \log(\pi_K(u|x))\right].$$

Under the metric induced by the Hessian, the steepest descent direction of $J(K)$ is given by

$$-\widetilde{\nabla} J(K) = -F(K)^{-1}\cdot \nabla_K J(K) = -2F(K)^{-1}\cdot G_K D_K,$$

where for $F(K)^{-1}$, we view the tensor $F(K)$ as a linear operator $\mathbb{R}^{k\times d}\to\mathbb{R}^{k\times d}$, so $F(K)^{-1}$ is the inverse operator. The following property gives a simple expression of $\widetilde{\nabla} J(K)$. The proof is in later sections.

Proposition 3. *We have*

$$\widetilde{\nabla} J(K) = 2\sigma^2 G_K. \tag{4.24}$$

Recall that $G_K = (R + B^\top P_K B)K - B^\top P_K A$. Hence, $G_K = \theta_K^{22}K - \theta_K^{21}$ where θ_K is the true parameter w.r.t. policy π_K, given by (4.16). Therefore, the actor update is given by

$$K_{t+1} = K_t - \beta_t(\theta_t^{22}K_t - \theta_t^{21}) =: K_t - \beta_t \widehat{G}_{K_t}, \tag{4.25}$$

where the constant $2\sigma^2$ is absorbed in the step size β_t and we have defined $\widehat{G}_{K_t} := \theta_t^{22}K_t - \theta_t^{21}$. Recall that we use $\mathcal{G}_t$ to denote the filtration generated by the training process. Since K_{t+1} is deterministic in θ_t and K_t, K_{t+1} is $\mathcal{G}_t$-measurable.

4.3.3 Assumptions and main result

Here we state some technical assumptions for our result.

Assumption 1. *We assume that*

1. *There exists a constant $\rho \in (0,1)$ such that $\rho(A - BK_t) = \rho(E_t) \le \rho$, for all t.*
2. *There exist constants $c_A, c_E, c_\theta, c_K > 0$ such that $\|A - BK_t\| \le c_A$, $\|E_t\| \le c_E$, $\|\theta_t\|_F \le c_\theta$, and $\|K^*\|, \|K_t\| \le c_K$ for all t.*
3. *D_ϵ is positive definite with minimum eigenvalue $\sigma_{min}(D_\epsilon) > 0$.*

Remark. *In the assumption, E_t is defined by* (4.12) *with K replaced by K_t. The first assumption is common in the analysis of the LQR problem [FGKM18, YCHW19]. A theoretical guarantee for this condition is hard to obtain, while we will present some numerical examples to support this assumption. The second assumption gives upper bounds for several matrices, which is made to avoid technical tedious works to control the probability of the random trajectory hitting unfavorable regions. One potential way to alleviate this assumption is to define a projection map that reduces the size of θ_t or K_t whenever it is out of range [KT00, BSGL09], which is left for future work. The third assumption is necessary to make the problem non-degenerate (cf. Lemma 7 below).*

Next, we specify the choice of parameters in the algorithm. We initialize $\theta_0 = 0$, $K_0 = 0$ for simplicity. Fixing the error tolerance $\varepsilon > 0$, we set the step sizes α_t and β_t to be constant in t:

$$\alpha_t = \frac{\sigma_{min}(D_\epsilon)}{16 c_L^2 c_3 \kappa}\varepsilon \qquad \beta_t = \frac{\sigma_{min}(D_\epsilon)}{16 c_L^2 c_3 \kappa^2}\varepsilon \tag{4.26}$$

where

$$\kappa = \max\left(\frac{3\sigma_{min}(D_\epsilon)}{2c_3\mu_\sigma}, \frac{4c_1^2}{\mu_\sigma \sigma_{min}(D_\epsilon)}, \frac{3c_D c_K^2}{\mu_\sigma}\right). \tag{4.27}$$

Here, every parameter appearing in (4.26) and (4.27), except α_t, β_t, or ε, are constants of order $\mathcal{O}(1)$:

1. c_L^2 is the upper bound for $\mathbb{E}[\|\widehat{\nabla L_t}(\theta_t)\|_F^2 \mid \mathcal{G}_t]$ that is in Lemma 3;
2. c_3 illustrates the geometry of $J(K)$, with details in Lemma 6;
3. In Lemma 2, we will show that the critic loss is μ_σ-strongly convex;
4. c_1 is a Lipschitz constant for θ_K w.r.t. K that is specified in Lemma 4;
5. c_D is an upper bound for $\|D_{K_t}\|$ and $\|D_{K^*}\|$ that is specified in Lemma 1.

It is easy to verify that the step sizes satisfies the following inequalities:

$$\begin{aligned}
&\frac{\sigma_{min}(D_\epsilon)}{c_3}\beta_t \leq \frac{2}{3}\mu_\sigma\alpha_t \\
&\frac{\sigma_{min}(D_\epsilon)}{\beta_t} \geq (\frac{3}{\alpha_t\mu_\sigma}+2)c_1^2 + (\|R\| + c_P\|B\|^2) \\
&\frac{1}{3}\alpha_t\mu_\sigma \geq \beta_t c_D c_K^2,
\end{aligned} \tag{4.28}$$

where we need to assume that ε is small enough such that $1/(\mu_\sigma\alpha_t) \geq 2 + (\|R\| + c_P\|B\|^2)/c_1^2$ for the second inequality. The total number of iterations is

$$T = \mathcal{O}(\frac{1}{\varepsilon}\log(\frac{1}{\varepsilon}))$$

such that

$$(1-\beta_t c_4)^T L_0 < \varepsilon,$$

where $L_0 = \mathcal{O}(1)$ is the initial Lyapunov function that is specified at the beginning of the proof for Theorem 1 and $c_4 = \mathcal{O}(1)$ is a positive constant that is also specified in the proof for Theorem 1. The number of samples N, the length of trajectory N_0 each step, and the sub-sample size N_1, are set to be $N = \mathcal{O}(1)$, $N_0 = \mathcal{O}(\log(\frac{1}{\varepsilon}))$, and

$N_1 = \mathcal{O}(1)$, in order to achieve desired accuracy for the sample of critic gradient, with details in Lemma 3. Here, $\frac{\alpha_t}{\beta_t} = \kappa = \mathcal{O}(1)$ implies that our algorithm has single timescale. In such algorithm, the actor and the critic are interdependent, which makes the analysis challenging. We summarize the actor-critic algorithm in Algorithm 3.

Algorithm 3 Single timescale actor-critic algorithm for LQR

Input: Training steps T, step sizes α_t, β_t, sample size N, N_0, and N_1
Output: critic parameter θ_T, actor parameter K_T
initialization: critic parameter $\theta_0 = 0$ and actor parameter $K_0 = 0$
for $t = 0$ **to** $T - 1$ **do**
Sample $\widehat{\nabla L_t}(\theta_t)$ according to (4.21) {critic steps}
$\theta_{t+1} = \theta_t - \alpha_t \widehat{\nabla L_t}(\theta_t)$
$K_{t+1} = K_t - \beta_t(\theta_t^{22} K_t - \theta_t^{21})$ {actor steps}
end for

The main result of our work is the following convergence theorem.

Theorem 1 (Main theorem)**.** *Under Assumption 1, for any $\varepsilon > 0$ that is sufficiently small, Algorithm 3, with the choice of parameters discussed above, has sample complexity $\mathcal{O}(\frac{1}{\varepsilon}\log(\frac{1}{\varepsilon})^2)$. Moreover, the terminal error satisfies*

$$\mathbb{E}[\|\theta_T - \theta_{K_T}\|_F^2] \leq \varepsilon \quad \textit{and} \quad \mathbb{E}[J(K_T) - J(K^*)] \leq \varepsilon.$$

Remark. *The number of steps is $T = \mathcal{O}(\frac{1}{\varepsilon}\log(\frac{1}{\varepsilon}))$ and the one-step complexity is $\mathcal{O}(\log(\frac{1}{\varepsilon}))$. Therefore, the total complexity is $\mathcal{O}(\frac{1}{\varepsilon}\ \log(\frac{1}{\varepsilon})^2)$. This theorem tells us that we have small error for both the critic and the actor. If we want error estimate for $\|K_T - K^*\|_F$ or $\|\theta_T - \theta^*\|_F$, we will need extra assumption such as strong convexity of $J(K)$ in K.*

We believe the complexity $\mathcal{O}(\frac{1}{\varepsilon}\log(\frac{1}{\varepsilon})^2)$ is nearly optimal (up to a log factor). Even for a simple stochastic gradient descent (SGD) algorithm, we need $\mathcal{O}(\varepsilon^{-1})$ sample to achieve ε-optimal solution [Bot12]. The LQR problem is bilevel, with the critic part similar to SGD. Thus, the problem is more complicated than SGD and

expects to require higher sample complexity. The convergence rate is also confirmed by the numerical examples below.

4.4 Proof sketch of the main theorem

In this section, we give a sketch of the proof of Theorem 1 and postpone the details to later sections. The lemmas used in the proof are stated in the later part of this section.

Proof Sketch of Theorem 1. First, we show in Lemma 2 that the critic loss is strongly convex. Then, we show in Lemma 3 that we can obtain the sample of gradient with small bias:

$$\left\| \mathbb{E}\left[\widehat{\nabla L_t}(\theta_t) - \nabla L_t(\theta_t) | \mathcal{G}_t \right] \right\|_F \leq \delta$$

With these two lemmas, we show in Lemma 5 that there is an improvement of critic error in each step:

$$\begin{aligned} &\mathbb{E}\left[\|\theta_{t+1} - \theta_{K_{t+1}}\|_F^2 | \mathcal{G}_t \right] - \|\theta_t - \theta_{K_t}\|_F^2 \\ &\quad \leq -\frac{4}{3}\alpha_t \mu_\sigma \|\theta_t - \theta_{K_t}\|_F^2 + \frac{1}{4}\frac{\sigma_{min}(D_\epsilon)}{c_3}\beta_t \varepsilon + \big(\frac{3}{\alpha_t \mu_\sigma} + 2\big)\|\theta_{K_t} - \theta_{K_{t+1}}\|_F^2. \end{aligned} \tag{4.29}$$

Here, the term $\frac{1}{4}\frac{\sigma_{min}(D_\epsilon)}{c_3}\beta_t\varepsilon$ comes from the sample error in Lemma 3 and $(\frac{3}{\alpha_t\mu_\sigma} + 2)\|\theta_{K_t} - \theta_{K_{t+1}}\|_F^2$ is due to the actor update. Intuitively, we expect $\|\theta_{t+1} - \theta_{K_t}\|_F$ to be smaller than $\|\theta_t - \theta_{K_t}\|_F$, recall that $\|\theta_t - \theta_{K_t}\|_F$ measures the error of θ_t w.r.t. the current policy parameter K_t, while the last term in (4.29) takes into account the update of K_t to K_{t+1} in the actor step.

Furthermore, we establish the improvement of the actor in Lemma 7:

$$\begin{aligned} J(K_{t+1}) - J(K_t) &\leq -\beta_t \frac{\sigma_{min}(D_\epsilon)}{c_3}(J(K_t) - J(K^*)) \\ &\quad - \beta_t \left[\sigma_{min}(D_\epsilon) - \beta_t c_D(\|R\| + c_P\|B\|^2)\right] \|\widehat{G}_{K_t}\|_F^2 + \beta_t c_D \|G_{K_t} - \widehat{G}_{K_t}\|_F^2 \end{aligned} \tag{4.30}$$

where the last term comes from the critic error.

To establish the convergence, we define a Lyapunov function

$$\mathcal{L}_t = \mathcal{L}(\theta_t, K_t) := \|\theta_t - \theta_{K_t}\|_F^2 + J(K_t) - J(K^*),$$

which is the sum of critic and actor errors. Direct computation shows that the last term in (4.29) can be bounded by the second term in (4.30) and the last term in (4.30) can be bounded by $\frac{1}{4}$ of the first term in (4.29). Therefore, combining (4.29) and (4.30), we obtain the decay estimate of the Lyapunov function

$$\begin{aligned}\mathbb{E}[\mathcal{L}_{t+1} - \mathcal{L}_t] \leq -\mathbb{E}\left[\alpha_t \mu_\sigma \|\theta_t - \theta_{K_t}\|_F^2 + \beta_t \frac{\sigma_{min}(D_\epsilon)}{c_3}(J(K_t) - J(K^*))\right] \\ + \frac{1}{4}\frac{\sigma_{min}(D_\epsilon)}{c_3}\beta_t \varepsilon.\end{aligned} \tag{4.31}$$

Notice that the last term (sample error) in (4.31) can be bounded by the first term if $\mathbb{E}[\|\theta_t - \theta_{K_t}\|_F^2] \geq \frac{\varepsilon}{2}$ (according to the first inequality of (4.28)) or by the second term if $\mathbb{E}[J(K_t) - J(K^*)] \geq \frac{\varepsilon}{2}$ and we will obtain a contraction rate for the Lyapunov function:

$$\mathcal{L}_{t+1} - \mathcal{L}_t \leq -\mathcal{O}(\beta_t)\mathcal{L}_t.$$

If both $\mathbb{E}[\|\theta_t - \theta_{K_t}\|_F^2] < \frac{\varepsilon}{2}$ and $\mathbb{E}[J(K_t) - J(K^*)] < \frac{\varepsilon}{2}$, then $\mathbb{E}[\mathcal{L}_t] < \varepsilon$ and we can easily show that $\mathbb{E}[\mathcal{L}_{t+1}]$ is also less than ε. This finishes the proof. □

In summary, the key point of the proof is that we can bound the positive term in the critic improvement by the negative term in the actor improvement and vice versa. In this way, we obtain a contraction rate of the Lyapunov function.

Before we turn to the analysis of critic and actor parts, we state the following lemma which provides bounds for matrices D_{K_t}, P_{K_t}, and Σ_{K_t}.

Lemma 1. *Under Assumption 1, the matrix D_{K_t}, P_{K_t} and Σ_{K_t} satisfy*

$$\sigma_{min}(D_\epsilon) \leq D_{K_t} \leq c_D, \quad P_{K_t} \leq c_P, \quad \textit{and} \quad \Sigma_{K_t} \leq c_\Sigma \tag{4.32}$$

where the three constants $c_D, c_P, c_\Sigma = \mathcal{O}(1)$ *only depend on* A, B, D_ϵ, Q, R, ρ, σ, *and* c_A. *Furthermore, the first inequality also holds with* D_{K_t} *replaced by* D_{K^*}.

4.4.1 Analysis of the critic part

In this subsection, we analyze the critic part of the algorithm. All the proofs are deferred to later sections. Let us start with the following lemma, which gives the strong convexity property of the critic loss.

Lemma 2 (Strong convexity of critic loss)**.** *Suppose that* $\rho(E) \le \rho < 1$, $L_K(\theta)$ *is* μ_σ*-strongly convex in* θ, *where* $\mu_\sigma > 0$ *only depends on* A, B, D_ϵ, ρ, σ, c_K, *and* c_Σ. *Moreover,* $\mu_\sigma = \mathcal{O}(\sigma^4)$ *when* σ *is small.*

Actually, one technical reason of using a stochastic policy for exploration is to guarantee the strong convexity. The next lemma gives a quantitative description of the accuracy of critic gradient sampling proposed in §4.3.1.

Lemma 3 (Gradient sample accuracy)**.** *Under Assumption 1, for any* $\delta > 0$ *that is sufficiently small, let* $\widehat{\nabla L_t}(\theta_t)$ *be the sample of* $\nabla L_t(\theta_t)$ *with complexity* $N, N_1 = \mathcal{O}(1)$ *and* $N_0 = \mathcal{O}(\log \frac{1}{\delta})$. *Then, we have*

$$\left\| \mathbb{E}\left[\widehat{\nabla L_t}(\theta_t) - \nabla L_t(\theta_t) \;\middle|\; \mathcal{G}_t \right] \right\|_F \le \delta \tag{4.33}$$

and

$$\mathbb{E}\left[\|\widehat{\nabla L_t}(\theta_t)\|_F^2 \;\middle|\; \mathcal{G}_t \right] \le c_L^2, \tag{4.34}$$

where $c_L = \mathcal{O}(1)$ *is a positive constant that only depends on* A, B, D_ϵ, Q, R, σ, c_K, *and* c_θ.

Remark. *When we apply this lemma later, we will set* $\delta^2 = \frac{1}{24}\frac{\sigma_{min}(D_\epsilon)}{\kappa c_3}\mu_\sigma \varepsilon$, *and thus* $\delta = \mathcal{O}(\varepsilon^{\frac{1}{2}})$. *By definition of the step sizes* (4.26), *we have*

$$2\alpha_t^2 \mathbb{E}\left[\|\widehat{\nabla L_t}(\theta_t)\|_F^2 \;\middle|\; \mathcal{G}_t \right] \le \frac{1}{8}\beta_t \frac{\sigma_{min}(D_\epsilon)}{c_3}\varepsilon. \tag{4.35}$$

when (4.34) *holds. This inequality* (4.35) *will be used later and we can see that the step size has to be of order* $\mathcal{O}(\varepsilon)$ *to guarantee* (4.35).

Next, we show a Lipschitz property for θ_K with respect to K.

Lemma 4. *For any two actor parameters* K *and* K' *such that* $\|K\|, \|K'\| \leq c_K$, $\|A - BK\|, \|A - BK'\| \leq c_A$, *and* $\rho(A - BK), \rho(A - BK') \leq \rho < 1$, *we have*

$$\|\theta_K - \theta_{K'}\|_F \leq c_1 \|K - K'\|_F,$$

where the constant $c_1 = \mathcal{O}(1)$ *only depends on* A, B, R, ρ, c_A, c_K, *and* c_P.

With the above lemmas, we can establish the improvement by the critic update.

Lemma 5. *Let the step size be defined as in* (4.26) *and Assumption 1 hold. For any* $\varepsilon > 0$ *that is sufficiently small, assume that* (4.33) *and* (4.34) *hold with* $\delta^2 = \frac{1}{24} \frac{\sigma_{min}(D_\epsilon)}{\kappa c_3} \mu_\sigma \varepsilon$ *for all* t, *then we have*

$$\begin{aligned} &\mathbb{E}\left[\|\theta_{t+1} - \theta_{K_{t+1}}\|_F^2 \;\middle|\; \mathcal{G}_t\right] - \|\theta_t - \theta_{K_t}\|_F^2 \\ &\qquad \leq -\frac{4}{3}\alpha_t \mu_\sigma \|\theta_t - \theta_{K_t}\|_F^2 + \frac{1}{4}\frac{\sigma_{min}(D_\epsilon)}{c_3}\beta_t \varepsilon + \big(\frac{3}{\alpha_t \mu_\sigma} + 2\big)\|\theta_{K_t} - \theta_{K_{t+1}}\|_F^2. \quad (4.36) \end{aligned}$$

Recall that K_{t+1} is $\mathcal{G}_t$-measurable.

4.4.2 Analysis of the actor part

In this subsection, we give the convergence result for the actor part. All proofs are deferred to later sections. The first lemma demonstrates that the cost functional is roughly quadratic in G_K. Inequality (4.37) has also been established in earlier works [FGKM18, FYW20].

Lemma 6. *Let* K *be an actor parameter such that* $\rho(A - BK) < 1$, *we have*

$$c_2 \operatorname{Tr}(G_K G_K^\top) \leq J(K) - J(K^*) \leq c_3 \operatorname{Tr}(G_K G_K^\top), \quad (4.37)$$

with positive constants $c_2 = \frac{\sigma_{min}(D_\epsilon)}{\|R\| + c_P \|B\|^2}$ *and* $c_3 = \frac{\|D_{K^*}\|}{\sigma_{min}(R)}$.

We recall that $\|\cdot\|$ denotes the operator norm of a matrix. We also recall that K^* is the optimal control parameter that is given by $K^* = (R + B^\top P^* B)^{-1} B^\top P^* A$ (see (4.2) for definition of P^*). Next lemma establishes the improvement of the actor update.

Lemma 7 (Improvement in the actor update)**.** *Let the actor update be defined by* (4.25) *and Assumption 1 hold, then*

$$J(K_{t+1}) - J(K_t) \leq -\beta_t \frac{\sigma_{min}(D_\epsilon)}{c_3}(J(K_t) - J(K^*))$$
$$- \beta_t \left[\sigma_{min}(D_\epsilon) - \beta_t c_D(\|R\| + c_P\|B\|^2)\right] \|\widehat{G}_{K_t}\|_F^2 + \beta_t c_D \|G_{K_t} - \widehat{G}_{K_t}\|_F^2$$

Remark. *This lemma for actor improvement is a generalization of Lemma 15 in [FGKM18]. Their lemma shows an improvement of policy gradient with accurate critic, while our lemma shows that there are extra terms when we have stochastic estimate of the critic.*

4.5 Numerical examples

In this section, we present some numerical examples to validate our theoretical results. We consider two examples: the first one has $d = 2$ and $k = 3$:

$$A = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix}, \quad B = \begin{bmatrix} 0.2 & 0 & 0.1 \\ 0 & 0.2 & 0.1 \end{bmatrix}, \quad Q = \begin{bmatrix} 1 & 0 \\ 0 & 0.8 \end{bmatrix},$$

$$R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0.5 \end{bmatrix}, \quad D_\xi = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},$$

and $\sigma = 1$. The other one has $d = 4$ and $k = 3$:

$$A = \begin{bmatrix} 0.5 & 0.1 & 0 & 0 \\ 0.1 & 0.5 & 0.1 & 0 \\ 0 & 0.1 & 0.5 & 0 \\ 0 & 0 & 0 & 0.5 \end{bmatrix}, \quad B = \begin{bmatrix} 0.3 & 0.1 & 0 \\ 0.1 & 0.3 & 0.1 \\ 0 & 0.1 & 0.3 \\ 0.1 & 0.1 & 0.1 \end{bmatrix}, \quad Q = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0.1 & 0 \\ 0 & 0.1 & 1 & 0.1 \\ 0 & 0 & 0.1 & 1 \end{bmatrix},$$

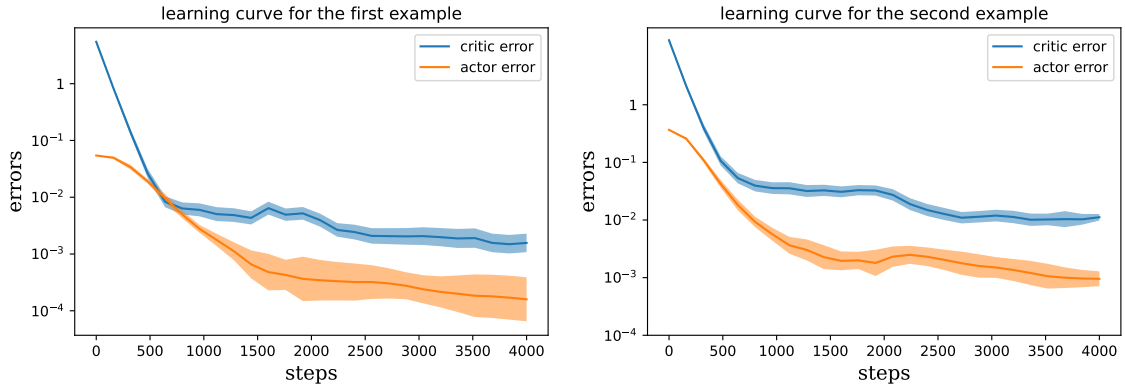

Figure 4.1: Error curves.

$$R = \begin{bmatrix} 1 & 0.1 & 0 \\ 0.1 & 1 & 0.1 \\ 0 & 0.1 & 1 \end{bmatrix}, \quad D_\xi = \begin{bmatrix} 1 & 0 & 0.1 & 0 \\ 0 & 1 & 0 & 0 \\ 0.1 & 0 & 1 & 0.1 \\ 0 & 0 & 0.1 & 1 \end{bmatrix},$$

and $\sigma = 1$. In all the tests, we set $N = N_0 = N_1 = 100$ for simplicity. We test for $T = 125, 250, 500, 1000, 2000, 4000$. In each example, we set the step sizes to be $\alpha_t = \beta_t = \frac{4}{T}$. In order to save time, we multiply the step sizes by 3 for the first $T/2$ steps.

Figure 4.1 shows the learning curves for the two example with step size $\alpha_t = \beta_t = 0.001$. The error is the average of 10 independent runs, and we also show the standard deviations. In the beginning, the error curves are nearly straight lines, which coincide with our one-step improvement analysis in the previous section. Then the errors become static because the algorithm has reached its capacity.

In order to obtain a convergence rate, we also test different step sizes, which is shown in Figure 4.2. In the tests, we keep $T\alpha_t = T\beta_t$ as a constant. The horizontal axis marks the number of steps T, ranging from 125 to 4000. We take a log_2 transform of T. The vertical axis is the final critic and actor errors (after a log_2 transform). A linear regression indicates that the slopes of the four error curves are all -1.0, which confirms our theoretical results in the previous section.

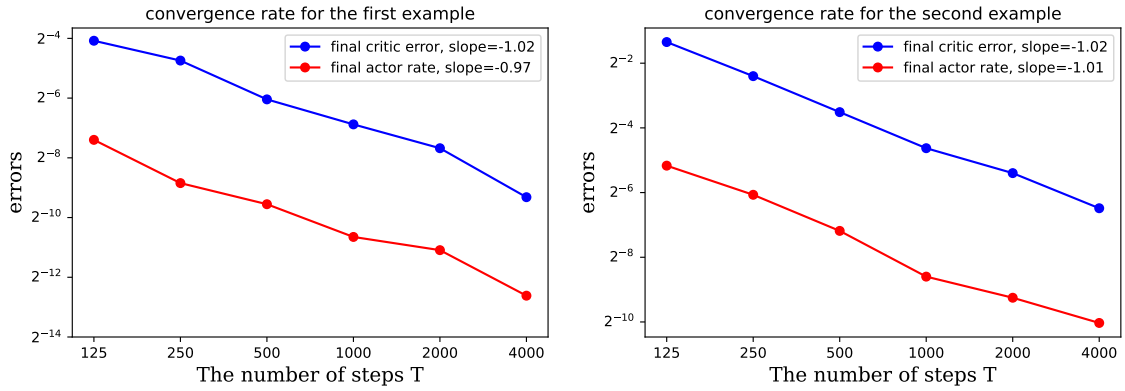

Figure 4.2: Convergence rate.

We also track the norm in Assumption 1. In the numerical tests, the maximum of $\rho(A - BK_t)$, $\|A - BK_t\|$, $\|E_t\|$, $\|K_t\|$, and $\|\theta_t\|_F$ for the first and second examples are 0.524, 0.529, 0.586, 0.329, 2.641 and 0.662, 0.662, 0.867, 0.498, 4.254 respectively. This further confirms that Assumption 1 is reasonable.

4.6 Proofs for the results

Throughout the proof, we will frequently use two basic properties in linear algebra. So we state them here. The first one is that if X is a (symmetric and) positive semi-definite matrix and Y is of the same shape, then $\mathrm{Tr}(XY) \leq \mathrm{Tr}(X)\|Y\|$, where we recall that $\|\cdot\|$ is the operator norm of a matrix. The second property is a direct corollary of the first one: for any matrices X and Y of proper shapes, we have $\|XY\|_F \leq \|X\| \, \|Y\|_F$

4.6.1 Proofs for results in Section 4.2 and Section 4.3

Proof of Proposition 1. Since $\rho(A - BK) < 1$, we know from definition (4.8) that the expression for P_K in series is

$$P_K = \sum_{s=0}^{\infty} ((A - BK)^\top)^s (Q + K^\top R K)(A - BK)^s. \tag{4.38}$$

Give the state x_s, the conditional expectation of one-step cost is

$$\begin{aligned}\mathbb{E}[c(x_s,u_s)|x_s] &= x_s^\top Q x_s + \mathbb{E}_{\omega_s\sim N(0,Id)}[(-Kx_s+\sigma\omega_s)^\top R(-Kx_s+\sigma\omega_s)] \\ &= x_s^\top (Q+K^\top RK)x_s + \sigma^2\,\mathrm{Tr}(R).\end{aligned} \tag{4.39}$$

So the total cost is

$$\begin{aligned}&J(K)\\ &= \lim_{S\to\infty}\mathbb{E}_K\left[\frac{1}{S}\sum_{s=0}^{S-1}c(x_s,u_s)\right] = \lim_{S\to\infty}\mathbb{E}_K\left[\frac{1}{S}\sum_{s=0}^{S-1}\mathbb{E}[c(x_s,u_s)|x_s]\right]\\ &= \lim_{S\to\infty}\mathbb{E}_K\left[\frac{1}{S}\sum_{s=0}^{S-1}x_s^\top(Q+K^\top RK)x_s\right] + \sigma^2\,\mathrm{Tr}(R)\\ &= \mathbb{E}_K[x^\top(Q+K^\top RK)x] + \sigma^2\,\mathrm{Tr}(R)\\ &= \mathrm{Tr}\left[\mathbb{E}_K[xx^\top](Q+K^\top RK)\right] + \sigma^2\,\mathrm{Tr}(R) = \mathrm{Tr}\left[D_K(Q+K^\top RK)\right] + \sigma^2\,\mathrm{Tr}(R)\\ &= \mathrm{Tr}\left[D_K(P_K-(A-BK)^\top P_K(A-BK))\right] + \sigma^2\,\mathrm{Tr}(R)\\ &= \mathrm{Tr}\left[(D_K-(A-BK)D_K(A-BK)^\top)P_K\right] + \sigma^2\,\mathrm{Tr}(R) = \mathrm{Tr}[D_\epsilon P_K] + \sigma^2\,\mathrm{Tr}(R).\end{aligned}$$

So (4.9) holds. Next, we derive the expression for $\nabla_K J(K)$. We need a simple formula: if the shape of M is the same as the shape of K, then

$$\nabla_K\,\mathrm{Tr}(M^\top K) = \nabla_K\,\mathrm{Tr}(MK^\top) = M.$$

Since $J(K) = \mathrm{Tr}\left[D_K(Q+K^\top RK)\right] + \sigma^2\,\mathrm{Tr}(R)$, we have

$$\nabla_K J(K) = 2RKD_K + \nabla_K\,\mathrm{Tr}[D_K Q_0]|_{Q_0=Q+K^\top RK}. \tag{4.40}$$

We recall that

$$D_K = D_\epsilon + (A-BK)D_K(A-BK)^\top.$$

Therefore,

$$\begin{aligned}
&\nabla_K \operatorname{Tr}[D_K Q_0] = \nabla_K \operatorname{Tr}[(D_\epsilon + (A-BK)D_K(A-BK)^\top)Q_0] \\
&= -B^\top(Q_0 + Q_0^\top)(A-BK)D_K + \nabla_K \operatorname{Tr}[D_K Q_1]|_{Q_1=(A-BK)^\top Q_0(A-BK)} \\
&= -2B^\top Q_0(A-BK)D_K + \nabla_K \operatorname{Tr}[D_K Q_1]|_{Q_1=(A-BK)^\top Q_0(A-BK)}
\end{aligned} \tag{4.41}$$

where we used $Q_0 = Q_0^\top$ in the last equality. Therefore, we can apply (4.41) recursively and obtain

$$\begin{aligned}
&\nabla_K \operatorname{Tr}[D_K Q_0]|_{Q_0=Q+K^\top RK} \\
&= -2B^\top(Q+K^\top RK)(A-BK)D_K \\
&\quad + \nabla_K \operatorname{Tr}[D_K Q_1]|_{Q_1=(A-BK)^\top(Q+K^\top RK)(A-BK)} \\
&= -2B^\top(Q+K^\top RK)(A-BK)D_K \\
&\quad - 2B^\top(A-BK)^\top(Q+K^\top RK)(A-BK)^2 D_K \\
&\quad + \nabla_K \operatorname{Tr}[D_K Q_2]|_{Q_2=((A-BK)^\top)^2(Q+K^\top RK)(A-BK)^2} \\
&= \cdots \\
&= -\sum_{s=0}^{\infty} 2B^\top((A-BK)^\top)^s(Q+K^\top RK)(A-BK)^{s+1}D_K \\
&= -2B^\top P_K(A-BK)D_K
\end{aligned} \tag{4.42}$$

where the assumption $\rho(A-BK) < 1$ guarantees that the series converges and the remaining term vanishes. Substituting (4.42) into (4.40), we obtain

$$\nabla_K J(K) = 2RKD_K - 2B^\top P_K(A-BK)D_K = 2\left[(R+B^\top P_K B)K - B^\top P_K A\right]D_K.$$

□

Proof of Proposition 2. If we start with $x_0 = x$, since the state dynamic is

$$x_{s+1} = (A-BK)x_s + \epsilon_s$$

with $\epsilon_s \sim N(0, D_\epsilon)$, the state distribution is

$$\begin{aligned} x_s &\sim N\left((A-BK)^s x,\ \sum_{i=0}^{s-1}(A-BK)^i D_\epsilon((A-BK)^\top)^i\right) \\ &=: N\left((A-BK)^s x,\ D_K^{(s)}\right). \end{aligned}$$

Therefore, by definition, the value function is

$$\begin{aligned} & V_K(x) \\ &= \sum_{s=0}^{\infty}\left\{\mathbb{E}_K\left[c(x_s,u_s)\mid x_0=x\right]-J(K)\right\} \\ &= \sum_{s=0}^{\infty}\left\{\mathbb{E}_K\left[x_s^\top(Q+K^\top RK)x_s\mid x_0=x\right]+\sigma^2\operatorname{Tr}(R)-J(K)\right\} \\ &= \sum_{s=0}^{\infty}\left\{\operatorname{Tr}\left(\mathbb{E}_K\left[x_s x_s^\top\mid x_0=x\right](Q+K^\top RK)\right)-\operatorname{Tr}[D_\epsilon P_K]\right\} \\ &= \sum_{s=0}^{\infty}\left\{\operatorname{Tr}\left[\left((A-BK)^s xx^\top((A-BK)^\top)^s+D_K^{(s)}\right)(Q+K^\top RK)\right]-\operatorname{Tr}[D_\epsilon P_K]\right\}, \end{aligned}$$

where the second equality is by (4.39), the third equality is by (4.9). Therefore,

$$
\begin{aligned}
&V_K(x) \\
&= x^\top P_K x + \sum_{s=0}^{\infty} \Bigg\{ \mathrm{Tr}\left[\left(\sum_{i=0}^{s-1} (A-BK)^i D_\epsilon ((A-BK)^\top)^i \right) (Q + K^\top R K) \right] \\
&\qquad\qquad - \mathrm{Tr}\left[D_\epsilon \left(\sum_{i=0}^{\infty} ((A-BK)^\top)^i (Q + K^\top R K)(A-BK)^i \right) \right] \Bigg\} \\
&= x^\top P_K x - \sum_{s=0}^{\infty} \mathrm{Tr}\left[\left(\sum_{i=s}^{\infty} (A-BK)^i D_\epsilon ((A-BK)^\top)^i \right) (Q + K^\top R K) \right] \\
&= x^\top P_K x - \sum_{s=0}^{\infty} \sum_{j=0}^{\infty} \mathrm{Tr}\left[\left((A-BK)^s D_\epsilon ((A-BK)^\top)^s \right) \right. \\
&\qquad\qquad \left. \left(((A-BK)^\top)^j (Q + K^\top R K)(A-BK)^j \right) \right] \\
&= x^\top P_K x - \sum_{s=0}^{\infty} \left\{ \mathrm{Tr}\left[\left((A-BK)^s D_\epsilon ((A-BK)^\top)^s \right) P_K \right] \right\} = x^\top P_K x - \mathrm{Tr}[D_K P_K],
\end{aligned}
$$

where we have used the series expressions for P_K (4.38) and D_K (4.5). The assumption $\rho(A-BK) < 1$ guarantees that all the series above converge. Next, we compute the state-value function $Q_K(x,u)$. Recall that $Q_K(x,u)$ is the expected extra cost if we start at $x_0 = x$, take a first action $u_0 = u$ and then follow the policy π_K.

Therefore,

$$
\begin{aligned}
&Q_K(x,u) = c(x,u) - J(K) + \mathbb{E}[V_K(x') \mid x,u] \\
&= x^\top Q x + u^\top R u - \operatorname{Tr}[D_\epsilon P_K] - \sigma^2 \operatorname{Tr}(R) + \mathbb{E}_{x'\sim N(Ax+Bu,D_\xi)}[x'^\top P_K x' - \operatorname{Tr}[D_K P_K]] \\
&= x^\top Q x + u^\top R u - \operatorname{Tr}[D_\epsilon P_K] - \sigma^2 \operatorname{Tr}(R) \\
&\quad + \operatorname{Tr}\left[\mathbb{E}_{x'\sim N(Ax+Bu,D_\xi)}[x'x'^\top] P_K\right] - \operatorname{Tr}[D_K P_K] \\
&= x^\top Q x + u^\top R u - \operatorname{Tr}[D_\epsilon P_K + \sigma^2 R + D_K P_K] \\
&\quad + \operatorname{Tr}\left[\left((Ax+Bu)(Ax+Bu)^\top + D_\xi\right) P_K\right] \\
&= x^\top Q x + u^\top R u - \operatorname{Tr}[(D_\epsilon - D_\xi) P_K + \sigma^2 R + D_K P_K] + (Ax+Bu)^\top P_K (Ax+Bu) \\
&= \begin{bmatrix} x^\top & u^\top \end{bmatrix} \begin{bmatrix} Q + A^\top P_K A & A^\top P_K B \\ B^\top P_K A & R + B^\top P_K B \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix} - \sigma^2 \operatorname{Tr}(R + P_K B B^\top) - \operatorname{Tr}(D_K P_K).
\end{aligned}
$$

□

Proof of Proposition 3. The distribution of policy is $\pi_K(u|x) \sim N(-Kx, \sigma^2 I_k)$, with probability density

$$\pi_K(u|x) = (2\pi\sigma^2)^{-k/2} \exp\left(-\frac{1}{2\sigma^2}|u+Kx|^2\right).$$

Therefore,

$$\log \pi_K(u|x) = -\frac{k}{2}\log(2\pi\sigma^2) - \frac{1}{2\sigma^2}|u+Kx|^2$$

and

$$\nabla_K \log \pi_K(u|x) = -\frac{1}{\sigma^2}(u+Kx)x^\top.$$

Therefore, by the definition in (4.23), the Fisher information matrix at state x is

$$
\begin{aligned}
F_x(K) &= \int_{\mathbb{R}^k} (2\pi\sigma^2)^{-k/2} \exp\left(-\frac{1}{2\sigma^2}|u+Kx|^2\right) \frac{1}{\sigma^4}[(u+Kx)x^\top] \otimes [(u+Kx)x^\top] \mathrm{d}u \\
&= \int_{\mathbb{R}^k} (2\pi\sigma^2)^{-k/2} \exp\left(-\frac{1}{2\sigma^2}|u|^2\right) \frac{1}{\sigma^4}[ux^\top] \otimes [ux^\top] \mathrm{d}u.
\end{aligned}
$$

Recall that the stationary state distribution is $N(0, D_K)$. Hence, the Fisher information matrix is

$$F(K) = \int_{\mathbb{R}^d} (2\pi)^{-d/2} (\det(D_K))^{-1/2} \exp\left(-\frac{1}{2} x^\top D_K^{-1} x\right) F_x(K) \mathrm{d}x$$

$$= \int_{\mathbb{R}^d} (2\pi)^{-d/2} (\det(D_K))^{-1/2} \exp\left(-\frac{1}{2} x^\top D_K^{-1} x\right)$$

$$\int_{\mathbb{R}^k} (2\pi\sigma^2)^{-k/2} \exp\left(-\frac{1}{2\sigma^2}|u|^2\right) \frac{1}{\sigma^4} [ux^\top] \otimes [ux^\top] \mathrm{d}u\mathrm{d}x$$

Note that we can compute the integration w.r.t. x and u separately with

$$\int_{\mathbb{R}^d} (2\pi)^{-d/2} (\det(D_K))^{-1/2} \exp\left(-\frac{1}{2} x^\top D_K^{-1} x\right) xx^\top \mathrm{d}x = D_K$$

and

$$\int_{\mathbb{R}^k} (2\pi\sigma^2)^{-k/2} \exp\left(-\frac{1}{2\sigma^2}|u|^2\right) uu^\top \mathrm{d}u = \sigma^2 I_k.$$

Therefore, by an elementwise analysis, we obtain

$$\sigma^2 F(K) \cdot G_K = G_K D_K.$$

Therefore, (4.24) holds. □

4.6.2 Proofs for results in Section 4.3.3

We first prove the lemmas and then the main theorem 1.

Proof of Lemma 1. Firstly

$$D_{K_t} = D_\epsilon + (A - BK_t) D_{K_t} (A - BK_t)^\top \geq D_\epsilon \geq \sigma_{min}(D_\epsilon).$$

D_{K_t} also has an expression in series:

$$D_{K_t} = \sum_{s=0}^{\infty} (A - BK_t)^s D_\epsilon ((A - BK_t)^\top)^s.$$

Since $\lim_{k\to\infty}\|(A-BK_t)^k\|^{\frac{1}{k}} = \rho(A-BK_t) \le \rho < 1$ and $\|A-BK_t\| \le c_A$, (with an argument similar to the proof in Lemma 2 below,) we have

$$D_{K_t} = \sum_{s=0}^{\infty}(A-BK_t)^s D_\epsilon((A-BK_t)^\top)^s \lesssim \frac{1}{1-\rho^2}\|D_\epsilon\|$$

with the constant depending on c_A and d. Therefore, the first inequality in (4.32) holds. The constant c_D is proportional to $\frac{1}{1-\rho^2}\|D_\epsilon\|$ and also depends on c_A and d. The argument above also holds for K^*, so the inequality also holds with K_t replaced by K^*. For P_{K_t}, we also have an expression in series:

$$P_{K_t} = \sum_{s=0}^{\infty}((A-BK_t)^\top)^s(Q+B^\top RB)(A-BK_t)^s.$$

So the argument to prove the second inequality of (4.32) is the same. Finally, since Σ_{K_t} has expression (4.13) with $\|D_{K_t}\| \le c_D$ and $\|K_t\| \le c_K$, $\|\Sigma_{K_t}\|$ has a bound $c_\Sigma = (1+c_K)^2 c_D + \sigma^2$ automatically. □

Proofs for critic

Here we prove the results for the critic.

Proof of Lemma 2. In order to show $\nabla^2 L_K(\theta) = \mathbb{E}_K\left[\psi(x,u)\otimes\psi(x,u)\right] \ge \mu_\sigma$, we only need to show that for any $M \in \mathbb{R}^{(d+k)\times(d+k)}$, we have

$$\mathbb{E}_K\left[(\mathrm{Tr}[M\psi(x,u)])^2\right] \ge \mu_\sigma\|M\|_F^2.$$

Since $\psi(x,u)$ is symmetric, we have $\mathrm{Tr}[M\psi(x,u)] = \mathrm{Tr}[M^\top\psi(x,u)] = \mathrm{Tr}[\frac{1}{2}(M+M^\top)\psi(x,u)]$. We also have $2\|\frac{1}{2}(M+M^\top)\|_F^2 \ge \|M\|_F^2$. Therefore, we only need to show

$$\mathbb{E}_K\left[(\mathrm{Tr}[M\psi(x,u)])^2\right] \ge 2\mu_\sigma\|M\|_F^2 \tag{4.43}$$

for all symmetric matrix M. Recall that

$$z_{s+1} = Ez_s + \widetilde{\epsilon}_s.$$

Since

$$\psi(z) = \mathbb{E}_K[(Ez+\widetilde{\epsilon})(Ez+\widetilde{\epsilon})^\top] - zz^\top = Ezz^\top E^\top + \Sigma_\epsilon - zz^\top,$$

we have

$$\mathrm{Tr}[M\psi(x,u)] = \mathrm{Tr}[MEzz^\top E^\top + M\Sigma_\epsilon - Mzz^\top] = z^\top(E^\top ME - M)z + \mathrm{Tr}[M\Sigma_\epsilon].$$

Recall that $z \sim N(0, \Sigma_K)$ under the stationary distribution where Σ_K is defined in (4.13). By definition, for any $x \in \mathbb{R}^d$, $u \in \mathbb{R}^k$, and $\gamma \neq 0$, we have

$$\begin{aligned}\begin{bmatrix} x^\top & u^\top \end{bmatrix} \Sigma_K \begin{bmatrix} x \\ u \end{bmatrix} &= (\gamma x - \frac{1}{\gamma}K^\top u)^\top D_K(\gamma x - \frac{1}{\gamma}K^\top u) \\ &\quad + (1-\gamma^2)x^\top D_K x + u^\top[\sigma^2 I_k - (\frac{1}{\gamma^2}-1)KD_KK^\top]u. \quad (4.44)\end{aligned}$$

Therefore, we can smartly choose a $\gamma \in (0,1)$ s.t. $(1-\gamma^2)D_K \geq \mu_\Sigma$ and $\sigma^2 I_k - (\frac{1}{\gamma^2}-1)KD_KK^\top \geq \mu_\Sigma$ for some positive constant $\mu_\Sigma \in \mathbb{R}$. Therefore, $\Sigma_K \geq \mu_\Sigma$. Using the same method, we can also show that $\Sigma_\epsilon \geq \mu_\Sigma$. This μ_Σ depends on σ, $\sigma_{min}(D_K)$ ($\sigma_{min}(D_\epsilon)$ for Σ_ϵ) and $\|K\|$. Since $\sigma_{min}(D_K) \geq \sigma_{min}(D_\epsilon) = \mathcal{O}(1)$, μ_Σ is of order $\mathcal{O}(1)$ as long as we have an upper bound for $\|K\|$. We can also find that $\mu_\Sigma = \mathcal{O}(\sigma^2)$ when σ is small. Next, we start to compute (4.43).

$$\begin{aligned}&\mathbb{E}_K\left[(\mathrm{Tr}[M\psi(x,u)])^2\right] \\ &= \mathbb{E}_K\left[\left(z^\top(E^\top ME - M)z + \mathrm{Tr}[M\Sigma_\epsilon]\right)\left(z^\top(E^\top ME - M)z + \mathrm{Tr}[M\Sigma_\epsilon]\right)\right] \\ &= \mathbb{E}_K\left[z^\top(E^\top ME - M)zz^\top(E^\top ME - M)z\right. \\ &\quad \left. + 2z^\top(E^\top ME - M)z\,\mathrm{Tr}[M\Sigma_\epsilon] + \mathrm{Tr}[M\Sigma_\epsilon]^2\right].\end{aligned} \quad (4.45)$$

We will compute each term respectively. We recall the stationary distribution is $z \sim N(0, \Sigma_K)$. If we define $w = \Sigma_K^{-\frac{1}{2}} z$, then $w \sim N(0, I_{d+k})$. Denote $(m_{ij}) = \widetilde{M} = \Sigma_K^{\frac{1}{2}}(E^\top M E - M)\Sigma_K^{\frac{1}{2}}$, then $\widetilde{M}$ is symmetric and

$$\begin{aligned}
&\mathbb{E}_K\left[z^\top (E^\top M E - M) z z^\top (E^\top M E - M) z\right] \\
&= \mathbb{E}_{w\sim N(0,I_{d+k})}\left[w^\top \Sigma_K^{\frac{1}{2}}(E^\top M E - M)\Sigma_K^{\frac{1}{2}} w w^\top \Sigma_K^{\frac{1}{2}}(E^\top M E - M)\Sigma_K^{\frac{1}{2}} w\right] \\
&= \mathbb{E}_{w\sim N(0,I_{d+k})}\left[w^\top \widetilde{M} w w^\top \widetilde{M} w\right] \\
&= \int_{\mathbb{R}^{d+k}} (2\pi)^{-\frac{d+k}{2}} w^\top \widetilde{M} w w^\top \widetilde{M} w \exp\left(-\frac{|w|^2}{2}\right) \mathrm{d}w \qquad (4.46) \\
&= 3\sum_{i=1}^{d+k} m_{ii}^2 + \sum_{i\neq j} m_{ii} m_{jj} + 2\sum_{i\neq j} m_{ij}^2 = 2\,\mathrm{Tr}[\widetilde{M}^2] + \mathrm{Tr}[\widetilde{M}]^2 \\
&= 2\,\mathrm{Tr}\left[\Sigma_K(E^\top M E - M)\Sigma_K(E^\top M E - M)\right] + \mathrm{Tr}\left[\Sigma_K(E^\top M E - M)\right]^2.
\end{aligned}$$

Also,

$$\mathbb{E}_K\left[z^\top (E^\top M E - M) z\right] = \mathbb{E}_K\left[\mathrm{Tr}(z z^\top (E^\top M E - M))\right] = \mathrm{Tr}[\Sigma_K(E^\top M E - M))]. \qquad (4.47)$$

Recall that $\Sigma_K = \Sigma_\epsilon + E\Sigma_K E^\top$, so

$$\mathrm{Tr}[\Sigma_K(E^\top M E - M))] = -\,\mathrm{Tr}[M(\Sigma_K - E\Sigma_K E^\top)] = -\,\mathrm{Tr}[M\Sigma_\epsilon] \qquad (4.48)$$

Therefore, substituting (4.46), (4.47) and (4.48) into (4.45), we obtain

$$\begin{aligned}
&\mathbb{E}_K\left[(\mathrm{Tr}[M\psi(x,u)])^2\right] \\
&= 2\,\mathrm{Tr}\left[\Sigma_K(E^\top M E - M)\Sigma_K(E^\top M E - M)\right] \\
&\quad + \mathrm{Tr}\left[M\Sigma_\epsilon\right]^2 - 2\,\mathrm{Tr}\left[M\Sigma_\epsilon\right]^2 + \mathrm{Tr}\left[M\Sigma_\epsilon\right]^2 \\
&= 2\,\mathrm{Tr}\left[\Sigma_K(E^\top M E - M)\Sigma_K(E^\top M E - M)\right] \qquad (4.49) \\
&\geq 2\mu_\Sigma\,\mathrm{Tr}\left[(E^\top M E - M)\Sigma_K(E^\top M E - M)\right] \\
&\geq 2\mu_\Sigma^2 \left\|E^\top M E - M\right\|_F^2
\end{aligned}$$

for all symmetric matrix M. Next, we want to show $\|M\|_F \lesssim \|E^\top ME - M\|_F$. Since the Frobenius norm is equivalent to the operator norm (with the constant depending on the dimension), we only need to show $\|M\| \lesssim \|E^\top ME - M\|$. Note that this step makes μ_σ depend polynomially on $d + k$. We define an operator $\mathcal{T}_E : \mathbb{R}^{(d+k)\times(d+k)} \to \mathbb{R}^{(d+k)\times(d+k)}$ such that

$$\mathcal{T}_E(X) = \sum_{s=0}^{\infty} (E^\top)^s X E^s.$$

Since $1 > \rho \geq \rho(E) = \lim_{s\to\infty} \|E^s\|^{\frac{1}{s}}$, the norm of the operator should satisfy

$$\|\mathcal{T}_E\| = sup_{X\neq 0} \frac{\|\mathcal{T}_E(X)\|}{\|X\|} \leq \frac{c}{1-\rho^2}$$

where c depends on $\|E\|$ and $d + k$. Notice that

$$\mathcal{T}_E(M - E^\top ME) = \sum_{s=0}^{\infty} (E^\top)^s (M - E^\top ME) E^s = M,$$

we conclude that

$$\|M\| = \|\mathcal{T}_E(M - E^\top ME)\| \leq \|\mathcal{T}_E\| \|M - E^\top ME\| \leq \frac{c}{1-\rho^2} \|M - E^\top ME\|.$$

So, $\|M\|_F \lesssim \|E^\top ME - M\|_F$. Therefore, by (4.49),

$$\nabla^2 L_K(\theta) = \mathbb{E}_K \left[\psi(x, u) \otimes \psi(x, u)\right] \geq \mu_\sigma$$

holds with μ_σ proportional to $\sigma^4/(1-\rho^2)$ and depending on $\|E\|$ and $d+k$. Moreover, μ_σ grows polynomially as $d + k$ becomes large. □

Proof of Lemma 3. Similar to (4.19), we define

$$\nabla L_t(\theta_t) = \mathbb{E}_{K_t}[f(x, u)],$$

where f depends on both θ_t and K_t. We denote $\mathbb{E}_{N_0}[f(x,u)]$ the expectation of the same function under the distribution of (x_{N_0}, u_{N_0}), which starts at $x_0 = 0$ and follows the policy π_{K_t}. We prove (4.34) first. We recall that the feature matrix $\phi(x,u)$ defined in (4.14) is quadratic in (x,u). So, $\psi(x,u) = \mathbb{E}\left[\phi(x',u')|x,u\right] - \phi(x,u)$ also grows at most quadratically in (x,u) since (x',u') are normally distributed. Therefore, $f(x,u)$, defined in (4.19) grows at most quartically in (x,u). By assumption 1, $\|\theta_t\|_F \leq c_\theta = \mathcal{O}(1)$ and $\|K_t\| \leq c_K = \mathcal{O}(1)$, so the coefficients for this quadratic growth are of order $\mathcal{O}(1)$. A similar argument tells us that $\widehat{f}(x_{N_0}^{(i)}, u_{N_0}^{(i)})$ defined in (4.20) grows at most quartically in $\{(x^{(i,j)}, u^{(i,j)})\}_{j=1}^{N_1}$ and $(x_{N_0}^{(i)}, u_{N_0}^{(i)})$, with $\mathcal{O}(1)$ coefficients. Note that $\{(x^{(i,j)}, u^{(i,j)})\}_{j=1}^{N_1}$ and $(x_{N_0}^{(i)}, u_{N_0}^{(i)})$ are normally distributed with 0 mean and $\mathcal{O}(1)$ covariance matrix. Therefore,

$$\mathbb{E}\left[\left\|\widehat{\nabla L_t}(\theta_t)\right\|_F^2 \;\middle|\; \mathcal{G}_t\right] = \mathbb{E}\left[\left\|\frac{1}{N}\sum_{i=1}^{N}\widehat{f}(x_{N_0}^{(i)}, u_{N_0}^{(i)})\right\|_F^2 \;\middle|\; \mathcal{G}_t\right] = \mathcal{O}(1)$$

So (4.34) holds with $c_L = \mathcal{O}(1)$. We also see that $c_L = \text{poly}(d+k)$ as the dimensions increase. We will show (4.33) next. By definition,

$$\left\|\mathbb{E}\left[\widehat{\nabla L_t}(\theta_t) - \nabla L_t(\theta_t) \;\middle|\; \mathcal{G}_t\right]\right\|_F = \|\mathbb{E}_{N_0}[f(x,u)] - \mathbb{E}_{K_t}[f(x,u)]\|_F. \tag{4.50}$$

Here, we remind the reader that the expectation on the left in (4.50) is taken w.r.t. the training filtration $\mathcal{G}_t$ while those on the right are taken w.r.t. the state and action distributions.

We remark that existing results [AA68] bound (4.50) directly. However, it can be computed directly, so we give an elementary proof. Recall that the state trajectory is given by

$$x_{s+1} = (A - BK_t)x_s + \epsilon_s$$

with $x_0 = 0$ where $\epsilon_s \sim N(0, D_\epsilon)$. Therefore, the distribution of x_{N_0} is

$$x_{N_0} \sim N\left(0, \sum_{s=0}^{N_0-1} (A - BK_t)^s D_\epsilon ((A - BK_t)^\top)^s\right) =: N\left(0, D_{K_t}^{(N_0)}\right)$$

and the stationary distribution of x_s is

$$x_\infty \sim N\left(0, \sum_{s=0}^{\infty} (A - BK_t)^s D_\epsilon ((A - BK_t)^\top)^s\right) = N\left(0, D_{K_t}\right).$$

Since $\rho(A - BK_t) \leq \rho < 1$, $D_\epsilon > 0$, and $N_0 = \mathcal{O}(\log \frac{1}{\delta})$, we have $D_{K_t} > \sigma_{min}(D_\epsilon)$, $D_{K_t}^{(N_0)} > \sigma_{min}(D_\epsilon)$, $D_{K_t} - D_{K_t}^{(N_0)} \geq 0$ and $\|D_{K_t} - D_{K_t}^{(N_0)}\|_F \lesssim \delta$. Since

$$u_s \sim N(-K_t x_s, \sigma^2 I_k),$$

we have the joint distribution for $z_{N_0} = (x_{N_0}^\top, u_{N_0}^\top)^\top$

$$z_{N_0} \sim N\left(0, \begin{bmatrix} D_{K_t}^{(N_0)} & -D_{K_t}^{(N_0)} K_t^\top \\ -K_t D_{K_t}^{(N_0)} & K_t D_{K_t}^{(N_0)} K_t^\top + \sigma^2 I_k \end{bmatrix}\right) =: N\left(0, \Sigma_{K_t}^{(N_0)}\right)$$

and the joint stationary distribution

$$z \sim N\left(0, \begin{bmatrix} D_{K_t} & -D_{K_t} K_t^\top \\ -K_t D_{K_t} & K_t D_{K_t} K_t^\top + \sigma^2 I_k \end{bmatrix}\right) =: N\left(0, \Sigma_{K_t}\right)$$

Since $\|D_{K_t} - D_{K_t}^{(N_0)}\|_F \lesssim \delta$ and $\|K_t\| \leq c_K$, we have $\|\Sigma_{K_t} - \Sigma_{K_t}^{(N_0)}\|_F \leq c_6 \delta$. Here the positive constant $c_6 = \mathcal{O}(1)$ decrease geometrically as N_0 increases algebraically. Furthermore, using the same argument when we prove $\Sigma_K \geq \mu_\Sigma$ in Lemma 2, we can

find a positive constant $\mu_\Sigma = \mathcal{O}(1)$ such that $\Sigma_{K_t} \geq \mu_\Sigma$ and $\Sigma_{K_t}^{(N_0)} \geq \mu_\Sigma$. Therefore

$$
\begin{aligned}
&\|\mathbb{E}_{N_0}[f(x,u)] - \mathbb{E}_{K_t}[f(x,u)]\|_F \\
&= \left\| \int_{\mathbb{R}^{d+k}} f(z)(2\pi)^{-\frac{d+k}{2}} \left[\det(\Sigma_{K_t}^{(N_0)})^{-\frac{1}{2}} \exp\left(-\frac{1}{2} z^\top (\Sigma_{K_t}^{(N_0)})^{-1} z\right) \right.\right. \\
&\qquad \left.\left. - \det(\Sigma_{K_t})^{-\frac{1}{2}} \exp\left(-\frac{1}{2} z^\top (\Sigma_{K_t})^{-1} z\right) \right] \mathrm{d}z \right\|_F \\
&\leq \int_{\mathbb{R}^{d+k}} c(1+|z|^4) \left| \det(\Sigma_{K_t}^{(N_0)})^{-\frac{1}{2}} \exp\left(-\frac{1}{2} z^\top (\Sigma_{K_t}^{(N_0)})^{-1} z\right) \right. \\
&\qquad \left. - \det(\Sigma_{K_t})^{-\frac{1}{2}} \exp\left(-\frac{1}{2} z^\top (\Sigma_{K_t})^{-1} z\right) \right| \mathrm{d}z \qquad (4.51) \\
&\leq \int_{\mathbb{R}^{d+k}} c(1+|z|^4) \left[\det(\Sigma_{K_t}^{(N_0)})^{-\frac{1}{2}} - \det(\Sigma_{K_t})^{-\frac{1}{2}} \right] \exp\left(-\frac{1}{2} z^\top (\Sigma_{K_t}^{(N_0)})^{-1} z\right) \mathrm{d}z \\
&\quad + \int_{\mathbb{R}^{d+k}} c(1+|z|^4) \det(\Sigma_{K_t})^{-\frac{1}{2}} \\
&\quad \left[\exp\left(-\frac{1}{2} z^\top (\Sigma_{K_t})^{-1} z\right) - \exp\left(-\frac{1}{2} z^\top (\Sigma_{K_t}^{(N_0)})^{-1} z\right) \right] \mathrm{d}z
\end{aligned}
$$

There is no absolute value at the end of (4.51) because each term is non-negative. Next, we will bound the two integrals respectively. For the first one, we have

$$
\begin{aligned}
&\det(\Sigma_{K_t}^{(N_0)})^{-\frac{1}{2}} - \det(\Sigma_{K_t})^{-\frac{1}{2}} \\
&= \frac{\det(\Sigma_{K_t}) - \det(\Sigma_{K_t}^{(N_0)})}{\sqrt{\det(\Sigma_{K_t}^{(N_0)}) \det(\Sigma_{K_t})} \left(\sqrt{\det(\Sigma_{K_t})} + \sqrt{\det(\Sigma_{K_t}^{(N_0)})} \right)} \\
&= \mathcal{O}(1) \left(\det(\Sigma_{K_t}) - \det(\Sigma_{K_t}^{(N_0)}) \right).
\end{aligned}
$$

Next, we will show $\det(\Sigma_{K_t}) - \det(\Sigma_{K_t}^{(N_0)}) = \mathcal{O}(\delta)$. We can find a unitary matrix U such that $U^\top \Sigma_{K_t}^{(N_0)} U$ is a diagonal matrix,

$$
\left\| U^\top \Sigma_{K_t} U - U^\top \Sigma_{K_t}^{(N_0)} U \right\|_F = \left\| \Sigma_{K_t} - \Sigma_{K_t}^{(N_0)} \right\|_F \leq c_6 \delta,
$$

and

$$\det(\Sigma_{K_t}) - \det(\Sigma_{K_t}^{(N_0)}) = \det(U\Sigma_{K_t}U^\top) - \det(U\Sigma_{K_t}^{(N_0)}U^\top).$$

If we assume that the diagonal element of $U\Sigma_{K_t}U^\top$ to be $a_1, \cdots, a_{d+k}$ and

$$U\Sigma_{K_t}^{(N_0)}U^\top = \mathrm{diag}(b_1, \cdots, b_{d+k}).$$

Then $a_i \geq b_i$ and $a_i - b_i = \mathcal{O}(\delta)$. Therefore

$$0 \leq \det(U\Sigma_{K_t}U^\top) - \det(U\Sigma_{K_t}^{(N_0)}U^\top) \leq \prod_{i=1}^{d+k} a_i - \prod_{i=1}^{d+k} b_i = \mathcal{O}(\delta).$$

Therefore, $\det(\Sigma_{K_t}) - \det(\Sigma_{K_t}^{(N_0)}) = \mathcal{O}(\delta)$ and hence

$$\det(\Sigma_{K_t}^{(N_0)})^{-\frac{1}{2}} - \det(\Sigma_{K_t})^{-\frac{1}{2}} \leq c\delta$$

with positive constant c being as small as we want (through increasing N_0). Therefore, the first integral in (4.51) satisfies

$$\begin{aligned} &\int_{\mathbb{R}^{d+k}} c(1+|z|^4)\left[\det(\Sigma_{K_t}^{(N_0)})^{-\frac{1}{2}} - \det(\Sigma_{K_t})^{-\frac{1}{2}}\right] \exp\left(-\frac{1}{2}z^\top(\Sigma_{K_t}^{(N_0)})^{-1}z\right) \mathrm{d}z \\ &\leq c\delta \int_{\mathbb{R}^{d+k}} (1+|z|^4) \exp\left(-\frac{1}{2}z^\top(\Sigma_{K_t}^{(N_0)})^{-1}z\right) \mathrm{d}z = c\delta\mathcal{O}(1) \leq \frac{1}{2}\delta. \end{aligned} \tag{4.52}$$

Here, again, the constant c may differ according to the context. A more detailed computation shows that

$$\det(\Sigma_{K_t}^{(N_0)})^{-\frac{1}{2}} - \det(\Sigma_{K_t})^{-\frac{1}{2}} \leq \det(\Sigma_{K_t}^{(N_0)})^{-\frac{1}{2}} \ \mathrm{poly}(d+k)\ c_6\delta.$$

Therefore, N_0 should scale with $\log(d+k)$ as the dimensions increase. Next, we

bound the second integration in (4.51). Using the inequality $1 - e^{-x} \leq x$, we have

$$
\begin{aligned}
&\exp\left(-\frac{1}{2}z^\top(\Sigma_{K_t})^{-1}z\right) - \exp\left(-\frac{1}{2}z^\top(\Sigma_{K_t}^{(N_0)})^{-1}z\right) \\
&= \exp\left(-\frac{1}{2}z^\top(\Sigma_{K_t})^{-1}z\right)\left[1 - \exp\left(-\frac{1}{2}z^\top\left((\Sigma_{K_t}^{(N_0)})^{-1} - (\Sigma_{K_t})^{-1}\right)z\right)\right] \\
&\leq \frac{1}{2}z^\top\left((\Sigma_{K_t}^{(N_0)})^{-1} - (\Sigma_{K_t})^{-1}\right)z\exp\left(-\frac{1}{2}z^\top(\Sigma_{K_t})^{-1}z\right) \\
&= \frac{1}{2}\exp\left(-\frac{1}{2}z^\top(\Sigma_{K_t})^{-1}z\right)\mathrm{Tr}\left[\left((\Sigma_{K_t}^{(N_0)})^{-1} - (\Sigma_{K_t})^{-1}\right)zz^\top\right] \\
&= \frac{1}{2}\exp\left(-\frac{1}{2}z^\top(\Sigma_{K_t})^{-1}z\right)\mathrm{Tr}\left[(\Sigma_{K_t}^{(N_0)})^{-1}\left(\Sigma_{K_t} - \Sigma_{K_t}^{(N_0)}\right)(\Sigma_{K_t})^{-1}zz^\top\right] \\
&\leq \frac{1}{2}\exp\left(-\frac{1}{2}z^\top(\Sigma_{K_t})^{-1}z\right)\|(\Sigma_{K_t}^{(N_0)})^{-1}\left(\Sigma_{K_t} - \Sigma_{K_t}^{(N_0)}\right)(\Sigma_{K_t})^{-1}\|\ \mathrm{Tr}[zz^\top] \\
&\leq \frac{1}{2}\exp\left(-\frac{1}{2}z^\top(\Sigma_{K_t})^{-1}z\right)\frac{1}{\mu_\Sigma^2}c_6\delta|z|^2.
\end{aligned}
$$

Therefore, the second integration in (4.51) satisfies

$$
\begin{aligned}
&\int_{\mathbb{R}^{d+k}} c(1+|z|^4)\det(\Sigma_{K_t})^{-\frac{1}{2}} \\
&\left[\exp\left(-\frac{1}{2}z^\top(\Sigma_{K_t})^{-1}z\right) - \exp\left(-\frac{1}{2}z^\top(\Sigma_{K_t}^{(N_0)})^{-1}z\right)\right]\mathrm{d}z \\
&\leq \delta\int_{\mathbb{R}^{d+k}} c(|z|^2+|z|^6)\det(\Sigma_{K_t})^{-\frac{1}{2}}\exp\left(-\frac{1}{2}z^\top(\Sigma_{K_t})^{-1}z\right)\mathrm{d}z = \delta c\mathcal{O}(1) \leq \frac{1}{2}\delta.
\end{aligned}
\tag{4.53}
$$

Plugging (4.52) and (4.53) into (4.51), we obtain

$$
\|\mathbb{E}_{N_0}[f(x,u)] - \mathbb{E}_{K_t}[f(x,u)]\|_F \leq \delta.
$$

□

Proof of Lemma 4. By definition

$$\theta_K - \theta_{K'} = \begin{bmatrix} A^\top(P_K - P_{K'})A & A^\top(P_K - P_{K'})B \\ B^\top(P_K - P_{K'})A & B^\top(P_K - P_{K'})B \end{bmatrix} = \begin{bmatrix} A^\top \\ B^\top \end{bmatrix} \left[P_K - P_{K'}\right] \begin{bmatrix} A & B \end{bmatrix}$$

Therefore,

$$\begin{aligned} &\|\theta_K - \theta_{K'}\|_F^2 = \mathrm{Tr}[(\theta_K - \theta_{K'})^\top(\theta_K - \theta_{K'})] \\ &= \mathrm{Tr}\left([(AA^\top + BB^\top)(P_K - P_{K'})]^2\right) \leq (\|A\|^2 + \|B\|^2)^2 \|P_K - P_{K'}\|_F^2 \end{aligned} \tag{4.54}$$

Therefore, our goal is to bound $\|P_K - P_{K'}\|_F$ by $\|K - K'\|_F$. By definition in (4.8),

$$\begin{aligned} &P_K - P_{K'} \\ &= K^\top RK - K'^\top RK' + (A - BK)^\top P_K(A - BK) - (A - BK')^\top P_{K'}(A - BK') \\ &= K^\top RK - K^\top RK' + K^\top RK' - K'^\top RK' \\ &\quad + (A - BK)^\top P_K(A - BK) - (A - BK)^\top P_K(A - BK') \\ &\quad + (A - BK)^\top P_K(A - BK') - (A - BK)^\top P_{K'}(A - BK') \\ &\quad + (A - BK)^\top P_{K'}(A - BK') - (A - BK')^\top P_{K'}(A - BK') \\ &= K^\top R(K - K') + (K - K')^\top RK' - (A - BK)^\top P_K B(K - K') \\ &\quad + (A - BK)^\top (P_K - P_{K'})(A - BK') - (K - K')^\top B^\top P_{K'}(A - BK') \end{aligned}$$

Therefore,

$$\begin{aligned} &P_K - P_{K'} - (A - BK)^\top (P_K - P_{K'})(A - BK') \\ &= K^\top R(K - K') + (K - K')^\top RK' \\ &\quad - (A - BK)^\top P_K B(K - K') - (K - K')^\top B^\top P_{K'}(A - BK') \end{aligned} \tag{4.55}$$

Next, we want to take $\|\cdot\|_F$ on both sides of (4.55). For the left hand side, since $\rho(A - BK), \rho(A - BK') \leq \rho < 1$ and $\|A - BK\|, \|A - BK'\| \leq c_A$, we can repeat the last part in the proof of Lemma 2 and prove that

$$\|P_K - P_{K'}\|_F \leq c\|(P_K - P_{K'}) - (A - BK)^\top (P_K - P_{K'})(A - BK')\|_F \tag{4.56}$$

where c is proportional to $1/(1-\rho^2)$ and also depends on c_A and d. For the right hand side of (4.55), since $\|P_K\| \le c_P$, $\|P_{K'}\| \le c_P$, $\|K\| \le c_K$ and $\|K'\| \le c_K$,

$$\begin{aligned}
&\|K^\top R(K-K') + (K-K')^\top RK' \\
&- (A-BK)^\top P_K B(K-K') - (K-K')^\top B^\top P_{K'}(A-BK')\|_F \qquad (4.57)\\
&\le 2(c_K\|R\| + c_P c_A\|B\|)\,\|K-K'\|_F.
\end{aligned}$$

Plugging (4.56) and (4.57) into (4.55), we obtain

$$\|P_K - P_{K'}\|_F \le 2c(c_K\|R\| + c_P c_A\|B\|)\,\|K-K'\|_F. \qquad (4.58)$$

Finally, combining (4.54) and (4.58), we obtain

$$\|\theta_K - \theta_{K'}\|_F \le c_1\|K-K'\|_F \qquad (4.59)$$

with $c_1 = 2c(c_K\|R\| + c_P c_A\|B\|)\,(\|A\|^2 + \|B\|^2)$. This c_1 grows polynomially as the dimensions increase. □

Proof of Lemma 5. Note that

$$\begin{aligned}
&\|\theta_{t+1} - \theta_{K_{t+1}}\|_F^2 = \|\theta_t - \alpha_t\widehat{\nabla L_t}(\theta_t) - \theta_{K_t} + \theta_{K_t} - \theta_{K_{t+1}}\|_F^2\\
&= \|\theta_t - \theta_{K_t}\|_F^2 - 2\alpha_t \operatorname{Tr}\left[(\theta_t - \theta_{K_t})^\top \widehat{\nabla L_t}(\theta_t)\right]\\
&\quad + \alpha_t^2\|\widehat{\nabla L_t}(\theta_t)\|_F^2 + \|\theta_{K_t} - \theta_{K_{t+1}}\|_F^2 + 2\operatorname{Tr}\left[(\theta_{K_t} - \theta_{K_{t+1}})^\top(\theta_t - \theta_{K_t} - \alpha_t\widehat{\nabla L_t}(\theta_t))\right]\\
&= \|\theta_t - \theta_{K_t}\|_F^2 - 2\alpha_t \operatorname{Tr}\left[(\theta_t - \theta_{K_t})^\top \nabla L_t(\theta_t)\right]\\
&\quad + 2\alpha_t \operatorname{Tr}\left[(\theta_t - \theta_{K_t})^\top(\nabla L_t(\theta_t) - \widehat{\nabla L_t}(\theta_t))\right] + \alpha_t^2\|\widehat{\nabla L_t}(\theta_t)\|_F^2\\
&\quad + \|\theta_{K_t} - \theta_{K_{t+1}}\|_F^2 + 2\operatorname{Tr}\left[(\theta_{K_t} - \theta_{K_{t+1}})^\top(\theta_t - \theta_{K_t} - \alpha_t\widehat{\nabla L_t}(\theta_t))\right].
\end{aligned}$$

So

$$
\begin{aligned}
&\|\theta_{t+1} - \theta_{K_{t+1}}\|_F^2 \\
&\leq (1 - 2\alpha_t\mu_\sigma)\|\theta_t - \theta_{K_t}\|_F^2 + 2\alpha_t \operatorname{Tr}\left[(\theta_t - \theta_{K_t})^\top(\nabla L_t(\theta_t) - \widehat{\nabla L_t}(\theta_t))\right] \\
&\quad + \alpha_t^2\|\widehat{\nabla L_t}(\theta_t)\|_F^2 + \|\theta_{K_t} - \theta_{K_{t+1}}\|_F^2 \\
&\quad + 2\operatorname{Tr}\left[(\theta_{K_t} - \theta_{K_{t+1}})^\top(\theta_t - \theta_{K_t})\right] - 2\alpha_t \operatorname{Tr}\left[(\theta_{K_t} - \theta_{K_{t+1}})^\top\widehat{\nabla L_t}(\theta_t)\right] \qquad (4.60) \\
&\leq (1 - \frac{5}{3}\alpha_t\mu_\sigma)\|\theta_t - \theta_{K_t}\|_F^2 + 2\alpha_t \operatorname{Tr}\left[(\theta_t - \theta_{K_t})^\top(\nabla L_t(\theta_t) - \widehat{\nabla L_t}(\theta_t))\right] \\
&\quad + 2\alpha_t^2\|\widehat{\nabla L_t}(\theta_t)\|_F^2 + (\frac{3}{\alpha_t\mu_\sigma} + 2)\|\theta_{K_t} - \theta_{K_{t+1}}\|_F^2
\end{aligned}
$$

The first inequality is because $L_t(\theta)$ is $\mu_\sigma-$ strongly convex and hence

$$
\operatorname{Tr}\left[(\theta_t - \theta_{K_t})^\top\nabla L_t(\theta_t)\right] = \operatorname{Tr}\left[(\theta_t - \theta_{K_t})^\top(\nabla L_t(\theta_t) - \nabla L_t(\theta_{K_t}))\right] \geq \mu_\sigma\|\theta_t - \theta_{K_t}\|_F^2.
$$

The second inequality in (4.60) is a simple application of Cauchy-Schwartz inequality. Taking expectation w.r.t. $\mathcal{G}_t$ in (4.60), we obtain

$$
\begin{aligned}
&\mathbb{E}\left[\|\theta_{t+1} - \theta_{K_{t+1}}\|_F^2 \;\middle|\; \mathcal{G}_t\right] \\
&\leq (1 - \frac{5}{3}\alpha_t\mu_\sigma)\|\theta_t - \theta_{K_t}\|_F^2 + 2\alpha_t \operatorname{Tr}\left[(\theta_t - \theta_{K_t})^\top\mathbb{E}\left[\nabla L_t(\theta_t) - \widehat{\nabla L_t}(\theta_t) \;\middle|\; \mathcal{G}_t\right]\right] \\
&\quad + 2\alpha_t^2\mathbb{E}\left[\left\|\widehat{\nabla L_t}(\theta_t)\right\|_F^2 \;\middle|\; \mathcal{G}_t\right] + (\frac{3}{\alpha_t\mu_\sigma} + 2)\|\theta_{K_t} - \theta_{K_{t+1}}\|_F^2 \\
&\leq (1 - \frac{4}{3}\alpha_t\mu_\sigma)\|\theta_t - \theta_{K_t}\|_F^2 + \frac{3\alpha_t}{\mu_\sigma}\left\|\mathbb{E}\left[\nabla L_t(\theta_t) - \widehat{\nabla L_t}(\theta_t) \;\middle|\; \mathcal{G}_t\right]\right\|_F^2 \\
&\quad + 2\alpha_t^2\mathbb{E}\left[\left\|\widehat{\nabla L_t}(\theta_t)\right\|_F^2 \;\middle|\; \mathcal{G}_t\right] + (\frac{3}{\alpha_t\mu_\sigma} + 2)\|\theta_{K_t} - \theta_{K_{t+1}}\|_F^2.
\end{aligned}
$$

Therefore,

$$\begin{aligned}
&\mathbb{E}\left[\|\theta_{t+1}-\theta_{K_{t+1}}\|_F^2 \mid \mathcal{G}_t\right]-\|\theta_t-\theta_{K_t}\|_F^2 \\
\leq& -\frac{4}{3}\alpha_t\mu_\sigma\|\theta_t-\theta_{K_t}\|_F^2+\frac{3\alpha_t}{\mu_\sigma}\left\|\mathbb{E}\left[\nabla L_t(\theta_t)-\widehat{\nabla L_t}(\theta_t)\mid \mathcal{G}_t\right]\right\|_F^2 \\
&+2\alpha_t^2\mathbb{E}\left[\left\|\widehat{\nabla L_t}(\theta_t)\right\|_F^2 \mid \mathcal{G}_t\right]+(\frac{3}{\alpha_t\mu_\sigma}+2)\|\theta_{K_t}-\theta_{K_{t+1}}\|_F^2.
\end{aligned}$$

Combining with (4.33), (4.34), and the definition of α_t, we obtain (4.36):

$$\begin{aligned}
&\mathbb{E}\left[\|\theta_{t+1}-\theta_{K_{t+1}}\|_F^2 \mid \mathcal{G}_t\right]-\|\theta_t-\theta_{K_t}\|_F^2 \\
\leq& -\frac{4}{3}\alpha_t\mu_\sigma\|\theta_t-\theta_{K_t}\|_F^2+\frac{1}{4}\frac{\sigma_{min}(D_\epsilon)}{c_3}\beta_t\varepsilon+\big(\frac{3}{\alpha_t\mu_\sigma}+2\big)\|\theta_{K_t}-\theta_{K_{t+1}}\|_F^2.
\end{aligned}$$

□

Proofs for the Actor

Next, we prove the results for the actor.

Proof of Lemma 6. We prove the upper bound first. According to (4.9),

$$J(K)-J(K^*)=\mathrm{Tr}((P_K-P_{K^*})D_\epsilon)=\mathbb{E}_{x\sim N(0,D_\epsilon)}[x^\top(P_K-P_{K^*})x] \tag{4.61}$$

where we recall that $P_K=(Q+K^\top RK)+(A-BK)^\top P_K(A-BK)$ and P_{K^*} satisfies a similar equation. So, P_{K^*} also has the following expression in series

$$P_{K^*}=\sum_{s=0}^{\infty}[(A-BK^*)^s]^\top(Q+K^{*\top}RK^*)(A-BK^*)^s.$$

Therefore, if we define a sequence $\{y_s\}_{s=0}^\infty$ with $y_0=x$ and $y_{s+1}=(A-BK^*)y_s$, then

$$\begin{aligned}
x^\top P_{K^*}x&=\sum_{s=0}^{\infty}x^\top[(A-BK^*)^s]^\top(Q+K^{*\top}R^*K)(A-BK^*)^sx \\
&=\sum_{s=0}^{\infty}y_s^\top(Q+K^{*\top}RK^*)y_s.
\end{aligned}$$

Combining with

$$x^\top P_K x = \sum_{s=0}^{\infty} \left(y_s^\top P_K y_s - y_{s+1}^\top P_K y_{s+1} \right) = \sum_{s=0}^{\infty} y_s^\top (P_K - (A - BK^*)^\top P_K (A - BK^*)) y_s$$

and (4.61), we obtain

$$\begin{aligned}
&J(K) - J(K^*) \\
&= \mathbb{E}_{D_\epsilon, K^*} \left[\sum_{s=0}^{\infty} y_s^\top \left(-Q - K^{*\top} R K^* + P_K - (A - BK^*)^\top P_K (A - BK^*) \right) y_s \right] \\
&= \mathrm{Tr} \left[\mathbb{E}_{D_\epsilon, K^*} \left[\sum_{s=0}^{\infty} y_s y_s^\top \right] \cdot \left(-Q - K^{*\top} R K^* + P_K - (A - BK^*)^\top P_K (A - BK^*) \right) \right]
\end{aligned} \tag{4.62}$$

where $\mathbb{E}_{D_\epsilon, K^*}$ denotes the expectation with $y_0 \sim N(0, D_\epsilon)$ and $y_{s+1} = (A - BK^*) y_s$. Next, we analyze the two terms in (4.62) respectively. The first term is easy, recall that D_{K^*} is the solution of

$$D_{K^*} = D_\epsilon + (A - BK^*) D_{K^*} (A - BK^*)^\top$$

so that

$$D_{K^*} = \sum_{s=0}^{\infty} (A - BK^*)^s D_\epsilon [(A - BK^*)^\top]^s.$$

Therefore,

$$\mathbb{E}_{D_\epsilon, K^*} \left[\sum_{s=0}^{\infty} y_s y_s^\top \right] = \mathbb{E}_{x \sim N(0, D_\epsilon)} \left[\sum_{s=0}^{\infty} (A - BK^*)^s x x^\top [(A - BK^*)^\top]^s \right] = D_{K^*}. \tag{4.63}$$

Next, we consider the second term in (4.62). By direct computation,

$$
\begin{aligned}
&-Q - K^{*\top} R K^* + P_K - (A - BK^*)^\top P_K (A - BK^*) \\
&= -Q - (K^* - K + K)^\top R (K^* - K + K) + P_K \\
&\quad - (A - BK + BK - BK^*)^\top P_K (A - BK + BK - BK^*) \\
&= (K - K^*)^\top (RK - B^\top P_K (A - BK)) + (RK - B^\top P_K (A - BK))^\top (K - K^*) \\
&\quad - (K - K^*)^\top (R + B^\top P_K B)(K - K^*) \\
&= (K - K^*)^\top G_K + G_K^\top (K - K^*) - (K - K^*)^\top (R + B^\top P_K B)(K - K^*) \\
&= G_K^\top (R + B^\top P_K B)^{-1} G_K - (K - K^* - (R + B^\top P_K B)^{-1} G_K)^\top \\
&\quad (R + B^\top P_K B)(K - K^* - (R + B^\top P_K B)^{-1} G_K) \\
&\leq G_K^\top (R + B^\top P_K B)^{-1} G_K
\end{aligned}
\tag{4.64}
$$

where we have used the equation (4.8) for P_K in the second equality and the definition of G_K (4.22) in the third equality. The $\leq$ above means the difference of the two matrix is positive semi-definite. Plugging (4.63) and (4.64) into (4.62), we obtain

$$
J(K) - J(K^*) \leq \operatorname{Tr}(D_{K^*} G_K^\top (R + B^\top P_K B)^{-1} G_K) \leq \|D_{K^*}\| / \sigma_{min}(R) \operatorname{Tr}(G_K G_K^\top).
$$

This finishes the proof of the upper bound. Next, we prove the lower bound. Note that the argument above does not rely on the optimality of K^*. Therefore, we can obtain a general formula (that is useful in the proof later):

$$
\begin{aligned}
J(K) - J(K') = \operatorname{Tr}\big[D_{K'} \big(&(K - K')^\top G_K && (4.65) \\
&+ G_K^\top (K - K') - (K - K')^\top (R + B^\top P_K B)(K - K') \big) \big]. && (4.66)
\end{aligned}
$$

Specifically, we can set $K' = K - (R + B^\top P_K B)^{-1} G_K$ (i.e., let (4.64) hold with

equality), then by the optimality of K^* and (4.65), we obtain

$$J(K) - J(K^*) \geq J(K) - J(K') = \mathrm{Tr}(D_{K'} G_K^\top (R + B^\top P_K B)^{-1} G_K)$$

$$\geq \sigma_{min}(D_\epsilon) \, \|R + B^\top P_K B\|^{-1} \, \mathrm{Tr}(G_K G_K^\top) \geq \frac{\sigma_{min}(D_\epsilon)}{\|R\| + c_P \|B\|^2} \, \mathrm{Tr}(G_K G_K^\top)$$

□

Proof of Lemma 7. By (4.65),

$$\begin{aligned}
&J(K_t) - J(K_{t+1}) \\
&= \mathrm{Tr}\left[D_{K_{t+1}} \left((K_t - K_{t+1})^\top G_{K_t} + G_{K_t}^\top (K_t - K_{t+1}) \right.\right. \\
&\quad \left.\left. -(K_t - K_{t+1})^\top (R + B^\top P_{K_t} B)(K_t - K_{t+1})\right)\right] \\
&= \mathrm{Tr}\left[D_{K_{t+1}} \left(\beta_t \widehat{G}_{K_t}^\top G_{K_t} + \beta_t G_{K_t}^\top \widehat{G}_{K_t} - \beta_t^2 \widehat{G}_{K_t}^\top (R + B^\top P_{K_t} B) \widehat{G}_{K_t}\right)\right]
\end{aligned}$$

Therefore,

$$\begin{aligned}
&J(K_{t+1}) - J(K_t) \\
&= -\beta_t \, \mathrm{Tr}\left[D_{K_t} \left(\widehat{G}_{K_t}^\top G_{K_t} + G_{K_t}^\top \widehat{G}_{K_t} - \beta_t \widehat{G}_{K_t}^\top (R + B^\top P_{K_t} B) \widehat{G}_{K_t}\right)\right] \\
&= -\beta_t \, \mathrm{Tr}\left[D_{K_t} \left(G_{K_t}^\top G_{K_t} + \widehat{G}_{K_t}^\top \widehat{G}_{K_t} - (G_{K_t} - \widehat{G}_{K_t})^\top (G_{K_t} - \widehat{G}_{K_t}) \right.\right. \\
&\quad \left.\left. -\beta_t \widehat{G}_{K_t}^\top (R + B^\top P_{K_t} B) \widehat{G}_{K_t}\right)\right]
\end{aligned}$$

Recall that we proved

$$\sigma_{min}(D_\epsilon) I_d \leq D_{K_t} \leq c_D I_d \quad \text{and} \quad P_{K_t} \leq c_P$$

in Lemma 1. Therefore,

$$\mathrm{Tr}\left[D_{K_t} G_{K_t}^\top G_{K_t}\right] \geq \sigma_{min}(D_\epsilon) \|G_{K_t}\|_F^2,$$

$$\mathrm{Tr}\left[D_{K_t} \widehat{G}_{K_t}^\top \widehat{G}_{K_t}\right] \geq \sigma_{min}(D_\epsilon) \|\widehat{G}_{K_t}\|_F^2,$$

$$\mathrm{Tr}\left[D_{K_t}\widehat{G}_{K_t}^\top(R+B^\top P_{K_t}B)\widehat{G}_{K_t}\right] \le c_D(\|R\|+c_P\|B\|^2)\|\widehat{G}_{K_t}\|_F^2,$$

and

$$\mathrm{Tr}\left[D_{K_t}(G_{K_t}-\widehat{G}_{K_t})^\top(G_{K_t}-\widehat{G}_{K_t})\right] \le c_D\|G_{K_t}-\widehat{G}_{K_t}\|_F^2.$$

Therefore,

$$\begin{aligned}J(K_{t+1})-J(K_t) \le &-\beta_t\sigma_{min}(D_\epsilon)(\|G_{K_t}\|_F^2+\|\widehat{G}_{K_t}\|_F^2)+\beta_t c_D\|G_{K_t}-\widehat{G}_{K_t}\|_F^2\\ &+\beta_t^2 c_D(\|R\|+c_P\|B\|^2)\|\widehat{G}_{K_t}\|_F^2\end{aligned}$$

Finally, by Lemma 6, we can conclude that

$$\begin{aligned}J(K_{t+1})-J(K_t) \le &-\beta_t\frac{\sigma_{min}(D_\epsilon)}{c_3}(J(K_t)-J(K^*))\\ &-\beta_t\left[\sigma_{min}(D_\epsilon)-\beta_t c_D(\|R\|+c_P\|B\|^2)\right]\|\widehat{G}_{K_t}\|_F^2+\beta_t c_D\|G_{K_t}-\widehat{G}_{K_t}\|_F^2\end{aligned}$$

□

Proofs for the main theorem

Finally we can prove our main theorem.

Proof of Theorem 1. By lemma 3, (4.33) and (4.34) hold for all $t \le T$. We define a Lyapunov function

$$\mathcal{L}_t = \mathcal{L}(\theta_t, K_t) = \|\theta_t-\theta_{K_t}\|_F^2 + J(K_t) - J(K^*).$$

Firstly, $\mathcal{L}_0 = \mathcal{O}(1)$ because

$$\|\theta_0-\theta_{K_0}\|_F^2 = \|\theta_{K_0}\|_F^2 = \left\|\begin{bmatrix}Q+A^\top P_{K_0}A & A^\top P_{K_0}B\\ B^\top P_{K_0}A & R+B^\top P_{K_0}B\end{bmatrix}\right\|_F^2 = \mathcal{O}(1)$$

(note that $P_{K_0} = Q + A^\top P_{K_0}A$ implies $\|P_{K_0}\|_F = \mathcal{O}(1)$) and

$$J(K_0)-J(K^*) \le J(K_0) = \mathrm{Tr}(D_\epsilon P_{K_0}) + \sigma^2\,\mathrm{Tr}(R) \le c_P\,\mathrm{Tr}[D_\epsilon] + \sigma^2\,\mathrm{Tr}(R) = \mathcal{O}(1).$$

Next, we want to show a decrease rate of the Lyapunov function. According to Lemma 5 and Lemma 7,

$$\begin{aligned}
&\mathbb{E}\left[\mathcal{L}_{t+1} \mid \mathcal{G}_t\right] - \mathcal{L}_t \\
\leq& -\frac{4}{3}\alpha_t\mu_\sigma\|\theta_t - \theta_{K_t}\|_F^2 + \frac{1}{4}\frac{\sigma_{min}(D_\epsilon)}{c_3}\beta_t\varepsilon + (\frac{3}{\alpha_t\mu_\sigma} + 2)\|\theta_{K_t} - \theta_{K_{t+1}}\|_F^2 \\
&- \beta_t\frac{\sigma_{min}(D_\epsilon)}{c_3}(J(K_t) - J(K^*)) - \beta_t\left[\sigma_{min}(D_\epsilon) - \beta_t c_D(\|R\| + c_P\|B\|^2)\right]\|\widehat{G}_{K_t}\|_F^2 \\
&+ \beta_t c_D\|G_{K_t} - \widehat{G}_{K_t}\|_F^2.
\end{aligned} \tag{4.67}$$

Fortunately, we can use the negative term in the actor estimate to bound the positive term in the critic estimate and use the negative term in the critic estimate to bound the positive term in the actor estimate. Specifically, by Lemma 4,

$$\|\theta_{K_t} - \theta_{K_{t+1}}\|_F^2 \leq c_1^2\|K_t - K_{t+1}\|_F^2 = c_1^2\beta_t^2\|\widehat{G}_{K_t}\|_F^2.$$

So, by the second inequality in (4.28)

$$\beta_t\left[\sigma_{min}(D_\epsilon) - \beta_t c_D(\|R\| + c_P\|B\|^2)\right]\|\widehat{G}_{K_t}\|_F^2 \geq (\frac{3}{\alpha_t\mu_\sigma} + 2)\|\theta_{K_t} - \theta_{K_{t+1}}\|_F^2. \tag{4.68}$$

In addition,

$$\|G_{K_t} - \widehat{G}_{K_t}\|_F^2 = \|(\theta_{K_t}^{22} - \theta_t^{22})K_t - (\theta_{K_t}^{21} - \theta_t^{21})\|_F^2 \leq c_K^2\|\theta_t - \theta_{K_t}\|_F^2.$$

So, by the third inequality in (4.28)

$$\frac{1}{3}\alpha_t\mu_\sigma\|\theta_t - \theta_{K_t}\|_F^2 \geq \beta_t c_D\|G_{K_t} - \widehat{G}_{K_t}\|_F^2. \tag{4.69}$$

Substituting (4.68) and (4.69) into (4.67), we obtain

$$\begin{aligned}
&\mathbb{E}\left[\mathcal{L}_{t+1} \mid \mathcal{G}_t\right] - \mathcal{L}_t \\
\leq& -\alpha_t\mu_\sigma\|\theta_t - \theta_{K_t}\|_F^2 + \frac{1}{4}\frac{\sigma_{min}(D_\epsilon)}{c_3}\beta_t\varepsilon - \beta_t\frac{\sigma_{min}(D_\epsilon)}{c_3}(J(K_t) - J(K^*)).
\end{aligned}$$

Taking expectation, we obtain

$$\begin{aligned}&\mathbb{E}[\mathcal{L}_{t+1}-\mathcal{L}_t]\\&\leq -\mathbb{E}\left[\alpha_t\mu_\sigma\|\theta_t-\theta_{K_t}\|_F^2+\beta_t\frac{\sigma_{min}(D_\epsilon)}{c_3}(J(K_t)-J(K^*))\right]+\frac{1}{4}\frac{\sigma_{min}(D_\epsilon)}{c_3}\beta_t\varepsilon.\end{aligned}\tag{4.70}$$

Next, we consider three cases. The first case is when $\mathbb{E}[\|\theta_t-\theta_{K_t}\|_F^2]\geq\frac{1}{2}\varepsilon$. In this case, by (4.70) and the first inequality of (4.28),

$$\mathbb{E}[\mathcal{L}_{t+1}-\mathcal{L}_t]\leq -\mathbb{E}\left[\frac{1}{3}\alpha_t\mu_\sigma\|\theta_t-\theta_{K_t}\|_F^2+\beta_t\frac{\sigma_{min}(D_\epsilon)}{c_3}(J(K_t)-J(K^*))\right].$$

The second case is when $\mathbb{E}[J(K_t)-J(K^*)]\geq\frac{1}{2}\varepsilon$. In this case

$$\mathbb{E}[\mathcal{L}_{t+1}-\mathcal{L}_t]\leq -\mathbb{E}\left[\alpha_t\mu_\sigma\|\theta_t-\theta_{K_t}\|_F^2+\frac{1}{2}\beta_t\frac{\sigma_{min}(D_\epsilon)}{c_3}(J(K_t)-J(K^*))\right].$$

In both the first and the second cases, we have

$$\mathbb{E}[\mathcal{L}_{t+1}-\mathcal{L}_t]\leq -\mathbb{E}\left[\frac{1}{3}\alpha_t\mu_\sigma\|\theta_t-\theta_{K_t}\|_F^2+\frac{1}{2}\beta_t\frac{\sigma_{min}(D_\epsilon)}{c_3}(J(K_t)-J(K^*))\right].$$

Note that $\frac{1}{2}\beta_t\frac{\sigma_{min}(D_\epsilon)}{c_3}\leq\frac{1}{3}\alpha_t\mu_\sigma$, we obtain a contraction rate for the Lyaponov function in both cases:

$$\mathbb{E}[\mathcal{L}_{t+1}-\mathcal{L}_t]\leq -\frac{1}{2}\beta_t\frac{\sigma_{min}(D_\epsilon)}{c_3}\mathbb{E}[\mathcal{L}_t]=:-\beta_t c_4\mathbb{E}[\mathcal{L}_t]$$

where we remind the reader that $L(\theta_{K^*},K^*)=0$. Let us rewrite it into a contraction form

$$\mathbb{E}[\mathcal{L}_{t+1}]\leq(1-\beta_t c_4)\mathbb{E}[\mathcal{L}_t].\tag{4.71}$$

Next, we consider the third case, when both $\mathbb{E}[\|\theta_t-\theta_{K_t}\|_F^2]<\frac{1}{2}\varepsilon$ and $\mathbb{E}[J(K_t)-$

$J(K^*)] < \frac{1}{2}\varepsilon$. In this case we have $\mathbb{E}[\mathcal{L}_t] < \varepsilon$. Therefore, by (4.70), we obtain

$$\begin{aligned}
&\mathbb{E}[\mathcal{L}_{t+1}] \\
&\leq (1-\alpha_t\mu_\sigma)\mathbb{E}\left[\|\theta_t - \theta_{K_t}\|_F^2\right] + \frac{1}{4}\frac{\sigma_{min}(D_\epsilon)}{c_3}\beta_t\varepsilon \\
&\quad + \left(1 - \beta_t\frac{\sigma_{min}(D_\epsilon)}{c_3}\right)\mathbb{E}\left[(J(K_t) - J(K^*))\right] \\
&< \frac{1}{2}\varepsilon + \frac{1}{2}\varepsilon\left(\frac{1}{2}\frac{\sigma_{min}(D_\epsilon)}{c_3}\beta_t + 1 - \beta_t\frac{\sigma_{min}(D_\epsilon)}{c_3}\right) < \varepsilon.
\end{aligned}$$

Therefore, we have shown that under (4.33) and (4.34), the Lyapunov function is decreasing at rate (4.71) as long as $\mathbb{E}[\|\theta_t - \theta_{K_t}\|_F^2] \geq \frac{1}{2}\varepsilon$ or $\mathbb{E}[J(K_t) - J(K^*)] \geq \frac{1}{2}\varepsilon$, or else, the Lyapunov function will keep being smaller than ε. Since $(1-\beta_t c_4)^T\mathcal{L}_0 < \varepsilon$ (recall that β_t is constant in t), we have $\mathbb{E}[\mathcal{L}_T] \leq \varepsilon$. Since $\mathbb{E}[\mathcal{L}_T]$ is the sum of two non-negative numbers, both of them are less than ε. □

Chapter 5

A Policy Gradient Framework for Stochastic Optimal Control Problems with Global Convergence Guarantee

5.1 Introduction

The stochastic optimal control problem is an important field of study and has a wide range of applications, such as financial portfolio investment [Pha09], manufacturing system management [SYZZ02], climate policy decisions [BHM08], disease control and prevention [LLLO22], and multiagent path finding [ONL+22], just to name a few. It also has natural connections with machine learning, as it can be viewed as a reinforcement learning problem with continuous time [WZZ20].

Given its importance, extensive research has been devoted to solving the optimal control problem, based on fundamental tools as dynamic programming [Ber12], Pontryagin's maximum principle [Kop62], and Hamilton–Jacobi–Bellman (HJB) equations [BD+97]. Conventional methods, such as finite difference [BZ03], finite elements [Kus90], and method of successive approximations (MSA) [CL82], face difficulties when the dimensionality of the problem becomes high. Thus, in recent years, machine learning based methods have emerged as powerful tools to solve the optimal control problem, with studies such as Ji et al. applying the stochastic maximum principle with neural network parametrization [JPPZ22], Ruthotto et al. using deep learning to solve deterministic control problems via characteristic lines [ROL+20], and Zhou et al. developing an actor-critic framework to solve the HJB equation through deep learning [ZHL21].

Despite the great empirical success in solving the optimal control problem, the theoretical studies of such algorithms are still lacking. In this work, we analyze the theoretical properties of a policy gradient method for the optimal control problem in a quite general setting. In particular, our analysis covers stochastic optimal control with controlled diffusion (the diffusion coefficient is part of the control), which leads to a fully nonlinear HJB equation for the value function, making the analysis significantly more difficult. In order to obtain the exact optimal control, we set the control as a deterministic function of time and state, without entropy regularization.

We consider a policy gradient method for solving the control problem, in the continuous time limit, the algorithm can be viewed as a gradient flow of the control function with respect to the cost functional. We establish the global convergence of the gradient flow under relatively mild regularity assumptions, while the problem is not convex. The proof for the convergence of the gradient flow is based on the construction of a barrier function, which is motivated by the uniqueness theory of the viscosity solution to the HJB equation. We also design a local optimal control function in order to distinguish a key criterion for proving a convergence rate. This local optimal function is crucial for the proof of global convergence as the cost functional is not convex, so that one cannot apply standard tools in convex optimization.

5.1.1 Related works

Before we present the framework and our results. We summarize a few theoretical studies on the optimal control problem that are related to ours.

For the analysis of numerical methods for optimal control, many studies focus on specific settings, such as the linear quadratic regulator (LQR) problem, which is thoroughly studied due to its simple structure. Its optimal control has an explicit expression w.r.t. the value function, which makes the HJB equation semi-linear.

Wang et al. analyze the global convergence of a policy gradient method for LQR [WHYW21]. Gobet and Grangereau develop a Newton's method for control problems with linear dynamics and show quadratic convergence [GG22].

Another well-studied scenario is the control problem with a soft policy to encourage exploration. Such global search makes the convergence analysis more feasible [Zho21]. For instance, in the context of mean-field games, the convergence of policies to the Nash equilibrium is guaranteed under mild assumptions [DEJM$^+$20, FJ22, GXZ22]. Tang et al. analyze the property of a class of soft policy algorithms where the rate of exploration decreases to 0 [TZZ22].

In more general settings, a variety of recent works focus on studying the convergence of algorithms. Carmona and Laurière analyze the approximation errors with neural networks for linear quadratic (LQ) mean-field games [CL21] and general mean-field games [CL22]. Kerimkulov et al. study the convergence and stability of a Howard's policy improvement algorithm [KSS20a]. Ito et al. investigate an iterative method with a superlinear convergence rate [IRZ21]. However, these studies do not involve controlled diffusion. As Yong and Zhou point out, the presence of control in diffusion will make the control problem significantly more difficult, even in the LQ scenario [YZ99].

Perhaps the research by Kerimkulov et al. is the most related to ours. They study the convergence rate of a MSA algorithm in a controlled diffusion setting [KŠS21]. However, their work assumes solution of a global optimization to update the control, and is hence not a practical algorithm.

5.2 Theoretical background: the stochastic optimal control problem

We clarify some notations first. Let $\mathcal{X}$ be an n-dimensional unit flat torus, i.e., $\mathcal{X} = [0,1]^n$ with periodic boundary condition. We use $|\cdot|$ to denote the absolute value of a scalar, the l^2 norm of a vector, the Frobenius norm of a matrix, or the square root of the square sum of a higher order tensor according to the context. $\|\cdot\|_{L^1}$ and $\|\cdot\|_{L^2}$ denote the L^1 and L^2 norm of a function. $\|\cdot\|_2$ denotes the l^2 operator norm (i.e. the largest singular value) of a matrix. $\langle\cdot,\,\cdot\rangle$ denotes the inner product between two vectors, and $\langle\cdot,\,\cdot\rangle_{L^2}$ denotes the inner product between two L^2 functions. $\mathrm{Tr}(\cdot)$ denotes the trace of a squared matrix.

In this work, we consider the optimal control problem on $\mathcal{X}$ during a time period $t \in [0,T]$. Let $(\Omega, \mathcal{F}, \{\mathcal{F}_t\}, \mathbb{P})$ be a filtered probability space. Let x_t be the state trajectory in $\mathcal{X}$ satisfying the stochastic differential equation (SDE)

$$\mathrm{d}x_t = b(x_t, u_t)\mathrm{d}t + \sigma(x_t, u_t)\mathrm{d}W_t, \tag{5.1}$$

where $b(x,u) : \mathcal{X}\times\mathbb{R}^{n'} \to \mathbb{R}^n$ and $\sigma(x,u) : \mathcal{X}\times\mathbb{R}^{n'} \to \mathbb{R}^{n\times m}$ are the drift and diffusion coefficients, $u_t \in \mathbb{R}^{n'}$ is an $\mathcal{F}_t$-adapted control process, and W_t is an m-dimensional $\mathcal{F}_t$-Brownian motion. With a slight abuse of notation, the letter u may refer to the control process or a vector in $\mathbb{R}^{n'}$. The initial point x_0 is uniformly distributed in $\mathcal{X}$ unless otherwise specified. Throughout the paper, we assume that the matrix valued function $\frac{1}{2}\sigma\sigma^\top =: D \in \mathbb{R}^{n\times n}$ is uniformly elliptic with minimum eigenvalue $\lambda_{\min} \geq \sigma_0 > 0$. The goal of the optimal control problem is to minimize the cost functional

$$J[u] = \mathbb{E}\left[\int_0^T r(x_t, u_t)\,\mathrm{d}t + h(x_T)\right] \tag{5.2}$$

over all admissible controls, where $r(x,u) : \mathcal{X}\times\mathbb{R}^{n'} \to \mathbb{R}$ is the running cost and $h(x) : \mathcal{X} \to \mathbb{R}$ is the terminal cost. To study the optimal control problem, the value

function is very important and useful, which is defined as the expected cost if we start at a certain time and location:

$$V_u(t,x) = \mathbb{E}\left[\int_t^T r(x_s,u_s)\,\mathrm{d}s + h(x_T) \;\middle|\; x_t = x\right]. \tag{5.3}$$

Here, the subscript u indicates V_u is the value function w.r.t. the control process u. By the Markov property, we can verify that $V_u(t,x)$ satisfies the Bellman equation

$$V_u(t_1,x) = \mathbb{E}\left[\int_{t_1}^{t_2} r(x_t,u_t)\,\mathrm{d}t + V_u(t_2,x_{t_2}) \;\middle|\; x_{t_1} = x\right] \tag{5.4}$$

for any $0 \le t_1 \le t_2 \le T$ and $x \in \mathcal{X}$.

The existence of an optimal control that minimizes (5.2) is well-studied (see e.g., the book [YZ99]). In this work, we assume the optimal control exists and we denote u_t^* the optimal control, x_t^* the optimal state process, and $V^*(t,x) = V_{u^*}(t,x)$ the optimal value function. By the dynamic programming principle [FR12],

$$V^*(t_1,x) = \inf_u \mathbb{E}\left[\int_{t_1}^{t_2} r(x_t,u_t)\,\mathrm{d}t + V^*(t_2,x_{t_2}) \;\middle|\; x_{t_1} = x\right],$$

where the infimum is taken over all the admissible control that coincide with u^* in $[0,t_1] \cup [t_2,T]$. This dynamic programming principle informs us that we have the optimal solution from t_1 to T if we optimize the control from t_1 to t_2 w.r.t. the loss $\mathbb{E}[\int_{t_1}^{t_2} r(x_t,u_t)\,\mathrm{d}t + V^*(t_2,x_{t_2})]$ and apply u^* after t_2. Based on this principle, let $t_2 \to t_1$, we see that the optimal control u^* at time t is a deterministic function of the state x_t, at least heuristically. For a rigorous argument, we refer the reader to the verification theorem (see, for example, [YZ99] section 5.3.5). Therefore, we will only consider the controls as a function of t and x and we will use the shorthand u_t for $u(t,x_t)$ when there is no confusion. The objective function becomes

$$J[u] = \mathbb{E}\left[\int_0^T r(x_t,u(t,x_t))\,\mathrm{d}t + h(x_T)\right],$$

where u belongs to a class of admissible control functions to be specified later.

Let $\rho^u(t,x)$ be the density function of x_t under control function u, then ρ^u satisfies the Fokker Planck equation (see e.g., [Ris96])

$$\partial_t \rho^u(t,x) = -\nabla_x \cdot [b(x,u(t,x))\rho^u(t,x)] + \sum_{i,j=1}^{n} \partial_i \partial_j \left[D_{ij}(x,u(t,x))\rho^u(t,x)\right], \quad (5.5)$$

where we denote $\partial_i = \partial_{x_i}$ for simplicity. The initial condition $\rho^u(0,\cdot)$ is the density of x_0. For example, $\rho^u(0,\cdot) \equiv 1$ if $x_0 \sim \mathrm{Unif}(\mathcal{X})$ and $\rho^u(0,\cdot) = \delta_{x_0}$ if x_0 is deterministic. If we denote $\mathcal{G}_u$ the infinitesimal generator of the SDE (5.1) under control u, then, the Fokker Planck equation becomes $\partial_t \rho^u = \mathcal{G}_u^\dagger \rho^u$, where $\mathcal{G}_u^\dagger$ is the adjoint operator of $\mathcal{G}_u$.

One important tool to study the optimal control problem is the adjoint equation. We introduce the adjoint state $p_t = -\nabla_x V_u(t,x_t)$ (also known as the shadow price [AC79]) and also $q_t = -\nabla_x^2 V_u(t,x_t)\sigma(x_t,u_t)$. Then, $(p_t,q_t) \in \mathbb{R}^n \times \mathbb{R}^{n\times m}$ is the unique solution to the following backward stochastic differential equations (BSDE) [PP92]

$$\begin{cases} \mathrm{d}p_t = -\left[\nabla_x b(x_t,u_t)^\top p_t + \nabla_x \operatorname{Tr}\left(\sigma(x_t,u_t)^\top q_t\right) - \nabla_x r(x_t,u_t)\right]\mathrm{d}t + q_t\ \mathrm{d}W_t \\ \qquad = -\nabla_x H\left(t,x_t,u_t,p_t,q_t\right)\mathrm{d}t + q_t\ \mathrm{d}W_t \\ p_T = -\nabla_x h(x_T), \end{cases} \quad (5.6)$$

known as the first order adjoint equation. Here,

$$H(t,x,u,p,q) := \operatorname{Tr}\left(q^\top \sigma(x,u)\right) + \langle p,\, b(x,u)\rangle - r(x,u)$$

is the Hamiltonian. We also define the generalized Hamiltonian as

$$G(t,x,u,p,P) := \frac{1}{2}\operatorname{Tr}\left(P\sigma(t,x,u)\sigma(t,x,u)^\top\right) + \langle p,\, b(x,u)\rangle - r(x,u). \quad (5.7)$$

Then, the value function V_u in (5.3) is also the solution to the Hamilton–Jacobi (HJ)

equation

$$\begin{cases} -\partial_t V_u(t,x) + G\left(t, x, u(t,x), -\nabla_x V_u(t,x), -\nabla_x^2 V_u(t,x)\right) = 0 \\ V_u(T,x) = h(x). \end{cases} \tag{5.8}$$

Moreover, the optimal value function V^* satisfies the HJB equation

$$\begin{cases} -\partial_t V^*(t,x) + \sup_u G\left(t, x, u, -\nabla_x V^*(t,x), -\nabla_x^2 V^*(t,x)\right) = 0 \\ V^*(T,x) = h(x). \end{cases} \tag{5.9}$$

Therefore, one necessary condition for a control u to be optimal is that

$$u(t,x) = \arg\max_{u \in \mathbb{R}^m} G\left(t, x, u, -\nabla_x V_u(t,x), -\nabla_x^2 V_u(t,x)\right). \tag{5.10}$$

Since σ involves the control u, this HJB equation is fully nonlinear, making it hard to solve. The existence and uniqueness for the (viscosity) solution of HJB equation is well established (see [BD$^+$97] for example). To simplify notation, We will denote $G(t,x,u,-\nabla_x V(t,x), -\nabla_x^2 V(t,x))$ by $G(t,x,u,-\nabla_x V, -\nabla_x^2 V)$ in the rest of the paper. We define the $H^2([0,T];\mathcal{X})$ norm of a value function $V(t,x)$ by

$$\|V\|^2_{(T,H^2)} := \int_0^T \int_{\mathcal{X}} \left(|V(t,x)|^2 + |\nabla_x V(t,x)|^2 + \left|\nabla_x^2 V(t,x)\right|^2 \right) \mathrm{d}x \, \mathrm{d}t.$$

5.3 A policy gradient method for the control problem

Policy gradient is one of the most popular methods for reinforcement learning problems [SMSM99]. It updates parametrized policy using gradient based method such as gradient descent. This method also applies to the control problem with continuous time [Mun06], which can be viewed as a instantaneous but local dynamic programming method through gradient descent. In order to design a policy gradient method,

we present the following proposition, giving an explicit expression for the derivative of the cost functional (5.2) w.r.t. the control function.

Proposition 4. *Let u be a control function and V_u be the corresponding value function. Let the state process x_t start with uniform distribution in $\mathcal{X}$ and follows the SDE (5.1) with control u. Let $\rho^u(t,\cdot)$ be the density function of x_t. Then,*

$$\frac{\delta J}{\delta u}(t,x) = -\rho^u(t,x)\ \nabla_u G\left(t,x,u(t,x),-\nabla_x V_u,-\nabla_x^2 V_u\right). \tag{5.11}$$

As a generalization,

$$\frac{\delta V_u(s,y)}{\delta u}(t,x) = -\mathbb{1}_{\{t\geq s\}}\, p^u(t,x;s,y)\ \nabla_u G\left(t,x,u(t,x),-\nabla_x V_u,-\nabla_x^2 V_u\right) \tag{5.12}$$

for all $x,y\in\mathcal{X}$, where $p^u(t,x;s,y)$ is the fundamental solution to the Fokker-Planck equation (5.5).

Proof sketch. We pick an arbitrary perturbation function $\phi(t,x)$, and perturb the control infinitesimally, then

$$\frac{\mathrm{d}}{\mathrm{d}\varepsilon} J[u+\varepsilon\phi]\bigg|_{\varepsilon=0} = \left\langle \frac{\delta J}{\delta u},\, \phi \right\rangle_{L^2}.$$

Therefore, in order to show (5.11), it suffices to show

$$\frac{\mathrm{d}}{\mathrm{d}\varepsilon} J[u+\varepsilon\phi]\bigg|_{\varepsilon=0} = -\int_0^T \int_{\mathcal{X}} \rho^u(t,x)\nabla_u G\left(t,x,u(t,x),-\nabla_x V_u,-\nabla_x^2 V_u\right)\phi(t,x)\ \mathrm{d}x\mathrm{d}t.$$

The rest of work is to substitute in the definitions and do some tedious calculation, which is postponed to Section 5.6. (5.12) can be established in an analogous way. □

Remark. *We remark that a similar result is established in [CL21] (Proposition 2) under a different setting. That work considers an infinite horizon mean-field games with constant diffusion coefficient, while our setting is finite horizon stochastic control problem with controlled diffusion.*

Motivated by proposition 4, we want to construct a continuous-time version of the policy gradient, i.e., the gradient flow of J with respect to u. Let us recall the gradient descent method in discrete time first. Using τ_k as time steps for the policy gradient, for any $(t,x) \in [0,T] \times \mathcal{X}$, the gradient descent method is

$$\begin{aligned} u^{\tau_{k+1}}(t,x) &= u^{\tau_k}(t,x) - \Delta\tau_k \frac{\delta J}{\delta u^{\tau_k}}(t,x) \\ &= u^{\tau_k}(t,x) + \Delta\tau_k \ \rho^{u^{\tau_k}}(t,x) \ \nabla_u G\left(t,x,u^{\tau_k}(t,x), -\nabla_x V_{u^{\tau_k}}, -\nabla_x^2 V_{u^{\tau_k}}\right), \end{aligned} \tag{5.13}$$

where $\Delta\tau_k = \tau_{k+1} - \tau_k$ is the step size and $V_{u^{\tau_k}}$ is the value function w.r.t. the control u^{τ_k}.

This gradient descent method (5.13) can be implemented in practice noticing that $\rho^{u^{\tau_k}}(t,x)$ is the density function. One just needs to sample the gradient $\nabla_u G$ using trajectories of x_t. For example, under some parametrization of the control function $u(t,x;\theta)$, [1] where θ denotes the collective parameters. We seek to update the parameters $\theta_{k+1} = \theta_k + \Delta\theta$ such that

$$u(t,x;\theta_{k+1}) \approx u(t,x;\theta_k) + \Delta\tau_k \rho^{u_k}(t,x) \nabla_u G(t,x,u_k, -\nabla_x V_{u_k}, -\nabla_x^2 V_{u_k}) \tag{5.14}$$

for all $(t,x) \in [0,T] \times \mathcal{X}$, where $u_k = u(t,x;\theta_k)$.

To sample the trajectories, we can use the Euler-Maruyama scheme

$$x_{i+1} = x_i + b(x_i, u_k(t_i,x_i))\Delta t + \sigma(x_i, u_k(t_i,x_i))\sqrt{\Delta t}\,\xi \tag{5.15}$$

with $x_0 \sim \text{Unif}(\mathcal{X})$ and $\xi \sim N(0, I_m)$ to approximate the SDE (5.1) numerically. Here $\Delta t = T/N$ is the time step size and $t_i = i\Delta t$ $(i = 0, 1, \ldots, N)$. Multiple numerical trajectories can be sampled from (5.15), which provides a set

$$\mathcal{S}_k := \{(t_i, x_i^{(j)}) \mid j \geq 1, \ i = 0, 1, \ldots N-1\}$$

[1] This parametrization could be grid, finite element, or neural network (see [HLZ20] for an example of neural network with periodic boundary condition).

where j is the index of samples. Then the update (5.14) can be achieved by a least square regression

$$\min_{\Delta\theta} \sum_{(t,x)\in\mathcal{S}_k} \left| \frac{\partial u(t,x;\theta_k)^\top}{\partial \theta} \Delta\theta - \Delta\tau_k \, \nabla_u G(t,x,u_k,-\nabla_x V_{u_k}, -\nabla_x^2 V_{u_k}) \right|^2 .$$

Let us return to the continuous dynamic. Let $\Delta_\tau = \max_k \Delta\tau_k \to 0$, the update scheme of the control (5.13) converges to the gradient flow in continuous time

$$\frac{\mathrm{d}}{\mathrm{d}\tau} u^\tau(t,x) = -\frac{\delta J}{\delta u^\tau}(t,x) = \rho^{u^\tau}(t,x) \; \nabla_u G\left(t,x,u^\tau(t,x), -\nabla_x V_{u^\tau}, -\nabla_x^2 V_{u^\tau}\right). \quad (5.16)$$

By the definition of value function (5.3) and Proposition 4, we can show that $V_{u^\tau}(t,x)$ is decreasing in τ (for fixed t and x).

Proposition 5 (Monotonicity of value function in τ)**.** *Under the policy gradient dynamic* (5.16)*, the value function $V_{u^\tau}(t,x)$ is decreasing in τ for all $(t,x) \in [0,T] \times \mathcal{X}$.*

Proof. By Proposition 4, for any $(t,x) \in [0,T] \times \mathcal{X}$,

$$\frac{\mathrm{d}}{\mathrm{d}\tau} V_{u^\tau}(t,x) = \left\langle \frac{\delta V_{u^\tau}(t,x)}{\delta u}, \frac{\mathrm{d}}{\mathrm{d}\tau} u^\tau \right\rangle_{L^2}$$

$$= -\int_t^T \int_{\mathcal{X}} p^{u^\tau}(s,y;t,x) \rho^{u^\tau}(s,y) \left| \nabla_u G\left(s,y,u^\tau(s,y), -\nabla_x V_{u^\tau}, -\nabla_x^2 V_{u^\tau}\right) \right|^2 \mathrm{d}r \, \mathrm{d}s \le 0,$$

where we conclude using the fact that density functions $p^{u^\tau}(s,y;t,x)$ and $\rho^{u^\tau}(s,y)$ are non-negative. □

Proposition 4 also indicates that the policy gradient method requires the derivatives of value function $\nabla_x V_u$ and $\nabla_x^2 V_u$ for a given control, which reduces to solving the linear parabolic HJ equation (5.8). Computing the value function with given policy is called a policy evaluation process in reinforcement learning [Hec92]. Such

structure naturally motivates us to use the actor-critic framework [KT99], where the policy gradient and policy evaluation are operated jointly. In our setting, the gradient flow (5.16) can be viewed as the limit of a two time-scale actor-critic method [WZXG20], where the speed of policy evaluation is infinitely faster than policy gradient. While it is possible to extend our analysis to the actor-critic method, we choose to focus on the dynamic (5.16) in this paper and leave that to future works.

5.4 Convergence of the policy gradient

In this section, we give the main results of this work on convergence of the policy gradient. First, we give some technical assumptions

Assumption 2. *1. r, b, and σ are smooth, with $C^4(\mathcal{X} \times \mathbb{R}^{n'})$ norm bounded by some constant K.*

2. h is smooth with $C^4(\mathcal{X})$ norm bounded by K.

3. There exists unique solution $V^* \in C^{1,2}([0,T];\mathcal{X})$ to the HJB equation (5.9) in the classical sense. The optimal control function u^* is smooth and

$$\|u^*\|_{C^{2,4}([0,T];\mathcal{X})} \leq K.$$

In order to avoid tedious technicality and focus on the main ideas of the analysis, we also make a regularity assumption on the control function through the gradient flow (5.16).

Assumption 3. *The control function u^τ remains smooth through the gradient flow (5.16), and there exists a constant K such that $\|u^\tau\|_{C^{2,4}([0,T];\mathcal{X})} \leq K$.*

Let us define a set

$$\mathcal{U} = \left\{ u(t,x) \;\middle|\; u \text{ is smooth and } \|u\|_{C^{2,4}([0,T];\mathcal{X})} \leq K \right\}$$

to include all the regular control functions we consider. We make a few remarks about the assumptions.

- It follows from definition that D is also smooth, and we also use K to denote its $C^4(\mathcal{X} \times \mathbb{R}^{n'})$ bound. Since the control function $u(t,x) \in \mathcal{U}$ is bounded, we just need r, b, σ, D has bounded derivative when the input vector $u \in \mathbb{R}^{n'}$ is within this bounded range. Similar boundedness assumptions are very common, such as in [ŠS20].

- When Assumption 2 holds and $u \in \mathcal{U}$, we have the solution $V_u \in C^{1,2}([0,T] \times \mathcal{X})$ to the HJ equation (5.8) in classical sense.

- Regarding the third condition in Assumption 2, the existence and uniqueness of viscousity solution is well established under mild assumption [BD$^+$97]. If the solution $V^* \in C^{1,2}([0,T]; \mathcal{X})$, it is also a classical solution.

- When Assumption 2 holds and $u \in \mathcal{U}$, we know from Schauder estimate [LSU88] that V_u has fourth order derivative in x. So, $V_u \in C^{1,4}([0,T]; \mathcal{X})$. Then, we observe that $G\left(t, x, u(t,x), -\nabla_x V_u(t,x), -\nabla_x^2 V_u(t,x)\right)$ in (5.8) is differentiable in t, which implies $V_u \in C^{2,4}([0,T]; \mathcal{X})$. We will also use K to denote the bound for $C^{2,4}([0,T]; \mathcal{X})$ norm of V_u. The same argument also holds for V^*, so $\|V^*\|_{C^{2,4}} \leq K$.

- We assume boundedness of r, b, and σ in Assumption 2 as the state space $\mathcal{X}$ is compact. In the setting with unbounded state space such as $\mathbb{R}^n$, the common assumption is that the functions grows at most linearly in $|x|$ [YZ99].

- In Assumption 2, the bounded derivative implies a Lipschitz condition. For example, if $|\nabla_x b| \leq K$, then $|b(x_1,u) - b(x_2,u)| \leq K\,|x_1 - x_2|$. In the proofs,

we will use L instead of K whenever we use the Lipschitz condition, in order to be more reader friendly.

Under these assumptions, we have a lower bound and an upper bound for the density function on our compact space $\mathcal{X}$.

Proposition 6. *Under Assumption 2, let $u \in \mathcal{U}$ and ρ^u be the solution to the Fokker-Planck equation* (5.5) *with initial condition $\rho^u(0,x) \equiv 1$. Then $\rho^u(t,x)$ has a positive lower bound ρ_0 and an upper bound ρ_1 that only depend on n, T, and K.*

The proof of this proposition can be found in Section 5.6.

We now state convergence results of the gradient flow (5.16). We give a warm-up theorem about its critical point.

Theorem 2 (Critical point for policy gradient)**.** *Under Assumption 2, assume further that G is strongly concave in u. Then, any critical point of the gradient flow* (5.16) *is an optimal control.*

Remark. *Similar to the remark under Assumptions 2, 3 above, we only require the concavity of G when $p = -\nabla_x V_u$ and $P = -\nabla_x^2 V_u$ are within a bounded range given by the Schauder estimates.*

Proof. Let u be a critical point of (5.16). Then since ρ^u is not vanishing by Proposition 6 (lower bound), we have

$$\nabla_u G\left(t, x, u(t,x), -\nabla_x V_u, -\nabla_x^2 V_u\right) = 0.$$

Since G is strongly concave in u, the control function $u(t,x)$ satisfies the maximum condition (5.10). Therefore, V_u is not only the solution to the HJ equation (5.8), but also the solution to the HJB equation (5.9). Then, by the uniqueness of HJB equation, $u(t,x)$ is the optimal control. □

In order to explain the necessity of the strong concavity assumption of G in u, we give a counter-example in Section 5.5, showing that there are multiple critical points of the policy gradient dynamics (5.16) when the concavity assumption does not hold. It is also clear that the commonly studied LQR problem satisfies this strong concavity assumption [WHYW21]. We shall emphasize that this concavity assumption does not imply that the optimal control problem (5.2) is convex in u. In fact the cost functional is in general non-convex.

Next, we state our main results, establishing the convergence guarantee of gradient flow (5.16).

Theorem 3 (Convergence of the policy gradient)**.** *Let Assumption 2 and 3 hold. Further assume that G is uniformly strongly concave in u. Then, the gradient flow* (5.16) *of control satisfies*

$$\lim_{\tau\to\infty} J[u^\tau] = J[u^*]. \tag{5.17}$$

Here, by uniformly strongly concave, we mean that there exists an absolute constant $\mu_G > 0$ such that the family of functions $G(t, x, \cdot, p, P)$ is μ_G-strongly concave for all (t, x, p, P) within the range we care. Given Theorem 3, one natural question is whether one can establish a convergence rate for the policy gradient. The answer is yes, with a mild extra assumption to avoid flatness.

Assumption 4. *There exists a modulus of continuity $\omega : [0, \infty) \to [0, \infty)$ such that*

$$\|u - u^*\|_{L^2} \le \omega(J[u] - J[u^*])$$

for any $u \in \mathcal{U}$. Here u^* is the optimal control.

With this assumption, Theorem 3 guarantees us that $\|u^\tau - u^*\|_{L^2} \to 0$ as $\tau \to \infty$. Therefore, we just have to analyze when u^τ is sufficient close to u^* in order to get a global convergence rate.

Theorem 4 (Convergence rate of the policy gradient)**.** *Let Assumption 2, 3, and 4 hold. Further assume that G is uniformly strongly concave in u. Then, the gradient flow* (5.16) *of control satisfies*

$$J[u^\tau] - J[u^*] \le e^{-c\tau}\left(J[u^0] - J[u^*]\right) \tag{5.18}$$

for some positive constant c. As a direct corollary, $\|u^\tau - u^*\|_{L^2} \to 0$.

The proofs for Theorem 3 and 4 overlap quite a bit, so we will prove them together. We give a sketch of the proof here and leave a detailed version to Section 5.8. Throughout all the proofs, we will use C to denote some absolute constant that only depends on n, K, T, which may change according to the context.

Proof sketch for Theorems 3 and 4. For a control function $u(t,x)$, we define a corresponding "local optimal" control function as

$$u^\diamond(t,x) := \arg\max_{u\in\mathbb{R}^{n'}} G(t,x,u,-\nabla_x V_u(t,x), -\nabla_x^2 V_u(t,x)). \tag{5.19}$$

We call $u^\diamond$ "local optimal" because the maximum condition (5.10) is satisfied, but w.r.t. the current value function u instead of $u^\diamond$. So, $|u(t,x) - u^\diamond(t,x)|$ can measure the distance between the current control u and the local optimal one at (t,x). Under the strong convexity assumption of G in u, we know that $u^\diamond$ is well defined and

$$\left|\nabla_u G(t,x,u(t,x), -\nabla_x V_u, -\nabla_x^2 V_u)\right| \ge \mu_G \left|u(t,x) - u^\diamond(t,x)\right|.$$

For each control function u^τ along the gradient flow (5.16), we thus define a corresponding "local optimal" control function $u^{\tau\diamond}$.

For the convergence analysis, recall that establishing the Polyak-Łojasiewicz (PL) condition [KNS16] is one typical way to show the convergence of gradient descent (or gradient flow). Accordingly, we separate into two scenarios through a condition (see (5.20)). Indeed, the local optimal control function (5.19) is introduced to establish a

condition for u^τ to distinguish the two scenarios. When the assumption (5.20) holds (*Case 1*), it directly implies the PL condition, and thus the convergence analysis follows easily. The non-trivial task is of course to analyze the scenario when this assumption does not hold (*Case 2*), which consists of the main technical work.

Case 1. There exist positive constants μ and τ_0 such that

$$\|u^\tau - u^{\tau\diamond}\|_{L^2} \geq \mu \|u^\tau - u^*\|_{L^2} \tag{5.20}$$

for all $\tau \geq \tau_0$. Under such condition, we have

$$\begin{aligned}
&\frac{\mathrm{d}}{\mathrm{d}\tau} J[u^\tau] = \left\langle \frac{\delta J}{\delta u}[u^\tau],\, \frac{\mathrm{d}}{\mathrm{d}\tau} u^\tau \right\rangle \\
&= -\left\| \rho^{u^\tau}(t,x) \nabla_u G(t,x,u^\tau(t,x), -\nabla_x V_{u^\tau}, -\nabla_x^2 V_{u^\tau}) \right\|_{L^2}^2 \\
&\leq -\rho_0^2 \left\| \nabla_u G(t,x,u^\tau(t,x), -\nabla_x V_{u^\tau}, -\nabla_x^2 V_{u^\tau}) \right\|_{L^2}^2 \leq -\rho_0^2 \mu_G^2 \|u^\tau - u^{\tau\diamond}\|_{L^2}^2 \\
&\leq -\rho_0^2 \mu_G^2 \mu^2 \|u^\tau - u^*\|_{L^2}^2 \leq -\rho_0^2 \mu_G^2 \mu^2 \frac{1}{C_3} \left(J[u^\tau] - J[u^*] \right),
\end{aligned}$$

where the last inequality, stating that the cost function J grows at most quadratically in u, is established in Section 5.7 Lemma 10. This is nothing but the PL inequality, leading to exponential convergence of the cost function. Therefore, the two theorems hold.

Case 2. Next, we consider the case when assumption (5.20) does not hold. Then, we can find a sequence $\{\tau_k\}$, increasing to infinity, such that for each k

$$\|u^{\tau_k} - u^{\tau_k\diamond}\|_{L^2} \leq \frac{1}{k} \|u^{\tau_k} - u^*\|_{L^2}.$$

Denoting u^{τ_k} and $V_{u^{\tau_k}}$ by u_k and V_k, we rewrite the above as

$$\|u_k - u_k^\diamond\|_{L^2} \leq \frac{1}{k} \|u_k - u^*\|_{L^2}. \tag{5.21}$$

By Proposition 5, the value function $V_k(t,x)$ is decreasing in k, so it has a pointwise limit $V_\infty(t,x)$. Since $V_k(t,x) \geq V^*(t,x)$, we have $V_\infty(t,x) \geq V^*(t,x)$. We claim that

$$V_\infty(t,x) \equiv V^*(t,x). \tag{5.22}$$

This claim is proved in Section 5.8 Lemma 13, and a proof sketch can be found below.

When (5.22) holds, we know that $V_{u^\tau}(0,\cdot)$ converges to $V^*(0,\cdot)$ uniformly using the Lipschitz condition and Arzelá–Ascoli theorem. Therefore, using the relationship

$$J[u^\tau] = \int_{\mathcal{X}} \rho^{u^\tau}(0,x) V_{u^\tau}(0,x)\,\mathrm{d}x \quad \text{and} \quad J[u^*] = \int_{\mathcal{X}} \rho^{u^*}(0,x) V^*(0,x)\,\mathrm{d}x,$$

where $\rho^{u^\tau}(0,\cdot) = \rho^{u^*}(0,\cdot)$ (as the initial distribution of state does not depend on control), we establish (5.17) and thus Theorem 3.

Next, we prove the convergence rate as in Theorem 4. By Assumption 4 and the result of Theorem 3 we just proved, we know that $\lim_{\tau\to\infty} \|u^\tau - u^*\|_2 = 0$, hence $\lim_{k\to\infty} \|u_k - u^*\|_{L^2} = 0$. For fixed (t,x), $u^\diamond(t,x)$ defined in (5.19) can be viewed as an implicit function of $-\nabla_x V_u(t,x)$ and $-\nabla_x^2 V_u(t,x)$, given by the equation

$$\nabla_u G(t,x,u^\diamond(t,x), -\nabla_x V_u(t,x), -\nabla_x^2 V_u(t,x)) = 0.$$

We show a Lipschitz condition of this implicit function in Section 5.7 Lemma 12:

$$|u_k^\diamond(t,x) - u^*(t,x)| \leq C_5 \left(|\nabla_x V_k(t,x) - \nabla_x V^*(t,x)| + \left|\nabla_x^2 V_k(t,x) - \nabla_x^2 V^*(t,x)\right| \right).$$

Therefore,

$$\|u_k^\diamond - u^*\|_{L^2} \leq \sqrt{2} C_5 \|V_k - V^*\|_{(T;H^2)} \leq \sqrt{2} C_5 C_4 \|u_k - u^*\|_{L^2}^{1+\alpha},$$

where the second inequality is established in Section 5.7 Lemma 11, with $\alpha = \frac{1}{n+3} > 0$.

Therefore, we obtain

$$\|u_k - u^*\|_{L^2} \leq \|u_k - u_k^\diamond\|_{L^2} + \|u_k^\diamond - u^*\|_{L^2} \leq \frac{1}{k} \|u_k - u^*\|_{L^2} + C \|u_k - u^*\|_{L^2}^{1+\alpha}.$$

However, this cannot hold when k is sufficiently large (i.e., when $\|u_k - u^*\|_{L^2}$ is sufficiently small), so we reach a contradiction. Therefore, (5.21) cannot hold under Assumption 4. Hence only *Case 1* is possible, so Theorem 4 holds. □

Proof sketch for the claim (5.22). The idea to prove (5.22) comes from the proof for the uniqueness of the (viscosity) solution of the HJB equation, where a barrier function is constructed (see e.g., [YZ99] Chapter 4.6). V^* is the unique solution of the HJB equation. V_∞ is the limit of V_k, with u_k satisfying (5.21). Therefore, V_k is very "close" to the solution of the HJB equation in the sense that u_k is very close to $u_k^\diamond$, i.e., the maximum condition (5.10) is "nearly" satisfied.

We assume to the contrary that there exists $(\bar{t}, \bar{x}) \in [0, T] \times \mathcal{X}$ s.t. $V_\infty(\bar{t}, \bar{x}) - V^*(\bar{t}, \bar{x}) \geq \eta > 0$. For any ε, δ, β, $\lambda \in (0, 1)$, we define a barrier function

$$\varphi(t, x, s, y) := \frac{1}{2\varepsilon} |t - s|^2 + \frac{1}{2\delta} |x - y|^2 - \beta(t + s) + \frac{\lambda}{t} + \frac{\lambda}{s}$$

and define

$$\Phi_k(t, x, s, y) := V_k(t, x) - V^*(s, y) - \varphi(t, x, s, y).$$

We assume $\Phi_k(t, x, s, y)$ achieves its maximum at (t_k, x_k, s_k, y_k), which depends on ε, δ, β, λ, and k. This maximum condition implies $|t_k - s_k|, |x_k - y_k| \to 0$ as $\varepsilon, \delta \to 0$.

When t_k or s_k are close to T, we can use the fact that $V_\infty(T, \cdot) = V^*(T, \cdot) = h(\cdot)$ to derive a contradiction. So we focus on the case when $t_k \vee s_k$ is not close to T. We define a new function

$$\begin{aligned}\widehat{\Phi}_k(t, x, s, y) = \Phi_k(t, x, s, y) - \frac{\mu}{2} \left(|t - t_k|^2 + |x - x_k|^2 + |s - s_k|^2 + |y - y_k|^2 \right) \\ + q(t - t_k) + \langle p, x - x_k \rangle + \widehat{q}(s - s_k) + \langle \widehat{p}, y - y_k \rangle,\end{aligned}$$

where $\mu > 0$ and $(q, p, \widehat{q}, \widehat{p}) \in \mathbb{R}^{1+n+1+n}$ are small. This μ aims to make (t_k, x_k, s_k, y_k) a strict maximum of Φ_k, and $(q, p, \widehat{q}, \widehat{p})$ is a linear perturbation. The motivation to

make this perturbation is that we only have a relation in norm (5.21). So, we want to integrate the inequality we get later over all the local perturbations so that we can compare with (5.21). Let $\widehat{\Phi}_k$ reaches its maximum at $(\widehat{t}_k, \widehat{x}_k, \widehat{s}_k, \widehat{y}_k)$. We can show that $(\widehat{t}_k, \widehat{x}_k, \widehat{s}_k, \widehat{y}_k)$ lies in the interior of the domain because $t_k \vee s_k$ is not close to T. Therefore, we have first and second necessary conditions for optimality

$$\nabla_{t,x,s,y} \widehat{\Phi}_k(\widehat{t}_k, \widehat{x}_k, \widehat{s}_k, \widehat{y}_k) = 0 \quad \text{and} \quad \nabla^2_{x,y} \widehat{\Phi}_k(\widehat{t}_k, \widehat{x}_k, \widehat{s}_k, \widehat{y}_k) \leq 0. \tag{5.23}$$

Next, we estimate the quantity $\partial_s V^*(\widehat{s}_k, \widehat{y}_k) - \partial_t V_k(\widehat{t}_k, \widehat{x}_k)$ in two ways using optimality condition (5.23) and the HJ equations (5.8) for V^* and V_k, which gives us

$$\begin{aligned} 2\beta + 2\lambda/T^2 \leq\ & 8L^4(\varepsilon + \delta) + \mu K(|\widehat{s}_k - s_k| + \left|\widehat{t}_k - t_k\right| + |\widehat{x}_k - x_k| + |\widehat{y}_k - y_k|) \\ & + \mu K^2 + K(|p| + |\widehat{p}| + |q| + |\widehat{q}|) + L\left|u_k(\widehat{t}_k, \widehat{x}_k) - u_k^\diamond(\widehat{t}_k, \widehat{x}_k)\right|. \end{aligned}$$

In order to use the assumption (5.21), we making an integration over a box of side length r, and get

$$\begin{aligned} 2\beta + 2\lambda/T^2 \leq\ & 8L^4(\varepsilon + \delta) + \mu K(1 + \sqrt{n})r + \mu K^2 \\ & + KC\left(\frac{1}{\varepsilon} + \frac{1}{\delta}\right)\sqrt{2n+2}\, r + r^{-2n-2} L\sqrt{T}\, \|u_k - u_k^\diamond\|_{L^2}, \end{aligned}$$

where our regularity assumptions guarantee that $|(q, p, \widehat{q}, \widehat{p})|$ is controlled by r. Finally, we reach a contradiction by choosing appropriate β, λ, μ, ε, δ, r, and k consecutively. □

5.5 A counter example for multiple critical points of the gradient flow

In this section, we give a counter example to show the necessity of the strong concavity of G in u in Assumption 2. Let $n = n' = m = 1$. Consider the HJB equation

$$-\partial_t V(t,x) + \sup_u \left[-\partial_x^2 V(t,x) - \frac{1}{3}u^3 \partial_x V(t,x) - \frac{1}{4}u^4 + \frac{1}{2}u^2\right] = 0$$

with some nice terminal condition $V(T,x) = h(x)$. To simplify the notation, we use V_t, V_{xx}, V_x to denote the derivatives. In order to obtain the optimal control, we need to find the maximum of G. i.e., we seek for the minimum of the quartic function $g(u; V_x) := \frac{1}{4}u^4 + \frac{1}{3}V_x u^3 - \frac{1}{2}u^2$ w.r.t. u for any given $V_x \in \mathbb{R}$ (cf. (5.10)). This quartic function has a local maximum $u = 0$ and two local minimums

$$u_{1,2} = \frac{1}{2}\left(-V_x \pm \sqrt{V_x^2 + 4}\right).$$

With some standard calculus, we obtain the optimal control

$$u^* = \frac{1}{2}\left(-V_x - \text{sign}(V_x)\sqrt{V_x^2 + 4}\right).$$

The HJB equation becomes

$$-V_t(t,x) - V_{xx}(t,x) - g\left(\frac{1}{2}\left(-V_x - \text{sign}(V_x)\sqrt{V_x^2 + 4}\right); V_x\right) = 0, \tag{5.24}$$

which is semilinear. We define a second control (implicitly) through

$$\widetilde{u} = \frac{1}{2}\left(-V_x + \text{sign}(V_x)\sqrt{V_x^2 + 4}\right).$$

The the HJ equation corresponding to this control is

$$-V_t(t,x) - V_{xx}(t,x) - h\left(\frac{1}{2}\left(-V_x + \text{sign}(V_x)\sqrt{V_x^2 + 4}\right); V_x\right) = 0 \tag{5.25}$$

According to standard results in semi-linear parabolic PDE (see [Tay12] Chapter 15 for example), we have unique solutions V^* and $\widetilde{V}$ to (5.24) and (5.25) respectively (if T is not too large). Note that $\widetilde{u}$ is at a local but not global optimal almost everywhere and $\widetilde{V}$ is the value function for $\widetilde{u}$. Therefore, if our policy gradient algorithm reaches $\widetilde{u}$, it becomes static at this local solution. This example demonstrate the necessity of the concavity assumption of the generalized Hamiltonian G in u.

5.6 Proofs for the propositions

Proof of Proposition 4. We fix an arbitrary perturbation function $\phi(t,x)$, then

$$\frac{\mathrm{d}}{\mathrm{d}\varepsilon}J[u+\varepsilon\phi]\bigg|_{\varepsilon=0} = \left\langle \frac{\delta J}{\delta u},\, \phi \right\rangle.$$

We denote x_t^ε the SDE (5.1) under control function $u^\varepsilon := u + \varepsilon\phi$ that start with $x_0^\varepsilon \sim \mathrm{Unif}(\mathcal{X})$. Let $\rho^\varepsilon(t,x)$ be its density. We also denote the corresponding value function by $V^\varepsilon(t,x) := V_{u^\varepsilon}(t,x)$ for simplicity. Then, ρ^ε and V^ε depend continuously on ε (cf. [YZ99] section 4.4.1). By definition of the cost functional (5.2),

$$\begin{aligned} J[u+\varepsilon\phi] &= \mathbb{E}\left[\int_0^T r(x_t^\varepsilon, u^\varepsilon(t,x_t^\varepsilon))\mathrm{d}t + h(x_T^\varepsilon)\right] \\ &= \int_0^T \int_{\mathcal{X}} r(x,u^\varepsilon(t,x))\rho^\varepsilon(t,x)\,\mathrm{d}x\,\mathrm{d}t + \int_{\mathcal{X}} V^\varepsilon(T,x)\rho^\varepsilon(T,x)\,\mathrm{d}x. \end{aligned}$$

Taking derivative w.r.t. ε, and note that $V^\varepsilon(T,x) = h(x)$ does not depend on ε, we obtain

$$\begin{aligned} &\frac{\mathrm{d}}{\mathrm{d}\varepsilon}J[u+\varepsilon\phi] = \int_{\mathcal{X}} V^\varepsilon(T,x)\partial_\varepsilon\rho^\varepsilon(T,x)\,\mathrm{d}x + \\ &+ \int_0^T \int_{\mathcal{X}} \left[\nabla_u r\left(x,u^\varepsilon(t,x)\right)^\top \phi(t,x)\rho^\varepsilon(t,x) + r\left(x,u^\varepsilon(t,x)\right)\partial_\varepsilon\rho^\varepsilon(t,x)\right]\mathrm{d}x\,\mathrm{d}t. \end{aligned} \tag{5.26}$$

In order to compute $V^\varepsilon(T,x)\partial_\varepsilon\rho^\varepsilon(T,x)$ in (5.26), we write down the integral equation

$$V^\varepsilon(T,x)\rho^\varepsilon(T,x) = V^\varepsilon(0,x)\rho^\varepsilon(0,x) + \int_0^T \left[\partial_t\rho^\varepsilon(t,x)V^\varepsilon(t,x) + \rho^\varepsilon(t,x)\partial_t V^\varepsilon(t,x)\right]\mathrm{d}t. \tag{5.27}$$

We also have

$$\begin{aligned} &\partial_\varepsilon V^\varepsilon(0,x)\rho^\varepsilon(0,x) + \int_0^T \left[\partial_t\rho^\varepsilon(t,x)\partial_\varepsilon V^\varepsilon(t,x) + \rho^\varepsilon(t,x)\partial_\varepsilon\partial_t V^\varepsilon(t,x)\right]\mathrm{d}t \\ &= \partial_\varepsilon V^\varepsilon(T,x)\rho^\varepsilon(T,x) = \partial_\varepsilon h(x)\rho^\varepsilon(T,x) = 0. \end{aligned} \tag{5.28}$$

Next, taking derivative of (5.27) w.r.t. ε (note that $V^\varepsilon(T,x)=h(x)$ and $\rho^\varepsilon(0,\cdot)\equiv 1$ do not depend on ε), we obtain

$$\begin{aligned}
&V^\varepsilon(T,x)\,\partial_\varepsilon\rho^\varepsilon(T,x)\\
&=\partial_\varepsilon V^\varepsilon(0,x)\rho^\varepsilon(0,x)+\int_0^T\partial_\varepsilon\left[\partial_t\rho^\varepsilon(t,x)V^\varepsilon(t,x)+\rho^\varepsilon(t,x)\partial_t V^\varepsilon(t,x)\right]\mathrm{d}t\\
&=\int_0^T\left[\partial_\varepsilon\partial_t\rho^\varepsilon(t,x)V^\varepsilon(t,x)+\partial_\varepsilon\rho^\varepsilon(t,x)\partial_t V^\varepsilon(t,x)\right]\mathrm{d}t\\
&=\int_0^T\left[\partial_\varepsilon\partial_t\rho^\varepsilon(t,x)V^\varepsilon(t,x)+\partial_\varepsilon\rho^\varepsilon(t,x)G\left(t,x,u(t,x),-\nabla_x V^\varepsilon,-\nabla_x^2V^\varepsilon\right)\right]\mathrm{d}t
\end{aligned}\tag{5.29}$$

where we used (5.28) and the HJ equation (5.8) in the second and third equality respectively. Substitute (5.29) into (5.26), we obtain

$$\begin{aligned}
&\frac{\mathrm{d}}{\mathrm{d}\varepsilon}J[u+\varepsilon\phi]\\
&=\int_0^T\int_{\mathcal{X}}\left[\partial_\varepsilon\rho^\varepsilon(t,x)G\left(t,x,u(t,x),-\nabla_x V^\varepsilon,-\nabla_x^2V^\varepsilon\right)+\partial_\varepsilon\partial_t\rho^\varepsilon(t,x)V^\varepsilon(t,x)\right]\mathrm{d}x\,\mathrm{d}t\\
&\quad+\int_0^T\int_{\mathcal{X}}\left[\nabla_u r\left(x,u^\varepsilon(t,x)\right)\phi(t,x)\rho^\varepsilon(t,x)+r\left(x,u^\varepsilon(t,x)\right)\partial_\varepsilon\rho^\varepsilon(t,x)\right]\mathrm{d}x\,\mathrm{d}t
\end{aligned}$$

Taking derivative of the Fokker Planck equation (5.5) w.r.t. ε, we obtain

$$\begin{aligned}
&\partial_\varepsilon\partial_t\rho^\varepsilon(t,x)\\
&=-\nabla_x\cdot\left[\nabla_u b\left(x,u^\varepsilon(t,x)\right)^\top\phi(t,x)\rho^\varepsilon(t,x)+b\left(x,u^\varepsilon(t,x)\right)\partial_\varepsilon\rho^\varepsilon(t,x)\right]\\
&\quad+\sum_{i,j=1}^n\partial_i\partial_j\left[\nabla_u D_{ij}\left(x,u^\varepsilon(t,x)\right)^\top\phi(t,x)\rho^\varepsilon(t,x)+D_{ij}\left(x,u^\varepsilon(t,x)\right)\partial_\varepsilon\rho^\varepsilon(t,x)\right]
\end{aligned}$$

Therefore

$$\frac{\mathrm{d}}{\mathrm{d}\varepsilon}J[u+\varepsilon\phi] = \int_0^T \int_{\mathcal{X}} \left\{\partial_\varepsilon \rho^\varepsilon(t,x) G\left(t,x,u(t,x),-\nabla_x V^\varepsilon,-\nabla_x^2 V^\varepsilon\right)\right.$$

$$+ V^\varepsilon(t,x)\{-\nabla_x \cdot \left[\nabla_u b\left(x,u^\varepsilon(t,x)\right)\phi(t,x)\rho^\varepsilon(t,x) + b\left(x,u^\varepsilon(t,x)\right)\partial_\varepsilon\rho^\varepsilon(t,x)\right]$$

$$+ \sum_{i,j=1}^n \partial_i\partial_j\left[\nabla_u D_{ij}\left(x,u^\varepsilon(t,x)\right)\phi(t,x)\rho^\varepsilon(t,x) + D_{ij}\left(x,u^\varepsilon(t,x)\right)\partial_\varepsilon\rho^\varepsilon(t,x)\right]\}$$

$$\left. + \left[\nabla_u r\left(x,u^\varepsilon(t,x)\right)\phi(t,x)\rho^\varepsilon(t,x) + r\left(x,u^\varepsilon(t,x)\right)\partial_\varepsilon\rho^\varepsilon(t,x)\right]\right\}\mathrm{d}x\,\mathrm{d}t.$$

Applying integration by part in x, we get

$$\frac{\mathrm{d}}{\mathrm{d}\varepsilon}J[u+\varepsilon\phi] = \int_0^T \int_{\mathcal{X}} \left\{\partial_\varepsilon \rho^\varepsilon(t,x) G\left(t,x,u(t,x),-\nabla_x V^\varepsilon,-\nabla_x^2 V^\varepsilon\right)\right.$$

$$\left. + \nabla_x V^\varepsilon(t,x)^\top\left[\nabla_u b\left(x,u^\varepsilon(t,x)\right)\phi(t,x)\rho^\varepsilon(t,x) + b\left(x,u^\varepsilon(t,x)\right)\partial_\varepsilon\rho^\varepsilon(t,x)\right]\right.$$

$$+ \sum_{i,j=1}^n \partial_i\partial_j V^\varepsilon(t,x)\left[\nabla_u D_{ij}\left(x,u^\varepsilon(t,x)\right)\phi(t,x)\rho^\varepsilon(t,x) + D_{ij}\left(x,u^\varepsilon(t,x)\right)\partial_\varepsilon\rho^\varepsilon(t,x)\right]$$

$$\left. + \left[\nabla_u r\left(x,u^\varepsilon(t,x)\right)\phi(t,x)\rho^\varepsilon(t,x) + r\left(x,u^\varepsilon(t,x)\right)\partial_\varepsilon\rho^\varepsilon(t,x)\right]\right\}\mathrm{d}x\,\mathrm{d}t.$$

Making a rearrangement, we get

$$\frac{\mathrm{d}}{\mathrm{d}\varepsilon}J[u+\varepsilon\phi] = \int_0^T \int_{\mathcal{X}} \left\{\partial_\varepsilon \rho^\varepsilon(t,x)\Big[G\left(t,x,u(t,x),-\nabla_x V^\varepsilon,-\nabla_x^2 V^\varepsilon\right)\right.$$

$$\left. + \nabla_x V^\varepsilon(t,x)^\top b\left(x,u^\varepsilon(t,x)\right) + \sum_{i,j=1}^n \partial_i\partial_j V^\varepsilon(t,x) D_{ij}\left(x,u^\varepsilon(t,x)\right) + r\left(x,u^\varepsilon(t,x)\right)\Big]\right.$$

$$+ \rho^\varepsilon(t,x)\Big[\nabla_x V^\varepsilon(t,x)^\top \nabla_u b\left(x,u^\varepsilon(t,x)\right) + \sum_{i,j=1}^n \partial_i\partial_j V^\varepsilon(t,x)\ \nabla_u D_{ij}\left(x,u^\varepsilon(t,x)\right)$$

$$\left.\left. + \nabla_u r\left(x,u^\varepsilon(t,x)\right)\Big]\phi(t,x)\right\}\mathrm{d}x\,\mathrm{d}t$$

Therefore, by the definition of G (5.7),

$$\begin{aligned}
&\frac{\mathrm{d}}{\mathrm{d}\varepsilon} J[u+\varepsilon\phi] \\
&= \int_0^T \int_{\mathcal{X}} \left[\partial_\varepsilon \rho^\varepsilon(t,x)\cdot[0] - \rho^\varepsilon(t,x)\nabla_u G\left(t,x,u^\varepsilon(t,x),-\nabla_x V^\varepsilon,-\nabla_x^2 V^\varepsilon\right)\phi(t,x)\right]\mathrm{d}x\,\mathrm{d}t \\
&= -\int_0^T \int_{\mathcal{X}} \rho^\varepsilon(t,x)\nabla_u G\left(t,x,u^\varepsilon(t,x),-\nabla_x V^\varepsilon,-\nabla_x^2 V^\varepsilon\right)\phi(t,x)\,\mathrm{d}x\,\mathrm{d}t.
\end{aligned}$$

Let $\varepsilon = 0$, we get

$$\frac{\mathrm{d}}{\mathrm{d}\varepsilon} J[u+\varepsilon\phi]\bigg|_{\varepsilon=0} = -\int_0^T \int_{\mathcal{X}} \rho(t,x)\nabla_u G\left(t,x,u(t,x),-\nabla_x V_u,-\nabla_x^2 V_u\right)\phi(t,x)\,\mathrm{d}x\,\mathrm{d}t.$$

Therefore,

$$\frac{\delta J}{\delta u}(t,x) = -\rho(t,x)\ \nabla_u G\left(t,x,u(t,x),-\nabla_x V(t,x),-\nabla_x^2 V(t,x)\right).$$

i.e. (5.11) holds. The proof for (5.12) is almost the same. Firstly, changing the control function at $t < s$ will not affect $V^u(s,\cdot)$ by definition, so we just need to show (5.12) when $t \geq s$. We recall the definition of value function (5.3)

$$\begin{aligned}
V_u(s,y) &= \mathbb{E}\left[\int_s^T r(x_t,u_t)\,\mathrm{d}t + h(x_T)\ \middle|\ x_s = y\right] \\
&= \int_s^T \int_{\mathcal{X}} r(x,u(t,x))p^u(t,x;s,y)\,\mathrm{d}x\,\mathrm{d}t + \int_{\mathcal{X}} h(x)p^u(T,x;s,y)\,\mathrm{d}x.
\end{aligned}$$

Here, $p^u(t,x;s,y)$ is the fundamental solution of the Fokker Planck equation (5.5), so $p^u(t,x;s,y)$, as a function of (t,x), is the density of x_t starting at $x_s = y$. Therefore, we only need to repeat the argument to prove (5.11). The only caveat we need to be careful is that $p^\varepsilon(s,\cdot;s,y) = \delta_y$, so the classical derivative does not exist. This is not an essential difficulty because we can pick an arbitrary smooth probability density

function $\psi(y)$ on $\mathcal{X}$ and compute

$$\frac{\mathrm{d}}{\mathrm{d}\varepsilon}\int_{\mathcal{X}} V^{\varepsilon}(s,y)\psi(y)\,\mathrm{d}y\Big|_{\varepsilon=0}. \tag{5.30}$$

For example, when $s=0$ and $\psi\equiv 1$, (5.30) becomes

$$\frac{\mathrm{d}}{\mathrm{d}\varepsilon}\int_{\mathcal{X}} V^{\varepsilon}(0,y)\,\mathrm{d}y\Big|_{\varepsilon=0} = \frac{\mathrm{d}}{\mathrm{d}\varepsilon}J[u^{\varepsilon}]\Big|_{\varepsilon=0}.$$

Therefore, we can repeat the argument to prove (5.11) and get

$$\begin{aligned}&\frac{\mathrm{d}}{\mathrm{d}\varepsilon}\int_{\mathcal{X}} V^{\varepsilon}(s,y)\psi(y)\,\mathrm{d}y\Big|_{\varepsilon=0}\\ =&-\int_s^T\int_{\mathcal{X}} \rho^{u,s,\psi}(t,x)\nabla_u G\left(t,x,u(t,x),-\nabla_x V_u,-\nabla_x^2 V_u\right)\phi(t,x)\,\mathrm{d}x\,\mathrm{d}t.\end{aligned} \tag{5.31}$$

where $\rho^{u,s,\psi}(t,x) := \int_{\mathcal{X}} p^u(t,x;s,y)\psi(y)\,\mathrm{d}y$ is the solution to the Fokker Planck equation with initial condition $\rho^{u,s,\psi}(s,x)=\psi(x)$ and is also the density function of x_t, which starts with x_s, who follows a distribution of ψ. The only difference between proving (5.11) and (5.31) is that we need to replace $\int_0^T$ by $\int_s^T$, replace $\rho^{\varepsilon}(t,x)$ by $\rho^{\varepsilon,s,\psi}(t,x) := \rho^{u^{\varepsilon},s,\psi}(t,x)$, and replace $\rho^{\varepsilon}(0,x)$ by $\rho^{\varepsilon,s,\psi}(s,x)$. Therefore,

$$\begin{aligned}&\frac{\delta\left(\int_{\mathcal{X}} V_u(s,y)\psi(y)\,\mathrm{d}y\right)}{\delta u}\\ =&-\int_{\mathcal{X}} p^u(t,x;s,y)\psi(y)\,\mathrm{d}y\,\nabla_u G\left(t,x,u(t,x),-\nabla_x V_u,-\nabla_x^2 V_u\right).\end{aligned}$$

Hence, (5.12) holds. □

Remark. *In the proof of this proposition, we have assumed sufficient regularity such that all the derivatives exist in the classical sense. We believe that the theorem still holds with weaker assumptions, which can be proved using the spike variation technique. See [YZ99] section 3.4 Theorem 4.4 for example.*

Proof for Proposition 6. The Fokker Planck equation has been well-studied. Let $p^u(t,x;s,y)$ denote the fundamental solution to (5.5). Aronson found that the fundamental solution of a linear parabolic equation can be upper and lower bounded by fundamental solutions of heat equation (i.e. Gaussian functions) with different thermal diffusivity constant [Aro67]. For example, let $\widetilde{p}^u(t,x;s,y)$ be the fundamental solution of the Fokker Planck equation (5.5) in $\mathbb{R}^n$ (where b and σ are extended periodically), then

$$C^{-1}(t-s)^{-n/2}\exp\Big(-\frac{C\,|x-y|^2}{t-s}\Big) \le \widetilde{p}^u(t,x;s,y) \le C(t-s)^{-n/2}\exp\Big(-\frac{C^{-1}\,|x-y|^2}{t-s}\Big) \tag{5.32}$$

for all $s<t\le T$ and $x,y\in\mathbb{R}^n$, where C only depends on n, T, and K. We are in the unit torus $\mathcal{X}$ instead of $\mathbb{R}^n$, so

$$p^u(t,x;s,y)=\sum_{z\in\mathbb{Z}^n}\widetilde{p}^u(t,x+z;s,y),$$

where the x,y on the left is in $\mathcal{X}$, and the x,y on the right can be viewed as their embedding into $\mathbb{R}^n$. Our solution to the Fokker Planck equation, starting at uniform distribution $\rho(0,x)\equiv 1$, can be represented by

$$\rho^u(t,x)=\int_{\mathcal{X}}p^u(t,x;0,y)\,\mathrm{d}y=\int_{[0,1]^n}\sum_{z\in\mathbb{Z}^n}\widetilde{p}^u(t,x+z;s,y)\,\mathrm{d}y$$

$$=\int_{[0,1]^n}\sum_{z\in\mathbb{Z}^n}\widetilde{p}^u(t,x;s,y-z)\,\mathrm{d}y=\int_{\mathbb{R}^n}\widetilde{p}^u(t,x;s,y)\,\mathrm{d}y$$

Substituting the lower and upper bound (5.32), we obtain

$$\rho^u(t,x)\ge\int_{\mathbb{R}^n}C^{-1}t^{-n/2}\exp\left(-\frac{C\,|x-y|^2}{t}\right)\mathrm{d}y=:\rho_0$$

and

$$\rho^u(t,x)\le\int_{\mathbb{R}^n}Ct^{-n/2}\exp\left(-\frac{C^{-1}\,|x-y|^2}{t}\right)\mathrm{d}y=:\rho_1.$$

Here, the two integrals above is invariant w.r.t. t because of a simple change of variable. Therefore, we obtain a uniform lower bound ρ_0 and upper bound ρ_1 for $\rho^u(t,x)$, which depend only on T, n, and K. □

5.7 Some auxiliary lemmas

We state and prove some lemmas in this section.

Lemma 8. *[Stochastic Gronwall inequality] Under Assumption 2, there exists a positive constant C_1 s.t. for any two control functions $u_1, u_2 \in \mathcal{U}$, we have*

$$\sup_{t\in[0,T]} \mathbb{E}\left|x_t^1 - x_t^2\right|^2 \le C_1 \mathbb{E}\left|x_0^1 - x_0^2\right|^2 + C_1 \mathbb{E}\left[\int_0^T \left|u_1(t,x_t^1) - u_2(t,x_t^1)\right|^2 \mathrm{d}t\right], \quad (5.33)$$

where x_t^1 and x_t^2 are the state process (5.1) under controls u_1 and u_2 respectively. As a direct corollary, if $u_1 = u_2$, then

$$\sup_{t\in[0,T]} \mathbb{E}\left|x_t^1 - x_t^2\right|^2 \le C_1 \mathbb{E}\left|x_0^1 - x_0^2\right|^2. \quad (5.34)$$

Moreover, if $x_0^1 = x_0^2 \sim$ Unif($\mathcal{X}$), then

$$\sup_{t\in[0,T]} \mathbb{E}\left|x_t^1 - x_t^2\right|^2 \le C_1 \left\|u_1 - u_2\right\|_{L^2}^2. \quad (5.35)$$

Proof. We denote $b_t^i = b(x_t^i, u_i(t,x_t^i))$, $\sigma_t^i = \sigma(x_t^i, u_i(t,x_t^i))$ for $i = 1,2$, so $\mathrm{d}x_t^i = b_t^i \mathrm{d}t + \sigma_t^i \mathrm{d}W_t$. By Itô's lemma,

$$\mathrm{d}\left|x_t^1 - x_t^2\right|^2 = \left[\left|\sigma_t^1 - \sigma_t^2\right|^2 + 2\left\langle x_t^1 - x_t^2,\, b_t^1 - b_t^2\right\rangle\right]\mathrm{d}t + 2(x_t^1 - x_t^2)^\top(\sigma_t^1 - \sigma_t^2)\mathrm{d}W_t.$$

Integrate and take expactation, we obtain

$$\mathbb{E}\left|x_T^1 - x_T^2\right|^2 = \mathbb{E}\left|x_0^1 - x_0^2\right|^2 + \mathbb{E}\int_0^T \left[\left|\sigma_t^1 - \sigma_t^2\right|^2 + 2\left\langle x_t^1 - x_t^2,\, b_t^1 - b_t^2\right\rangle\right]\mathrm{d}t. \quad (5.36)$$

By the Lipschitz condition in Assumption 2,

$$\begin{aligned}\left|b_t^1 - b_t^2\right| &\leq L\left|x_t^1 - x_t^2\right| + L\left|u_1(t, x_t^1) - u_2(t, x_t^2)\right| \\ &\leq (L + L^2)\left|x_t^1 - x_t^2\right| + L\left|u_1(t, x_t^1) - u_2(t, x_t^1)\right|.\end{aligned}$$

So,

$$\left|b_t^1 - b_t^2\right|^2 \leq 2(L + L^2)^2\left|x_t^1 - x_t^2\right|^2 + 2L^2\left|u_1(t, x_t^1) - u_2(t, x_t^1)\right|^2. \tag{5.37}$$

Similarly,

$$\left|\sigma_t^1 - \sigma_t^2\right|^2 \leq 2(L + L^2)^2\left|x_t^1 - x_t^2\right|^2 + 2L^2\left|u_1(t, x_t^1) - u_2(t, x_t^1)\right|^2. \tag{5.38}$$

Applying Cauchy's inequality, and substituting (5.37) and (5.38) into (5.36), we obtain

$$\begin{aligned}\mathbb{E}\left|x_T^1 - x_T^2\right|^2 &\leq \mathbb{E}\left|x_0^1 - x_0^2\right|^2 + \mathbb{E}\int_0^T \left[\left|\sigma_t^1 - \sigma_t^2\right|^2 + \left|x_t^1 - x_t^2\right|^2 + \left|b_t^1 - b_t^2\right|^2\right] \mathrm{d}t \\ &\leq \mathbb{E}\left|x_0^1 - x_0^2\right|^2 + \mathbb{E}\int_0^T \left[17L^4\left|x_t^1 - x_t^2\right|^2 + 4L^2\left|u_1(t, x_t^1) - u_2(t, x_t^1)\right|^2\right] \mathrm{d}t\end{aligned} \tag{5.39}$$

Note that (5.39) still holds if we replace T by some $T' < T$, so we can apply Gronwall's inequality and obtain

$$\mathbb{E}\left|x_T^1 - x_T^2\right|^2 \leq e^{17L^4T}\mathbb{E}\left|x_0^1 - x_0^2\right|^2 + 4L^2e^{17L^4T}\mathbb{E}\left[\int_0^T \left|u_1(t, x_t^1) - u_2(t, x_t^1)\right|^2 \mathrm{d}t\right]. \tag{5.40}$$

Again, (5.40) still holds if we replace T by some $T' < T$, so (5.33) holds. Moreover, if $x_0^1 = x_0^2 \sim \mathrm{Unif}(\mathcal{X})$, then by Proposition 6,

$$\begin{aligned}&\mathbb{E}\int_0^T \left|u_1(t, x_t^1) - u_2(t, x_t^1)\right|^2 \mathrm{d}t \\ &= \int_{\mathcal{X}}\int_0^T \rho^1(t, x)\left|u_1(t, x) - u_2(t, x)\right|^2 \mathrm{d}t\mathrm{d}x \leq \rho_1\left\|u_1 - u_2\right\|_{L^2}^2.\end{aligned}$$

Therefore, (5.35) holds. □

Lemma 9. *Under Assumption 2, there exists a positive constant C_2 s.t. for any two control functions $u_1, u_2 \in \mathcal{U}$, we have*

$$\|V_{u_1} - V_{u_2}\|_{(T;H^2)} \leq C_2 \|u_1 - u_2\|_{L^2} . \tag{5.41}$$

Proof. We firstly give some notations. Following the notations in the previous lemma, x_t^1 and x_t^2 are the state process w.r.t. controls u_1 and u_2, starting at $x_0^1 = x_0^2 \sim \text{Unif}(\mathcal{X})$. For $i = 1, 2$, we have $b_t^i = b(x_t^i, u_i(t, x_t^i))$, $\sigma_t^i = \sigma(x_t^i, u_i(t, x_t^i))$, and $r_t^i = r(x_t^i, u_i(t, x_t^i))$. We also define the following gradient processes

$$\nabla_x b_t^i = \nabla_x b(x_t^i, u_i(t, x_t^i)),$$

$$\nabla_x \sigma_t^i = \nabla_x \sigma(x_t^i, u_i(t, x_t^i)),$$

and

$$\nabla_x r_t^i = \nabla_x r(x_t^i, u_i(t, x_t^i)).$$

Here please note that ∇_x only operate on the first argument in b, σ, and r. By Assumption 2, for $f = b, \sigma, r, \nabla_x b, \nabla_x \sigma, \nabla_x r$, we have

$$\begin{aligned}
&\left|f_t^1 - f_t^2\right| = \left|f(x_t^1, u_1(t, x_t^1)) - f(x_t^2, u_2(t, x_t^2))\right| \\
\leq\ & L\left|x_t^1 - x_t^2\right| + L\left|u_1(t, x_t^1) - u_2(t, x_t^2)\right| \\
\leq\ & (L + L^2)\left|x_t^1 - x_t^2\right| + L\left|u_1(t, x_t^1) - u_2(t, x_t^1)\right|
\end{aligned}$$

and hence

$$\left|f_t^1 - f_t^2\right|^2 \leq 2(L + L^2)^2 \left|x_t^1 - x_t^2\right|^2 + 2L^2 \left|u_1(t, x_t^1) - u_2(t, x_t^1)\right|^2 .$$

If we make an integration and apply Lemma 8, we obtain

$$\mathbb{E}\int_0^T \left|f_t^1 - f_t^2\right|^2 \mathrm{d}t \leq C\mathbb{E}\left[\int_0^T \left|u_1(t, x_t^1) - u_2(t, x_t^1)\right|^2 \mathrm{d}t\right] \leq C\|u_1 - u_2\|_{L^2}^2 . \tag{5.42}$$

Next, we will show (5.41) step by step.

Step 1. We want to show

$$\|V_{u_1} - V_{u_2}\|_{L^2} \leq C \|u_1 - u_2\|_{L^2} . \tag{5.43}$$

Applying Itô's lemma on $V_{u_i}(t, x_t^i)$, we obtain

$$h(x_T^i) = V_{u_i}(0, x_0) + \int_0^T \left(\partial_t V_{u_i}(t, x_t^i) + \mathcal{G}_{u_i} V_{u_i}(t, x_t^i)\right) \mathrm{d}t + \int_0^T \nabla_x V_{u_i}(t, x_t^i)^\top \sigma_t^i \mathrm{d}W_t,$$

where $\mathcal{G}_{u_i}$ is the infinitesimal generator of the SDE (5.1) under control u_i. Applying the HJ equation (5.8) in the drift term and rearranging the terms, we get

$$V_{u_i}(0, x_0) = h(x_T^i) + \int_0^T r_t^i \mathrm{d}t - \int_0^T \nabla_x V_{u_i}(t, x_t^i)^\top \sigma_t^i \mathrm{d}W_t. \tag{5.44}$$

So,

$$V_{u_1}(0, x_0) - V_{u_2}(0, x_0) = \mathbb{E}\left[h(x_T^1) - h(x_T^2) + \int_0^T (r_t^1 - r_t^2)\mathrm{d}t \;\middle|\; x_0\right]. \tag{5.45}$$

Therefore,

$$\begin{aligned}
&\int_{\mathcal{X}} |V_{u_1}(0, x) - V_{u_2}(0, x)|^2 \, \mathrm{d}x = \mathbb{E}\left[(V_{u_1}(0, x_0) - V_{u_2}(0, x_0))^2\right] \\
&= \mathbb{E}\left[\left(\mathbb{E}\left[h(x_T^1) - h(x_T^2) + \int_0^T (r_t^1 - r_t^2)\mathrm{d}t \;\middle|\; x_0\right]\right)^2\right] \\
&\leq \mathbb{E}\left[\left(\mathbb{E}\left[L\left|x_T^1 - x_T^2\right| + \int_0^T \left|r_t^1 - r_t^2\right| \mathrm{d}t \;\middle|\; x_0\right]\right)^2\right] \\
&\leq \mathbb{E}\left[\left(L\left|x_T^1 - x_T^2\right| + \int_0^T \left|r_t^1 - r_t^2\right| \mathrm{d}t\right)^2\right] \\
&\leq \mathbb{E}\left[2L^2\left|x_T^1 - x_T^2\right|^2 + 2T\int_0^T \left|r_t^1 - r_t^2\right|^2 \mathrm{d}t\right] \leq C \|u_1 - u_2\|_{L^2}^2 ,
\end{aligned}$$

where we have consecutively used: $x_0 \sim \text{Unif}(\mathcal{X})$; equation (5.45); Lipschitz condition of h in Assumption 2; Jensen's inequality and tower property; Cauchy's inequality; Lemma 8 and (5.42). Therefore, we have shown

$$\|V_{u_1}(0,\cdot) - V_{u_2}(0,\cdot)\|_{L^2}^2 \le C \|u_1 - u_2\|_{L^2}^2 \tag{5.46}$$

where this constant C only depends on K, n, T. Also, (5.46) holds with the same C if the total time span T decreases. Therefore, we can reformulate the control problem such that it start at $t \in (0,T)$ instead of 0. Then the new state process starts at $x_t^i \sim \text{Unif}(\mathcal{X})$ and the new value function coincide with V_{u_i} on $[t,T]$ by definition (5.3). We also remark that the constants ρ_0, ρ_1 in Proposition 6 remain the same because T decreases. Applying the argument for (5.46) on the new control problem gives us

$$\|V_{u_1}(t,\cdot) - V_{u_2}(t,\cdot)\|_{L^2}^2 \le C \int_t^T \|u_1(s,\cdot) - u_2(s,\cdot)\|_{L^2}^2 \,\mathrm{d}s \le C \|u_1 - u_2\|_{L^2}^2 .$$

Making an integration in t gives us (5.43).

Step 2. We want to show

$$\|\nabla_x V_{u_1} - \nabla_x V_{u_2}\|_{L^2} \le C \|u_1 - u_2\|_{L^2} . \tag{5.47}$$

We recall that the first order adjoint equation is given by (5.6). We denote $p_t^i = -\nabla_x V_{u_i}(t, x_t^i)$ and $q_t^i = -\nabla_x^2 V_{u_i}(t, x_t^i)\sigma_t^i$ for $i = 1, 2$. They satisfy the equations

$$\begin{cases} \mathrm{d}p_t^i = -\left[(\nabla_x b_t^i)^\top p_t^i + \nabla_x \,\mathrm{Tr}\left((\sigma_t^i)^\top q_t^i\right) - \nabla_x r_t^i\right]\mathrm{d}t + q_t^i \,\mathrm{d}W_t \\ p_T^i = -\nabla_x h(x_T^i). \end{cases} \tag{5.48}$$

By assumption 2, we have $|p_t^i| \le K$ and $|q_t^i| \le K^2$. By (5.44),

$$\int_0^T \left((p_t^1)^\top \sigma_t^1 - (p_t^2)^\top \sigma_t^2\right) \mathrm{d}W_t$$

$$= h(x_T^1) - h(x_T^2) + \int_0^T (r_t^1 - r_t^2)\mathrm{d}t - \left(V_{u_1}(0,x_0) - V_{u_2}(0,x_0)\right).$$

Taking a square expectation and using a Cauchy inequality, we obtain

$$\begin{aligned}
&\mathbb{E}\int_0^T \left|(p_t^1)^\top \sigma_t^1 - (p_t^2)^\top \sigma_t^2\right|^2 \mathrm{d}t \\
&\leq 3\mathbb{E}\left[\left|h(x_T^1) - h(x_T^2)\right|^2 + T\int_0^T \left|r_t^1 - r_t^2\right|^2 \mathrm{d}t + \left|V_{u_1}(0,x_0) - V_{u_2}(0,x_0)\right|^2\right] \\
&\leq 3L^2\mathbb{E}\left|x_T^1 - x_T^2\right|^2 + 3T\,\mathbb{E}\int_0^T \left|r_t^1 - r_t^2\right|^2 \mathrm{d}t + 3\left\|V_{u_1}(0,\cdot) - V_{u_2}(0,\cdot)\right\|_{L^2}^2 \\
&\leq C\left\|u_1 - u_2\right\|_{L^2}^2,
\end{aligned} \tag{5.49}$$

where the last inequality is because of the Gronwall inequality, estimate (5.42), and the arguments in *Step 1.* Also note that

$$(p_t^1)^\top \sigma_t^1 - (p_t^2)^\top \sigma_t^2 = (p_t^1)^\top \sigma_t^1 - (p_t^2)^\top \sigma_t^1 + (p_t^2)^\top \sigma_t^1 - (p_t^2)^\top \sigma_t^2,$$

so

$$\left|(p_t^1)^\top \sigma_t^1 - (p_t^2)^\top \sigma_t^1\right|^2 \leq 2\left|(p_t^1)^\top \sigma_t^1 - (p_t^2)^\top \sigma_t^2\right|^2 + 2K^2\left|\sigma_t^1 - \sigma_t^2\right|^2. \tag{5.50}$$

Therefore, taking an integration and expectation, we obtain

$$\mathbb{E}\int_0^T \left|p_t^1 - p_t^2\right|^2 \mathrm{d}t \leq \frac{1}{2\sigma_0}\mathbb{E}\int_0^T \left|(p_t^1)^\top \sigma_t^1 - (p_t^2)^\top \sigma_t^1\right|^2 \mathrm{d}t \leq C\left\|u_1 - u_2\right\|_{L^2}^2, \tag{5.51}$$

where the first inequality is due to the uniform ellipticity assumption and the second is because of (5.50), (5.49), and (5.42). Next, since

$$p_t^1 - p_t^2 = \nabla_x V_{u_1}(t, x_t^1) - \nabla_x V_{u_2}(t, x_t^1) + \nabla_x V_{u_2}(t, x_t^1) - \nabla_x V_{u_2}(t, x_t^2),$$

we have

$$\begin{aligned}
&\left|\nabla_x V_{u_1}(t, x_t^1) - \nabla_x V_{u_2}(t, x_t^1)\right|^2 \\
&\leq 2\left|p_t^1 - p_t^2\right|^2 + 2\left|\nabla_x V_{u_2}(t, x_t^1) - \nabla_x V_{u_2}(t, x_t^2)\right|^2 \\
&\leq 2\left|p_t^1 - p_t^2\right|^2 + 2L^2\left|x_t^1 - x_t^2\right|^2.
\end{aligned} \tag{5.52}$$

Therefore,

$$\|\nabla_x V_{u_1} - \nabla_x V_{u_2}\|_{L^2}^2 \leq \frac{1}{\rho_0}\mathbb{E}\int_0^T \left|\nabla_x V_{u_1}(t, x_t^1) - \nabla_x V_{u_2}(t, x_t^1)\right|^2 \mathrm{d}t$$

$$\leq C\mathbb{E}\int_0^T \left(\left|p_t^1 - p_t^2\right|^2 + \left|x_t^1 - x_t^2\right|^2\right)\mathrm{d}t \leq C\left\|u_1 - u_2\right\|_{L^2}^2,$$

where we have consecutively used: Proposition 6; equation (5.52); equation (5.51) and Lemma 8. Therefore (5.47) holds.

Step 3. We want to show

$$\mathbb{E}\int_0^T \left|q_t^1 - q_t^2\right|^2 \mathrm{d}t \leq C\left\|u_1 - u_2\right\|_{L^2}^2. \tag{5.53}$$

The analysis for the second order derivative is the most difficult. We need to cut the interval $[0, T]$ into small pieces and estimate them separately. Let N be an integer s.t. $\delta_0 := T/N \leq 1/40K^2$. We denote $t_k = k\delta_0$ as time stamps. We will do the estimate for each interval $[t_{k-1}, t_k]$ for $k = 1, 2, \ldots, N$. Computing the difference of the two adjoint equations (5.48) gives us

$$\begin{aligned}(q_t^1 - q_t^2)\mathrm{d}W_t = \mathrm{d}(p_t^1 - p_t^2) - \left(\nabla_x r_t^1 - \nabla_x r_t^2\right)\mathrm{d}t + \left[(\nabla_x b_t^1)^\top p_t^1 - (\nabla_x b_t^2)^\top p_t^2\right]\mathrm{d}t \\ + \left[\nabla_x \operatorname{Tr}\left((\sigma_t^1)^\top q_t^1\right) - \nabla_x \operatorname{Tr}\left((\sigma_t^2)^\top q_t^2\right)\right]\mathrm{d}t.\end{aligned} \tag{5.54}$$

Step 3.1. We consider $k = N$ first. Let $\delta \in [\delta_0, 2\delta_0]$. If we integrate (5.54) on $[T - \delta, T]$ we obtain

$$\begin{aligned}\int_{T-\delta}^T \left(q_t^1 - q_t^2\right)\mathrm{d}W_t = &-\left(\nabla_x h(x_T^1) - \nabla_x h(x_T^2)\right) + \left(p_{T-\delta}^1 - p_{T-\delta}^2\right) \\ &- \int_{T-\delta}^T \left(\nabla_x r_t^1 - \nabla_x r_t^2\right)\mathrm{d}t + \int_{T-\delta}^T \left[(\nabla_x b_t^1)^\top p_t^1 - (\nabla_x b_t^2)^\top p_t^2\right]\mathrm{d}t \\ &+ \int_{T-\delta}^T \left[\nabla_x \operatorname{Tr}\left((\sigma_t^1)^\top q_t^1\right) - \nabla_x \operatorname{Tr}\left((\sigma_t^2)^\top q_t^2\right)\right]\mathrm{d}t.\end{aligned}$$

We take a square expectation, apply Cauchy's inequalities, and get

$$\begin{aligned}
&\mathbb{E}\int_{T-\delta}^{T}\left|q_t^1-q_t^2\right|^2\mathrm{d}t \le 5\mathbb{E}\left[\left|\nabla_x h(x_T^1)-\nabla_x h(x_T^2)\right|^2+\left|p_{T-\delta}^1-p_{T-\delta}^2\right|^2\right.\\
&\quad+\delta\int_{T-\delta}^{T}\left|\nabla_x r_t^1-\nabla_x r_t^2\right|^2\mathrm{d}t+\delta\int_{T-\delta}^{T}\left|(\nabla_x b_t^1)^\top p_t^1-(\nabla_x b_t^2)^\top p_t^2\right|^2\mathrm{d}t\\
&\quad\left.+\delta\int_{T-\delta}^{T}\left|\nabla_x \operatorname{Tr}\left((\sigma_t^1)^\top q_t^1\right)-\nabla_x \operatorname{Tr}\left((\sigma_t^2)^\top q_t^2\right)\right|^2\mathrm{d}t\right]\\
&=: (I)+(II)+(III)+(IV)+(V).
\end{aligned}\tag{5.55}$$

Next, we bound these terms one by one. For (I), we have

$$(I)\le 5L^2\mathbb{E}\left|x_T^1-x_T^2\right|^2\le 5L^2C_1\left\|u_1-u_2\right\|_{L^2}^2,\tag{5.56}$$

where we used the Lipschitz condition in Assumption 2 and Lemma 8. We skip (II). For (III), we have

$$(III)\le C\left\|u_1-u_2\right\|_{L^2}^2\tag{5.57}$$

because of (5.42). For (IV), we have

$$\begin{aligned}
&(IV)=5\delta\,\mathbb{E}\int_{T-\delta}^{T}\left|(\nabla_x b_t^1)^\top p_t^1-(\nabla_x b_t^2)^\top p_t^2\right|^2\mathrm{d}t\\
&\le 10\delta\,\mathbb{E}\int_{T-\delta}^{T}\left(\left|(\nabla_x b_t^1)^\top p_t^1-(\nabla_x b_t^1)^\top p_t^2\right|^2+\left|(\nabla_x b_t^1)^\top p_t^2-(\nabla_x b_t^2)^\top p_t^2\right|^2\right)\mathrm{d}t\\
&\le 10\delta\,\mathbb{E}\int_{T-\delta}^{T}K^2\left(\left|p_t^1-p_t^2\right|^2+\left|\nabla_x b_t^1-\nabla_x b_t^2\right|^2\right)\mathrm{d}t\le C\left\|u_1-u_2\right\|_{L^2}^2,
\end{aligned}\tag{5.58}$$

where the second inequality is because of the boundedness of $\nabla_x b$ and $\nabla_x V_{u_i}$ and the

third is because of (5.51) and (5.42). For (V), we have

$$
\begin{aligned}
(V) &= 5\delta\,\mathbb{E}\int_{T-\delta}^{T}\left|\nabla_x \operatorname{Tr}\left((\sigma_t^1)^\top q_t^1\right) - \nabla_x \operatorname{Tr}\left((\sigma_t^2)^\top q_t^2\right)\right|^2 \mathrm{d}t \\
&\le 10\delta\,\mathbb{E}\int_{T-\delta}^{T}\Big(\left|\nabla_x \operatorname{Tr}\left((\sigma_t^1)^\top q_t^1\right) - \nabla_x \operatorname{Tr}\left((\sigma_t^1)^\top q_t^2\right)\right|^2 \\
&\qquad + \left|\nabla_x \operatorname{Tr}\left((\sigma_t^1)^\top q_t^2\right) - \nabla_x \operatorname{Tr}\left((\sigma_t^2)^\top q_t^2\right)\right|^2\Big)\,\mathrm{d}t \qquad (5.59) \\
&\le 10\delta\,\mathbb{E}\int_{T-\delta}^{T}\left(K^2\left|q_t^1 - q_t^2\right|^2 + K^4\left|\sigma_t^1 - \sigma_t^2\right|\right)\mathrm{d}t \\
&\le \frac{1}{2}\mathbb{E}\int_{T-\delta}^{T}\left|q_t^1 - q_t^2\right|^2 \mathrm{d}t + C\left\|u_1 - u_2\right\|_{L^2}^2,
\end{aligned}
$$

where the second inequality is because boundedness of $\nabla_x\sigma$ and $\nabla_x^2 V_{u_i}$ and the third is because $\delta \le 2\delta_0 \le 1/20K^2$ and (5.42). Substituting (5.56), (5.57), (5.58), and (5.59) into (5.55), we obtain

$$
\frac{1}{2}\mathbb{E}\int_{T-\delta}^{T}\left|q_t^1 - q_t^2\right|^2 \mathrm{d}t \le 5\delta\mathbb{E}\left|p_{T-\delta}^1 - p_{T-\delta}^2\right|^2 + C\left\|u_1 - u_2\right\|_{L^2}^2. \qquad (5.60)
$$

Integrating (5.60) w.r.t. δ on $[\delta_0, 2\delta_0]$, we obtain

$$
\begin{aligned}
&\frac{1}{2}\delta_0\mathbb{E}\int_{T-\delta_0}^{T}\left|q_t^1 - q_t^2\right|^2 \mathrm{d}t \le \int_{\delta_0}^{2\delta_0} (5.60)\mathrm{LHS}\,\mathrm{d}\delta \le \int_{\delta_0}^{2\delta_0} (5.60)\mathrm{RHS}\,\mathrm{d}\delta \\
&\le C\mathbb{E}\int_{T-2\delta_0}^{T-\delta_0}\left|p_t^1 - p_t^2\right|^2 \mathrm{d}t + C\left\|u_1 - u_2\right\|_{L^2}^2 \le C\left\|u_1 - u_2\right\|_{L^2}^2,
\end{aligned}
$$

where the last inequality is due to (5.51). Therefore, we have

$$
\mathbb{E}\int_{T-\delta_0}^{T}\left|q_t^1 - q_t^2\right|^2 \mathrm{d}t \le C\left\|u_1 - u_2\right\|_{L^2}^2. \qquad (5.61)
$$

Step 3.2. We consider $k = 2, 3, \ldots, N-1$ next. Let $\delta \in [0, \delta_0]$. We integrate

(5.54) on $[t_{k-1}-\delta, t_{k+1}-\delta]$ and obtain

$$\begin{aligned}\int_{t_{k-1}-\delta}^{t_{k+1}-\delta}\left(q_t^1-q_t^2\right)\mathrm{d}W_t &= \left(p^1_{t_{k+1}-\delta}-p^2_{t_{k+1}-\delta}\right)-\left(p^1_{t_{k-1}-\delta}-p^2_{t_{k-1}-\delta}\right)\\ &-\int_{t_{k-1}-\delta}^{t_{k+1}-\delta}\left(\nabla_x r_t^1-\nabla_x r_t^2\right)\mathrm{d}t+\int_{t_{k-1}-\delta}^{t_{k+1}-\delta}\left[(\nabla_x b_t^1)^\top p_t^1-(\nabla_x b_t^2)^\top p_t^2\right]\mathrm{d}t\\ &+\int_{t_{k-1}-\delta}^{t_{k+1}-\delta}\left[\nabla_x \operatorname{Tr}\left((\sigma_t^1)^\top q_t^1\right)-\nabla_x \operatorname{Tr}\left((\sigma_t^2)^\top q_t^2\right)\right]\mathrm{d}t.\end{aligned}$$

We take a square expectation, apply Cauchy's inequalities, and get

$$\begin{aligned}&\mathbb{E}\int_{t_{k-1}-\delta}^{t_{k+1}-\delta}\left|q_t^1-q_t^2\right|^2\mathrm{d}t \le 5\mathbb{E}\left[\left|p^1_{t_{k+1}-\delta}-p^2_{t_{k+1}-\delta}\right|^2+\left|p^1_{t_{k-1}-\delta}-p^2_{t_{k-1}-\delta}\right|^2\right.\\ &+2\delta_0\int_{t_{k-1}-\delta}^{t_{k+1}-\delta}\left|\nabla_x r_t^1-\nabla_x r_t^2\right|^2\mathrm{d}t+2\delta_0\int_{t_{k-1}-\delta}^{t_{k+1}-\delta}\left|(\nabla_x b_t^1)^\top p_t^1-(\nabla_x b_t^2)^\top p_t^2\right|^2\mathrm{d}t\\ &\left.+2\delta_0\int_{t_{k-1}-\delta}^{t_{k+1}-\delta}\left|\nabla_x \operatorname{Tr}\left((\sigma_t^1)^\top q_t^1\right)-\nabla_x \operatorname{Tr}\left((\sigma_t^2)^\top q_t^2\right)\right|^2\mathrm{d}t\right]\\ &=: (I)+(II)+(III)+(IV)+(V)\end{aligned}\tag{5.62}$$

with a little abuse of notation for the five terms in (5.55). We bound these five terms next. The techniques are exactly the same as in *Step3.1.* We keep (I) and (II) unchanged. (III) also satisfies (5.57) with the same reason. (IV) also satisfies (5.58), where we only need to modify the interval for integration in the intermediate steps. For (V), using the same argument in (5.58), we obtain

$$(V)\le\frac{1}{2}\mathbb{E}\int_{t_{k-1}-\delta}^{t_{k+1}-\delta}\left|q_t^1-q_t^2\right|^2\mathrm{d}t+C\left\|u_1-u_2\right\|_{L^2}^2.$$

Combining the estimates into (5.62), we obtain

$$\begin{aligned}&\frac{1}{2}\mathbb{E}\int_{t_{k-1}-\delta}^{t_{k+1}-\delta}\left|q_t^1-q_t^2\right|^2\mathrm{d}t\\&\leq 5\delta\mathbb{E}\left[\left|p^1_{t_{k+1}-\delta}-p^2_{t_{k+1}-\delta}\right|^2+\left|p^1_{t_{k-1}-\delta}-p^2_{t_{k-1}-\delta}\right|^2\right]+C\left\|u_1-u_2\right\|^2_{L^2}.\end{aligned}\tag{5.63}$$

Integrating (5.63) w.r.t. δ on $[0,\delta_0]$, we obtain

$$\begin{aligned}&\frac{1}{2}\delta_0\mathbb{E}\int_{t_{k-1}}^{t_k}\left|q_t^1-q_t^2\right|^2\mathrm{d}t\leq\int_0^{\delta_0}(5.63)\mathrm{LHS}\,\mathrm{d}\delta\leq\int_0^{\delta_0}(5.63)\mathrm{RHS}\,\mathrm{d}\delta\\&\leq C\mathbb{E}\left(\int_{t_{k-2}}^{t_{k-1}}+\int_{t_k}^{t_{k+1}}\right)\left|p_t^1-p_t^2\right|^2\mathrm{d}t+C\left\|u_1-u_2\right\|^2_{L^2}\leq C\left\|u_1-u_2\right\|^2_{L^2},\end{aligned}$$

where the last inequality is due to (5.51). Therefore, for $k=2,3,\ldots,N-1$, we have

$$\mathbb{E}\int_{t_{k-1}}^{t_k}\left|q_t^1-q_t^2\right|^2\mathrm{d}t\leq C\left\|u_1-u_2\right\|^2_{L^2}.\tag{5.64}$$

Step 3.3. We consider $k=1$ next. In this case we cannot integrate (5.54) on $[t_{k-1}-\delta,t_{k+1}-\delta]$ because t should be non-negative. But we can repeat the argument in *Step 3.2*, with only a slight modification of our model. We extend the value function $V(t,x)$ to $t\in[-\delta_0,0)$ by considering a modification of the control problem starting at $-\delta_0$ instead of time 0.

We give detailed description of this extension to confirm that it works. Let us use a "hat" notation to denote the quantities for the new control problem. Firstly, the control functions $u_i(t,x)$ need to be extended to $[-\delta_0,T]\times\mathcal{X}$ such that $\widehat{u}_i(t,x)=u_i(t,x)$ on $[0,T]\times\mathcal{X}$. By definition (5.3), the new value functions $\widehat{V}_{u_i}:[-\delta_0,T]\times\mathcal{X}\to\mathbb{R}$ coincide with V_{u_i} on $[0,T]\times\mathcal{X}$. We also require the extension of u_i to be smooth such that the bounds for control functions in Assumption 3 still hold and

$$\left\|\widehat{u}_1-\widehat{u}_2\right\|^2_{L^2}\leq 2\left\|u_1-u_2\right\|^2_{L^2}.$$

The bounds for the value function (obtained from Schauder estimate) should still hold. The new state process start at $\widehat{x}^i_{-\delta_0} \sim \text{Unif}(\mathcal{X})$. The bounds ρ_0, ρ_1 in Proposition 6 may need to change because the total time span is increased from T to $T+\delta_0$, but they are still absolute constants if we follow the proof for Proposition 6. The Gronwall inequality should also hold, but with a larger constant C_1. Inequality (5.42) should still hold, with interval to be integrated replaced by $[-\delta_0, T]$.

With these clarifications, we can repeat the arguments in *Step 3.2* and obtain

$$\mathbb{E}\int_0^{\delta_0} \left|\widehat{q}^1_t - \widehat{q}^2_t\right|^2 \mathrm{d}t \le C \left\|\widehat{u}_1 - \widehat{u}_2\right\|^2_{L^2} \le 2C \left\|u_1 - u_2\right\|^2_{L^2}. \tag{5.65}$$

Therefore, we did not perfectly recover (5.53), but the results (5.61), (5.64), and (5.65) are enough for us to proceed the next step. We will finish proving (5.53) later.

Step 4. We want to show

$$\left\|\nabla^2_x V_{u_1} - \nabla^2_x V_{u_2}\right\|_{L^2} \le C \left\|u_1 - u_2\right\|_{L^2}. \tag{5.66}$$

Since

$$\begin{aligned} q^1_t - q^2_t = &-\nabla^2_x V_{u_1}(t, x^1_t)\sigma^1_t + \nabla^2_x V_{u_2}(t, x^2_t)\sigma^2_t = -\nabla^2_x V_{u_1}(t, x^1_t)\sigma^1_t + \nabla^2_x V_{u_2}(t, x^1_t)\sigma^1_t \\ &- \nabla^2_x V_{u_2}(t, x^1_t)\sigma^1_t + \nabla^2_x V_{u_2}(t, x^2_t)\sigma^1_t - \nabla^2_x V_{u_2}(t, x^2_t)\sigma^1_t + \nabla^2_x V_{u_2}(t, x^2_t)\sigma^2_t, \end{aligned} \tag{5.67}$$

we have

$$\begin{aligned} &\left|\nabla^2_x V_{u_1}(t, x^1_t)\sigma^1_t - \nabla^2_x V_{u_2}(t, x^1_t)\sigma^1_t\right|^2 \le 3\left|-\nabla^2_x V_{u_2}(t, x^1_t)\sigma^1_t - \nabla^2_x V_{u_2}(t, x^2_t)\sigma^1_t\right|^2 \\ &\qquad 3\left|\nabla^2_x V_{u_2}(t, x^2_t)\sigma^1_t - \nabla^2_x V_{u_2}(t, x^2_t)\sigma^2_t\right|^2 + 3\left|q^1_t - q^2_t\right|^2 \\ &\le 3K^2L^2\left|x^1_t - x^2_t\right|^2 + 3K^2\left|\sigma^1_t - \sigma^2_t\right|^2 + 3\left|q^1_t - q^2_t\right|^2 \end{aligned} \tag{5.68}$$

Therefore,

$$\int_{\delta_0}^{T} \left\|\nabla_x^2 V_{u_1}(t,\cdot) - \nabla_x^2 V_{u_2}(t,\cdot)\right\|_{L^2}^2 \mathrm{d}t \leq \frac{1}{\rho_0}\mathbb{E}\int_{\delta_0}^{T} \left|\nabla_x^2 V_{u_1}(t,x_t^1) - \nabla_x^2 V_{u_2}(t,x_t^1)\right|^2 \mathrm{d}t$$

$$\leq \frac{1}{2\sigma_0\rho_0}\mathbb{E}\int_{\delta_0}^{T} \left|\nabla_x^2 V_{u_1}(t,x_t^1)\sigma_t^1 - \nabla_x^2 V_{u_2}(t,x_t^1)\sigma_t^1\right|^2 \mathrm{d}t$$

$$\leq C\mathbb{E}\int_{\delta_0}^{T} \left(\left|x_t^1 - x_t^2\right|^2 + \left|\sigma_t^1 - \sigma_t^2\right|^2 + \left|q_t^1 - q_t^2\right|^2\right) \mathrm{d}t \leq C\left\|u_1 - u_2\right\|_{L^2}^2 . \tag{5.69}$$

where we have consecutively used: Proposition 6; uniform ellipticity of σ; equation (5.68); Lemma 8, equations (5.42), (5.64) and (5.61). Applying the same argument to the new control problem that start at $t = -\delta_0$, we obtain

$$\int_{0}^{\delta_0} \left\|\nabla_x^2 V_{u_1}(t,\cdot) - \nabla_x^2 V_{u_2}(t,\cdot)\right\|_{L^2}^2 \mathrm{d}t \leq C\left\|\widehat{u}_1 - \widehat{u}_2\right\|_{L^2}^2 \leq 2C\left\|u_1 - u_2\right\|_{L^2}^2 . \tag{5.70}$$

Combining (5.69) and (5.70), we obtain (5.66). As a follow up, with (5.66) holds, we can use (5.67) to bound $|q_t^1 - q_t^2|$, and obtain (5.53). Therefore, the result for *Step 3* is perfectly proved.

Finally, combining (5.43), (5.47), and (5.66), we get (5.41), so the lemma is proved. We remark that we can also write down the second order adjoint equation (see [Pen90] for example) and prove that

$$\left\|\nabla_x^3 V_{u_1} - \nabla_x^3 V_{u_2}\right\|_{L^2} \leq C\left\|u_1 - u_2\right\|_{L^2} ,$$

using the same method in *Step 3-4*. □

Lemma 10. *Under Assumption 2, there exists a constant positive C_3 s.t.*

$$J[u] - J[u^*] \leq C_3\left\|u - u^*\right\|_{L^2}^2 \tag{5.71}$$

for any $u \in \mathcal{U}$.

Proof. Denote $\varepsilon_0 = \|u - u^*\|_{L^2}$ and let $u = u^* + \varepsilon_0\phi$, then $\|\phi\|_{L^2} = 1$. We denote $u^\varepsilon = u^* + \varepsilon\phi$. Denote the corresponding value function V_{u^ε} by V^ε. Denote the corresponding density function by ρ^ε, with initial condition $\rho^\varepsilon(0,\cdot) \equiv 1$. By Proposition 4,

$$\begin{aligned} &\frac{\mathrm{d}}{\mathrm{d}\varepsilon} J[u^\varepsilon] = \left\langle \frac{\delta J}{\delta u}[u^\varepsilon],\, \phi \right\rangle \\ &= -\int_0^T \int_{\mathcal{X}} \rho^\varepsilon(t,x) \left\langle \nabla_u G(t,x,u^\varepsilon(t,x), -\nabla_x V^\varepsilon, -\nabla_x^2 V^\varepsilon),\, \phi(t,x) \right\rangle \mathrm{d}x\, \mathrm{d}t. \end{aligned} \tag{5.72}$$

In order to show (5.71), it is sufficient to show that $\frac{\mathrm{d}}{\mathrm{d}\varepsilon} J[u^\varepsilon] \le C\varepsilon$ for some uniform constant C (that does not depend on ϕ), because

$$J[u^* + \varepsilon_0\phi] - J[u^*] = \int_0^{\varepsilon_0} \frac{\mathrm{d}}{\mathrm{d}\varepsilon} J[u^\varepsilon]\, \mathrm{d}\varepsilon.$$

We estimate $\nabla_u G$ in (5.72) first.

$$\begin{aligned} &\left|\nabla_u G(t,x,u^\varepsilon(t,x), -\nabla_x V^\varepsilon, -\nabla_x^2 V^\varepsilon)\right| \\ &= \left|\nabla_u G(t,x,u^\varepsilon(t,x), -\nabla_x V^\varepsilon, -\nabla_x^2 V^\varepsilon) - \nabla_u G(t,x,u^*(t,x), -\nabla_x V^*, -\nabla_x^2 V^*)\right| \\ &\le \left|\nabla_u G(t,x,u^\varepsilon(t,x), -\nabla_x V^\varepsilon, -\nabla_x^2 V^\varepsilon) - \nabla_u G(t,x,u^*(t,x), -\nabla_x V^\varepsilon, -\nabla_x^2 V^\varepsilon)\right| \\ &\quad + \left|\nabla_u G(t,x,u^*(t,x), -\nabla_x V^\varepsilon, -\nabla_x^2 V^\varepsilon) - \nabla_u G(t,x,u^*(t,x), -\nabla_x V^*, -\nabla_x^2 V^*)\right| \\ &=: (I) + (II), \end{aligned} \tag{5.73}$$

where we used the optimality condition in the first inequality. Recall that we have denoted $D = \frac{1}{2}\sigma\sigma^\top$. Let us also denote $u^\varepsilon(t,x)$ and $u^*(t,x)$ by u^ε and u^* for simplicity. For (I), we have

$$\begin{aligned} (I) &\le \left|\nabla_u r(x,u^\varepsilon) - \nabla_u r(x,u^*)\right| + \left|\left(\nabla_u b(x,u^\varepsilon) - \nabla_u b(x,u^*)\right)^\top \nabla_x V^\varepsilon\right| \\ &\quad + \left|\nabla_u \operatorname{Tr}\left[\left(D(x,u^\varepsilon) - D(x,u^*)\right)\nabla_x^2 V^\varepsilon\right]\right| \\ &\le L\varepsilon\left|\phi(t,x)\right| + L\varepsilon\left|\phi(t,x)\right| K + L\varepsilon\left|\phi(t,x)\right| K \le C\varepsilon\left|\phi(t,x)\right|, \end{aligned} \tag{5.74}$$

where we have used the Lipschitz conditions in Assumption 2 and boundedness of the value function's derivatives. For (II), we have

$$\begin{aligned}(II) &\leq \left|\nabla_u b(x,u^*)^\top \left(\nabla_x V^\varepsilon - \nabla_x V^*\right)\right| + \left|\nabla_u \operatorname{Tr}\left[D(x,u^*)\left(\nabla_x^2 V^\varepsilon - \nabla_x^2 V^*\right)\right]\right| \\ &\leq K\left(\left|\nabla_x V^\varepsilon - \nabla_x V^*\right| + \left|\nabla_x^2 V^\varepsilon - \nabla_x^2 V^*\right|\right).\end{aligned} \tag{5.75}$$

Combining (5.74) and (5.75) into (5.73), we obtain

$$\begin{aligned}&\left|\nabla_u G(t,x,u^\varepsilon(t,x),-\nabla_x V^\varepsilon,-\nabla_x^2 V^\varepsilon)\right| \\ &\leq C\left(\varepsilon\left|\phi(t,x)\right| + \left|\nabla_x V^\varepsilon - \nabla_x V^*\right| + \left|\nabla_x^2 V^\varepsilon - \nabla_x^2 V^*\right|\right).\end{aligned} \tag{5.76}$$

Therefore, (5.72) has estimate

$$\begin{aligned}&\left|\frac{\mathrm{d}}{\mathrm{d}\varepsilon} J[u^\varepsilon]\right| \\ &\leq \rho_1 \int_0^T \int_{\mathcal{X}} \left(\frac{1}{\varepsilon}\left|\nabla_u G(t,x,u^\varepsilon(t,x),-\nabla_x V^\varepsilon,-\nabla_x^2 V^\varepsilon)\right|^2 + \varepsilon\left|\phi(t,x)\right|^2\right) \mathrm{d}x\,\mathrm{d}t \\ &\leq C\left(\varepsilon\left\|\phi\right\|_{L^2}^2 + \frac{1}{\varepsilon}\left\|\nabla_x V^\varepsilon - \nabla_x V^*\right\|_{L^2}^2 + \frac{1}{\varepsilon}\left\|\nabla_x^2 V^\varepsilon - \nabla_x^2 V^*\right\|_{L^2}^2 + \varepsilon\left\|\phi\right\|_{L^2}^2\right) \leq C\varepsilon,\end{aligned}$$

where we have consecutively used: Proposition 6 and Cauchy's inequality; inequality (5.76); Lemma 9. Therefore, (5.71) holds. □

Lemma 11. *Under Assumption 2, there exists a positive constant C_4 s.t. for any control function $u \in \mathcal{U}$, we have*

$$\left\|V_u - V^*\right\|_{(T;H^2)} \leq C_4 \left\|u - u^*\right\|_{L^2}^{1+\alpha}, \tag{5.77}$$

with $\alpha = \frac{1}{n+3}$.

Remark. *We believe that* (5.77) *holds with $\alpha = 1$, but encountered some technical difficulty to prove it. We give the intuition here. Following the notation in the*

previous lemma, we denote $u^\varepsilon = u^* + \varepsilon\phi$. *Since* $V^\varepsilon(t,x)$ *reaches its minimum at* $\varepsilon = 0$ *for any* (t,x), *we have*

$$\partial_\varepsilon V^\varepsilon(t,x)\big|_{\varepsilon=0} = 0.$$

With sufficient regularity, we have

$$\left(\partial_\varepsilon \nabla_x V^\varepsilon\right)|_{\varepsilon=0} = \left(\nabla_x \partial_\varepsilon V^\varepsilon\right)|_{\varepsilon=0} = \nabla_x\left(\partial_\varepsilon V^\varepsilon\right)|_{\varepsilon=0} = 0,$$

$$\left(\partial_\varepsilon \nabla_x^2 V^\varepsilon\right)|_{\varepsilon=0} = \left(\nabla_x^2 \partial_\varepsilon V^\varepsilon\right)|_{\varepsilon=0} = \nabla_x^2\left(\partial_\varepsilon V^\varepsilon\right)|_{\varepsilon=0} = 0.$$

Making a local Taylor expansion w.r.t. ε, *we know that* $\nabla_x V^\varepsilon - \nabla_x V^*$ *and* $\nabla_x^2 V^\varepsilon - \nabla_x^2 V^*$ *are of order* $\mathcal{O}(\varepsilon^2)$, *which implies* (5.77) *holds with* $\alpha = 1$.

Proof. We will inherit some notations from the lemma. Denote $\varepsilon_0 = \|u - u^*\|_{L^2}$ and let $u = u^* + \varepsilon_0\phi$, then $\|\phi\|_{L^2} = 1$. We denote $u^\varepsilon = u^* + \varepsilon\phi$. Denote the corresponding value function V_{u^ε} by V^ε. Denote the corresponding density function by ρ^ε, with initial condition $\rho^\varepsilon(0,\cdot) \equiv 1$. The key difficulty for the proof is that $\phi(t,x)$ may not lie in $\mathcal{U}$ like $u^*(t,x)$ or $u^\varepsilon(t,x)$, which has K as a bound for itself and its derivatives. $\phi = (u^\varepsilon - u^*)/\varepsilon$ do have some regularity, but the constant for the bounds has a factor of $2/\varepsilon$. We will prove the lemma in three steps, which is similar to Lemma 9.

Step 1. We want to show

$$\|V_u - V^*\|_{L^2} \le C\|u - u^*\|_{L^2}^{1+\alpha}. \tag{5.78}$$

Note that $V_u \ge V^*$, so

$$\int_{\mathcal{X}} |V_u(0,x) - V^*(0,x)|\,\mathrm{d}x = \int_{\mathcal{X}} (V_u(0,x) - V^*(0,x))\,\mathrm{d}x$$
$$= J[u] - J[u^*] \le C_3\|u - u^*\|_{L^2}^2,$$

where we used Lemma 10 in the last inequality. i.e.,

$$\|V^\varepsilon(0,\cdot) - V^*(0,\cdot)\|_{L^1} \le C_3\|u - u^*\|_{L^2}^2.$$

A similar argument with $t \in (0,T)$ as the starting time gives us

$$\|V^\varepsilon(t,\cdot) - V^*(t,\cdot)\|_{L^1} \le C \|u - u^*\|_{L^2}^2 .$$

Therefore, we have

$$\|V^\varepsilon - V^*\|_{L^1} \le C \|u - u^*\|_{L^2}^2 ,$$

which implies (5.78) because V^ε, V^*, u, u^* are bounded.

Step 2. We want to show

$$\|\nabla_x V_u - \nabla_x V^*\|_{L^2} \le C \|u - u^*\|_{L^2}^{1+\alpha} . \tag{5.79}$$

It is sufficient to show the partial derivative in each dimension at $t = 0$ satisfies the estimate in L^1 norm:

$$\|\partial_i V_u(0,\cdot) - \partial_i V^*(0,\cdot)\|_{L^1} \le C \|u - u^*\|_{L^2}^{1+\alpha} , \tag{5.80}$$

because we can repeat the argument in other dimensions and for other $t \in (0,T)$, and the derivatives of the value functions are bounded.

Step 2.1. We reformulate the problem using finite difference in this step. Let $x_1 \in \mathcal{X}$ be a variable and denote $x_2 = x_1 + \delta e_i$ a perturbation. We assume $\delta > 0$ without loss of generality. We have

$$\begin{aligned}
&\|\partial_i V_u(0,\cdot) - \partial_i V^*(0,\cdot)\|_{L^1} = \int_{\mathcal{X}} |\partial_i V_u(0,x_1) - \partial_i V^*(0,x_1)| \,\mathrm{d}x_1 \\
&= \int_{\mathcal{X}} \left| \int_0^{\varepsilon_0} \partial_\varepsilon \partial_i V^\varepsilon(0,x_1) \mathrm{d}\varepsilon \right| \mathrm{d}x_1 = \int_{\mathcal{X}} \left| \int_0^{\varepsilon_0} \partial_i \partial_\varepsilon V^\varepsilon(0,x_1) \mathrm{d}\varepsilon \right| \mathrm{d}x_1 \\
&= \int_{\mathcal{X}} \left| \int_0^{\varepsilon_0} \lim_{\delta \to 0} \frac{1}{\delta} \left(\partial_\varepsilon V^\varepsilon(0,x_2) - \partial_\varepsilon V^\varepsilon(0,x_1) \right) \mathrm{d}\varepsilon \right| \mathrm{d}x_1 \\
&\le \liminf_{\delta \to 0} \frac{1}{\delta} \int_0^{\varepsilon_0} \int_{\mathcal{X}} |\partial_\varepsilon V^\varepsilon(0,x_2) - \partial_\varepsilon V^\varepsilon(0,x_1)| \,\mathrm{d}x_1 \,\mathrm{d}\varepsilon,
\end{aligned} \tag{5.81}$$

where the last inequality is because of Fatou's lemma. Now, we denote $x_t^{1,\varepsilon}$ and $x_t^{2,\varepsilon}$ the state processes under control u^ε that start at $x_0^{1,\varepsilon} = x_1$ and $x_0^{2,\varepsilon} = x_2$. Here $x_t^{1,\varepsilon}$

and $x_t^{2,\varepsilon}$ share the same realization of Brownian motion. By Proposition 6,

$$\begin{aligned}\partial_\varepsilon V^\varepsilon(0,x_1) &= \mathbb{E}\left[\int_0^T \left\langle \nabla_u G(t, x_t^{1,\varepsilon}, u^\varepsilon(t, x_t^{1,\varepsilon}), -\nabla_x V^\varepsilon, -\nabla_x^2 V^\varepsilon),\, \phi(t, x_t^{1,\varepsilon})\right\rangle \mathrm{d}t\right] \\ &=: \mathbb{E}\left[\int_0^T \left\langle \nabla_u G_t^{1,\varepsilon},\, \phi_t^{1,\varepsilon}\right\rangle \mathrm{d}t\right].\end{aligned}$$

Similarly, $\partial_\varepsilon V^\varepsilon(0,x_2) = \mathbb{E}\left[\int_0^T \left\langle \nabla_u G_t^{2,\varepsilon},\, \phi_t^{2,\varepsilon}\right\rangle \mathrm{d}t\right]$. So

$$\begin{aligned}&\int_{\mathcal{X}} |\partial_\varepsilon V^\varepsilon(0,x_2) - \partial_\varepsilon V^\varepsilon(0,x_1)|\, \mathrm{d}x_1 \\ &= \int_{\mathcal{X}} \left|\mathbb{E}\left[\int_0^T \left\langle \nabla_u G_t^{1,\varepsilon},\, \phi_t^{1,\varepsilon}\right\rangle - \left\langle \nabla_u G_t^{2,\varepsilon},\, \phi_t^{2,\varepsilon}\right\rangle \mathrm{d}t\right]\right| \mathrm{d}x_1 \qquad (5.82) \\ &\le \int_{\mathcal{X}} \mathbb{E}\left[\int_0^T \left|\left\langle \nabla_u G_t^{1,\varepsilon},\, \phi_t^{1,\varepsilon}\right\rangle - \left\langle \nabla_u G_t^{2,\varepsilon},\, \phi_t^{2,\varepsilon}\right\rangle\right| \mathrm{d}t\right] \mathrm{d}x_1\end{aligned}$$

By (5.81) and (5.82), in order to show (5.80), it is sufficient to show that

$$\int_{\mathcal{X}} \mathbb{E}\left[\int_0^T \left|\left\langle \nabla_u G_t^{1,\varepsilon},\, \phi_t^{1,\varepsilon}\right\rangle - \left\langle \nabla_u G_t^{2,\varepsilon},\, \phi_t^{2,\varepsilon}\right\rangle\right| \mathrm{d}t\right] \mathrm{d}x_1 \le C\delta\varepsilon^\alpha \qquad (5.83)$$

for some uniform constant C. We can assume $\delta \le \varepsilon$ because $\nabla_u G^\varepsilon|_{\varepsilon=0} = 0$, hence (5.83) is obvious when $\varepsilon = 0$.

Step 2.2. We split into two sub-tasks to show (5.83) in this step. Let us denote $\rho^{1,\varepsilon}(t,x)$ and $\rho^{2,\varepsilon}(t,x)$ the density functions of $x_t^{1,\varepsilon}$ and $x_t^{2,\varepsilon}$, then $\rho^{j,\varepsilon}(0,\cdot) = \delta_{x_j}$ ($j = 1,2$) and (5.83) can be rewritten as

$$\begin{aligned}\int_{\mathcal{X}}\int_0^T\int_{\mathcal{X}} \Big| &\left\langle \nabla_u G(t,x,u^\varepsilon, -\nabla_x V^\varepsilon, -\nabla_x^2 V^\varepsilon),\, \phi(t,x)\right\rangle \\ &\left(\rho^{1,\varepsilon}(t,x) - \rho^{2,\varepsilon}(t,x)\right)\Big| \,\mathrm{d}x\,\mathrm{d}t\,\mathrm{d}x_1 \le C\delta\varepsilon^\alpha.\end{aligned} \qquad (5.84)$$

The idea to prove *Step 2* is to decompose the time interval $[0,T]$ into two sub-intervals $[0,\varepsilon^{2\alpha}]$ and $(\varepsilon^{2\alpha},T]$ and prove (5.83) and (5.84) with $\int_0^T$ replaced by the corresponding

intervals respectively. For the first part, we take advantage that $\varepsilon^{2\alpha}$ is small, while for the second part, we we use the fact that $\rho^{1,\varepsilon}(t,x)$ and $\rho^{2,\varepsilon}(t,x)$ are nicely mixed.

Step 2.3. We estimate the integration in the interval $[0,\varepsilon^{2\alpha}]$ in this step. We want to show

$$\int_{\mathcal{X}} \mathbb{E}\left[\int_0^{\varepsilon^{2\alpha}} \left|\left\langle \nabla_u G_t^{1,\varepsilon},\, \phi_t^{1,\varepsilon}\right\rangle - \left\langle \nabla_u G_t^{2,\varepsilon},\, \phi_t^{2,\varepsilon}\right\rangle\right| \mathrm{d}t\right] \mathrm{d}x_1 \leq C\delta\varepsilon^{\alpha}. \tag{5.85}$$

Using the Lipschitz property and boundedness of $\nabla_u r$, $\nabla_u b$, $\nabla_u D$, $\nabla_x V^\varepsilon$, and $\nabla_x^2 V^\varepsilon$, we can show

$$\left|\nabla_u G_t^{1,\varepsilon} - \nabla_u G_t^{2,\varepsilon}\right| \leq C\left|x_t^{1,\varepsilon} - x_t^{2,\varepsilon}\right|.$$

Also, we have $\left|\phi_t^{1,\varepsilon} - \phi_t^{2,\varepsilon}\right| \leq \dfrac{2L}{\varepsilon}\left|x_t^{1,\varepsilon} - x_t^{2,\varepsilon}\right|$. Therefore,

$$\begin{aligned}
&\left|\left\langle \nabla_u G_t^{1,\varepsilon},\, \phi_t^{1,\varepsilon}\right\rangle - \left\langle \nabla_u G_t^{2,\varepsilon},\, \phi_t^{2,\varepsilon}\right\rangle\right| \\
&= \left|\left\langle \nabla_u G_t^{1,\varepsilon} - \nabla_u G_t^{2,\varepsilon},\, \phi_t^{1,\varepsilon}\right\rangle + \left\langle \nabla_u G_t^{2,\varepsilon},\, \phi_t^{1,\varepsilon} - \phi_t^{2,\varepsilon}\right\rangle\right| \\
&\leq \left|\nabla_u G_t^{1,\varepsilon} - \nabla_u G_t^{2,\varepsilon}\right|\,\left|\phi_t^{1,\varepsilon}\right| + \left|\nabla_u G_t^{2,\varepsilon}\right|\,\left|\phi_t^{1,\varepsilon} - \phi_t^{2,\varepsilon}\right| \\
&\leq C\left(\left|\phi_t^{1,\varepsilon}\right| + \left|\phi_t^{2,\varepsilon}\right| + \frac{1}{\varepsilon}\left|\nabla_x V^\varepsilon(t,x_t^{2,\varepsilon}) - \nabla_x V^*(t,x_t^{2,\varepsilon})\right| + \right. \\
&\qquad \left. \frac{1}{\varepsilon}\left|\nabla_x^2 V^\varepsilon(t,x_t^{2,\varepsilon}) - \nabla_x^2 V^*(t,x_t^{2,\varepsilon})\right|\right)\left|x_t^{1,\varepsilon} - x_t^{2,\varepsilon}\right|
\end{aligned}$$

where we have used (5.76) to estimate $\left|\nabla_u G_t^{2,\varepsilon}\right|$ in the last inequality. Substituting

the estimate above into (5.85) left, we obtain

$$
\begin{aligned}
&\int_{\mathcal{X}} \mathbb{E}\left[\int_0^{\varepsilon^{2\alpha}} \left|\left\langle \nabla_u G_t^{1,\varepsilon},\, \phi_t^{1,\varepsilon}\right\rangle - \left\langle \nabla_u G_t^{2,\varepsilon},\, \phi_t^{2,\varepsilon}\right\rangle\right| \mathrm{d}t\right] \mathrm{d}x_1 \\
&\leq C \int_{\mathcal{X}} \mathbb{E}\left[\int_0^{\varepsilon^{2\alpha}} \left(\delta\varepsilon^{\alpha} \left|\phi_t^{1,\varepsilon}\right|^2 + \delta\varepsilon^{\alpha} \left|\phi_t^{2,\varepsilon}\right|^2 + \delta\varepsilon^{\alpha-2} \left|\nabla_x V^{\varepsilon}(t, x_t^{2,\varepsilon}) - \nabla_x V^{*}(t, x_t^{2,\varepsilon})\right|^2\right.\right. \\
&\qquad \left.\left. \delta\varepsilon^{\alpha-2} \left|\nabla_x^2 V^{\varepsilon}(t, x_t^{2,\varepsilon}) - \nabla_x^2 V^{*}(t, x_t^{2,\varepsilon})\right|^2 + \frac{1}{\delta\varepsilon^{\alpha}} \left|x_t^{1,\varepsilon} - x_t^{2,\varepsilon}\right|^2\right) \mathrm{d}t\right] \mathrm{d}x_1 \\
&\leq C\rho_1 \left(2\delta\varepsilon^{\alpha} \left\|\phi\right\|_{L^2}^2 + \delta\varepsilon^{\alpha-2} \left\|V^{\varepsilon} - V^{*}\right\|_{(T;H^2)}^2\right) + C \int_{\mathcal{X}} \frac{1}{\delta\varepsilon^{\alpha}} \int_0^{\varepsilon^{2\alpha}} \mathbb{E}\left|x_t^{1,\varepsilon} - x_t^{2,\varepsilon}\right|^2 \mathrm{d}t\, \mathrm{d}x_1 \\
&\leq C\left(\delta\varepsilon^{\alpha} + \delta\varepsilon^{\alpha-2}\varepsilon^2\right) + C \int_{\mathcal{X}} \frac{1}{\delta\varepsilon^{\alpha}} \int_0^{\varepsilon^{2\alpha}} C_1 \left|x_1 - x_2\right|^2 \mathrm{d}t\, \mathrm{d}x_1 \\
&\leq C\delta\varepsilon^{\alpha} + \frac{C}{\delta\varepsilon^{\alpha}}\varepsilon^{2\alpha}\delta^2 \leq C\delta\varepsilon^{\alpha}.
\end{aligned}
\tag{5.86}
$$

Here, the first inequality is just Cauchy's inequality. For the third inequality, we used Lemma 9 and the Gronwall inequality (5.34). In the fourth inequality, we used $|x_1 - x_2| = \delta$. We give an explanation of the second inequality in (5.86) next. After confirming this second inequality, we get (5.85).

We will use a similar argument many times later in this proof. Although $x_0^{1,\varepsilon} = x_1$ and $x_0^{2,\varepsilon} = x_2$ are fixed points, we are integrating x_1 over $\mathcal{X}$ (with $x_2 - x_1 = \delta e_i$ fixed). So, we can define two new processes $\overline{x}_t^{1,\varepsilon}$ and $\overline{x}_t^{2,\varepsilon}$ that have the same dynamic as $x_t^{1,\varepsilon}$ and $x_t^{2,\varepsilon}$, but start at uniform distribution in $\mathcal{X}$, with $\overline{x}_t^{2,\varepsilon} - \overline{x}_t^{1,\varepsilon} \equiv \delta e_i$. The densities for $\overline{x}_t^{2,\varepsilon}$ and $\overline{x}_t^{1,\varepsilon}$ (denoted by $\overline{\rho}^{1,\varepsilon}(t,x)$ and $\overline{\rho}^{2,\varepsilon}(t,x)$) satisfies the estimate in

Proposition 6. Therefore,

$$\int_{\mathcal{X}} \mathbb{E}\left[\int_0^{\varepsilon^{2\alpha}} \left|\phi_t^{1,\varepsilon}\right|^2 \mathrm{d}t\right] \mathrm{d}x_1 \equiv \int_{\mathcal{X}} \mathbb{E}\left[\int_0^{\varepsilon^{2\alpha}} \left|\phi(t, x_t^{1,\varepsilon})\right|^2 \mathrm{d}t \;\middle|\; x_0^{1,\varepsilon} = x_1\right] \mathrm{d}x_1$$

$$= \mathbb{E}_{\overline{x}_0^{1,\varepsilon} \sim \mathrm{Unif}(\mathcal{X})} \mathbb{E}\left[\int_0^{\varepsilon^{2\alpha}} \left|\phi(t, \overline{x}_t^{1,\varepsilon})\right|^2 \mathrm{d}t \;\middle|\; \overline{x}_0^{1,\varepsilon}\right] = \mathbb{E}\int_0^{\varepsilon^{2\alpha}} \left|\phi(t, \overline{x}_t^{1,\varepsilon})\right|^2 \mathrm{d}t \tag{5.87}$$

$$= \int_{\mathcal{X}} \int_0^{\varepsilon^{2\alpha}} |\phi(t,x)|^2 \, \overline{\rho}^{1,\varepsilon}(t,x) \, \mathrm{d}t \, \mathrm{d}x \leq \rho_1 \int_{\mathcal{X}} \int_0^{\varepsilon^{2\alpha}} |\phi(t,x)|^2 \, \mathrm{d}t \, \mathrm{d}x \leq \rho_1 \left\|\phi\right\|_{L^2}^2 .$$

$\phi_t^{2,\varepsilon}$ satisfies the same inequality. The analysis for the $\nabla_x V$ and $\nabla_x^2 V$ terms are exactly the same. Therefore, we can apply Peoposition 6, and the second inequality in (5.86) holds. Hence, we confirm that (5.85) holds.

Step 2.4. We estimate the integration in the interval $[\varepsilon^{2\alpha}, T]$ in this step. We want to show

$$\int_{\varepsilon^{2\alpha}}^T \int_{\mathcal{X}} \left|\left\langle \nabla_u G(t,x,u^\varepsilon, -\nabla_x V^\varepsilon, -\nabla_x^2 V^\varepsilon),\, \phi(t,x)\right\rangle \left(\rho^{1,\varepsilon}(t,x) - \rho^{2,\varepsilon}(t,x)\right)\right| \mathrm{d}x \, \mathrm{d}t$$

$$\leq C\delta\varepsilon^\alpha. \tag{5.88}$$

We recall that ρ is the solution of the Fokker Planck equation $\partial_t \rho = \mathcal{G}_\varepsilon^\dagger \rho$, where $\mathcal{G}_\varepsilon$ is the infinitesimal generator of the state process with control u^ε and $\mathcal{G}_\varepsilon^\dagger$ is its adjoint. Let us use $p^\varepsilon(t,x;s,y)$ $(t \geq s)$ to denote the fundamental solution of this PDE. Then, $\rho^{j,\varepsilon}(t,x) = p^\varepsilon(t,x;0,x_j)$ for $j = 1,2$. The fundamental solution of linear parabolic PDE is well-studied, and a comprehensive description can be found in [Fri08]. A key observation of the fundamental solution p^ε is that $q^\varepsilon(t,x;s,y) := p^\varepsilon(s,y;t,x)$ is the fundamental solution of the backward Kolmogorov equation $\partial_t \psi + \mathcal{G}_\varepsilon \psi = 0$ [Itô53]. Therefore, the regularity of $p^\varepsilon(t,x;s,y)$ in y (here $t \geq s$) is equivalent to the regularity of $q^\varepsilon(t,x;s,y)$ in x (here $s \geq t$). Aronson proved (in [Aro59] Lemma 4.2)

that

$$\left|\nabla_x^k q^\varepsilon(t,x;s,y)\right| \le C^{(k)}(s-t)^{-(n+k)/2}. \tag{5.89}$$

Applying a standard mean value theorem and this lemma (5.89) with $k=1$ to q^ε, we obtain

$$\begin{aligned}
&\left|\rho^{1,\varepsilon}(t,x)-\rho^{2,\varepsilon}(t,x)\right| = \left|q^\varepsilon(0,x_1;t,x)-q^\varepsilon(0,x_2;t,x)\right| \\
&= \left|\left\langle \nabla_x q^\varepsilon(0,(1-c)x_1+cx_2;t,x),\, x_1-x_2\right\rangle\right| \\
&=\le Ct^{-(1+n)/2}\left|x_1-x_2\right| = Ct^{-(1+n)/2}\delta,
\end{aligned} \tag{5.90}$$

where we clarify that ∇_x is operated on the second (not fourth) argument on

$$q^\varepsilon(t,x;s,y).$$

Therefore,

$$\begin{aligned}
&\int_{\varepsilon^{2\alpha}}^T\int_{\mathcal{X}} \left|\left\langle \nabla_u G(t,x,u^\varepsilon,-\nabla_x V^\varepsilon,-\nabla_x^2 V^\varepsilon),\, \phi(t,x)\right\rangle\left(\rho^{1,\varepsilon}(t,x)-\rho^{2,\varepsilon}(t,x)\right)\right| \mathrm{d}x\,\mathrm{d}t \\
&\le C\int_{\varepsilon^{2\alpha}}^T\int_{\mathcal{X}} \left(\varepsilon\left|\phi(t,x)\right| + \left|\nabla_x V^\varepsilon-\nabla_x V^*\right| + \left|\nabla_x^2 V^\varepsilon-\nabla_x^2 V^*\right|\right) \\
&\quad \left|\phi(t,x)\right| t^{-(1+n)/2}\delta\,\mathrm{d}x\,\mathrm{d}t \\
&\le C\delta\varepsilon^{-\alpha-n\alpha}\int_{\varepsilon^{2\alpha}}^T\int_{\mathcal{X}}\left(\varepsilon\left|\phi(t,x)\right|^2+\frac{1}{\varepsilon}\left|\nabla_x V^\varepsilon-\nabla_x V^*\right|^2+\frac{1}{\varepsilon}\left|\nabla_x^2 V^\varepsilon-\nabla_x^2 V^*\right|^2\right)\mathrm{d}x\,\mathrm{d}t \\
&\le C\delta\varepsilon^{-\alpha-n\alpha}\left(\varepsilon\left\|\phi\right\|_{L^2}^2+\frac{1}{\varepsilon}\left\|V^\varepsilon-V^*\right\|_{(T;H^2)}^2\right) \le C\delta\varepsilon^{1-\alpha-n\alpha}\le C\delta\varepsilon^{\alpha}.
\end{aligned}$$

We used (5.76) and (5.90) in the first inequality, and used Lemma 9 in the thourth inequality. So, (5.88) holds.

To conclude, we combine (5.85) and (5.88) and recover (5.83). Therefore, (5.80), hence (5.79) holds.

Step 3. We want to show

$$\left\|\nabla_x^2 V_u-\nabla_x^2 V^*\right\|_{L^2} \le C\left\|u-u^*\right\|_{L^2}^{1+\alpha}. \tag{5.91}$$

The idea to show (5.91) is similar as in *Step 2*. It is sufficient to show

$$\left\|\nabla_x^2 V_u(0,\cdot) - \nabla_x^2 V^*(0,\cdot)\right\|_{L^1} \leq C \left\|u - u^*\right\|_{L^2}^{1+\alpha} \tag{5.92}$$

because the same argument applies to other $t \in (0,T)$. We will use the idea of finite difference and cut $[0,T]$ into two intervals with separate estimate.

Step 3.1. We reformulate the problem using finite difference in this step. Let $x_1 \in \mathcal{X}$ be a variable.

$$\begin{aligned}
&\left\|\nabla_x^2 V_u(0,\cdot) - \nabla_x^2 V^*(0,\cdot)\right\|_{L^1} = \int_{\mathcal{X}} \left|\nabla_x^2 V_u(0,x_1) - \nabla_x^2 V^*(0,x_1)\right| \mathrm{d}x_1 \\
&= \int_{\mathcal{X}} \left| \int_0^{\varepsilon_0} \partial_\varepsilon \nabla_x^2 V^\varepsilon(0,x_1) \mathrm{d}\varepsilon \right| \mathrm{d}x_1 = \int_{\mathcal{X}} \left| \int_0^{\varepsilon_0} \nabla_x^2 \partial_\varepsilon V^\varepsilon(0,x_1) \mathrm{d}\varepsilon \right| \mathrm{d}x_1 \\
&\leq \int_0^{\varepsilon_0} \int_{\mathcal{X}} \left|\nabla_x^2 \partial_\varepsilon V^\varepsilon(0,x_1)\right| \mathrm{d}x_1 \, \mathrm{d}\varepsilon.
\end{aligned}$$

So, it is sufficient to show

$$\int_{\mathcal{X}} \left|\nabla_x^2 \partial_\varepsilon V^\varepsilon(0,x_1)\right| \mathrm{d}x_1 \leq C\varepsilon^\alpha \tag{5.93}$$

for all $\varepsilon \in (0,\varepsilon_0)$ in order to recover (5.92). To compute the Hessian, let z be a perturbation vector with $|z| = 1$. Denote $x_0 = x_1 - \delta z$ and $x_2 = x_1 + \delta z$. Without loss of generality, we just consider $\delta > 0$. Then,

$$\lim_{\delta \to 0} \frac{1}{\delta^2} \left(\partial_\varepsilon V^\varepsilon(0,x_0) + \partial_\varepsilon V^\varepsilon(0,x_2) - 2\partial_\varepsilon V^\varepsilon(0,x_1)\right) = z^\top \nabla_x^2 \partial_\varepsilon V^\varepsilon(0,x_1) z.$$

We recall that $\|\cdot\|_2$ denotes the matrix spectrum norm. Since

$$\left|\nabla_x^2 \partial_\varepsilon V^\varepsilon(0,x_1)\right| \leq \sqrt{n} \left\|\nabla_x^2 \partial_\varepsilon V^\varepsilon(0,x_1)\right\|_2,$$

we only need to show that there exists C that does not depend on z, such that

$$\int_{\mathcal{X}} \left|z^\top \nabla_x^2 \partial_\varepsilon V^\varepsilon(0,x_1) z\right| \mathrm{d}x_1 \leq C\varepsilon^\alpha \tag{5.94}$$

for all $|z| = 1$, in order to get (5.93). The left hand side of (5.94) satisfies

$$\begin{aligned}
&\int_{\mathcal{X}} \left| z^\top \nabla_x^2 \partial_\varepsilon V^\varepsilon(0, x_1) z \right| \, \mathrm{d}x_1 \\
&= \int_{\mathcal{X}} \lim_{\delta \to 0} \frac{1}{\delta^2} \left| \partial_\varepsilon V^\varepsilon(0, x_0) + \partial_\varepsilon V^\varepsilon(0, x_2) - 2\partial_\varepsilon V^\varepsilon(0, x_1) \right| \, \mathrm{d}x_1 \qquad (5.95) \\
&\leq \liminf_{\delta \to 0} \frac{1}{\delta^2} \int_{\mathcal{X}} \left| \partial_\varepsilon V^\varepsilon(0, x_0) + \partial_\varepsilon V^\varepsilon(0, x_2) - 2\partial_\varepsilon V^\varepsilon(0, x_1) \right| \, \mathrm{d}x_1,
\end{aligned}$$

where we used Fatou's lemma in the last inequality. And

$$\begin{aligned}
&\int_{\mathcal{X}} \left| \partial_\varepsilon V^\varepsilon(0, x_0) + \partial_\varepsilon V^\varepsilon(0, x_2) - 2\partial_\varepsilon V^\varepsilon(0, x_1) \right| \, \mathrm{d}x_1 \\
&= \int_{\mathcal{X}} \left| \mathbb{E} \left[\int_0^T \left\langle \nabla_u G_t^{0,\varepsilon}, \, \phi_t^{0,\varepsilon} \right\rangle + \left\langle \nabla_u G_t^{2,\varepsilon}, \, \phi_t^{2,\varepsilon} \right\rangle - 2 \left\langle \nabla_u G_t^{1,\varepsilon}, \, \phi_t^{1,\varepsilon} \right\rangle \mathrm{d}t \right] \right| \, \mathrm{d}x_1 \\
&\leq \int_{\mathcal{X}} \mathbb{E} \left[\int_0^T \left| \left\langle \nabla_u G_t^{0,\varepsilon}, \, \phi_t^{0,\varepsilon} \right\rangle + \left\langle \nabla_u G_t^{2,\varepsilon}, \, \phi_t^{2,\varepsilon} \right\rangle - 2 \left\langle \nabla_u G_t^{1,\varepsilon}, \, \phi_t^{1,\varepsilon} \right\rangle \right| \mathrm{d}t \right] \, \mathrm{d}x_1 \\
&\leq \int_{\mathcal{X}} \int_0^T \int_{\mathcal{X}} \left| \left\langle \nabla_u G(t, x, u^\varepsilon, -\nabla_x V^\varepsilon, -\nabla_x^2 V^\varepsilon), \, \phi(t, x) \right\rangle \right| \left| \rho^{0,\varepsilon} + \rho^{2,\varepsilon} - 2\rho^{1,\varepsilon} \right| \mathrm{d}x \, \mathrm{d}t \, \mathrm{d}x_1,
\end{aligned} \qquad (5.96)$$

where we used (5.12) in Proposition 4 for the first equality. Combining (5.95) and (5.96), it is sufficient to show

$$\int_{\mathcal{X}} \mathbb{E} \left[\int_0^{\varepsilon^{2\alpha}} \left| \left\langle \nabla_u G_t^{0,\varepsilon}, \, \phi_t^{0,\varepsilon} \right\rangle + \left\langle \nabla_u G_t^{2,\varepsilon}, \, \phi_t^{2,\varepsilon} \right\rangle - 2 \left\langle \nabla_u G_t^{1,\varepsilon}, \, \phi_t^{1,\varepsilon} \right\rangle \right| \mathrm{d}t \right] \, \mathrm{d}x_1 \leq C\delta^2 \varepsilon^\alpha \qquad (5.97)$$

and

$$\int_{\varepsilon^{2\alpha}}^T \int_{\mathcal{X}} \left| \left\langle \nabla_u G(t, x, u^\varepsilon, -\nabla_x V^\varepsilon, -\nabla_x^2 V^\varepsilon), \, \phi(t, x) \right\rangle \right| \left| \rho^{0,\varepsilon} + \rho^{2,\varepsilon} - 2\rho^{1,\varepsilon} \right| \mathrm{d}x \, \mathrm{d}t \leq C\delta^2 \varepsilon^\alpha \qquad (5.98)$$

in order to recover (5.94) and hence (5.93). These two inequalities are tasks for later steps. Again, we only need to verify them when $\delta \leq \varepsilon$.

Step 3.2. We derive some generalizations of mean value theorem and Gronwall inequalities in this step. Let $f(s) : \mathbb{R} \to \mathbb{R}$ be a smooth function. Then

$$f(1) + f(-1) - 2f(0) = \int_0^1 (f'(s) - f'(-s))\,\mathrm{d}s = \int_0^1 \int_{-s}^{s} f''(\tau)\,\mathrm{d}\tau\,\mathrm{d}s.$$

Using the mean value theorem, we know that there exists $c \in [-1, 1]$ s.t. $f(1) + f(-1) - 2f(0) = f''(c)$. More generally, let $x_j \in \mathcal{X}$ $(j = 0, 1, 2)$ with $2x_1 = x_0 + x_2$ and let $f : s \mapsto g(x_1 + s(x_2 - x_1))$ for some smooth function g. Then

$$\begin{aligned} &g(x_2) + g(x_0) - 2g(x_1) = f(1) + f(-1) - 2f(0) = f''(c) \\ &= (x_2 - x_1)^\top \nabla_x^2 g((1-c)x_1 + cx_2)(x_2 - x_1) \end{aligned} \tag{5.99}$$

for some $c \in [-1, 1]$. We will use this result in later steps. We give some Gronwall inequalities next.

Let x_t^j $(j = 0, 1, 2)$ be the state processes that start at $x_0^j = x_j$ with $x_0 = x_1 - \delta z$ and $x_2 = x_1 + \delta z$ and $|z| = 1$. Here x_t^j share the same control function u, which could be u^* or u^ε. As two corollaries of (5.34), We want to show

$$\sup_{t\in[0,T]} \mathbb{E}\left|x_t^1 - x_t^2\right|^4 \le C\mathbb{E}\left|x_0^1 - x_0^2\right|^4 = C\delta^4 \tag{5.100}$$

and

$$\sup_{t\in[0,T]} \mathbb{E}\left|x_t^0 + x_t^2 - 2x_t^1\right|^2 \le C\delta^4. \tag{5.101}$$

By Itô's formula,

$$\begin{aligned} \mathrm{d}\left|x_t^1 - x_t^2\right|^4 = \Big[&4(x_t^1 - x_t^2)^\top \left(\sigma_t^1 - \sigma_t^2\right)\left(\sigma_t^1 - \sigma_t^2\right)^\top (x_t^1 - x_t^2) + 2\left|x_t^1 - x_t^2\right|^2 \\ &\left(\left|\sigma_t^1 - \sigma_t^2\right|^2 + 2\left\langle x_t^1 - x_t^2,\, b_t^1 - b_t^2\right\rangle\right)\Big]\,\mathrm{d}t + 4\left|x_t^1 - x_t^2\right|^2 (x_t^1 - x_t^2)^\top(\sigma_t^1 - \sigma_t^2)\mathrm{d}W_t, \end{aligned}$$

where we have inherit the notation $b_t^j = b(x_t^j, u(t, x_t^j))$ and $\sigma_t^j = \sigma(x_t^j, u(t, x_t^j))$ in the proof of Lemma 8. Integrating, taking expectation, and using (5.37) and (5.38), we

obtain

$$\mathbb{E}\left|x_T^1 - x_T^2\right|^4 \leq \mathbb{E}\left|x_0^1 - x_0^2\right|^4 + C\mathbb{E}\int_0^T \mathbb{E}\left|x_t^1 - x_t^2\right|^4 = \delta^4 + C\mathbb{E}\int_0^T \mathbb{E}\left|x_t^1 - x_t^2\right|^4,$$

where we used the Lipschitz condition for b and σ in Assumption 2. This inequality also holds for $T' < T$. Therefore, applying a Gronwall's inequality gives us $\mathbb{E}\left|x_T^1 - x_T^2\right|^4 \leq C\delta^4$. Therefore, (5.100) holds.

We show (5.101) next. By Itô's formula,

$$\begin{aligned}\mathrm{d}\left|x_t^0 + x_t^2 - 2x_t^1\right|^2 = &\left[\left|\sigma_t^0 + \sigma_t^2 - 2\sigma_t^1\right|^2 + 2\left\langle x_t^0 + x_t^2 - 2x_t^1,\, b_t^0 - b_t^2 - 2b_t^1\right\rangle\right]\mathrm{d}t \\ &+ 2(x_t^0 + x_t^2 - 2x_t^1)^\top(\sigma_t^0 + \sigma_t^2 - 2\sigma_t^1)\mathrm{d}W_t.\end{aligned} \tag{5.102}$$

We pick the $i-$th entry of the vector valued function b and denote the map $(t,x) \mapsto b_i(x, u(t,x))$ by $B_i(t,x)$ for $i = 1, 2, \ldots, n$. By Assumption 2, $B_i(t,x)$ is Lipschitz in x and has bounded Hessian in x. Apply the mean value theorem (5.99), we get

$$\begin{aligned}&\left|B_i(t, x_t^0) + B_i(t, x_t^2) - 2B_i(t, x_t^1)\right| \\ \leq &\left|B_i(t, 2x_t^1 - x_t^2) + B_i(t, x_t^2) - 2B_i(t, x_t^1)\right| + \left|B_i(t, 2x_t^1 - x_t^2) - B_i(t, x_t^0)\right| \\ \leq &(x_t^2 - x_t^1)^\top \nabla_x^2 B_i(t, x_t^1 + c(x_t^2 - x_t^1))\,(x_t^2 - x_t^1) + C\left|x_t^0 + x_t^2 - 2x_t^1\right| \\ \leq &C\left(\left|x_t^2 - x_t^1\right|^2 + \left|x_t^0 + x_t^2 - 2x_t^1\right|\right).\end{aligned}$$

Therefore,

$$\left|b_t^0 - b_t^2 - 2b_t^1\right| \leq C\left(\left|x_t^2 - x_t^1\right|^2 + \left|x_t^0 + x_t^2 - 2x_t^1\right|\right). \tag{5.103}$$

Similarly, we have

$$\left|\sigma_t^0 - \sigma_t^2 - 2\sigma_t^1\right| \leq C\left(\left|x_t^2 - x_t^1\right|^2 + \left|x_t^0 + x_t^2 - 2x_t^1\right|\right). \tag{5.104}$$

Integrating (5.102), taking expectation, we obtain

$$\begin{aligned}
&\mathbb{E}\left|x_T^0 + x_T^2 - 2x_T^1\right|^2 = \mathbb{E}\left|x_0^0 + x_0^2 - 2x_0^1\right|^2 \\
&+ \mathbb{E}\int_0^T \left(\left|\sigma_t^0 + \sigma_t^2 - 2\sigma_t^1\right|^2 + 2\left\langle x_t^0 + x_t^2 - 2x_t^1,\, b_t^0 - b_t^2 - 2b_t^1\right\rangle\right) \mathrm{d}t \\
&\leq 0 + \mathbb{E}\int_0^T \left(\left|\sigma_t^0 + \sigma_t^2 - 2\sigma_t^1\right|^2 + \left|x_t^0 + x_t^2 - 2x_t^1\right|^2 + \left|b_t^0 - b_t^2 - 2b_t^1\right|^2\right) \mathrm{d}t \\
&\leq C\mathbb{E}\int_0^T \left(\left|x_t^2 - x_t^1\right|^4 + \left|x_t^0 + x_t^2 - 2x_t^1\right|^2\right) \mathrm{d}t \leq C\delta^4 + C\mathbb{E}\int_0^T \left|x_t^0 + x_t^2 - 2x_t^1\right|^2 \mathrm{d}t,
\end{aligned} \tag{5.105}$$

where we used (5.103) and (5.104) in the second inequality, and used (5.100) in the third. Applying Gronwall's inequality on (5.105), we get

$$\mathbb{E}\left|x_T^0 + x_T^2 - 2x_T^1\right|^2 \leq C\delta^4$$

and hence recover (5.101).

Step 3.3. We reformulate (5.97) and estimate the first two terms in this step. Let us rewrite (5.97) first.

$$\begin{aligned}
&\int_{\mathcal{X}} \mathbb{E}\left[\int_0^{\varepsilon^{2\alpha}} \left|\left\langle \nabla_u G_t^{0,\varepsilon},\, \phi_t^{0,\varepsilon}\right\rangle + \left\langle \nabla_u G_t^{2,\varepsilon},\, \phi_t^{2,\varepsilon}\right\rangle - 2\left\langle \nabla_u G_t^{1,\varepsilon},\, \phi_t^{1,\varepsilon}\right\rangle\right| \mathrm{d}t\right] \mathrm{d}x_1 \\
&= \int_{\mathcal{X}} \mathbb{E}\left[\int_0^{\varepsilon^{2\alpha}} \left|\left\langle \nabla_u G_t^{0,\varepsilon} + \nabla_u G_t^{2,\varepsilon} - 2\nabla_u G_t^{1,\varepsilon},\, \phi_t^{1,\varepsilon}\right\rangle + \left\langle \nabla_u G_t^{1,\varepsilon},\, \phi_t^{0,\varepsilon} + \phi_t^{2,\varepsilon} - 2\phi_t^{1,\varepsilon}\right\rangle\right.\right. \\
&\qquad \left.\left. \left\langle \nabla_u G_t^{2,\varepsilon} - \nabla_u G_t^{1,\varepsilon},\, \phi_t^{2,\varepsilon} - \phi_t^{1,\varepsilon}\right\rangle + \left\langle \nabla_u G_t^{0,\varepsilon} - \nabla_u G_t^{1,\varepsilon},\, \phi_t^{0,\varepsilon} - \phi_t^{1,\varepsilon}\right\rangle\right| \mathrm{d}t\right] \mathrm{d}x_1 \\
&\leq: (I) + (II) + (III) + (IV),
\end{aligned} \tag{5.106}$$

where we use triangle inequality in the last step and bound (5.106) by four separate integrals, which are denoted by $(I)-(IV)$. Because of Assumption 2 and the regularity of V^ε, the map $(t,x) \mapsto \nabla_u G^\varepsilon(t,x,u^\varepsilon(t,x),-\nabla_x V^\varepsilon,-\nabla_x^2 V^\varepsilon)$ is lipschitp in x and

has bounded Hessian in x. So, we can repeat the argument for (5.103) and obtain

$$\left|\nabla_u G_t^{0,\varepsilon} + \nabla_u G_t^{2,\varepsilon} - 2\nabla_u G_t^{1,\varepsilon}\right| \le C\left(\left|x_t^{2,\varepsilon} - x_t^{1,\varepsilon}\right|^2 + \left|x_t^{0,\varepsilon} + x_t^{2,\varepsilon} - 2x_t^{1,\varepsilon}\right|\right), \quad (5.107)$$

where we have used boundedness of $\nabla_x^4 V^\varepsilon$. Therefore, we can estimate (I) through

$$\begin{aligned}
(I) &= \int_{\mathcal{X}} \mathbb{E}\left[\int_0^{\varepsilon^{2\alpha}} \left|\left\langle \nabla_u G_t^{0,\varepsilon} + \nabla_u G_t^{2,\varepsilon} - 2\nabla_u G_t^{1,\varepsilon},\, \phi_t^{1,\varepsilon}\right\rangle\right| \mathrm{d}t\right] \mathrm{d}x_1 \\
&\le \int_{\mathcal{X}} \mathbb{E}\left[\int_0^{\varepsilon^{2\alpha}} \left(\frac{1}{\delta^2\varepsilon^\alpha}\left|\nabla_u G_t^{0,\varepsilon} + \nabla_u G_t^{2,\varepsilon} - 2\nabla_u G_t^{1,\varepsilon}\right|^2 + \delta^2\varepsilon^\alpha\left|\phi_t^{1,\varepsilon}\right|^2\right) \mathrm{d}t\right] \mathrm{d}x_1 \\
&\le \int_{\mathcal{X}} \int_0^{\varepsilon^{2\alpha}} \mathbb{E}\left[\frac{1}{\delta^2\varepsilon^\alpha}\left|x_t^{2,\varepsilon} - x_t^{1,\varepsilon}\right|^2 + \frac{1}{\delta^2\varepsilon^\alpha}\left|x_t^{0,\varepsilon} + x_t^{2,\varepsilon} - 2x_t^{1,\varepsilon}\right|^2\right] \mathrm{d}t\, \mathrm{d}x_1 + \delta^2\varepsilon^\alpha\rho_1 \left\|\phi\right\|_{L^2}^2 \\
&\le \int_{\mathcal{X}} \int_0^{\varepsilon^{2\alpha}} \left[\frac{1}{\delta^2\varepsilon^\alpha} C\delta^4 + \frac{1}{\delta^2\varepsilon^\alpha} C\delta^4\right] \mathrm{d}t\, \mathrm{d}x_1 + \delta^2\varepsilon^\alpha\rho_1 \le C\delta^2\varepsilon^\alpha.
\end{aligned} \quad (5.108)$$

In the second inequality above, we used (5.107). Also, for the ϕ term, the argument is the same as in *Step 2.3*, see (5.87). In the third inequality above, we used (5.100) and (5.101).

Let us consider (II) next. Similar to (5.107), we can estimate the ϕ term in (II), but note that the constant should scaled by $2/\varepsilon$. (This is explained at the beginning of the proof). So, we have

$$\left|\phi_t^{0,\varepsilon} + \phi_t^{2,\varepsilon} - 2\phi_t^{1,\varepsilon}\right| \le \frac{C}{\varepsilon}\left(\left|x_t^{2,\varepsilon} - x_t^{1,\varepsilon}\right|^2 + \left|x_t^{0,\varepsilon} + x_t^{2,\varepsilon} - 2x_t^{1,\varepsilon}\right|\right). \quad (5.109)$$

Therefore

$$
\begin{aligned}
(II) &= \int_{\mathcal{X}} \mathbb{E}\left[\int_0^{\varepsilon^{2\alpha}} \left|\left\langle \nabla_u G_t^{1,\varepsilon},\, \phi_t^{0,\varepsilon} + \phi_t^{2,\varepsilon} - 2\phi_t^{1,\varepsilon}\right\rangle\right| \mathrm{d}t\right] \mathrm{d}x_1 \\
&\leq \int_{\mathcal{X}} \int_0^{\varepsilon^{2\alpha}} \mathbb{E}\left[\left(\frac{\delta^2}{\varepsilon^{2-\alpha}} \left|\nabla_u G_t^{1,\varepsilon}\right|^2 + \frac{\varepsilon^{2-\alpha}}{\delta^2} \left|\phi_t^{0,\varepsilon} + \phi_t^{2,\varepsilon} - 2\phi_t^{1,\varepsilon}\right|^2\right)\right] \mathrm{d}t\, \mathrm{d}x_1 \\
&\leq C \int_{\mathcal{X}} \int_0^{\varepsilon^{2\alpha}} \mathbb{E}\left[\frac{\delta^2}{\varepsilon^{2-\alpha}} \left(\left|\nabla_x V^\varepsilon(t, x_t^{1,\varepsilon}) - \nabla_x V^*(t, x_t^{1,\varepsilon})\right|^2 + \left|\nabla_x^2 V^\varepsilon(t, x_t^{1,\varepsilon}) - \right.\right.\right. \\
&\left.\left.\left. \nabla_x^2 V^*(t, x_t^{1,\varepsilon})\right|^2 + \varepsilon^2 \left|\phi_t^{1,\varepsilon}\right|^2\right) + \frac{1}{\delta^2 \varepsilon^\alpha} \left(\left|x_t^{2,\varepsilon} - x_t^{1,\varepsilon}\right|^4 + \left|x_t^{0,\varepsilon} + x_t^{2,\varepsilon} - 2x_t^{1,\varepsilon}\right|^2\right)\right] \mathrm{d}t\, \mathrm{d}x_1 \\
&\leq C \frac{\delta^2}{\varepsilon^{2-\alpha}} \left(\rho_1 \left\|V^\varepsilon - V^*\right\|^2_{(T;H^2)} + \varepsilon^2 \rho_1 \left\|\phi\right\|^2_{L^2}\right) + C \int_{\mathcal{X}} \int_0^{\varepsilon^{2\alpha}} \frac{1}{\delta^2 \varepsilon^\alpha} C\delta^4 \mathrm{d}t\, \mathrm{d}x_1 \leq C\delta^2 \varepsilon^\alpha,
\end{aligned}
\tag{5.110}
$$

where we have consecutively used: Cauchy's inequality; the estimate of $\nabla_u G$ in (5.76) and ϕ terms in (5.109); the argument at the end of *Step 2.3* for ϕ in (5.87) and the Gronwall inequalities (5.100), (5.101); Lemma 9.

Step 3.4. We reformulate (III) in (5.106) in this step. (IV) can be analyzed in the same way. We want establish the following estimate in the following few steps.

$$
(III) = \int_{\mathcal{X}} \mathbb{E}\left[\int_0^{\varepsilon^{2\alpha}} \left|\left\langle \nabla_u G_t^{2,\varepsilon} - \nabla_u G_t^{1,\varepsilon},\, \phi_t^{2,\varepsilon} - \phi_t^{1,\varepsilon}\right\rangle\right| \mathrm{d}t\right] \mathrm{d}x_1 \leq C\delta^2 \varepsilon^\alpha. \tag{5.111}
$$

The idea is similar to the derivation of (5.76). Denote $u_t^{j,*} = u^*(t, x_t^{j,\varepsilon})$ for $j = 0, 1, 2$. Denote $f_t^{j,\varepsilon} = f(x_t^{j,\varepsilon}, u^\varepsilon(t, x_t^{j,\varepsilon}))$ and $f_t^{j,*} = f(x_t^{j,\varepsilon}, u^*(t, x_t^{j,\varepsilon}))$ for $f = r, b, \sigma, D$ and $j = 0, 1, 2$. Denote $\nabla_u G_t^{j,*} = \nabla_u G(t, x_t^{j,\varepsilon}, u_t^{j,*}, -\nabla_x V^*, -\nabla_x^2 V^*)$ for $j = 0, 1, 2$. Note that $\nabla_u G_t^{j,*} = 0$ due to maximum condition (5.10). Denote $V_t^{j,\varepsilon} = V^\varepsilon(t, x_t^{j,\varepsilon})$ and

$V_t^{j,*} = V^*(t, x_t^{j,\varepsilon})$. By definition of G (5.7),

$$\begin{aligned}
&\nabla_u G_t^{2,\varepsilon} - \nabla_u G_t^{1,\varepsilon} = \left(\nabla_u G_t^{2,\varepsilon} - \nabla_u G_t^{2,*}\right) - \left(\nabla_u G_t^{1,\varepsilon} - \nabla_u G_t^{1,*}\right) \\
&= \left(\nabla_u r_t^{1,\varepsilon} - \nabla_u r_t^{1,*}\right) - \left(\nabla_u r_t^{2,\varepsilon} - \nabla_u r_t^{2,*}\right) \\
&\quad + \left(\nabla_u b_t^{1,\varepsilon\top} \nabla_x V_t^{1,\varepsilon} - \nabla_u b_t^{1,*\top} \nabla_x V_t^{1,*}\right) - \left(\nabla_u b_t^{2,\varepsilon\top} \nabla_x V_t^{2,\varepsilon} - \nabla_u b_t^{2,*\top} \nabla_x V_t^{2,*}\right) \\
&\quad + \nabla_u \operatorname{Tr}\left(D_t^{1,\varepsilon} \nabla_x^2 V_t^{1,\varepsilon} - D_t^{1,*} \nabla_x^2 V_t^{1,*}\right) - \nabla_u \operatorname{Tr}\left(D_t^{2,\varepsilon} \nabla_x^2 V_t^{2,\varepsilon} - D_t^{2,*} \nabla_x^2 V_t^{2,*}\right)
\end{aligned}$$

Note that the ∇_u only operate on D in the last two terms. Therefore,

$$\int_{\mathcal{X}} \mathbb{E}\left[\int_0^{\varepsilon^{2\alpha}} \left|\left\langle \nabla_u G_t^{2,\varepsilon} - \nabla_u G_t^{1,\varepsilon},\, \phi_t^{2,\varepsilon} - \phi_t^{1,\varepsilon}\right\rangle\right| \mathrm{d}t\right] \mathrm{d}x_1 \le (V) + (VI) + (VII) \quad (5.112)$$

where

$$(V) := \int_{\mathcal{X}} \mathbb{E}\left[\int_0^{\varepsilon^{2\alpha}} \left|\left\langle \left(\nabla_u r_t^{1,\varepsilon} - \nabla_u r_t^{1,*}\right) - \left(\nabla_u r_t^{2,\varepsilon} - \nabla_u r_t^{2,*}\right),\, \phi_t^{2,\varepsilon} - \phi_t^{1,\varepsilon}\right\rangle\right| \mathrm{d}t\right] \mathrm{d}x_1,$$

$$\begin{aligned}
(VI) := \int_{\mathcal{X}} \mathbb{E}\Bigg[\int_0^{\varepsilon^{2\alpha}} \Bigg|\Bigg\langle &\left(\nabla_u b_t^{1,\varepsilon\top} \nabla_x V_t^{1,\varepsilon} - \nabla_u b_t^{1,*\top} \nabla_x V_t^{1,*}\right) \\
&- \left(\nabla_u b_t^{2,\varepsilon\top} \nabla_x V_t^{2,\varepsilon} - \nabla_u b_t^{2,*\top} \nabla_x V_t^{2,*}\right), \phi_t^{2,\varepsilon} - \phi_t^{1,\varepsilon}\Bigg\rangle\Bigg| \mathrm{d}t\Bigg] \mathrm{d}x_1,
\end{aligned}$$

and

$$\begin{aligned}
(VII) := \int_{\mathcal{X}} \mathbb{E}\Bigg[\int_0^{\varepsilon^{2\alpha}} \Big|\Big\langle &\nabla_u \operatorname{Tr}\left(D_t^{1,\varepsilon} \nabla_x^2 V_t^{1,\varepsilon} - D_t^{1,*} \nabla_x^2 V_t^{1,*}\right) \\
&- \nabla_u \operatorname{Tr}\left(D_t^{2,\varepsilon} \nabla_x^2 V_t^{2,\varepsilon} - D_t^{2,*} \nabla_x^2 V_t^{2,*}\right), \phi_t^{2,\varepsilon} - \phi_t^{1,\varepsilon}\Big\rangle\Big| \mathrm{d}t\Bigg] \mathrm{d}x_1.
\end{aligned}$$

We want to show that each term above is less than $C\delta^2\varepsilon^\alpha$ in order to recover (5.111).

Step 3.5. In this step, we want to show

$$\varepsilon \int_{\mathcal{X}} \mathbb{E}\left[\int_0^{\varepsilon^{2\alpha}} \left|\phi_t^{2,\varepsilon} - \phi_t^{1,\varepsilon}\right|^2 \mathrm{d}t\right] \mathrm{d}x_1 \le C\delta^2\varepsilon^\alpha. \quad (5.113)$$

We first give a generalization of the argument for (5.87). Like before, we define three new processes $\overline{x}_t^{j,\varepsilon}$ for $j = 0, 1, 2$ that start at $\overline{x}_0^{j,\varepsilon} \sim \text{Unif}(\mathcal{X})$ with $\overline{x}_0^{2,\varepsilon} - \overline{x}_0^{1,\varepsilon} \equiv \overline{x}_0^{1,\varepsilon} - \overline{x}_0^{2,\varepsilon} \equiv \delta z$. $\overline{x}_t^{j,\varepsilon}$ follow the same dynamic as $x_t^{j,\varepsilon}$. Again, denote $\overline{\rho}^{j,\varepsilon}(t,x)$ the density for $\overline{x}_t^{j,\varepsilon}$. Then, $\overline{\rho}^{0,\varepsilon}(t,x) = \overline{\rho}^{1,\varepsilon}(t,x) = \overline{\rho}^{2,\varepsilon}(t,x)$ because they share the same distribution. So,

$$
\begin{aligned}
&\int_{\mathcal{X}} \mathbb{E}\left[\int_0^{\varepsilon^{2\alpha}} \left\langle \phi_t^{1,\varepsilon},\, \phi_t^{1,\varepsilon} \right\rangle \mathrm{d}t\right] \mathrm{d}x_1 \\
&\equiv \int_{\mathcal{X}} \mathbb{E}\left[\int_0^{\varepsilon^{2\alpha}} \left\langle \phi(t, x_t^{1,\varepsilon}),\, \phi(t, x_t^{1,\varepsilon}) \right\rangle \mathrm{d}t \;\middle|\; x_0^{1,\varepsilon} = x_1\right] \mathrm{d}x_1 \\
&= \mathbb{E}_{\overline{x}_0^{1,\varepsilon} \sim \text{Unif}(\mathcal{X})} \mathbb{E}\left[\int_0^{\varepsilon^{2\alpha}} \left\langle \phi(t, \overline{x}_t^{1,\varepsilon}),\, \phi(t, \overline{x}_t^{1,\varepsilon}) \right\rangle \mathrm{d}t \;\middle|\; \overline{x}_0^{1,\varepsilon}\right] \\
&= \mathbb{E}\left[\int_0^{\varepsilon^{2\alpha}} \left\langle \phi(t, \overline{x}_t^{1,\varepsilon}),\, \phi(t, \overline{x}_t^{1,\varepsilon}) \right\rangle \mathrm{d}t\right] = \int_{\mathcal{X}} \int_0^{\varepsilon^{2\alpha}} \left\langle \phi(t,x),\, \phi(t,x) \right\rangle \overline{\rho}^{1,\varepsilon}(t,x) \mathrm{d}t\, \mathrm{d}x \\
&= \int_{\mathcal{X}} \int_0^{\varepsilon^{2\alpha}} \left\langle \phi(t,x),\, \phi(t,x) \right\rangle \overline{\rho}^{2,\varepsilon}(t,x) \mathrm{d}t\, \mathrm{d}x = \int_{\mathcal{X}} \mathbb{E}\left[\int_0^{\varepsilon^{2\alpha}} \left\langle \phi_t^{2,\varepsilon},\, \phi_t^{2,\varepsilon} \right\rangle \mathrm{d}t\right] \mathrm{d}x_1.
\end{aligned}
\tag{5.114}
$$

Similarly, since $(\overline{x}_t^{0,\varepsilon}, \overline{x}_t^{1,\varepsilon})$ and $(\overline{x}_t^{1,\varepsilon}, \overline{x}_t^{2,\varepsilon})$ share the same joint distribution, we can show

$$
\int_{\mathcal{X}} \mathbb{E}\left[\int_0^{\varepsilon^{2\alpha}} \left\langle \phi_t^{1,\varepsilon},\, \phi_t^{0,\varepsilon} \right\rangle \mathrm{d}t\right] \mathrm{d}x_1 = \int_{\mathcal{X}} \mathbb{E}\left[\int_0^{\varepsilon^{2\alpha}} \left\langle \phi_t^{2,\varepsilon},\, \phi_t^{1,\varepsilon} \right\rangle \mathrm{d}t\right] \mathrm{d}x_1. \tag{5.115}
$$

Therefore,

$$
\begin{aligned}
&\varepsilon \int_{\mathcal{X}} \mathbb{E}\left[\int_0^{\varepsilon^{2\alpha}} \left|\phi_t^{2,\varepsilon} - \phi_t^{1,\varepsilon}\right|^2 \mathrm{d}t\right] \mathrm{d}x_1 \\
&= \varepsilon \int_{\mathcal{X}} \mathbb{E}\left[\int_0^{\varepsilon^{2\alpha}} \left(\left\langle \phi_t^{2,\varepsilon},\, \phi_t^{2,\varepsilon}\right\rangle + \left\langle \phi_t^{1,\varepsilon},\, \phi_t^{1,\varepsilon}\right\rangle - 2\left\langle \phi_t^{2,\varepsilon},\, \phi_t^{1,\varepsilon}\right\rangle\right) \mathrm{d}t\right] \mathrm{d}x_1 \\
&= \varepsilon \int_{\mathcal{X}} \mathbb{E}\left[\int_0^{\varepsilon^{2\alpha}} \left(2\left\langle \phi_t^{1,\varepsilon},\, \phi_t^{1,\varepsilon}\right\rangle - \left\langle \phi_t^{2,\varepsilon},\, \phi_t^{1,\varepsilon}\right\rangle - \left\langle \phi_t^{1,\varepsilon},\, \phi_t^{0,\varepsilon}\right\rangle\right) \mathrm{d}t\right] \mathrm{d}x_1 \\
&= \int_{\mathcal{X}} \mathbb{E}\left[\int_0^{\varepsilon^{2\alpha}} \left\langle \phi_t^{1,\varepsilon},\, \varepsilon\left(2\phi_t^{1,\varepsilon} - \phi_t^{2,\varepsilon} - \phi_t^{0,\varepsilon}\right)\right\rangle \mathrm{d}t\right] \mathrm{d}x_1 \\
&\leq C \int_{\mathcal{X}} \mathbb{E}\left[\int_0^{\varepsilon^{2\alpha}} \left[\delta^2\varepsilon^{\alpha}\left|\phi_t^{1,\varepsilon}\right|^2 + \frac{1}{\delta^2\varepsilon^{\alpha}}\left(\left|2x_t^{1,\varepsilon} - x_t^{2,\varepsilon} - x_t^{0,\varepsilon}\right|^2 + \left|x_t^{1,\varepsilon} - x_t^{2,\varepsilon}\right|^4\right)\right] \mathrm{d}t\right] \mathrm{d}x_1 \\
&\leq C\left(\delta^2\varepsilon^{\alpha}\rho_1 \left\|\phi\right\|_{L^2}^2 + \delta^{-2}\varepsilon^{2\alpha-\alpha}\delta^4\right) \leq C\delta^2\varepsilon^{\alpha}.
\end{aligned}
$$

We used (5.114) and (5.115) in the second equality above. In the first inequality, we used estimate for ϕ in (5.109) and Cauchy's inequality. The second inequality is because of (5.87) and the Gronwall inequalities (5.100) (5.101). Therefore, (5.113) holds.

Step 3.6. We estimate (V) in this step. Let us pick one dimension $i \leq m$ and use mean value theorem. We get

$$
\begin{aligned}
&\partial_{u_i} r_t^{1,\varepsilon} - \partial_{u_i} r_t^{2,\varepsilon} \equiv \partial_{u_i} r(x_t^{1,\varepsilon}, u_t^{1,\varepsilon}) - \partial_{u_i} r(x_t^{2,\varepsilon}, u_t^{2,\varepsilon}) \\
&= \left\langle \nabla_{x,u}\partial_{u_i} r\left((1-c)x_t^{1,\varepsilon} + cx_t^{2,\varepsilon}, (1-c)u_t^{1,\varepsilon} + cu_t^{2,\varepsilon}\right),\, (x_t^{1,\varepsilon} - x_t^{2,\varepsilon}, u_t^{1,\varepsilon} - u_t^{2,\varepsilon})\right\rangle
\end{aligned}
$$

for some $c \in [0,1]$. Therefore,

$$
\begin{aligned}
&\left|\left(\partial_{u_i} r_t^{1,\varepsilon} - \partial_{u_i} r_t^{2,\varepsilon}\right) - \left(\partial_{u_i} r_t^{1,*} - \partial_{u_i} r_t^{2,*}\right)\right| \\
&= \left|\left\langle \nabla_{x,u}\partial_{u_i} r\left((1-c)x_t^{1,\varepsilon} + cx_t^{2,\varepsilon}, (1-c)u_t^{1,\varepsilon} + cu_t^{2,\varepsilon}\right), (x_t^{1,\varepsilon} - x_t^{2,\varepsilon}, u_t^{1,\varepsilon} - u_t^{2,\varepsilon})\right\rangle\right. \\
&\quad \left. - \left\langle \nabla_{x,u}\partial_{u_i} r\left((1-c')x_t^{1,\varepsilon} + c'x_t^{2,\varepsilon}, (1-c')u_t^{1,*} + c'u_t^{2,*}\right), (x_t^{1,\varepsilon} - x_t^{2,\varepsilon}, u_t^{1,*} - u_t^{2,*})\right\rangle\right| \\
&\leq L\left(\left|x_t^{1,\varepsilon} - x_t^{2,\varepsilon}\right| + \left|u_t^{1,\varepsilon} - u_t^{2,\varepsilon}\right| + \left|u_t^{1,\varepsilon} - u_t^{1,*}\right| + \left|u_t^{2,\varepsilon} - u_t^{2,*}\right|\right) \cdot \\
&\quad \left(\left|x_t^{1,\varepsilon} - x_t^{2,\varepsilon}\right| + \left|u_t^{1,\varepsilon} - u_t^{2,\varepsilon}\right|\right) + K\left|\left(u_t^{1,\varepsilon} - u_t^{2,\varepsilon}\right) - \left(u_t^{1,*} - u_t^{2,*}\right)\right| \\
&\leq L\left[(L+1)\left|x_t^{1,\varepsilon} - x_t^{2,\varepsilon}\right| + \varepsilon\left(\left|\phi_t^{1,\varepsilon}\right| + \left|\phi_t^{2,\varepsilon}\right|\right)\right](L+1)\left|x_t^{1,\varepsilon} - x_t^{2,\varepsilon}\right| + K\varepsilon\left|\phi_t^{1,\varepsilon} - \phi_t^{2,\varepsilon}\right| \\
&\leq C\left(\left|x_t^{1,\varepsilon} - x_t^{2,\varepsilon}\right|^2 + \varepsilon\left|x_t^{1,\varepsilon} - x_t^{2,\varepsilon}\right|\left(\left|\phi_t^{1,\varepsilon}\right| + \left|\phi_t^{2,\varepsilon}\right|\right) + \varepsilon\left|\phi_t^{1,\varepsilon} - \phi_t^{2,\varepsilon}\right|\right).
\end{aligned}
$$

In the first equality, we apply the mean value theorem twice, with $c, c' \in [0,1]$. In the first inequality, we used:

$$
|\langle a_1, b_1\rangle - \langle a_2, b_2\rangle| \leq |a_1 - a_2|\,|b_1| + |a_2|\,|b_1 - b_2|,
$$

and the Lipschitz and boundedness property (in Assumption 2) of the derivatives for r. In the second inequality, we use $u^\varepsilon = u^* + \varepsilon\phi$ and the Lipschitz property of u^ε. Applying the same argument in all the dimensions, we get

$$
\begin{aligned}
&\left|\left(\nabla_u r_t^{1,\varepsilon} - \nabla_u r_t^{1,*}\right) - \left(\nabla_u r_t^{2,\varepsilon} - \nabla_u r_t^{2,*}\right)\right| \\
&\leq C\left(\left|x_t^{1,\varepsilon} - x_t^{2,\varepsilon}\right|^2 + \varepsilon\left|x_t^{1,\varepsilon} - x_t^{2,\varepsilon}\right|\left(\left|\phi_t^{1,\varepsilon}\right| + \left|\phi_t^{2,\varepsilon}\right|\right) + \varepsilon\left|\phi_t^{1,\varepsilon} - \phi_t^{2,\varepsilon}\right|\right).
\end{aligned}
$$

Therefore

$$
\begin{aligned}
(V) &= \int_{\mathcal{X}} \mathbb{E}\left[\int_0^{\varepsilon^{2\alpha}} \left|\left\langle\left(\nabla_u r_t^{1,\varepsilon} - \nabla_u r_t^{1,*}\right) - \left(\nabla_u r_t^{2,\varepsilon} - \nabla_u r_t^{2,*}\right), \phi_t^{2,\varepsilon} - \phi_t^{1,\varepsilon}\right\rangle\right| \mathrm{d}t\right] \mathrm{d}x_1 \\
&\leq C\int_{\mathcal{X}} \mathbb{E}\left[\int_0^{\varepsilon^{2\alpha}} \left(\left|x_t^{1,\varepsilon} - x_t^{2,\varepsilon}\right|^2 + \varepsilon\left|x_t^{1,\varepsilon} - x_t^{2,\varepsilon}\right|\left(\left|\phi_t^{1,\varepsilon}\right| + \left|\phi_t^{2,\varepsilon}\right|\right)\right.\right. \\
&\qquad \left.\left. + \varepsilon\left|\phi_t^{1,\varepsilon} - \phi_t^{2,\varepsilon}\right|\right)\left|\phi_t^{2,\varepsilon} - \phi_t^{1,\varepsilon}\right| \mathrm{d}t\right] \mathrm{d}x_1.
\end{aligned}
$$

Continuing the analysis, we get

$$
\begin{aligned}
(V) \leq C \int_{\mathcal{X}} \mathbb{E}\Bigg[\int_0^{\varepsilon^{2\alpha}} \Bigg(\frac{1}{\varepsilon}\left|x_t^{1,\varepsilon} - x_t^{2,\varepsilon}\right|^3 + \delta^2\varepsilon^{\alpha}\left(\left|\phi_t^{1,\varepsilon}\right|^2 + \left|\phi_t^{2,\varepsilon}\right|^2\right) \\
+\frac{1}{\delta^2\varepsilon^{\alpha}}\left|x_t^{1,\varepsilon} - x_t^{2,\varepsilon}\right|^4 + \varepsilon\left|\phi_t^{2,\varepsilon} - \phi_t^{1,\varepsilon}\right|^2\Bigg)\mathrm{d}t\Bigg]\mathrm{d}x_1 \\
\leq C\left(C\varepsilon^{2\alpha-1}\delta^3 + 2\delta^2\varepsilon^{\alpha}\rho_1\left\|\phi\right\|_{L^2}^2 + C\varepsilon^{2\alpha-\alpha}\delta^{-2+4} + \delta^2\varepsilon^{\alpha}\right) \leq C\delta^2\varepsilon^{\alpha}.
\end{aligned}
\tag{5.116}
$$

In the first inequality, we used the Lipschitz condition

$$
\left|\phi_t^{1,\varepsilon} - \phi_t^{2,\varepsilon}\right| \leq \frac{2L}{\varepsilon}\left|x_t^{1,\varepsilon} - x_t^{2,\varepsilon}\right|
$$

and Cauchy's inequality. In the second inequality, we used Gronwall inequality of order 3, (5.87), Gronwall inequality of order 4 (5.100), and (5.113) in *Step 3.5*. We did not prove Gronwall inequality of order 3, but it can be obtained directly from the order 2 (5.34) and order 4 (5.100) inequalities using Cauchy's inequality. Therefore, we finished estimation of (V).

Step 3.7. We estimate (VI) and (VII) in this step. Recall the definition

$$
\begin{aligned}
(VI) = \int_{\mathcal{X}} \mathbb{E}\Bigg[\int_0^{\varepsilon^{2\alpha}} \Bigg|\Big\langle\Big(\nabla_u b_t^{1,\varepsilon\top}\nabla_x V_t^{1,\varepsilon} - \nabla_u b_t^{1,*\top}\nabla_x V_t^{1,*}\Big) \\
-\Big(\nabla_u b_t^{2,\varepsilon\top}\nabla_x V_t^{2,\varepsilon} - \nabla_u b_t^{2,*\top}\nabla_x V_t^{2,*}\Big), \phi_t^{2,\varepsilon} - \phi_t^{1,\varepsilon}\Big\rangle\Bigg|\mathrm{d}t\Bigg]\mathrm{d}x_1.
\end{aligned}
$$

We further decompose (VI) into two parts. A simple triangle inequality gives us

$$
(VI) \leq (VIII) + (IX),
$$

where

$$
\begin{aligned}
(VIII) := \int_{\mathcal{X}} \mathbb{E}\Bigg[\int_0^{\varepsilon^{2\alpha}} \Bigg|\Big\langle\Big(\nabla_u b_t^{1,\varepsilon\top}\nabla_x V_t^{1,\varepsilon} - \nabla_u b_t^{1,*\top}\nabla_x V_t^{1,\varepsilon}\Big) \\
-\Big(\nabla_u b_t^{2,\varepsilon\top}\nabla_x V_t^{2,\varepsilon} - \nabla_u b_t^{2,*\top}\nabla_x V_t^{2,\varepsilon}\Big), \phi_t^{2,\varepsilon} - \phi_t^{1,\varepsilon}\Big\rangle\Bigg|\mathrm{d}t\Bigg]\mathrm{d}x_1.
\end{aligned}
\tag{5.117}
$$

and

$$(IX) := \int_{\mathcal{X}} \mathbb{E}\left[\int_0^{\varepsilon^{2\alpha}} \Big|\Big\langle \left(\nabla_u b_t^{1,*\top}\nabla_x V_t^{1,\varepsilon} - \nabla_u b_t^{1,*\top}\nabla_x V_t^{1,*}\right) - \left(\nabla_u b_t^{2,*\top}\nabla_x V_t^{2,\varepsilon} - \nabla_u b_t^{2,*\top}\nabla_x V_t^{2,*}\right), \phi_t^{2,\varepsilon} - \phi_t^{1,\varepsilon}\Big\rangle\Big| \,\mathrm{d}t\right] \mathrm{d}x_1. \tag{5.118}$$

The analysis for $(VIII)$ is exactly the same as the analysis for (V) in *Step 3.6*, except that the function $\nabla_u r$ is replaced by $\nabla_u b^\top \nabla_x V^\varepsilon$. By Assumption 2, and the regularity for the value functions, $\nabla_u b^\top \nabla_x V^\varepsilon$ and $\nabla_u r$ share the same properties that are necessary to prove (5.116). Therefore, we can show

$$(VIII) \le C\delta^2 \varepsilon^\alpha. \tag{5.119}$$

We consider (IX) next. We want to show

$$(IX) = \int_{\mathcal{X}} \mathbb{E}\left[\int_0^{\varepsilon^{2\alpha}} \Big|\Big\langle \nabla_u b_t^{1,*\top}\left(\nabla_x V_t^{1,\varepsilon} - \nabla_x V_t^{1,*}\right) - \nabla_u b_t^{2,*\top}\left(\nabla_x V_t^{2,\varepsilon} - \nabla_x V_t^{2,*}\right), \phi_t^{2,\varepsilon} - \phi_t^{1,\varepsilon}\Big\rangle\Big| \,\mathrm{d}t\right] \mathrm{d}x_1 \le C\delta^2\varepsilon^\alpha. \tag{5.120}$$

Again, we consider one single dimension. We pick $i \le m$, $l \le n$ and denote $b_t^{j,*,l}$ the l-th entry of $b_t^{j,*}$ for $j = 0, 1, 2$. We have

$$\begin{aligned} &\left|\partial_{u_i} b_t^{1,*,l}\left(\partial_{x_l} V_t^{1,\varepsilon} - \partial_{x_l} V_t^{1,*}\right) - \partial_{u_i} b_t^{2,*,l}\left(\partial_{x_l} V_t^{2,\varepsilon} - \partial_{x_l} V_t^{2,*}\right)\right| \\ \le &\left|\left(\partial_{u_i} b_t^{1,*,l} - \partial_{u_i} b_t^{2,*,l}\right)\left(\partial_{x_l} V_t^{1,\varepsilon} - \partial_{x_l} V_t^{1,*}\right)\right| \\ &+ \left|\partial_{u_i} b_t^{2,*,l}\left[\left(\partial_{x_l} V_t^{1,\varepsilon} - \partial_{x_l} V_t^{1,*}\right) - \left(\partial_{x_l} V_t^{2,\varepsilon} - \partial_{x_l} V_t^{2,*}\right)\right]\right|. \end{aligned} \tag{5.121}$$

We estimate the two terms in (5.121) next. For the first, we have

$$\left|\left(\partial_{u_i} b_t^{1,*,l} - \partial_{u_i} b_t^{2,*,l}\right)\left(\partial_{x_l} V_t^{1,\varepsilon} - \partial_{x_l} V_t^{1,*}\right)\right| \le L\left|x_t^{1,\varepsilon} - x_t^{2,\varepsilon}\right|\left|\nabla_x V_t^{1,\varepsilon} - \nabla_x V_t^{1,*}\right| \tag{5.122}$$

because of Assumption 2. For the second, we use the technique in *Step 3.6*. We have

$$\begin{aligned}
&\left|\left(\partial_{x_l} V_t^{1,\varepsilon} - \partial_{x_l} V_t^{2,\varepsilon}\right) - \left(\partial_{x_l} V_t^{1,*} - \partial_{x_l} V_t^{2,*}\right)\right| \\
&= \left|\left\langle \nabla_x \partial_{x_l} V^{\varepsilon}\left(t, (1-c)x_t^{1,\varepsilon} + cx_t^{2,\varepsilon}\right), x_t^{1,\varepsilon} - x_t^{2,\varepsilon}\right\rangle\right. \\
&\quad \left. - \left\langle \nabla_x \partial_{x_l} V^{*}\left(t, (1-c')x_t^{1,\varepsilon} + c'x_t^{2,\varepsilon}\right), x_t^{1,\varepsilon} - x_t^{2,\varepsilon}\right\rangle\right| \\
&\leq \left|\nabla_x \partial_{x_l} V^{\varepsilon}\left(t, (1-c)x_t^{1,\varepsilon} + cx_t^{2,\varepsilon}\right) - \nabla_x \partial_{x_l} V^{*}\left(t, (1-c')x_t^{1,\varepsilon} + c'x_t^{2,\varepsilon}\right)\right| \left|x_t^{1,\varepsilon} - x_t^{2,\varepsilon}\right| \\
&\leq \left(\left|\nabla_x \partial_{x_l} V_t^{1,\varepsilon} - \nabla_x \partial_{x_l} V_t^{1,*}\right| + L(c+c')\left|x_t^{1,\varepsilon} - x_t^{2,\varepsilon}\right|\right)\left|x_t^{1,\varepsilon} - x_t^{2,\varepsilon}\right|,
\end{aligned} \tag{5.123}$$

where we have consecutively used (two) mean value theorems, Cauchy's inequality, and the Lipschitz property for the derivatives of V^{ε} and V^{*}. Combining (5.122), (5.123), and $|\nabla_u b| \leq K$ into (5.121), and repeat the same argument in all the dimensions, we obtain

$$\begin{aligned}
&\left|\nabla_u b_t^{1,*\top}\left(\nabla_x V_t^{1,\varepsilon} - \nabla_x V_t^{1,*}\right) - \nabla_u b_t^{2,*\top}\left(\nabla_x V_t^{2,\varepsilon} - \nabla_x V_t^{2,*}\right)\right| \\
&\leq C\left|x_t^{1,\varepsilon} - x_t^{2,\varepsilon}\right|\left(\left|\nabla_x V_t^{1,\varepsilon} - \nabla_x V_t^{1,*}\right| + \left|\nabla_x^2 V_t^{1,\varepsilon} - \nabla_x^2 V_t^{1,*}\right| + \left|x_t^{1,\varepsilon} - x_t^{2,\varepsilon}\right|\right).
\end{aligned} \tag{5.124}$$

Therefore,

$$\begin{aligned}
(IX) &\leq \int_{\mathcal{X}} \mathbb{E}\left[\int_0^{\varepsilon^{2\alpha}} \left|\nabla_u b_t^{1,*\top}\left(\nabla_x V_t^{1,\varepsilon} - \nabla_x V_t^{1,*}\right) - \nabla_u b_t^{2,*\top}\left(\nabla_x V_t^{2,\varepsilon} - \nabla_x V_t^{2,*}\right)\right|\right. \\
&\qquad \left.\left|\phi_t^{2,\varepsilon} - \phi_t^{1,\varepsilon}\right| \mathrm{d}t\right] \mathrm{d}x_1 \\
&\leq C\int_{\mathcal{X}} \mathbb{E}\left[\int_0^{\varepsilon^{2\alpha}} \left|x_t^{1,\varepsilon} - x_t^{2,\varepsilon}\right|\left(\left|\nabla_x V_t^{1,\varepsilon} - \nabla_x V_t^{1,*}\right| + \left|\nabla_x^2 V_t^{1,\varepsilon} - \nabla_x^2 V_t^{1,*}\right|\right.\right. \\
&\qquad \left.\left. + \left|x_t^{1,\varepsilon} - x_t^{2,\varepsilon}\right|\right)\frac{2L}{\varepsilon}\left|x_t^{1,\varepsilon} - x_t^{2,\varepsilon}\right| \mathrm{d}t\right] \mathrm{d}x_1,
\end{aligned}$$

where we used (5.124) and the Lipschitz condition for ϕ

$$\left|\phi_t^{1,\varepsilon} - \phi_t^{2,\varepsilon}\right| \leq \frac{2L}{\varepsilon}\left|x_t^{1,\varepsilon} - x_t^{2,\varepsilon}\right|.$$

Continuing the analysis, we get

$$(IX) \leq C \int_{\mathcal{X}} \mathbb{E}\left[\int_0^{\varepsilon^{2\alpha}} \left|x_t^{1,\varepsilon} - x_t^{2,\varepsilon}\right|^2 \frac{1}{\varepsilon}\left(\left|\nabla_x V_t^{1,\varepsilon} - \nabla_x V_t^{1,*}\right| + \left|\nabla_x^2 V_t^{1,\varepsilon} - \nabla_x^2 V_t^{1,*}\right|\right)\right.$$
$$\left. + \frac{1}{\varepsilon}\left|x_t^{1,\varepsilon} - x_t^{2,\varepsilon}\right|^3 \mathrm{d}t\right] \mathrm{d}x_1,$$
$$\leq C \int_{\mathcal{X}} \int_0^{\varepsilon^{2\alpha}} \mathbb{E}\left[\frac{1}{\delta^2\varepsilon^\alpha}\left|x_t^{1,\varepsilon} - x_t^{2,\varepsilon}\right|^4 + \frac{\delta^2\varepsilon^\alpha}{\varepsilon^2}\left|\nabla_x V_t^{1,\varepsilon} - \nabla_x V_t^{1,*}\right|^2\right.$$
$$\left. + \frac{\delta^2\varepsilon^\alpha}{\varepsilon^2}\left|\nabla_x^2 V_t^{1,\varepsilon} - \nabla_x^2 V_t^{1,*}\right|^2 + \frac{1}{\varepsilon}\left|x_t^{1,\varepsilon} - x_t^{2,\varepsilon}\right|^3\right] \mathrm{d}t\mathrm{d}x_1,$$

where we used Cauchy's inequality in the second inequality. Next, using Gronwall inequalities and the analysis for the $\nabla_x V$ and $\nabla_x^2 V$ terms in *Step 2.3*, we have

$$(IX) \leq C \int_{\mathcal{X}} \int_0^{\varepsilon^{2\alpha}} \left(\frac{\delta^4}{\delta^2\varepsilon^\alpha} + \frac{\delta^3}{\varepsilon}\right) \mathrm{d}t\, \mathrm{d}x_1 + C\frac{\delta^2\varepsilon^\alpha}{\varepsilon^2} \left\|V^\varepsilon - V^*\right\|_{(T;H^2)}^2 \leq C\delta^2\varepsilon^\alpha.$$

Note that we used $\delta \leq \varepsilon$ and Lemma 9 in the last inequality. Therefore, (5.120) holds. Combining (5.119) and (5.120), we obtain

$$(VI) = \int_{\mathcal{X}} \mathbb{E}\left[\int_0^{\varepsilon^{2\alpha}} \left|\left\langle \left(\nabla_u b_t^{1,\varepsilon\top} \nabla_x V_t^{1,\varepsilon} - \nabla_u b_t^{1,*\top} \nabla_x V_t^{1,*}\right)\right.\right.$$
$$\left.\left. - \left(\nabla_u b_t^{2,\varepsilon\top} \nabla_x V_t^{2,\varepsilon} - \nabla_u b_t^{2,*\top} \nabla_x V_t^{2,*}\right), \phi_t^{2,\varepsilon} - \phi_t^{1,\varepsilon}\right\rangle\right| \mathrm{d}t\right] \mathrm{d}x_1 \leq C\delta^2\varepsilon^\alpha. \tag{5.125}$$

The analysis for (VII) is the same as the analysis for (VI). We decompose (VII) into two parts like (5.117) and (5.118). The first one can be analyzed using the same technique in *Step 3.6*. The second one be analyzed using the same technique in this

step: considering each dimension separately and combine them. Therefore,

$$(VII) = \int_{\mathcal{X}} \mathbb{E}\left[\int_0^{\varepsilon^{2\alpha}} \left|\left\langle \nabla_u \operatorname{Tr}\left(D_t^{1,\varepsilon}\nabla_x^2 V_t^{1,\varepsilon} - D_t^{1,\varepsilon}\nabla_x^2 V_t^{1,\varepsilon}\right)\right.\right.$$
$$\left.\left. -\nabla_u \operatorname{Tr}\left(D_t^{2,\varepsilon}\nabla_x^2 V_t^{2,\varepsilon} - D_t^{2,\varepsilon}\nabla_x^2 V_t^{2,\varepsilon}\right), \phi_t^{2,\varepsilon} - \phi_t^{1,\varepsilon}\right\rangle\right| \mathrm{d}t\right] \mathrm{d}x_1 \le C\delta^2\varepsilon^{\alpha}. \tag{5.126}$$

Combining (5.116), (5.125), and (5.126) into (5.112), we obtain (5.111). i.e., we get the bound for (III). (IV) can be analyzed in exactly the same way, so we also have

$$(IV) = \int_{\mathcal{X}} \mathbb{E}\left[\int_0^{\varepsilon^{2\alpha}} \left|\left\langle \nabla_u G_t^{0,\varepsilon} - \nabla_u G_t^{1,\varepsilon},\, \phi_t^{0,\varepsilon} - \phi_t^{1,\varepsilon}\right\rangle\right| \mathrm{d}t\right] \mathrm{d}x_1 \le C\delta^2\varepsilon^{\alpha}. \tag{5.127}$$

Combining (5.108), (5.110), (5.111), and (5.127) into (5.106), we obtain (5.97).

Step 3.8. We want to show (5.98)

$$\int_{\varepsilon^{2\alpha}}^{T}\int_{\mathcal{X}} \left|\left\langle \nabla_u G(t,x,u^{\varepsilon},-\nabla_x V^{\varepsilon},-\nabla_x^2 V^{\varepsilon}),\, \phi(t,x)\right\rangle\right| \left|\rho^{0,\varepsilon} + \rho^{2,\varepsilon} - 2\rho^{1,\varepsilon}\right| \mathrm{d}x\,\mathrm{d}t \le C\delta^2\varepsilon^{\alpha}$$

in this step. The spirit and notation are much the same as *Step 2.4*. $\rho^{j,\varepsilon}(t,x) = p^{\varepsilon}(t,x;0,x_j)$ for $j = 0,1,2$, where $p^{\varepsilon}(t,x;s,y)$ is the fundamental solution of the Fokker Planck equation $\partial_t\rho = \mathcal{G}_{\varepsilon}^{\dagger}\rho$. $q^{\varepsilon}(t,x;s,y) := p^{\varepsilon}(s,y;t,x)$ is the fundamental solution of the backward Kolmogorov equation $\partial_t\psi + \mathcal{G}_{\varepsilon}\psi = 0$. Applying the generalized mean value theorem (5.99) and lemma (5.89) with $k = 2$ to q^{ε}, we obtain

$$\begin{aligned} &\left|\rho^{0,\varepsilon}(t,x) - \rho^{2,\varepsilon}(t,x) - 2\rho^{1,\varepsilon}(t,x)\right| \\ &= \left|q^{\varepsilon}(0,x_0;t,x) + q^{\varepsilon}(0,x_2;t,x) - 2q^{\varepsilon}(0,x_1;t,x)\right| \\ &= \left|(x_2-x_1)^{\top}\nabla_x^2 q^{\varepsilon}(0,(1-c)x_1 + cx_2;t,x)\,(x_2-x_1)\right| \\ &\le Ct^{-(2+n)/2}\left|x_1-x_2\right|^2 = Ct^{-(2+n)/2}\delta^2. \end{aligned} \tag{5.128}$$

Therefore,

$$\int_{\varepsilon^{2\alpha}}^{T}\int_{\mathcal{X}} \left|\left\langle \nabla_u G(t,x,u^\varepsilon,-\nabla_x V^\varepsilon,-\nabla_x^2 V^\varepsilon),\, \phi(t,x)\right\rangle\right| \left|\rho^{0,\varepsilon}+\rho^{2,\varepsilon}-2\rho^{1,\varepsilon}\right| \mathrm{d}x\,\mathrm{d}t$$

$$\leq C\int_{\varepsilon^{2\alpha}}^{T}\int_{\mathcal{X}} \left(\varepsilon\left|\phi(t,x)\right| + \left|\nabla_x V^\varepsilon - \nabla_x V^*\right| + \left|\nabla_x^2 V^\varepsilon - \nabla_x^2 V^*\right|\right)$$

$$\left|\phi(t,x)\right| t^{-(2+n)/2}\delta^2 \,\mathrm{d}x\,\mathrm{d}t \leq C\delta^2\varepsilon^{-(2+n)\alpha}$$

$$\int_{\varepsilon^{2\alpha}}^{T}\int_{\mathcal{X}} \left(\varepsilon\left|\phi(t,x)\right|^2 + \frac{1}{\varepsilon}\left|\nabla_x V^\varepsilon - \nabla_x V^*\right|^2 + \frac{1}{\varepsilon}\left|\nabla_x^2 V^\varepsilon - \nabla_x^2 V^*\right|^2\right) \mathrm{d}x\,\mathrm{d}t$$

$$\leq C\delta^2\left(\varepsilon^{1-(2+n)\alpha}\left\|\phi\right\|_{L^2}^2 + \varepsilon^{-1-(2+n)\alpha}\left\|V^\varepsilon - V^*\right\|_{(T;H^2)}^2\right) \leq C\delta^2\varepsilon^{1-(2+n)\alpha} = C\delta^2\varepsilon^{\alpha}.$$

In the first inequality, we used (5.76) and (5.128). In the second inequality, we extract a constant $\delta^2\varepsilon^{-(2+n)\alpha} \geq \delta^2 t^{-(2+n)/2}$ and use Cauchy's inequality. In the last inequality, we used Lemma 9. Therefore, (5.98) holds. To conclude, combining (5.97) and (5.98), we recover (5.93), which implies (5.92). Hence, (5.91) holds, and we finish *Step 3*.

Finally, combining the three steps (5.78), (5.79), and (5.91), we get (5.77) and finish proving Lemma 11. □

We recall the definition of G for the reader convenience:

$$G(t,x,u,p,P) = \mathrm{Tr}\left(P\,D(x,u)\right) + \left\langle p,\, b(x,u)\right\rangle - r(x,u).$$

For a control function $u(t,x)$, we define a corresponding new function through

$$u^\diamond(t,x) := \arg\max_{u\in\mathbb{R}^m} G(t,x,u,-\nabla_x V_u(t,x),-\nabla_x^2 V_u(t,x)). \tag{5.129}$$

This $u^\diamond$ is well-defined because G is strongly concave in u. By the uniqueness of the solution to the HJB equation, $u = u^\diamond$ if and only if u is the optimal control. Also,

using the μ_G-strong concavity, we have

$$\begin{aligned}
&\left|\nabla_u G(t,x,u(t,x),-\nabla_x V_u,-\nabla_x^2 V_u)\right| \\
&= \left|\nabla_u G(t,x,u(t,x),-\nabla_x V_u,-\nabla_x^2 V_u) - \nabla_u G(t,x,u^\diamond(t,x),-\nabla_x V_u,-\nabla_x^2 V_u)\right| \\
&\geq \mu_G \left|u(t,x)-u^\diamond(t,x)\right|
\end{aligned} \tag{5.130}$$

We give the following lemma regarding this implicit function

Lemma 12 (Lipschitz condition of the implicit function induced by G)**.** *Let Assumption 2 holds. Then there exists a constant $C_5 > 0$ such that for any two control functions $u_1, u_2 \in \mathcal{U}$, and any $(t,x) \in [0,T] \times \mathcal{X}$,*

$$\begin{aligned}
&\left|u_1^\diamond(t,x) - u_2^\diamond(t,x)\right| \\
&\leq C_5 \left(\left|\nabla_x V_{u_1}(t,x) - \nabla_x V_{u_2}(t,x)\right| + \left|\nabla_x^2 V_{u_1}(t,x) - \nabla_x^2 V_{u_2}(t,x)\right|\right).
\end{aligned} \tag{5.131}$$

Proof. By strong concavity of G in u, $u^\diamond(t,x)$ in (5.129) is given by the equation

$$\nabla_u G(t,x,u^\diamond(t,x),-\nabla_x V_u(t,x),-\nabla_x^2 V_u(t,x)) = 0.$$

Therefore, for fixed (t,x), we can view $u^\diamond = u^\diamond(t,x)$ as an implicit function of $p = -\nabla_x V_u(t,x)$ and $P = -\nabla_x^2 V_u(t,x)$. So, (5.131) is nothing but the Lipschitz condition of this implicit function. Therefore, it is sufficient to show the boundedness of the Jacobian of this implicit function. Compute the Jacobian of $\nabla_u G(t,x,u^\diamond,p,P) = 0$ w.r.t. $(p,P) \in \mathbb{R}^{n+n^2}$, we obtain

$$0 = \nabla_u^2 G(t,x,u^\diamond,p,P) \cdot \frac{\partial u^\diamond}{\partial(p,P)} + (\nabla_u b(x,u),\ \nabla_u D(x,u)).$$

So

$$\frac{\partial u^\diamond}{\partial(p,P)} = -\left(\nabla_u^2 G(t,x,u^\diamond,p,P)\right)^{-1} (\nabla_u b(x,u),\ \nabla_u D(x,u)). \tag{5.132}$$

By assumption 2, $|\nabla_u b(x,u)| \leq K$ and $|\nabla_u D(x,u)| \leq K$. Since G is μ_G-strongly concave in u, $-(\nabla_u^2 G)^{-1}$ is positive definite with spectrum norm less than $1/\mu_G$. Therefore, (5.132) implies

$$\left|\frac{\partial u^\diamond}{\partial(p,P)}\right| \leq \frac{2K}{\mu_G}.$$

So (5.131) holds. □

5.8 Proof for the theorems

Now, we are ready to prove Theorem 3 and 4.

Proof of Theorem 3 and 4. Case 1. We firstly consider an easy case where there exists positive constants μ and τ_0 such that

$$\|u^\tau - u^{\tau\diamond}\|_{L^2} \geq \mu \|u^\tau - u^*\|_{L^2} \tag{5.133}$$

for all $\tau \geq \tau_0$. Under such condition, we have

$$\begin{aligned}
&\frac{\mathrm{d}}{\mathrm{d}\tau} J[u^\tau] = \left\langle \frac{\delta J}{\delta u}[u^\tau],\, \frac{\mathrm{d}}{\mathrm{d}\tau} u^\tau \right\rangle \\
&= -\left\| \rho^{u^\tau}(t,x) \nabla_u G(t,x,u^\tau(t,x), -\nabla_x V_{u^\tau}, -\nabla_x^2 V_{u^\tau}) \right\|_{L^2}^2 \\
&\leq -\rho_0^2 \left\| \nabla_u G(t,x,u^\tau(t,x), -\nabla_x V_{u^\tau}, -\nabla_x^2 V_{u^\tau}) \right\|_{L^2}^2 \leq -\rho_0^2 \mu_G^2 \|u^\tau - u^{\tau\diamond}\|_{L^2}^2 \\
&\leq -\rho_0^2 \mu_G^2 \mu^2 \|u^\tau - u^*\|_{L^2}^2 \leq -\rho_0^2 \mu_G^2 \mu^2 \frac{1}{C_3} \left(J[u^\tau] - J[u^*]\right),
\end{aligned} \tag{5.134}$$

where we have consecutively used: chain rule; proposition 4 and the control dynamic (5.16); proposition 6; inequality (5.130); assumption (5.133); and Lemma 10 respectively. Equation (5.134) implies

$$J[u^\tau] - J[u^*] \leq e^{-c(\tau-\tau_0)} \left(J[u^{\tau_0}] - J[u^*]\right)$$

holds with $c = \rho_0^2 \mu_G^2 \mu^2 \frac{1}{C_2}$. So (5.18) holds. Therefore, the two theorems hold under this easy case.

Case 2. Next, we focus on the harder case when (5.133) does not hold. Then we can find a sequence $\{\tau_k\}$, increasing to infinity, such that

$$\|u^{\tau_k} - u^{\tau_k \diamond}\|_{L^2} \leq \frac{1}{k} \|u^{\tau_k} - u^*\|_{L^2} .$$

For notational simplicity, we denote u^{τ_k} by u_k and the corresponding value function $V_{u^{\tau_k}}$ by V_k. So we have

$$\|u_k - u_k^\diamond\|_{L^2} \leq \frac{1}{k} \|u_k - u^*\|_{L^2} . \tag{5.135}$$

By Proposition 5, the value function $V_{u^\tau}(t,x)$ is decreasing in τ, so it has a pointwise limit $V_\infty(t,x)$. Since $V_{u^\tau}(t,x) \geq V^*(t,x)$, we have $V_\infty(t,x) \geq V^*(t,x)$. We claim that

$$V_\infty(t,x) \equiv V^*(t,x). \tag{5.136}$$

The proof of this claim is quite long and technical, so we leave it to the next lemma 13 and focus on the rest of the proof first. With the claim holds, we know that $V_{u^\tau}(0,\cdot)$ converges to $V^*(0,\cdot)$ uniformly using the Lipschitz condition and Arzelá–Ascoli theorem. Therefore, using the relationship

$$J[u^\tau] = \int_{\mathcal{X}} \rho^{u^\tau}(0,x) V_{u^\tau}(0,x)\,\mathrm{d}x = \int_{\mathcal{X}} V_{u^\tau}(0,x)\,\mathrm{d}x \quad \text{and} \quad J[u^*] = \int_{\mathcal{X}} V^*(0,x)\,\mathrm{d}x,$$

we can show (5.17) and thus confirm Theorem 3.

Next, we show the convergence rate in Theorem 4. By Assumption 4,

$$\lim_{\tau\to\infty} \|u^\tau - u^*\|_{L^2} = 0,$$

hence $\lim_{k\to\infty} \|u_k - u^*\|_{L^2} = 0$. By Lemma 12, we have

$$|u_k^\diamond(t,x) - u^*(t,x)| \leq C_5 \left(|\nabla_x V_k(t,x) - \nabla_x V^*(t,x)| + \left|\nabla_x^2 V_k(t,x) - \nabla_x^2 V^*(t,x)\right| \right),$$

hence

$$\|u_k^\diamond - u^*\|_{L^2}^2 \leq 2C_5^2 \left(\|\nabla_x V_k - \nabla_x V^*\|_{L^2}^2 + \left\|\nabla_x^2 V_k - \nabla_x^2 V^*\right\|_{L^2}^2 \right).$$

Therefore, by lemma 11, we have

$$\|u_k^\diamond - u^*\|_{L^2} \leq \sqrt{2} C_5 \|V_k - V^*\|_{(T;H^2)} \leq \sqrt{2} C_5 C_4 \|u_k - u^*\|_{L^2}^{1+\alpha}.$$

Therefore, we obtain

$$\|u_k - u^*\|_{L^2} \leq \|u_k - u_k^\diamond\|_{L^2} + \|u_k^\diamond - u^*\|_{L^2} \leq \frac{1}{k} \|u_k - u^*\|_{L^2} + C \|u_k - u^*\|_{L^2}^{1+\alpha}.$$

However, this cannot hold when k is sufficiently large (i.e., when $\|u_k - u^*\|_{L^2}$ is sufficiently small) because we assume (5.135). Therefore, the assumption (5.135) cannot hold under Assumption 4. Hence (5.133) must hold and Theorem 4 is proved.

□

Lemma 13 (claim (5.136))**.** *Under assumption* (5.135) *and all the assumptions in theorem 3,* (5.136) *holds.*

Proof. We assume to the contrary that there exists $(\bar{t}, \bar{x}) \in [0, T] \times \mathcal{X}$ s.t. $V_\infty(\bar{t}, \bar{x}) - V^*(\bar{t}, \bar{x}) \geq \eta > 0$. This implies that

$$V_k(\bar{t}, \bar{x}) - V^*(\bar{t}, \bar{x}) \geq \eta > 0 \quad \forall k. \tag{5.137}$$

By the Arzelá–Ascoli theorem, V_k converges to V_∞ uniformly and V_∞ (hence $V_\infty - V^*$) is continuous. So, we can assume $\bar{t} > 0$. For any ε, δ, β, $\lambda \in (0, 1)$, we define two continuous functions on $(0, T] \times \mathcal{X} \times (0, T] \times \mathcal{X}$

$$\varphi(t, x, s, y) := \frac{1}{2\varepsilon} |t - s|^2 + \frac{1}{2\delta} |x - y|^2 - \beta(t + s) + \frac{\lambda}{t} + \frac{\lambda}{s} \tag{5.138}$$

and

$$\Phi_k(t, x, s, y) := V_k(t, x) - V^*(s, y) - \varphi(t, x, s, y). \tag{5.139}$$

Since the domain of Φ_k is bounded and $\lim_{t\wedge s\to 0+}\Phi_k(t,x,s,y)=-\infty$, $\Phi_k(t,x,s,y)$ achieves its maximum at some point $(t_k,x_k,s_k,y_k)\in(0,T]\times\mathcal{X}\times(0,T]\times\mathcal{X}$. Note that (t_k,x_k,s_k,y_k) depends on ε, δ, β, λ, and k. Using the inequality

$$2\Phi_k(t_k,x_k,s_k,y_k)\geq\Phi_k(t_k,x_k,t_k,x_k)+\Phi_k(s_k,y_k,s_k,y_k),$$

we obtain

$$\begin{aligned}\frac{1}{\varepsilon}\left|t_k-s_k\right|^2+\frac{1}{\delta}\left|x_k-y_k\right|^2&\leq V_k(t_k,x_k)-V_k(s_k,y_k)+V^*(t_k,x_k)-V^*(s_k,y_k)\\&\leq 2L\left|(t_k,x_k)-(s_k,y_k)\right|,\end{aligned}$$

where we used the Lipschitz condition of V^* and V_k in the second inequality. Therefore,

$$\frac{1}{\varepsilon+\delta}\left|(t_k,x_k)-(s_k,y_k)\right|^2=\frac{1}{\varepsilon+\delta}\left(\left|t_k-s_k\right|^2+\left|x_k-y_k\right|^2\right)\leq 2L\left|(t_k,x_k)-(s_k,y_k)\right|.$$

Hence,

$$\left|(t_k,x_k)-(s_k,y_k)\right|\leq 2L(\varepsilon+\delta)\tag{5.140}$$

and

$$\frac{1}{\varepsilon}\left|t_k-s_k\right|^2+\frac{1}{\delta}\left|x_k-y_k\right|^2\leq 4L^2(\varepsilon+\delta)\tag{5.141}$$

hold. We can also see that $\left|t_k-s_k\right|,\ \left|x_k-y_k\right|\to 0$ as $\varepsilon,\ \delta\to 0$. Another direct result we have is that

$$\begin{aligned}&V_k(\bar{t},\bar{x})-V^*(\bar{t},\bar{x})-\varphi(\bar{t},\bar{x},\bar{t},\bar{x})=\Phi_k(\bar{t},\bar{x},\bar{t},\bar{x})\\&\leq\Phi_k(t_k,x_k,s_k,y_k)=V_k(t_k,x_k)-V^*(s_k,y_k)-\varphi(t_k,x_k,s_k,y_k),\end{aligned}$$

which implies

$$V_k(\bar{t},\bar{x})-V^*(\bar{t},\bar{x})+2\beta\bar{t}-\frac{2\lambda}{\bar{t}}\leq V_k(t_k,x_k)-V^*(s_k,y_k)+\beta(t_k+s_k)-\frac{\lambda}{t_k}-\frac{\lambda}{s_k}.\tag{5.142}$$

Next, we separate into two cases. The idea is that: when t_k or s_k are close to T, we use the fact that $V_\infty(T,\cdot) = V^*(T,\cdot) = h(\cdot)$ to derive a contradiction; when t_k and s_k are not close to T, we use positivity of β and λ to derive a contradiction.

Case 1. For any K_0 and $\alpha > 0$, we can find $k \geq K_0$ and $\beta, \varepsilon, \delta, \lambda < \alpha$ s.t. $t_k \vee s_k \geq T - \frac{\eta}{3L}$. Under this assumption, we can find a sequence $\{k_i, \beta_i, \varepsilon_i, \delta_i, \lambda_i\}_{i=1}^\infty$ s.t. k_i increases to infinity, $(\beta_i, \varepsilon_i, \delta_i, \lambda_i)$ decrease to 0s, and the corresponding $t_{k_i}, x_{k_i}, s_{k_i}, y_{k_i}$ satisfies $t_{k_i} \vee s_{k_i} \geq T - \frac{\eta}{3L}$ for all i. Since $[0,T] \times \mathcal{X}$ is bounded, we can pick a subsequence of this $\{k_i, \beta_i, \varepsilon_i, \delta_i, \lambda_i\}_{i=1}^\infty$ (without changing notations) such that $t_{k_i}, x_{k_i}, s_{k_i}, y_{k_i}$ all converge.

Let $i \to \infty$. Note that (5.140) implies s_{k_i}, t_{k_i} converge to some same limit $t_\infty \geq T - \frac{\eta}{3L}$ and x_{k_i}, y_{k_i} converge to some same limit x_∞. So, (5.142) becomes

$$\begin{aligned} &V_\infty(\bar{t},\bar{x}) - V^*(\bar{t},\bar{x}) \leq V_\infty(t_\infty, x_\infty) - V^*(t_\infty, x_\infty) \\ &= V_\infty(t_\infty, x_\infty) - V_\infty(T, x_\infty) + V^*(T, x_\infty) - V^*(t_\infty, x_\infty) \\ &\leq L\,|T - t_\infty| + L\,|T - t_\infty| \leq 2L\frac{\eta}{3L} = \frac{2}{3}\eta < \eta, \end{aligned}$$

which contradicts to (5.137).

Case 2. There exist K_0 and $\alpha > 0$ s.t. for any $k \geq K_0$ and $\beta, \varepsilon, \delta, \lambda < \alpha$, we have $t_k \vee s_k < T - \frac{\eta}{3L}$. In this second case, we will only focus on the situation when $k \geq K_0$ and $\beta, \varepsilon, \delta, \lambda < \alpha$. Without loss of generality, we assume $K_0 \geq 1$ and $\alpha \leq 1$. We fix $\beta < \alpha$ and $\lambda < \alpha$ and let k, ε, δ vary. Define $M := 4K + 2T + 2/\bar{t}$ and $r_0 := \min\{\frac{\lambda}{M(M+1)}, \frac{\eta}{6L}\}$. Note that λ is fixed, so r_0 is an absolute constant. We also define

$$Q_0 := \{(t,x,s,y) \mid \lambda/M \leq t, s \leq T - 2r_0\}.$$

Then Φ_k achieves its maximum (t_k, x_k, s_k, y_k) in Q_0 because (5.142) cannot hold if $t_k \leq \lambda/M$ or $s_k \leq \lambda/M$. Next, we define

$$Q := \{(t,x,s,y) \mid \lambda/(M+1) < t, s < T - r_0\}. \tag{5.143}$$

We find $Q_0 \subset Q$. Also,

$$t_k - \frac{\lambda}{M+1} \geq \frac{\lambda}{M} - \frac{\lambda}{M+1} = \frac{\lambda}{M(M+1)} \geq r_0,$$

and $T - r_0 - t_k \geq r_0$. s_k also satisfies the two inequalities. Restricted in Q, Φ_k has bounded derivatives. Next, we want to show that the maximum is still in Q if we make a small perturbation on Φ_k.

We define $\mu > 0$ by

$$\mu K(K + \sqrt{n} + 1) = \frac{1}{2}\beta + \frac{\lambda}{2T^2}. \tag{5.144}$$

Let $r_1 := \mu r_0/4$. We pick $(q, p, \widehat{q}, \widehat{p}) \in \mathbb{R}^{1+n+1+n}$ s.t. $|p|, |q|, |\widehat{p}|, |\widehat{q}| \leq r_1$. Then we define a new function

$$\begin{aligned}\widehat{\Phi}_k(t, x, s, y) =& \Phi_k(t, x, s, y) - \frac{\mu}{2}\left(|t - t_k|^2 + |x - x_k|^2 + |s - s_k|^2 + |y - y_k|^2\right) \\ &+ q(t - t_k) + \langle p,\, x - x_k\rangle + \widehat{q}(s - s_k) + \langle \widehat{p},\, y - y_k\rangle .\end{aligned} \tag{5.145}$$

If we do not have the second line in (5.145), then $\widehat{\Phi}_k$ achieves a strict maximum at (t_k, x_k, s_k, y_k). This second line can be viewed as a linear perturbation. So, $\widehat{\Phi}_k$ achieves a maximum at some other point in $\mathbb{R}^{1+n+1+n}$, denoted by $(\widehat{t_k}, \widehat{x}_k, \widehat{s}_k, \widehat{y}_k)$. By this optimality, $(\widehat{t_k}, \widehat{x}_k, \widehat{s}_k, \widehat{y}_k)$ must lie in the set

$$\begin{aligned}\Big\{(t, x, s, y) \;\Big|\; &\frac{\mu}{2}\left(|t - t_k|^2 + |x - x_k|^2 + |s - s_k|^2 + |y - y_k|^2\right) \\ &\leq q(t - t_k) + \langle p,\, x - x_k\rangle + \widehat{q}(s - s_k) + \langle \widehat{p},\, y - y_k\rangle\Big\}.\end{aligned}$$

So,

$$\frac{\mu}{2}\left|(\widehat{t_k}, \widehat{x}_k, \widehat{s}_k, \widehat{y}_k) - (t_k, x_k, s_k, y_k)\right|^2 \leq \left|(\widehat{t_k}, \widehat{x}_k, \widehat{s}_k, \widehat{y}_k) - (t_k, x_k, s_k, y_k)\right| \cdot |(q, p, \widehat{q}, \widehat{p})|.$$

Therefore,

$$\left|(\widehat{t_k}, \widehat{x}_k, \widehat{s}_k, \widehat{y}_k) - (t_k, x_k, s_k, y_k)\right| \leq \frac{2}{\mu}\,|(q, p, \widehat{q}, \widehat{p})| \leq \frac{2}{\mu}\; 2r_1 = r_0.$$

So $\left|\widehat{t}_k - t_k\right|, |\widehat{s}_k - s_k| \leq r_0$, which implies $(\widehat{t}_k, \widehat{x}_k, \widehat{s}_k, \widehat{y}_k) \in Q$. More importantly, $(\widehat{t}_k, \widehat{x}_k, \widehat{s}_k, \widehat{y}_k)$ lies in the interior of $(0,T] \times \mathcal{X} \times (0,T] \times \mathcal{X}$. So, by the optimality of $(\widehat{t}_k, \widehat{x}_k, \widehat{s}_k, \widehat{y}_k)$, we have

$$\begin{cases} 0 = \partial_t \widehat{\Phi}_k(\widehat{t}_k, \widehat{x}_k, \widehat{s}_k, \widehat{y}_k) = \partial_t V_k(\widehat{t}_k, \widehat{x}_k) - \partial_t \varphi(\widehat{t}_k, \widehat{x}_k, \widehat{s}_k, \widehat{y}_k) - \mu(\widehat{t}_k - t_k) + q \\ 0 = \partial_s \widehat{\Phi}_k(\widehat{t}_k, \widehat{x}_k, \widehat{s}_k, \widehat{y}_k) = -\partial_s V^*(\widehat{s}_k, \widehat{y}_k) - \partial_s \varphi(\widehat{t}_k, \widehat{x}_k, \widehat{s}_k, \widehat{y}_k) - \mu(\widehat{s}_k - s_k) + \widehat{q} \\ 0 = \nabla_x \widehat{\Phi}_k(\widehat{t}_k, \widehat{x}_k, \widehat{s}_k, \widehat{y}_k) = \nabla_x V_k(\widehat{t}_k, \widehat{x}_k) - \nabla_x \varphi(\widehat{t}_k, \widehat{x}_k, \widehat{s}_k, \widehat{y}_k) - \mu(\widehat{x}_k - x_k) + p \\ 0 = \nabla_y \widehat{\Phi}_k(\widehat{t}_k, \widehat{x}_k, \widehat{s}_k, \widehat{y}_k) = -\nabla_y V^*(\widehat{s}_k, \widehat{y}_k) - \nabla_y \varphi(\widehat{t}_k, \widehat{x}_k, \widehat{s}_k, \widehat{y}_k) - \mu(\widehat{y}_k - y_k) + \widehat{p} \\ \begin{pmatrix} \nabla_x^2 V_k(\widehat{t}_k, \widehat{x}_k) & 0 \\ 0 & -\nabla_y^2 V^*(\widehat{s}_k, \widehat{y}_k) \end{pmatrix} \leq \nabla^2_{x,y}\varphi + \mu I_{2n} = \frac{1}{\delta}\begin{pmatrix} I_n & -I_n \\ -I_n & I_n \end{pmatrix} + \mu I_{2n} \end{cases} \tag{5.146}$$

as first and second order necessary conditions. Note that $(\widehat{t}_k, \widehat{x}_k, \widehat{s}_k, \widehat{y}_k)$ depend on ε, δ, β, λ, q, p, $\widehat{q}$, $\widehat{p}$, μ, and k. Also, recall that β, λ, and μ are fixed.

For given ε, δ and k, we can view $(\widehat{t}_k, \widehat{x}_k, \widehat{s}_k, \widehat{y}_k)$ as an implicit function of $(q, p, \widehat{q}, \widehat{p})$, given by the equation $\nabla\widehat{\Phi}_k(\widehat{t}_k, \widehat{x}_k, \widehat{s}_k, \widehat{y}_k) = 0$, i.e.,

$$(q, p, \widehat{q}, \widehat{p}) = -\nabla\Phi_k(\widehat{t}_k, \widehat{x}_k, \widehat{s}_k, \widehat{y}_k) + \mu(\widehat{t}_k - t_k, \widehat{x}_k - x_k, \widehat{s}_k - s_k, \widehat{y}_k - y_k). \tag{5.147}$$

Here, the gradient is taken w.r.t. (t, x, s, y). We claim that we can consider it inversely and view $(q, p, \widehat{q}, \widehat{p})$ as an implicit function of $(\widehat{t}_k, \widehat{x}_k, \widehat{s}_k, \widehat{y}_k)$ locally, still given by (5.147). The Jacobian of this implicit function is given by

$$A_k := \frac{\partial(q, p, \widehat{q}, \widehat{p})}{\partial(\widehat{t}_k, \widehat{x}_k, \widehat{s}_k, \widehat{y}_k)} = \mu I_{2n+2} - \nabla^2\Phi_k(\widehat{t}_k, \widehat{x}_k, \widehat{s}_k, \widehat{y}_k). \tag{5.148}$$

We will show the claim by proving that A_k is nonsingular locally.

Now we require that $\left|\widehat{t}_k - t_k\right|, |\widehat{x}_k - x_k|, |\widehat{s}_k - s_k|, |\widehat{y}_k - y_k| < r$ where

$$0 < r < \mu/\left[8\left(L + 3(M+1)^4/\lambda^3\right)\right]. \tag{5.149}$$

Later, this r will change according to ε and δ, but will be independent with k. We give an estimate next.

$$\begin{aligned}
&\left|\nabla^2\Phi_k(t_k,x_k,s_k,y_k) - \nabla^2\Phi_k(\widehat{t}_k,\widehat{x}_k,\widehat{s}_k,\widehat{y}_k)\right| \\
&\le \left|\nabla^2 V_k(\widehat{t}_k,\widehat{x}_k) - \nabla^2 V_k(t_k,x_k)\right| + \left|\nabla^2 V^*(\widehat{s}_k,\widehat{y}_k) - \nabla^2 V^*(s_k,y_k)\right| \\
&\quad + 2\lambda\left(\left|\widehat{t}_k^{-3} - t_k^{-3}\right| + \left|\widehat{s}_k^{-3} - s_k^{-3}\right|\right) \\
&\le L\left|(\widehat{t}_k,\widehat{x}_k) - (t_k,x_k)\right| + L\left|(\widehat{s}_k,\widehat{y}_k) - (s_k,y_k)\right| \\
&\quad + 2\lambda\left(3\left|\widehat{t}_k - t_k\right|\left(\min\{\widehat{t}_k,t_k\}\right)^{-4} + 3\left|\widehat{s}_k - s_k\right|\left(\min\{\widehat{s}_k,s_k\}\right)^{-4}\right) \\
&\le 4Lr + 12\lambda r\left(\lambda/(M+1)\right)^{-4} \le \frac{1}{2}\mu.
\end{aligned} \tag{5.150}$$

Here, we used the the definition of φ (5.138) and Φ_k (5.139) in the first inequality. The third inequality is due to the range of $\widehat{t}_k$, $\widehat{s}_k$, t_k, s_k given by (5.143). The fourth inequality comes from the range for r in (5.149). In the second inequality, we used the Lipschitz condition for the derivatives of the value functions and a mean value theorem. Note that the ∇ in (5.150) operates on all the inputs, so we also used the Lipschitz condition of $\partial_t^2 V_k(t,x)$ (and $\partial_s^2 V^*(s,y)$), which was not mentioned before (but is easy to show). If we take derivative of the HJ equation (5.8) w.r.t. t, we get

$$\begin{aligned}
\partial_t^2 V_k(t,x) = &-\partial_t \operatorname{Tr}\left(D(x,u(t,x))\nabla_x^2 V_u(t,x)\right) \\
&- \partial_t\left\langle b(x,u(t,x)),\, \nabla_x V_u(t,x)\right\rangle - \partial_t r(x,u(t,x)).
\end{aligned} \tag{5.151}$$

Expanding the right hand side of (5.151) with chain rule and product rule, we find that each term is bounded and Lipschitz in t and x, so $\partial_t^2 V_k(t,x)$ (and $\partial_s^2 V^*(s,y)$) is Lipschitz. We also remark that this part (5.150) makes the analysis not easy to generalize to the viscosity solution of the HJB equation, which does not have sufficient

regularity in general. Therefore,

$$\begin{aligned}
&A_k = \mu I_{2n+2} - \nabla^2\Phi_k(\widehat{t}_k, \widehat{x}_k, \widehat{s}_k, \widehat{y}_k) \\
&= \mu I_{2n+2} - \nabla^2\Phi_k(t_k, x_k, s_k, y_k) + \left(\nabla^2\Phi_k(t_k, x_k, s_k, y_k) - \nabla^2\Phi_k(\widehat{t}_k, \widehat{x}_k, \widehat{s}_k, \widehat{y}_k)\right) \\
&\geq \mu I_{2n+2} - \left|\nabla^2\Phi_k(t_k, x_k, s_k, y_k) - \nabla^2\Phi_k(\widehat{t}_k, \widehat{x}_k, \widehat{s}_k, \widehat{y}_k)\right| \cdot I_{2n+2} \geq \frac{1}{2}\mu I_{2n+2}.
\end{aligned}$$

Here, the inequality $\geq$ between two symmetric matrix means that their difference is positive semi-definite. In the first inequality, we use the fact that

$$\nabla^2\Phi_k(t_k, x_k, s_k, y_k) \leq 0,$$

coming form the optimality of (t_k, x_k, s_k, y_k). In the second equality, we use the estimate (5.150). Therefore, the Jacobian A_k in (5.148) always nonsingular when (5.149) holds and we confirm the claim after (5.147).

Next, we also want to derive an upper bound for this Jacobian. A direct calculation from (5.148) gives us

$$\begin{aligned}
\|A_k\|_2 \leq& \mu + \left\|\nabla_x^2 V_k(\widehat{t}_k, \widehat{x}_k)\right\|_2 + \left\|\nabla_y^2 V^*(\widehat{s}_k, \widehat{y}_k)\right\|_2 \\
&+ \frac{1}{\varepsilon}\left\|\begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}\right\|_2 + \frac{1}{\delta}\left\|\begin{pmatrix} I_n & -I_n \\ -I_n & I_n \end{pmatrix}\right\|_2 + \frac{4\lambda}{(\lambda/M)^2} \\
\leq& \mu + 2K + \frac{2}{\varepsilon} + \frac{2}{\delta} + \frac{4M^2}{\lambda} \leq C\left(\frac{1}{\varepsilon} + \frac{1}{\delta}\right).
\end{aligned}$$

Note that the notation $\|\cdot\|_2$ is the matrix norm (instead of the Frobenius norm). The first inequality above is by definition of φ (5.138) and Φ_k (5.139). The second is by boundedness of the derivatives of the value functions. The third is because λ is fixed while ε and δ are small and are going to 0s later. Therefore, we obtain an estimate

$$|(q, p, \widehat{q}, \widehat{p})| \leq C\left(\frac{1}{\varepsilon} + \frac{1}{\delta}\right)\left|(\widehat{t}_k - t_k, \widehat{x}_k - x_k, \widehat{s}_k - s_k, \widehat{y}_k - y_k)\right|. \tag{5.152}$$

Therefore, we also require that

$$r \leq \left(\frac{1}{\varepsilon} + \frac{1}{\delta}\right) \frac{r_1}{2C} \tag{5.153}$$

where the C in (5.153) is the same as the C in (5.152), in order to guarantee $|(q, p, \widehat{q}, \widehat{p})| \leq r_1$. Now we can see that the r depends on ε and δ, but it is independent of k.

Next, we consider the quantity

$$B_k := \partial_s V^*(\widehat{s}_k, \widehat{y}_k) - \partial_t V_k(\widehat{t}_k, \widehat{x}_k),$$

which depends on ε, δ, β, λ, q, p, $\widehat{q}$, $\widehat{p}$, μ, and k. On the one hand, by the optimality condition (5.146),

$$\begin{aligned} B_k &= -\partial_s \varphi(\widehat{t}_k, \widehat{x}_k, \widehat{s}_k, \widehat{y}_k) - \partial_t \varphi(\widehat{t}_k, \widehat{x}_k, \widehat{s}_k, \widehat{y}_k) - \mu \left[(\widehat{s}_k - s_k) + (\widehat{t}_k - t_k)\right] + q + \widehat{q} \\ &= 2\beta + \lambda/\widehat{t}_k^{\,2} + \lambda/\widehat{s}_k^{\,2} - \mu \left[(\widehat{s}_k - s_k) + (\widehat{t}_k - t_k)\right] + q + \widehat{q} \\ &\geq 2\beta + 2\lambda/T^2 - \mu \left(|\widehat{s}_k - s_k| + \left|\widehat{t}_k - t_k\right|\right) + q + \widehat{q}, \end{aligned} \tag{5.154}$$

where the terms with ε in $\partial_s \varphi$ and $\partial_t \varphi$ cancel each other.

On the other hand, using the HJ equations that V^* and V_k satisfy, we have

$$\begin{aligned} B_k &= G(\widehat{s}_k, \widehat{y}_k, u^*(\widehat{s}_k, \widehat{y}_k), -\nabla_y V^*, -\nabla_y^2 V^*) - G(\widehat{t}_k, \widehat{x}_k, u_k(\widehat{t}_k, \widehat{x}_k), -\nabla_x V_k, -\nabla_x^2 V_k) \\ &= \sup_u G(\widehat{s}_k, \widehat{y}_k, u, -\nabla_y V^*, -\nabla_y^2 V^*) - G(\widehat{t}_k, \widehat{x}_k, u_k(\widehat{t}_k, \widehat{x}_k), -\nabla_x V_k, -\nabla_x^2 V_k) \\ &\leq \sup_u G(\widehat{s}_k, \widehat{y}_k, u, -\nabla_y V^*, -\nabla_y^2 V^*) - \sup_u G(\widehat{t}_k, \widehat{x}_k, u, -\nabla_x V_k, -\nabla_x^2 V_k) \\ &\quad + L \left|u_k(\widehat{t}_k, \widehat{x}_k) - u_k^\diamond(\widehat{t}_k, \widehat{x}_k)\right| \\ &\leq \sup_u \left[G(\widehat{s}_k, \widehat{y}_k, u, -\nabla_y V^*, -\nabla_y^2 V^*) - G(\widehat{t}_k, \widehat{x}_k, u, -\nabla_x V_k, -\nabla_x^2 V_k)\right] \\ &\quad + L \left|u_k(\widehat{t}_k, \widehat{x}_k) - u_k^\diamond(\widehat{t}_k, \widehat{x}_k)\right|, \end{aligned}$$

where we have consecutively used: HJ equations for V^* and V_k; the optimality condition (5.10) for u^*; the definition of $u_k^\diamond$ (5.129) and the Lipschitz condition of G in u; a simple inequality. Therefore, by the definition of G (5.7),

$$\begin{aligned} B_k \leq \sup_u \Bigg\{ & \frac{1}{2} \operatorname{Tr}\left[\nabla_x^2 V_k(\widehat{t}_k, \widehat{x}_k)\sigma\sigma^\top(\widehat{x}_k, u) - \nabla_y^2 V^*(\widehat{s}_k, \widehat{y}_k)\sigma\sigma^\top(\widehat{y}_k, u)\right] \\ & + \left[\left\langle \nabla_x V_k(\widehat{t}_k, \widehat{x}_k),\, b(\widehat{x}_k, u)\right\rangle - \left\langle \nabla_y V^*(\widehat{s}_k, \widehat{y}_k),\, b(\widehat{y}_k, u)\right\rangle\right] \\ & + r(\widehat{x}_k, u) - r(\widehat{y}_k, u) \Bigg\} + L\left|u_k(\widehat{t}_k, \widehat{x}_k) - u_k^\diamond(\widehat{t}_k, \widehat{x}_k)\right| \\ =: \sup_u & \{(I) + (II) + (III)\} + L\left|u_k(\widehat{t}_k, \widehat{x}_k) - u_k^\diamond(\widehat{t}_k, \widehat{x}_k)\right|. \end{aligned} \tag{5.155}$$

Next, we bound the three terms in (5.155). Using the same argument for Φ_k on $\widehat{\Phi}_k$, we can show that (5.140) and (5.141) also hold for $(\widehat{t}_k, \widehat{x}_k, \widehat{s}_k, \widehat{y}_k)$. i.e.

$$\left|(\widehat{t}_k, \widehat{x}_k) - (\widehat{s}_k, \widehat{y}_k)\right| \leq 2L(\varepsilon + \delta) \tag{5.156}$$

and

$$\frac{1}{\varepsilon}\left|\widehat{t}_k - \widehat{s}_k\right|^2 + \frac{1}{\delta}\left|\widehat{x}_k - \widehat{y}_k\right|^2 \leq 4L^2(\varepsilon + \delta) \tag{5.157}$$

For (III), we have

$$(III) = r(\widehat{x}_k, u) - r(\widehat{y}_k, u) \leq L\left|\widehat{x}_k - \widehat{y}_k\right| \leq 2L^2(\varepsilon + \delta), \tag{5.158}$$

where we used Lipschitz condition of r in Assumption 2 and (5.156).

For (II), we have

$$\begin{aligned}(II) &= \left\langle \nabla_x \varphi(\widehat{t}_k, \widehat{x}_k, \widehat{s}_k, \widehat{y}_k) + \mu(\widehat{x}_k - x_k) - p,\, b(\widehat{x}_k, u) \right\rangle \\ &\quad + \left\langle \nabla_y \varphi(\widehat{t}_k, \widehat{x}_k, \widehat{s}_k, \widehat{y}_k) + \mu(\widehat{y}_k - y_k) - \widehat{p},\, b(\widehat{y}_k, u) \right\rangle \\ &= \left\langle \frac{1}{\delta}(\widehat{x}_k - \widehat{y}_k) + \mu(\widehat{x}_k - x_k) - p,\, b(\widehat{x}_k, u) \right\rangle \\ &\quad + \left\langle \frac{1}{\delta}(\widehat{y}_k - \widehat{x}_k) + \mu(\widehat{y}_k - y_k) - \widehat{p},\, b(\widehat{y}_k, u) \right\rangle \\ &\leq \frac{L}{\delta} |\widehat{x}_k - \widehat{y}_k|^2 + \mu K(|\widehat{x}_k - x_k| + |\widehat{y}_k - y_k|) + K(|p| + |\widehat{p}|) \\ &\leq 4L^3(\varepsilon + \delta) + \mu K(|\widehat{x}_k - x_k| + |\widehat{y}_k - y_k|) + K(|p| + |\widehat{p}|),\end{aligned} \tag{5.159}$$

where we have consecutively used: the optimality condition (5.146); the definition of φ in (5.138); boundness and Lipschitz condition of b in Assumption 2; the bound (5.157).

For (I), we have

$$\begin{aligned}(I) &= \frac{1}{2} \operatorname{Tr}\left[\begin{pmatrix} \sigma(\widehat{x}_k, u) \\ \sigma(\widehat{y}_k, u) \end{pmatrix}^\top \begin{pmatrix} \nabla_x^2 V_k(\widehat{t}_k, \widehat{x}_k) & 0 \\ 0 & -\nabla_y^2 V^*(\widehat{s}_k, \widehat{y}_k) \end{pmatrix} \begin{pmatrix} \sigma(\widehat{x}_k, u) \\ \sigma(\widehat{y}_k, u) \end{pmatrix} \right] \\ &\leq \frac{1}{2} \operatorname{Tr}\left[\begin{pmatrix} \sigma(\widehat{x}_k, u) \\ \sigma(\widehat{y}_k, u) \end{pmatrix}^\top \left(\frac{1}{\delta} \begin{pmatrix} I_n & -I_n \\ -I_n & I_n \end{pmatrix} + \mu I_{2n} \right) \begin{pmatrix} \sigma(\widehat{x}_k, u) \\ \sigma(\widehat{y}_k, u) \end{pmatrix} \right] \\ &= \frac{1}{2\delta} |\sigma(\widehat{x}_k, u) - \sigma(\widehat{y}_k, u)|^2 + \frac{\mu}{2} \left(|\sigma(\widehat{x}_k, u)|^2 + |\sigma(\widehat{y}_k, u)|^2 \right) \\ &\leq \frac{L^2}{2\delta} |\widehat{x}_k - \widehat{y}_k|^2 + \mu K^2 \leq 2L^4(\varepsilon + \delta) + \mu K^2,\end{aligned} \tag{5.160}$$

where we have consecutively used: a simple transform in linear algebra; the second order optimality condition in (5.146); a simple calculation; boundness and Lipschitz condition of σ in Assumption 2; the bound (5.157).

Combining (5.154), (5.155), (5.158), (5.159), and (5.160), we obtain

$$2\beta + 2\lambda/T^2 - \mu\left(|\widehat{s}_k - s_k| + \left|\widehat{t}_k - t_k\right|\right) + q + \widehat{q} \leq 2L^4(\varepsilon+\delta) + \mu K^2 + 4L^3(\varepsilon+\delta)$$
$$+\mu K(|\widehat{x}_k - x_k| + |\widehat{y}_k - y_k|) + K(|p| + |\widehat{p}|) + 2L^2(\varepsilon+\delta) + L\left|u_k(\widehat{t}_k, \widehat{x}_k) - u_k^\diamond(\widehat{t}_k, \widehat{x}_k)\right|,$$

which simplifies to

$$\begin{aligned} 2\beta + 2\lambda/T^2 \leq 8L^4(\varepsilon+\delta) + \mu K(|\widehat{s}_k - s_k| + \left|\widehat{t}_k - t_k\right| + |\widehat{x}_k - x_k| + |\widehat{y}_k - y_k|) \\ +\mu K^2 + K(|p| + |\widehat{p}| + |q| + |\widehat{q}|) + L\left|u_k(\widehat{t}_k, \widehat{x}_k) - u_k^\diamond(\widehat{t}_k, \widehat{x}_k)\right|. \end{aligned} \tag{5.161}$$

Next, we pick a box in $\mathbb{R}^{2n+2}$ that centered at (t_k, x_k, s_k, y_k) and have side length r. Then $|\widehat{x}_k - x_k|, |\widehat{y}_k - y_k| \leq \sqrt{n}r/2$, so we require $\sqrt{n}r/2$ (instead of r) satisfies (5.149) and (5.153). If we integrate (5.161) over the box w.r.t. $(\widehat{t}_k, \widehat{x}_k, \widehat{s}_k, \widehat{y}_k)$ and divided it by r^{2n+2}, we obtain

$$\begin{aligned} 2\beta + 2\lambda/T^2 \leq 8L^4(\varepsilon+\delta) + \mu K(1+\sqrt{n})r + \mu K^2 \\ + KC\left(\frac{1}{\varepsilon} + \frac{1}{\delta}\right)\sqrt{2n+2}\, r + r^{-2n-2}L\left\|u_k - u_k^\diamond\right\|_{L^1} \end{aligned} \tag{5.162}$$

where we have used (5.152). We recall that β and λ are fixed at first. We also recall that the definition of μ in (5.144) ensures that

$$\mu K(1+\sqrt{n})r + \mu K^2 \leq \frac{1}{2}\beta + \frac{\lambda}{2T^2}. \tag{5.163}$$

Therefore, if we firstly set ε and δ to be small such that

$$8L^4(\varepsilon+\delta) < \frac{1}{2}\beta + \frac{\lambda}{2T^2}. \tag{5.164}$$

Then we set r to be small such that

$$KC\left(\frac{1}{\varepsilon} + \frac{1}{\delta}\right)\sqrt{2n+2}\, r < \frac{1}{2}\beta + \frac{\lambda}{2T^2}, \tag{5.165}$$

where the C in (5.165) is the same as the C in (5.162). Next, note that $\|u_k - u_k^\diamond\|_{L^1} \leq \sqrt{T}\,\|u_k - u_k^\diamond\|_{L^2}$. So, by (5.135), we can set k to be large enough such that

$$r^{-2n-2} L \,\|u_k - u_k^\diamond\|_{L^1} < \frac{1}{2}\beta + \frac{\lambda}{2T^2}. \tag{5.166}$$

Finally, substituting (5.163), (5.164), (5.165), and (5.166) into (5.162), we obtain an contradiction, so Lemma 13 is proved. □

5.9 Conclusion and future directions

In conclusion, we study the stochastic optimal control problem with controlled diffusion in continuous time. We propose a policy gradient framework to solve the problem, where the control dynamic follows the gradient flow of the cost functional (5.16). We design a local optimal control function to analyze the convergence property of the algorithm and prove that the algorithm converges to the optimum under some regularity assumptions.

Our analysis can be extended in several directions. In this work, we concentrate on the time-invariant optimal control problem. It should not be too difficult to extend to more general time-dependent scenarios, although it may require additional regularity assumptions.

Regarding regularity, we have focused on the classical solutions to the HJ and HJB equations. It is natural to ask about viscosity solutions with less stringent regularity assumptions, which is an important future research direction.

As already mentioned in Section 5.3, our setting can be viewed as a limiting case under the actor-critic framework with two time scales. It is of interest to establish the full convergence of the actor-critic method with the critic (policy evaluation) dynamics included. It is also interesting to extend the analysis to a single time-scale actor-critic method, which couples together the control dynamics with policy

evaluation. We will leave these for future works.

Chapter 6

An Actor-Critic Framework for Stochastic Optimal Control Problems with Global Convergence Guarantee

This work is a follow up for the work in Chapter 5, and will also uses the techniques in Chapter 3 and 4. So I will omit the overlapped parts and focus on the innovative contents. This is still an ongoing work, so some details will be omitted and I will focus on the main idea.

6.1 Introduction

As is mentioned in section 5.9, it is more practical to consider the actor-critic method, with policy evaluation involved in the algorithm. In this work, we construct a VR-LSTD method for policy evaluation, which is based on the deep BSDE method. We will show a global convergence of the joint actor-critic dynamic under mild assumptions.

6.2 Policy evaluation for the critic and the joint actor-critic dynamic

In this section, we consider the policy evaluation process for a fixed control $u_t = u(t, x_t)$. Motivated by (5.4), we define the temporal difference (TD) (w.r.t. given $t_1 < t_2$) as

$$\mathrm{TD}_{vanilla} = \int_{t_1}^{t_2} r(x_t, u_t)\,\mathrm{d}t + \widehat{V}(t_2, x_{t_2}) - \widehat{V}(t_1, x_{t_1}), \tag{6.1}$$

where $\widehat{V}$ is some current estimate of the value function. This temporal difference can be understood as the inconsistency between the current value estimate and its sampled trajectory, which is commonly used in RL.

There is a generalized version of TD defined in [ZHL21], which is in the same spirit of the Deep BSDE method [HJE18]. A direction computation gives us

$$\begin{aligned} V(t_2, x_{t_2}) &= V(t_1, x_{t_1}) + \int_{t_1}^{t_2} \nabla V(t, x_t)^\top \sigma(x_t, u_t)\, \mathrm{d}W_t \\ &\quad + \int_{t_1}^{t_2} \left[\partial_t V(t, x_t) + \frac{1}{2} \operatorname{Tr}\left(\sigma\sigma^\top \nabla^2 V\right)(t, x_t, u_t) + b(x_t, u_t)^\top \nabla V(t, x_t) \right] \mathrm{d}t \\ &= V(t_1, x_{t_1}) - \int_{t_1}^{t_2} r(x_t, u_t)\mathrm{d}t + \int_{t_1}^{t_2} \nabla V(t, x_t)^\top \sigma(x_t, u_t)\, \mathrm{d}W_t, \end{aligned} \tag{6.2}$$

where we use Itô's Lemma and the HJ equation (5.8) in the first and second equation respectively. Therefore, the modified TD is defined as

$$\mathrm{TD}_{VR} = \int_{t_1}^{t_2} r(x_t, u_t)\, \mathrm{d}t + \widehat{V}(t_2, x_{t_2}) - \widehat{V}(t_1, x_{t_1}) - \int_{t_1}^{t_2} \widehat{\nabla V}(t, x_t)^\top \sigma(x_t, u_t)\, \mathrm{d}W_t. \tag{6.3}$$

The VR in (6.3) is short for "variance reduced", because this version of TD have lower variance in the optimization process [ZHL21]. From now on, we will use the variance reduced version of TD

$$\mathrm{TD} = \int_0^T r(x_t, u_t)\, \mathrm{d}t + h(x_T) - \widehat{V}(0, x_0) - \int_0^T \widehat{\nabla V}(t, x_t)^\top \sigma(x_t, u_t)\, \mathrm{d}W_t \tag{6.4}$$

to evaluate the value function. In the policy evaluation process, we apply a least square temporal difference (LSTD) method, which minimizes the following expected squared loss

$$\mathcal{L} = \frac{1}{2}\mathbb{E}\left[TD^2\right]. \tag{6.5}$$

We treat the approximation for $V(0,\cdot)$ and ∇V separately. We define the evolution of the value function and its gradient as the L^2 gradient flows of the loss (6.5)

$$\partial_\tau V^\tau(0,x) = -\frac{\delta \mathcal{L}}{\delta V(0,\cdot)}(x) = \mathbb{E}\left[TD\,\delta_{x_0}(x)\right] = \rho(0,x)\,\mathbb{E}\left[TD \mid x_0 = x\right] \tag{6.6}$$

$$\partial_\tau (\nabla V)^\tau(y) = -\frac{\delta \mathcal{L}}{\delta(\nabla V)}(y) = \mathbb{E}\left[TD \int_0^T \delta_{t,x_t}(y)\sigma(x_t,u_t)\,\mathrm{d}W_t\right] \tag{6.7}$$

As for the numerical implementation, we will parametrize the initial value function $V(0,\cdot)$ and the spatial gradient $V(\cdot,\cdot)$ by two neural networks. This parametrization will be analyzed in future works ane we will focus on the gradient flow in this work. Although these two functions are closely related, we state the following proposition to justify that it is fine to treat them separately.

Proposition 7. *The critic loss is* 0 *if and only if both the initial value function and the spatial gradient are exact.*

We do not have access to V_u in practice, but have its estimate V^τ instead. So, the actual evolution equation for the actor is

$$\partial_s u^\tau(t,x) = \rho^{u^\tau}(t,x)\ \nabla_u G\left(t,x,u(t,x),-\nabla_x V^\tau(t,x),-\nabla_x^2 V^\tau(t,x)\right).$$

Here, our policy evaluation has estimate of the value function and its gradient. So we will only consider the control problem with uncontrolled diffusion, i.e., σ is a function of x. In this case, $\nabla_u G$ in (5.7) does not depend on P and the actor dynamic becomes

$$\partial_s u^\tau(t,x) = \rho^{u^\tau}(t,x)\ \nabla_u G\left(t,x,u(t,x),-\nabla_x V^\tau(t,x)\right). \tag{6.8}$$

We summarize the joint dynamic of the actor and the critic

$$\begin{aligned}
\partial_\tau u^\tau(t,x) &= \rho(t,x)\ \nabla_u G\left(t,x,u^\tau(t,x),-\nabla_x V^\tau(t,x)\right)\\
\partial_\tau V^\tau(0,x) &= -\frac{\delta \mathcal{L}}{\delta V(0,\cdot)}(x)\\
\partial_\tau (\nabla V)^\tau(t,x) &= -\frac{\delta \mathcal{L}}{\delta(\nabla V)}(t,x)
\end{aligned} \tag{6.9}$$

6.3 Theoretical analysis

The critic convergence reduce to the convergence of the DeepBSDE method to solve the (parabolic) HJ equation.

Theorem 5 (critic)**.** *For any fixed control function u, we have*

$$\lim_{s\to\infty} \|V^s - V_u\|_{H^1} = 0.$$

Proof. The TD is linear in the value function and the critic loss is just TD^2. So the critic loss is convex in the value function. The minimizer of the critic loss is unique because $\min(TD^2) = 0$ is achieved only when V satisfies the HJ equation. □

Theorem 6 (critic convergence rate)**.** *The critic loss and critic error decays exponentially.*

proof of theorem 6. We denote V_u the true value function w.r.t. the control, which is also the solution of the HJ equation. By (6.2),

$$0 = \int_0^T r(x_t, u_t)\,\mathrm{d}t + h(x_T) - V_u(0, x_0) - \int_0^T \nabla V_u(t, x_t)^\top \sigma(x_t, u_t)\,\mathrm{d}W_t$$

So, the TD (6.4) can be rewritten as

$$TD = V_u(0, x_0) - V^\tau(0, x_0) + \int_0^T \left(\nabla V_u(t, x_t) - (\nabla V)^\tau(t, x_t)\right)^\top \sigma(x_t, u_t)\,\mathrm{d}W_t.$$

Therefore, the critic loss becomes

$$\begin{aligned}
&\mathcal{L} = \frac{1}{2}\mathbb{E}\left[TD^2\right] \\
&= \mathbb{E}\left[\left(V_u(0,x_0) - V^\tau(0,x_0) + \int_0^T \left(\nabla V_u(t,x_t) - (\nabla V)^\tau(t,x_t)\right)^\top \sigma(x_t,u_t)\,\mathrm{d}W_t\right)^2\right] \\
&= \frac{1}{2}\mathbb{E}\left[\left(V^\tau(0,x_0) - V_u(0,x_0)\right)^2 + \int_0^T \left\|\sigma(t,x_t,u_t)^\top\left((\nabla V)^\tau(t,x_t) - \nabla V_u(t,x_t)\right)\right\|^2 \mathrm{d}t\right] \\
&=: \mathcal{L}_1 + \mathcal{L}_2.
\end{aligned} \tag{6.10}$$

where we applied Itô's isometry in the second last equality. Therefore, the critic dynamic is decomposed into two separate L^2 gradient flow of $\mathcal{L}_1$ and $\mathcal{L}_2$:

$$\frac{\delta\mathcal{L}}{\delta V(0,\cdot)} = \frac{\delta\mathcal{L}_1}{\delta V(0,\cdot)} \quad \text{and} \quad \frac{\delta\mathcal{L}}{\delta(\nabla V)} = \frac{\delta\mathcal{L}_2}{\delta(\nabla V)}.$$

For $\mathcal{L}_1$,

$$\begin{aligned}
\partial_\tau V^\tau(0,x) &= -\frac{\delta\mathcal{L}_1}{\delta V(0,\cdot)}(x) = -\mathbb{E}\left[\left(V^\tau(0,x_0) - V_u(0,x_0)\right)\delta_{x_0}(x)\right] \\
&= -\rho(0,x)\left(V^\tau(0,x) - V_u(0,x)\right).
\end{aligned}$$

So

$$\begin{aligned}
\partial_\tau\mathcal{L}_1 &= \mathbb{E}\left[\left(V^\tau(0,x_0) - V_u(0,x_0)\right)\partial_\tau V^\tau(0,x_0)\right] \\
&= -\mathbb{E}\left[\rho(0,x_0)\left(V^\tau(0,x_0) - V_u(0,x_0)\right)^2\right] = -2\mathcal{L}_1
\end{aligned} \tag{6.11}$$

where we recall that $\rho(0,\cdot) \equiv 1$. So $\mathcal{L}_1$ decays exponentially.

For $\mathcal{L}_2$,

$$\begin{aligned}
&\frac{\delta\mathcal{L}_2}{\delta(\nabla V)}(s,x) = \mathbb{E}\left[\int_0^T \sigma\sigma^\top(t,x_t,u_t)\left((\nabla V)^\tau(t,x_t) - \nabla V_u(t,x_t)\right)\delta_{t,x_t}(s,x)\,\mathrm{d}t\right] \\
&= \mathbb{E}\left[\sigma\sigma^\top(s,x_s,u_s)\left((\nabla V)^\tau(s,x_s) - \nabla V_u(s,x_s)\right)\delta_{x_s}(x)\right] \\
&= \rho(s,x)\,\sigma\sigma^\top(s,x,u(s,x))\left((\nabla V)^\tau(s,x) - \nabla V_u(s,x)\right).
\end{aligned}$$

So

$$
\begin{aligned}
&\partial_s \mathcal{L}_2 \\
&= \mathbb{E}\left[\int_0^T \left((\nabla V)^s(t,x_t) - \nabla V_u(t,x_t)\right)^\top \sigma\sigma^\top(x_t,u_t)\ \partial_s(\nabla V)^s(t,x_t)\,\mathrm{d}t\right] \\
&= -\mathbb{E}\left[\int_0^T \left((\nabla V)^s(t,x_t) - \nabla V_u(t,x_t)\right)^\top (\sigma\sigma^\top)^2(x_t,u_t)\ \rho(t,x)\right. \\
&\quad \left.\left((\nabla V)^s(t,x_t) - \nabla V_u(t,x_t)\right)\,\mathrm{d}t\right]
\end{aligned}
$$

hence

$$
\begin{aligned}
&\partial_s \mathcal{L}_2 \\
&\le -\rho_0\,\sigma_0\,\mathbb{E}\left[\int_0^T \left(\nabla V^s(t,x_t) - (\nabla V)_u(t,x_t)\right)^\top \sigma\sigma^\top(x_t,u_t)\right. \\
&\quad \left.\left((\nabla V)^s(t,x_t) - \nabla V_u(t,x_t)\right)\,\mathrm{d}t\right] \\
&= -2\rho_0\,\sigma_0\,\mathcal{L}_2,
\end{aligned}
\tag{6.12}
$$

where have used Proposition 6 and the uniform ellipticity in the inequality. Combining (6.11) and (6.12), we show a exponential convergence rate of the critic loss. □

Next, we show the result for the joint dynamic of actor-critic algorithm.

Theorem 7 (joint dynamic)**.** *Assume that the actor-critic joint dynamic* (6.9) *reach a stable state V^∞ and u^∞, then u^∞ is the optimal control and V^∞ is the solution to the HJB equation and hence the optimal value function.*

Proof. Since the critic is static, V^∞ is the value function w.r.t. u^∞. Therefore, V^∞ is the solution of the HJ equation. Since the actor is static and G is concave in u, V^∞ is also the solution of the HJB equation. Therefore, V^∞ is also the optimal value function. □

We also have the result for a rate of convergence.

Theorem 8 (joint convergence rate)**.** *Let Assumptions 2, 3, and 4 holds. Assume that G is uniformly μ_G-strongly concave in u. Then both actor and critic converge exponentially.*

Proof sketch. We have shown that the policy evaluation algorithm converges exponentially in Theorem 6. We have also shown a rate of convergence for the policy gradient in 4. Therefore, in order to show the convergence of the joint dynamic, we need to estimate two quantities, the change of critic loss due to actor update, and the deviation of the control dynamic from the policy gradient due to error of value function. Thanks to the technique developed in Theorem 1, these two quantities can be bounded by the improvement of the actor and critic respectively. Therefore, we obtain a convergence rate of the joint dynamic. □

6.4 Conclusion and future research

This work extend the policy gradient algorithm in Chapter 5 to a practical actor-critic method and preserves its theoretical property.

One important direction of future research is to improve the policy evaluation method so that it is able to solve for the Hessian of the value function. In this case, the algorithm is able to solve the problem with controlled diffusion. One naïve way is to use auto differentiation for the neural network for ∇V. However, there is no theoretical guarantee for such method. In practice, (6.4) is usually discretized by the Euler-Maruyama scheme. However, the high order Milstein scheme [Mil75] can also be used, which improve the accuracy and involves the Hessian.

Another future direction of research is to study the neural network parametrization of the algorithm. For example, we can apply the single layer mean-field network

[CB18]. The control function is parametrized by a neural network

$$u_\mu : y \mapsto \int_\Theta \sigma_N(\langle \theta,\, y \rangle)\, \mathrm{d}\mu(\theta), \tag{6.13}$$

with derivative

$$\frac{\delta u_\mu}{\delta \mu}(\theta) = \overrightarrow{\sigma_N}(\langle \theta,\, \cdot \rangle). \tag{6.14}$$

As an analog for (5.16) and (6.8), the ideal and real dynamic for μ are

$$\partial_\tau \mu(\theta) = \int_0^T \mathbb{E}\left[\overrightarrow{\sigma_N}(\langle \theta,\, (t, x_t) \rangle\ \nabla_u G\left(t, x_t, u_\mu(t, x_t), \nabla V_{u_\mu}(t, x_t)\right)\right] \mathrm{d}t \tag{6.15}$$

and

$$\partial_\tau \mu(\theta) = \int_0^T \mathbb{E}\left[\overrightarrow{\sigma_N}(\langle \theta,\, (t, x_t) \rangle\ \nabla_u G\left(t, x_t, u_\mu(t, x_t), \nabla V^\tau(t, x_t)\right)\right] \mathrm{d}t \tag{6.16}$$

In practice, the expectation can be further approximated by Monte Carlo sampling.

Let $\Theta = \mathbb{R} \oplus \Theta_0$. The initial value function $V(0, \cdot)$ is parametrized by a Borel measure ν_0 on Θ_0:

$$V_{\nu_0} : x \mapsto \int_{\Theta_0} \sigma_N(\langle \theta_0,\, x \rangle)\, \mathrm{d}\nu_0(\theta_0). \tag{6.17}$$

We denote $y = (t, x)$. The spatial gradient of the value function is parametrized by ν, the direct product of d Borel measures on Θ:

$$(\nabla V)_\nu : y \mapsto \int_\Theta \sigma_N(\langle \theta,\, y \rangle)\, \mathrm{d}\nu(\theta). \tag{6.18}$$

Here, σ_N is an elementwise activation function and ν can be viewed as a vector valued measure. So

$$\frac{\delta V_{\nu_0}}{\delta \nu_0}(\theta_0) = \sigma_N(\langle \theta_0,\, \cdot \rangle) \quad \text{and} \quad \frac{\delta (\nabla V)_\nu}{\delta \nu}(\theta) = \overrightarrow{\sigma_N}(\langle \theta,\, \cdot \rangle), \tag{6.19}$$

where $\overrightarrow{\sigma_N}(\langle\theta, \cdot\rangle)$ should be understood as a function with vector value or diagonal matrix value according to the context. The derivative of temporal difference are

$$\frac{\delta TD}{\delta \nu_0} = -\sigma_N(\langle\cdot, x_0\rangle) \tag{6.20}$$

and

$$\frac{\delta TD}{\delta \nu} = -\int_0^T \overrightarrow{\sigma_N}(\langle\cdot, (t, x_t)\rangle)^\top \sigma(x_t, u_t)\, \mathrm{d}W_t. \tag{6.21}$$

The dynamic of the measures ν_0 and ν are

$$\begin{aligned} \partial_\tau \nu_0 &= -\mathbb{E}\left[TD \frac{\delta TD}{\delta \nu_0}\right] = \mathbb{E}\left[TD\, \sigma_N(\langle\cdot, x_0\rangle)\right] \\ \partial_\tau \nu &= -\mathbb{E}\left[TD \frac{\delta TD}{\delta \nu}\right] = \mathbb{E}\left[TD \int_0^T \overrightarrow{\sigma_N}(\langle\cdot, (t, x_t)\rangle)^\top \sigma(x_t, u_t)\, \mathrm{d}W_t\right] \end{aligned} \tag{6.22}$$

Chapter 7

Conclusions

This thesis collects my works on machine learning to solve traditional scientific computing problems during my Ph.D. studies, which include a wide range of partial differential equation (PDE) problems and optimal control problems.

The eigenvalue solver based on deep learning could accurately solve the eigenvalue problem. The actor-critic framework to solve the optimal control problem and the HJB equation has both numerical success and theoretical guarantees. The actor-critic framework could also solve the LQR problem efficiently.

The future directions of research has been mentioned at the end of each chapter.

Bibliography

[AA68] Vladimir Igorevich Arnold and André Avez. *Ergodic problems of classical mechanics*, volume 9. Benjamin, 1968.

[AB07] Mohammed Shahid Abdulla and Shalabh Bhatnagar. Parametrized actor-critic algorithms for finite-horizon MDPs. In *2007 American Control Conference*, pages 534–539. IEEE, 2007.

[ABC+16] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In *12th USENIX symposium on operating systems design and implementation (OSDI 16)*, pages 265–283, 2016.

[AC79] Jean-Pierre Aubin and FH Clarke. Shadow prices and duality for a class of optimal control problems. *SIAM Journal on Control and Optimization*, 17(5):567–586, 1979.

[AM07] Brian DO Anderson and John B Moore. *Optimal control: linear quadratic methods*. Courier Corporation, 2007.

[Aro59] DG Aronson. The fundamental solution of a linear parabolic equation containing a small parameter. *Illinois Journal of Mathematics*, 3(4):580–619, 1959.

[Aro67] Donald Gary Aronson. Bounds for the fundamental solution of a parabolic equation. *Bulletin of the American Mathematical society*, 73(6):890–896, 1967.

[BA06] Shalabh Bhatnagar and Mohammed Shahid Abdulla. A reinforcement learning based algorithm for finite horizon Markov decision processes. In *Proceedings of the 45th IEEE Conference on Decision and Control*, pages 5519–5524. IEEE, 2006.

[Bar13] Jonathan F Bard. *Practical bilevel optimization: algorithms and applications*, volume 30. Springer Science & Business Media, 2013.

[BB96] Steven J Bradtke and Andrew G Barto. Linear least-squares algorithms for temporal difference learning. *Machine learning*, 22(1):33–57, 1996.

[BCJ19] Sebastian Becker, Patrick Cheridito, and Arnulf Jentzen. Deep optimal stopping. *Journal of Machine Learning Research*, 20:74, 2019.

[BD+97] Martino Bardi, Italo Capuzzo Dolcetta, et al. *Optimal control and viscosity solutions of Hamilton-Jacobi-Bellman equations*, volume 12. Springer, 1997.

[Bel66] Richard Bellman. Dynamic programming. *Science*, 153(3731):34–37, 1966.

[Ber12] Dimitri Bertsekas. *Dynamic programming and optimal control: Volume I*, volume 1. Athena scientific, 2012.

[Ber19] Dimitri Bertsekas. *Reinforcement learning and optimal control*. Athena Scientific, 2019.

[BHJK20] Christian Beck, Martin Hutzenthaler, Arnulf Jentzen, and Benno Kuckuck. An overview on deep learning-based approximation methods for partial differential equations. *arXiv preprint arXiv:2012.12348*, 2020.

[BHM08] Olivier Bahn, Alain Haurie, and Roland Malhamé. A stochastic control model for optimal timing of climate policies. *Automatica*, 44(6):1545–1558, 2008.

[BJ02] Guy Barles and Espen Robstad Jakobsen. On the convergence rate of approximation schemes for Hamilton–Jacobi–Bellman equations. *ESAIM: Mathematical Modelling and Numerical Analysis*, 36(1):33–54, 2002.

[Bot12] Léon Bottou. Stochastic gradient descent tricks. In *Neural networks: Tricks of the trade*, pages 421–436. Springer, 2012.

[Boy99] Justin A Boyan. Least-squares temporal difference learning. In *ICML*, pages 49–56. Citeseer, 1999.

[BP03] FM Buchmann and WP Petersen. Solving Dirichlet problems numerically using the Feynman–Kac representation. *BIT Numerical Mathematics*, 43(3):519–540, 2003.

[BSGL09] Shalabh Bhatnagar, Richard S Sutton, Mohammad Ghavamzadeh, and Mark Lee. Natural actor-critic algorithms. *Automatica*, 45(11):2471–2482, 2009.

[BSS81] Richard Blankenbecler, DJ Scalapino, and RL Sugar. Monte Carlo calculations of coupled boson-fermion systems. I. *Physical Review D*, 24(8):2278, 1981.

[BSW97] Randal W Beard, George N Saridis, and John T Wen. Galerkin approximations of the generalized Hamilton–Jacobi–Bellman equation. *Automatica*, 33(12):2159–2177, 1997.

[BSW98] Randal W Beard, George N Saridis, and John T Wen. Approximate solutions to the time-invariant Hamilton–Jacobi–Bellman equation. *Journal of Optimization theory and Applications*, 96(3):589–626, 1998.

[BZ03] J Frédéric Bonnans and Housnaa Zidani. Consistency of generalized finite difference schemes for the stochastic hjb equation. *SIAM Journal on Numerical Analysis*, 41(3):1008–1021, 2003.

[CB18] Lenaic Chizat and Francis Bach. On the global convergence of gradient descent for over-parameterized models using optimal transport. *Advances in neural information processing systems*, 31, 2018.

[CCK77] David Ceperley, Geoffrey V. Chester, and M.H. Kalos. Monte Carlo simulation of a many-fermion study. *Physical Review B*, 16(7):3081–3099, 1977.

[CIL92] Michael G Crandall, Hitoshi Ishii, and Pierre-Louis Lions. User's guide to viscosity solutions of second order partial differential equations. *Bulletin of the American mathematical society*, 27(1):1–67, 1992.

[CL82] Felix L Chernousko and AA Lyubushin. Method of successive approximations for solution of optimal control problems. *Optimal Control Applications and Methods*, 3(2):101–114, 1982.

[CL21] René Carmona and Mathieu Laurière. Convergence analysis of machine learning algorithms for the numerical solution of mean field control and games i: the ergodic case. *SIAM Journal on Numerical Analysis*, 59(3):1455–1485, 2021.

[CL22] René Carmona and Mathieu Laurière. Convergence analysis of machine learning algorithms for the numerical solution of mean field control and games: Ii—the finite horizon case. *The Annals of Applied Probability*, 32(6):4065–4105, 2022.

[CMC20] Kenny Choo, Antonio Mezzacapo, and Giuseppe Carleo. Fermionic neural-network states for ab-initio electronic structure. *Nature communications*, 11(1):2368, 2020.

[CSXY22] Tianyi Chen, Yuejiao Sun, Quan Xiao, and Wotao Yin. A single-timescale method for stochastic bilevel optimization. In *International Conference on Artificial Intelligence and Statistics*, pages 2466–2488. PMLR, 2022.

[CWNMW19] Quentin Chan-Wai-Nam, Joseph Mikael, and Xavier Warin. Machine learning for semi linear PDEs. *Journal of Scientific Computing*, 79(3):1667–1712, 2019.

[DCH+16] Yan Duan, Xi Chen, Rein Houthooft, John Schulman, and Pieter Abbeel. Benchmarking deep reinforcement learning for continuous control. In *International Conference on Machine Learning*, pages 1329–1338. PMLR, 2016.

[DEJM+20] Carles Domingo-Enrich, Samy Jelassi, Arthur Mensch, Grant Rotskoff, and Joan Bruna. A mean-field analysis of two-player zero-sum games. *Advances in Neural Information Processing Systems*, 33:20215–20226, 2020.

[DKK21] Sergey Dolgov, Dante Kalise, and Karl K Kunisch. Tensor decomposition methods for high-dimensional hamilton–jacobi–bellman equations. *SIAM Journal on Scientific Computing*, 43(3):A1625–A1650, 2021.

[DMM+20] Sarah Dean, Horia Mania, Nikolai Matni, Benjamin Recht, and Stephen Tu. On the sample complexity of the linear quadratic regulator. *Foundations of Computational Mathematics*, 20(4):633–679, 2020.

[DWS12] Thomas Degris, Martha White, and Richard Sutton. Off-policy actor-critic. In *International Conference on Machine Learning*, 2012.

[EH16] Weinan E and Jiequn Han. Deep learning approximation for stochastic control problems. *arXiv preprint arXiv:1611.07422*, 2016.

[EHJ17] Weinan E, Jiequn Han, and Arnulf Jentzen. Deep learning-based numerical methods for high-dimensional parabolic partial differential equations and backward stochastic differential equations. *Communications in Mathematics and Statistics*, 5(4):349–380, 2017.

[ESJB10] OS Ebrahim, MF Salem, PK Jain, and MA Badr. Application of linear quadratic regulator theory to the stator field-oriented control of induction motors. *IET Electric Power Applications*, 4(8):637–646, 2010.

[EY18] Weinan E and Bing Yu. The deep Ritz method: a deep learning-based numerical algorithm for solving variational problems. *Communications in Mathematics and Statistics*, 6(1):1–12, 2018.

[FGKM18] Maryam Fazel, Rong Ge, Sham Kakade, and Mehran Mesbahi. Global convergence of policy gradient methods for the linear quadratic regulator. In *International Conference on Machine Learning*, pages 1467–1476. PMLR, 2018.

[FJ22] Dena Firoozi and Sebastian Jaimungal. Exploratory lqg mean field games with entropy regularization. *Automatica*, 139:110177, 2022.

[FL07] Peter A Forsyth and George Labahn. Numerical methods for controlled Hamilton–Jacobi–Bellman PDEs in finance. *Journal of Computational Finance*, 11(2):1, 2007.

[FMNR01] W.M.C. Foulkes, Lubos Mitas, Richarad J. Needs, and Gunaretnam Rajagopal. Quantum Monte Carlo simulations of solids. *Reviews of Modern Physics*, 73(1):33, 2001.

[FR12] Wendell H Fleming and Raymond W Rishel. *Deterministic and stochastic optimal control*, volume 1. Springer Science & Business Media, 2012.

[Fri08] Avner Friedman. *Partial differential equations of parabolic type*. Courier Dover Publications, 2008.

[FYW20] Zuyue Fu, Zhuoran Yang, and Zhaoran Wang. Single-timescale actor-critic provably finds globally optimal policy. *arXiv preprint arXiv:2008.00483*, 2020.

[GG22] Emmanuel Gobet and Maxime Grangereau. Newton method for stochastic control problems. *SIAM Journal on Control and Optimization*, 60(5):2996–3025, 2022.

[Gob00] Emmanuel Gobet. Weak approximation of killed diffusion using euler schemes. *Stochastic processes and their applications*, 87(2):167–197, 2000.

[Gro61] Eugene P Gross. Structure of a quantized vortex in boson systems. *Il Nuovo Cimento*, 20(3):454–477, 1961.

[GRS95] Walter R Gilks, Sylvia Richardson, and David Spiegelhalter. *Markov chain Monte Carlo in practice*. CRC press, 1995.

[GXZ22] Xin Guo, Renyuan Xu, and Thaleia Zariphopoulou. Entropy regularization for mean field games with learning. *Mathematics of Operations Research*, 2022.

[Has19] Aamir Hashim. Optimal speed control for direct current motors using linear quadratic regulator. *Journal of Engineering and Computer Science (JECS)*, 14(2):48–56, 2019.

[Hec92] James J Heckman. Policy evaluation. *Evaluating welfare and training programs*, page 201, 1992.

[HH20] Jiequn Han and Ruimeng Hu. Deep fictitious play for finding markovian Nash equilibrium in multi-agent games. In *Mathematical and Scientific Machine Learning*, pages 221–245. PMLR, 2020.

[HH21] Jiequn Han and Ruimeng Hu. Recurrent neural networks for stochastic control problems with delay. *Mathematics of Control, Signals, and Systems*, 33:775–795, 2021.

[HHL20] Jiequn Han, Ruimeng Hu, and Jihao Long. Convergence of deep fictitious play for stochastic differential games. *arXiv preprint arXiv:2008.05519*, 2020.

[HJE18] Jiequn Han, Arnulf Jentzen, and Weinan E. Solving high-dimensional partial differential equations using deep learning. *Proceedings of the National Academy of Sciences*, 115(34):8505–8510, 2018.

[HL17] Pierre Henry-Labordere. Deep primal-dual algorithm for BSDEs: Applications of machine learning to CVA and IM. *Available at SSRN 3071506*, 2017.

[HL20] Jiequn Han and Jihao Long. Convergence of the deep bsde method for coupled FBSDEs. *Probability, Uncertainty and Quantitative Risk*, 5(1):1–33, 2020.

[HLZ20] Jiequn Han, Jianfeng Lu, and Mo Zhou. Solving high-dimensional eigenvalue problems using deep neural networks: A diffusion monte carlo like approach. *Journal of Computational Physics*, 423:109792, 2020.

[HNS20] Jihun Han, Mihai Nica, and Adam R Stinchcombe. A derivative-free method for solving elliptic partial differential equations with deep neural networks. *Journal of Computational Physics*, 419:109672, 2020.

[HSN20] Jan Hermann, Zeno Schätzle, and Frank Noé. Deep-neural-network solution of the electronic schrödinger equation. *Nature Chemistry*, 12(10):891–897, 2020.

[HZE19] Jiequn Han, Linfeng Zhang, and Weinan E. Solving many-electron Schrödinger equation using deep neural networks. *Journal of Computational Physics*, 399:108929, 2019.

[HZRS16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[IRZ21] Kazufumi Ito, Christoph Reisinger, and Yufei Zhang. A neural network-based policy iteration algorithm with global h 2-superlinear convergence for stochastic games on domains. *Foundations of Computational Mathematics*, 21(2):331–374, 2021.

[IS15] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pages 448–456, 2015.

[Itô53] Seizô Itô. The fundamental solution of the parabolic equation in a differentiable manifold. *Osaka Mathematical Journal*, 5(1):75–92, 1953.

[JPPZ20] Shaolin Ji, Shige Peng, Ying Peng, and Xichuan Zhang. Three algorithms for solving high-dimensional fully coupled FBSDEs through deep learning. *IEEE Intelligent Systems*, 35(3):71–84, 2020.

[JPPZ22] Shaolin Ji, Shige Peng, Ying Peng, and Xichuan Zhang. Solving stochastic optimal control problem via stochastic maximum principle with deep learning method. *Journal of Scientific Computing*, 93(1):30, 2022.

[Kak01] Sham M Kakade. A natural policy gradient. *Advances in neural information processing systems*, 14, 2001.

[KB15] Diederik Kingma and Jimmy Ba. Adam: a method for stochastic optimization. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015.

[KBP13] Jens Kober, J Andrew Bagnell, and Jan Peters. Reinforcement learning in robotics: A survey. *The International Journal of Robotics Research*, 32(11):1238–1274, 2013.

[KK18] Dante Kalise and Karl Kunisch. Polynomial approximation of high-dimensional Hamilton–Jacobi–Bellman equations and applications to feedback control of semilinear parabolic PDEs. *SIAM Journal on Scientific Computing*, 40(2):A629–A652, 2018.

[Kle05] Fima C Klebaner. *Introduction to stochastic calculus with applications*. World Scientific Publishing Company, 2005.

[KNS16] Hamed Karimi, Julie Nutini, and Mark Schmidt. Linear convergence of gradient and proximal-gradient methods under the polyak-łojasiewicz condition. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2016, Riva del Garda, Italy, September 19-23, 2016, Proceedings, Part I 16*, pages 795–811. Springer, 2016.

[Kop62] Richard E Kopp. Pontryagin maximum principle. In *Mathematics in Science and Engineering*, volume 5, pages 255–279. Elsevier, 1962.

[KSS20a] Bekzhan Kerimkulov, David Siska, and Lukasz Szpruch. Exponential convergence and stability of howard's policy improvement algorithm for controlled diffusions. *SIAM Journal on Control and Optimization*, 58(3):1314–1340, 2020.

[KSS20b] Stefan Kremsner, Alexander Steinicke, and Michaela Szölgyenyi. A deep neural network algorithm for semilinear elliptic PDEs with applications in insurance mathematics. *Risks*, 8(4):136, 2020.

[KŠS21] Bekzhan Kerimkulov, David Šiška, and Lukasz Szpruch. A modified msa for stochastic control problems. *Applied Mathematics & Optimization*, 84(3):3417–3436, 2021.

[KT99] Vijay Konda and John Tsitsiklis. Actor-critic algorithms. *Advances in neural information processing systems*, 12, 1999.

[KT00] Vijay R Konda and John N Tsitsiklis. Actor-critic algorithms. In *Advances in Neural Information Processing Systems*, pages 1008–1014. Citeseer, 2000.

[Kus90] Harold J Kushner. Numerical methods for stochastic control problems in continuous time. *SIAM Journal on Control and Optimization*, 28(5):999–1048, 1990.

[KVX04] Karl Kunisch, Stefan Volkwein, and Lei Xie. HJB-POD-based feedback design for the optimal control of evolution problems. *SIAM Journal on Applied Dynamical Systems*, 3(4):701–722, 2004.

[KW17] Wei Kang and Lucas C Wilcox. Mitigating the curse of dimensionality: sparse grid characteristics method for optimal feedback control and HJB equations. *Computational Optimization and Applications*, 68(2):289–315, 2017.

[LLLO22] Wonjun Lee, Siting Liu, Wuchen Li, and Stanley Osher. Mean field control problems for vaccine distribution. *Research in the Mathematical Sciences*, 9(3):51, 2022.

[LSU88] Olga A Ladyzenskaja, Vsevolod Alekseevich Solonnikov, and Nina N Uralceva. *Linear and quasi-linear equations of parabolic type*, volume 23. American Mathematical Soc., 1988.

[Lyg04] John Lygeros. On reachability and minimum cost optimal control. *Automatica*, 40(6):917–927, 2004.

[LZBY20] Yanli Liu, Kaiqing Zhang, Tamer Basar, and Wotao Yin. An improved analysis of (variance-reduced) policy gradient and natural policy gradient methods. In *NeurIPS*, 2020.

[MBT05] Ian M Mitchell, Alexandre M Bayen, and Claire J Tomlin. A time-dependent Hamilton–Jacobi formulation of reachable sets for continuous dynamic games. *IEEE Transactions on automatic control*, 50(7):947–957, 2005.

[McM65] William Lauchlin McMillan. Ground state of liquid He 4. *Physical Review*, 138(2A):A442, 1965.

[Mil75] GN Mil'shtejn. Approximate integration of stochastic differential equations. *Theory of Probability & Its Applications*, 19(3):557–562, 1975.

[MKS+13] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

[MSBS10] Hamid R Maei, Csaba Szepesvári, Shalabh Bhatnagar, and Richard S Sutton. Toward off-policy learning control with function approximation. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 719–726, 2010.

[MT03] Ian M Mitchell and Claire J Tomlin. Overapproximating reachable sets by Hamilton–Jacobi projections. *Journal of Scientific Computing*, 19(1):323–346, 2003.

[Mun06] Rémi Munos. Policy gradient in continuous time. *Journal of Machine Learning Research*, 7:771–791, 2006.

[MZC+21] Cameron Martin, Hongyuan Zhang, Julia Costacurta, Mihai Nica, and Adam R Stinchcombe. Solving elliptic equations with brownian motion: Bias reduction and temporal difference learning. *Methodology and Computing in Applied Probability*, pages 1–24, 2021.

[MZSJ21] Hesameddin Mohammadi, Armin Zare, Mahdi Soltanolkotabi, and Mihailo R Jovanovic. Convergence and sample complexity of gradient methods for the model-free linear quadratic regulator problem. *IEEE Transactions on Automatic Control*, 2021.

[NR21] Nikolas Nüsken and Lorenz Richter. Solving high-dimensional hamilton–jacobi–bellman pdes using neural networks: perspectives from the theory of controlled diffusions and measures on path space. *Partial differential equations and applications*, 2:1–48, 2021.

[NTDR09] Richarad J. Needs, Michael D. Towler, Neil D. Drummond, and P. López Ríos. Continuum variational and diffusion quantum Monte Carlo calculations. *Journal of Physics: Condensed Matter*, 22(2):023201, 2009.

[NZGK21] Tenavi Nakamura-Zimmerer, Qi Gong, and Wei Kang. Adaptive deep learning for high-dimensional hamilton–jacobi–bellman equations. *SIAM Journal on Scientific Computing*, 43(2):A1221–A1247, 2021.

[ONL+22] Derek Onken, Levon Nurbekyan, Xingjian Li, Samy Wu Fung, Stanley Osher, and Lars Ruthotto. A neural network approach for high-dimensional optimal control applied to multiagent path finding. *IEEE Transactions on Control Systems Technology*, 31(1):235–251, 2022.

[OS88] Stanley Osher and James A Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton–Jacobi formulations. *Journal of computational physics*, 79(1):12–49, 1988.

[OSS19] Mathias Oster, Leon Sallandt, and Reinhold Schneider. Approximating the stationary Hamilton–Jacobi–Bellman equation by hierarchical tensor products. *arXiv preprint arXiv:1911.00279*, 2019.

[Par98] Étienne Pardoux. Backward stochastic differential equations and viscosity solutions of systems of semilinear parabolic and elliptic PDEs of second order. In *Stochastic Analysis and Related Topics VI*, pages 79–127. Springer, 1998.

[Pen90] Shige Peng. A general stochastic maximum principle for optimal control problems. *SIAM Journal on control and optimization*, 28(4):966–979, 1990.

[Pen91] Shige Peng. Probabilistic interpretation for systems of quasilinear parabolic partial differential equations. *Stochastics and Stochastics Reports*, 37(1-2):61–74, 1991.

[Pha09] Huyên Pham. *Continuous-time stochastic control and optimization with financial applications*, volume 61. Springer Science & Business Media, 2009.

[Pit61] Lev P Pitaevskii. Vortex lines in an imperfect Bose gas. *Soviet Physics—JETP*, 13(2):451–454, 1961.

[PP90] Etienne Pardoux and Shige Peng. Adapted solution of a backward stochastic differential equation. *Systems & Control Letters*, 14(1):55–61, 1990.

[PP92] Etienne Pardoux and Shige Peng. Backward stochastic differential equations and quasilinear parabolic partial differential equations. In *Stochastic partial differential equations and their applications*, pages 200–217. Springer, 1992.

[PPA+19] David Pfau, Stig Petersen, Ashish Agarwal, David G. T. Barrett, and Kimberly L. Stachenfeld. Spectral inference networks: Unifying deep and spectral learning. In *International Conference on Learning Representations*, 2019.

[PS08] Jan Peters and Stefan Schaal. Natural actor-critic. *Neurocomputing*, 71(7-9):1180–1190, 2008.

[PSMF20] David Pfau, James S Spencer, Alexander GDG Matthews, and W Matthew C Foulkes. Ab initio solution of the many-electron schrödinger equation with deep neural networks. *Physical Review Research*, 2(3):033429, 2020.

[PWET19] Marcus A Pereira, Ziyi Wang, Ioannis Exarchos, and Evangelos A. Theodorou. Learning deep stochastic optimal control policies using forward-backward SDEs. In *Robotics: science and systems*, 2019.

[PWG21] Huyen Pham, Xavier Warin, and Maximilien Germain. Neural networks-based backward scheme for fully nonlinear PDEs. *SN Partial Differential Equations and Applications*, 2(1):1–24, 2021.

[Rao09] Anil V Rao. A survey of numerical methods for optimal control. *Advances in the Astronautical Sciences*, 135(1):497–528, 2009.

[Ris96] Hannes Risken. Fokker-planck equation. In *The Fokker-Planck Equation*, pages 63–95. Springer, 1996.

[ROL+20] Lars Ruthotto, Stanley J Osher, Wuchen Li, Levon Nurbekyan, and Samy Wu Fung. A machine learning framework for solving high-dimensional mean field game and mean field control problems. *Pro-*

ceedings of the National Academy of Sciences, 117(17):9183–9193, 2020.

[RW06] Steven Richardson and Song Wang. Numerical solution of Hamilton–Jacobi–Bellman equations by an exponentially fitted finite volume method. *Optimization*, 55(1-2):121–140, 2006.

[SB18] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[SHM+16] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, and Marc Lanctot. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016.

[SLH+14] David Silver, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra, and Martin Riedmiller. Deterministic policy gradient algorithms. In *International Conference on Machine Learning*, pages 387–395. PMLR, 2014.

[SMD17] Ankur Sinha, Pekka Malo, and Kalyanmoy Deb. A review on bilevel optimization: from classical to evolutionary approaches and applications. *IEEE Transactions on Evolutionary Computation*, 22(2):276–295, 2017.

[SMSM99] Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. *Advances in neural information processing systems*, 12, 1999.

[ŠS20] David Šiška and Łukasz Szpruch. Gradient flows for regularized stochastic control problems. *arXiv preprint arXiv:2006.05956*, 2020.

[SYZZ02] Suresh P Sethi, Houmin Yan, Hanqin Zhang, and Qing Zhang. Optimal and hierarchical controls in dynamic stochastic manufacturing systems: A survey. *Manufacturing & Service Operations Management*, 4(2):133–170, 2002.

[Tay12] Michael Eugene Taylor. *Partial differential equations. 3, Nonlinear equations*. Springer, 2012.

[TR18] Stephen Tu and Benjamin Recht. Least-squares temporal difference learning for the linear quadratic regulator. In *International Conference on Machine Learning*, pages 5005–5014. PMLR, 2018.

[TZZ22] Wenpin Tang, Yuming Paul Zhang, and Xun Yu Zhou. Exploratory HJB equations and their convergence. *SIAM Journal on Control and Optimization*, 60(6):3191–3216, 2022.

[VL10] Kyriakos G Vamvoudakis and Frank L Lewis. Online actor-critic algorithm to solve the continuous-time infinite horizon optimal control problem. *Automatica*, 46(5):878–888, 2010.

[WBH+16] Ziyu Wang, Victor Bapst, Nicolas Heess, Volodymyr Mnih, Remi Munos, Koray Kavukcuoglu, and Nando de Freitas. Sample efficient actor-critic with experience replay. *arXiv preprint arXiv:1611.01224*, 2016.

[WHJ21] E Weinan, Jiequn Han, and Arnulf Jentzen. Algorithms for solving high dimensional pdes: from nonlinear monte carlo to machine learning. *Nonlinearity*, 35(1):278, 2021.

[WHYW21] Weichen Wang, Jiequn Han, Zhuoran Yang, and Zhaoran Wang. Global convergence of policy gradient for linear-quadratic mean-field control/game in continuous time. In *International Conference on Machine Learning*, pages 10772–10782. PMLR, 2021.

[Wie00] Marco A Wiering. Multi-agent reinforcement learning for traffic light control. In *Machine Learning: Proceedings of the Seventeenth International Conference (ICML'2000)*, pages 1151–1158, 2000.

[WJT03] Song Wang, Les S Jennings, and Kok Lay Teo. Numerical solution of Hamilton–Jacobi–Bellman equations by an upwind finite volume method. *Journal of Global Optimization*, 27(2):177–192, 2003.

[WZXG20] Yue Frank Wu, Weitong Zhang, Pan Xu, and Quanquan Gu. A finite-time analysis of two time-scale actor-critic methods. *Advances in Neural Information Processing Systems*, 33:17617–17628, 2020.

[WZZ20] Haoran Wang, Thaleia Zariphopoulou, and Xun Yu Zhou. Reinforcement learning in continuous time and space: A stochastic control approach. *J. Mach. Learn. Res.*, 21(198):1–34, 2020.

[XGZ+11] Yong Xu, Rencai Gu, Huiqing Zhang, Wei Xu, and Jinqiao Duan. Stochastic bifurcations in a bistable Duffing–Van der Pol oscillator with colored noise. *Physical Review E*, 83(5):056215, 2011.

[YCHW19] Zhuoran Yang, Yongxin Chen, Mingyi Hong, and Zhaoran Wang. Provably global convergence of actor-critic: A case for linear quadratic regulator with ergodic cost. *Advances in neural information processing systems*, 32, 2019.

[YZ99] Jiongmin Yong and Xunyu Zhou. *Stochastic controls: Hamiltonian systems and HJB equations*, volume 43. Springer, 1999.

[ZCG97] Shiwei Zhang, Joseph Carlson, and James E. Gubernatis. Constrained path Monte Carlo method for fermion ground states. *Physical Review B*, 55(12):7464, 1997.

[ZDR21] Sihan Zeng, Thinh T Doan, and Justin Romberg. A two-time-scale stochastic optimization framework with applications in control and reinforcement learning. *arXiv preprint arXiv:2109.14756*, 2021.

[ZHL21] Mo Zhou, Jiequn Han, and Jianfeng Lu. Actor-critic method for high dimensional static hamilton–jacobi–bellman partial differential equations based on neural networks. *SIAM Journal on Scientific Computing*, 43(6):A4043–A4066, 2021.

[Zho21] Xun Yu Zhou. Curse of optimality, and how we break it. *Available at SSRN 3845462*, 2021.

[ZL22] Mo Zhou and Jianfeng Lu. Single time-scale actor-critic method to solve the linear quadratic regulator with convergence guarantees. *arXiv preprint arXiv:2202.00048*, 2022.

[ZL23] Mo Zhou and Jianfeng Lu. A policy gradient framework for stochastic optimal control problems with global convergence guarantee. *arXiv preprint arXiv:2302.05816*, 2023.

Biography

Mo Zhou is a driven and dedicated scholar with a passion for mathematics. From a young age, he excelled in mathematics, and his natural talent for the subject led them to pursue an academic career in the field.

After completing his secondary education, Mo Zhou enrolled at Tsinghua University to study mathematics. He showed exceptional ability in a wide range of mathematical disciplines, from calculus and linear algebra to numerical analysis and partial differential equations. He also participated in various research projects, collaborating with faculty members on topics such as signal decomposition and machine learning.

Upon graduation, Mo Zhou was awarded a prestigious fellowship to pursue a Ph.D. in mathematics at Duke University. Under the guidance of his advisor, Prof. Jianfeng Lu, he developed a deep interest in the intersection of traditional scientific computing problem and machine learning, particularly in the area of deep learning method to solve PDE problems. His research focused on the development of new techniques for studying both the theoretical and numerical aspects of the application of deep learning on scientific computing problems.

Throughout his Ph.D. studies, Mo Zhou was awarded several research grants and fellowships, and were invited to present his research at numerous national and international conferences. His works have been published in numerous high-impact academic journals, such as Journal of Computational Physics and Journal on Scientific Computing. He also served as an instructor for various courses, demonstrating a natural talent for communicating complex mathematical concepts to students of all levels.