Convergence of the Deep Galerkin Method for the Mean Field Control Problem

U(M) Math REU – Summer 2023

Jake Hofgard, mentored by Asaf Cohen

August 10, 2023

Abstract

We consider the convergence of the deep Galerkin method (DGM), a deep learning-based scheme for solving high-dimensional nonlinear PDE, for Hamilton-Jacobi-Bellman (HJB) equations that arise from the study of mean field control problems (MFCP). Based on a recent characterization of the value function of the MFCP as the unique viscosity solution of an HJB equation on the simplex, we establish both an existence and convergence result for the DGM. First, we show that the loss functional of the DGM can be made arbitrarily small given that the value function of the MFCP possesses sufficient regularity. Then, we show that if the loss functional of the DGM converges to zero, the corresponding neural network approximators must converge uniformly to the true value function on the simplex. We also provide numerical experiments demonstrating the DGM's ability to generalize to high-dimensional HJB equations.

Contents

1 Introduction **2**

2 The N-Agent Optimization and Mean Field Control Problems **4**
2.1 N-agent Optimization Problem . . . 4
2.2 Mean Field Control Problem . . . 5
2.3 Assumptions for the MFCP . . . 6
2.4 Convergence, Existence, and Uniqueness results for the MFCP . . . 7

3 Deep Galerkin Method (DGM) **8**
3.1 Original Algorithm . . . 8
3.2 Modified Algorithm . . . 9

4 Universal Approximation with Two-Layer Feedforward Neural Networks **10**

5 Convergence of the DGM **11**
5.1 Approximation Via Two-Layer Neural Networks . . . 11
5.2 Convergence of Neural Network Approximators to Value Function . . . 17
5.3 Equicontinuity of Neural Network Approximators . . . 20

6 Numerical Results **21**

References **25**

A Derivations of HJB Equations **27**

1 Introduction

Mean field control problems (MFCPs) describe the limiting behavior of cooperative games with a finite number of interacting agents attempting to minimize a common cost. The N-agent optimization problem, which limits to the MFCP as N approaches infinity, is a continuous-time, finite state optimization problem with N agents $\mathbf{X} = (X^1, \dots, X^N)$ belonging to the state space $[\![d]\!] := \{1, \dots, d\}$. The agents follow the dynamics given by the continuous-time Markov chain

$$\mathbb{P}(X^k_{t+h} = j \mid \mathbf{X}_t = \mathbf{x}) = Q_{x_k,j}(t, \beta_k(t, \mathbf{x}), \mu^N_{\mathbf{x}})h + o(h), \tag{1}$$

where μ^N is the empirical measure of the N agents and the agents choose feedback controls $\boldsymbol{\beta} = (\beta_1, \dots, \beta_N)$ to minimize the common cost

$$J^N(\boldsymbol{\beta}) := \frac{1}{N}\sum_{k=1}^{N} \mathbb{E}\left[\int_0^T f(t, X^k_t, \beta_k(t, X^k_t), \mu^N_t)dt + g(X^T_k, \mu^N_T)\right]. \tag{2}$$

As is standard, f denotes the running cost of the stochastic control problem while g denotes the terminal cost. In practice, finding the optimal control in N-agent stochastic control problems quickly becomes intractable as N grows. As a result, the MFCP is of theoretical and practical interest in stochastic control. More rigorously, as N approaches infinity, the N-agent optimization problem resembles an optimization problem involving a *single* agent that evolves via that process

$$\mathbb{P}(X_{t+h} = j \mid X_t = i) = Q_{i,j}(t, \alpha^i(t), \mathrm{Law}(X_t))h + o(h), \tag{3}$$

where α is now the feedback control of the single agent, and $\mathrm{Law}(X_0) = m_0$ is predetermined. Similar to the N-agent case, the single agent aims to minimize the cost

$$J(\alpha) := \mathbb{E}\left[\int_0^T f(t, X_t, \alpha(t, X_t), \mathrm{Law}(X_t))dt + g(X_T, \mathrm{Law}(X_T))\right]. \tag{4}$$

This problem can be considered as a deterministic control problem in terms of the Fokker–Planck equation for $\mu_t := \mathrm{Law}(X_t)$. In particular, μ solves the system

$$\begin{aligned} &\frac{d}{dt}\mu^i_t = \sum_{j\in[\![d]\!]} \left(\mu^j_t Q_{j,i}(t, \alpha^j(t), \mu_t) - \mu^i_t Q_{i,j}(t, \alpha^i(t), \mu_t)\right), \\ &\mu_0 = m_0, \end{aligned} \tag{5}$$

and the cost functional for the deterministic control problem becomes

$$J(\alpha) := \int_0^T \sum_{i\in[\![d]\!]} f(s, i, \alpha^i(s), \mu_s)\mu^i_s ds + \sum_{i\in[\![d]\!]} g^i(\mu_T)\mu^i_T. \tag{6}$$

By a standard optimal control argument via an appropriate dynamic programming principle, this gives rise to an HJB equation on $[0, T] \times S_d$, where S_d is the $(d-1)$-dimensional simplex. Specifically, the value function $V : [0, T] \times S_d \to \mathbb{R}$ for the MFCP solves the HJB equation

$$\begin{aligned} &-\partial_t V(t, m) + \sum_{i\in[\![d]\!]} m_i H^i(t, m, D^i V(t, m)) = 0, \\ &V(T, m) = \sum_{i\in[\![d]\!]} m_i g^i(m), \end{aligned} \tag{7}$$

where $D^i V(t, m)$ denotes the vector $(\partial_{m_j - m_i} V(t, m))_{j\in[\![d]\!]}$. Note that on the simplex, directional derivatives are only permitted in the directions $e_j - e_i$, where e_i denotes the ith standard basis vector in $\mathbb{R}^d$. It is Equation (7) that we focus on solving numerically due to the connection between the MFCP and the N-agent optimization problem.

Importantly, recent work rigorously establishes the connection between the N-agent stochastic control problem

and the corresponding MFCP. While we restrict ourselves to the HJB equation that arises from the MFCP with feedback controls, the case of open-loop controls is considered in [1], where a forward-backward system of McKean-Vlasov SDEs is the primary object of interest. In the case of Markovian feedback controls, Lacker [2] shows that the optimal control-state pairs of the MFCP, also referred to as control of McKean-Vlasov dynamics, is the limit in distribution of near-optimal control-state pairs of the N-agent control problem.

Taking a slightly different approach, Cecchin [3] reformulates the MFCP in terms of the deterministic optimal control of a Fokker–Planck equation as in Equations (5) and (6) above. The resulting deterministic control problem yields a dynamic programming principle and Hamilton-Jacobi-Bellman (HJB) equation solved by the value function of the MFCP. By formulating the N-agent optimization problem in a similar manner, one obtains two HJB equations: one for the N-agent optimization problem and one for the MFCP. Both HJB equations have unique viscosity solutions, as shown in [3]. In turn, [3] establishes an explicit rate of convergence between V^N, the value function for the N-agent problem, and V, the value function for the MFCP. In particular, a convergence rate of $1/\sqrt{N}$ is established. In a similar vein, Kolokoltsov [4] obtains a convergence rate of $1/N$ under additional regularity assumptions. Notably, [3] places no regularity assumptions on the value function V (other than Lipschitz continuity), and in the most general case, does not require convexity of the running or terminal costs of the MFCP.

We aim to construct a numerical scheme for efficiently solving the HJB equation associated with the MFCP. In particular, as the dimension d increases, the so-called "curse of dimensionality" prevents standard numerical schemes (e.g., Monte Carlo methods, mesh-based algorithms, etc.) from solving the HJB equation in a tractable manner. However, recent advancements in deep learning present promising options for solving high-dimensional, non-linear PDE such as the HJB equation in Equation (7).

Two leading methods have been presented for parabolic PDE that resemble the HJB equation: the deep Galerkin method (DGM) and deep backwards stochastic differential equations (BSDE). The primary focus of this paper is the DGM. This approach, first introduced in [5], models itself after classical finite element methods for solving low-dimensional PDE. However, the DGM is a mesh-free method; instead of creating basis functions that approximate the solution to a PDE from a mesh, the DGM utilizes neural network approximations that only depend on the parameters, the architecture, and the activation function between layers of the network. The loss functional of the original DGM attempts to minimize the L^2-error of both the PDE and the terminal condition during training, ultimately learning the parameters that best approximate the solution of the PDE. In [5], the authors introduce the DGM, illustrate its ability to numerically solve high-dimensional nonlinear PDE, and provide a convergence guarantee for second-order nonlinear parabolic PDE.

The second popular deep learning-based method for solving high-dimensional PDE, introduced in [6] and expanded upon in [7, 8], exploits the connection between nonlinear parabolic PDE and backward SDEs can be exploited via a so-called non-linear Feynman-Kac formula. In turn, the resulting backward SDE can be solved numerically by recursively using a sequence of neural networks to solve the SDE along a specified discretization of the time interval in question, starting with the terminal condition of the original PDE. Although this method, often referred to as deep BSDE, is likely applicable to our context, we defer further consideration of deep backward schemes to future work.

We show that the convergence guarantee provided in [5] extends to the class of HJB equations associated with the MFCP, relying on the theory of viscosity solutions for HJB equations to obtain the desired convergence. Importantly, our class of HJB equations does not fall into the class of second-order nonlinear parabolic PDE considered in [5], so we require a different proof technique. We first show that a sequence of neural networks taking the loss functional of the DGM to zero exists. Then, we prove that for such a sequence of networks, the neural network approximators converge uniformly to the true value function for the MFCP on $[0, T] \times S_d$.

We structure the remainder of the paper as follows. Section 2 introduces the MFCP and related HJB equation in full detail and provides a connection between the N-agent optimization problem and the MFCP. Additionally, we present the relevant assumptions placed upon the MFCP that ensure sufficient regularity of the value function. Section 3 presents the DGM in both its original form and the modified version of the algorithm for the HJB equation that we use in our numerical experiments and convergence proof. Before presenting the convergence proof, we provide an overview of universal approximation theorems in Section 4, a key concept from machine learning theory that we require in order to show convergence of the DGM. In Section 5 we present the

full convergence proof of the DGM for the HJB equation in addition to a discussion of neural networks with bounded weights and equicontinuous families of neural networks. Finally, Section 6 presents selected numerical results from a TensorFlow implementation of the DGM.

2 The N-Agent Optimization and Mean Field Control Problems

Below, we outline the N-agent optimization problem, the resulting MFCP, and the recent existence, uniqueness, and convergence results for the MFCP proven in [3]. Derivations of the HJB equations for both the N-agent problem and the MFCP can be found in Appendix A. We also present the relevant assumptions imposed upon the MFCP. This section utilizes the background on stochastic control and HJB equations provided in [9].

2.1 N-agent Optimization Problem

In the N-agent problem, we assume that the dynamics in

$$\mathbb{P}(X^k_{t+h} = j \mid \mathbf{X}_t = \mathbf{x}) = Q_{x_k,j}(t, \beta_k(t, \mathbf{x}), \mu^N_{\mathbf{x}})h + o(h) \tag{8}$$

hold, and N agents aim to minimize the common cost

$$J^N(\boldsymbol{\beta}) = \frac{1}{N}\sum_{k=1}^{N} \mathbb{E}\left[\int_0^T f(t, X^k_t, \beta_k(t, X^k_t), \mu^N_t)dt + g(X^T_k, \mu^N_T)\right], \tag{9}$$

where the running cost f and terminal cost g depend on the empirical distribution of the agents, with coordinates given by

$$\mu^N_{i,t} = \frac{1}{N}\sum_{k=1}^{N} \mathbb{1}_{X^k_t = i}$$

for $i \in \{1, \dots, d\}$. Let A^N denote the set of admissible controls, which we take to be measurable, Markovian feedback controls. Then, the associated value function (which we aim to describe as the solution to an appropriate HJB equation) is given by

$$v^N(t, \mathbf{x}) = \inf_{\boldsymbol{\beta} \in A^N} \frac{1}{N}\sum_{k=1}^{N} \mathbb{E}\left[\int_t^T f(s, X^k_s, \beta_k(s, X^k_s), \mu^N_s)ds + g(X^T_k, \mu^N_T) \,\middle|\, \mathbf{X}_t = \mathbf{x}\right] =: \inf_{\boldsymbol{\beta} \in A^N} J^N(t, \boldsymbol{\beta}, \mathbf{x}), \tag{10}$$

assuming that minimizing the cost in (2) is the goal of the agents. We make the same assumptions as [3], discussed in detail below, so that the value function v belongs to $\mathcal{C}^{1,1}([0,T] \times S_d)$; see [3, Theorem 3.5]. In particular, this ensures that the HJB equation for the N-agent optimization problem (and the MFCP) has a classical solution.

At this point, we introduce some relevant notation used throughout the paper. In particular, let $[\![d]\!]^N := \{1, \dots, d\}^N$. Then, given $\mathbf{x} \in [\![d]\!]^N$, we define $[\mathbf{x}^{-k}, j] \in [\![d]\!]^N$ for $j \in [\![d]\!]^N$ by

$$[\mathbf{x}^{-k}, j]_\ell = \begin{cases} x_\ell & \ell \neq k, \\ j & \ell = k. \end{cases}$$

In turn, given $u : [\![d]\!]^N \to \mathbb{R}^d$, we define $\Delta^k u \in \mathbb{R}^d$ by $\Delta^k u(\mathbf{x})_j = u([\mathbf{x}^{-k}, j]) - u(\mathbf{x})$.

We utilize the notation $Q_{i,\bullet}$ to denote the ith row of the transition rate matrix $(Q_{i,j})_{i,j \in [\![d]\!]^N}$. From this, we define the pre-Hamiltonian $\mathcal{H}^i : [0,T] \times A \times S_d \times \mathbb{R}^d \to \mathbb{R}$ for each $i \in [\![d]\!]$ by

$$\mathcal{H}^i(t, a, m, z) := -\langle Q_{i,\bullet}(t, a, m), z\rangle - f(t, i, a, m). \tag{11}$$

In turn, the corresponding Hamiltonian is given by

$$H^i(t, m, z) = \sup_{a \in A} \mathcal{H}^i(t, a, m, z) \tag{12}$$

for each $i \in [\![d]\!]$.

With this notation out of the way, we can state the HJB equation for the N-agent optimization problem.

Proposition 2.1. *The value function v^N defined in Equation (10) is $\mathcal{C}^1$ in time and uniquely solves the HJB equation*

$$\begin{aligned} &-\frac{\partial v^N}{\partial t}(t,\mathbf{x}) + \frac{1}{N}\sum_{k=1}^{N} H^{x_k}(t,\mu_{\mathbf{x}}^N, N\Delta^k v^N(t,\mathbf{x})) = 0, \\ &v^N(T,x) = \frac{1}{N}\sum_{k=1}^{N} g(x_k,\mu_{\mathbf{x}}^N). \end{aligned} \tag{13}$$

In [3], it is shown that the above N-agent optimization problem is in fact equivalent to a single optimization problem in terms of the empirical distribution of the agents. This problem is given by a time-inhomogeneous Markov chain with dynamics given by

$$\mathbb{P}\left(\mu_{t+h}^N = m + \frac{1}{N}(\delta_j - \delta_i) \;\middle|\; \mu_t^N = m\right) = N m_i Q_{i,j}(t,\alpha_N(t,i,m),m)h + o(h), \tag{14}$$

where $m \in S_d^N$, $i \neq j \in [\![d]\!]$, and each $\alpha^N(t,\cdot,m) \in A^d$ is now a control that depends only on the state of each agent and the empirical distribution of the agents. The cost functional the problem is now given by the expression

$$J^N(\alpha_N,t,m) = \mathbb{E}\left[\int_t^T \sum_{i\in[\![d]\!]} m_{i,t}\mu_{i,s}^N f(s,i,\alpha_N(s,i,\mu_s^N),\mu_s^N)ds + \sum_{i\in[\![d]\!]} \mu_{i,T}^N g^i(\mu_T^N)\right]. \tag{15}$$

Defining

$$D_j^{N,i}v(m) := N\left(v(m+\frac{1}{N}(\delta_j-\delta_i)) - v(m)\right),$$

we have the following analogous proposition.

Proposition 2.2. *The value function V^N for the control problem described by Equations (14) – (15) is $\mathcal{C}^1$ in time and uniquely solves the HJB equation*

$$\begin{aligned} &-\frac{\partial V^N}{\partial t}(t,m) + \sum_{i\in[\![d]\!]} m_i H^i(t,m,D^{N,i}V^N(t,m)) = 0, \\ &V^N(T,m) = \sum_{i\in[\![d]\!]} m_i g^i(m). \end{aligned} \tag{16}$$

As noted above and shown in [3, Proposition 2.6], the HJB equations in Proposition 2.1 and Proposition 2.2 are equivalent. Specifically, the original N-agent optimization problem and the reformulated problem in which the agents only interact through their empirical distribution are equivalent control problems.

2.2 Mean Field Control Problem

In the case of the MFCP, we instead consider a single agent that follows that dynamics

$$\mathbb{P}(X_{t+h} = j \mid X_t = i) = Q_{i,j}(t,\alpha^i(t),\mathrm{Law}(X_t))h + o(h),$$

with $\mathrm{Law}(X_0) = m_0$. The associated cost functional is then

$$J(\alpha) = \mathbb{E}\left[\int_0^T f(t,X_t,\alpha(t,X_t),\mathrm{Law}(X_t))dt + g(X_T,\mathrm{Law}(X_T))\right] \tag{17}$$

As in [3], this can be considered as a deterministic control problem, where we aim to minimize the cost

$$J(t,\alpha,\mu) = \int_t^T \sum_{i\in[\![d]\!]} f(s,i,\alpha^i(s),\mu_s)\mu_s^i ds + \sum_{i\in[\![d]\!]} g^i(\mu_T)\mu_T^i, \tag{18}$$

where f is the running cost and g is the terminal cost exactly as above. Additionally, μ satisfies the dynamics

$$\begin{aligned} \frac{d}{dt}\mu_t^i &= \sum_{j\in[\![d]\!]} \left(\mu_t^j Q_{j,i}(t,\alpha^j(t),\mu_t) - \mu_t^i Q_{i,j}(t,\alpha^i(t),\mu_t)\right), \\ \mu_0 &= m_0, \end{aligned} \tag{19}$$

which arise from the Fokker-Planck equation for the process $(X_t)_{t\in[0,T]}$. From a standard control argument, we have the following proposition for the MFCP.

Proposition 2.3. *Under the assumptions presented in Section 2.3 below, the value function $V \in \mathcal{C}^{1,1}([0,T]\times S_d)$ for the MFCP is the unique classical solution of the HJB equation*

$$\begin{aligned} -\partial_t V(t,m) + \sum_{i\in[\![d]\!]} m_i H^i(t,m,D^i V(t,m)) &= 0, \\ V(T,m) = \sum_{i\in[\![d]\!]} m_i g^i(m). \end{aligned} \tag{20}$$

All three HJB equations presented in Propositions 2.1 – 2.3 are derived in Appendix A.

2.3 Assumptions for the MFCP

Under the assumptions presented in [3], Equation 7 has a unique classical solution $V \in \mathcal{C}^{1,1}([0,T]\times S_d)$. In particular, we start by assuming the following, drawing upon some of the notation introduced throughout the preceding section.

Assumption A. *(A1) The action space A forms a compact metric space when equipped with some metric ρ.*

(A2) The transition rate $Q_{i,j}$ is continuous on $[0,T]\times A\times S_d$ and Lipschitz in (t,m), with Lipschitz constant $C>0$ that doesn't depend on a:

$$|Q_{i,j}(t,a,m) - Q_{i,j}(s,a,p)| \le C(|t-s| + |m-p|).$$

(A3) The functions F and G, defined by

$$F(t,a^1,\dots,a^d,m) := \sum_{i\in[\![d]\!]} m_i f(t,i,a^i,m)$$

and

$$G(m) = \sum_{i\in[\![d]\!]} m_i g^i(m),$$

are such that F is continuous on $[0,T]\times A\times S_d$ and

$$\begin{aligned} |F(t,a,m) - F(s,a,p)| &\le C(|t-s| + |m-p|) \\ |G(m) - G(p)| &\le C|m-p|. \end{aligned}$$

Assumption B. *In addition to Assumption (A), we have that:*

(B1) $A = [0,M]^d$ for some constant $M>0$.

(B2) The transition rate is homogeneous and given by $Q_{i,j}(t,a,m) = a_j$.

(B3) For each $i\in[\![d]\!]$, the running cost f is continuously differentiable in a, $\nabla_a f$ is Lipschitz continuous with respect to m, and there exists $\lambda>0$ such that

$$f(t,i,b,m) \ge f(t,i,a,m) + \langle \nabla_a f(t,i,a,m), b-a\rangle + \lambda|b-a|^2.$$

Assumption C. *In addition to both Assumption (A), and Assumption (B), we have that:*

(C1) $F(\cdot,a,\cdot)\in\mathcal{C}^{1,1}([0,T]\times S_d)$ and $G\in\mathcal{C}^{1,1}(S_d)$.

(C2) The function

$$[0,T] \times [0,\infty)^{d\times d} \times Int(S_d) \ni (t,w,m) \mapsto \sum_{i\in[\![d]\!]} m_i f\left(t, i, \left(\frac{w_{i,j}}{m_i}\right)_{j\neq i}, m\right)$$

is convex in (w,m).

(C3) G is convex in m.

2.4 Convergence, Existence, and Uniqueness results for the MFCP

Under the above assumptions, Cecchin [3] derived a series of useful results for the MFCP. We begin with a uniqueness result for solutions to the above HJB equations, from [3, Theorem 2.9].

Theorem 2.4. *Let V be the value function for the deterministic control problem in (6). Then, if:*

(1) Assumption (A) holds, V is the unique viscosity solution of Equation (7) on S_d and V is Lipschitz continuous in (t,m).

(2) Assumptions (A) and (B) hold, there exists an optimal control to the deterministic MFCP.

(3) Assumptions (A) – (C) hold, then $V \in \mathcal{C}^{1,1}([0,T]\times S_d)$ is the unique classical solution of the HJB equation in Equation (7).

Although the above theorem is the most important result from [3] for our work, Cecchin also presents several convergence results connecting the MFCP to the N-agent optimization problem. In particular, we have, from [3, Theorem 2.10] the following result concerning convergence of the value function V^N for the N-agent optimization problem in (1) to the value function V for the MFCP. Note that this result only requires the standard stochastic control assumptions from Assumption (A).

Theorem 2.5. *Under Assumption (A), we have that*

$$\max_{(t,m)\in[0,T]\times S_d} \left|V^N(t,m) - V(t,m)\right| \leq \frac{C}{\sqrt{N}}$$

for all $N \in \mathbb{N}$.

Next, [3, Theorem 2.11] contains a similar convergence result for the cost obtained by the optimal control for the MFCP.

Theorem 2.6. *Let $\varepsilon > 0$ and $N \in \mathbb{N}$. Then, under Assumption (A), if $\alpha : [0,T] \to A^d$ is an ε-optimal control for the MFCP,*

$$J^N(\alpha) \leq \inf_{\alpha^N \in A} J^N(\alpha^N) + \frac{C}{\sqrt{N}} + \varepsilon,$$

where J^N is the cost functional for the N-agent optimization problem in (2).

Finally, Cecchin also presents a propagation of chaos of result, describing the connection between the optimal trajectory of the N-agent optimization problem and the MFCP in [3, Theorem 2.13].

Theorem 2.7. *If Assumption (B) holds $V \in \mathcal{C}^{1,1}([0,T]\times S_d)$, then*

$$\mathbb{E}\left[\sup_{t\in[0,T]} |\mu_t^N - \mu_t|\right] \leq \frac{C}{N^{1/9}},$$

where μ^N is the process in (14) and μ is the optimal trajectory of the MFCP, satisfying (19).

3 Deep Galerkin Method (DGM)

3.1 Original Algorithm

In this section, we present the deep Galerkin method (DGM) algorithm, first proposed by [5], in the context of the HJB equation for the MFCP. Specifically, consolidating the discussion in the previous section, we aim to approximate solutions to the following first-order, nonlinear PDE:

$$\begin{aligned} -\partial_t V(t,m) + \sum_{i\in[\![d]\!]} m_i H^i(t,m,D^i V(t,m)) = 0, \\ V(T,m) = \sum_{i\in[\![d]\!]} m_i g^i(m). \end{aligned} \tag{21}$$

The DGM aims to efficiently approximate a solution to the above equation using a deep learning-based approach. Specifically, the method learns model parameters $\theta \in \mathbb{R}^K$, where K depends on the dimension d of the simplex S_d, the number of layers in the neural network used, and the number of nodes in each layer. The DGM learns the model parameters θ by minimizing an objective functional, given by

$$L(\varphi) := \left\| -\partial_t \varphi(t,m;\theta) + \sum_{i\in[\![d]\!]} m_i H^i(t,m,D^i\varphi(t,m;\theta)) \right\|^2_{2,[0,T]\times S_d,\nu_1} + \|\varphi(T,m;\theta) - \sum_{i\in[\![d]\!]} m_i g^i(m)\|^2_{2,S_d,\nu_2}. \tag{22}$$

Above, ν_1 and ν_2 are taken to be probability measures on $[0,T]\times S_d$ and S_d respectively. Below, we typically refer to this objective as the L^2-error of the neural network approximator $\varphi(t,m;\theta)$. Analytically minimizing the objective functional with respect to the model parameters θ is computationally intractable in higher dimensions, as numerically integrating over S_d requires the generation of a mesh.

Instead, the method proposed in [5] suggests utilizing stochastic gradient descent (SGD), a common technique in machine learning that involves sampling to minimize an objective such as the one in (22). Thus, we obtain a mesh-free algorithm for minimizing (22) with respect to the model parameters θ as follows:

(1) Initialize model parameters $\theta^{(0)} \in \mathbb{R}^K$. These may be initialized randomly or according to some heuristic.

(2) At each step, sample points $(t^{(n)}, m^{(n)}) \in [0,T]\times S_d$ and $p^{(n)} \in S_d$ from the corresponding densities ν_1 and ν_2.

(3) Calculate the squared error

$$\begin{aligned} G(t^{(n)},m^{(n)},p^{(n)},\theta^{(n)}) = &\left(-\partial_t\varphi(t^{(n)},m^{(n)};\theta^{(n)}) + \sum_{i\in[\![d]\!]} m_i H^i(t^{(n)},m^{(n)},D^i\varphi(t^{(n)},m^{(n)};\theta^{(n)}) \right)^2 \\ &+ \left(\varphi(T,p^{(n)};\theta^{(n)}) - \sum_{i\in[\![d]\!]} p_i^{(n)} g^i(p^{(n)}) \right)^2 \end{aligned}$$

(4) Update the model parameters $\theta^{(n)}$ from the sampled point $(t^{(n)},m^{(n)},p^{(n)}) \in [0,T]\times S_d \times S_d$ according to the update rule

$$\theta^{(n+1)} = \theta^{(n)} - \alpha^{(n)} \nabla_\theta G(t^{(n)},m^{(n)},p^{(n)},\theta^{(n)}).$$

(5) Repeat steps (2) – (4) until a convergence criterion, such as $G(\cdot,\theta^{(n)}) < \delta$ for some small $\delta > 0$, is satisfied.

Note that the derivative of the squared error with respect to the model parameters θ can be computed via the standard machine learning algorithm of backpropagation. Additionally, at each step, the stochastic gradient $\nabla_\theta G(t^{(n)},m^{(n)},p^{(n)},\theta^{(n)})$ is an unbiased estimate of the gradient $\nabla_\theta L(\varphi(\cdot;\theta^{(n)}))$. Specifically, we have that

$$\mathbb{E}\left[\nabla_\theta G(t^{(n)},m^{(n)},p^{(n)},\theta^{(n)}) \mid \theta^{(n)} \right] = \nabla_\theta L(\varphi(\cdot;\theta^{(n)})).$$

This ensures that, in expectation, step (4) above will update the model parameters according to a descent direction. In step (4), the learning rate $\alpha^{(n)}$ is selected according to a schedule that is decreasing in n; the authors in [5] present a heuristic schedule that they found to work particularly well. The above algorithm guarantees that as $n \to \infty$,

$$|\nabla_\theta L(\varphi(\cdot;\theta^{(n)}))| \to 0,$$

meaning that the L^2-error will converge to a critical point with respect to the model parameters. Deep neural networks typically possess local minima, so the above convergence guarantee does *not* ensure that $\varphi(t,m;\theta) = V(t,m)$ upon convergence. In Section 5 below, we present a more comprehensive overview of the theoretical guarantees that the DGM provides when applied to Equation (7).

3.2 Modified Algorithm

Although the above algorithm, using the L^2-error as defined in [5], seems to work quite well in practice, the structure of the PDE in Equation (7) prohibits us from using the same argument as [5] to prove convergence of the original DGM algorithm. Instead, by slightly changing the loss function used to train the neural network approximation in DGM, we are able to prove convergence of a modified DGM algorithm to the unique value function of the MFCP. Specifically, by utilizing a loss function that approximates the uniform norm of the PDE and terminal condition rather than the squared error of the PDE and the terminal condition, we are able to leverage the theory of viscosity solutions for first-order HJB equations from [10] in our convergence proof. Consequently, the loss function for the modified algorithm is given by

$$\tilde{L}(\varphi) = \max_{(t,m)\in[0,T]\times S_d} \left| -\partial_t \varphi(t,m;\theta) + \sum_{i\in[\![d]\!]} m_i H^i(t,m,D^i\varphi(t,m;\theta)) \right| + \max_{m\in S_d} \left| \varphi(T,m;\theta) - \sum_{i\in[\![d]\!]} m_i g^i(m) \right|. \quad (23)$$

The maxima over $[0,T]\times S_d$ and S_d are approximated by sampling, as demonstrated in the following algorithm. As with the original DGM algorithm, we utilize stochastic gradient descent (SGD) to find the parameter $\theta \in \mathbb{R}^K$ that minimizes the above loss. Note that in the following algorithm, the architecture of the neural network is fixed, and only the parameter θ is updated by SGD. As a proxy for the true loss functional, given a set of M samples $(t^{(j)},m^{(j)},p^{(j)})_{j=1,\dots,M} \in [0,T]\times S_d$, we define

$$\begin{aligned}\tilde{G}((t^{(j)},m^{(j)},p^{(j)})_{j=1,\dots,M},\theta) := &\max_{j=1,\dots,M} \left| -\partial_t \varphi(t^{(j)},m^{(j)};\theta) + \sum_{i\in[\![d]\!]} m_i^{(j)} H^i(t^{(j)},m^{(j)},D^i\varphi(t^{(j)},m^{(j)};\theta)) \right| \\ &+ \max_{j=1,\dots,M} \left| \varphi(T,p^{(j)};\theta) - \sum_{i\in[\![d]\!]} p_i^{(j)} g^i(p^{(j)}) \right|.\end{aligned} \quad (24)$$

Algorithm 1 Uniform DGM

Initialize parameters $\theta^{(0)} \in \mathbb{R}^K$
Initialize tolerance $\delta \in (0,1)$
$n \leftarrow 0$
while $\tilde{G}(\theta^{(n)}) \geq \delta$ **do**
  Sample $(t^{(j)},m^{(j)},p^{(j)})_{j=1,\dots,M} \in [0,T]\times S_d$.
  $\theta^{(n+1)} \leftarrow \theta^{(n)} - \alpha^{(n)} \nabla_\theta \tilde{G}((t^{(j)},m^{(j)},p^{(j)})_{j=1,\dots,M},\theta^{(n)})$
  $n \leftarrow n+1$
end while

In practice, the performance of the above algorithm may vary depending on the sample size M at each step. Additionally, the learning rate schedule $\alpha^{(n)}$ may determine the convergence rate of the algorithm as before. Using an optimizer such as AdaGrad or Adam may help speed up convergence. Finally, instead of using a tolerance $\delta \in (0,1)$ to determine the convergence of the algorithm, one may instead specify a fixed number of SGD iterations to carry out.

Remark 3.1. As noted above, we approximate that maxima over $[0,T] \times S_d$ and S_d respectively by uniformly sampling M points in each region. Now, denote

$$(t^\star, m_1^\star) := \operatorname*{argmax}_{(t,m)\in[0,T]\times S_d} \left| -\partial_t \varphi(t,m;\theta) + \sum_{i\in[\![d]\!]} m_i H^i(t,m,D^i\varphi(t,m;\theta)) \right|$$

and

$$m_2^\star := \operatorname*{argmax}_{m\in S_d} \left| \varphi(T,m;\theta) - \sum_{i\in[\![d]\!]} m_i g^i(m) \right|,$$

Similarly, let

$$j_1^\star := \operatorname*{argmax}_{j=1,\dots,M} \left| -\partial_t \varphi(t^{(j)},m^{(j)};\theta) + \sum_{i\in[\![d]\!]} m_i^{(j)} H^i(t^{(j)},m^{(j)},D^i\varphi(t^{(j)},m^{(j)};\theta)) \right|$$

and

$$j_2^\star := \operatorname*{argmax}_{j=1,\dots,M} \left| \varphi(T,p^{(j)};\theta) - \sum_{i\in[\![d]\!]} p_i^{(j)} g^i(p^{(j)}) \right|,$$

By the law of large numbers, we have that

$$\mathbb{E}\left|t^\star - t_{j_1^\star}\right| + \mathbb{E}|m_1^\star - m_{j_1^\star}| + \mathbb{E}|m_2^\star - m_{j_2^\star}| = \mathcal{O}(M^{-1}).$$

Thus, by taking the sample size M larger, we can approximate the maximizers of (23) via (24).

4 Universal Approximation with Two-Layer Feedforward Neural Networks

In this section, we provide a brief, non-comprehensive review of universal approximation theorems, with the goal of summarizing results relevant to the DGM. By a "two-layer" feedforward neural network, we refer to a neural network with one input layer and *one* hidden layer as defined by (25). Although such networks are relatively simple compared modern deep neural networks, their universal approximation power is sufficient for the purposes of the convergence result in Section 5. Theorem 5.1 is just one of many results that describe the approximation power of this class of neural networks; see [11] for a similar survey, albeit with a focus on implementation.

Universal approximation results relevant to this paper fall broadly into two categories: approximations in L^p spaces for $p < \infty$ (or in $W^{k,p}$ spaces if the approximation of derivatives is also desired) and approximations in $\mathcal{C}^k$ for $k \geq 0$. We also discuss bounds on the weights of neural networks and the impact that this has on their approximation power. Indeed, in order to establish equicontinuity as in Theorem 5.12, we require approximation by a sequence of neural networks with bounded, summable weights. In the case of bounded neural networks, however, approximation of a function *and* its derivatives does not seem to be covered by the current literature.

Modern universal approximation results largely stem from those established by Hornik, Stinchcombe, White, and Cybenko in the early 1990s. In particular, the former three authors first showed that neural networks with (possibly discontinuous) "squashing function" activations are uniformly dense on compact sets in $\mathcal{C}(\mathbb{R})$, with respect to the uniform norm [12]. Cybenko then showed that neural networks with sigmoidal activations share the same approximation power [13]. Hornik, Stinchombe, and White then extended the result of Cybenko to neural networks with non-sigmoidal (indeed, bounded and nonconstant) activations, also providing estimates of the derivatives of an unknown function $f \in W^{k,p}(\Omega)$ for $k \geq 1$ and $p \geq 1$, with $\Omega \subset \mathbb{R}^n$ compact [14]. Theorem 5.1 is due to Hornik [15], who subsequently showed that functions with k continuous derivatives can be uniformly approximated on compact sets in $\mathcal{C}^k$-norm in addition to the standard norm on $W^{k,p}$ [15]. Out of the universal approximation theorems shown in the early 1990s, Hornik's result is the most general. However,

his proof, relying on the Hahn–Banach theorem, is not constructive and does not extend to neural networks with bounded weights. Conversely, Stinchcombe and White showed that continuous functions can be approximated, in uniform norm, by neural networks with bounded weights using an argument based on the Stone–Weierstrass theorem [16].

Although the vast majority of results concerning universal approximation by feedforward neural networks were stated and proved in the early 1990s, several authors have made recent attempts to extend classical results to more complicated architectures, more general convergence guarantees, and provide explicit constructions of universal approximators. For instance, Mhaskar and Micchelli show in [17] that functions and their derivatives can be uniformly approximated in $L^p([-1,1]^d)$ for $p \geq 1$ using a constructive, Fourier-analytic approach via approximation by periodic functions.

Additionally, there has been recent interest in exploring the approximation power of neural networks with specific activation functions such as hyperbolic tangent neural networks. For example, [18] considers the approximation power of tanh neural networks in the standard norm on $W^{k,p}(\Omega)$ for a compact domain $\Omega \subset \mathbb{R}^d$, deducing asymptotic bounds on the weights of the neural network and much more explicit bounds on the approximation power of a network with a fixed number of hidden units. In a similar vein, [19] derive promising results for networks with piecewise quadratic activations (specifically ReQU neural networks), demonstrating that ReQU networks with bounded weights *can* approximate functions and their derivatives in Hölder norms. In particular, [19] shows that if $f \in \mathcal{C}^{2,\alpha}([0,1]^d)$ for $\alpha \in (0,1]$, then for any $\varepsilon > 0$, there exists a *deep* neural network φ_f with weights in $[-1,1]$ such that $\|f - \varphi_f\|_{\mathcal{C}^{2,\alpha}([0,1]^d)} < \varepsilon$. This construction, however, is not currently considered in the proof in Section 5, as we limit ourselves to smooth activation functions here.

5 Convergence of the DGM

This section aims to establish the following two results in the context of the HJB equation for the MFCP. We employ the following outline to obtain our existence and convergence results.

(1) By an appropriate version of the universal approximation theorem, we can arbitrarily approximate $V \in \mathcal{C}^{1,1}([0,T] \times S_d)$ with neural network approximators; see Theorem 5.1. Moreover, there exists a sequence of neural network approximators $\{\varphi^n(t,m;\theta)\}_{n\in\mathbb{N}}$ such that both $L(\varphi^n)$ and $\tilde{L}(\varphi^n) \to 0$ as $n \to \infty$; see Theorem 5.2

(2) If $\tilde{L}(\varphi^n) \to 0$ as $n \to \infty$ for a sequence of neural network approximators $\{\varphi^n(t,m;\theta)\}_{n\in\mathbb{N}}$, then $\varphi^n \to V$ uniformly on $[0,T] \times S_d$; see Theorem 5.8.

5.1 Approximation Via Two-Layer Neural Networks

In this section, we utilize the universial approximation result shown by Hornik in [15]. For some context, we operate in the setting of a two-layer feedward network with real-valued outputs. Specifically, denote any set of weights by $\theta := (\beta_1, \ldots, \beta_n, \alpha_{1,1}, \ldots, \alpha_{d+1,n}, c_1, \ldots, c_n) \in \mathbb{R}^{2n+n(d+1)}$. The class of such networks with n hidden units and common activation function σ, aiming to approximate an arbitrary mapping $f : \mathbb{R}^{d+1} \to \mathbb{R}$, is denoted by:

$$\mathfrak{C}_{d+1}^{(n)}(\sigma) := \left\{ \varphi : \mathbb{R}^{d+1} \to \mathbb{R} \;\middle|\; \varphi(t,x;\theta) = \sum_{i=1}^{n} \beta_i \sigma\left(\alpha_{1,i} t + \sum_{j=1}^{d} \alpha_{j+1,i} x_j + c_i \right),\ \theta \in \mathbb{R}^{2n+n(d+1)} \right\}. \tag{25}$$

Each network in the above class has weights given by $\theta = (\beta_1, \ldots, \beta_n, \alpha_{1,1}, \ldots, \alpha_{d+1,n}, c_1, \ldots, c_n) \in \mathbb{R}^{2n+n(d+1)}$. As in [15], we then denote

$$\mathfrak{C}_{d+1}(\sigma) := \bigcup_{n=1}^{\infty} \mathfrak{C}_{d+1}^{(n)}(\sigma).$$

With the above notation in mind, we can apply the universal approximation theorem [15, Theorem 3], stated as follows:

Theorem 5.1. *If $\sigma \in \mathcal{C}^m(\mathbb{R})$ is nonconstant and bounded, then $\mathfrak{C}_{d+1}$ is uniformly m-dense on compact sets in $\mathcal{C}^m(\mathbb{R}^{d+1})$. In particular, for all $f \in \mathcal{C}^m(\mathbb{R}^{d+1})$, all compact subsets $K \subset \mathbb{R}^{d+1}$, and any $\varepsilon > 0$, there exists $\varphi = \varphi(f, K, \varepsilon) \in \mathfrak{C}_{d+1}$ such that $\|f - \varphi\|_{\mathcal{C}^m(K)} < \varepsilon$.*

As is standard, we use the definition

$$\|f\|_{\mathcal{C}^m(K)} := \max_{|\alpha| \leq m} \sup_{x \in K} |D^\alpha f(x)|.$$

In the implementation in Section 6, we take $\sigma(y) = \tanh(y)$, a typical choice of activation function that is smooth, nonconstant, and bounded and therefore satisfies all the criteria of Theorem 5.1. Note also that with this choice of σ, any element of $\mathfrak{C}_{d+1}(\sigma)$ is smooth (as a linear combination of smooth functions), ensuring that any $\varphi \in \mathfrak{C}_{d+1}(\sigma)$ has Lipschitz-continuous first derivative. See [18] for further justification of this choice of activation function in terms of the approximation guarantees that it brings.

With the above background in mind, we move towards approximating solutions to the HJB equation for the MFCP using the DGM. Recall that we aim to approximate solutions to the following HJB equation, which describes the value function for the MFCP.

$$\begin{aligned} -\partial_t V(t,m) + \sum_{i \in [\![d]\!]} m_i H^i(t, m, D^i V(t,m)) &= 0, \\ V(T,m) = \sum_{i \in [\![d]\!]} m_i g^i(m), & \end{aligned}$$

where $(D^i V(t,m))_j = \partial_{m_j - m_i} V(t,m)$ as above. Throughout this section, we use the notation $\mathcal{C}^{1,1}([0,T] \times S_d)$ to denote (continuously-differentiable) functions that have Lipschitz-continuous first derivatives; see [3] for a more thorough discussion of this class of functions as they relate to the HJB equation in Equation (7). Through assumptions (A) – (C) in [3], we have the following useful properties for proving convergence of DGM to the solution of the HJB equation.

(1) Under assumption (A) in [3], the Hamiltonian in Equation (7) is Lipschitz continuous in all three arguments. Specifically, we have that

$$\begin{aligned} \left| \sum_{i \in [\![d]\!]} m_i H^i(t_1, m_1, z_1) - \sum_{i \in [\![d]\!]} m_i H^i(t_2, m_2, z_2) \right| &\leq L|(t_1 - t_2, m_1 - m_2, z_1 - z_2)| \\ &\leq L(|t_1 - t_2| + |m_1 - m_2| + |z_1 - z_2|) \end{aligned}$$

for some constant $L > 0$, where $|\cdot|$ denotes the Euclidean norm. In fact, we only need the following, which also follows from the assumptions presented in [3]: for each $i \in [\![d]\!]$ and fixed $(t,m) \in [0,T] \times S_d$, the map $p \mapsto H^i(t,m,p)$ is Lipschitz continuous with common Lipschitz constant $L > 0$.

Proof. By the definition of the Hamiltonian provided in [3], we have that

$$\begin{aligned} H^i(t,m,p) - H^i(t,m,p') &= \max_{a \in A}(-\langle Q_{i,\bullet}(t,a,m), p\rangle - f(t,i,a,m)) - \max_{a \in A}(-\langle Q_{i,\bullet}(t,a,m), p'\rangle - f(t,i,a,m)) \\ &\leq \max_{a \in A} |-\langle Q_{i,\bullet}(t,a,m), p\rangle + \langle Q_{i,\bullet}(t,a,m), p'\rangle| \\ &= \max_{a \in A} |\langle Q_{i,\bullet}(t,a,m), p' - p\rangle| \\ &\leq \max_{a \in A} |Q_{i,\bullet}(t,a,m)||p' - p| \\ &\leq C|p' - p| \end{aligned}$$

by applying the Cauchy–Schwarz inequality in the penultimate line. In the last line, we note that under assumption (A), the transition rate $Q_{i,j} : [0,T] \times A \times S_d \to \mathbb{R}$ is uniformly continuous and bounded so that for all $i \in [\![d]\!]$,

$$|Q_{i,\bullet}(t,a,m)| = \left(\sum_{j=1}^{d} Q_{i,j}(t,a,m)^2 \right)^{1/2} \leq \sqrt{d} \max_{i,j \in [\![d]\!]} \max_{(t,a,m) \in [0,T] \times A \times S_d} Q_{i,j}(t,a,m) := C$$

is bounded above uniformly for all $i \in [\![d]\!]$ by some constant $C > 0$. Switching the roles of p and p' above completes the proof. □

(2) By [3, Theorem 9], Equation (7) admits a unique classical solution $V \in \mathcal{C}^{1,1}([0,T] \times S_d)$.

Finally, recall that the original DGM algorithm aims to minimize the L^2-error of the approximate solution to the PDE in question. Specifically, DGM learns an approximator $\varphi(t, m; \theta)$, parametrized by θ, by minimizing the L^2-error of the HJB equation:

$$L(\varphi) := \left\| -\partial_t \varphi(t,m;\theta) + \sum_{i \in [\![d]\!]} m_i H^i(t, m, D^i \varphi(t,m;\theta)) \right\|^2_{2,[0,T]\times S_d, \nu_1} + \|\varphi(T,m;\theta) - \sum_{i \in [\![d]\!]} m_i g^i(m)\|^2_{2, S_d, \nu_2}.$$

Above, ν_1 and ν_2 are probability densities on $[0,T] \times S_d$ and S_d respectively. Furthermore, we show below that by utilizing the uniform error, given by

$$\tilde{L}(\varphi) = \max_{(t,m)\in[0,T]\times S_d} \left| -\partial_t \varphi(t,m;\theta) + \sum_{i \in [\![d]\!]} m_i H^i(t, m, D^i \varphi(t,m;\theta)) \right| + \max_{m \in S_d} \left| \varphi(T,m;\theta) - \sum_{i \in [\![d]\!]} m_i g^i(m) \right|,$$

we can obtain the desired convergence result. With the ultimate goal of showing that a two-layer neural network can approximate the value function $V(t,m)$ on $[0,T] \times S_d$ arbitrarily well in the uniform norm by taking n sufficiently large, we first show that following intermediate result, which holds for both the original and the modified DGM loss functions. For simplicity below, we define the operator $\mathcal{L}$ by

$$\mathcal{L}[\varphi](t,m) := -\partial_t \varphi(t,m) + \sum_{i \in [\![d]\!]} m_i H^i(t, m, D^i \varphi(t,m;\theta)).$$

The following theorem presents our main existence result, establishing the existence of a sequence of two-layer neural networks that can make either the original or modified DGM loss arbitrarily small.

Theorem 5.2. *Let $\sigma \in \mathcal{C}^1(\mathbb{R})$ be bounded and nonconstant. For every $\varepsilon > 0$, there exist constants $K = K(d,T,C) > 0$ and $\tilde{K} = \tilde{K}(d,T,C) > 0$, where d is the dimension of the simplex S_d, T is the finite time horizon of the mean field control problem, and C is the Lipschitz constant of the Hamiltonian in Equation (7), such that for some $\varphi \in \mathfrak{C}_{d+1}(\sigma)$, the L^2-error satisfies $L(\varphi) \le K\varepsilon^2$ and $\tilde{L}(\varphi) \le \tilde{K}\varepsilon$.*

Proof. Note that $\Omega_T := [0,T] \times S_d$ is a compact set in $\mathbb{R}^{d+1}$. Thus, by Theorem 5.1 above, we know that for the unique solution $V \in \mathcal{C}^{1,1}(\Omega_T)$ to Equation (7) and any $\varepsilon > 0$, there exists $\varphi \in \mathfrak{C}_{d+1}(\sigma)$ such that

$$\sup_{(t,m)\in\Omega_T} |V(t,m) - \varphi(t,m;\theta)| + \sup_{(t,m)\in\Omega_T} |\partial_t V(t,m) - \partial_t \varphi(t,m;\theta)| + \sup_{(t,m)\in\Omega_T} |\nabla_m V(t,m) - \nabla_m \varphi(t,m;\theta)| < \varepsilon \tag{26}$$

For such $\varphi \in \mathfrak{C}_{d+1}(\sigma)$, the Lipschitz continuity of the Hamiltonian in Equation (7), we observe that

$$\begin{aligned}
&\int_{\Omega_T} \left| \sum_{i \in [\![d]\!]} m_i H^i(t,m,D^i\varphi(t,m;\theta)) - \sum_{i \in [\![d]\!]} m_i H^i(t,m,D^i V(t,m)) \right|^2 d\nu_1(t,m) \\
&= \int_{\Omega_T} \left| \sum_{i \in [\![d]\!]} m_i (H^i(t,m,D^i\varphi(t,m;\theta)) - H^i(t,m,D^i V(t,m))) \right|^2 d\nu_1(t,m) \\
&\le d \sum_{i \in [\![d]\!]} \int_{\Omega_T} \left| m_i (H^i(t,m,D^i\varphi(t,m;\theta)) - H^i(t,m,D^i V(t,m))) \right|^2 d\nu_1(t,m) \\
&\le dC^2 \sum_{i \in [\![d]\!]} \int_{\Omega_T} |D^i\varphi(t,m;\theta) - D^i V(t,m)|^2 d\nu_1(t,m).
\end{aligned} \tag{27}$$

Above, we apply the Cauchy–Schwarz inequality in the second-to-last line above and note that $|m_i| \le 1$ for any $m \in S_d$. Now, for each $i \in [\![d]\!]$ and any $(t,m) \in \Omega_T$, observe that by denoting the standard basis of $\mathbb{R}^d$ by $\{e_i\}_{i\in[\![d]\!]}$, we have that

$$|D^i\varphi(t,m;\theta) - D^i V(t,m)|^2 = \frac{1}{2} \sum_{j=1}^{d} |(\nabla_m \varphi(t,m;\theta) - \nabla_m V(t,m)) \cdot (e_j - e_i)|^2$$

$$\leq \frac{1}{2}\sum_{j=1}^{d} |\nabla_m V(t,m) - \nabla_m \varphi(t,m;\theta)|^2 |e_j - e_i|^2$$
$$\leq d|\nabla_m V(t,m) - \nabla_m \varphi(t,m;\theta)|^2,$$

again by applying the Cauchy–Schwarz inequality. In turn, we can bound (27) by

$$dC^2 \sum_{i\in[\![d]\!]} \int_{\Omega_T} |D^i\varphi(t,m;\theta) - D^i V(t,m)|^2 d\nu_1(t,m) \leq d^3C^2 \int_{\Omega_T} |\nabla_m V(t,m) - \nabla_m \varphi(t,m;\theta)|^2 d\nu_1(t,m)$$
$$\leq K\varepsilon^2$$

for some positive constant $K = K(d,T,C) > 0$ by the construction of φ. Finally, because the value function V satisfies $\mathcal{L}[V](t,m) = 0$ for all $(t,m) \in \Omega_T$ (in addition to the terminal condition of Equation (7)), this allows us to conclude that

$$\begin{aligned}
L(\varphi) &= \|\mathcal{L}[\varphi](t,m)\|^2_{2,\Omega_T,\nu_1} + \|\varphi(T,m;\theta) - \sum_{i\in[\![d]\!]} m_i g^i(m)\|^2_{2,S_d,\nu_2} \\
&= \|\mathcal{L}[\varphi](t,m) \ - \mathcal{L}[V](t,m)\|^2_{2,\Omega_T,\nu_1} + \|\varphi(T,m;\theta) - V(T,m)\|^2_{2,S_d,\nu_2} \\
&\leq 2\int_{\Omega_T} \left| \sum_{i\in[\![d]\!]} m_i H^i(t,m,D^i\varphi(t,m;\theta)) - \sum_{i\in[\![d]\!]} m_i H^i(t,m,D^iV(t,m)) \right|^2 d\nu_1(t,m) \\
&\quad + 2\int_{\Omega_T} |\partial_t V(t,m) - \partial_t\varphi(t,m;\theta)|^2 \, d\nu_1(t,m) \\
&\quad + \int_{S_d} |\varphi(T,m;\theta) - V(T,m)|^2 d\nu_2(m) \\
&\leq K\varepsilon^2
\end{aligned}$$

by applying the Cauchy–Schwarz inequality yet again, taking K larger if necessary, and noting that the estimate in (26) provides bounds on the two remaining terms in the above expression.

On the other hand, the same estimate as in (27) shows that

$$\max_{(t,m)\in\Omega_T} \left| m_i H^i(t,m,D^i\varphi(t,m;\theta)) - \sum_{i\in[\![d]\!]} m_i H^i(t,m,D^iV(t,m)) \right| \leq C \max_{(t,m)\in\Omega_T} |D^i\varphi(t,m;\theta) - D^iV(t,m)||$$
$$\leq \tilde{K}|\nabla_m V(t,m) - \nabla_m\varphi(t,m;\theta)|$$

for some $\tilde{K} = \tilde{K}(d,T,C) > 0$. Consequently, we observe that

$$\begin{aligned}
\tilde{L}(\varphi) &= \max_{(t,m)\in\Omega_T} |\mathcal{L}[\varphi](t,m)| + \max_{m\in S_d} \left| \varphi(T,m;\theta) - \sum_{i\in[\![d]\!]} m_i g^i(m) \right| \\
&= \max_{(t,m)\in\Omega_T} |\mathcal{L}[\varphi](t,m) \ - \mathcal{L}[V](t,m)| + \max_{m\in S_d} |\varphi(T,m;\theta) - V(T,m)| \\
&\leq \max_{(t,m)\in\Omega_T} \left| m_i H^i(t,m,D^i\varphi(t,m;\theta)) - \sum_{i\in[\![d]\!]} m_i H^i(t,m,D^iV(t,m)) \right| \\
&\quad + \max_{(t,m)\in\Omega_T} |\partial_t V(t,m) - \partial_t\varphi(t,m;\theta)| \\
&\quad + \max_{m\in S_d} |\varphi(T,m;\theta) - V(T,m)| \\
&\leq \tilde{K}\varepsilon
\end{aligned}$$

by applying the approximation result from (26) and taking $\tilde{K} > 0$ larger if necessary. □

Remark 5.3. In the case of the original DGM L^2-error, the measures ν_1 and ν_2, regardless of the densities that they correspond to, are defined as probability measures on $[0,T] \times S_d$ and S_d respectively. Thus, the above result is *independent* of the choice of densities ν_1 and ν_2, as we simply use the bounds

$$\int_{\Omega_T} |\partial_t V(t,m) - \partial_t \varphi(t,m;\theta)|^2 \, d\nu_1(t,m) \leq \varepsilon^2 \nu_1(\Omega_T) = \varepsilon^2$$

and

$$\int_{S_d} |\varphi(T,m;\theta) - V(T,m)|^2 d\nu_2(m) \leq \varepsilon^2 \nu_2(S_d) = \varepsilon^2$$

respectively.

The following proposition, which also relates to the existence of an approximating sequence of neural networks, utilizes the same universal approximation theorem as Theorem 5.2. In particular, we can obtain a sequence of neural networks that satisfies a corresponding sequence of PDE with a measurable error term that uniformly converges to zero.

Proposition 5.4. *There exists a sequence of neural networks $\varphi^n \in \mathcal{C}^{1,1}([0,T] \times S_d)$ such that $\varphi^n \to V$ uniformly as $n \to \infty$, where V is the unique classical solution to Equation (7). Additionally, each φ^n satisfies*

$$\begin{aligned} -\partial_t \varphi^n(t,m) + \sum_{i \in [\![d]\!]} m_i H^i(t,m,D^i\varphi^n(t,m)) &= \varepsilon^n(t,m), \\ \varphi^n(T,m) = \sum_{i \in [\![d]\!]} m_i g^i(m) &+ \varepsilon^n(T,m). \end{aligned} \tag{28}$$

for some measurable $\varepsilon^n : [0,T] \times S_d \to \mathbb{R}$ such that $\|\varepsilon^n\|_\infty \to 0$ as $n \to \infty$.

Proof. From Theorem 5.1, we know that for any $n \in \mathbb{N}$, there exists a neural network $\varphi^n \in \mathfrak{C}_{d+1}(\sigma) \subset \mathcal{C}^{1,1}([0,T] \times S_d)$ such that (26) holds for $\varepsilon = n^{-1}$. Thus, it immediately follows that $\|\varphi^n - V\|_{\mathcal{C}([0,T]\times S)d)} \to 0$ as $n \to \infty$, yielding a sequence φ^n of neural networks that converges uniformly to V, the classical solution to Equation (7). Now, for $(t,m) \in [0,T) \times S_d$, define $\varepsilon^n : [0,T] \times S_d \to \mathbb{R}$ by

$$\varepsilon^n(t,m) := -\partial_t \varphi^n(t,m) + \sum_{i \in [\![d]\!]} m_i H^i(t,m,D^i\varphi^n(t,m)). \tag{29}$$

For $m \in S_d$, instead define

$$\varepsilon^n(T,m) := \varphi^n(T,m) - \sum_{i \in [\![d]\!]} m_i g^i(m).$$

By construction, φ^n satisfies the PDE in (28). Furthermore, because each $\varepsilon^n : [0,T] \times S_d \to \mathbb{R}$ is continuous on both $[0,T) \times S_d$ and $\{T\} \times S_d$, it is evidently measurable.

Finally, to see that $\|\varepsilon^n\|_\infty \to 0$ as $n \to \infty$, we reuse many of the estimates from the proof of Theorem 5.2. In particular, we can write

$$\begin{aligned} \varepsilon^n(t,m) &= -\partial_t \varphi^n(t,m) + \sum_{i \in [\![d]\!]} m_i H^i(t,m,D^i\varphi^n(t,m)) \\ &= \partial_t V(t,m) - \partial_t \varphi^n(t,m) + \sum_{i \in [\![d]\!]} m_i \left[H^i(t,m,D^i\varphi^n(t,m)) - H^i(t,m,D^iV(t,m)) \right], \end{aligned}$$

for $(t,m) \in [0,T) \times S_d$, using the fact that V solves Equation (7). By the fact that $m \in S_d$ and the Lipschitz continuity of H^i for each $i \in [\![d]\!]$, we again have that

$$\left| \sum_{i \in [\![d]\!]} m_i \left[H^i(t,m,D^i\varphi^n(t,m)) - H^i(t,m,D^iV(t,m)) \right] \right| \leq \sum_{i \in [\![d]\!]} \left| H^i(t,m,D^i\varphi^n(t,m)) - H^i(t,m,D^iV(t,m)) \right|$$

$$\leq C \sum_{i \in [\![d]\!]} |D^i \varphi^n(t,m) - D^i V(t,m)|$$
$$\leq 2dC |\nabla_m V(t,m) - \nabla_m \varphi^n(t,m)|$$

Thus, we have that for all $(t,m) \in [0,T) \times S_d$

$$|\varepsilon^n(t,m)| \leq 2dC|\nabla_m V(t,m) - \nabla_m \varphi^n(t,m)| + |\partial_t V(t,m) - \partial_t \varphi^n(t,m)| \leq \frac{2dC}{n} + \frac{1}{n}$$

by the construction of φ^n via (26). Similarly, at the terminal time $t = T$, observe that

$$|\varepsilon^n(T,m)| = \left| \varphi^n(T,m) - \sum_{i \in [\![d]\!]} m_i g^i(m) \right| = |\varphi^n(T,m) - V(T,m)| \leq \frac{1}{n}.$$

Thus, it follows that $\|\varepsilon^n\|_\infty \to 0$ as $n \to \infty$ as claimed. □

Remark 5.5. The above sequence of neural networks satisfies several properties based on its construction. Specifically, note that by the converse of the Arzelà–Ascoli theorem, $\{\varphi^n\}_{n \in \mathbb{N}}$ is uniformly bounded and equicontinuous on $[0,T] \times S_d$. In fact, the construction of φ^n implies that it is uniformly bounded with respect to the standard norm on $\mathcal{C}^1([0,T] \times S_d)$.

Remark 5.6. Below, it will be useful to instead consider the HJB equation for $x \in \widehat{S}_d$, where

$$\widehat{S}_d := \left\{ (x_1, \ldots, x_{d-1}) \in \mathbb{R}^{d-1} : x_j \geq 0 \text{ for all } j = 1, \ldots, d-1, \sum_{j=1}^{d-1} x_j \leq 1 \right\}.$$

Then, the simplex S_d can be expressed as

$$S_d = \left\{ (x, x^{-d}) : x \in \widehat{S}_d,\ x^{-d} = 1 - \sum_{j=1}^{d-1} x_j \right\}.$$

In turn, any function $v \in \mathcal{C}^1(S_d)$ is such that $\widehat{v} \in \mathcal{C}^1(\widehat{S}_d)$, where $v(x, x^{-d}) = \widehat{v}(x)$ and $\nabla_x \widehat{v}(x) = (\partial_{x_j} \widehat{v}(x))_{j=1}^{d-1}$ satisfies $\partial_{x_j} \widehat{v}(x) = \partial_{m_j - m_d} v(m)$. With this modification, the solution to Equation (7) can be written $V(t, x, x^{-d}) = \widehat{V}(t,x)$, where $\widehat{V} \in \mathcal{C}^{1,1}(\widehat{S}_d)$ is the unique solution to the modified HJB equation

$$\begin{aligned} &- \partial_t \widehat{V}(t,x) + \sum_{i \in [\![d-1]\!]} x_i \widehat{H}^i(t, x, \nabla_x \widehat{V}(t,x)) + x^{-d} \widehat{H}^d(t, x, \nabla_x \widehat{V}(t,x)) = 0, \\ &\widehat{V}(T,x) = \sum_{i \in [\![d-1]\!]} x_i g^i(x, x^{-d}) + x^{-d} g^d(x, x^{-d}). \end{aligned} \tag{30}$$

Above, the modified Hamiltonians take inputs in $[0,T] \times \widehat{S}_d \times \mathbb{R}^{d-1}$ and are defined by

$$\widehat{H}^i(t,x,p) = H^i(t, x, x^{-d}, p_1 - p_i, \ldots, p_{d-1} - p_i, -p_i) \quad \text{and} \quad \widehat{H}^d(t,x,p) = H^d(t, x, x^{-d}, p, 0).$$

As shown in [3], Equation (30) has a unique solution $\widehat{V} \in \mathcal{C}^{1,1}([0,T] \times \widehat{S}_d)$. Additionally, $\widehat{S}_d$ is a compact subset of $\mathbb{R}^{d-1}$, allowing us to apply the universal approximation theorem exactly as above, now on $\widehat{S}_d$. By the definitions of the modified Hamiltonians (in terms of the original Hamiltonians), we can also apply the exact same argument as above to obtain a result equivalent to Theorem 5.2 on $\widehat{S}_d$. In particular, observe that for all $i \in [\![d-1]\!]$ and $p, p' \in \widehat{S}_d$, we have that

$$|\widehat{H}^i(t,x,p) - \widehat{H}^i(t,x,p')|^2 = |H^i(t, x, x^{-d}, p_1 - p_i, \ldots, p_{d-1} - p_i, -p_i) - H^i(t, x, x^{-d}, p'_1 - p'_i, \ldots, p'_{d-1} - p'_i, -p'_i)|^2$$
$$\leq C^2 \left(\sum_{j=1}^{d-1} ((p_j - p_i) - (p'_j - p'_i))^2 + (p_i - p'_i)^2 \right)$$

$$
\begin{aligned}
&\leq C^2 \left(2\sum_{j=1}^{d-1}(p_j - p'_j)^2 + (2(d-1)+1)(p_i - p'_i)^2 \right) \\
&\leq C^2(2(d-1)+1) \left(\sum_{j=1}^{d-1}(p_j - p'_j)^2 + (p_i - p'_i)^2 \right) \\
&= D^2|p-p'|^2,
\end{aligned}
$$

where $D^2 = 2C^2(2(d-1)+1)$. This shows that $\widehat{H}^i$ is Lipschitz continuous in p with Lipschitz constant $D \geq C > 0$ for $i \in [\![d-1]\!]$, and we similarly observe that

$$
|\widehat{H}^d(t,x,p) - \widehat{H}^d(t,x,p')| = |H^d(t,x,x^{-d},p,0) - H^d(t,x,x^{-d},p',0)| \leq C|p-p'| \leq D|p-p'|
$$

so that $\widehat{H}^i$ is Lipschitz continuous in p, with common Lipschitz constant $D > 0$, for all $i \in [\![d]\!]$. From this, the proof of a modified version of Theorem 5.2, now on $[0,T] \times \widehat{S}_d$, can proceed exactly as before. Note that the original value function defined on the simplex can be recovered via $V(t,x,x^{-d}) = \widehat{V}(t,x)$ for $x \in \widehat{S}_d$.

For the remainder of the paper, we only consider the modified DGM algorithm with uniform loss.

5.2 Convergence of Neural Network Approximators to Value Function

We now discuss convergence of a sequence of neural network approximators φ^n to V, the unique classical solution of the HJB equation in Equation (7). With the ultimate goal of establishing *uniform* convergence of the neural network approximators φ^n to the value function V, we rely on the theory of viscosity solutions to first-order nonlinear PDE. This powerful theory, developed by Crandall, Evans, and Lions in the 1980s for the explicit purpose of approaching HJB equations (which often lack classical, differentiable solutions) [10], will allow us to relate the neural network approximators $\varphi^n(t,m;\theta)$ to the value function V via a sequence of first-order nonlinear PDE. Then, using a version of the comparison principle for viscosity solutions, we obtain the desired convergence.

First, however, we must reframe the problem in terms of Equation (30), which possesses a unique classical solution $\widehat{V} \in \mathcal{C}^{1,1}([0,T] \times \widehat{S}_d)$. Recall that $\widehat{S}_d \subset \mathbb{R}^{d-1}$ is the preimage of the simplex in $\mathbb{R}^d$ under the chart introduced in Remark 5.6. Working with Equation (30) rather than Equation (7) allows us to cite results from the theory of viscosity solutions that require the domain of the relevant PDE to be open; note that $\text{Int}(\widehat{S}_d)$ is an open subset of $\mathbb{R}^{d-1}$ whereas S_d has empty interior in $\mathbb{R}^d$. Additionally, the following result demonstrates that the convergence result on $\widehat{S}_d$ translates to S_d without any issues.

Proposition 5.7. *Assume that $\widehat{V} \in \mathcal{C}([0,T] \times \widehat{S}_d)$ and $\widehat{\varphi}^n \in \mathcal{C}([0,T] \times \widehat{S}_d)$ are such that $\|\widehat{V} - \widehat{\varphi}^n\|_\infty \to 0$ as $n \to \infty$. Then, $\|V - \varphi^n\|_\infty \to 0$ as $n \to \infty$, where $V, \varphi^n \in \mathcal{C}([0,T] \times S_d)$ are given by $V(t,x,x^{-d}) = \widehat{V}(t,x)$ and $\varphi^n(t,x,x^{-d}) = \widehat{\varphi}^n(t,x)$ for all $(t,x,x^{-d}) \in [0,T] \times S_d$ and all $n \in \mathbb{N}$.*

Proof. This is a simple consequence of the definition of $\widehat{S}_d$. Indeed, if $\|\widehat{V} - \widehat{\varphi}^n\|_\infty \to 0$ as $n \to \infty$. Then, for any $\varepsilon > 0$, we have that for all $n \in \mathbb{N}$ sufficiently large,

$$
\sup_{(t,x) \in [0,T] \times \widehat{S}_d} |\widehat{V}(t,x) - \widehat{\varphi}^n(t,x)| < \varepsilon.
$$

Consequently for all $n \in \mathbb{N}$ sufficiently large, we have that

$$
\begin{aligned}
\sup_{(t,x,x^{-d}) \in [0,T] \times S_d} |V(t,x,x^{-d}) - \varphi^n(t,x,x^{-d})| &= \sup_{(t,x,x^{-d}) \in [0,T] \times S_d} |\widehat{V}(x) - \widehat{\varphi}^n(t,x)| \\
&= \sup_{(t,x) \in [0,T] \times \widehat{S}_d} |\widehat{V}(t,x) - \widehat{\varphi}^n(t,x)| \\
&< \varepsilon.
\end{aligned}
$$

This means precisely that $\|V - \varphi^n\|_\infty \to 0$ as $n \to \infty$. □

As a consequence of the above proposition, it suffices to show the uniform convergence of a sequence of neural network approximators $\widehat{\varphi}^n$ to the unique classical solution $\widehat{V}$ of Equation (30) on $\widehat{S}_d$, as we can then recover uniform convergence on the simplex. From Remark 5.6, we also know that Theorem 5.2 holds on $\widehat{S}_d$, yielding the existence of a sequence of neural networks $\{\widehat{\varphi}^n(t,x;\theta)\}_{n\in\mathbb{N}}$ such that $\tilde{L}(\widehat{\varphi}^n)\to 0$ as $n\to\infty$. In turn, each network $\widehat{\varphi}^n(t,x;\theta)$ satisfies its own "perturbed" PDE, of the form

$$\begin{cases}\mathcal{L}[\widehat{\varphi}^n](t,x)=\varepsilon^n(t,x) & (t,x)\in[0,T]\times\widehat{S}_d,\\ \widehat{\varphi}^n(T,x;\theta)=\widehat{G}^n(x) & x\in\widehat{S}_d,\end{cases} \tag{31}$$

where $\widehat{G}^n(x):=\widehat{\varphi}^n(T,x;\theta)$ for all $x\in\widehat{S}_d$. For notational simplicity, we take

$$\widehat{G}(x):=\sum_{i\in[\![d-1]\!]}x_i g^i(x,x^{-d})+x^{-d}g^d(x,x^{-d})$$

in this section to denote the terminal condition of the HJB equation on $\widehat{S}_d$, and the operator $\mathcal{L}$ is given by

$$\mathcal{L}[\widehat{\varphi}](t,x):=-\partial_t\widehat{\varphi}(t,x)+\sum_{i\in[\![d-1]\!]}x_i\widehat{H}^i(t,x,\nabla_x\widehat{\varphi}(t,x))+x^{-d}\widehat{H}^d(t,x,\nabla_x\widehat{\varphi}(t,x)).$$

Denoting $\widehat{\Omega}_T:=[0,T]\times\widehat{S}_d$ as in the previous section, Theorem 5.2 above implies that Equation (31) satisfies

$$\max_{(t,x)\in[0,T]\times\widehat{S}_d}|\varepsilon^n(t,x)|+\max_{x\in\widehat{S}_d}\left|\widehat{G}^n(x)-\widehat{G}(x)\right|\to 0$$

as $n\to\infty$. With this context in mind, we state the primary convergence result of this section.

Theorem 5.8. *The family of neural network approximators $\{\widehat{\varphi}^n(t,x;\theta)\}_{n\in\mathbb{N}}$ satisfying Equation (31) converges uniformly to $\widehat{V}\in\mathcal{C}^{1,1}([0,T]\times\widehat{S}_d)$, the unique classical solution of Equation (30), in the sense that*

$$\sup_{(t,x)\in[0,T]\times\widehat{S}_d}|\widehat{\varphi}^n(t,x)-\widehat{V}(t,x)|\to 0$$

as $n\to\infty$.

To prove the above theorem, we argue via the comparison principle for viscosity solutions to (7) presented in [3]. To this end, we require a suitable definition of viscosity solutions of Equation (30) on $\widehat{S}_d$.

Definition 5.9. *A function $\widehat{v}\in\mathcal{C}((0,T)\times Int(\widehat{S}_d))$ is:*

(i) a viscosity subsolution of Equation (30) if for any $\widehat{\varphi}\in\mathcal{C}^1((0,T)\times Int(\widehat{S}_d))$,

$$-\partial_t\widehat{\varphi}(t_0,x_0)+\sum_{i\in[\![d-1]\!]}x_i\widehat{H}^i(t_0,x_0,\nabla_x\widehat{\varphi}(t_0,x_0))+x^{-d}\widehat{H}^d(t_0,x_0,\nabla_x\widehat{\varphi}(t_0,x_0))\leq 0$$

for every local maximum $(t_0,x_0)\in(0,T)\times Int(\widehat{S}_d)$ of $\widehat{v}-\widehat{\varphi}$ on $(0,T)\times Int(\widehat{S}_d)$.

(ii) a viscosity supersolution of Equation (30) if for any $\widehat{\varphi}\in\mathcal{C}^1((0,T)\times Int(\widehat{S}_d))$,

$$-\partial_t\widehat{\varphi}(t_0,x_0)+\sum_{i\in[\![d-1]\!]}x_i\widehat{H}^i(t_0,x_0,\nabla_x\widehat{\varphi}(t_0,x_0))+x^{-d}\widehat{H}^d(t_0,x_0,\nabla_x\widehat{\varphi}(t_0,x_0))\geq 0$$

for every local minimum $(t_0,x_0)\in(0,T)\times Int(\widehat{S}_d)$ of $\widehat{v}-\widehat{\varphi}$ on $(0,T)\times Int(\widehat{S}_d)$.

(iii) a viscosity solution of Equation (30) if $\widehat{v}$ is both a viscosity subsolution and viscosity supersolution.

Remark 5.10. When viscosity solutions are introduced in [3], the author also allows for test functions on $[0,T)\times S_d$ (resp. $[0,T)\times\widehat{S}_d$), noting that $[0,T)\times S_d$ (resp. $[0,T)\times\widehat{S}_d$) is no longer an open subdomain of $\mathbb{R}^{d+1}$ (resp. $\mathbb{R}^d$). However, in order to utilize [10, Theorem 3.3], the standard comparison principle for viscosity solutions, we must consider viscosity solutions on open subdomains of $\mathbb{R}^d$. As noted in [3], it is also not immediately clear that a classical solution to Equation (7) is a viscosity solution if the latter is defined on a closed set.

We could alternatively cite the comparison principle from [3, Theorem 3.4] that utilizes the definition of viscosity solutions on closed sets presented therein. However, in order to utilize the clearly-presented stability properties of viscosity solutions under uniform limits presented in [10, 20], we opt for the standard definition in Definition 5.9.

In order to establish Theorem 5.8, we proceed using a standard comparison principle argument for viscosity solutions that also leverages the fact that $\widehat{V} \in \mathcal{C}^{1,1}([0,T] \times \widehat{S}_d)$ is the unique viscosity solution to Equation 30 from [3].

Proof of Theorem 5.8. For each $n \in \mathbb{N}$, we may define an operator

$$\mathcal{L}^n[\widehat{\varphi}](t,x) := -\partial_t \widehat{\varphi}(t,x) + \sum_{i \in [\![d-1]\!]} x_i \widehat{H}^i(t,x,\nabla_x \widehat{\varphi}(t,x)) + x^{-d} \widehat{H}^d(t,x,\nabla_x \widehat{\varphi}(t,x)) - \varepsilon^n(t,x),$$

corresponding to the sequence of PDE described in (31). Because $\mathcal{L}^n[\widehat{\varphi}]$ depends only on the derivatives of $\widehat{\varphi}$ (and not on $\widehat{\varphi}$ itself), we observe that $\mathcal{L}^n$ is *proper* in the sense of [10]. This fact also ensures that the technical conditions preceding the comparison principle [10, Theorem 3.3] are satisfied.

Now, note that $\max_{(t,x) \in [0,T] \times \widehat{S}_d} |\varepsilon^n(t,x)| \to 0$ as $n \to \infty$, meaning that ε^n converges uniformly to zero on $[0,T] \times \widehat{S}_d$. Now, for each $n \in \mathbb{N}$, define $T^n : [0,T] \times \widehat{S}_d \times \mathbb{R} \times \mathbb{R}^{d-1} \to \mathbb{R}$ by

$$T^n(t,x,p_0,p) := -p_0 + \sum_{i \in [\![d-1]\!]} x_i \widehat{H}^i(t,x,p) + p^{-d} \widehat{H}^d(t,x,p) - \varepsilon^n(t,x).$$

We then have that T^n converges uniformly on $[0,T] \times \widehat{S}_d \times \mathbb{R} \times \mathbb{R}^{d-1}$ to

$$T(t,x,p_0,p) := -p_0 + \sum_{i \in [\![d-1]\!]} x_i \widehat{H}^i(t,x,p) + p^{-d} \widehat{H}^d(t,x,p).$$

These definitions are motivated by the fact that Equation (7) can be written succinctly as

$$T(t,x,\partial_t \widehat{\varphi}, \nabla_x \widehat{\varphi}) = 0,$$

while Equation (31) is given by

$$T^n(t,x,\partial_t \widehat{\varphi}, \nabla_x \widehat{\varphi}) = 0$$

for each $n \in \mathbb{N}$. Now, following [10, Remark 6.3], we note that because $\widehat{\varphi}^n$ is a classical solution (and therefore a viscosity solution) to the equation $T^n(t,x,\partial_t \widehat{\varphi}, \nabla_x \widehat{\varphi}) = 0$ on $(0,T) \times \mathrm{Int}(\widehat{S}_d)$, then

$$\overline{V}(t,x) := \lim_{j \to \infty} \sup\{\widehat{\varphi}^n(s,y) : n \geq j,\ (t,x) \in [0,T) \times \widehat{S}_d,\ |(s,y)-(t,x)| \leq 1/j\}$$

is a viscosity subsolution to the equation $T(t,x,\partial_t \widehat{\varphi}, \nabla_x \widehat{\varphi}) = 0$, as we have that

$$T(t,x,p_0,p) = \liminf_{n \to \infty} T^n(t,x,p,p_0).$$

On the other hand, we also observe that

$$\underline{V}(t,x) := \lim_{j \to \infty} \inf\{\widehat{\varphi}^n(s,y) : n \geq j,\ (t,x) \in [0,T) \times \widehat{S}_d,\ |(s,y)-(t,x)| \leq 1/j\}$$

is a viscosity supersolution to the equation $T(t,x,\partial_t \widehat{\varphi}, \nabla_x \widehat{\varphi}) = 0$ by the same reasoning. By construction, observe that $\underline{V} \leq \overline{V}$ on $[0,T) \times \widehat{S}_d$. Note also that both $\underline{V}$ and $\overline{V}$ are well-defined on $\{0\} \times \partial \widehat{S}_d$ by their construction. However, by the comparison principle presented in [10, Theorem 3.3], the fact that $\underline{V}$ is a viscosity supersolution and $\overline{V}$ is a viscosity subsolution is sufficient to conclude that $\overline{V} \leq \underline{V}$ on $[0,T) \times \widehat{S}_d$, observing that the comparison principle still holds the closure of the domain $(0,T) \times \mathrm{Int}(\widehat{S}_d)$.

In particular, $\overline{V} = \underline{V}$ is a viscosity solution. As shown in [3, Theorem 9], Equation (7) has a unique viscosity solution $\widehat{V}$, showing that $\overline{V} = \underline{V} = \widehat{V}$. Now, [10, Remark 6.4] implies that $\lim_{n \to \infty} \widehat{\varphi}^n(t,x) = \widehat{V}(t,x)$ uniformly on $[0,T) \times \widehat{S}_d$.

Finally, note that we also have that

$$\max_{x \in \widehat{S}_d} \left| \widehat{G}^n(x) - \widehat{G}(x) \right| = \max_{x \in \widehat{S}_d} \left| \widehat{\varphi}^n(T,x) - \widehat{V}(T,x) \right| \to 0,$$

from the construction of the modified DGM loss, allowing us to conclude uniform convergence of $\widehat{\varphi}^n \to \widehat{V}$ on the entire region $[0,T] \times \widehat{S}_d$ as claimed. □

Remark 5.11. At this point, we can clarify the reasons for the modification to the DGM algorithm made in Section 3. The authors introducing the DGM algorithm in [5] formulated the L^2-error in (22) because of the natural connection between the class of equations that they considered and convergence in L^2. A key step in the proof of their analogue to Theorem 5.8 involves obtaining a uniform bound on $\{\varphi^n(t,m;\theta)\}_{n\in\mathbb{N}}$ in $L^\infty(0,T;L^2(\Omega))\cap L^2(0,T;W_0^{1,2}(\Omega))$, where Ω is the open domain on which the PDE is considered. In turn, this arises from an energy bound on quasilinear parabolic equations such as the one presented in [21], or in more generality in [22]. However, such a bound only holds for equations of the form

$$\begin{cases} \partial_t u - \operatorname{div}(a(t,x,u,\nabla u)) = H(t,x,\nabla u) & (t,x)\in(0,T)\times\Omega, \\ u = 0 & (t,x)\in(0,T)\times\partial\Omega, \\ u(0,x) = u_0(x) & x\in\Omega. \end{cases}$$

that satisfy the Leray-Lions conditions. Namely, there must exist $\alpha > 0$ such that

$$\alpha|\xi|^p \le a(t,x,p,\xi)\cdot\xi$$

for all $\xi\in\mathbb{R}^d$ and some $1 < p < d$. Clearly, this fails if a is identically zero, even though our HJB equation otherwise satisfies the structure conditions in [22]. Following the discussion in Section 4, it may be possible to obtain a similar uniform bound by bounding the networks and their weights without losing any universal approximation guarantees, but we do not currently consider this approach.

Translating the argument in [5] to our context is also complicated by the fact that the class of quasilinear parabolic PDE for which they prove convergence of the DGM possesses a standard notion of weak solutions that, via the dominated convergence theorem, cooperates with convergence in L^2. In the case of HJB equations, however, viscosity solutions take the place of weak solutions and instead behave nicely with respect to uniform convergence. Thus, we require that $\varepsilon^n \to 0$ uniformly on $[0,T]\times\widehat{S}_d$ in Equation (31), but the original formulation of the DGM with L^2-error only implies convergence of the error term in L^2.

Finally, [5] only concludes uniform convergence of the neural network approximators to the true solution of the PDE after imposing additional assumptions of uniform boundedness and equicontinuity on the neural networks. Although this may be a reasonable assumption to include, we find that the modified DGM algorithm and the theory of viscosity solutions provide a more direct route to uniform convergence; see Section 5.3 for a more detailed discussion of the equicontinuity of neural network approximators and the potential issues with such an approach.

5.3 Equicontinuity of Neural Network Approximators

This section presents necessary conditions for the equicontinuity of neural network approximators, which would allow us to instead replicate the argument of [5] and obtain uniform convergence of the approximators to the true value function using the original DGM algorithm with L^2-error.

The primary condition that allows for equicontinuity is boundedness of the weights in the hidden layer(s) of the neural network used to approximate some continuous function. As shown in [16], a general universal approximation theorem for continuous, bounded functions such as Theorem 5.1 above still holds for neural networks with bounded weights. In particular, it is likely possible to approximate $V\in\mathcal{C}^{1,1}([0,T]\times S_d)$ (in $\mathcal{C}^1$) via a sequence of neural networks with bounded weights. In turn, we may apply the equicontinuity results established in [23], obtaining an *equicontinuous* sequence of neural network approximators to the value function V. Finally, the argument of [5] provides for uniform boundedness, allowing us to apply the Arzelà–Ascoli theorem to obtain uniform convergence. This modification is stated more precisely below.

Theorem 5.12. *Take $\mathfrak{C}_{d+1}(\sigma)$ as defined above and consider any function $f\in\mathcal{C}^m(K)$ for a compact set $K\subset\mathbb{R}^{d+1}$. Let $M>0$ be such that $\sup_{x\in K}|f(x)|\le M$. Now, let $\mathfrak{C}'_{d+1}(\sigma)$ denote the subset of networks in $\mathfrak{C}_{d+1}(\sigma)$ with weights $\theta=(\beta_1,\dots,\beta_n,\alpha_{1,1},\dots,\alpha_{d+1,n},c_1,\dots,c_n)\in\mathbb{R}^{2n+n(d+1)}$ satisfying $|\alpha_{j,i}|\le M$, $|c_i|\le M$, and $|\beta_i|\le M$ for all $i=1,\dots,n$ and $j=1,\dots,d+1$. Then, for any $m\in\mathbb{N}$, there exists a sequence of neural networks $\{\varphi\}_{k\in\mathbb{N}}$ with an increasing number of hidden units such that $\|f-\varphi^k\|_{\mathcal{C}^m(K)}<1/k$ for all $k\in\mathbb{N}$. Furthermore, the sequence $\{\varphi^k(t,m;\theta)\}_{k\in\mathbb{N}}$ is equicontinuous with respect to the inputs $(t,m)\in[0,T]\times S_d$.*

Proof. As shown in [16], any continuous, bounded functions can be uniformly approximated by neural networks with bounded weights. Assuming that we can in fact obtain a universal approximation result for a neural

networks in $\mathfrak{C}'_{d+1}$, [23, Proposition 8] shows that the weights of any network $\varphi \in \mathfrak{C}'_{d+1}$ with n hidden units such that $\|f - \varphi\|_{\mathcal{C}(K)} < 1$ must have weights satisfying

$$\left| \sum_{i=1}^{n} \beta_i \right| \leq M + 1.$$

In turn, the above condition allows us to apply [23, Theorem 20], which states that the subset of $\mathfrak{C}'_{d+1}(\sigma)$ satisfying the above summability condition is equicontinuous with respect to the input space. Specifically, this implies that the sequence $\{\varphi^k(t, m; \theta)\}_{k \in \mathbb{N}}$ is equicontinuous with respect to the inputs $(t, m) \in [0, T] \times S_d$ as claimed. □

In order to utilize Theorem 5.12 in our context, however, we require a suitable version of the universal approximation theorem for neural networks with bounded weights. For the time being, the modified DGM algorithm and Theorem 5.8 provide a workaround for this issue, which we hope to resolve in a future work.

6 Numerical Results

In this section, we present numerical results for DGM applied to a simple example case of the MFCP, as presented in [3, Example 2]. In particular, we consider the quadratic running cost

$$f(t, i, \alpha, m) = \frac{1}{2} \sum_{j \neq i} c_{i,j} \alpha_{i,j}^2 + f_0^i(m),$$

with $f_0^i(m) := m_i$ and $\{c_{i,j}\}_{i,j \in [\![d]\!]} \in \mathbb{R}^{d \times d}$ a cost matrix that encodes the cost of transitioning from state i to state j for $i \neq j$. Finally, we consider the linear terminal condition given by

$$g^i(m) = m_i$$

for $i \in [\![d]\!]$. With this choice of terminal cost, we obtain the terminal condition

$$V(T, m) = G(m) = \sum_{i \in [\![d]\!]} m_i^2.$$

In this simple example, the Hamiltonian is explicitly given

$$H^i(t, m, z) = \sum_{j \neq i} \left(-\mathfrak{a}^*(-z_j) z_j - \frac{1}{2} (\mathfrak{a}^*(-z_j))^2 \right) - f_0^i(m),$$

where

$$\mathfrak{a}^*(s) = \begin{cases} 0 & s \leq 0, \\ s & 0 \leq s \leq M, \\ M & s \geq M. \end{cases}$$

Recall from Assumption (B) that $M > 0$ is some constant such that $A = [0, M]^d$, where A is the action space for the MFCP. Under this construction, all of the convexity and Lipschitz continuity constraints in Assumptions (A) – (C) are satisfied.

For this example setup, we compare the performance of DGM with the original L^2-error from [5] and the modified algorithm with a uniform error metric presented above. Additionally, we explore the dependence of the performance of Algorithm 1 on the number of samples drawn from $[0, T] \times S_d \times S_d$. Finally, we demonstrate that the modified DGM algorithm scales well with dimension, providing numerical tests up to dimension $d = 500$.

The plots in Figure 1 and Figure 2 below demonstrate the value functions approximated by DGM with L^2 and uniform error respectively. Figures 3 and 4 contain the corresponding loss curves. Both the original and modified DGM algorithms perform similarly, but both are susceptible to local minima using the LSTM architecture outlined in [5]. In particular, after initially large decreases in training loss, both the original and modified

algorithms fall into local minima in their respective loss functions. However, given sufficient training time and appropriate hyperparameter tuning of the SGD optimizer and learning rate schedule used, both algorithms can closely approximate the true terminal condition of the example problem as demonstrated in Figure 5.

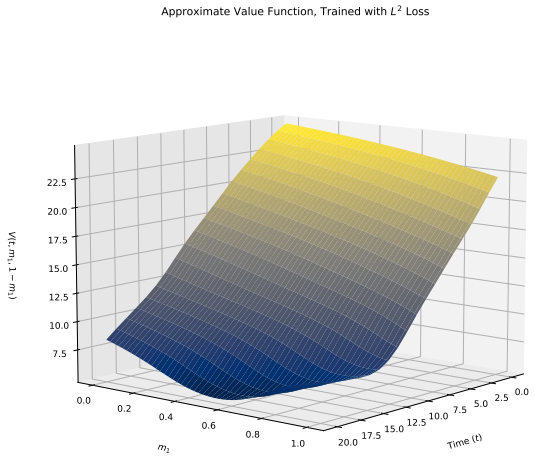

Figure 1: Approximate value function, L^2-error.

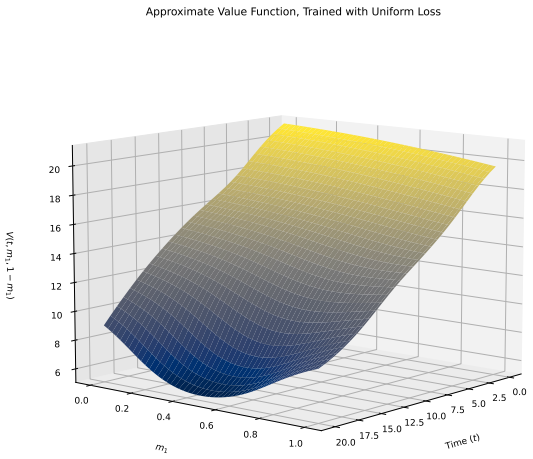

Figure 2: Approximate value function, uniform error.

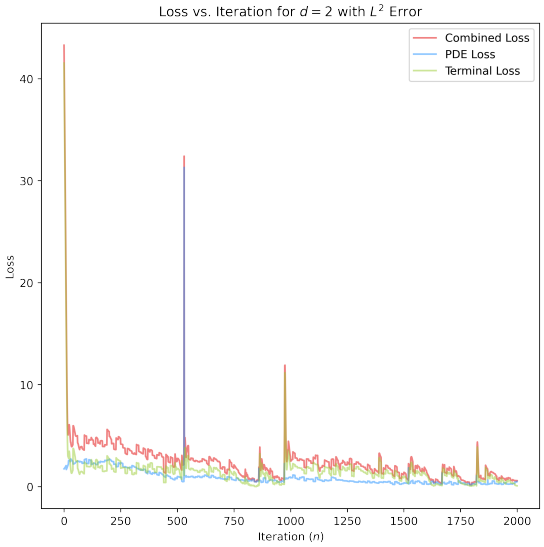

Figure 3: L^2 training loss, $d = 2$.

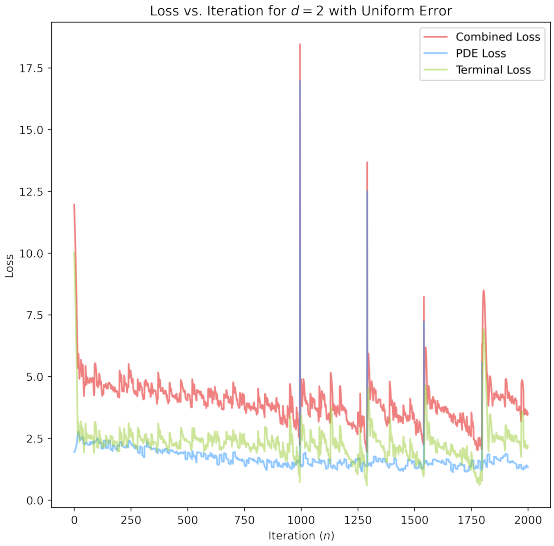

Figure 4: Uniform training loss, $d = 2$.

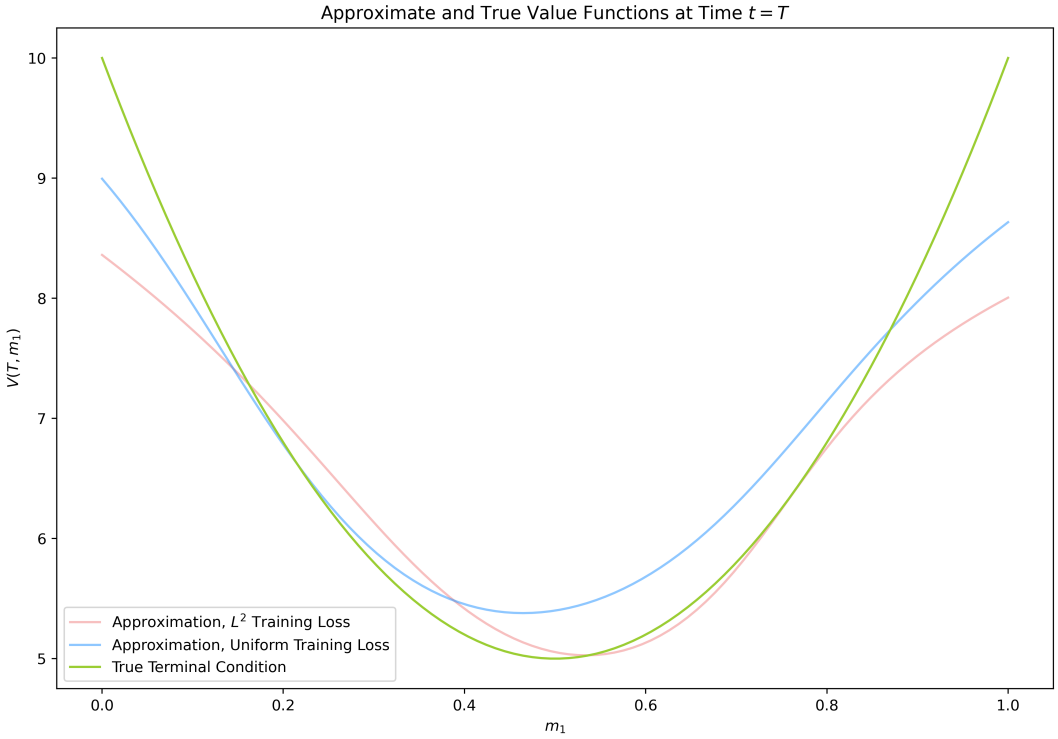

Figure 5: Approximate and true terminal conditions for both DGM algorithms.

In Figure 6 below, we see that the uniform DGM algorithm is far more stable as the number of samples M increases as expected. Unless otherwise specified, all numerical tests are carried out with $M = 100$, as the tradeoff between stability and runtime becomes worse as the number of samples exceeds $M = 100$.

Finally, in Table 1, we demonstrate the scalability of the DGM algorithm, implemented in TensorFlow and run with TPU acceleration. By exploiting TensorFlow's built-in vectorization and TPU acceleration functionality, the DGM algorithm scales well to dimension $d = 500$, with roughly linear increases in runtime past dimension $d = 100$. By runtime, we refer to the time that it takes to train the DGM network with uniform loss for 40 epochs, with 10 SGD steps in each epoch, and $M = 100$ samples at each step. All numerical experiments were run a 2019 MacBook Pro 1.4 GHz Quad-Core Intel Core i5 with 8 GB RAM, and all code can be found in the GitHub repository for this paper.

Dimension d	Total Training Time (s)	PDE Loss	Terminal Loss	Combined Loss
2	438.94	1.9422	2.3989	4.3411
5	432.13	0.3080	1.079	1.3874
10	427.05	0.1280	0.3490	0.4770
20	365.42	0.1882	0.1004	0.2887
50	429.75	0.0617	0.0323	0.0941
100	469.09	0.0220	0.0135	0.0355
200	610.24	0.0095	0.0039	0.0134
500	1621.6	0.0004	0.0011	0.0015

Table 1: Uniform DGM training times and losses as dimension d increases.

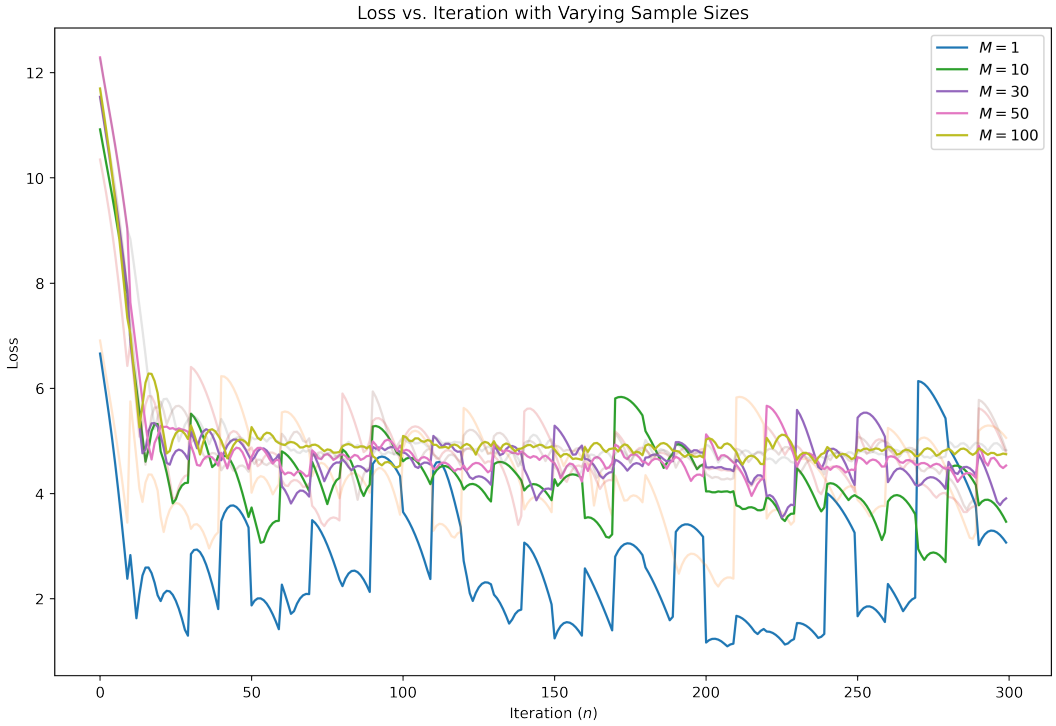

Figure 6: Dependence of loss on number of samples M for uniform DGM.

References

[1] René Carmona and François Delarue. Forward-backward stochastic differential equations and controlled McKean-Vlasov dynamics. *Ann. Probab.*, 43(5):2647–2700, 2015.

[2] Daniel Lacker. Limit theory for controlled McKean-Vlasov dynamics. *SIAM J. Control Optim.*, 55(3):1641–1672, 2017.

[3] Alekos Cecchin. Finite state N-agent and mean field control problems. *ESAIM: COCV*, 27:31, 2021.

[4] Vassili N. Kolokoltsov. Nonlinear markov games on a finite state space (mean-field and binary interactions), Apr 2012.

[5] Justin Sirignano and Konstantinos Spiliopoulos. DGM: A deep learning algorithm for solving partial differential equations. *Journal of Computational Physics*, 375:1339–1364, 2018.

[6] Weinan E, Jiequn Han, and Arnulf Jentzen. Deep learning-based numerical methods for high-dimensional parabolic partial differential equations and backward stochastic differential equations. *Commun. Math. Stat.*, 5(4):349–380, 2017.

[7] Huyên Pham. Feynman-Kac representation of fully nonlinear PDEs and applications. *Acta Math. Vietnam.*, 40(2):255–269, 2015.

[8] Côme Huré, Huyên Pham, and Xavier Warin. Deep backward schemes for high-dimensional nonlinear PDEs. *Math. Comp.*, 89(324):1547–1579, 2020.

[9] Huyên Pham. *Continuous-time stochastic control and optimization with financial applications*, volume 61 of *Stochastic Modelling and Applied Probability*. Springer-Verlag, Berlin, 2009.

[10] Michael G. Crandall, Hitoshi Ishii, and Pierre-Louis Lions. User's guide to viscosity solutions of second order partial differential equations. *Bull. Amer. Math. Soc. (N.S.)*, 27(1):1–67, 1992.

[11] Franco Scarselli and Ah Chung Tsoi. Universal approximation using feedforward neural networks: A survey of some existing methods, and some new results. *Neural Networks*, 11(1):15–37, 1998.

[12] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989.

[13] G. Cybenko. Approximation by superpositions of a sigmoidal function. *Math. Control Signals Systems*, 2(4):303–314, 1989.

[14] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks. *Neural Networks*, 3(5):551–560, 1990.

[15] Kurt Hornik. Approximation capabilities of multilayer feedforward networks. *Neural Networks*, 4(2):251–257, 1991.

[16] M. Stinchcombe and H. White. Approximating and learning unknown mappings using multilayer feedforward networks with bounded weights. In *1990 IJCNN International Joint Conference on Neural Networks*, pages 7–16 vol.3, 1990.

[17] H. N. Mhaskar and Charles A. Micchelli. Degree of approximation by neural and translation networks with a single hidden layer. *Adv. in Appl. Math.*, 16(2):151–183, 1995.

[18] Tim De Ryck, Samuel Lanthaler, and Siddhartha Mishra. On the approximation of functions by tanh neural networks. *Neural Networks*, 143:732–750, 2021.

[19] Denis Belomestny, Alexey Naumov, Nikita Puchkin, and Sergey Samsonov. Simultaneous approximation of a smooth function and its derivatives by deep neural networks with piecewise-polynomial activations. *Neural Networks*, 161:242–253, 2023.

[20] M. G. Crandall, L. C. Evans, and P. L. Lions. Some properties of viscosity solutions of Hamilton-Jacobi equations. *Transactions of the American Mathematical Society*, 282(2):487–502, 1984.

[21] Maria Michaela Porzio. Existence of solutions for some "noncoercive" parabolic equations. *Discrete and Continuous Dynamical Systems*, 5(3):553 – 568, 1999.

[22] Martina Magliocca. Existence results for a Cauchy–Dirichlet parabolic problem with a repulsive gradient term. *Nonlinear Analysis*, 166:102–143, 2018.

[23] P. Chandra and Y. Singh. Feedforward sigmoidal networks - equicontinuity and fault-tolerance properties. *IEEE Transactions on Neural Networks*, 15(6):1350–1366, 2004.

[24] Olav Kallenberg. *Foundations of modern probability*. Probability and its Applications (New York). Springer-Verlag, New York, second edition, 2002.

[25] W.H. Fleming and H.M. Soner. *Controlled Markov Processes and Viscosity Solutions*. Stochastic Modelling and Applied Probability. Springer New York, 2006.

A Derivations of HJB Equations

In this appendix, we present the derivations of the HJB equations for both the N-agent optimization problem and the MFCP, first introduced in Section 2. We begin with the HJB equation in Proposition 2.1, corresponding to the standard N-agent optimization problem.

Proof of Proposition 2.1. Recall the definition of the generator of a continuous-time Markov chain; if $\varphi : \{1,\dots,d\}^N \to \mathbb{R}$ and $T_{t,h}\varphi(\mathbf{x}) := \mathbb{E}[\varphi(\mathbf{X}_{t+h}) \mid \mathbf{X}_t = \mathbf{x}]$ denotes the Feller semigroup of the process, then we define the infinitesimal generator of the process by

$$\mathcal{L}_t^{N,\boldsymbol{\beta}}\varphi(\mathbf{x}) := \lim_{h\downarrow 0} \frac{T_{t,h}\varphi(\mathbf{x}) - \varphi(\mathbf{x})}{h}. \tag{32}$$

We can now explicitly compute the limit in (32) given the dynamics in (1). In particular, we observe that

$$T_{t+h}\varphi(\mathbf{x}) = \mathbb{E}[\varphi(\mathbf{X}_h) \mid \mathbf{X}_t = \mathbf{x}] = \sum_{k=1}^N \sum_{j\neq x_k} \left[Q_{x_k,j}(h,\beta_k(h,\mathbf{x}),\mu_{\mathbf{x}}^N)h + o(h)\right]\varphi([\mathbf{x}^{-k},j]).$$

Assuming that $\varphi : \{1,\dots,d\}^N \to \mathbb{R}$ is bounded, the fact that $o(h)/h \to 0$ as $h \downarrow 0$ now implies that

$$\begin{aligned}
\mathcal{L}_t^{N,\boldsymbol{\beta}}\varphi(\mathbf{x}) = \lim_{h\downarrow 0} \frac{T_{t,h}\varphi(\mathbf{x}) - \varphi(\mathbf{x})}{h} &= \sum_{k=1}^N \sum_{j\neq x_k} Q_{x_k,j}(t,\beta_k(t,\mathbf{x}),\mu_{\mathbf{x}}^N)\left[\varphi([\mathbf{x}^{-k},j]) - \varphi(\mathbf{x})\right] \\
&= \sum_{k=1}^N \langle Q_{x_k,\bullet}(t,\beta_k(t,\mathbf{x},\mu_{\mathbf{x}}^N)), \Delta^k\varphi(\mathbf{x})\rangle \\
&= \frac{1}{N}\sum_{k=1}^N \langle Q_{x_k,\bullet}(t,\beta_k(t,\mathbf{x},\mu_{\mathbf{x}}^N)), N\Delta^k\varphi(\mathbf{x})\rangle.
\end{aligned}$$

From Assumption (A), we have that the transition rate $Q_{i,j}$ is continuous on $[0,T]\times A \times S_d$, allowing us to pass the limit inside the transition rate above.

The next step is the key step that allows us to apply Itô's formula, or rather, Dynkin's formula. In particular, recall that if $(X_t)_{t\in[0,T]}$ is a Feller process (as is the case here) and $\mathcal{L}_0^{N,\boldsymbol{\beta}}\varphi$ is well-defined, then the process given by

$$M_t^{\varphi} := \varphi(X_t) - \varphi(X_0) - \int_0^t \mathcal{L}_0^{N,\boldsymbol{\beta}}\varphi(X_s)ds.$$

is a martingale adapted to the canonical filtration of the process $(X_t)_{t\in[0,T]}$; see [24, Chapter 17], for instance. Now, if v^N denotes the value function of the N-agent stochastic control problem, we can now apply Dynkin's formula (noting that $(M_t^{v^N})_{t\in[0,T]}$ is now a martingale even if $(X_t)_{t\in[0,T]}$ may not be) to see that

$$v^N(t+h,\mathbf{X}_{t+h}) = v^N(t,\mathbf{X}_t) + \int_t^{t+h}\left(\frac{\partial v^N}{\partial t} + \mathcal{L}_t^{N,\boldsymbol{\beta}}v^N\right)(s,\mathbf{X}_s)ds.$$

Taking conditional expectations, we find that

$$\mathbb{E}[v^N(t+h,\mathbf{X}_{t+h}) \mid \mathbf{X}_t = \mathbf{x}] = v^N(t,\mathbf{x}) + \mathbb{E}\left[\int_t^{t+h}\left(\frac{\partial v^N}{\partial t} + \mathcal{L}_t^{N,\boldsymbol{\beta}}v^N\right)(s,\mathbf{X}_s)ds \,\middle|\, \mathbf{X}_t = \mathbf{x}\right]. \tag{33}$$

Here, we remark that the above expression from Itô's formula may also contain a local martingale that disappears upon taking expectations.

Now, define for any control $\boldsymbol{\beta}'$, we observe that for all $0 \le t < r \le T$,

$$v^N(t,\mathbf{x}) = \inf_{\boldsymbol{\beta}\in A^N} J^N(t,\boldsymbol{\beta},\mathbf{x}) \le J^N(t,\boldsymbol{\beta}',\mathbf{x})$$

$$
\begin{aligned}
&= \frac{1}{N}\sum_{k=1}^{N}\mathbb{E}\left[\int_t^T f(s, X_s^k, \beta_k'(s, X_s^k), \mu_s^N)ds + g(X_k^T, \mu_T^N) \;\middle|\; \mathbf{X}_t = \mathbf{x}\right] \\
&= \frac{1}{N}\sum_{k=1}^{N}\mathbb{E}\left[\int_r^T f(s, X_s^k, \beta_k'(s, X_s^k), \mu_s^N)ds + g(X_k^T, \mu_T^N) \;\middle|\; \mathbf{X}_t = \mathbf{x}\right] \\
&\quad + \frac{1}{N}\sum_{k=1}^{N}\mathbb{E}\left[\int_t^r f(s, X_s^k, \beta_k'(s, X_s^k), \mu_s^N)ds \;\middle|\; \mathbf{X}_t = \mathbf{x}\right] \\
&= \mathbb{E}[J^N(r, \boldsymbol{\beta}', \mathbf{X}_r) \mid \mathbf{X}_t = \mathbf{x}] + \frac{1}{N}\sum_{k=1}^{N}\mathbb{E}\left[\int_t^r f(s, X_s^k, \beta_k'(s, X_s^k), \mu_s^N)ds \;\middle|\; \mathbf{X}_t = \mathbf{x}\right]
\end{aligned}
$$

by the Markov property and the tower property. Now, assume that the control $\boldsymbol{\beta}'$ is after time r so that $J^N(r, \boldsymbol{\beta}', \mathbf{X}_r) = v^N(r, \mathbf{X}_r)$. We then obtain, with $r = t + h$,

$$
v^N(t, \mathbf{x}) \geq \mathbb{E}[v^N(t+h, \mathbf{X}_{t+h}) \mid \mathbf{X}_t = \mathbf{x}] + \frac{1}{N}\sum_{k=1}^{N}\mathbb{E}\left[\int_t^{t+h} f(s, X_s^k, \beta_k'(s, X_s^k), \mu_s^N)ds \;\middle|\; \mathbf{X}_t = \mathbf{x}\right]. \tag{34}
$$

Applying the law of iterated expectation and plugging this inequality into the inequality in (33), we see that

$$
\mathbb{E}\left[\int_t^{t+h} \frac{1}{N}\sum_{k=1}^{N} f(s, X_s^k, \beta_k'(s, X_s^k), \mu_s^N) + \left(\frac{\partial v^N}{\partial t} + \mathcal{L}_t^{N,\boldsymbol{\beta}} v^N\right)(s, \mathbf{X}_s)ds\right] \geq 0.
$$

Dividing by h and taking the limit as $h \downarrow 0$ as in [9], we obtain via the mean value theorem that

$$
\begin{aligned}
0 &\geq -\frac{\partial v^N}{\partial t}(t, \mathbf{x}) + \left(-\mathcal{L}_t^{N,\boldsymbol{\beta}} v^N(t, \mathbf{x}) - \frac{1}{N}\sum_{k=1}^{N} f(t, x_k, \beta_k(t, \mathbf{x}), \mu_t^N)\right) \\
&= -\frac{\partial v^N}{\partial t}(t, \mathbf{x}) + \frac{1}{N}\left(-\sum_{k=1}^{N}\langle Q_{x_k,\bullet}(t, \beta_k(t, \mathbf{x}, \mu_{\mathbf{x}}^N)), N\Delta^k v^N(t, \mathbf{x})\rangle - \sum_{k=1}^{N} f(t, x_k, \beta_k(t, \mathbf{x}), \mu_t^N)\right),
\end{aligned}
$$

with equality when the optimal control $\boldsymbol{\beta}^\star$ is chosen. As a result, we have that

$$
0 = -\frac{\partial v^N}{\partial t}(t, \mathbf{x}) + \sup_{\boldsymbol{\beta} \in A^N}\left(-\sum_{k=1}^{N}\langle Q_{x_k,\bullet}(t, \beta_k(t, \mathbf{x}, \mu_{\mathbf{x}}^N)), N\Delta^k v^N(t, \mathbf{x})\rangle - \sum_{k=1}^{N} f(t, x_k, \beta_k(t, \mathbf{x}), \mu_t^N)\right).
$$

Thus, by the preceding definition of the Hamiltonian, we therefore conclude that the value function v^N satisfies

$$
-\frac{\partial v^N}{\partial t}(t, \mathbf{x}) + \frac{1}{N}\sum_{k=1}^{N} H^{x_k}(t, \mu_{\mathbf{x}}^N, N\Delta^k v^N(t, \mathbf{x})) = 0.
$$

Additionally, because the terminal cost is given by $\frac{1}{N}\sum_{k=1}^{N} g(\mathbf{X}_k^T, \mu_T^N)$, the associated terminal condition for the above system of ODEs must be

$$
v^N(T, x) = \frac{1}{N}\sum_{k=1}^{N} g(x_k, \mu_{\mathbf{x}}^N).
$$

The above derivation yields the HJB equation for the N-agent optimization problem. Uniqueness and the stated regularity then follow from [3, Proposition 2.3]. □

Next, we derive the HJB equation for reformulation of the N-agent problem, presented in Proposition 2.2.

Proof of Proposition 2.2. The generator of this Markov chain is instead given by

$$
\mathcal{L}_t^{N,\alpha_N} v(m) = N\sum_{i,j \in [\![d]\!]} m_i Q_{i,j}(t, \alpha_N(t, i, m), m)\left[v(m + \frac{1}{N}(\delta_j - \delta_i) - v(m)\right], \tag{35}
$$

but the HJB equation for the new value function $V^N(t,m)$ can be derived precisely as above. Consequently, the same reasoning as in the proof of Proposition 2.1 (albeit with a different generator) yields the HJB equation

$$\begin{aligned} 0 &= -\frac{\partial V^N}{\partial t}(t,m) + \sup_{\alpha_N \in A^d} \left(-\mathcal{L}_t^{N,\alpha_N} V^N(t,m) - \sum_{i \in [\![d]\!]} m_i f(t,i,a^i,m) \right) \\ &= -\frac{\partial V^N}{\partial t}(t,m) + \sum_{i \in [\![d]\!]} m_i H^i(t,m,D^{N,i}V^N(t,m)) \end{aligned}$$

in terms of the Hamiltonian defined in [3]. Finally, we have the following terminal condition, coming from the terminal cost term in (15):

$$V^N(T,m) = \sum_{i \in [\![d]\!]} m_i g^i(m).$$

Again, uniqueness and the stated regularity are shown in [3, Proposition 2.6]. □

Finally, it remains to derive the HJB equation for the MFCP itself, as stated in Proposition 2.3.

Proof of Proposition 2.3. In Section 2, we reduced the MFCP to a deterministic control problem. Thus, we may apply the dynamic programming principle (DPP) as usual. If the value function of the MFCP problem is given by

$$V(t,\mu) = \inf_{\alpha} J(t,\alpha,\mu),$$

then the DPP (as stated and derived for the deterministic control problem in [25]) states that

$$V(t,\mu_t) = \inf_{\alpha} \left(\int_t^{t+h} \sum_{i \in [\![d]\!]} f(s,i,\alpha^i(s),\mu_s)\mu_s^i ds + V(t+h,\mu_{t+h}) \right).$$

Above, the control α is such that $\alpha : [0,T] \to A^d$. Subtracting $V(t,\mu_t)$ from both sides and dividing by h, we obtain

$$0 = \inf_{\alpha} \left(\frac{1}{h} \int_t^{t+h} \sum_{i \in [\![d]\!]} f(s,i,\alpha^i(s),\mu_s)\mu_s^i ds + \frac{V(t+h,\mu_{t+h}) - V(t,\mu_t)}{h} \right)$$

Taking the limit as $h \downarrow 0$ then yields

$$\begin{aligned} 0 &= \inf_{a \in A^d} \left(\sum_{i \in [\![d]\!]} f(t,i,a,\mu_t)\mu_t^i + \frac{\partial V}{\partial t}(t,\mu_t) + \nabla_m V(t,\mu_t) \cdot \frac{d}{dt}\mu_t(t,a) \right) \\ &= \frac{\partial V}{\partial t}(t,\mu_t) + \inf_{a \in A^d} \left(\sum_{i \in [\![d]\!]} \mu_t^i f(t,i,a,\mu_t) + \sum_{i,j \in [\![d]\!]} \mu_t^i Q_{ij}(t,a^i,\mu_t) \partial_{m_j - m_i} V(t,\mu_t) \right), \end{aligned}$$

where in the last step, we utilize the dynamics of the system from above and recall that on the interior of the simplex, only derivatives in the directions $(e_j - e_i)_{i,j \in [\![d]\!]}$ are considered in [3]. This last step is justified in [3, Section 2], where the simplex is represented as a $(d-1)$-dimensional submanifold of $\mathbb{R}^d$ via the obvious local chart. Rearranging and notationally replacing μ_t with m, we obtain the HJB equation

$$-\frac{\partial V}{\partial t}(t,m) + \sup_{a \in A^d} \left(-\sum_{i \in [\![d]\!]} m_i f(t,i,a,m) - \sum_{i,j \in [\![d]\!]} m_i Q_{ij}(t,a^i,m) \partial_{m_j - m_i} V(t,m) \right) = 0.$$

Finally, using more notation from [3], we write $D_j^i V(t,m) = \partial_{m_j - m_i} V(t,m)$ so that the we can write the second term above in terms of the relevant pre-Hamiltonian:

$$-\sum_{i \in [\![d]\!]} m_i f(t,i,a,m) - \sum_{i,j \in [\![d]\!]} m_i Q_{ij}(t,a^i,m) \partial_{m_j - m_i} V(t,m) = \sum_{i \in [\![d]\!]} m_i \left(-\langle Q_{i,\bullet}(t,a,m), D^i V(t,m) \rangle - f(t,i,a,m) \right).$$

In turn, we arrive at the final HJB equation for the MFCP, given by

$$-\frac{\partial V}{\partial t}(t,m) + \sum_{i \in [\![d]\!]} m_i H^i(t, m, D^i V(t,m)) = 0,$$

from the definition of the Hamiltonian associated with the above pre-Hamiltonian. As before, the associated boundary condition comes directly from the terminal cost of the problem, and is given by

$$V(T,m) = \sum_{i \in [\![d]\!]} m_i g^i(m).$$

This concludes the derivation of the HJB equation for the MFCP. We now obtain uniqueness and the regularity of V from Theorem 2.4. □