University of California
Santa Barbara

Topics in Signature, Directed Chain SDEs and Applications in Machine Learning

A dissertation submitted in partial satisfaction
of the requirements for the degree

Doctor of Philosophy
in
Statistics and Applied Probability

by

Ming Min

Committee in charge:

Professor Tomoyuki Ichiba, Chair
Professor Jean-Pierre Fouque
Professor Ruimeng Hu

June 2023

The Dissertation of Ming Min is approved.

Professor Jean-Pierre Fouque

Professor Ruimeng Hu

Professor Tomoyuki Ichiba, Committee Chair

April 2023

Topics in Signature, Directed Chain SDEs and Applications in Machine Learning

I would like to dedicate this thesis to my family.

Acknowledgements

When I look back on the last five years of my PhD journey, I feel encouraged and supported by my advisor, friends and family.

I am indebted to my advisor, Professor Tomoyuki Ichiba, for his unwavering support, expert guidance and constant encouragement. Not only did he teach me so many knowledge in Theoretical Probability with patience, but also encourage me to come up with my own immature research ideas with countless tolerances. Professor Ichiba set me an example of being a humble, dedicated and successful researcher. I am so lucky to be advised by Professor Ichiba.

I would like to thank Professor Jean-Pierre Fouque for being my committee member and giving me insightful research advice. I also want to extend my heartfelt thanks to my committee member Professor Ruimeng Hu. Ruimeng is not only a teacher, but also a friend who gives me so many invaluable suggestions on our research projects and my career growth.

My sincere appreciation goes to Shan Jiang, Hanzhao Wang, Zhenyu Qiu, Yue He, Weiyi Li, Menglin Li, Jiawei Wang, Yi Zheng, Zhuoli Jin and Mengye Liu for their unconditional friendship. It is from their companionship that my PhD life could be so enjoyable.

Most of all, I am grateful to my parents, Yugen Min and Renhe Zhang. They may not understand very much of Mathematics and Probability, but their unselfish support is the most important thing in my life. Finally, I am so fortunate to meet my fiancée Yichen Feng, your love fills every day of my life with happiness and laughter.

Abstract

Topics in Signature, Directed Chain SDEs and Applications in Machine Learning

by

Ming Min

Stochastic analysis, stochastic processes and machine learning of dynamical systems have depicted strong connection in many aspects. This thesis aims to study such relationship from two directions. The first direction is using signature and deep learning techniques to propose efficient algorithm learning Mean Field Games with common noises. The second direction deploy the distributional invariance property of directed chain stochastic differential equations to design a novel time series generator with excellent simulation ability.

In the first part, we introduce signature, borrowed from Rough Paths theory, as an efficient feature extraction technique and propose a novel algorithm to address the *curse of dimensionality* issue.

In the second part, we propose an application of signature. In the problem of learning Mean Field Games with common noises, traditional algorithms admit a nested loop structure due to the appearance of individual and common noises. Our proposed algorithm (Sig-DFP), utilize the universality property of signatures, has only single loop, which improves the efficiency from quadratic to linear in both time and space complexity.

In the third part, we first study smoothing property of directed chain stochastic differential equations via partial Malliavin calculus, and then propose a novel generative adversarial network based time series generator. We also point out the independence issue of this directed chain generator, and solve it via branching scheme.

Contents

Chapter 1

Introduction

1.1 Motivation

The emergence of machine learning has increasingly great impact to scientific research. With strong power, machine learning techniques (Deep Learning, Reinforcement Learning, Gaussian Process, Kernel Methods etc.) have been applied to solve scientific computing problems successfully that were difficult to solve in the past. As an essential field of machine learning, dynamical system analysis is naturally closely connected to stochastic processes and stochastic analysis, which are important components of theoretical probability that plays an important role as the theoretical support of machine learning algorithms.

On one hand, stochastic analysis provide numerous applications to propose and exam novel machine learning methods for dynamical systems. Recurrent neural networks, neural differential equations and reinforcement learning have been applied to solve Stochastic Control and Mean Field Game (MFG) problems; Gaussian Process have been implemented in pricing American option that is also known as free boundary problem; Solving high dimensional parabolic partial differential equations, which is equivalent to a corresponding high dimensional stochastic control problem, has draw great attentions in recent years and are solved by deep backward stochastic differential equations(BSDE) algorithm; In [63], deep neural network is used to approximate the Radon-Nikodym derivative in the problem of systemic risk measures to find the optimal Q-measure.

On the other hand, stochastic analysis and theory provides many well studied models that turns out to be excellent machine learning algorithms. Rough paths theory introduces neural rough differential equations(neural RDE) as an extension of neural ordinary differential equations(neural ODE) to simulating the dynamical system; signature, a crucial object from Rough paths theory, can be treated as an efficient feature extraction tool for time series; stochastic differential equations inspired researchers to develop neural SDE, again an extension of neural RDE and neural ODE, to better simulate dynamical systems with high volatility such as stock prices. Indeed, the mentioned deep BSDE solver is supported by the BSDE theory designed as pure probabilistic object.

The machine learning algorithms of dynamical system and stochastic analysis depicts strong bond as shown in the above examples. In this thesis, we follow this idea as our backbone. Firstly, we introduce Rough paths theory in its lightest version and signature. We then point out that signature suffers *the curse of dimensionality* and propose our novel convolutional neural network based algorithm to address this problem.

Secondly, we investigate another application of signature: In learning Mean Field Games with common noises, all existing algorithms suffer from the quadratic complexity issue because of the appearance of both individual and common noises. With the favor of signature, we propose a linear complexity algorithms for learning MFG with common noises.

Thirdly, we theoretically study smoothing property of directed chain stochastic differential equations(directed chain SDE) via partial Malliavin calculus. Driven by the distribution invariance property of directed chain SDE, we design a novel time series generator named after directed chain GAN and justify its superiority by both synthetic and real world datasets.

1.2 Originality, Contribution Summary and Open Questions

The content of this thesis is either my original work with collaborators, or relevant prior or concurrent work included for reference.

1. The partial content of Chapter 2 is the result of a collaboration with Dr. Tomoyuki Ichiba, and has previously appeared in [121] entitled as “Convolutional Signature for Sequential Data”.

2. The content of Chapter 3 comes from the collaboration with Dr. Ruimeng Hu, and has previously appeared in [119] entitled as “Signatured Deep Fictitious Play for Mean Field Games with Common Noises”.

3. The content of Chapter 4 is the result of a collaboration with Dr. Tomoyuki Ichiba, Ruimeng Hu and has previously appeared in [83] entitled as “Smoothness of Directed Chain Stochastic Differential Equations” and [120] entitled as “Directed Chain Generative Adversarial Networks”.

In ”Convolutional Signature for Sequential Data“, we proposed a algorithm (CNN-Sig) to address the well known *curse of dimensionaly* issue in using signature as feature map for high-dimension time series data. Our technique considers using convolutional layer and is purely data-driven. We examined the feasibility of our algorithms on several high-dimension datasets, including NLP dataset of IMDB reviews.

In “Signatured Deep Fictitious Play for Mean Field Games with Common Noises”, we took advantage of the universality property of signatures and fictitous play scheme to designed a novel algorithm (Sig-DFP) for solving Mean Field Games with common noises. Our algorithm dramatically improves the complexity by one order, quadratically to linearly, and is tested on multiple cases, including heterogeneous extended form Mean Field Games.

In “Smoothness of Directed Chain Stochastic Differential Equations”, we proved the existence and smoothness of the density of directed chain SDEs via partial Malliavin Calculus. Based on these results, we derived partial differential equations associated with directed chain SDEs.

In “Directed Chain Generative Adversarial Networks”, we proposed a time series generator based on the idea of directed chain SDEs and showed its advantage through several experiments. Using the previous theoretical work as a toolbox, we proved the expressiveness and dependence decay property of the propsed generator.

Building upon our works, we list two open questions as our future research directions.

1. From the results of smoothness of directed chain SDEs, can we generalize the results to the solutions of stochastic differential game problem built on directed chain system proposed in [59]? If the answer is positive, we may use partial differential equation approach (eg. master equations) to solve the game problem, instead of probabilisitc approach used in [59]. There still exists some gaps between our work and the game problem in [59]. The infinite player game problem admits solution of the form that each player has dependence on all the players to infinite along the direction of the chain. However, our approach is suitable for the case that the each player has dependence only on a bounded number of his/her neighborhoods. The difficulty here is how to extend our approach from finite to infinite dependences.

2. Data privacy has become a common concern along with the development of artificial intelligence. One open problem inspired by directed chain SDEs is that can we use it as a time series generator to address data privacy problems? For instance, what is the performance of using directed chain SDEs to generate fake healthy data? Can we protect users data if the directed chain SDEs generator is embedded into the machine learning application pipelines? The answers to these questions could drive widespread applications of directed chain SDEs.

Chapter 2

Signature

2.1 Signature and Geometric Rough Paths

Let us introduce some backgrounds in order to explain the signature method, following [115]. Given a Banach space E with a norm $\|\cdot\|$, we define the tensor algebra

$$T((E)) := \{(a_i)_{i\geq 0} : a_i \in E^{\otimes i} \text{ for every } i\} \tag{2.1}$$

associated with the sum $+$ and with the tensor product $\otimes$ defined by

$$(a_i)_{i\geq 0} + (b_i)_{i\geq 0} := (a_i + b_i)_{i\geq 0}, \quad (a_i)_{i\geq 0} \otimes (b_i)_{i\geq 0} := (c_i)_{i\geq 0},$$

where the jth element $c_j := \sum_{k=0}^{j} a_k \otimes b_{j-k}$ is the convolution of the first j elements of $(a_i)_{i\geq 0}$ and $(b_i)_{i\geq 0}$ in $T((E))$. Similarly, let us define its subset

$$T(E) := \{(a_i)_{i\geq 0} : a_i \in E^{\otimes i} \text{ and } \exists N \in \mathbb{N} \text{ such that } a_i = 0 \ \forall i \geq N\} \tag{2.2}$$

of $T((E))$ for those with finite number of non-zero elements. Note that $T(E) \subset T((E))$. Also, we shall consider the truncated tensor algebra of order $m \in \mathbb{N}$, i.e.,

$$T^m(E) := \{(a_i)_{i=0}^m : a_i \in E^{\otimes i} \text{ for } \forall i \leq m\}, \tag{2.3}$$

which is a subalgebra of $T((E))$. Then as we shall see, the signatures and the m-th order truncated signatures lie in these spaces $T((E))$ and $T^m(E)$, respectively.

Now with $E := \mathbb{R}^d$ and the usual Euclidean norm $\|\cdot\|$, we shall define the space $\mathcal{V}^p([0,T],E)$ of the d-dimensional continuous paths of finite p-th variation over the time interval $[0,T]$ and the signatures of the paths in $\mathcal{V}^p([0,T],E)$.

Definition 2.1.1 (The space of finite p-variation paths) *Fix $p \geq 1$ and the interval $[0,T]$. The p-variation of a d-dimensional path $X : [0,T] \to E := \mathbb{R}^d$ is defined by*

$$\|X\|_p := \left(\sup_{D_n \subset [0,T]} \sum_{i=0}^{n-1} \|X_{t_{i+1}} - X_{t_i}\|^p \right)^{1/p}.$$

Here, the supremum is taken over all the possible partitions of the form $D_n := \{t_i\}_{1 \leq i \leq n}$ of $[0,T]$ with $0 = t_0 < t_1 < \cdots < t_n \leq T$, $n \geq 1$. X is said to be of finite p-variation, if $\|X\|_p < \infty$. We denote the set of continuous paths $X : [0,T] \to E$ of finite p-variation by $\mathcal{V}^p([0,T],E)$.

We use the supremum norm $\|\cdot\|_\infty$ for continuous functions on $[0,T]$, i.e., $\|f\|_\infty := \sup_{x \in [0,T]} |f(x)|$. It can be shown that if we equip the space $\mathcal{V}^p([0,T],E)$ with the norm $\|X\|_{\mathcal{V}^p([0,T],E)} := \|X\|_p + \|X\|_\infty$, then $\mathcal{V}^p([0,T],E)$ is a Banach space. Now the signature and truncated signature are defined as follows.

Definition 2.1.2 (Signatures) *The signature $S(X)$ of a path $X \in \mathcal{V}^p([0,T],E)$, $p \geq 1$*

is defined by $S(X) := (1, X^1, X^2, ...) \in T((E))$, *where the* k*-th element*

$$X^k := \int \cdots \int_{0<t_1<\cdots<t_n<T} \mathrm{d}X_{t_1} \otimes \cdots \otimes \mathrm{d}X_{t_n} \in E^{\otimes k} \tag{2.4}$$

is the k*-fold, iterated integral for* $k \geq 1$, *if the iterated integrals are well defined.*

The truncated signature is naturally defined as $S^m(X) := (1, X^1, X^2, ..., X^m) \in T^m(E)$ *for every* $m \geq 1$ *including the* 0*-th term* $S^0(X) = 1$.

Remark 2.1.3 *The integrals in* (2.4) *depend on the nature of the paths. Here are some typical examples:*

1. *If* X *is of* 1*-variation path, then the integrals* (2.4) *of the signature can be understood as the Stieltjes integral;*

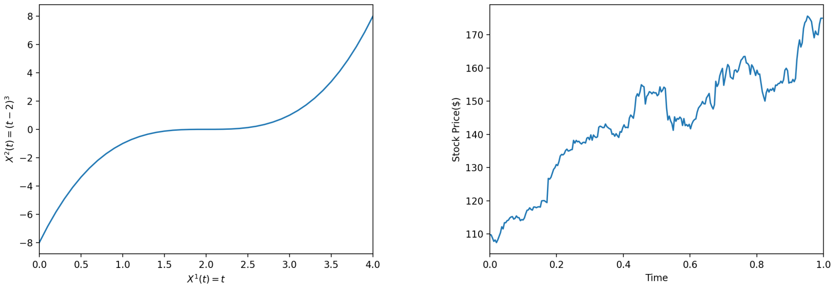
2. *If* X *is of* p*-variation path with* $1 < p < 2$, *then it can be defined in the sense of Young (e.g., see [114]).*

3. *If* X *is a Brownian motion, then we can use the Itô integral or the Stratonovtich integral. As we will explain later, when extending from a Brownian motion path or a semimartingale to a geometric rough path, we choose the Stratonovitch integral rather than the Itô integral.*

Example 2.1.4 (Smooth paths and piece-wise linear paths) *For* $p \geq 1$ *the path space* $\mathcal{V}^p([0,T], E)$ *contains the smooth functions and the piece-wise linear functions. We give the following two examples of paths in* $\mathcal{V}^p([0,T], E)$, *as shown in Figure 2.1. In its left panel, we plot the smooth path* $X_t = (t, (t-2)^3), t \in [0, 4]$. *In its right panel, we represent the discrete data: daily AAPL adjusted close stock price from Nov 28, 2016 to Nov 24, 2017 by interpolating the path linearly between each successive two days. The first* 2 *degree signatures* X^1 *and* X^2 *of these two paths in* (2.4) *are calculated and given in Table 2.1.*

X	$(t,(t-2)^3)$	AAPL
X^1	$(4,16)$	$(1,65.52)$
X^2	$\begin{pmatrix} 8 & 32 \\ 32 & 128 \end{pmatrix}$	$\begin{pmatrix} 0.5 & 31.17 \\ 34.00 & 2123.3 \end{pmatrix}$

Table 2.1: The corresponding signatures for the smooth path $(t,(t-2)^3)$ and the piece-wise linear path of the augmented AAPL adjusted price in Figure 2.1, respectively.



(a) Smooth path

(b) Piecewise linear path

Figure 2.1: Examples of $\mathcal{V}^p([0,T],E)$, $p \geq 1$: (a) Plot of a smooth path $X_t = (t,(t-2)^3), t \in [0,4]$. (b) Plot of linear interpolation of daily AAPL adjusted close stock price from Nov 28, 2016 to Nov 24, 2017.

Geometric Rough Paths and Linear Functionals

Here we introduce rough paths and geometric rough paths briefly, following [114, 115, 64]. Denote by Δ_T the simplex $\{(s,t) \in [0,T]^2 : 0 \leq s \leq t \leq T\}$, set $E = \mathbb{R}^d$ and hence $T^n(\mathbb{R}^d) = \bigoplus_{k=0}^n (\mathbb{R}^d)^{\otimes k}$ the truncated tensor algebra.

Definition 2.1.5 (Multiplicative Functional) *Let $\mathbb{X} : \Delta_T \to T^n(\mathbb{R}^d)$, with $n \geq 1$ as an integer. For each $(s,t) \in \Delta_T$, $\mathbb{X}_{s,t}$ denotes the image of (s,t) under the mapping $\mathbb{X}$, and we write*

$$\mathbb{X}_{s,t} = (\mathbb{X}^0_{s,t}, \mathbb{X}^1_{s,t}, \ldots, \mathbb{X}^n_{s,t}) \in T^n(\mathbb{R}^d).$$

The function $\mathbb{X}$ is called a multiplicative functional of degree n in $\mathbb{R}^d$ if $\mathbb{X}^0_{s,t} = 1$ for all

$(s,t) \in \Delta_T$ and

$$\mathbb{X}_{s,u} \otimes \mathbb{X}_{u,t} = \mathbb{X}_{s,t},\ \forall s,u,t \in [0,T],\ s \leq u \leq t, \tag{2.5}$$

which is called Chen's identity.

Rough paths will be defined as a multiplicative functional with extra regularization conditions.

Definition 2.1.6 (Control) *A control function on $[0,T]$ is a continuous non-negative function ω on the simplex Δ_T which is supper-additive in the sense that*

$$\omega(s,u) + \omega(u,t) \leq \omega(s,t) \quad \forall\, 0 \leq s \leq u \leq t \leq T.$$

It is easy to see that $\omega(t,t) = 0$ for any control ω. In the following, we use the notation $x! = \Gamma(x+1)$, where $\Gamma(\cdot)$ is the Gamma function and x is a positive real number.

Definition 2.1.7 *Let $p \geq 1$ be a real number and $n \geq 1$ be an integer. Denote $\omega : \Delta_T \to [0,+\infty)$ as a control and $\mathbb{X} : \Delta_T \to T^n(\mathbb{R}^d)$ as a multiplicative functional. Then we say that $\mathbb{X}$ has finite p-variation on Δ_T controlled by ω if*

$$\|\mathbb{X}^i_{s,t}\| \leq \frac{\omega(s,t)^{\frac{i}{p}}}{\beta(\frac{i}{p})!} \quad \forall i = 1, \ldots, n, \quad \forall (s,t) \in \Delta_T, \tag{2.6}$$

where $\|\cdot\|$ is the tensor norm induced by the norm on $\mathbb{R}^d$. We will call that $\mathbb{X}$ has finite p-variation in short if there exists a control ω such that (2.6) is satisfied.

Note that in (2.6), β is a constant depending only on p. We are now ready to define the rough paths.

Definition 2.1.8 (Rough Path) *Let $p \geq 1$ be a real number. A p-rough path in $\mathbb{R}^d$ is a multiplicative functional of degree $\lfloor p \rfloor$ with finite p-variation. The space of p-rough paths is denoted by $\Omega_p(\mathbb{R}^d)$.*

Given a continuous path $X : [0,T] \to \mathbb{R}^d$ with bounded p-variation, one can construct a $\lfloor p \rfloor$-rough path $\mathbb{X}$ with $\mathbb{X}^1_{s,t} = X_t - X_s$ for any $s \leq t$. In particular, truncated siganture $S^{\lfloor p \rfloor}(X) \in T^{\lfloor p \rfloor}(\mathbb{R}^d)$ is a p-rough path. The following fundamental theorem of rough paths allows us to make extension of a p-rough path,

Theorem 2.1.9 (Extension Theorem, [114]) *Let $p \geq 1$ be a real number and $n \geq 1$ an integer. Denote $\mathbb{X} : \Delta_T \to T^n(\mathbb{R}^d)$ as a multiplicative functional with finite p-variation controlled be a control ω. Assume that $n \geq \lfloor p \rfloor$, then there exists a unique extension of $\mathbb{X}$ to a multiplicative functional $\Delta_T \to T((\mathbb{R}^d))$ which possesses finite p-variation.*

More precisely, for every $m \geq \lfloor p \rfloor + 1$, there exists a unique continuous function $\mathbb{X}^m : \Delta_T \to (\mathbb{R}^d)^{\bigotimes m}$ such that

$$(s,t) \to \mathbb{X}_{s,t} = \Big(1, \mathbb{X}^1_{s,t}, \dots, \mathbb{X}^{\lfloor p \rfloor}_{s,t}, \dots, \mathbb{X}^m_{s,t}, \dots\Big) \in T((\mathbb{R}^d))$$

is a multiplicative functional with finite p-variation controlled by ω. By this we mean that

$$\|\mathbb{X}^i_{s,t}\| \leq \frac{\omega(s,t)^{\frac{i}{p}}}{\beta(\frac{i}{p})!} \quad \forall i \geq 1, \quad \forall (s,t) \in \Delta_T. \tag{2.7}$$

Signature can be seen as an extension of rough path, and its factorial decay property follows by (2.7). The control function is related to p-variation of path. Given that $x \in \mathcal{V}^p([0,T], \mathbb{R}^d)$, $S^{\lfloor p \rfloor}(x)$ is a p-rough path and one candidate for its control function is

$$\omega(s,t) = \sum_{i=1}^{\lfloor p \rfloor} \sup_{D \subset [s,t]} \sum_k \|x^i_{t_{k+1}} - x^i_{t_k}\|^{p/i}, \tag{2.8}$$

where the norm is the tensor norm induced by Euclidean norm in $\mathbb{R}^d$.

Let $S^{\lfloor p \rfloor}(\Omega_1) = \{S^{\lfloor p \rfloor}(x) : x \in \Omega_1(\mathbb{R}^d)\}$, and $\mathbb{Y}$ be a p-rough path. We call $\mathbb{Y}$ a p-geometric rough path if $\mathbb{Y}$ is in the closure of $S^{\lfloor p \rfloor}(\Omega_1)$ under p-variation metric, where

p-variation metric is given by

$$d_{p\text{-var}}(\mathbb{X}, \mathbb{Y}) := \left(\sup_D \sum_{t_i \in D} \|\mathbb{X}_{t_i,t_{i+1}} - \mathbb{Y}_{t_i,t_{i+1}}\|^p \right)^{1/p}, \qquad \mathbb{X}, \mathbb{Y} \in \Omega_p(\mathbb{R}^d). \tag{2.9}$$

Instead of $T((E))$ in (2.1), the p-rough paths and the geometric p-rough paths are objects in $T^{\lfloor p \rfloor}(E)$ in (2.3) for some real number $p\,(\geq 1)$. The Extension Theorem 2.1.9 stated above implies that there exists a continuous unique lift from $T^{\lfloor p \rfloor}(E)$ to $T((E))$. This lift is made in an iterated integral, and consequently, it gives us the signature of rough paths.

We denote the space of the p-rough paths by Ω_p. The space $G\Omega_p$ of the geometric p-rough paths is defined by the p-variational closure (*cf.* [115] Chapter 3.2) of $S^{\lfloor p \rfloor}(\Omega_1)$. For a path $X : [0, T] \to \mathbb{R}^d$ with the bounded p-variation, the truncated signature belongs to the space of the p-rough paths, i.e., $S^{\lfloor p \rfloor}(X) \in \Omega_p$. If X is of bounded 1-variation, then the truncated signature belongs to the space of the geometric p-rough paths, i.e., $S^{\lfloor p \rfloor}(X) \in G\Omega_p$ for any $p\,(\geq 1)$.

It is manifested that the signature enjoys many nice properties. For example, signature characterizes paths up to tree-like equivalence [13] that are parametrization invariant. Here is a precise statement.

Proposition 2.1.10 (Parametrization Invariance, Lemma 2.12 of [108]) *Denote $X : [0, T] \to \mathbb{R}^d$ a path with bounded variation and $\psi : [0, T] \to [0, T]$ a reparametrization of the time parameter. If we define $\tilde{X}$ by $\tilde{X}_t := X_{\psi(t)}$, then each term in $S(\tilde{X})$ is equal to the corresponding term in $S(X)$, i.e. $S(\tilde{X}) = S(X)$.*

Moreover, if there exists a monotone increasing dimension in the path with bounded variation or geometric rough path, we can get rid of tree-like equivalence [13, 71, 108]. Also, it is easy to specify one path among the parametrization invariance by adding timestamps. In other words, provided that an extra time dimension included, signature characterize geometric rough path uniquely. Another useful fact from rough path theory [39, 114] is that signature terms enjoy a factorial decay as the depth increases, which makes truncating signature reasonable. The following remark shows an example of the factorial decay for bounded 1-variation paths.

Remark 2.1.11 (Factorial Decay, Proposition 2.2 of [115]) *Let $X : [0,T] \to \mathbb{R}^d$ be a continuous path with bounded* 1*-variation, then for every* $k \geq 1$

$$\left\| \int \cdots \int_{0 \leq t_1 < \cdots < t_k \leq T} dX_{t_1} \otimes \cdots \otimes dX_{t_k} \right\| \leq \frac{\|X\|_1^k}{k!}, \tag{2.10}$$

where $\| \cdot \|$ *is the tensor norm.*

All these properties motivate us to use the signature as a feature map in Data Science. We shall then define the linear forms on the signatures.

For simplicity, let us fix $E = \mathbb{R}^d$, and let $\{e_i\}_{i=1}^d$ ($\{e_i^*\}_{i=1}^d$, respectively) be a basis of $\mathbb{R}^d$ (a basis of the dual space $(\mathbb{R}^d)^*$ of $\mathbb{R}^d$, respectively). For every $n \in \mathbb{N}$ and indexes $(i_1, \dots, i_n) \in \{1, \dots, d\}^n$, $(e_{i_1}^* \otimes \cdots \otimes e_{i_n}^*)$ can be naturally extended to $(E^*)^{\otimes n}$ with the basis ($e_I^* = e_{i_1}^* \otimes \cdots \otimes e_{i_n}^*$), and we call $I = i_1 \cdots i_n$ a *word* of length n. The linear actions of $(E^*)^{\otimes n}$ on $E^{\otimes n}$ extends naturally a linear mapping $(E^*)^{\otimes n} \to T((E))^*$ by

$$e_I^*(\mathbf{a}) := e_I^*(a_n), \tag{2.11}$$

for every word I and every element $\mathbf{a} = (a_0, a_1, \dots, a_n, \dots) \in T((E))$.

Let A^* be the collection of all words of length n for all $n \in \mathbb{N}$. Then $\{e_I^*\}_{I \in A^*}$ forms a basis of $T(E^*) = T((\mathbb{R}^d)^*)$. Let $I, J \in A^*$ be two words of lengths m and n with $I = i_1 \cdots i_m$ and $J = j_1 \cdots j_n$, respectively. We say a permutation σ in the symmetric group $\mathfrak{S}_{m+n}$ of $\{1, \ldots, m+n\}$ is a *shuffle* of $\{1, \ldots, m\}$ and $\{m+1, \ldots, m+n\}$, if $\sigma(1) < \cdots < \sigma(m)$ and $\sigma(m+1) < \cdots < \sigma(m+n)$. We denote the collection of all *shuffles* of $\{1, \ldots, m\}$ and $\{1, \ldots, n\}$ by *Shuffles*(m, n).

Definition 2.1.12 (Shuffle Product) *For every pair $I = i_1 \cdots i_m$, $J = j_1 \cdots j_n$ of words of length m and n, the shuffle product $e_I^* \sqcup\!\sqcup\, e_J^*$ of e_I^* and e_J^* is given by*

$$e_I^* \sqcup\!\sqcup\, e_J^* := \sum_{\sigma \in \text{Shuffles}(m,n)} e^*_{(k_{\sigma^{-1}(1)} \cdots k_{\sigma^{-1}(m+n)})}, \tag{2.12}$$

where $k_1 \cdots k_{m+n} = i_1 \cdots i_m j_1 \cdots j_n$.

Denote $T((\mathbb{R}^d))^*$ as the space of linear forms on $T((\mathbb{R}^d))$ induced by $T((\mathbb{R}^d)^*)$. The shuffle product between $f, g \in T((\mathbb{R}^d))^*$ denoted by $f \sqcup\!\sqcup\, g$ can be defined via natural extension of (2.12), by the bi-linearity of $\sqcup\!\sqcup$. It can be shown that $T((\mathbb{R}^d))^*$ is an algebra equipped with shuffle product and element-wise addition restricted to the geometric rough path space $S(\mathcal{V}^p([0,T], \mathbb{R}^d))$, see Theorem 2.15 of [115]. The following proposition motivates us to use the signature as a feature map.

Proposition 2.1.13 (Universal Approximation) *Fix $p \geq 1$, a continuous function $f : \mathcal{V}^p([0,T], \mathbb{R}^d) \to \mathbb{R}$ of finite p-variation, and a compact subset K of $\mathcal{V}^p([0,T], \mathbb{R}^d)$. If $S(x)$ is a p-geometric rough path for each $x \in K$, then for every $\epsilon > 0$, there exists a linear form $l^\epsilon \in T((\mathbb{R}^d))^*$, such that*

$$\sup_{x \in K} |f(x) - \langle l^\epsilon, S(x) \rangle| < \epsilon. \tag{2.13}$$

Proof: The proof follows directly from the uniqueness of signature transform for geometric rough paths and the Stone-Weierstrass theorem. See [113] and Theorem 4.2

in [7] for more details. ■

Remark 2.1.14 (A curse of dimensionality) *By Definition 2.1.2, the truncated signature $S^m(X)$ has a total of $\mathbf{d}_m := \sum_{k=0}^{m} d^k = (d^{m+1}-1)/(d-1)$ many terms for $m \geq 0$. The signature transform is an efficient feature reduction technique, when we have the d dimensional path sampled with high frequency in time. However, when the dimension d is large, the number of signature terms to be computed increases exponentially fast and makes the signature not easily applicable in practice.*

To our best knowledge at this time, only [91] and [141] introduce new algorithms of calculating the kernel of the signatures and [143] discuss the application of the kernel methods to fix this high dimensional problem. We introduce Convolutional Neural Network (CNN) to solve this problem in Section 2.3.

2.2 Classification via Signature

Before we discuss the convolutional neural network in Section 2.3, we consider the application of the signatures to classification problems. In classification problems, we estimate the probability of an object belonging to each class. This estimation problem for the sequential data classification can be solved via the signature.

On a probability space $(\Omega, \mathcal{F}, \mathbb{P})$ consider k classes, *class* 1, *class* 2, ..., *class* k, and n paired independent data $(x^i, y^i)_{1 \leq i \leq n}$, where each $x^i : [0,T] \to \mathbb{R}^d$ is the path data and the corresponding label $y^i \in \{1, \ldots, k\}$ is the class which x^i belongs to. We assume that the labels $y^1, \ldots, y^n$ are sampled from a common distribution and the conditional probability $\mathbb{P}(x^i \in \cdot \mid y^i)$ of x^i, given the class y^i, is a common probability distribution for $i = 1, 2, \ldots, n$. Since we often observe the path dataset at discrete time stamps and we use piece-wise linear interpolations to connect among them, it is reasonable to assume that each path x in the dataset is of bounded 1-variation. Hence, its signature $S(x^i)$ is a geometric 1 rough path.

Definition 2.2.1 (Classification problem) *Our sequential classification problem is stated as follows: given training data $(x^i, y^i)_{1\leq i\leq n}$, derive a classifier g for predicting the labels for unseen data (x, y). Let $p_j(x) := \mathbb{P}(y = j|x)$ for $j = 1, \dots, k$. Our goal is to estimate these conditional probability $p_j(x)$ by $\hat{p}_j(x)$ for the path x of bounded 1-variation and classify x in the class $\arg\max_j \hat{p}_j(x)$ for $j = 1, \dots, k$ as accurate as possible.*

Since the signature $S(x)$ of x determines the path x uniquely, it is reasonable to consider the signature $S(x)$ and a nonlinear continuous function $g : T((\mathbb{R}^d)) \to [0, 1]^k$, such that

$$g(S(x)) = (\hat{p}_1(x), \dots, \hat{p}_k(x))^{\mathrm{T}}, \tag{2.14}$$

where $\hat{p}_j$'s are estimator of p_j's, subject to $\sum_{j=1}^k \hat{p}_j(x) = 1$. Here, T represents the transpose of the vector.

For practical use, we use the truncate signature transforms, thanks to the factorial decay property (Remark 2.1.11) of the signature. With the truncation depth m, we obtain the estimate

$$g(S^m(x)) = (\hat{p}_1(x), \dots, \hat{p}_k(x))^{\mathrm{T}}, \tag{2.15}$$

where $g : T^m(\mathbb{R}^d) \to [0, 1]^k$ is a nonlinear continuous function, and then the predicted label is given by

$$\hat{y} = \arg\max_j \hat{p}_j(x). \tag{2.16}$$

Definition 2.2.2 (Signature Classifier) *We call $h : T((\mathbb{R}^d)) \to [0, 1]$ of the form (2.14) a signature classifier, where $T((\mathbb{R}^d))$ is the tensor algebra and h is a nonlinear continuous function. Naturally, a truncated signature classifier of degree $m \in \mathbb{N}$ is $h : T^m(\mathbb{R}^d) \to [0, 1]$ of the form (2.15).*

In the simple case with only 2 classes, *class* 0 and *class* 1, we consider the following concentration inequalities for classification via signature. We first restate the classification problem for the two classes. Suppose we have the pairwise, independent, identically distributed samples $(X^1, Y^1), \dots, (X^n, Y^n)$ where $Y^i \in \{0,1\}$ and $X^i \in \mathcal{V}^1([0,T], \mathbb{R}^d)$. Let $h : \mathcal{V}^1([0,T], \mathbb{R}^d) \to \{0,1\}$ be a classifier. The training error $\hat{R}_n(h)$ and the true error $R(h)$ are defined by

$$\hat{R}_n(h) = \frac{1}{n}\sum_{i=1}^{n} I(Y^i \neq h(X^i))\,, \qquad \text{and} \qquad R(h) = \mathbb{P}(Y \neq h(X)). \tag{2.17}$$

Here, $I(\cdot)$ is the indicator function. Correspondingly, $R(h) = \mathbb{P}(Y \neq I(h(X) > 0.5))$ and $\hat{R}_n(h) = \frac{1}{n}\sum_{i=1}^{n} I(Y^i \neq I(h(X^i) > 0.5))$. We shall see that $\hat{R}_n(\hat{h}) := \inf_{h\in\mathcal{H}} \hat{R}_n(h)$ is close to $R(h_*) := \inf_{h\in\mathcal{H}} R(h)$, where $\mathcal{H}$ is the collection of the signature classifiers and we assume that $h_* \in \mathcal{H}$. Denote the set

$$\mathcal{E} := \{\sup_{h\in\mathcal{H}} |\hat{R}_n(h) - R(h)| \leq \epsilon\}$$

to be the event that the training error $\hat{R}_n(h)$ is close to the true error $R(h)$ for all classifiers $h \in \mathcal{H}$ in the range of ε, given a fixed $\varepsilon > 0$.

From now on, we assume $\mathcal{H}$ is a compact set of truncated signature classifiers of degee m equipped with metric ρ. The following definition comes from [147].

Definition 2.2.3 (δ-net and covering number) *A set H is called a δ-net for $(\mathcal{H}, \rho)$ if for every $h \in \mathcal{H}$, there exists $\pi(h) \in H$ such that $\rho(h, \pi(h)) < \delta$. The smallest cardinality of a δ-net for $(\mathcal{H}, \rho)$ is called the covering number*

$$N(\mathcal{H}, \rho, \delta) := \inf\{|H| : H \textit{ is a } \delta\textit{-net for } (\mathcal{H}, \rho)\}. \tag{2.18}$$

In our case, we may take the uniform norm ρ, for example. Indeed, by the Ascoli-

Arzelà theorem, we only need $\mathcal{H}$ to be equicontinuous to make it compact, and hence $N_\delta := N(\mathcal{H}, \rho, \delta)$ is always finite for any $\delta > 0$. Let H_δ be a δ-net of $\mathcal{H}$ with cardinality N_δ.

Theorem 2.2.4 *For every $\epsilon > 0$, $\epsilon_0 > 0$, there exist $\delta > 0$ and a corresponding finite covering number N_δ, such that*

$$\mathbb{P}(\sup_{h \in \mathcal{H}} |\hat{R}_n(h) - R(h)| > \epsilon) \leq 2N_\delta \, e^{-2n\epsilon} + \epsilon_0. \tag{2.19}$$

Proof: Take a δ-net H_δ of $\mathcal{H}$ with cardinality N_δ. By the Markov inequality and the definition of the covering number, we have

$$\begin{aligned}\mathbb{P}(\sup_{h \in H_\delta} (\hat{R}_n(h) - R(h)) > \epsilon) &\leq e^{-t\epsilon}\mathbb{E}[\sup_{h \in H_\delta} e^{t(\hat{R}_n(h) - R(h))}] \\ &\leq N_\delta e^{-t\epsilon} \sup_{h \in H_\delta} \mathbb{E}[e^{t(\hat{R}_n(h) - R(h))}].\end{aligned}$$

Since $\hat{R}_n(h)$ is the sum (2.17) of independent random variables, by Hoeffding's inequality [78], we have $e^{-t\epsilon}\mathbb{E}[e^{t(\hat{R}_n(h) - R(h))}] \leq e^{-2n\epsilon}$ for $h \in H_\delta$, $t \geq 0$ and $n \geq 1$. Hence, for every $n \geq 1$ and δ-net H_δ of $\mathcal{H}$, we have

$$\mathbb{P}(\sup_{h \in H_\delta} (\hat{R}_n(h) - R(h)) > \epsilon) \leq N_\delta e^{-t\epsilon} \sup_{h \in H_\delta} \mathbb{E}[e^{t(\hat{R}_n(h) - R(h))}] \leq N_\delta e^{-2n\epsilon}.$$

By a similar argument, we also have $\mathbb{P}(\sup_{h \in H_\delta} (R(h) - \hat{R}_n(h)) > \epsilon) \leq N_\delta e^{-2n\epsilon}$ for every $n \geq 1$ and δ-net H_δ of $\mathcal{H}$.

Combining the above two inequalities, we obtain that for every $n \geq 1$ and δ-net H_δ of $\mathcal{H}$

$$\mathbb{P}(\sup_{h \in H_\delta} |\hat{R}_n(h) - R(h)| > \epsilon) \leq 2N_\delta e^{-2n\epsilon}.$$

By approximating the supremum over $\mathcal{H}$ by the supremum over the sets H_δ with cardinality N_δ, that is,

$$\mathbb{P}(\sup_{h\in\mathcal{H}} |\hat{R}_n(h) - R(h)| > \epsilon) = \lim_{\delta\to 0} \mathbb{P}(\sup_{h\in H_\delta} |\hat{R}_n(h) - R(h)| > \epsilon),$$

we conclude (2.19) that for any $\epsilon_0 > 0$, there exits a $\delta > 0$,

$$\begin{aligned}\mathbb{P}(\sup_{h\in\mathcal{H}} |\hat{R}_n(h) - R(h)| > \epsilon) &< \mathbb{P}(\sup_{h\in H_\delta} |\hat{R}_n(h) - R(h)| > \epsilon) + \epsilon_0 \\ &\le 2N_\delta e^{-2n\epsilon} + \epsilon_0.\end{aligned}$$

■

By Theorem 2.2.4, the event $\mathcal{E}$ holds with high probability provided that n is sufficiently large. On the set $\mathcal{E}$, we have by definitions

$$R(h_*) \le R(\hat{h}) \le \hat{R}_n(\hat{h}) + \epsilon \le \hat{R}_n(h_*) + \epsilon \le R(h_*) + 2\epsilon. \tag{2.20}$$

Thus, it follows that $|R(\hat{h}) - R(h_*)| \le 2\epsilon$ on the set $\mathcal{E}$. Thus, on $\mathcal{E}$, the best empirical signature classifier $\hat{h}$ is close to the best true signature classifier h_* as in (2.20). The connection between signature classifier and general classifier can be constructed by the uniqueness of the signature transform.

This covering number N_δ in Definition 2.2.3 plays an essential role here. The study of the covering number $N(\mathcal{H}, \rho, \delta)$ for the compact set $\mathcal{H}$ of the truncated signature classifiers is still in progress. If we can quantify this number, then the number of training samples n needed for fixed error can be calculated from (2.19).

Example 2.2.5 (GARCH time series) *We give an example of two classes of time* *series, $\{x^n\}_{n=1}^N$, generated by GARCH(2,2) model. The time series are given by*

$$
\begin{aligned}
x_k^n &= \sigma_k \epsilon_k, \\
\sigma_k^2 &= w + \sum_{i=1}^{2} \alpha_i x_{k-i}^n + \sum_{j=1}^{2} \beta_j \sigma_{k-j}^2,
\end{aligned}
$$

where $w > 0$, $\alpha_i \geq 0$, $\beta_j \geq 0$ and ϵ_k's are I.I.D. standard normal distributed. Denote $\boldsymbol{\alpha} = (\alpha_1, \alpha_2)$ and $\boldsymbol{\beta} = (\beta_1, \beta_2)$. 2 classes of GARCH time series are generated by setting parameters in Table 2.2.

class	w	$\boldsymbol{\alpha}$	$\boldsymbol{\beta}$
1	0.5	(0.4, 0.1)	(0.7, 0.5)
2	0.2	(0.8, 0.5)	(0.4, 0.1)

Table 2.2: Parameters for GARCH(2,2) time series.

For paths x^n generated by the first row parameters in Table 2.2, we label $y^n = 1$ (class 1), for the rest paths x^n generated by the second row parameters in Table 2.2, we label them by $y^n = 2$ (class 2). Thus, we generate paired data $\{(x^n, y^n)\}_{n=1}^N$.

Remark 2.2.6 *It is important to note that we cannot directly apply Proposition 2.1.13 here, because this $p(x)$ may not be continuous in x. Intuitively, it is better to add non-linearity on classifier $h(\cdot)$. The experiment in Section 2.3.3 verifies this intuition.*

In practice, the signature classifier (2.15) and its truncation (2.16) can be applied to find the classification model $g(\cdot)$ to estimate $\hat{y}$ in other contexts. In Section 2.3.3, we shall apply the logistic regression to Example 2.2.5, and the result shows that the use of the truncated signature to classify this GARCH(2,2) time series is significantly efficient.

2.3 Convolutional Signature for High Dimensional Sequential Data

The main goal of this section is to introduce the Convolutional Signature (CNN-Sig) model. As we have seen in Remark 2.1.14, the truncated signature suffers from the exponential growth of the number $\mathbf{d}_m$ of terms, when the dimension d is large, and in this case both space and time complexity increase dramatically. We will use Convolutional Neural Network (CNN) to reduce this exponential growth to at most linear growth. CNN has been mostly used in analyzing visual imagery, where it takes advantage of the hierarchical patterns in image and assembles complex patterns by focusing on many small pieces of the picture. Convolutional layer convolves the input data with a small rectangular kernel, and the output data can be masked with an activation function. As there are some patterns between channels of a path, this motivates us to consider the signature with CNN to address the high dimensional problem.

Before introducing the CNN-Sig model, we shall explain that the signature transform can be viewed as a layer in the deep neural network model.

2.3.1 Signature as a Layer

Signature transform can be viewed as a layer in deep neural networks and this is firstly proposed in [86]. In the background of Python package **signatory** [88], signature transform takes input tensor of shape (b, n, d), corresponding to a batch of size b of paths in $\mathbb{R}^d$ with n observing points at times $\{t_j\}_{j=1}^n$, and returns a tensor of shape $(b, \mathbf{d}_m)$ or a stream like tensor of shape $(b, n, \mathbf{d}_m)$, where $\mathbf{d}_m$ is defined in Remark 2.1.14. Usually it omits the first term 1 of the signature transform. Since the signature is also differentiable numerically with respect to each data points, the backpropagation calculation is available.

In this way, the signature can be viewed as a layer in neural network.

2.3.2 Convolutional Signature Model

CNN, which has been proved to be a powerful tool in computer vision, is an efficient feature extraction technique. This idea has been used in [111] as well as the **"Augment"** module [88] (but only 1D CNNs are used). There are two cases of using 1D CNNs. The first case is to extract new sequential features of original paths and then paste them to the original path as extra dimensions. This method is not helpful in the high dimensional case and causes extra difficulty. The second case is that we use extracted sequential features directly from the 1D CNN. It works as a dimension reduction technique but the challenge is that it causes loss of information.

With the favor of the 2D CNN, we are able to reduce the number of signature features and capture all information in the original path at the same time. Since the convolution here is different from the convolution concept in mathematics, we define it and present Example 2.3.2 to show the computational details for those who are not so familiar with CNN.

Definition 2.3.1 (2D Convolution) *Let $*$ be an operation of element-wise matrix multiplication and summation between two matrices of the same shape, that is,*

$$A * B = \sum_{i=1}^{m} \sum_{j=1}^{n} a_{i,j} b_{i,j}, \tag{2.21}$$

where $A := (a_{i,j})_{1 \leq i \leq m, 1 \leq j \leq n}$ and $B := (b_{i,j})_{1 \leq i \leq m, 1 \leq j \leq n}$ of the same size. Suppose the input tensor is $M := (M_{i,j})_{1 \leq i \leq I, 1 \leq j \leq J}$, a kernel window $K := (k_{i,j})_{1 \leq i \leq m, 1 \leq j \leq n}$ and a stride window (s, t). The output $O := (o_{p,q})$ of 2D convolution is given by

$$o_{p,q} := (M_{i,j})_{1+(p-1)s \leq i \leq m+(p-1)s, 1+(q-1)t \leq j \leq n+(q-1)t} * K. \tag{2.22}$$

The shape of the output O depends on how we treat the boundary specifically and does not play a crucial role here.

Example 2.3.2 (2D Convolution) *Let us consider a tensor* $M := (M_{i,j})_{1\leq i,j\leq 5}$ *and a kernel window* $K := (k_{i,j})_{1\leq i,j\leq 3}$*,*

$$M := \begin{pmatrix} 2 & 1 & 0 & 2 & 0 \\ 0 & 1 & 2 & 2 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 2 & 0 & 0 & 2 & 2 \\ 0 & 2 & 0 & 1 & 1 \end{pmatrix}, \quad \textit{and} \quad K := \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & -1 \\ -1 & -1 & -1 \end{pmatrix}, \textit{ respectively,}$$

and a stride window $(1,1)$*. The output will be a* 3×3 *tensor, denoted by* $O = (o_{ij})_{1\leq i,j\leq 3}$*, where each element* $o_{i,j}$ *of* O *is given by the element-wise multiplication and summation of*

$$\widetilde{M}^{i,j} := (M_{k,\ell})_{i\leq k\leq i+2, j\leq \ell\leq j+2}$$

and K*, i.e.,* $o_{i,j} = \widetilde{M}^{i,j} * K$ *for* $1 \leq i,j \leq 3$*. For example,*

$$o_{11} = \begin{pmatrix} 2 & 1 & 0 \\ 0 & 1 & 2 \\ 0 & 0 & 0 \end{pmatrix} * \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & -1 \\ -1 & -1 & -1 \end{pmatrix} = 2\cdot 0 + 1\cdot 1 + 0\cdot 0 + \cdots + 0\cdot(-1) = -1,$$

$$o_{12} = \begin{pmatrix} 1 & 0 & 2 \\ 1 & 2 & 2 \\ 0 & 0 & 1 \end{pmatrix} * \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & -1 \\ -1 & -1 & -1 \end{pmatrix} = 1\cdot 0 + 0\cdot 1 + 2\cdot 0 + \cdots + 1\cdot(-1) = -2,$$

$$o_{13} = \begin{pmatrix} 0 & 2 & 0 \\ 2 & 2 & 1 \\ 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & -1 \\ -1 & -1 & -1 \end{pmatrix} = 1, o_{21} = \begin{pmatrix} 0 & 1 & 2 \\ 0 & 0 & 0 \\ 2 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & -1 \\ -1 & -1 & -1 \end{pmatrix} = -1,$$

and so on. Therefore, the output O is given by

$$O = \begin{pmatrix} -1 & -2 & 1 \\ -1 & -1 & -1 \\ 0 & -5 & -3 \end{pmatrix}.$$

The Convolutional Signature model uses the 2D CNN before the signature transform, and the structure of the convolutional signature model can be described in Figure 2.2. The convolution is implemented in channels. Since the signature is efficient in the time direction, we do not have to convolute the time direction.

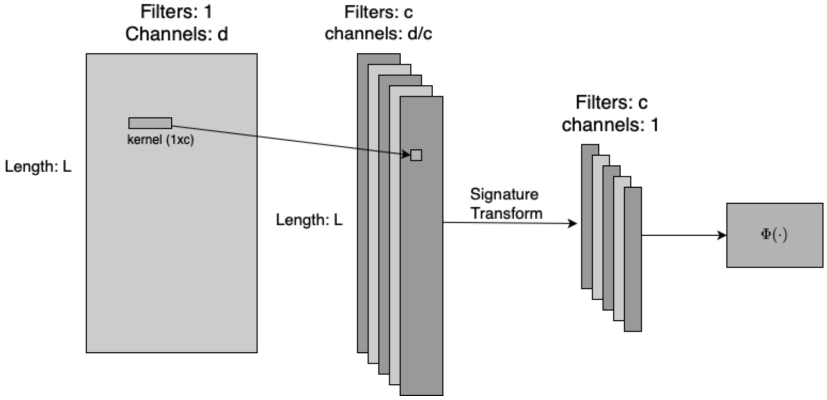


Figure 2.2: Convolutional neural network and signature transform connected by Φ.

Number of Features

Suppose $c\,(\leq d)$ is an integer such that d is divisible by c and let us fix the ratio $\gamma = d/c \in \mathbb{N}$. For the sake of simplicity of explanations, we set the number of features with kernel window of size $(1\times c)$ and stride $(1\times c)$. We illustrate our idea in the following example.

Example 2.3.3 *Let us consider a tensor* $M := (M_{i,j})_{1\leq i\leq 5, 1\leq j\leq 4}$ *and* 2 *kernel windows* $K_1 := (k_i^1)_{1\leq i\leq 2}$, $K_2 := (k_i^2)_{1\leq i\leq 2}$,

$$M := \begin{pmatrix} 2 & 1 & 0 & 2 \\ 0 & 1 & 2 & 2 \\ 0 & 0 & 0 & 1 \\ 2 & 0 & 0 & 2 \\ 0 & 2 & 0 & 1 \end{pmatrix}, \quad K_1 := \begin{pmatrix} -1 & 1 \end{pmatrix} \quad \textit{and} \quad K_2 := \begin{pmatrix} 1 & 2 \end{pmatrix}.$$

By using a stride window $(1,2)$*, we calculate the output* $O = \{O_1, O_2\}$ *with* $O_l = (o_{i,j}^l)_{1\leq i\leq 5, 1\leq j\leq 2}$, $l = 1, 2$. *The computation is done in the same way as in Example 2.3.2:* $o_{1,1}^1 = (2,1) * (-1,1) = -2+1 = -1$, $o_{2,2}^1 = (2,2) * (-1,1) = -2+2 = 0$, $o_{1,1}^2 = (2,1) * (1,2) = 2+2 = 4$, $o_{2,2}^2 = (2,2) * (1,2) = 2+4 = 6$. *Therefore, the output* O *is given by*

$$O_1 = \begin{pmatrix} -1 & 2 \\ 1 & 0 \\ 0 & 1 \\ -2 & 2 \\ 2 & 1 \end{pmatrix}, \quad O_2 = \begin{pmatrix} 4 & 4 \\ 2 & 6 \\ 0 & 2 \\ 2 & 4 \\ 4 & 2 \end{pmatrix}.$$

In this example, since K_1 *and* K_2 *are linear independent, we fully recover the input* M *given* K_1, K_2 *and output* O.

Notice that since the first term in signature transform is always 1, we can omit that, in order to save the computational memory. As shown in Figure 2.2, we start from one d-dimensional path with length L, by using such a convolutional layer, and we are resulted in c paths with each of d/c-dimensional. Then we augment each path with extra time dimension and apply signature transform to each path truncated at depth m, which gives us the number of features

$$N_f := c \cdot \frac{(d/c+1)^{m+1} - d/c - 1}{d/c + 1 - 1} = \frac{(\gamma+1)^{m+1} - \gamma - 1}{\gamma^2} \cdot d \tag{2.23}$$

many features by concatenating all c filters. These features can be used in any following neural network model. For example, a fully connected neural network in the simplest case, or a recurrent neural network (RNN) if we compute the sequence of the signature transform.

The number N_f of features grows linearly in d by increasing c linearly and fixing γ. Instead of optimizing this N_f by setting $\gamma = \arg\min N_f$ directly, we can think γ as a hyperparameter to be tuned to avoid overfitting problem. It can be easily seen that by setting $\gamma = 1$, we reach a minimum of N_f when $m \geq 3$. However, lower γ will give us higher c, which increase the number of parameters in the CNN step. We consider the sum $N_f + (\frac{d}{\gamma})^2$ of number of features and the number of parameters in CNN. Moreover, we can add a multiplier α to the second term, and then define a regularized number on γ,

$$N^\alpha(\gamma) := \frac{(\gamma+1)^m - 1}{\gamma^2} \cdot (\gamma+1) \cdot d + \alpha \cdot \frac{d^2}{\gamma^2}. \tag{2.24}$$

We can select a large real positive number α. This will help us avoid the overfitting problem, when we are concerned about that the CNN layer fits the original paths too well and it sacrifices the prediction power.

One-to-one Mapping

Under the setup in Section 2.3.2, we can generalize Example 2.3.3 and prove such a convolutional layer preserves all information of the original path. Suppose that $\{k^i\}_{i=1}^c$ are all c convolutional kernels with $k^i = (k_1^i, \dots, k_c^i)$ for $i = 1, \dots, c$. Denote the square matrix

$$\mathbf{K} := \begin{pmatrix} k_1^1 & \dots & k_c^1 \\ \vdots & \vdots & \vdots \\ k_1^c & \dots & k_c^c \end{pmatrix}.$$

Let the original path be $\mathbf{x} = (x_{t_1}, \dots, x_{t_n})^{\mathrm{T}}$, $x_{t_j} = \left(x_{t_j}^1, \dots, x_{t_j}^d\right)$ and the output path $\{\tilde{x}_i\}_{i=1}^c$, where $\tilde{x}_i = (\tilde{x}_{t_1,i}, \dots, \tilde{x}_{t_n,i})^{\mathrm{T}}$ with $\tilde{x}_{t_j,i} = \left(\tilde{x}_{t_j,i}^1, \dots, \tilde{x}_{t_j,i}^\gamma\right)$, $1 \le i \le c$. The CNN layer can be represented in equation as

$$\mathbf{K} \cdot \left(x_{t_j}^{lc+1}, \dots, x_{t_j}^{(l+1)c}\right)^{\mathrm{T}} = \left(\tilde{x}_{t_j,1}^l, \dots, \tilde{x}_{t_j,c}^l\right)^{\mathrm{T}}, \;\; 1 \le l \le \gamma, \; 1 \le j \le n. \tag{2.25}$$

Lemma 2.3.4 *If* $\mathbf{K}$ *is of full rank, then this CNN layer is a one-to-one map.*

Proof: Since $\mathbf{K}$ is square and of full rank, it is invertible.

$$\left(x_{t_j}^{lc+1}, \dots, x_{t_j}^{(l+1)c}\right)^{\mathrm{T}} = \mathbf{K}^{-1} \cdot \left(\tilde{x}_{t_j,1}^l, \dots, \tilde{x}_{t_j,c}^l\right)^{\mathrm{T}}, \;\; 1 \le l \le \gamma, \; 1 \le j \le n.$$

If follows that the original path $\mathbf{x}$ can be fully recovered by $\tilde{x} := \{\tilde{x}_i\}_{i=1}^c$. ■

We denote the CNN layer transform as $\mathbf{K} : \mathcal{V}^1([0,T], \mathbb{R}^d) \to \mathcal{V}^1([0,T], \mathbb{R}^{d/c+1})^c$. Here, plus 1 in the dimension $(d/c)+1$ comes from the time dimension we add to each convoluted paths.

In accordance with practical case, we consider approximating functions with domain in a subspace of $\mathcal{V}^1([0,T], \mathbb{R}^d)$ that is observed at finite time stamps and connected by linear interpolation between consecutive points. More precisely, define

$$\begin{aligned}\mathcal{V}^1_D([0,T],\mathbb{R}^d) := \{x \in \mathcal{V}^1([0,T],\mathbb{R}^d) : &\text{ there exist } n \in \mathbb{N} \text{ and } 0 = t_0 < \cdots < t_n = T \\ &\text{such that } x(t) = \frac{t_i - t}{t_i - t_{i-1}} x(t_{i-1}) + \frac{t - t_{i-1}}{t_i - t_{i-1}} x(t_i) \\ &\text{for } t_{i-1} \le t \le t_i, i = 1, \ldots, n\}. \end{aligned} \tag{2.26}$$

Suppose $f : \mathcal{V}^1_D([0,T],\mathbb{R}^d) \to \mathbb{R}$ is the continuous function we need to estimate. Then we have the following theorem.

Theorem 2.3.5 (Approximation by the CNN-Sig model) *Let K be a compact set in $\mathcal{V}^1_D([0,T],\mathbb{R}^d)$. Suppose that f is Lipschitz in K. For any $\epsilon > 0$ there exist a CNN layer $\mathbf{K}$, an integer m, and a neural network model Φ such that*

$$\sup_{x \in K} |f(x) - \Phi \circ S^m \circ \mathbf{K}(x)| < \epsilon.$$

Proof: For every $x \in \mathcal{V}^1_D([0,T],\mathbb{R}^d)$, we rewrite $f(x)$ as a function of $\tilde{x} = \{\tilde{x}_i\}_{i=1}^c$ in (2.25):

$$f(x) = f(\mathbf{K}^{-1}(\tilde{x})) = f \circ \mathbf{K}^{-1}(\tilde{x}) =: h(\tilde{x}). \tag{2.27}$$

It follows that $h = f \circ \mathbf{K}^{-1}$ is a continuous function. Since $S(\tilde{x}_i)$ is a geometric rough path and characterize the path $\tilde{x}_i$ uniquely for each $1 \le i \le c$, there exists a continuous function $\hat{h} : (T(\mathbb{R}))^c \to \mathbb{R}$ such that

$$h(\tilde{x}) = \hat{h}(S(\tilde{x}_1), \ldots, S(\tilde{x}_c)).$$

The existence follows from the compactness and that the signature map is continuous and one-to-one. Moreover, since f is Lipschitz, we have that h is Lipschitz and hence $\hat{h}$

is also Lipschitz. The compactness of K implies that the image of $S \circ \mathbf{K}$ is also compact, hence $h(\tilde{x})$ can be approximate arbitrarily well be truncated signatures up to a uniform truncation depth m for all data in the set K. The existence of such m is induced by the proof of [119, Lemma 4.1] and Lipschitz property. That is, there exists an integer m, such that

$$\sup_{x\in K} |\hat{h}(S(\tilde{x}_1), \dots, S(\tilde{x}_c)) - \hat{h}(S^m(\tilde{x}_1), \dots, S^m(\tilde{x}_c))| \leq \frac{\epsilon}{2}. \tag{2.28}$$

This $\hat{h}$ is not necessarily linear, because there might be some dependence among $\{\tilde{x}_i\}_{i=1}^c$, but it can be approximated by a neural network model arbitrarily well. A wide range of Φ can be chosen. For example, a fully connected shallow neural network with one wide enough hidden layer and some activation function would work, see [65], [48]. That is, there exists Φ such that

$$\sup_{x\in K} \left|\Phi(S^m(\tilde{x}_1), \dots, S^m(\tilde{x}_c)) - \hat{h}(S^m(\tilde{x}_1), \dots, S^m(\tilde{x}_c))\right| \leq \frac{\epsilon}{2}. \tag{2.29}$$

By combining (2.27), (2.28), (2.29) together, we get the desired result. ■

In the CNN-Sig model, the CNN layer can be understood as data dependent encoder which help us find the best way of encoding original path to several lower dimensional paths. On one hand, a large c will result in overfitting problem of CNN layer. On the other hand, small c will produce large number of features for Φ, and then Φ may has the overfitting problem. This tradeoff can be balanced by minimizing $N^\alpha(\gamma)$ in equation (2.24). Thus, although the choice of c does not affect the universality of the model, it could help with resolving the overfitting problem.

Remark 2.3.6 *When we do experiments of the CNN-Sig model, this model works even better compare to plain signature transform of original path on testing data, it is because the CNN-Sig model reduces the number of features and thus overcome the overfitting* *problem better than direct signature transform.*

Moreover, the signature transform can be performed in a sequential way. Then we can choose a RNN model (GRU or LSTM) for Φ. Some other candidates for Φ can be Attention model like Transformer, $1d$-CNN and so on, which might help us get better predictions. Thus, this CNN-Sig model is quite flexible and can be incorporated with many other well developed deep learning model as Φ, which depends specifically on the task. In practice, we can use a different stride size to allow some overlap during convolution and reduce the number of filters. The one-to-one mapping property may be lost in this case if we choose small number of filters, but it results in less overfitting. Another alternative is that we can also convolute over time dimension, provided that correlation over time is of importance to the sequential data.

2.3.3 Experiments

In this section, several results of the experiments are provided for the purpose of exhibiting the performance of the signature classifier and the CNN-Sig model. Sections 2.3.3 and 2.3.3 show that the signature classifier can be a nice candidate for the time series classification problem. In sections 2.3.3 and 2.3.3, we apply the CNN-Sig model to high-dimensional tasks, including the standard high-dimension datasets, approximation of maximum-call European payoff and sentimental analysis.

Classification of GARCH Time Series

The generalized autoregressive conditional heteroskedasticity (GARCH) process is usually used in econometrics to describe the time-varying volatility of financial time series [14, 58]. GARCH provides a more real-world context than other models when predicting the financial time series, compare to other time series model like ARIMA. We apply logistic regression to Example 2.2.5, i.e. the goal is to estimate $g(S^m(x)) = (\hat{p}_0, \hat{p}_1)$ in (2.15), where

$$\log \frac{\hat{p}_1}{1-\hat{p}_1} = \langle l, S^m(x) \rangle, \tag{2.30}$$

subject to $\hat{p}_0 + \hat{p}_1 = 1$, l is a linear functional on $T^m(\mathbb{R}^d)$ to be chosen such that the cross entropy

$$E(l) = -\sum_{i=1}^{N} (y^i \log \hat{p}_i + (1-y^i) \log(1-\hat{p}_i)) \tag{2.31}$$

is minimized, and we predict labels by $\hat{y}^i = \arg\max_i \hat{p}_i$. 500 samples are generated for each class and we use 70% of each class as training data and 30% of each as testing data. By using $m = 4$, we get training accuracy 96.4% and testing accuracy 97.0%. The confusion matrix is given below in Table 2.3.

Predicted True	0	1
0	343	7
1	18	332

Predicted True	0	1
0	147	3
1	6	144

Table 2.3: Training (left) and testing (right) confusion matrics.

Classification of Directed Chain Discrete Time Series

In the study of mean-field interaction and financial systemic risk problems, [51] propose a countably many particle system of diffusion processes, coupled through an infinite, chain-like directed graph, and discuss a detection problem of mean-field interactions among diffusive particles. In Remark 4.5 of [51], a discrete time analogue of the mean-reverting diffusions on the directed chain is also proposed.

We shall discuss a classification problem of such time series data partially observed from the directed chain graph. More specifically, we analyze an identically distributed time series data $\{X_n\}_{n\geq 1}$ and $\{\widetilde{X}_n\}_{n\geq 1}$ parametrized by $a, u \in [0,1]$ and defined recursively by

$$X_n = aX_{n-1} + (1-a)(u\widetilde{X}_{n-1} + (1-u)\mathbb{E}[X_{n-1}]) + \varepsilon_n, \quad n \geq 1, \tag{2.32}$$

where we assume that $X_0 = \widetilde{X}_0 = 0$ for simplicity, the distribution of $\{X_n, n \geq 0\}$ is identical with that of $\{\widetilde{X}_n, n \geq 0\}$ and ε_n, $n \geq 1$ are independent, identically distributed standard normal random variables, independent of $\{\widetilde{X}_n\}_{n\geq 1}$. The parameter $u \in [0,1]$ measures how much X_n depends on its neighborhood and $1-u$ measures how much X_n depends on the common distribution. X and $\widetilde{X}$ have the same distribution with the moving average representation:

$$\begin{aligned} X_n &= \sum_{0\leq l\leq k\leq n-1} \binom{k}{l} u^l (1-a)^l a^{k-l} \epsilon_{n-k,l}, \\ \widetilde{X}_n &= \sum_{0\leq l\leq k\leq n-1} \binom{k}{l} u^l (1-a)^l a^{k-l} \epsilon_{n-k,l+1}, \quad n \geq 1, \end{aligned} \tag{2.33}$$

where $\{\varepsilon_{n,k},\, n, k \geq 0\}$ is an independent, identically distributed array of standard normal random variables.

Suppose that our only observation is $\{X_n\}_{n\geq 1}$, but both $\{\widetilde{X}_n\}_{n\geq 1}$ and u are hidden to us. Our question is that given the access to $\{X_n\}_{n\geq 1}$ generated by different u, can we determine their classes?

In this part, we first set the default parameters and generate training and testing paths according to (2.33). First we initial some parameters: $a = 0.5$, $u = 0.2$ or 0.8 for classification task, $N = 100$ is the time steps, $1/N$ is the variance of ϵ.In order to generate paths, we generate a $n \times (n+1)$ matrix $\mathcal{E}$ of the error terms ϵ, and then pick the column we need for each n. The summation takes time $O(N^2)$ and we have to range

n from 1 to N. The time complexity is the order of $O(N^3)$. We simulate 2000 training paths and 400 testing paths for this task.

Method 1: Logistic Regression In this method, we use 2000 training paths: 1000 for $u = 0.2$ and 1000 for $u = 0.8$. Calculating the signature transform of these paths, augmented with time dimension, up to degree 9, we build a Logistic Regression model on the signatures of training data and test this model, see equation (2.30).

The result is shown in Table 2.4. We observe that signature does capture useful features for u in these special time series.

Training Acc	Testing Acc
0.7465	0.7375

Table 2.4: Training accuracy and testing accuracy on Logistic regression.

Method 2: Deep Neural Network We build a Neural Network model in order to get a better result. We use 4 hidden layers with $256, 256, 128, 2$ units respectively. For first 3 layers, we use "ReLu" as activation function, for last layer, we use "Softmax" activation function as the approximated probability values. After training for 20 epochs, the result is shown in Table 2.5.

Training Acc	Testing Acc
0.8930	0.8925

Table 2.5: Training accuracy and testing accuracy on NN.

This 4 layer neural network model produces better accuracy than logistic regression. The reason follows Remark 2.2.6. Logistic regression trains a linear classifier, but it cannot be used to estimate $p(\cdot)$ efficiently, because $p(\cdot)$ is not continuous in x. This DNN model add nonlinearity to $h(\cdot)$,s and hence works better.

High Dimensional Time Series

Signature is an efficient tool as a feature map for high frequency sequential data to reduces the number of features. However, the number of signature terms increases exponentially as dimension (or channels in the language of PyTorch) increasing. In Section 2.3, we proposed the CNN-Sig model to address this problem. We test our model by applying it in both regression and classification problem.

Experiments - Regression Problem for Maximum-Call Payoff

We investigate our model on a specific rainbow option, high-dimension European type maximum call option. In other words, we want to use our CNN-Sig model to estimate the payoff

$$\max_{1\leq k\leq d}((X_T^k - K)^+),$$

where T is terminal time, K is strike price, superscript k represents the k-th coordinate of this d-dimension path. If X_T^k is smaller than K for all $1 \leq k \leq d$, this payoff is zero. Otherwise the payoff would be the maximum of $(X_T^k - K)$ over those k satisfies $X_T^k \geq K$. Result of this experiment may motivate us to use CNN-Sig model in high dimensional optimal stopping problem from financial mathematics.

Because of the limitation of exponential growth in the number of features, we use lower $d = 6, 10, 12, 20$ to compare the performance between plain signature transform and CNN-Sig model. Then we apply this model to test its performance with higher dimension $d = 50$.

We generate 1000 training paths and 1000 testing paths for cases of $d = 6, 10, 12$, and generate 3000 training paths and 1000 testing paths for case $d = 20$. All stock price paths follows Black-Scholes model.

For all 4 cases, we consider $m = 4$ as the signature depth. For Φ in the CNN-Sig

	Sig+LR				CNN-Sig			
	Training		Testing		Training		Testing	
d	MAE	R^2	MAE	R^2	MAE	R^2	MAE	R^2
6 $(\gamma = 2)$	0.001	1.000	0.101	0.538	0.020	0.986	0.030	0.972
10$(\gamma = 2)$	0.000	1.000	0.124	0.806	0.033	0.988	0.062	0.962
12$(\gamma = 2)$	0.000	1.000	0.153	0.821	0.048	0.981	0.111	0.924
20$(\gamma = 1)$	0.000	1.000	0.225	0.838	0.177	0.916	0.203	0.892

Table 2.6: Training and testing mean absolute error(MAE) and R^2 for the direct signature transform plus linear regression (Sig+LR) and the CNN-Sig model with Φ as a fully connected neural network.

model, we use the same structure, 2 fully connected layers followed by ReLu activation function and then a fully connected layer. We did not apply any technique for avoiding overfitting problem in the CNN-Sig model to make this comparison fair. The result for comparison is shown in Table 2.6. We can see that for all these 4 cases, the CNN-Sig model beat direct signature transform. Since the CNN-Sig model reduce the number of features, it can help avoid overfitting problem compare to Sig+LR. We produce the QQ plots for training and testing results of the CNN-Sig model, see Figure 2.3.

For $d = 50$, where the plain Sig+LR becomes not applicable, we use the same CNN-Sig structure as lower d cases for training. The training MAE is 0.206 with $R^2 = 0.982$ and testing MAE is 0.751 with $R^2 = 0.797$. The QQ plot of training and testing results is in Figure 2.4. In this experiment, we show that CNN-Sig algorithm could be a good candidate in the high dimensional regression problem where plain signature is not applicable. But since CNN-Sig will add non-linearity here, we are not able to price this option in the same way as [7]. This will be left as our future research.

Experiments - Classification

We apply the CNN-Sig model to different high dimensional times series from [9] and [135]. As suggested in [135], all experiments are compared with a benchmark model ROCKET [49]. The results are evaluated over 5 independent trials and listed in Table

2.7. ROCKET is known to be a fast and accurate classification method, the experiment results show that the CNN-Sig model is competitive and fast after a model selection procedure via k-fold cross validation.[1]

Datasets	ROCKET	CNN-Sig
PEMS-SF	0.810(0.014)	**0.817(0.010)**
JapaneseVowels	**0.960(0.002)**	0.940(0.017)
FingerMovement	0.500(0.01)	**0.514(0.034)**
FaceDetection	**0.597(0.004)**	0.553(0.001)
PhonemeSpectra	0.035(0.002)	**0.152(0.006)**
MotorImagery	**0.620(0.007)**	0.524(0.05)
Heartbeat	**0.729(0.011)**	0.723(0.017)
Training Time	353.5	**209.1**

Table 2.7: Testing accuracy, standard deviation and total training time (s) for all high dimensional time series datasets.

Sentiment Analysis by Signature

In Natural Language Processing (NLP), text sentence can be regarded as sequential data. A conventional way to represent words is using high dimensional vector, which is called word embedding. These kind of word embedding is usually of $50, 100, 300$ dimension. Using plain signature transform becomes extremely difficult because of these high dimensions. We apply our CNN-Sig model to address this problem. The dataset we use is IMDB movie reviews, [116].

This IMDB dataset contains 50,000 movie reviews, each of them is labelled by either ”pos” or ”neg”, which represent **Positive** for **Negative** respectively. The IMDB dataset is split into training and testing evenly. For training part, we use 17500 samples for training the model, and use the other 7500 samples as validation dataset. A 100-dimension word embedding GloVe 100d [131] is used as the initial embedding, this high dimension

[1] All experiments are trained on a server with Intel Core i9-9820X (3.30GHz) and four RTX 2080 Ti GPUs

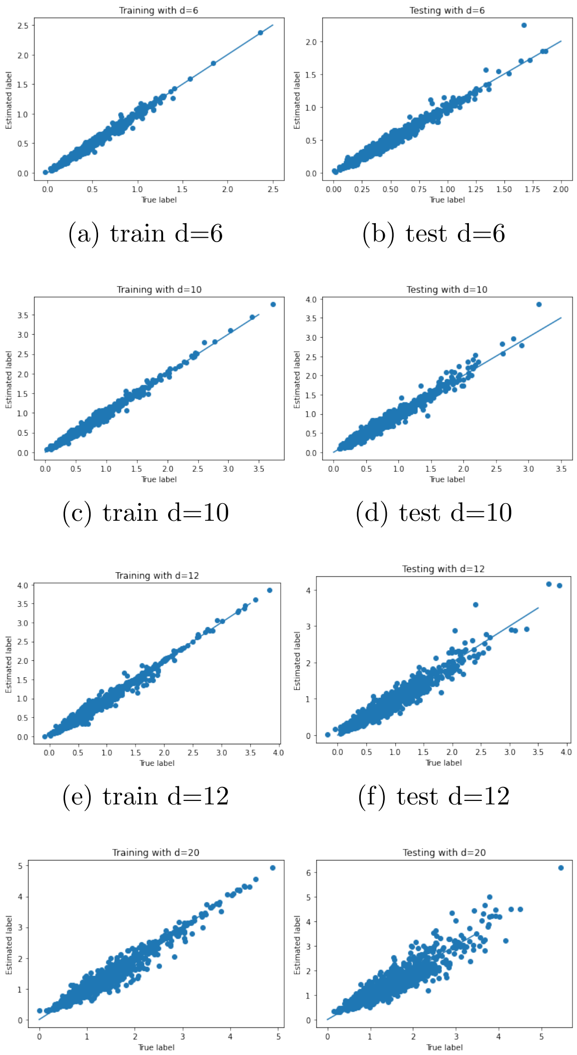
restricts us to use plain signature transform. In our model, by setting γ to be small, we use 1 convolutional 2d layer to reduce the dimension from 100 to c paths with each of $\gamma + 1$ dimensional augmented by extra time dimension. The architecture is shown in Figure 2.5.

The result is shown in Table 2.8 and the testing accuracy has been improved to **86.9%** which is higher than the result in [143] (83%) and Bidirectional LSTM (Bi-LSTM) with 2 hidden layers (0.846%). Moreover, CNN-Sig is a more efficient structure compare to Bi-LSTM in terms of training time and GPU memory usage.

	Bi-LSTM	CNN-Sig
Accuracy	0.846(0.013)	**0.869(0.002)**
Memory	6.8	**1.3**
Time	401.5	**292.5**

Table 2.8: Testing accuracy, GPU memory usage(Gb) during training and total training time(s) on IMDB dataset.

We believe that the CNN-Sig model is a good candidate for feature mapping and easy to be embraced into more complex models. By applying more complicated structure, such as using attention model for Φ and a sliding window, e.g., see [122], for calculating a sequential signature transform, the accuracy can be improved.



(a) train d=6

(b) test d=6

(c) train d=10

(d) test d=10

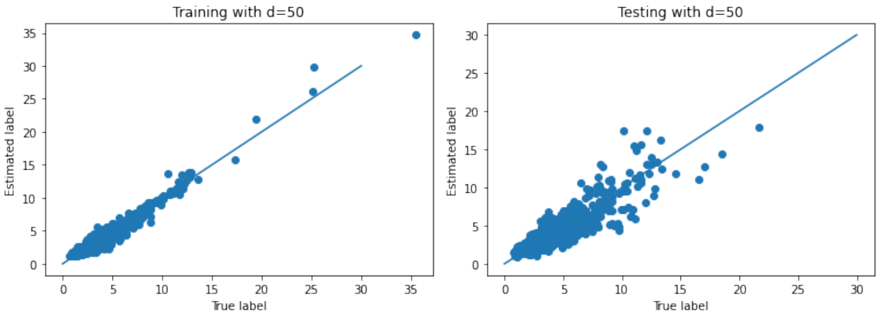
(e) train d=12

(f) test d=12

(g) train d=20

(h) test d=20

Figure 2.3: QQ plot for training and testing result for lower dimensional regression with $d = 6, 10, 12, 20$ using the CNN-Sig model.



(a) d=50

(b) d=50

Figure 2.4: QQ plot for training and testing result for regression task with $d = 50$ using CNN-Sig model.

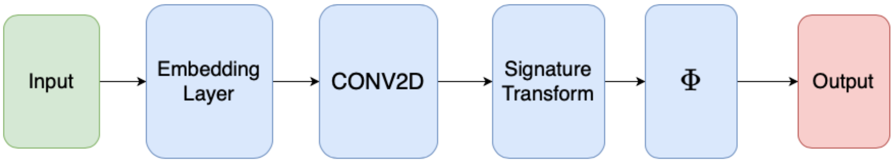


Figure 2.5: Convolutional Signature neural network model for IMDB dataset.

Chapter 3

Signatured Deep Fictitious Play for Mean Field Game with Common Noises

3.1 Background

Stochastic differential games study the strategic interaction of rational decision-makers in an uncertain dynamical system, and have been widely applied to many areas, including social science, system science, and computer science. For realistic models, the problem usually lacks tractability and needs numerical methods. With a large number of players resulting in high-dimensional problems, conventional algorithms soon lose efficiency and one may resort to recently developed machine learning tools [79, 74, 72]. On the other hand, one could utilize its limiting mean-field version, mean-field games (MFGs), to approximate the n-player game for large n (*e.g.*, [75]). Introduced independently in [81, 106], MFGs study the decision making problem of a continuum of agents, aiming to provide asymptotic analysis of the finite player model in which players interact through their empirical distribution. In an MFG, each agent is infinitesimal, whose decision can not affect the population law. Therefore, the problem can be solved by focusing on the optimal decision of a representative agent in response to the average behavior of the entire population and a fixed-point problem (*cf.* equation (3.5)). The MFG model has inspired tremendous applications, not only in finance and economics, such as system risk [32], high-frequency trading [99] and crowd trading [27], but also to population dynamics [1, 54, 2] and sanitary vaccination [82, 56], to list a few. For a systematical introduction of MFGs, see [23, 29, 30].

In MFGs, the random shocks to the dynamical system can be from two sources: idiosyncratic to the individual players and common to all players, *i.e.*, decision-makers face correlated randomness. While MFGs were initially introduced with only idiosyncratic noise as seen in most of the literature, games with common noise, referred to as *MFGs with common noise*, have attracted significant attention recently [104, 31, 4, 68]. The inclusion of common noise is natural in many contexts, such as multi-agent trading in a common stock market, or systemic risk induced through inter-bank lending/borrowing. In reality, players make decisions in a common environment (*e.g.*, trade in the same stock market). Therefore, their states are subject to correlated random shocks, which can be modeled by individual noises and a common noise. In this modeling, observing the state dynamics will be sufficient, and one does not need to observe the noises. These applications make it crucial to develop efficient and accurate algorithms for computing MFGs with common noise.

Theoretically, MFGs with common noise can be formulated as an infinite-dimensional master equation, which is the type of second-order nonlinear Hamilton-Jacobi-Bellman equation involving derivatives with respect to a probability measure. Therefore, direct simulation is infeasible due to the difficulty of discretizing the probability space. An alternative way of solving MFGs with common noise is to formulate it into a stochastic Fokker-Planck/Hamilton-Jacobi-Bellman system, which has a complicated form with common noise, forward-backward coupling, and second-order differential operators. The third kind of approaches turns it into forward backward stochastic differential equations (FBSDE) of McKean-Vlasov type (*cf.* [30, Chapter 2]), which in general requires convexity of the Hamiltonian. For all three approaches, the common assumption is the monotonicity condition that ensures uniqueness. Regarding simulation, existing deep learning methods fix the sampling common noise paths and then solve the corresponding MFGs, which leads to a nested-loop structure with millions of simulations of common noise paths to produce accurate predictions for unseen common shock realizations. Then the computational cost becomes prohibitive and limits the applications to a large extent.

In this paper, we solve MFGs with common noise by directly parameterizing the optimal control using deep neural networks in spirit of [73], and conducting a global optimization. We integrate the signature from rough path theory, and fictitious play from game theory for efficiency and accuracy, and term the algorithm *Signatured Deep Fictitious Play* (Sig-DFP). The proposed algorithm avoids solving the three aforementioned complicated equations (master equation, Stochastic FP/HJB, FBSDE) and does not have uniqueness issues.

Contribution. We design a novel efficient single-loop deep learning algorithm, Sig-DFP, for solving MFGs with common noise by integrating fictitious play [17] and Signature [115] from rough path theory. To our best knowledge, this is the first work focusing on the common noise setting, which can address heterogeneous MFGs and heterogeneous extended MFGs, both with common noise.

We prove that the Sig-DFP algorithm can reach mean-field equilibria as both the depth M of the truncated signature and the stage n of the fictitious play approaching infinity, subject to the universal approximation of neural networks. We demonstrate its convergence superiority on three benchmark examples, including homogeneous MFGs, heterogeneous MFGs, and heterogeneous extended MFGs, all with common noise, and with assumptions even beyond the technical requirements in the theorems. Moreover, the algorithm has the following advantages:

1. Temporal and spacial complexity are $\mathcal{O}(NLp + Np^2)$ and $\mathcal{O}(NLp)$, compared to $\mathcal{O}(N^2L)$ (for both time and space) in existing machine learning algorithms, with N as the sample size, L as the time discretization size, $p = \mathcal{O}(n_0^M)$, n_0 as the dimension of common noise.

2. Easy to apply the fictitious play strategy: only need to average over linear functionals with $\mathcal{O}(1)$ complexity.

Related Literature. After MFGs firstly introduced by [81] and [106] under the setting of a continuum of homogeneous players but without common noise, it has been extended to many applicable settings, *e.g.*, heterogeneous players games [105, 102] and major-minor players games [80, 127, 34]. A recent line of work studies MFGs with common noise [32, 12, 4, 25]. Despite its theoretical progress and importance for applications, efficient numerical algorithms focusing on common noise settings are still missing. Our work will fill this gap by integrating machine learning tools with learning procedures from game theory and signature from rough path theory.

Fictitious play was firstly proposed in [17, 18] for normal-form games, as a learning procedure for finding Nash equilibria. It has been widely used in the Economic literature, and adapted to MFGs [26, 15] and finite-player stochastic differential games [79, 74, 72, 148].

Using machine learning to solve MFGs has also been considered, for both model-based setting [33, 136, 112] and model-free reinforcement learning setting [69, 142, 5, 57], most of which did not consider common noise. Existing machine learning methods for MFGs with common noise were studied in [132], which have a nested-loop structure and require millions of simulations of common noise paths to produce accurate predictions for unseen common shock realizations.

3.2 Mean Field Games with Common Noise

We first introduce the following notations to precisely define MFGs with common noise. For a fixed time horizon T, let $(W_t)_{0\leq t\leq T}$ and $(B_t)_{0\leq t\leq T}$ be independent n- and n_0-dimensional Brownian motions defined on a complete filtered probability space $(\Omega, \mathcal{F}, \mathbb{F} = \{\mathcal{F}_t\}_{0\leq t\leq T}, \mathbb{P})$. We shall refer W as the *idiosyncratic noise* and B as the *common noise* of the system. Let $\mathcal{F}_t^B$ be the filtration generated by $(B_t)_{0\leq t\leq T}$, and $\mathcal{P}^p(\mathbb{R}^d)$ be the collection of probability measures on $\mathbb{R}^d$ with finite p^{th} moment, *i.e.*, $\mu \in \mathcal{P}^p(\mathbb{R}^d)$ if

$$\left(\int_{\mathbb{R}^d} \|x\|^p \,\mathrm{d}\mu(x)\right)^{1/p} < \infty. \tag{3.1}$$

We denote by $\mathcal{M}([0,T];\mathcal{P}^2(\mathbb{R}^d))$ the space of continuous $\mathcal{F}^B$-adapted stochastic flow of probability measures with the finite second moment, and by $\mathcal{H}^2([0,T];\mathbb{R}^m)$ the set of all $\mathcal{F}$-progressively measurable $\mathbb{R}^m$-valued square-integrable processes.

Next, we introduce the concept of MFGs with common noise. Given an initial distribution $\mu_0 \in \mathcal{P}^2(\mathbb{R}^d)$, and a stochastic flow of probability measures $\mu = (\mu_t)_{0\leq t\leq T} \in \mathcal{M}([0,T];\mathcal{P}^2(\mathbb{R}^d))$, we consider the stochastic control

$$\inf_{(\alpha_t)_{0\leq t\leq T}} \mathbb{E}[\int_0^T f(t, X_t, \mu_t, \alpha_t)\,\mathrm{d}t + g(X_T, \mu_T)], \tag{3.2}$$

$$\begin{aligned}\text{where } \mathrm{d}X_t &= b(t, X_t, \mu_t, \alpha_t)\,\mathrm{d}t + \sigma(t, X_t, \mu_t, \alpha_t)\,\mathrm{d}W_t \\ &\quad + \sigma^0(t, X_t, \mu_t, \alpha_t)\,\mathrm{d}B_t, \end{aligned} \tag{3.3}$$

with $X_0 \sim \mu_0$. Here the representative agent controls his dynamics X_t through a $\mathbb{R}^m$-dimensional control process α_t, and the drift coefficient b, diffusion coefficients σ and

σ^0, running cost f and terminal cost g are all measurable functions, with (b, σ, σ^0, f) : $[0,T] \times \mathbb{R}^d \times \mathcal{P}^2(\mathbb{R}^d) \times \mathbb{R}^m \to \mathbb{R}^d \times \mathbb{R}^{d\times n} \times \mathbb{R}^{d\times n_0} \times \mathbb{R}$, and $g : \mathbb{R}^d \times \mathcal{P}^2(\mathbb{R}^d) \to \mathbb{R}$.

Note that since μ is stochastic, (3.2)–(3.3) is a control problem with random coefficients.

Definition 3.2.1 (Mean-field equilibrium) *The control-distribution flow pair $\alpha^* = (\alpha_t^*)_{0\le t\le T} \in \mathcal{H}^2([0,T];\mathbb{R}^m)$, $\mu^* \in \mathcal{M}([0,T];\mathcal{P}^2(\mathbb{R}^d))$ is a mean-field equilibrium to the MFG with common noise, if α^* solves* (3.2) *given the stochastic measure flow μ^*, and the conditional marginal distribution of the optimal path $X_t^{\alpha^*}$ given the common noise B coincides with the measure flow μ^*:*

$$\mu_t^* = \mathcal{L}(X_t^{\alpha^*}|\mathcal{F}_t^B), \tag{3.4}$$

where $\mathcal{L}(\cdot|\mathcal{F})$ is the conditional law given a filtration $\mathcal{F}$.

We remark that, with a continuum of agents, the measure μ^* is not affected by a single agent's choice, and the MFG is a standard control problem plus an additional fixed-point problem. More precisely, denote by $\hat{\alpha}^\mu$ the optimal control of (3.2)–(3.3) given the stochastic measure flow $\mu \in \mathcal{M}([0,T];\mathcal{P}^2(\mathbb{R}^d))$, then μ^* is a fixed point of

$$\mu_t = \mathcal{L}(X_t^{\hat{\alpha}^\mu}|\mathcal{F}_t^B). \tag{3.5}$$

MFGs without common noise: Note that with $\sigma^0 \equiv 0$, (3.2)–(3.3) is a MFG without common noise, and the flow of measures μ_t becomes deterministic.

Extended MFGs: In extended mean field games, the interactions between the representative agent and the population happen via both the states and controls, thus the functions $(b, \sigma, \sigma^0, f, g)$ can also depend on $\mathcal{L}(\alpha_t|\mathcal{F}_t^B)$.

3.3 Fictitious Play

The Signatured Deep Fictitious Play (Sig-DFP) algorithm is built on fictitious play, and propagates conditional distributions $\mu = \{\mu_t\}_{0\leq t\leq T} \in \mathcal{M}([0,T];\mathcal{P}^2(\mathbb{R}^d))$ by signatures. This section briefly introduces these two ingredients.

In the learning procedure of *fictitious play*, players myopically choose their best responses against the empirical distribution of others' actions at every subsequent stage after arbitrary initial moves. When [26, 27] extended it to mean-field settings, the empirical distribution of actions is naturally replaced by the average of distribution flows. More precisely, let $\bar{\mu}^{(0)} \in \mathcal{M}([0,T];\mathcal{P}^2(\mathbb{R}^d))$ be the initial guess of μ^* in (3.4), and consider the following iterative algorithm: (1) take $\bar{\mu}^{(n-1)} \in \mathcal{P}^2(\mathbb{R}^d)$ as the given flow of measures in (3.2)–(3.3) for the n-th iteration, and solve the optimal control in (3.2) denoted by $\alpha^{(n)}$; (2) solve the controlled stochastic differential equation (SDE) (3.3) for $X^{\alpha^{(n)}}$ and then infer the conditional distribution flow $\mu^{(n)} = \mathcal{L}(X^{\alpha^{(n)}}|\mathcal{F}_t^B)$; (3) average distributions $\bar{\mu}^{(n)} = \frac{n-1}{n}\bar{\mu}^{(n-1)} + \frac{1}{n}\mu^{(n)}$ and pass $\bar{\mu}^{(n)}$ to the next iteration. If $\mu^{(n)}$ converges and the strategy corresponding to the limiting measure flow is admissible, then by construction, it is a fixed-point of (3.5) and thus a mean-field equilibrium.

3.4 The Sig-DFP Algorithm

We introduce two shorthand notations: if x is a path indexed by $t \in [0,T]$, then $x := (x_t)_{0\leq t\leq T}$ denotes the whole path and $x_{s:t} := (x_u)_{s\leq u\leq t}$ denotes the path between s and t.

3.4.1 Propagation of Distribution with Signatures

With the presence of common noise, existing algorithms mostly consider a nested-loop structure, with the inner one for idiosyncratic noise W and the outer one for common noise B. More precisely, if one works with N idiosyncratic Brownian paths $\{W^k\}_{k=1}^N$ and N common Brownian paths $\{B^k\}_{k=1}^N$, then for each B^j, one needs to simulate N paths $\{X^{i,j}\}_{i=1}^N$ defined by (3.3) over all idiosyncratic Brownian paths and solve the problem (3.2) associated to B^j. This requires a total of N^2 simulations of (3.3). With a sufficiently large N, $\mu_t = \mathcal{L}(X_t|\mathcal{F}_t^B)$ is approximated well by $\frac{1}{N^2}\sum_{i,j=1}^N \delta_{X_t^{i,j}} \mathbb{1}_{\omega^{(0,j)}}$ with $\omega^{0,j} \in \Omega$ corresponding to the trajectory B^j. The double summation is of $\mathcal{O}(N^2)$ which is computationally expensive for large N.

We shall address the aforementioned numerical difficulties by signatures. The key idea is to approximate μ_t by

$$\begin{aligned}&\mu_t \equiv \mathcal{L}(X_t|\mathcal{F}_t^B) = \mathcal{L}(X_t|S(\hat{B}_t)) \approx \mathcal{L}(X_t|S^M(\hat{B}_t)),\\ &\text{with } \hat{B}_t = (t, B_t),\end{aligned} \tag{3.6}$$

where the equal sign comes from the unique characterization of signatures $S(\hat{B})$ to the paths $B_{0:t}$, and the approximation is accurate for large M due to the factorial decay property of the signature. The last term is then computed by machine learning methods, *e.g.*, by Generative Adversarial Networks (GANs). In addition, if the agents interact via some population average subject to common noise: $\mu_t = \mathbb{E}[\iota(X_t)|\mathcal{F}_t^B]$, the approximation in (3.6) can be arbitrarily close to the true measure flow for sufficiently large M. The following lemma gives a precise statement.

Lemma 3.4.1 *Suppose $\mu_t = \mathbb{E}[\iota(X_t)|\mathcal{F}_t^B]$ where $\iota : \mathbb{R}^d \to \mathbb{R}$ is a measurable function. View μ_t as $\mu(t, B_{0:t})$ with $\mu : \mathcal{V}^p([0,T], \mathbb{R}^{n_0+1}) \to \mathbb{R}$ continuous for some $p \in (2,3)$, and* *let $K \subset \mathcal{V}^p([0,T], \mathbb{R}^{n_0+1})$ be a compact set, then for any $\epsilon > 0$, there exist a positive integer M and a linear functional $l \in T((\mathbb{R}^{n_0+1}))^*$, such that*

$$\sup_{t\in[0,T]} \sup_{\hat{B}\in K} |\mu_t - \langle l, S^M(\hat{B}_{0:t})\rangle| < \epsilon. \tag{3.7}$$

Proof: See Appendix A.1 for details due to the page limit. ■

With all the above preparations, we now explain how the approximation to $\mu = \{\mu_t\}_{0\leq t\leq T}$ using signatures is implemented. Given N pairs of idiosyncratic and common Brownian paths (W^i, B^i) and assume α_t in (3.3) is already obtained (which will be explained in Section 3.4.2), we first sample the optimized state processes $(X_t^i)_{0\leq t\leq T}$, producing N samples $\{X^i\}_{i=1}^N$. Then the linear functional l in Lemma 3.4.1 is approximated by implementing linear regressions on $\{S^M(\hat{B}_{0:t}^i)\}_{i=1}^N$ with dependent variable $\{\iota(X_t^i)\}_{i=1}^N$ at several time stamps t, *i.e.*,

$$\begin{aligned} \hat{l} =& \arg\min_{\boldsymbol{\beta}} \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|^2, \\ &\boldsymbol{y} = \{\iota(X_t^i)\}_{i=1}^N,\ \boldsymbol{X} = \{S^M(\hat{B}_{0:t}^i)\}_{i=1}^N. \end{aligned} \tag{3.8}$$

In all experiments in Section 3.5, we get decent approximations of μ on $[0,T]$ by considering only three time stamps $t = 0, \frac{T}{2}, T$. Note that such a framework can also deal with multi-dimensional ι, where the regression coefficients become a matrix.

The choice in (3.8) is mainly motivated by Lemma 3.4.1 stating l is a linear functional, and by the probability model underlying ordinary linear regression (OLS) which interprets that the least square minimization (3.8) gives the best prediction of $E[\boldsymbol{y}|\boldsymbol{X}]$ restricting to linear relations. There are other benefits for choosing OLS: Once $\hat{l}$ is obtained in (3.8), the prediction for unseen common paths is efficient: $\mu_t(\tilde{\omega}) \approx \langle \hat{l}, S^M(\hat{B}_{0:t}(\tilde{\omega}))\rangle$ for any $\tilde{\omega}$ and t. Moreover, it is easy to integrate with fictitious play: averaging $\mu_t^{(n)}$ from different iterations, commonly needed in fictitious play, now means simply averaging $\hat{l}^{(n)}$ over n. Next, we analyze the temporal and spatial complexity of using signatures and linear regression as below.

Temporal Complexity: Suppose we discretize $[0, T]$ into L time stamps: $0 = t_0 \leq t_1 \leq \ldots \leq t_L = T$, and simulate N paths of W, B and X_t. The simulation cost is of $\mathcal{O}(NL)$. For computing the truncated signature $S^M(\hat{B})$ of depth M, we use the Python package Signatory [89], yielding a complexity of $\mathcal{O}(NLp)$ where $p = \frac{(n_0+1)^{M+1}-1}{n_0} = \mathcal{O}(n_0^M)$. Note that one can choose a large N and reuse all sampled common noise paths B for each iteration of fictitious play, thus the computation of $S^M(B)$ is done only once, and $S^M(\hat{B}_{0:t})$ is accessible in constant time for all t. The linear regression[1] (or Ridge regression) takes time $\mathcal{O}(Np^2)$. Thus, the total temporal complexity is of $\mathcal{O}(NLp + Np^2)$, which is linear in N given[2] $p \ll N$. Comparing to the nested-loop algorithm, where the cost of simulating SDEs is $\mathcal{O}(N^2 L)$ and computing conditional distribution flows takes time $\mathcal{O}(N^2 L)$, we claim that our algorithm reduced the temporal complexity by a factor of the sample size N by using signatures.

Spatial Complexity: In fictitious play, one may choose to average all past flow of measures $\mu^{(n)}$ as the given measures in (3.2)–(3.3) for the current iteration. Using signatures simplifies it to average $\hat{l}^{(n)}$. To update it between iterations, one needs to store the current average which costs $\mathcal{O}(p)$ of the memory. Combining $\mathcal{O}(NL)$ and $\mathcal{O}(NLp)$ for storing SDEs and truncated signatures, the overall spacial complexity is $\mathcal{O}(NLp)$. The complexity of the nested-loop case is again $\mathcal{O}(N^2 L)$, which we reduce by a factor of N.

We conclude this section by the following remark: For the general case $\mu_t = \mathcal{L}(X_t | \mathcal{F}_t^B)$, though the linear regression is no longer available, the one-to-one mapping between μ and $S(\hat{B})$ persists. Therefore, one can train a Generative Adversarial Network (GAN,

[1] We use the Python package scikit-learn [130] to do the linear regression.

[2] M is usually small due to the factorial decay property of the signature. For n_0 not large, we have $p \ll N$.

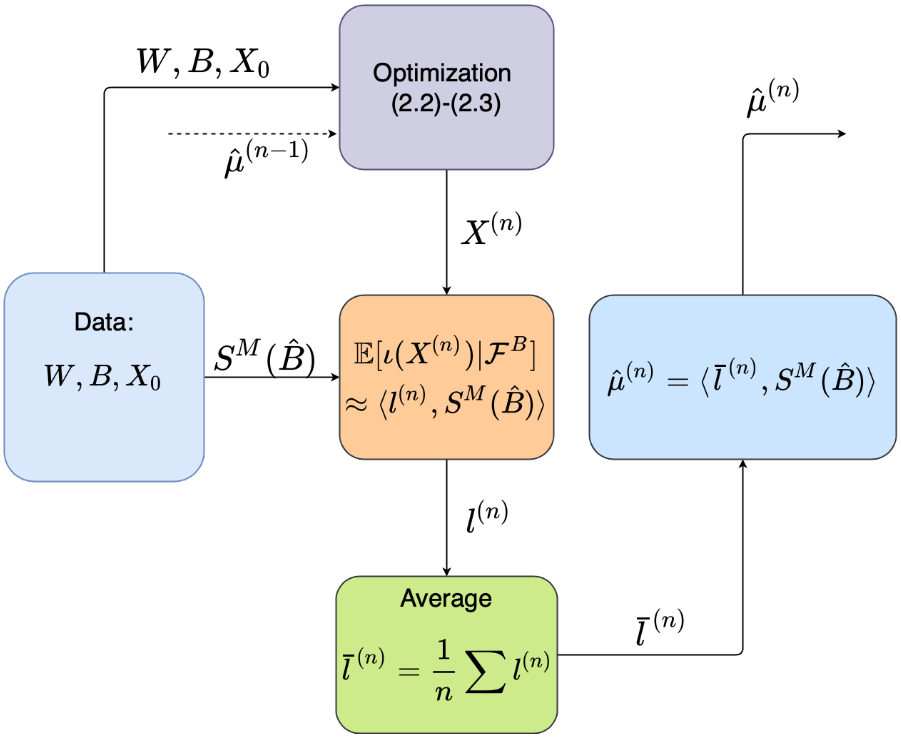

W, B, X_0
Optimization (2.2)-(2.3)
$\hat{\mu}^{(n-1)}$
$\hat{\mu}^{(n)}$
$X^{(n)}$
Data:
W, B, X_0
$S^M(\hat{B})$
$\mathbb{E}[\iota(X^{(n)})|\mathcal{F}^B] \approx \langle l^{(n)}, S^M(\hat{B})\rangle$
$\hat{\mu}^{(n)} = \langle \bar{l}^{(n)}, S^M(\hat{B})\rangle$
$l^{(n)}$
Average
$\bar{l}^{(n)} = \frac{1}{n}\sum l^{(n)}$
$\bar{l}^{(n)}$

Figure 3.1: Flowchart of one iteration in the Sig-DFP Algorithm. Input: idiosyncratic noise W, common noise B, initial position X_0 and measure flow $\hat{\mu}^{(n-1)}$ from the last iteration. Output: measure flow $\hat{\mu}^{(n)}$ for the next iteration.

[67]) for generating samples following the distribution μ by taking truncated signatures as part of the network inputs.

3.4.2 Deep Learning Algorithm

Having explained the key idea on how to approximate μ efficiently, we describe the Sig-DFP algorithm in this subsection. The algorithm consists of repeatedly solving (3.2)–(3.3) for a given measure flow μ using deep learning in the spirit of [73], and passing the yielded μ to the next iteration by using signatures. The flowchart of the idea is illustrated in Figure 3.1. Consider a partition π of $[0, T] : 0 = t_0 < \cdots < t_L = T$, denote by $\hat{\mu}^{(n-1)}$ the given flow of measures at stage n, the stochastic optimal control problem (3.2)–(3.3) is solved by

$$\begin{aligned} \inf_{\{\alpha_k\}_{k=0}^{N-1}} \frac{1}{N} \sum_{i=1}^{N} \bigg(\sum_{k=0}^{L-1} & f(t_k, X_k^i, \hat{\mu}_k^{(n-1)}(\omega^i), \alpha_k^i)\Delta_k \\ & + g(X_L^i, \hat{\mu}_L^{(n-1)}(\omega^i)) \bigg), \qquad (3.9) \\ \text{where } X_{k+1}^i = X_k^i & + b(t_k, X_k^i, \hat{\mu}_k^{(n-1)}(\omega^i), \alpha_k^i)\Delta_k \\ & + \sigma(t_k, X_k^i, \hat{\mu}_k^{(n-1)}(\omega^i), \alpha_k^i)\Delta W_k^i \\ & + \sigma^0(t_k, X_k^i, \hat{\mu}_k^{(n-1)}(\omega^i), \alpha_t^i)\Delta B_k^i, \qquad (3.10) \end{aligned}$$

where we replace the subscript t_k by k to simplify notations, and let $\Delta_k = t_{k+1} - t_k$, $\Delta W_k^i = W_{t_{k+1}}^i - W_{t_k}^i$, $\Delta B_k^i = B_{t_{k+1}}^i - B_{t_k}^i$. Here, we use the superscript i to represent the i^{th} sample path and $\hat{\mu}_k^{(n-1)}(\omega^i)$ to emphasize the stochastic measure's dependence on the i^{th} sample path of B up to time t_k. The control α_k is then parameterized by neural networks (NNs) in the feedback form:

$$\alpha_k^i := \alpha_\varphi(t_k, X_k^i, \hat{\mu}_k^{(n-1)}(\omega^i); \varphi), \qquad (3.11)$$

where α_φ denotes the NN map with parameters φ, and searching the infimum in (3.9) is translated into minimizing φ. The yielded optimizer φ^* gives $\alpha_k^{i,*}$, with which the optimized state process paths $\{X^{i,*}\}_{i=1}^N$ are simulated and its conditional law $\mathcal{L}(X^*|\mathcal{F}^B)$, denoted by $\mu^{(n)}$, is approximated using signatures as described in Section 3.4.1. This finishes one iteration of fictitious play. Denote by $\tilde{\mu}^{(n)}$ the approximation of $\mu^{(n)}$, we then pass $\tilde{\mu}^{(n)}$ to the next iteration via updating $\hat{\mu}^{(n)} = \frac{1}{n}\tilde{\mu}^{(n)} + \frac{n-1}{n}\hat{\mu}^{(n-1)}$ by averaging the coefficients in (3.8).

We summarize it in Algorithm 1, with implementation details deferred to Appendix A.2. Note that the simulation of $X^{i,(n)}$ and $J_B(\varphi, \bar{\mu}^{(n-1)})$ uses the equations (A.6) and (A.5)

Algorithm 1 The Sig-DFP Algorithm

```
Input: $b, \sigma, \sigma_0, f, g, \iota$ and $X_0^i, (W_{t_k}^i)_{k=0}^L, (B_{t_k}^i)_{k=0}^L$ for $i = 1, 2, \ldots, N$; $N_{\text{round}}$: rounds for FP;
$B$: minibatch size; $N_{\text{batch}}$: number of minibatches.
Compute the signatures of $\hat{B}^i_{0:t_k}$ for $i = 1, \ldots, N$, $k = 1, \ldots, L$;
Initialize $\hat{\mu}^{(0)}$, $\varphi$;
for $n = 1$ to $N_{\text{round}}$ do
    for $r = 1$ to $N_{\text{batch}}$ do
        Simulate the $r^{th}$ minibatch of $X^{i,(n)}$ using $\hat{\mu}^{(n-1)}$ and compute $J_B(\varphi, \hat{\mu}^{(n-1)})$;
        Minimize $J_B(\varphi, \hat{\mu}^{(n-1)})$ over $\varphi$, then update $\alpha_\varphi$;
    end for
    Simulate $X^{i,(n)}$ with the optimized $\alpha^*_\varphi$, for $i = 1, \ldots, N$;
    Regress $\iota(X_0^{i,(n)}), \iota(X_{L/2}^{i,(n)}), \iota(X_L^{i,(n)})$ on $S^M(\hat{B}^i_{0:0})$, $S^M(\hat{B}^i_{0:t_{L/2}})$, $S^M(\hat{B}^i_{0:t_L})$ to get $l^{(n)}$;
    Update $\bar{l}^{(n)} = \frac{n-1}{n}\bar{l}^{(n-1)} + \frac{1}{n}l^{(n)}$;
    Compute $\hat{\mu}^{(n)}$ by $\hat{\mu}^{(n)}_k(\omega^i) = \langle \bar{l}^{(n)}, S^M(\hat{B}^i_{0:t_k})\rangle$, for $i = 1, 2, \ldots, N, k = 1, \ldots, L$;
end for
Output: the optimized $\alpha^*_\varphi$ and $\bar{l}^{(N_{\text{round}})}$.
```

in Appendix A.2, respectively.

Theorem 3.4.2 (Convergence analysis) *Let (α^*,μ^*) be the mean-field equilibrium in Definition 3.2.1, $\alpha^{(n)}$ be the optimal control, and $\mu^{(n)}$ be the measure flow of the optimized state process after the n^{th} iteration of fictitious play, and $\tilde{\mu}^{(n)}$ be the approximation by truncated signatures. Under Assumption A.3.1 and $\sup_{t\in[0,T]} \mathbb{E}[\mathcal{W}_2^2(\tilde{\mu}_t^{(n)}, \mu_t^{(n)})] \leq \epsilon$, we have*

$$\sup_{t\in[0,T]} \mathbb{E}[\mathcal{W}_2^2(\tilde{\mu}_t^{(n)}, \mu_t^*)] + \int_0^T \mathbb{E}|\alpha_t^{(n)} - \alpha_t^*|^2 \,\mathrm{d}t$$
$$\leq C(q^n \sup_{t\in[0,T]} \mathbb{E}[\mathcal{W}_2^2(\mu_t^{(0)}, \mu_t^*)] + \epsilon),$$

for some constants $C > 0$ and $0 < q < 1$, where $\mathcal{W}_2$ denotes the 2-Wasserstein metric.

Moreover, if we consider a partition of $[0, T] : 0 = t_0 < \cdots < t_L = T$, and define $\pi(t) = t_k$ for $t \in [t_k, t_{k+1})$ with $\|\pi\| = \max_{1\leq k<L} |t_k - t_{k-1}|$, then

Theorem 3.4.3 (Convergence in discrete time) *Let $\mu_{t_k}^{(n)}$ be the conditional law of the discretized optimal process $X_{t_k}^{(n)}$ after the n^{th} iteration of fictitious play (cf.* (3.10)*), and $\tilde{\mu}_{t_k}^{(n)}$ be the approximation by truncated signatures. Under Assumption A.3.1 and $\sup_{0\leq k\leq L} \mathbb{E}[\mathcal{W}_2^2(\tilde{\mu}_{t_k}^{(n)}, \mu_{t_k}^{(n)})] \leq \epsilon$, one has*

$$\sup_{t\in[0,T]} \mathbb{E}[\mathcal{W}_2^2(\tilde{\mu}_{\pi(t)}^{(n)}, \mu_t^*)] + \int_0^T \mathbb{E}|\alpha_{\pi(t)}^{(n)} - \alpha_t^*|^2 \,\mathrm{d}t$$

$$\leq C(q^n \sup_{0\leq k\leq L} \mathbb{E}[\mathcal{W}_2^2(\mu_{t_k}^{(0)}, \mu_{t_k}^*)] + \epsilon + \|\pi\|),$$

for some constants $C > 0$ and $0 < q < 1$, where $\alpha_{t_k}^{(n)} = \hat{\alpha}(t_k, X_{t_k}, Y_{t_k}, \tilde{\mu}_{t_k}^{(n-1)})$, and (X_t, Y_t) solves (A.9) *with μ replaced by $\tilde{\mu}_{t_k}^{(n-1)}$.*

The proofs of Theorems 3.4.2 and 3.4.3 are given in Appendix A.3 due to the page limit.

Remark that the Sig-DFP framework is flexible. We choose to solve (3.2)-(3.3) by direct parameterizing control policies α_t for the sake of easy implementation and the possible exploration of multiple mean-field equilibria. If the equilibrium is unique, with proper conditions on the coefficients b, σ, σ^0, f and g, one can reformulate (3.2)-(3.3) into McKean-Vlasov FBSDEs or stochastic FP/HJB equations, and solve them by fictitious play and propagating the common noise using signatures.

3.5 Experiments

In this section, we present the performance of Sig-DFP for three examples: homogeneous, heterogeneous, and heterogeneous extended MFGs. A relative L^2 metric will be used for performance measurement, defined for progressively measurable random processes as

$$L_R^2(x,\hat{x}) := \sqrt{\frac{\mathbb{E}[\int_0^T \|x_t - \hat{x}_t\|^2 \, \mathrm{d}t]}{\mathbb{E}[\int_0^T \|x_t\|^2 \, \mathrm{d}t]}}, \tag{3.12}$$

where x is a benchmark process and $\hat{x}$ is its prediction. We shall use stochastic gradient descent (SGD) optimizer for all three experiments. Training processes are done on a server with Intel Core i9-9820X (10 cores, 3.30 GHz) and RTX 2080 Ti GPU, and training time will be reported in Appendix A.2. Implementation codes are available at `https://github.com/mmin0/SigDFP`.

Data Preparation. For all three experiments, the size of both training and test data is $N = 2^{15}$, and the size of validation data is $N/2$. We fix $T = 1$ and discretize $[0,1]$ by $t_k = \frac{k}{100}$, $k = 0, 1, \ldots, 100$. Initial states are generated independently by $X_0^i \sim \mu_0$, with $\mu_0 = U(0,1)$ as the uniform distribution. The idiosyncratic Brownian motions W and common noises B are generated by antithetic variates for variance reduction, *i.e.*, we generate the first half samples (W^i, B^i) and get the other half $(-W^i, -B^i)$ by flipping.

Benchmarks. The examples below are carefully chosen with analytical benchmark solutions. Due to the space limit, we provide the details in Appendix A.4.

Linear-Quadratic MFGs. We first consider a Linear-Quadratic MFG with common noise proposed in [32], formulated as below:

$$\begin{aligned} \inf_{\alpha} \mathbb{E}\bigg\{ \int_0^T \left[\frac{\alpha_t^2}{2} - q\alpha_t(m_t - X_t) + \frac{\epsilon}{2}(m_t - X_t)^2 \right] \mathrm{d}t \\ + \frac{c}{2}(m_T - X_T)^2 \bigg\}, \end{aligned} \tag{3.13}$$

$$\begin{aligned} \text{where } \mathrm{d}X_t = [a(m_t - X_t) + \alpha_t] \, \mathrm{d}t \\ + \sigma(\rho \, \mathrm{d}B_t + \sqrt{1-\rho^2} \, \mathrm{d}W_t). \end{aligned} \tag{3.14}$$

Here $m_t = \mathbb{E}[X_t|\mathcal{F}_t^B]$ is the conditional population mean, $\rho \in [0,1]$ characterizes the noise correlation between agents, and $q, \epsilon, c, a, \sigma$ are positive constants. The agents have homogeneous preferences and aim to minimize their individual costs. We assume $q \leq \epsilon^2$ so that the Hamiltonian is jointly convex in state and control variables, ensuring a unique mean-field equilibrium.

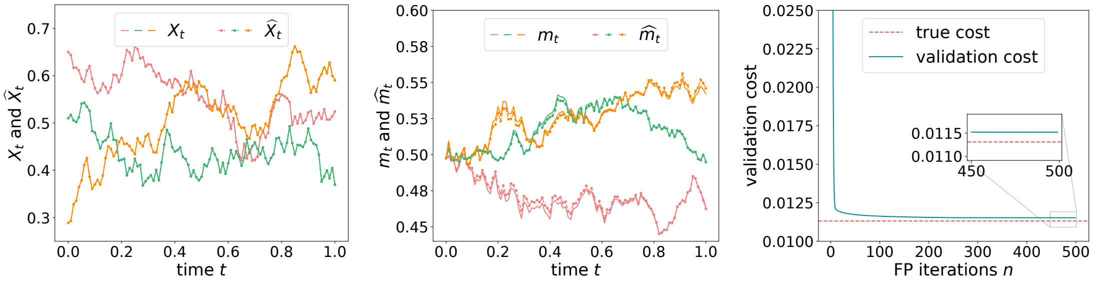
Training & Results. α_φ is a feedforward NN with two hidden layers of width 64. The truncated signature depth is chosen at $M = 2$. The model is trained for 500 iterations of fictitious play. The optimized state process $\hat{X}$ and its conditional mean $\hat{m}$ generated by test data are shown in Figures 3.2a and 3.2b. The minimized cost after each iteration computed using validation data is given in Figure 3.2c, where one can see a rapid convergence to the benchmark cost. During the experiments, we notice a slow convergence speed when using the average of $m^{(n)}$ in (3.14). This is because the initial guess $m^{(0)}$ is in general far from the truth. Therefore, for the first half of iterations, we simply use the previous-step result $m^{(n-1)}$. The learning rate is set as 0.1 for the first half and 0.01 for the second half of training. The relative L^2 errors for test data are listed in Table 3.1.

Table 3.1: Relative L^2 errors on test data for the LQ MFG.

	SDE X_t	Control α_t	Equilibrium m_t
L^2_R	0.0031	0.0044	0.058

Mean-Field Portfolio Game. Our second experiment is performed on a heterogeneous MFG proposed by [105], where the agent's preference is different, characterized by a type vector ζ which is random and drawn at time 0. They all aim to maximize their exponential utility of terminal wealth compared to the population average:

$$\sup_{\pi} \mathbb{E}\left[-\exp\left(-\frac{1}{\delta}(X_T - \theta m_T)\right)\right], \tag{3.15}$$



(a) X_t (b) $m_t = \mathbb{E}(X_t|\mathcal{F}_t^B)$ (c) Minimized Cost

Figure 3.2: Panels (a) and (b) give three trajectories of X_t, $m_t = \mathbb{E}[X_t|\mathcal{F}_b^B]$ (solid lines) and their approximations (dashed lines) using different (X_0, W, B) from test data. Panel (c) shows the minimized cost computed using validation data over fictitious play iterations. Parameter choices are: $\sigma = 0.2, q = 1, a = 1, \epsilon = 1.5, \rho = 0.2, c = 1$, $x_0 \sim U(0,1)$.

where the dynamics are

$$\mathrm{d}X_t = \pi_t(\mu\,\mathrm{d}t + \nu\,\mathrm{d}W_t + \sigma\,\mathrm{d}B_t), \quad X_0 = \xi. \tag{3.16}$$

Here m represents the conditional mean $m_t := \mathbb{E}[X_t|\mathcal{F}_t^B]$, and $\zeta = (\xi, \delta, \theta, \mu, \nu, \sigma)$ is random.

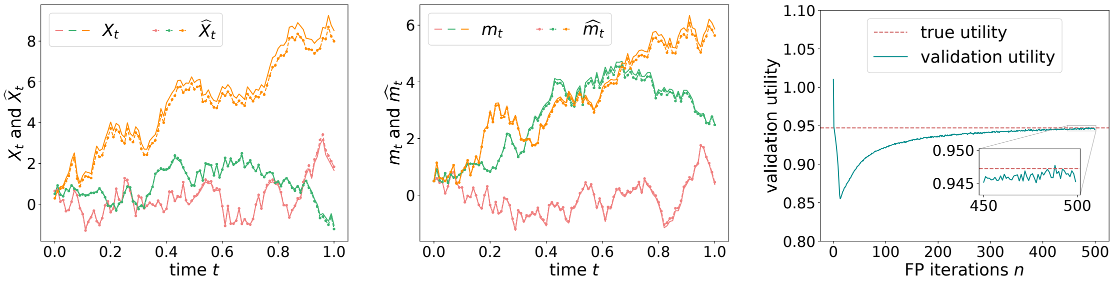
Training & Results. We use truncated signatures of depth $M = 2$ and a feedforward NN π_φ with 4 hidden layers[3] to approximate π. We train our model with 500 iterations of fictitious play. The learning rate starts at 0.1 and is reduced by a factor of 5 every 200 rounds. The relative L^2 errors evaluated under test data are listed in Table 3.2. Figure 3.3 compares X and m to their approximations, and plots the maximized utilities.

Table 3.2: Relative L^2 errors on test data for MF Portfolio Game.

	SDE X_t	Invest π_t	Equilibrium m_t
L^2_R	0.068	0.035	0.085

Mean-Field Game of Optimal Consumption and Investment. Our last ex-

[3] Since agents are heterogeneous characterized by their type vectors ζ, π_φ takes (ζ, t, X_t, m_t) as inputs. Hidden neurons in each layer are (64, 32, 32, 16).



(a) X_t (b) $m_t = \mathbb{E}(X_t|\mathcal{F}_t^B)$ (c) Maximized Utility

Figure 3.3: Panels (a) and (b) give three trajectories of X_t, $m_t = \mathbb{E}[X_t|\mathcal{F}_b^B]$ (solid lines) and their approximations (dashed lines) using different (X_0, W, B) from test data. Panel (c) shows the maximized utility computed using validation data over fictitious play iterations. Parameter choices are: $\delta \sim U(5, 5.5), \mu \sim U(0.25, 0.35), \nu \sim U(0.2, 0.4), \theta \sim U(0, 1), \sigma \sim U(0.2, 0.4)$, $\xi \sim U(0, 1)$.

periment considers an extended heterogeneous MFG proposed by [102], where agents interact via both states and controls. The setup is similar to [105] except for including consumption and using power utilities. More precisely, each agent is characterized by a type vector $\zeta = (\xi, \delta, \theta, \mu, \nu, \sigma, \epsilon)$, and the optimization problem reads

$$\sup_{\pi, c} \mathbb{E}\left[\int_0^T U(c_t X_t (\Gamma_t m_t)^{-\theta}; \delta)\, \mathrm{d}t + \epsilon U(X_T m_T^{-\theta}; \delta)\right], \tag{3.17}$$

where $U(x; \delta) = \frac{1}{1-\frac{1}{\delta}} x^{1-\frac{1}{\delta}}$, $\delta \neq 1$, X_t follows

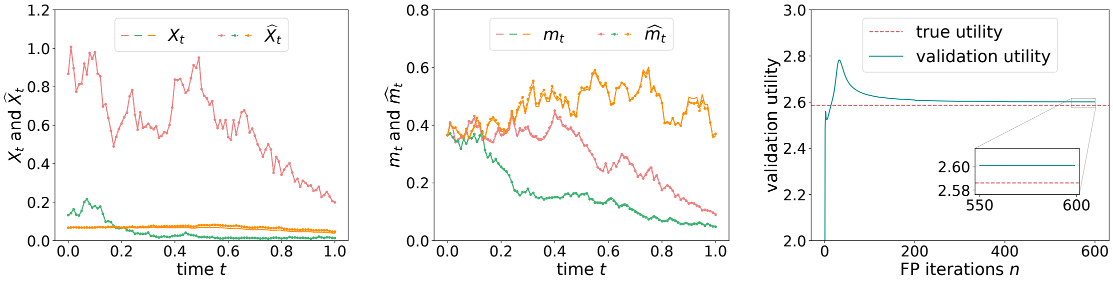
$$\mathrm{d}X_t = \pi_t X_t(\mu\, \mathrm{d}t + \nu\, \mathrm{d}W_t + \sigma\, \mathrm{d}B_t) - c_t X_t\, \mathrm{d}t, \tag{3.18}$$

and $X_0 = \xi$. Here $\Gamma_t = \exp \mathbb{E}[\log c_t|\mathcal{F}_t^B]$ and $m_t = \exp \mathbb{E}[\log X_t|\mathcal{F}_t^B]$ are the mean-field interactions from consumption and wealth.

Training & Results. For this experiment, we use truncated signatures of depth $M = 4$. The optimal controls $(\pi_t, c_t)_{0 \le t \le 1}$ are parameterized by two neural networks π_φ and c_φ, each with three hidden layers.[4] Due to the extended mean-field interaction term Γ_t, we will propagate two conditional distribution flows, *i.e.*, two linear functionals $\bar{l}^{(n)}, \bar{l}_c^{(n)}$ during each iteration of fictitious play. Instead of estimating m_t, Γ_t directly, we estimate $\mathbb{E}[\log X_t|\mathcal{F}_t^B], \mathbb{E}[\log c_t|\mathcal{F}_t^B]$ by $\langle \bar{l}^{(n)}, S^4(B_{0:t})\rangle$, $\langle \bar{l}_c^{(n)}, S^4(B_{0:t})\rangle$ and then take exponential to get m_t, Γ_t. To ensure the non-negativity condition of X_t, we evolve $\log X_t$ according to (A.22) and then take exponential to get X_t. For optimal consumption, c_φ is used to predicted $\log c_t$ and thus $\exp c_\varphi$ gives the predicted c_t. With 600 iterations of fictitious play and a learning rate of 0.1 decaying by a factor of 5 for every 200 iterations, the relative L^2 errors for test data are listed in Table 3.3. Figure 3.4 compares X and m to their approximations, and plots the maximized utilities. Plots of π_t, c_t, $\Gamma_t = \exp \mathbb{E}(\log c_t|\mathcal{F}_t^B)$ are provided in Appendix A.5.

Table 3.3: Relative L^2 errors on test data for Optimal Consumption and Investment MFG.

	Invest π_t	Consumption c_t	m_t	Γ_t
L_R^2	0.1126	0.0614	0.0279	0.0121



(a) X_t (b) $m_t = \exp \mathbb{E}(\log X_t|\mathcal{F}_t^B)$ (c) Maximized Utility

Figure 3.4: Panels (a) and (b) give three trajectories of X_t and $m_t = \exp \mathbb{E}(\log X_t|\mathcal{F}_t^B)$ (solid lines) and their approximation (dashed lines) using different (X_0, W, B) from test data. Panel (c) shows the maximized utility computed using validation data over fictitious play iterations. Parameter choices are: $\delta \sim U(2, 2.5), \mu \sim U(0.25, 0.35), \nu \sim U(0.2, 0.4), \theta, \xi \sim U(0, 1), \sigma \sim U(0.2, 0.4)$, $\epsilon \sim U(0.5, 1)$.

[4] Due to the nature of heterogeneous extended MFG, both α_φ and c_φ take $(\zeta_t, t, X_t, m_t, \Gamma_t)$ as inputs. Hidden neurons in each layer are (64, 64, 64).

Comparison with the nested algorithm. We run both Sig-DFP and the nested algorithm for the training data size of (INP, CNP)= $(2^4, 2^4)$, $(2^6, 2^6)$ $(2^8, 2^8)$, where INP means the number of individual noise paths and CNP means the number of common noise paths. From the comparisons of running time, memory, and relative L^2 errors in Tables 3.4 and 3.5, one can see that the accuracy is mainly affected by the size of (INP, CNP) used for training the neural network. Sig-DFP has the advantage of reducing memory request and running time, which allows it to use a larger size of data, *e.g.*, (INP, CNP)= $(2^{15}, 2^{15})$, to produce much better accuracy. The quadratic growth of memory in the nested algorithm, evidenced by the first three columns of data in Tables 3.4 (least squares growth rate ≈ 2), makes us unable to run the nested algorithm beyond $(2^8, 2^8)$ in our current computing environment due to its high demand for memory.

Table 3.4: Running time (hours) and Memory (GBs) comparisons between Sig-DFP and the nested algorithm for different (INP, CNP)'s. INP = # of individual noise paths, CNP = # of common noise paths, and NA = Not Available due to high demand for memory.

(INP, CNP)	$(2^4, 2^4)$	$(2^6, 2^6)$	$(2^8, 2^8)$	$(2^{12}, 2^{12})$	$(2^{15}, 2^{15})$
Nested Algorithm	$(0.09, 2.1)$	$(0.46, 4.1)$	$(4.3, 43.5)$	NA	NA
Sig-DFP	$(0.09, 1.9)$	$(0.1, 2.0)$	$(0.17, 2.3)$	$(0.33, 4.8)$	$(1.3, 27)$

Table 3.5: The comparisons of relative L^2 errors on (π, c) between Sig-DFP and the nested algorithm for different (INP, CNP)'s. INP = # of individual noise paths, CNP = # of common noise paths, and NA = Not Available due to high demand for memory.

(INP, CNP)	$(2^4, 2^4)$	$(2^6, 2^6)$	$(2^8, 2^8)$	$(2^{12}, 2^{12})$	$(2^{15}, 2^{15})$
Nested Algorithm	$(53\%, 44\%)$	$(36\%, 41\%)$	$(79.4\%, 16.2\%)$	NA	NA
Sig-DFP	$(85.8\%, 48.1\%)$	$(43.3\%, 44.9\%)$	$(49\%, 43\%)$	$(18\%, 38\%)$	$(11\%, 6\%)$

Table 3.6: The comparisons of running time (hours) for different signature depth M and dimension n_0 using (INP, CNP)= $(2^{15}, 2^{15})$.

(n_0, Depth M)	$(1,1)$	$(1,2)$	$(1,3)$	$(1,4)$	$(5,1)$	$(5,2)$	$(5,3)$	$(5,4)$
Running Time (hours)	1.2	1.2	1.2	1.3	1.2	1.3	1.5	2.6

Comparisons of running time for different signature depth M and dimension n_0. We choose the data size (INP, CNP)= $(2^{15}, 2^{15})$ and compare the running time for different (n_0, M)'s in Table 3.6. Choosing $M = 1, 2, 3, 4$ yield the relative L^2 errors of controls (π, c) as $(15.9\%, 9.5\%)$, $(11.4\%, 6.3\%)$, $(11.4\%, 6.3\%)$ and $(11.3\%, 6.1\%)$ for $n_0 = 1$, respectively. Note that, compared to $M = 1$, taking $M = 2$ improves the accuracy significantly but not $M = 3, 4$. This is because the curves of $\log(c_t)$ and $\log(X_t^*)$ are approximately either linear or quadratic in t, as shown in Figure A.1 in Appendix A.5 after taking a logarithm, which implies that the signatures of depth $M = 2$ will be sufficient to produce good accuracy. We remark that Sig-DFP has no difficulty computing high-dimensional problems, evidenced by the running time of $n_0 = 5$ cases in Table 3.6. We focus on one-dimensional problems since, to our best knowledge, the closed-form non-trivial solutions only exist in one-dimensional cases, which can serve as the benchmark solutions. More details about $n_0 = 5$ are given in Appendix A.6.

Chapter 4

Directed Chain SDEs

4.1 Background

4.1.1 Directed Chain SDE and Smootheness

The main objective of this chapter is to study the existence and regularity of the densities of the directed chain stochastic differential equations (DC-SDEs), and propose a deep learning based time series generator based on the idea of directed chain SDEs. Given a filtered probability space $(\Omega, \mathcal{F}, (\mathcal{F}_t)_{t\geq 0}, \mathbb{P})$, the directed chain McKean-Vlasov stochastic differential equation (or directed chain SDE for short) for a pair $(X^\theta_\cdot, \widetilde{X}_\cdot)$ of N-dimensional stochastic processes considered here is of the form

$$X^\theta_t = \theta + \int_0^t V_0(s, X^\theta_s, \mathrm{Law}(X^\theta_s), \widetilde{X}_s)\,\mathrm{d}s + \sum_{i=1}^d \int_0^t V_i(s, X^\theta_s, \mathrm{Law}(X^\theta_s), \widetilde{X}_s)\,\mathrm{d}B^i_s, \qquad (4.1)$$

for $t \geq 0$ with the distributional constraint

$$[X^\theta_t, t \geq 0] := \mathrm{Law}(X^\theta_t, t \geq 0) = \mathrm{Law}(\widetilde{X}_t, t \geq 0) =: [\widetilde{X}_t, t \geq 0],$$

where V_i, $i = 0, 1, \dots, d$ are some smooth coefficients, $B_\cdot := (B_\cdot^1, \dots, B_\cdot^d)$ is a standard d-dimensional Brownian motion independent of the initial state $X_0^\theta = \theta$ and of $\widetilde{X}_\cdot$, and X_0^θ is independent of $\widetilde{X}_0$. Throughout the section, $[\xi]$ denotes the law of a generic random element ξ. Here each coefficient V_i in (4.1) depends on time s, the value X_s^θ, its law $\mathrm{Law}(X_s^\theta) =: [X_s^\theta]$ and the other $\widetilde{X}_s$ of the pair for $s \ge 0$. The law $[X_\cdot^\theta]$ depends on the law $[\widetilde{X}_\cdot]$ through (4.1) and they are the same marginal law. We show that the above directed chain SDE has a unique weak solution in section 4.2.

This kind of directed chain structure was firstly proposed by [52] in a simpler form. Schematically, it can be written as an infinite chain of stochastic equations for $(X_{1,\cdot}, X_{2,\cdot}, \dots)$:

$$\begin{aligned}
\mathrm{d}X_{1,t} &= b(t, X_{1,t}, F_{1,t})\,\mathrm{d}t + \mathrm{d}B_{1,t}, \\
\mathrm{d}X_{2,t} &= b(t, X_{2,t}, F_{2,t})\,\mathrm{d}t + \mathrm{d}B_{2,t} \\
&\vdots \\
\mathrm{d}X_{i,t} &= b(t, X_{i,t}, F_{i,t})\,\mathrm{d}t + \mathrm{d}B_{i,t}, \\
&\vdots
\end{aligned} \tag{4.2}$$

where $F_{i,t} := u\delta_{X_{i+1,t}} + (1-u)\mu_{i,t}$ is the mixture distribution term of the measure-dependent drift coefficient b with the marginal law $\mu_{i,t} := \mathrm{Law}(X_{i,t})$ of $X_{i,t}$ for $t \ge 0$, $\delta_{X_{i+1,t}}$ is the Dirac measure at $X_{i+1,t}$, a fixed constant $u \in [0,1]$ measures the common amount of dependency of $X_{i,\cdot}$ on its neighborhood value $X_{i+1,\cdot}$, and $B_{1,\cdot}, B_{2,\cdot,\dots}$ are independent standard Brownian motions. We assume also that the initial value $X_{i,0}$ is independent of $B_{i,\cdot}$, and $X_{i+1,\cdot}$ and $B_{i,\cdot}$ are independent for $i = 1, 2, \dots$. In particular, the drift b in [52] has the following form $b(t, x, \mu) := \int_{\mathbb{R}} \widetilde{b}(t, x, y)\mu(\mathrm{d}y)$ with some Lipschitz continuous function $\widetilde{b}$. See also Figure 4.2 in section 4.4.

The stochastic processes on infinite graphs including the directed chain structure have drawn many attentions recently. Stochastic Differential Games on the directed chain have been studied in [59] and on the extended version of random directed networks in [61] as well as on the general random graph (e.g., [101]). [100] discuss the Markov random field property over both finite and countably infinite graph with local interactions through the drift coefficients. Another related topic is the Graphon particle system. There are a sequence of works in Graphon Mean Field Games, [10, 21, 22] just to name a few. [11] introduced the uniform-in-time exponential concentration bounds related to the graphon particle system and its finite particle approximations. Here, we are interested in the existence and smoothness of the density of directed chain SDE (4.3)-(4.4). It should be emphasized here that in this problem, we need notions of derivatives in the space of measures, which is used frequently in the theory of Mean Field Games.

In most cases, Malliavin calculus is a foundation to analyze the smoothness of the density of stochastic differential equations. It has been widely used in investigating the density of diffusions [95], [97], [98] and then applied into many different scenarios. The authors in [36] use Malliavin calculus to derive smoothing properties of solution to stochastic differential equations with jumps. The smoothing properties of McKean-Vlasov SDEs have been studied in [44], which is closely related to our purpose. However, because of the appearance of the auxiliary process $\widetilde{X}$, the crucial step making connections between the Malliavin derivative and the first order derivative of the state process fails, please see Question 4.3.3 for the detail. To our best knowledge, we did not find any works studying the smoothness property of such weak solutions of stochastic differential equations.

For the purpose of resolving this problem and utilizing the Malliavin derivatives, we should frozen the auxiliary process $\widetilde{X}$. This inspired us to consider another closely related, well-developed tool, partial Malliavin calculus. Partial Malliavin calculus is first introduced by [93] for the constant case, where the projections are taken on a fixed Hilbert subspace, and applied to prove some regularity results in Non-linear Filtering theory.

Another work developing the partial Malliavin calculus is [84], by which the authors were able to complete the proof of some results in [117] on the long-time asymptotic of the stochastic oscillatory integrals. We mainly adopt the framework from a later work by Nualart and Zakai [129], where the projection is taken on a family of the subspace which is defined as the orthogonal complement to the subspaces generated by $\widetilde{X}$ in (4.1). We remark that our method is potentially applied to analyze the smoothness property of weak solutions of stochastic differential equations in a general setting.

4.1.2 Directed Chain Generative Adversarial Networks

Generative models are important to overcome the limitation of data scarcity, privacy, and costs. In particular, medical data are not easy to get, use or share, due to privacy; and financial time series data are inadequate due to their nonstationarity nature. Times-series generative models, instead of seeking to learn the governing equations from real data, aim to discover and learn data automatically, and output new data that plausibly can be drawn from the original dataset. Some existing infinite-dimensional generative adversarial networks (GANs) (e.g., [87, 109]) showed successful performance in unimodal time series datasets. However, many real-world phenomena are multimodal distributed, e.g., data describing the opinion divergence in a community [144], the interspike interval distribution [138], and the oscillators' natural frequencies [139]. All these bring the necessity of developing new generative models for multimodal time series data.

In this project, we develop a novel time-series generator, named *directed chain GANs* (DC-GANs), motivated by the formulation of DC-SDEs introduced above and in [53]. The drift and diffusion coefficients in DC-SDEs depend on another stochastic process, which we call the neighborhood process, with distribution required to be the same as the SDEs' distribution. Different from other GANs, which only use real data in discriminators, our proposed algorithm naturally takes the dataset as the neighborhood process, giving generators access to data information. This feature enables our model to outperform the state-of-the-art methods on many datasets, particularly for the situation of multimodal time-series data.

Contribution. We propose a generator for multimodal distributed time series based on DC-SDEs (cf. Definition 4.5.1), and prove that our model can handle any distribution that Neural SDEs are capable of generating (see Theorem 4.5.2). To train the generator, we propose to use a combination of two types of discriminators: Sig-WGAN [125] and Neural CDEs [90].

We notice that data generated immediately from DC-GANs can be correlated, and propose an easy solution by walking along the directed chain in the path space for further steps (see Theorem 4.5.4). Combining branching the chain with different Brownian noises enables our model to generate unlimited independent fake data.

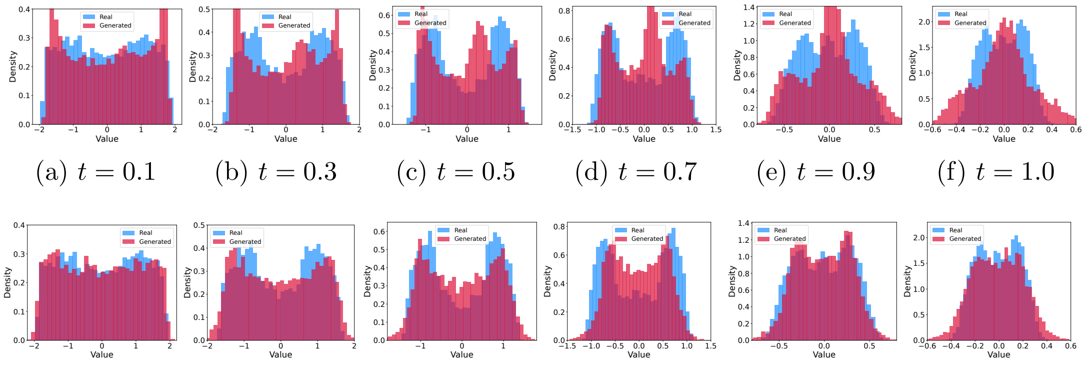
We test our algorithms in four different experiments and show that DC-GANs provide the best performance compared to existing popular models, including SigWGAN [125], CTFP [50], Neural SDEs [87], and TimeGAN [149].

Related Literature. Neural ordinary differential equations (Neural ODEs), introduced by [38], use neural networks to parameterize the vector fields of ODEs and bring a powerful tool for learning time series data. Later, significant effort has been put into improving Neural ODEs, e.g., [133, 150, 118, 77]. In fact, incorporating mathematical concepts into the Neural ODEs framework can provide the capability of analyzing and justifying its validity, leading to a deeper understanding of the framework itself. For example, [110] and [145] generalized the idea to neural stochastic differential equations (Neural SDEs), providing adjoint equations for efficient training. By integrating rough path theory [115], [90] proposed neural controlled differential equations (Neural CDEs) and [123] proposed neural rough differential equations for modeling time series. Other examples integrating profound mathematical concepts include using higher order kernel mean embeddings to capture information filtration [137], and solving high dimensional partial differential equations through backward stochastic differential equations [76], to name a few.

The closely related model to ours is the Neural SDEs by [87], which uses the Wasserstein GAN method to train stochastic diffusion evolving in a hidden space and gains great success in simulating time series data. Other successful GANs models for time-series data include [47, 146, 50, 87, 109]; see [16] for a recent review. Note that we find in the numerical experiments that the performances of Neural SDEs are limited in simulating multimodal distributed time series, e.g., as shown in Figure 4.1 from the stochastic opinion dynamics (Example 1 in Section 4.5.2).

To our best knowledge, [53] initiated the study of the SDE system on the directed chains, followed by [62, 60] for the analysis of stochastic differential games on such chains with (deterministic and random) interactions. Later on, more complicated graph structures are studied beyond directed chains. For example, [100] analyzed particle behaviors where the interaction only happens between neighborhoods in an undirected graph, and proved Markov random fields property and constructed Gibbs measure on path space when interactions appear only in drift; [103] considered stochastic differential games on transitive graphs; [28] studied games on a graphon which has infinitely many nodes. Despite numerous extensions, we find that the directed chain structure, although simple but rich enough for generating multimodal time series.

From another viewpoint, DC-SDEs can be understood as the reverse direction of mimicking theorems [70]. The idea of “mimicking” is that for a general SDE (even with path-dependence features), one can construct a Markovian one to mimic its marginal distribution; see [19] for details on mimicking aspects of Itô processes including the distributions of running maxima and running integrals. DC-SDEs work in the reverse direction: they can produce marginal distributions that are generated by Markovian SDEs (see Theorem 4.5.2 for a detailed statement). The benefit of using DC-SDEs, in particular in machine learning, is to have a more vital fitting ability by embedding data into a slightly more complicated system.



(a) $t = 0.1$ (b) $t = 0.3$ (c) $t = 0.5$ (d) $t = 0.7$ (e) $t = 0.9$ (f) $t = 1.0$

(g) $t = 0.1$ (h) $t = 0.3$ (i) $t = 0.5$ (j) $t = 0.7$ (k) $t = 0.9$ (l) $t = 1.0$

Figure 4.1: Marginal distributions of real data (blue) and generated data (red) from Example 1 (Stochastic opinion dynamics) at $t \in \{0.1, 0.3, 0.5, 0.7, 0.9, 1\}$ in Section 4.5.2. Figures (a)–(f) are generated by Neural SDEs, and Figures (g)–(l) are generated by DC-GANs. One can see from Figures (e) and (f) that Neural SDEs fail to capture the bimodal distribution.

This chapter is structured as follows: In section 4.2, we first introduce the differentiation in the space of measures and multi-index notation in section 4.2.1, and then prove the existence, uniqueness and some regularity results on the solutions of generalized directed chain SDEs in Propositions 4.2.2-4.2.3. In section 4.3, we prepare the notions of the partial Malliavin calculus and give the Kusuoka-Stroock process for the proof of our smoothness of densities, which will be stated in section 4.4. Our proofs follows the idea in [44], where we first derive integral by parts formulae for the directed chain SDEs via the partial Malliavin derivatives, instead of the Malliavin derivatives, as in [44]. The main result is stated in Theorem 4.4.11 with some applications in section 4.4. Finally, we introduce our DC-GANs algorithm in detail and experiments results in Section 4.5.

4.2 Preliminaries and Directed Chain SDEs

In this section, we first prepare some notations and the notion of differentiation in $\mathcal{P}_2$, where $\mathcal{P}_2$ is the space of all measures with finite second moments, and then establish the weak solutions of directed chain SDEs.

4.2.1 Notations and Basic Setup

To be consistent with the reference [44], we use $[\xi]$ to denote the law of a random variable ξ. Rather than the directed chain SDE of the type given in [52], we consider the SDE in a more general setup, allowing the diffusion coefficients non-constant. Given a probability space $(\Omega, \mathcal{F}, \mathbb{F} = (\mathcal{F}_t)_{t\geq 0}, \mathbb{P})$, the directed chain McKean-Vlasov SDE (or directed chain SDE for short) is of the form

$$X_t^\theta = \theta + \int_0^t V_0(s, X_s^\theta, [X_s^\theta], \widetilde{X}_s)\,\mathrm{d}s + \sum_{i=1}^d \int_0^t V_i(s, X_s^\theta, [X_s^\theta], \widetilde{X}_s)\,\mathrm{d}B_s^i, \tag{4.3}$$

$$\text{with the constraint} \quad [X_t^\theta, t \geq 0] = [\widetilde{X}_t, t \geq 0], \tag{4.4}$$

where $B_s := (B_s^1, \ldots, B_s^d)$ is a standard d dimensional Brownian motion and $\widetilde{X}_s \in L^2(\Omega \times [0,T], \mathbb{R}^N)$ is an adapted random process independent of all the Brownian motions $B^i, i = 1, \ldots, d$ and initial state $X_0^\theta \equiv \theta$.

In particular, the abstract directed chain system (4.3)-(4.4) takes initial pair $(X_0^\theta, \widetilde{X}_0)$ as input, and produces output $(X_\cdot^\theta, \widetilde{X}_\cdot)$ as the solution on a probability space $(\Omega, \mathcal{F}, \mathbb{P})$. We emphasize the Brownian motion B is independent of $(X_0^\theta, \widetilde{X}_\cdot)$ and the direction of directed chain system is determined by this independence property.

Moreover, we assume that $V_0, V_i : [0,T] \times \mathbb{R}^N \times \mathcal{P}_2(\mathbb{R}^N) \times \mathbb{R}^N \to \mathbb{R}^N$, where $\mathcal{P}_2(\mathbb{R}^N)$ is the set of measures on $\mathbb{R}^N$ with finite second moments for $i = 1, 2, \ldots, d$. We equip $\mathcal{P}_2(\mathbb{R}^N)$ with the 2-Wasserstein metric, W_2. For a general metric space (M, d), we define the 2-Wasserstein metric on $\mathcal{P}_2(M)$ by

$$W_2(\mu, \nu) = \inf_{\Pi \in \mathcal{P}_{\mu,\nu}} \left(\int_{M \times M} d(x,y)^2 \Pi(\,\mathrm{d}x,\,\mathrm{d}y) \right)^{1/2},$$

where $\mathcal{P}_{\mu,\nu}$ denotes the class of measures on $M \times M$ with marginals μ and ν.

We denote L^p norm on $(\Omega, \mathcal{F}, \mathbb{P})$ by $\|\cdot\|_p$, $p \geq 1$ and for every $t \geq 0$, we also introduce the space $\mathcal{S}_t^p$ of continuous $\mathbb{F}$ adapted process φ on $[0,t]$, satisfying

$$\|\varphi\|_{\mathcal{S}_t^p} = \big(\mathbb{E} \sup_{s \in [0,t]} |\varphi_s|^p\big)^{1/p} < \infty.$$

Let us introduce more notations in accordance with [44]. We will write $\theta = \delta_x$ if the initial state of this SDE is a fixed real vector $x \in \mathbb{R}^N$. We use $\mathcal{C}_{b,\mathrm{Lip}}^{k,k,k}(\mathbb{R}^+ \times \mathbb{R}^N \times \mathcal{P}_2(\mathbb{R}^N) \times \mathbb{R}^N; \mathbb{R}^N)$ for the class of functions that are k-times continuously differentiable with bounded Lipschitz derivatives in the the last three variables, where the notion of derivatives with respect to measure is adopted from P.-L. Lions' lecture notes at the *Collège de France*, recorded in a set of notes [24], very well exposed in [35] and also adopted by [44]. A precise definition for $\mathcal{C}_{b,\mathrm{Lip}}^{k,k,k}(\mathbb{R}^+ \times \mathbb{R}^N \times \mathcal{P}_2(\mathbb{R}^N) \times \mathbb{R}^N; \mathbb{R}^N)$ will be given in Definition 4.2.1.

Differentiability in $\mathcal{P}_2(\mathbb{R}^N)$. Lion's notion of differentiability with respect to measure of functions $U : \mathcal{P}_2(\mathbb{R}^N) \to \mathbb{R}$ is to define a lifted function U' on the Hilbert space $L^2(\Omega'; \mathbb{R}^N)$ over probability space $(\Omega', \mathcal{F}', \mathbb{P}')$, where Ω' is a Polish space and $\mathbb{P}'$ is an atomless measure, such that $U'(X') = U([X'])$ for $X' \in L^2(\Omega'; \mathbb{R}^N)$ and $[X'] = [X]$. Thus, we are able to express the derivative of U w.r.t. measure $\mu = [X]$ term as the

Fréchet derivative of U' w.r.t. X' whenever it exists, which can be written as an element of $L^2(\Omega';\mathbb{R}^N)$ by identifying $L^2(\Omega';\mathbb{R}^N)$ and its dual. This gradient in a direction $\gamma' \in L^2(\Omega';\mathbb{R}^N)$ is given by

$$\mathcal{D}U'(X')(\gamma') = \langle \mathcal{D}U'(X'), \gamma' \rangle = \mathbb{E}'\big[\mathcal{D}U'(X') \cdot \gamma'\big],$$

where $\mathbb{E}'$ is the expectation under $\mathbb{P}'$. By [24, Theorem 6.2], the distribution of this gradient depends only on the measure μ, exists uniquely and can be written as

$$\partial_\mu U(\mu, X') := \mathcal{D}U'(X') = \xi(X') \in L^2(\Omega';\mathbb{R}^N).$$

This definition of the derivative with respect to measure can be extended to higher orders by thinking of $\partial_\mu U(\mu,\cdot) : \mathcal{P}_2(\mathbb{R}^N) \times \mathbb{R}^N \to \mathbb{R}^N$ as a function, and the derivative is well defined for each of its components as in the following. For each $\mu \in \mathcal{P}_2(\mathbb{R}^N)$, there exists a unique version of such function $\partial_\mu U(\mu,\cdot)$ which is assumed to be a priori continuous (see the discussion in [44]).

Multi-index. To get a more general result, we extend the derivatives to higher order. For a function $f : \mathcal{P}_2(\mathbb{R}^N) \to \mathbb{R}^N$, we can apply the above discussion straightforwardly to each component $f = (f^1, \dots, f^N)$. Then the derivatives $\partial_\mu f^i, 1 \le i \le N$ takes values in $\mathbb{R}^N$, and we denote $(\partial_\mu f^i)_j : \mathcal{P}_2(\mathbb{R}^N) \times \mathbb{R}^N \to \mathbb{R}$ for $j = 1, \dots, N$. For a fixed $v \in \mathbb{R}^N$, we are able to differentiate $\mathcal{P}_2 \ni \mu \mapsto (\partial_\mu f^i)_j(\mu, v) \in \mathbb{R}$ again to get the second order derivative. If the derivative of this mapping exists and there is a continuous version of

$$\mathcal{P}_2(\mathbb{R}^N) \times \mathbb{R}^N \times \mathbb{R}^N \ni (\mu, v_1, v_2) \mapsto \partial_\mu (\partial_\mu f^i)_j(\mu, v_1, v_2) \in \mathbb{R}^N,$$

then it is unique. It is natural to have a multi-index notation $\partial_\mu^{(j,k)} f^i := (\partial_\mu(\partial_\mu f^i)_j)_k$ to ease the notation. Similarly, for higher derivatives, if for each $(i_0, \dots, i_n) \in \{1, \dots, N\}^{n+1}$,

$$\underbrace{\partial_\mu(\partial_\mu \dots (\partial_\mu}_{n \text{ times}} f^{i_0})_{i_1} \dots)_{i_n}$$

exists, we denote this $\partial_\mu^\alpha f^{i_0}$ with $\alpha = (i_1, \dots, i_n)$ and $|\alpha| = n$. Each derivative in μ is a function of an extra variable with $\partial_\mu^\alpha f^{i_0} : \mathcal{P}_2(\mathbb{R}^N) \times (\mathbb{R}^N)^n \to \mathbb{R}$. We always denote these variables, by $v_1, \dots, v_n$, i.e.,

$$\mathcal{P}_2(\mathbb{R}^N) \times (\mathbb{R}^N)^n \ni (\mu, v_1, \dots, v_n) \mapsto \partial_\mu^\alpha f^{i_0}(\mu, v_1, \dots, v_n) \in \mathbb{R}.$$

When there is no confusion, we will abbreviate $(v_1, \dots, v_n)$ to $\boldsymbol{v} \in (\mathbb{R}^N)^n$, so that

$$\partial_\mu^\alpha f^{i_0}(\mu, \boldsymbol{v}) = \partial_\mu^\alpha f^{i_0}(\mu, v_1, \dots, v_n),$$

and use notation

$$|\boldsymbol{v}| := |v_1| + \dots + |v_n|,$$

with $|\cdot|$ the Euclidean norm on $\mathbb{R}^N$. It then makes sense to discuss derivatives of the function $\partial_\mu^\alpha f^{i_0}$ with respect to variables $v_1, \dots, v_n$.

If, for some $j \in \{1, \dots, N\}$ and all $(\mu, v_1, \dots, v_{j-1}, v_{j+1}, \dots, v_n) \in \mathcal{P}_2(\mathbb{R}^N) \times (\mathbb{R}^N)^{n-1}$,

$$\mathbb{R}^N \ni v_j \mapsto \partial_\mu^\alpha f^{i_0}(\mu, v_1, \dots, v_n) \in \mathbb{R}$$

is l-times continuously differentiable, we denote the derivatives $\partial_{v_j}^{\beta_j} \partial_\mu^\alpha f^{i_0}$, for β_j a multi-index on $\{1, \dots, N\}$ with $|\beta_j| \leq l$. Similar to the above, we will denote by $\boldsymbol{\beta}$ the n-tuple of multi-indices $(\beta_1, \ldots, \beta_n)$. We also associate a length to $\boldsymbol{\beta}$ by

$$|\boldsymbol{\beta}| := |\beta_1| + \cdots + |\beta_n|,$$

and denote $\#\boldsymbol{\beta} := n$. Then we denote by $\mathcal{B}_n$ the collection of all such $\boldsymbol{\beta}$ with $\#\boldsymbol{\beta} = n$, and $\mathcal{B} := \cup_{n \geq 1} \mathcal{B}_n$. Again, to lighten the notation, we use

$$\partial_{\boldsymbol{v}}^{\boldsymbol{\beta}} \partial_\mu^\alpha f^i(\mu, \boldsymbol{v}) := \partial_{v_n}^{\beta_n} \cdots \partial_{v_1}^{\beta_1} \partial_\mu^\alpha f^i(\mu, v_1, \ldots, v_n).$$

The coefficients $V_0, \ldots, V_d : [0, T] \times \mathbb{R}^N \times \mathcal{P}_2(\mathbb{R}^N) \times \mathbb{R}^N \to \mathbb{R}^N$ depend on a time variable, two Euclidean variables as well as the measure variable. So whether the order of taking derivatives matters is a question. Fortunately, a result from [20, Lemma 4.1] tells us that derivatives commute when the mixed derivatives are Lipschitz continuous. However, it should be emphasized that we could not interchange the order of ∂_μ and ∂_v, since the coefficients would not depend on the extra variable $\boldsymbol{v}$ before taking derivatives with respect to measure.

Definition 4.2.1 **($\mathcal{C}_{b,\mathrm{Lip}}^{k,k,k}$)** *We have the following definitions:*

(a) We use $\partial_x, \tilde{\partial}$ to denote the derivative with respect to the second and fourth Euclidean variables in V_0, V_i's, respectively.

(b) Let $V : \mathbb{R}^+ \times \mathbb{R}^N \times \mathcal{P}_2(\mathbb{R}^N) \times \mathbb{R}^N \to \mathbb{R}^N$ with components $V^1, \ldots, V^N : \mathbb{R}^+ \times \mathbb{R}^N \times \mathcal{P}_2(\mathbb{R}^N) \times \mathbb{R}^N \to \mathbb{R}$. We say $V \in \mathcal{C}_{b,\mathrm{Lip}}^{1,1,1}([0, T] \times \mathbb{R}^N \times \mathcal{P}_2(\mathbb{R}^N) \times \mathbb{R}^N; \mathbb{R}^N)$ if the following is true: for each $i = 1, \ldots, N$, $\partial_\mu V^i$, $\partial_x V^i$ and $\tilde{\partial} V^i$ exist. Moreover, assume the boundedness of the derivatives for all $(t, x, \mu, y, v) \in [0, T] \times \mathbb{R}^N \times \mathcal{P}_2(\mathbb{R}^N) \times \mathbb{R}^N \times \mathbb{R}^N$,

$$|\partial_x V^i(t, x, \mu, y)| + |\tilde{\partial} V^i(t, x, \mu, y)| + |\partial_\mu V^i(t, x, \mu, y, v)| \leq C.$$

In addition, suppose that $\partial_\mu V^i$, $\partial_x V^i$ and $\tilde{\partial} V^i$ are all Lipschitz in the sense that for all $(t, x, \mu, y, v), (t, x', \mu', y', v') \in [0, T] \times \mathbb{R}^N \times \mathcal{P}_2(\mathbb{R}^N) \times \mathbb{R}^N \times \mathbb{R}^N$,

$$\begin{aligned} \big|\partial_\mu V^i(t, x, \mu, y, v) - \partial_\mu V^i(t, x', \mu', y', v')\big| \le& \\ & C(|x - x'| + |y - y'| + |v - v'| + W_2(\mu, \mu')), \\ \big|\partial_x V^i(t, x, \mu, y) - \partial_x V^i(t, x', \mu', y')\big| \le& \, C(|x - x'| + |y - y'| + W_2(\mu, \mu')), \\ \big|\tilde{\partial} V^i(t, x, \mu, y) - \tilde{\partial} V^i(t, x', \mu', y')\big| \le& \, C(|x - x'| + |y - y'| + W_2(\mu, \mu')), \end{aligned}$$

and V^i, $\partial_\mu V^i$, $\partial_x V^i$ and $\tilde{\partial} V^i$ all have linear growth property,

$$\begin{aligned} &|V^i(t, x, \mu, y)| + |\partial_x V^i(t, x, \mu, y)| + |\partial_\mu V^i(t, x, \mu, y, v)| + |\tilde{\partial} V^i(t, x, \mu, y)| \\ &\le C_T\big(1 + |x| + |y| + W_2(\mu, \mu_0) + |v|\big) \end{aligned}$$

for some fixed measure $\mu_0 \in \mathcal{P}_2(\mathbb{R}^N)$, and C_T is a constant that depends only on T.

(c) We write $V \in \mathcal{C}^{k,k,k}_{b,\mathrm{Lip}}([0, T] \times \mathbb{R}^N \times \mathcal{P}_2(\mathbb{R}^N) \times \mathbb{R}^N; \mathbb{R}^N)$, if the following holds true: for each $i = 1, \ldots, N$, and all multi-indices α, $\tilde{\gamma}$ and γ on $\{1, \ldots, N\}$ and all $\boldsymbol{\beta} \in \mathcal{B}$ satisfying $|\alpha| + |\boldsymbol{\beta}| + |\gamma| + |\tilde{\gamma}| \le k$, the derivative

$$\partial_x^\gamma \tilde{\partial}^{\tilde{\gamma}} \partial_{\boldsymbol{v}}^{\boldsymbol{\beta}} \partial_\mu^\alpha V^i(t, x, \mu, y, \boldsymbol{v})$$

exists and is bounded, Lipschitz continuous, and satisfies the linear growth condition.

(d) We write $h \in \mathcal{C}^{k,k}_{b,\mathrm{Lip}}([0, T] \times \mathbb{R}^N \times \mathbb{R}^N; \mathbb{R}^N)$, if the mapping h does not depend on a measure variable and all the other conditions are satisfied in (c).

4.2.2 Solutions of Directed Chain SDEs

The existence and uniqueness of weak solutions of directed chain SDEs are given in Proposition 4.2.2. The constraint (4.4) plays an essential role here to govern the uniqueness.

Proposition 4.2.2 *Suppose that $V_i, i = 0, 1, \ldots, d$ are Lipschitz in the sense that for every $T > 0$, there exists a constant C_T such that*

$$\sup_i |V_i(t, x_1, \mu_1, y_1) - V_i(t, x_2, \mu_2, y_2)| \leq C_T(|x_1 - x_2| + |y_1 - y_2| + W_2(\mu_1, \mu_2)), \quad 0 \leq t \leq T. \tag{4.5}$$

With the same constant C_T, let us also assume that V_i's have at most linear growth, i.e.

$$\sup_{0 \leq t \leq T} |V_i(t, x, \mu, y)| \leq C_T(1 + |x| + |y| + W_2(\mu, \mu_0)) \tag{4.6}$$

where $\mu_0 \in \mathcal{P}_2(\mathbb{R}^N)$ is fixed. Then there exists a unique weak solution to the directed chain stochastic differential equation (4.3)-(4.4).

The proof is similar to the proof for [52, Proposition 2.1] with a little generalization. Because of the appearance of the neighborhood process, we cannot expect a strong solution of the directed chain SDEs (4.3) (*cf.* Proposition 2.1 of [52]).

Proof: Let us first assume boundedness on all coefficients, i.e.

$$\sup_i |V_i(t, x_1, \mu_1, y_1) - V_i(t, x_2, \mu_2, y_2)| \leq C_T((|x_1 - x_2| + |y_1 - y_2| + W_2(\mu_1, \mu_2)) \wedge 1). \tag{4.7}$$

We shall evaluate the Wasserstein distance between two probability measures μ_1, μ_2 on the space $C([0, T], \mathbb{R}^N)$ of continuous functions, namely

$$D_t(\mu_1,\mu_2) := \inf\left\{ \int \big(\sup_{0\le s\le t} |X_s(\omega_1) - X_s(\omega_2)|^2 \wedge 1\big)\,\mathrm{d}\mu(\omega_1,\omega_2)\right\}^{1/2} \tag{4.8}$$

for $0 \le t \le T$, where the infimum is taken over all the joint measure μ on $C([0,T],\mathbb{R}^N) \times C([0,T],\mathbb{R}^N)$ such that their marginals are μ_1, μ_2, and the initial joint distribution is $\theta \otimes \theta$, the initial marginals are θ. Here, $X_s(\omega) = \omega(s), 0 \le s \le T$ is the coordinate map of $\omega \in C([0,T],\mathbb{R}^N)$. $D_T(\cdot,\cdot)$ defines a complete metric on $\mathcal{M}(C([0,T],\mathbb{R}^N))$, which gives the weak topology to it.

Given the distribution $m = \mathrm{Law}(\widetilde{X}) \in \mathcal{M}(C([0,T],\mathbb{R}^N))$ of $\widetilde{X}$ that is independent of B and X_0, it is well known that the following stochastic differential equation

$$\mathrm{d}X_t^m = V_0(t, X_t^m, m_t, \widetilde{X}_t)\,\mathrm{d}t + \sum_{i=1}^d V_i(t, X_t^m, m_t, \widetilde{X}_t)\,\mathrm{d}B_t^i \tag{4.9}$$

has a unique solution, based on the Lipschitz and linear growth condition on coefficients. Since $\widetilde{X}$ is independent of Brownian motion B, we can only expect the solution exists in weak sense.

Define a map $\Phi : \mathcal{M}(C([0,T],\mathbb{R}^N)) \to \mathcal{M}(C([0,T],\mathbb{R}^N))$ by $\Phi(m) := \mathrm{Law}(X_\cdot^m)$. We shall find a fixed point m^* for the map Φ such that $\Phi(m^*) = m^*$ to show the uniqueness of the solution in the weak sense.

Assume $m_1 = \mathrm{Law}(\widetilde{X}^1)$ and $m_2 = \mathrm{Law}(\widetilde{X}^2)$, then by rewriting (4.9) we have

$$X_t^{m_i} = \theta + \int_0^t V_0(t, X_t^{m_i}, m_{i,t}, \widetilde{X}_t^i)\,\mathrm{d}s + \sum_{i=1}^d \int_0^t V_i(t, X_t^{m_i}, m_{i,t}, \widetilde{X}_t^i)\,\mathrm{d}B_s^i, \quad i = 1, 2.$$

Note that here we fix the initial state to be the same θ for both X^{m_1} and X^{m_2}. Let m be a joint distribution of m_1, m_2 and $\mathbb{E}^m$ be the expectation under m. Under the stronger assumption (4.7),

$$
\begin{aligned}
&\mathbb{E}^m\big[\sup_{0\le s\le t}(X_s^{m_1}-X_s^{m_2})^2\big]\\
&\le 2\mathbb{E}^m\left[\sup_{0\le s\le t}\int_0^s \big(V_0(v,X_v^{m_1},m_{1,v},\widetilde{X}_v^1)-V_0(v,X_v^{m_2},m_{2,v},\widetilde{X}_v^2)\big)^2\,\mathrm{d}v\right]\\
&\qquad+2^d\sum_{i=1}^d\mathbb{E}^m\left[\sup_{0\le s\le t}\int_0^s \big(V_i(v,X_v^{m_1},m_{1,v},\widetilde{X}_v^1)-V_i(v,X_v^{m_2},m_{2,v},\widetilde{X}_v^2)\big)^2\,\mathrm{d}v\right]\\
&\le 2^{d+3}(d+1)C_T\mathbb{E}^m\left[\sup_{0\le s\le t}\int_0^s \big((X_v^{m_1}-X_v^{m_2})^2+W_2(m_{1,v},m_{2,v})^2+(\widetilde{X}_v^1-\widetilde{X}_v^2)^2\big)\wedge 1\,\mathrm{d}v\right]\\
&\le C\cdot\mathbb{E}^m\left[\int_0^t\sup_{0\le v\le s}(X_v^{m_1}-X_v^{m_2})^2\wedge 1\,\mathrm{d}s\right]+C\int_0^t W_2(m_{1,s},m_{2,s})^2\wedge 1\,\mathrm{d}s\\
&\quad+C\cdot\mathbb{E}^m\left[\int_0^t\sup_{0\le v\le s}(\widetilde{X}_v^1-\widetilde{X}_v^2)^2\wedge 1\,\mathrm{d}s\right]\\
&=C\int_0^t\mathbb{E}^m\big[\sup_{0\le v\le s}(X_v^{m_1}-X_v^{m_2})^2\wedge 1\big]\,\mathrm{d}s+C\int_0^t W_2(m_{1,s},m_{2,s})^2\wedge 1\,\mathrm{d}s\\
&\quad+C\int_0^t\mathbb{E}^m\big[\sup_{0\le v\le s}(\widetilde{X}_v^1-\widetilde{X}_v^2)^2\wedge 1\big]\,\mathrm{d}s
\end{aligned}
\tag{4.10}
$$

where we replace $2^{d+3}(d+1)C_T$ by C. Note that by construction,

$$W_2(m_{1,s},m_{2,s})^2\wedge 1\le D_s(m_1,m_2)^2.$$

By taking infimum over all m such that its marginals are m_1, m_2, the third term in (4.10) is bounded by

$$C\int_0^t D_s(m_1,m_2)^2\,\mathrm{d}s.$$

Hence we get

$$D_t(\Phi(m_1),\Phi(m_2))^2\le C\int_0^t D_s(\Phi(m_1),\Phi(m_2))^2\,\mathrm{d}s+2C\int_0^t D_s(m_1,m_2)^2\,\mathrm{d}s.$$

Then by applying Gronwall's lemma, we get

$$D_t(\Phi(m_1), \Phi(m_2))^2 \le 2Ce^{CT} \int_0^t D_s(m_1, m_2)^2 \, \mathrm{d}s. \tag{4.11}$$

For every $m \in \mathcal{M}(C([0,T], \mathbb{R}^N))$, let $m_1 = m$, $m_2 = \Phi(m)$, we get by iterating (4.11),

$$D_T(\Phi^{(k+1)}(m), \Phi^{(k)}(m)) \le \sqrt{\frac{(2CTe^{CT})^k}{k!}} D_T(\Phi(m), m), \quad \forall k \in \mathbb{N}. \tag{4.12}$$

This implies that $\{\Phi^{(k)}(m), k \in \mathbb{N}\}$ forms a Cauchy sequence converging to a fixed point m^*. This m^* is the weak solution to directed chain SDE (4.3)&(4.4). To relax the bounded condition (4.7) to (4.5), we can first cut $[0,T]$ to small time intervals such that the bounded assumption is satisfied on each interval, and establish the uniqueness on each interval and finally paste them together. ■

Proposition 4.2.3 (Regularity) *If $\theta \in L^2(\Omega)$, the solution of directed chain SDE* (4.3)-(4.4) *satisfies*

$$\|X^\theta\|_{\mathcal{S}_T^2} \le C(1 + \|\theta\|_2),$$

where $C = C(T)$, under the assumption of Proposition 4.2.2.

Proof: The proof follows from a similar procedures as [52, Proposition 2.2]. ■

4.2.3 Flow Property

In the last part of this section, we discuss the flow property of directed SDEs informally. After establishing the solution to the exact directed chain SDE (4.3), we also consider the process $X_\cdot^{x,[\theta]}$ that satisfies

$$X_\cdot^{x,[\theta]} = x + \int_0^\cdot V_0(s, X_s^{x,[\theta]}, [X_s^\theta], \widetilde{X}_s) \, \mathrm{d}s + \sum_{i=1}^d \int_0^\cdot V_i(s, X_s^{x,[\theta]}, [X_s^\theta], \widetilde{X}_s) \, \mathrm{d}B_s^i, \tag{4.13}$$

where $x \in \mathbb{R}^N$ is a fixed initial point and $\widetilde{X}_\cdot$ is the neighborhood process satisfying the constraints (4.4), i.e., $\mathrm{Law}(\widetilde{X}_\cdot) = \mathrm{Law}(X^\theta_\cdot) = [X^\theta_\cdot]$. Note that $X^{x,[\theta]}_\cdot$ in (4.13) is strongly solvable with pathwise uniqueness, given the unique, weak solution $(X^\theta_\cdot, \widetilde{X}_\cdot, B_\cdot)$ as in Proposition 4.2.2.

Proposition 4.2.4 (Regularity) *Under the assumption in Proposition 4.2.2, for every* $\theta \in L^2(\Omega)$, $T > 0$ *and* $p \geq 2$, *there exists a constant* $C = C(T, p)$ *such that the solution of* (4.13) *satisfies*

$$\|X^{x,[\theta]}\|_{\mathcal{S}^p_T} \leq C(1 + \|\theta\|_2 + |x|).$$

Proof: The proof follows from the Burkholder-Davis-Gundy inequality and Proposition 4.2.3, which is also satisfied by $\widetilde{X}$. ■

For the explanation purpose, we will add a superscript $\tilde{\theta}$ such that $X^{x,\theta,\tilde{\theta}}_t := X^{x,\theta}_t$ and $\widetilde{X}^{\tilde{\theta}}_t := \widetilde{X}_t$ to emphasize the neighborhood process start at $\tilde{\theta}$, independent of θ. This notation is only used in this subsection. Thus, with the notation $B^0_t \equiv t$, $t \geq 0$, (4.13) is read as

$$X^{x,[\theta],\tilde{\theta}}_t = x + \sum_{i=0}^{d} \int_0^t V_i(s, X^{x,[\theta],\tilde{\theta}}_s, [X^\theta_s], \widetilde{X}^{\tilde{\theta}}_s)\, \mathrm{d}B^i_s, \quad t \geq 0. \tag{4.14}$$

For different initial points x, x' and the corresponding solutions $X^{x,[\theta],\tilde{\theta}}_\cdot$ and $X^{x',[\theta],\tilde{\theta}}_\cdot$, we have the following estimate: there exists a constant $C > 0$ such that

$$\mathbb{E}\big[\sup_{t \leq s \leq T} \big|X^{x,[\theta],\tilde{\theta}}_s - X^{x',[\theta],\tilde{\theta}}_s\big|^2\big] \leq C|x - x'|^2$$

again by the Lipschitz continuity and the Burkholder-Davis-Gundy inequality. By the pathwise uniqueness of $X^{x,[\theta],\theta}_\cdot$, given the pair $(X^\theta_\cdot, \tilde{X}^{\tilde{\theta}}_\cdot)$, it follows

$$X^{x,[\theta],\tilde{\theta}}_s\Big|_{x=\theta} = X^\theta_s, \quad 0 \leq s \leq T. \tag{4.15}$$

Now, with some abuse of notations, we denote by $X_\cdot^{t,x,[\theta],\tilde{\theta}}$ the solution to (4.14) with $X_t^{t,x,[\theta],\tilde{\theta}} = x$, denote by $(X_\cdot^{t,\theta}, \widetilde{X}_\cdot^{t,\tilde{\theta}})$ the solution to (4.3) with $(X_t^{t,\theta}, \widetilde{X}_t^{t,\tilde{\theta}}) = (\theta, \tilde{\theta})$. It follows from (4.15) that by the strong Markov property, for $0 \le t \le s \le r \le T$, we have the flow property

$$(X_r^{s,X_s^{t,x,[\theta],\tilde{\theta}},[X_s^{t,\theta}],\widetilde{X}_s^{t,\tilde{\theta}}}, X_r^{s,X_s^{t,\theta}}, \widetilde{X}_r^{s,\widetilde{X}_s^{t,\tilde{\theta}}}) = (X_r^{t,x,[\theta],\tilde{\theta}}, X_r^{t,\theta}, \widetilde{X}_r^{t,\tilde{\theta}}). \tag{4.16}$$

We close section 4.2 at this point. After the introduction of the partial Malliavin derivatives, we will revisit the directed chain SDE and study the regularities of its derivatives.

4.3 Partial Malliavin Calculus

In this section, we will briefly review the Malliavin calculus, following [128], and introduce the partial Malliavin derivatives for our problem.

Malliavin Calculus. Let $H := L^2([0,T], \mathbb{R}^d)$ be the Hilbert space equipped with the norm $\| \cdot \|_H$, where we define Gaussian process, and $\mathcal{S}$ be the set of smooth functionals of the form

$$F(\omega) = f\bigg(\int_0^T h_1(t) \cdot \mathrm{d}B_t(\omega), \dots, \int_0^T h_n(t) \cdot \mathrm{d}B_t(\omega) \bigg),$$

where $f \in \mathcal{C}_p^\infty(\mathbb{R}^n; \mathbb{R})$ and $\int_0^T h_i(t) \cdot \mathrm{d}B_t = \sum_{j=1}^d \int_0^T h_i^j(t) \, \mathrm{d}B_t^j$.

Then the Malliavin derivative of F, denoted by $\boldsymbol{D}F \in L^2(\Omega; H)$ is given by:

$$\boldsymbol{D}F = \sum_{i=1}^n \partial^i f \bigg(\int_0^T h_1(t) \cdot \mathrm{d}B_t(\omega), \dots, \int_0^T h_n(t) \cdot \mathrm{d}B_t(\omega) \bigg) h_i. \tag{4.17}$$

As stated in [128], because of the isometry $L^2(\Omega \times [0,T]; \mathbb{R}^d) \simeq L^2(\Omega; H)$, we are able to identify $\boldsymbol{D}F$ with a process $(\boldsymbol{D}_r F)_{r\in[0,T]}$ taking values in $\mathbb{R}^d$. Moreover, the set of smooth functionals, denoted by $\mathcal{S}$, is dense in $L^p(\Omega)$ for any $p \geq 1$ and $\boldsymbol{D}$ is closable as operator from $L^p(\Omega)$ to $L^p(\Omega; H)$. We define $\mathbb{D}^{1,p}$ as the closure of the set $\mathcal{S}$ within $L^p(\Omega; \mathbb{R}^d)$ with respect to the norm

$$\|F\|_{\mathbb{D}^{1,p}} = \big(\mathbb{E}|F|^p + \mathbb{E}\|\boldsymbol{D}F\|_H^p\big)^{\frac{1}{p}}.$$

The higher order Malliavin derivatives are defined similarly, denoted by $\boldsymbol{D}^{(k)}F$, which is a random variable with values in $H^{\otimes k}$ defined as

$$\boldsymbol{D}^{(k)}F := \sum_{i_1,\dots,i_k=1}^{n} \partial^{(i_1,\dots,i_k)} f\bigg(\int_0^T h_1(t)\cdot \mathrm{d}B_t(\omega), \dots, \int_0^T h_n(t)\cdot \mathrm{d}B_t(\omega)\bigg).$$

We define $\mathbb{D}^{k,p}$ to be the closure of the set of smooth functions $\mathcal{S}$ with respect to the norm:

$$\|F\|_{\mathbb{D}^{k,p}} = \big(\mathbb{E}|F|^p + \sum_{j=1}^{k} \mathbb{E}\|\boldsymbol{D}^{(j)}F\|_H^p\big)^{\frac{1}{p}}.$$

The Malliavin derivative is also well defined for the general E-valued random variables, where E is some separable Hilbert space, and we write $\mathbb{D}^{1,p}(E)$ to be the closure of $\mathcal{S}$ under some appropriate metric with respect to E. We will use notation $\mathbb{D}^{1,\infty} = \cap_{p\geq 1}\mathbb{D}^{1,p}$. The adjoint operator of $\boldsymbol{D}$ is introduced as follows.

Definition 4.3.1 (Definition 1.3.1, [128]) *We denote by δ the adjoint of the operator $\boldsymbol{D}$. That is, δ is an unbounded operator on $L^2(\Omega; H)$ with values in $L^2(\Omega)$ such that*

1. *The domain of δ, denoted by $\operatorname{Dom}\delta$, is the set of H-valued square integrable random variables $u \in L^2(\Omega; H)$ such that*

$$\big|\mathbb{E}[\langle \boldsymbol{D}F, u\rangle_H]\big| \leq c\|F\|_2$$

for all $F \in \mathbb{D}^{1,2}$*, where* c *is some constant depending on* u*.*

2. *If* u *belongs to* $\operatorname{Dom}\delta$*, then* $\delta(u)$ *is the element of* $L^2(\Omega)$ *characterized by*

$$\mathbb{E}[F\delta(u)] = \mathbb{E}[\langle \boldsymbol{D}F, u\rangle_H]$$

for any $F \in \mathbb{D}^{1,2}$*.*

4.3.1 Partial Malliavin Calculus

The following remark motivate us to use partial Malliavin calculus.

Remark 4.3.2 *Because of the appearance of a neighborhood process* $\widetilde{X}_\cdot$*, we propose the following problem. We shall remark that almost everything satisfied by the McKean-Vlasov SDE in [44] is also satisfied by our directed chain SDE. However, we cannot directly apply their approach to argue the existence, contituity and differentiability of the density function of* $X_t^{x,[\theta]}$*. The reason is that a key step connecting the Malliavin derivative and* $\partial_x X_t^{x,[\theta]}$*, which is defined in* (4.13)*, may not hold in our case, i.e., in general, the identity*

$$\partial_x X_t^{x,[\theta]} = \boldsymbol{D}_r X_t^{x,[\theta]} \sigma^\top \left(\sigma\sigma^\top\right)^{-1}(r, X_r^{x,[\theta]}, [X_r^\theta], \widetilde{X}_r)\partial_x X_r^{x,[\theta]} \tag{4.18}$$

does not hold for any $r \le t$*. Thus, we cannot directly make use of the integration by parts formulae in [44], and hence, we cannot argue the smoothness of* $X_t^{x,[\theta]}$*.*

Question 4.3.3 *How can we make connections between the first order derivative* $\partial_x X_t^{x,[\theta]}$ *and the Malliavin derivatives similar to* (4.18)*, which would render us to apply integration by parts formula?*

To address Question 4.3.3, we consider the partial Malliavin derivative in [129]. Let $\mathcal{G} := \sigma(\{\widetilde{X}_{t_i}, \forall t_i \in \mathbb{Q}_T\})$ be the sigma algebra generated by the neighborhood process at all rational time, where $\mathbb{Q}_T = \mathbb{Q} \cap [0, T]$ denote the collection of all rational numbers in $[0, T]$. Thanks to the continuity of $\widetilde{X}$, considering all rational time stamps is equivalent to considering the whole time interval $[0, T]$, i.e. $\mathcal{G} = \sigma(\widetilde{X}_s, 0 \le s \le T)$. We associate to $\mathcal{G}$ the family of subspaces defined by the orthogonal complement to the subspace generated by $\{\boldsymbol{D}\widetilde{X}_{t_i}(\omega), t_i \in \mathbb{Q}_T\}$, i.e.,

$$K(\omega) = \langle \boldsymbol{D}\widetilde{X}_{t_i}(\omega), t_i \in \mathbb{Q}_T \rangle^{\perp}.$$

Since $\mathcal{G}$ is generated by countably many random variables, we say it is countably smoothly generated. Then the family

$$\mathcal{H} := \{K(\omega), \omega \in \Omega\}$$

has a measurable projection by this countably smoothness of $\mathcal{G}$. We define the partial Malliavin derivative operator as $\boldsymbol{D}^{\mathcal{H}}$.

Definition 4.3.4 (Definition 2.1, [129]) *We define the partial derivative operator*

$$\boldsymbol{D}^{\mathcal{H}} : \mathbb{D}^{1,2} \to L^2(\Omega, \mathcal{H})$$

as the projection of $\boldsymbol{D}$ on $\mathcal{H}$, namely, for any $F \in \mathbb{D}^{1,2}$,

$$\boldsymbol{D}^{\mathcal{H}} F = \mathrm{Proj}_{\mathcal{H}}(\boldsymbol{D}F) = \mathrm{Proj}_{K(\omega)}(\boldsymbol{D}F)(\omega).$$

This operator, similar to $\boldsymbol{D}$, admits an identification with a process $(\boldsymbol{D}^{\mathcal{H}}_r)_{r \in [0,T]}$. For the higher order Malliavin derivative $\boldsymbol{D}^{\mathcal{H},(j)} : \mathbb{D}^{1,2} \to L^2(\Omega, \mathcal{H}^{\otimes j})$, we can defined it in an *iterative manner, that is*

$$\boldsymbol{D}^{\mathcal{H},(j+1)}F = \mathrm{Proj}_{\mathcal{H}}(\boldsymbol{D}\boldsymbol{D}^{\mathcal{H},(j)}F).$$

Moreover, we define the norm associated with $\boldsymbol{D}^{\mathcal{H}}$ *by*

$$\|F\|_{\mathbb{D}^{k,p}_{\mathcal{H}}} = \big(\mathbb{E}|F|^p + \sum_{j=1}^{k} \mathbb{E}\|\boldsymbol{D}^{\mathcal{H},(j)}F\|_H^p\big)^{\frac{1}{p}},$$

where $\boldsymbol{D}^{\mathcal{H},(j)}$ *is defined as*

$$\boldsymbol{D}^{\mathcal{H},(j)}F = \mathrm{Proj}_{\mathcal{H}}(\boldsymbol{D}^{(j)}F) = \mathrm{Proj}_{K(\omega)}(\boldsymbol{D}^{(j)}F)(\omega).$$

Now we have the important fact that $\boldsymbol{D}^{\mathcal{H}}\widetilde{X}_t = 0$. This is because $\widetilde{X}_t$ is $\mathcal{G}$ measurable and hence equivalently

$$\boldsymbol{D}\widetilde{X}_t \in \langle \boldsymbol{D}\widetilde{X}_{t_i}, t_i \in \mathbb{Q}_T \cup \{t\}\rangle; \quad t \in [0,T]. \tag{4.19}$$

Then the projection of $\boldsymbol{D}\widetilde{X}_t$ on to the orthogonal complement of $\langle \boldsymbol{D}\widetilde{X}_{t_i}, t_i \in \mathbb{Q}_T \cup \{t\}\rangle$ must be zero. Similar to the common Malliavin calculus, we have an adjoint operator of $\boldsymbol{D}^{\mathcal{H}}$, which is denoted by $\delta_{\mathcal{H}}$, as well as the integration by parts formula for the partial Malliavin calculus.

Definition 4.3.5 (Definition 2.3, [129]) *Set* $\mathrm{Dom}\,\delta_{\mathcal{H}} = \{u \in L^2(\Omega; H) : \mathrm{Proj}_{\mathcal{H}}u \in \mathrm{Dom}\,\delta\}$. *For any* $u \in \mathrm{Dom}\,\delta_{\mathcal{H}}$, *set* $\delta_{\mathcal{H}}(u) = \delta(\mathrm{Proj}_{\mathcal{H}}u)$.

Following Definition 4.3.4 and 4.3.5, we have integration by parts formula for $\boldsymbol{D}^{\mathcal{H}}$ and $\delta_{\mathcal{H}}$

$$\mathbb{E}[\langle h, \boldsymbol{D}^{\mathcal{H}}F\rangle] = \mathbb{E}[\langle \mathrm{Proj}_{\mathcal{H}}h, \boldsymbol{D}F\rangle] = \mathbb{E}[F\delta_{\mathcal{H}}(h)]. \tag{4.20}$$

Kusuoka-Stroock Processes In order to derive the differentiability of the density function, we mimic the procedure in [44] and need to develop the integration-by-parts formulae introduced in the works of [96] and [94]. Kusuoka-Strook process is an important tool to analysis of stochastic differential equations with many applications, see [45, 42, 43, 46] for example.

Definition 4.3.6 (Definition 2.8 in [44]) *Let E be a separable Hilbert space and let $r \in \mathbb{R}$, $q, M \in \mathbb{N}$. We denote by $\mathbb{K}_r^q(E, M)$ the set of processes $\Psi : [0,T] \times \mathbb{R}^N \times \mathcal{P}_2(\mathbb{R}^N) \to \mathbb{D}^{M,\infty}(E)$ satisfying the following:*

1. *For any multi-indices $\alpha, \boldsymbol{\beta}, \gamma$ satisfying $|\alpha| + |\boldsymbol{\beta}| + |\gamma| \leq M$, the function*

$$[0,T] \times \mathbb{R}^N \times \mathcal{P}_2(\mathbb{R}^N) \ni (t, x, [\theta]) \mapsto \partial_x^\gamma \partial_{\boldsymbol{v}}^{\boldsymbol{\beta}} \partial_\mu^\alpha \Psi(t, x, [\theta], v) \in L^p(\Omega)$$

exists and is continuous for all $p \geq 1$.

2. *For any $p \geq 1$ and $m \in \mathbb{N}$ with $|\alpha| + |\boldsymbol{\beta}| + |\gamma| + m \leq M$, we have*

$$\sup_{v \in (\mathbb{R}^N)^{\#\boldsymbol{\beta}}} \sup_{t \in (0,T]} t^{-r/2} \left\| \partial_x^\gamma \partial_{\boldsymbol{v}}^{\boldsymbol{\beta}} \partial_\mu^\alpha \Psi(t, x, [\theta], v) \right\|_{\mathbb{D}_{\mathcal{H}}^{m,p}(E)} \leq C\left(1 + |x| + \|\theta\|_2\right)^q$$

In our discussion, we do not consider the differentiability of the process X with respect to the initial state of its neighborhood $\widetilde{X}$. This above definition of $\mathbb{K}_r^q(E)$ is almost the same as the definition in [44, Definition 2.8], except for the norm. The reason is that we only care about the existence and smoothing properties of the density function of $X^{x,[\theta]}$ and have to use the partial Malliavin calculus. We remark that although the norms are different, all the regularity results under the norm $\|\cdot\|_{\mathbb{D}^{k,p}}$ also holds under our norm $\|\cdot\|_{\mathbb{D}_{\mathcal{H}}^{k,p}}$ because of the Hölder's inequality. To get the smoothness of density functions of a process start from a fixed initial point, we use $\mathcal{K}_r^q(\mathbb{R}, M)$ as the class of Kusuoka-

Stroock processes which do not depend on a measure term. By [44, Lemma 2.11], if $\Psi \in \mathbb{K}_r^q(E, M)$, then $\Phi(t, x, y) := \Psi(t, x, \delta_x, y) \in \mathcal{K}_r^q(E, M)$.

4.4 Smoothness of Directed Chain SDEs

4.4.1 Regularities of Solutions of Directed Chain SDEs

For the purpose of establishing the integration by parts formulae for the directed chain SDEs and applying the results in [44, Theorem 6.1], we only need to check all the regularities conditions with respect to parameters (θ, x) contained in [44, Section 3].

Proposition 4.4.1 (First-order derivatives) *Suppose that $V_0, \ldots, V_d \in \mathcal{C}_{b,Lip}^{1,1,1}(\mathbb{R}^+ \times \mathbb{R}^N \times \mathcal{P}_2(\mathbb{R}^N) \times \mathbb{R}^N; \mathbb{R}^N)$. Then the following statements hold:*

1. *There exists a modification of $X^{x,[\theta]}$ such that for all $t \in [0, T]$, the map $x \mapsto X_t^{x,[\theta]}$ is $\mathbb{P}$-a.s. differentiable. We denote the derivative by $\partial_x X^{x,[\theta]}$ and note that it solves the following SDE*

$$\partial_x X_t^{x,[\theta]} = Id_N + \sum_{i=0}^{d} \int_0^t \left\{ \partial V_i(s, X_s^{x,[\theta]}, [X_s^\theta], \widetilde{X}_s) \partial_x X_s^{x,[\theta]} \right\} \mathrm{d}B_s^i \tag{4.21}$$

for every $t \in [0, T]$.

2. *For all $t \in [0, T]$, the maps $\theta \mapsto X_t^\theta$ and $\theta \mapsto X_t^{x,[\theta]}$ are Fréchet differentiable in $L^2(\Omega)$, i.e. there exists a linear continuous map $\mathcal{D}X_t^\theta : L^2(\Omega) \to L^2(\Omega)$ such that for all $\gamma \in L^2(\Omega)$,*

$$\|X_t^{\theta+\gamma} - X_t^\theta - \mathcal{D}X_t^\theta(\gamma)\|_2 = o(\|\gamma\|_2) \quad as \quad \|\gamma\|_2 \to 0,$$

and similarly for $X_t^{x,[\theta]}$. These processes satisfy the following stochastic differential *equations*

$$\begin{aligned}\mathcal{D}X_t^{x,[\theta]}(\gamma) = \sum_{i=0}^{d}\int_0^t \Big[&\partial V_i(s, X_s^{x,[\theta]}, [X_s^\theta], \widetilde{X}_s)\mathcal{D}X_s^{x,[\theta]}(\gamma)\\ &+ \tilde{\partial} V_i(s, X_s^{x,[\theta]}, [X_s^\theta], \widetilde{X}_s)\mathcal{D}\widetilde{X}_s(\gamma)\\ &+ \mathcal{D}V_i'(s, X_s^{x,[\theta]}, X_s^\theta, \widetilde{X}_s)(\mathcal{D}X_s^\theta(\gamma))\Big]\,\mathrm{d}B_s^i, \end{aligned} \tag{4.22}$$

$$\begin{aligned}\mathcal{D}X_t^{\theta}(\gamma) = \gamma + \sum_{i=0}^{d}\int_0^t \Big[&\partial V_i(s, X_s^{\theta}, [X_s^\theta], \widetilde{X}_s)\mathcal{D}X_s^{\theta}(\gamma) + \tilde{\partial} V_i(s, X_s^{\theta}, [X_s^\theta], \widetilde{X}_s)\mathcal{D}\widetilde{X}_s(\gamma)\\ &+ \mathcal{D}V_i'(s, X_s^{\theta}, X_s^\theta, \widetilde{X}_s)(\mathcal{D}X_s^\theta(\gamma))\Big]\,\mathrm{d}B_s^i \end{aligned} \tag{4.23}$$

where V_i' is the lifting of V_i. Moreover, for each $x \in \mathbb{R}^N$, $t \in [0,T]$, the map $\mathcal{P}_2 \ni [\theta] \mapsto X_t^{x,[\theta]} \in L^2(\Omega)$ is differentiable. So, $\partial_\mu X_t^{x,[\theta]}(v)$ exists and it satisfies the following equation

$$\begin{aligned}\partial_\mu X_t^{x,[\theta]}(v) = \sum_{i=0}^{d}\int_0^t \Big\{&\partial V_i\big(s, X_s^{x,[\theta]}, [X_s^\theta], \widetilde{X}_s\big)\partial_\mu X_s^{x,[\theta]}(v)\\ &+ \tilde{\partial} V_i\big(s, X_s^{x,[\theta]}, [X_s^\theta], \widetilde{X}_s\big)\partial_\mu \widetilde{X}_s(v)\\ &+ \mathbb{E}'\Big[\partial_\mu V_i\big(s, X_s^{x,[\theta]}, [X_s^\theta], \widetilde{X}_s, (X_s^{v,[\theta]})'\big)\partial_x (X_s^{v,[\theta]})'\Big]\\ &+ \mathbb{E}'\Big[\partial_\mu V_i\big(s, X_s^{x,[\theta]}, [X_s^\theta], \widetilde{X}_s, (X_s^{\theta'})'\big)\partial_\mu (X_s^{\theta',[\theta]})'(v)\Big]\Big\}\,\mathrm{d}B_s^i, \end{aligned} \tag{4.24}$$

where $(X_s^{\theta'})'$ is a copy of X_s^θ on the probability space $(\Omega', \mathcal{F}', \mathbb{P}')$. Similarly, $\partial_x (X_s^{v,[\theta]})'$ is a copy of $\partial_x X_s^{v,[\theta]}$ and $\partial_\mu (X_s^{\theta',[\theta]})' = \partial_\mu (X_s^{x,[\theta]})'\big|_{x=\theta'}$. Finally, the following representation holds for all $\gamma \in L^2(\Omega)$:

$$\mathcal{D}X_t^{x,[\theta]}(\gamma) = \mathbb{E}'[\partial_\mu X_t^{x,[\theta]}(\theta')\gamma']. \tag{4.25}$$

3. *For all* $t \in [0,T]$*,* $X_t^{x,[\theta]}, X_t^{\theta} \in \mathbb{D}^{1,\infty}$*. Moreover,* $\boldsymbol{D}_r^{\mathcal{H}} X^{x,[\theta]} = \Big(\boldsymbol{D}_r^{\mathcal{H},j} (X^{x,[\theta]})^i \Big)_{\substack{1 \le j \le N \\ 1 \le i \le d}}$ *satisfies, for* $0 \le r \le t$

$$\boldsymbol{D}_r^{\mathcal{H}} X_t^{x,[\theta]} = \sigma\big(r, X_r^{x,[\theta]}, [X_r^{\theta}], \widetilde{X}_r\big) + \sum_{i=0}^{d} \int_r^t \Big(\partial V_i(s, X_s^{x,[\theta]}, [X_s^{\theta}], \widetilde{X}_s) \boldsymbol{D}_r^{\mathcal{H}} X_s^{x,[\theta]} \Big) \, \mathrm{d}B_s^i, \tag{4.26}$$

where $\sigma\big(r, X_r^{x,[\theta]}, [X_r^{\theta}], \widetilde{X}_r\big)$ *is the* $N \times d$ *matrix with columns* $V_1, \ldots, V_d$*.*

Proof:

1. The SDE of $X^{x,[\theta]}$ satisfies a classical SDE with adapted coefficients, by [92, Theorem 7.6.5] there exists a modification of $X_t^{x,[\theta]}$ which is continuously differentiable in x, and the first derivative satisfies (4.21).

2. The maps $\theta \mapsto X_t^{\theta}$ and $\theta \mapsto X_t^{x,[\theta]}$ are Fréchet differentiable by [37, Lemma 4.17]. Then (4.22) and (4.23) follow from direct computation.

   Let us first rewrite the equation for $\mathcal{D}X_t^{\theta}(\gamma)$ in terms of the lifting V',

$$\begin{aligned}\mathcal{D}X_t^{\theta}(\gamma) = \gamma + \sum_{i=0}^{d} \int_0^t \Big[&\partial V_i(s, X_s^{\theta}, [X_s^{\theta}], \widetilde{X}_s) \mathcal{D}X_s^{\theta}(\gamma) + \tilde{\partial} V_i(s, X_s^{\theta}, [X_s^{\theta}], \widetilde{X}_s) \mathcal{D}\widetilde{X}_s(\gamma) \\ &+ \mathbb{E}'\big[\partial_\mu V_i'(s, X_s^{\theta}, [X_s^{\theta}], \widetilde{X}_s, (X_s^{\theta'})')(\mathcal{D}(X_s^{\theta'})'(\gamma'))\big] \Big] \, \mathrm{d}B_s^i. \end{aligned} \tag{4.27}$$

   We then consider the equation that we are going to prove for $\partial_\mu X_s^{\theta',[\theta]}(v)$, evaluated at $v = \theta''$ and multiplied by γ'' with both random variables defined on a probability space $(\Omega'', \mathcal{F}'', \mathbb{P}'')$. Then taking expectation with respect to $\mathbb{P}''$, we get

$$\begin{aligned}\mathbb{E}''\big[\partial_\mu X_t^{\theta',[\theta]}(\theta'')\gamma''\big] = \sum_{i=0}^{d} \int_0^t \Bigg\{ & \partial V_i(s, X_s^\theta, [X_s^\theta], \widetilde{X}_s)\mathbb{E}''[\partial_\mu X_s^{\theta',[\theta]}(\theta'')\gamma''] \\ & + \tilde{\partial} V_i(s, X_s^\theta, [X_s^\theta], \widetilde{X}_s)\mathbb{E}''[\partial_\mu \widetilde{X}_s \gamma''] \\ & + \mathbb{E}''\mathbb{E}'\Bigg[\partial_\mu V_i(s, X_s^\theta, [X_s^\theta], \widetilde{X}_s, (X_s^{\theta'',[\theta]})')\partial_x (X_s^{\theta'',[\theta]})'\gamma''\Bigg] \\ & + \mathbb{E}'\big[\partial_\mu V_i(s, X_s^\theta, [X_s^\theta], \widetilde{X}_s, (X_s^{\theta'})')\mathbb{E}''[\partial_x (X_s^{\theta',[\theta]})'(\theta'')\gamma'']\big]\Bigg\} \,\mathrm{d}B_s^i. \end{aligned} \tag{4.28}$$

Note that since (γ'', θ'') are defined on a separate probability space, we have

$$\mathbb{E}''[\partial_\mu \widetilde{X}_s \gamma''] = \mathcal{D}\widetilde{X}_s(\gamma)$$

and

$$\begin{aligned}&\mathbb{E}''\mathbb{E}'\big[\partial_\mu V_i(s, X_s^\theta, [X_s^\theta], \widetilde{X}_s, (X_s^{\theta'',[\theta]})')\partial_x (X_s^{\theta'',[\theta]})'\gamma''\big] = \\ &\qquad\qquad \mathbb{E}'[\partial_\mu V_i(s, X_s^\theta, [X_s^\theta], \widetilde{X}_s, (X_s^{\theta'})')\partial_x (X_s^{\theta',[\theta]})'\gamma'].\end{aligned}$$

Then the dynamic of $\mathbb{E}''[\partial_\mu X_t^{\theta',[\theta]}(\theta'')\gamma'']$ reduces to

$$\begin{aligned}\mathbb{E}''\big[\partial_\mu X_t^{\theta',[\theta]}(\theta'')\gamma''\big] = \sum_{i=0}^{d} \int_0^t \Bigg\{ & \partial V_i(s, X_s^\theta, [X_s^\theta], \widetilde{X}_s)\mathbb{E}''[\partial_\mu X_s^{\theta',[\theta]}(\theta'')\gamma''] \\ & + \tilde{\partial} V_i(s, X_s^\theta, [X_s^\theta], \widetilde{X}_s)\mathcal{D}\widetilde{X}_s(\gamma) \\ + \mathbb{E}'\big[\partial_\mu V_i(s, X_s^\theta, [X_s^\theta], \widetilde{X}_s, (X_s^{\theta'})')\big[\partial_x & (X_s^{\theta',[\theta]})'\gamma' + \mathbb{E}''[\partial_x (X_s^{\theta'',[\theta]})'(\theta'')\gamma'']\big]\Bigg\} \,\mathrm{d}B_s^i. \end{aligned} \tag{4.29}$$

By (4.21), we can evaluate the equation at $x = \theta$, multiply by x, and derive a dynamic of $\partial_x X_t^{\theta,[\theta]}\gamma$. It can be seen that $\partial_x X_t^{\theta,[\theta]}\gamma + \mathbb{E}''[\partial_\mu X_t^{\theta',[\theta]}(\theta'')\gamma'']$ is equal to

$$\gamma + \sum_{i=0}^{d}\int_0^t \Big\{ \partial V_i(s, X_s^\theta, [X_s^\theta], \widetilde{X}_s)\mathbb{E}''[\partial_\mu X_s^{\theta',[\theta]}(\theta'')\gamma''] + \tilde{\partial} V_i(s, X_s^\theta, [X_s^\theta], \widetilde{X}_s)\mathcal{D}\widetilde{X}_s(\gamma)$$
$$+ \mathbb{E}'\big[\partial_\mu V_i(s, X_s^\theta, [X_s^\theta], \widetilde{X}_s, (X_s^{\theta'})')\big[\partial_x (X_s^{\theta',[\theta]})'\gamma' + \mathbb{E}''[\partial_x (X_s^{\theta'',[\theta]})'(\theta'')\gamma'']\big] \Big\}\,\mathrm{d}B_s^i. \tag{4.30}$$

We observe that this dynamic is identical to the dynamic for $\mathcal{D}X_t^\theta(\gamma)$ in (4.27) and hence they are identical by uniqueness. Similarly, by using this result for $\mathcal{D}X_t^\theta(\gamma)$ and the same procedures, we are able to derive that $\mathbb{E}''[\partial_\mu X_t^{x,[\theta]}(\theta'')\gamma'']$ is equal to $\mathcal{D}X_t^{x,[\theta]}(\gamma)$. So (4.25) is proved. Moreover, $\partial_\mu X_t^{x,[\theta]}(v)$ exists and satisfies equation (4.24) by its definition.

3. We first deduce the Malliavin derivative for X^θ. Consider the Picard iteration given by

$$X_t^{\theta,0} = \theta,$$
$$X_t^{\theta,k+1} = \theta + \sum_{i=0}^{d}\int_0^t V_i(s, X_s^{\theta,k}, [\widetilde{X}_s^k], \widetilde{X}_s^k)\,\mathrm{d}B_s^i,$$

where $\widetilde{X}^k$ is a copy of $X^{\theta,k}$ independent of the Brownian motion and θ. We have shown that such iteration induces a Cauchy sequence $\{\Phi^{(k)}(\mathrm{Law}(X_t^{\theta,0})), k \in \mathbb{N}\}$ and a weak solution of the directed chain SDE. Since V_0, V_i are bounded continuously differentiable, we have

$$\boldsymbol{D}_r^{\mathcal{H},l}[V_i^j(s, X_s^{\theta,k}, [\widetilde{X}_s^k], \widetilde{X}_s^k)] = \partial V_i^j \boldsymbol{D}_r^{\mathcal{H},l} X_s^{\theta,k},$$

where we omit the arguments in V_i's for notation simplicity. Note that $|\partial V_i^j| \le K$

for some constant $K > 0$. We can then deduce that $V_i^j(s, X_s^{\theta,k}, [\widetilde{X}_s^k], \widetilde{X}_s^k) \in \mathbb{D}^{1,\infty}$ by [128, Proposition 1.5.5]. Moreover, the Ito integral

$$\int_0^t V_i^j(s, X_s^{\theta,k}, [\widetilde{X}_s^k], \widetilde{X}_s^k)\,\mathrm{d}B_s^i, \quad i = 1, \dots, d$$

belongs to $\mathbb{D}^{1,2}$ and for $r \le t$, we have

$$\begin{aligned}\boldsymbol{D}_r^{\mathcal{H},l}\Big[\int_0^t V_i^j(s, X_s^{\theta,k}, [\widetilde{X}_s^k], \widetilde{X}_s^k)\,\mathrm{d}B_s^i\Big] &= V_l^j(r, X_r^{\theta,k}, [\widetilde{X}_r^k], \widetilde{X}_r^k) \\ &\quad + \int_r^t \boldsymbol{D}_r^{\mathcal{H},l}[V_i^j(s, X_s^{\theta,k}, [\widetilde{X}_s^k], \widetilde{X}_s^k)]\,\mathrm{d}B_s^i.\end{aligned}$$

On the other hand, the Lebesgue integral $\int_0^t V_0^j(s, X_s^{\theta,k}, [\widetilde{X}_s^k], \widetilde{X}_s^k)\,\mathrm{d}s$ is also in the space $\mathbb{D}^{1,2}$ and have the dynamics

$$\boldsymbol{D}_r^{\mathcal{H},l}\Big[\int_0^t V_0^j(s, X_s^{\theta,k}, [\widetilde{X}_s^k], \widetilde{X}_s^k)\,\mathrm{d}s\Big] = \int_0^t \boldsymbol{D}_r^{\mathcal{H},l}[V_0^j(s, X_s^{\theta,k}, [\widetilde{X}_s^k], \widetilde{X}_s^k)]\,\mathrm{d}s.$$

Therefore, the dynamic of $\boldsymbol{D}_r^{\mathcal{H},l}[X_t^{\theta,k+1}]$ has exactly the form of (4.26) by the chain rule of Malliavin derivative. Due to the reason that $\widetilde{X}^k$ and $X^{\theta,k}$ has the same distribution, by Doob's maximal inequality and Burkholder's inequality,

$$\mathbb{E}[\sup_{0\le s\le t} |\boldsymbol{D}_r^{\mathcal{H},l} X_s^{\theta,k}|^p] \le c_1,$$

where c_1 is a constant that depends only on K, d, p for $p \ge 2$. Moreover, we define a metric similar to (4.8) but raise the power to general $p \ge 1$,

$$D_{t,p}(\mu_1, \mu_2) := \inf\Big\{\int (\sup_{0\le s\le t} |X_s(\omega_1) - X_s(\omega_2)|^p \wedge 1)\,\mathrm{d}\mu(\omega_1, \omega_2)\Big\}^{1/p}.$$

Note that the weak convergence of X^θ holds under any metric $D_{t,p}$ with $p \ge 2$.

We then have the following,

$$D_t(m^{k+1}, m^k)^2 \le c_1 \int_0^t D_{s,p}(\mathrm{Law}(X^{\theta,k}), \mathrm{Law}(X^{\theta,k-1}))\,\mathrm{d}s + c_2 \int_0^t D_s(m^k, m^{k-1})^2\,\mathrm{d}s,$$

by a similar approach as in the proof of Proposition 4.2.2, where c_1, c_2 are positive constants depending on K, d, p and $m^k = \mathrm{Law}(\boldsymbol{D}_r^{\mathcal{H},l} X^{\theta,k})$. By iteration, we get that $\{m^k, k \in \mathbb{N}\}$ forms a Cauchy sequence and has limit. We have now proved that

$$\boldsymbol{D}_r^{\mathcal{H}} X_t^\theta = \sigma\big(r, X_r^\theta, [X_r^\theta], \widetilde{X}_r\big) + \sum_{i=0}^d \int_r^t \left(\partial V_i(s, X_s^\theta, [X_s^\theta], \widetilde{X}_s) \boldsymbol{D}_r^{\mathcal{H}} X_s^\theta\right) \mathrm{d}B_s^i, \quad (4.31)$$

and the solution of $\boldsymbol{D}_r^{\mathcal{H}} X_t^\theta$ exists uniquely in the weak sense. In the iteration, it can be easily proved by induction that $X^{\theta,k} \in \mathbb{D}^{1,\infty}$ and the sequence $\boldsymbol{D}_r^{\mathcal{H}} X_t^{\theta,k}$ is uniformly bounded in $L^p(\Omega; H)$ for $p \ge 2$. Therefore, we have $X_t^\theta \in \mathbb{D}^{1,\infty}$. The proof for $X_t^{x,[\theta]}$ is similar, we can set $X_t^{x,[\theta],0} = \theta$ add another equation for $X_t^{x,[\theta],k}$ into the above Picard iteration

$$X_t^{x,[\theta],k+1} = x + \sum_{i=0}^d \int_0^t V_i(s, X_s^{x,[\theta],k}, [\widetilde{X}_s^k], \widetilde{X}_s^{x,k})\,\mathrm{d}B_s^i.$$

Then the procedures are the same as the deduction for $\boldsymbol{D}_r^{\mathcal{H}} X^\theta$.

■

For the purpose of more general applications, we want to make sure that the density for directed chain SDE is at least second order differentiable, hence we need to extend the above first order regularities to higher orders. Following [44], we provide a result for general case, which characterize $X_t^{x,[\theta]}$ as a Kusuoka-Stroock process.

Theorem 4.4.2 *Suppose* $V_0, \ldots, V_d \in \mathcal{C}_{b,\mathrm{Lip}}^{k,k,k}([0,T] \times \mathbb{R}^N \times \mathcal{P}_2(\mathbb{R}^N) \times \mathbb{R}^N; \mathbb{R}^N)$, *then* $(t, x, [\theta]) \mapsto X_t^{x,[\theta]} \in \mathbb{K}_0^1(\mathbb{R}^N, k)$. *If, in addition,* $V_0, \ldots, V_d$ *are uniformly bounded then*

$(t, x, [\theta]) \mapsto X_t^{x,[\theta]} \in \mathbb{K}_0^0(\mathbb{R}^N, k)$.

Note that [44, Proposition 6.7 and 6.8] can be extended our directed chain case, since the coefficients $V_i : [0,T] \times \mathbb{R}^N \times \mathcal{P}_2(\mathbb{R}^N) \times \mathbb{R}^N \to \mathbb{R}^N$ in directed chain SDEs can be written as a map of the form $\Omega \times [0,T] \times \mathbb{R}^N \times \mathcal{P}_2(\mathbb{R}^N) \ni (\omega, t, x, \mu) \mapsto a(\omega, t, x, \mu) \in \mathbb{R}^N$. This is because the auxiliary dependence on the neighborhood in the coefficients can be thought as the dependence on an initial state x, initial distribution μ and independent Brownian motions, which are implied in the term ω. Moreover, we are able to take care of the extra term with $\mathcal{D}\widetilde{X}_s$ due to the differentiability and regularity of V_i.

Similar to Proposition 4.4.1, each type of derivative (w.r.t. x, μ or v) of $X_t^{x,[\theta]}$ satisfies a linear equation. We will introduce a general linear equation, derive some a priori L^p estimates on the solution and then show that this linear equation is again differentiable under some conditions in the next Lemma. Whenever we say a_k, $k = 1, 2, 3$, we also mean $\tilde{a}_1$.

Lemma 4.4.3 *Let v_r be one element of the tuple $\boldsymbol{v} = (v_1, \ldots, v_{\#\boldsymbol{v}})$ and $Y^{x,[\theta]}(\boldsymbol{v})$ solve the following SDE*

$$\begin{aligned} Y_t^{x,[\theta]}(\boldsymbol{v}) = a_0 + \sum_{i=0}^{d} \int_0^t \Big\{ & a_1^i(s, x, [\theta]) Y_s^{x,[\theta]}(\boldsymbol{v}) + \tilde{a}_1^i(s, x, [\theta]) \widetilde{Y}_s(\boldsymbol{v}) + a_2(s, x, [\theta], \boldsymbol{v}) \\ & + \mathbb{E}'\big[a_3^i(s, x, [\theta], \theta')(Y_s^{\theta',[\theta]})'(\boldsymbol{v}) + \sum_{r=1}^{\#\boldsymbol{v}} a_3^i(s, x, [\theta], \theta')(Y_s^{v_r,[\theta]})'(\boldsymbol{v})\big] \Big\} \mathrm{d}B_s^i, \end{aligned} \tag{4.32}$$

where, for all $i = 1, \ldots, d$, the coefficients $(t, x, [\theta], \boldsymbol{v}) \mapsto a_k(t, x, [\theta], \boldsymbol{v})$ are continuously *in* $L^p(\Omega)$ $\forall p \geq 1$, $k = 1, 2, 3$ *and*

$$\begin{aligned}
&a_0 \in \mathbb{R}^N, \\
&a_1, \tilde{a}_1 : \Omega \times [0,T] \times \mathbb{R}^N \times \mathcal{P}_2(\mathbb{R}^N) \to \mathbb{R}^{N\times N} \\
&a_2 : \Omega \times [0,T] \times \mathbb{R}^N \times \mathcal{P}_2(\mathbb{R}^N) \times (\mathbb{R}^N)^{\#\boldsymbol{v}} \to \mathbb{R}^N \\
&a_3^i : \Omega' \times \Omega \times [0,T] \times \mathbb{R}^N \times \mathcal{P}_2(\mathbb{R}^N) \times \mathbb{R}^N \to \mathbb{R}^{N\times N}.
\end{aligned}$$

In (4.32), $(Y^{\theta',[\theta]})'$ *is a copy of* $Y^{x,[\theta]}$ *on the probability space* $(\Omega', \mathcal{F}', \mathbb{P}')$ *where the initial state is* θ'. *Similarly,* $(Y^{v_r,[\theta]})'$ *is a copy of* $Y^{x,[\theta]}$ *on the probability space* $(\Omega', \mathcal{F}', \mathbb{P}')$ *where the initial state is* v. $\widetilde{Y}$ *is the neighborhood process, which has the same law as* Y^θ *and independent with Brownian motion* B. *If we make the following boundedness assumptions*

1. $\sup_{x\in\mathbb{R}^N,[\theta]\in\mathcal{P}_2(\mathbb{R}^N),\boldsymbol{v}\in(\mathbb{R}^N)^{\#\boldsymbol{v}}} \|a_2(\cdot, x, [\theta], \boldsymbol{v})\|_{\mathcal{S}_T^p} < \infty$,
2. $a_1, \tilde{a}_1$ *and* a_3 *are uniformly bounded,*
3. $\sup_{x\in\mathbb{R}^N,[\theta]\in\mathcal{P}_2(\mathbb{R}^N),\boldsymbol{v}\in(\mathbb{R}^N)^{\#\boldsymbol{v}}} \|a_2(\cdot, x, [\theta], \boldsymbol{v})\|_{\mathcal{S}_T^2} < \infty$.

then we have the following estimate for $C = C(p, T, a_1, a_3)$

$$\|Y^{x,[\theta]}(\boldsymbol{v})\|_{\mathcal{S}_T^p} \leq C(|a_0| + \|a_2(\cdot, x, [\theta], \boldsymbol{v})\|_{\mathcal{S}_T^p} + \|a_2(\cdot, x, [\theta], \boldsymbol{v})\|_{\mathcal{S}_T^2}).$$

Moreover, we also get that the mapping

$$[0,T] \times \mathbb{R}^N \times \mathcal{P}_2(\mathbb{R}^N) \times (\mathbb{R}^N)^{\#\boldsymbol{v}} \ni (t, x, [\theta], \boldsymbol{v}) \mapsto Y_t^{x,[\theta]}(\boldsymbol{v}) \in L^p(\Omega)$$

is continuous.

Proof: Note that $\|\widetilde{Y}(\boldsymbol{v})\|_{\mathcal{S}_T^p} = \|(Y_s^{\theta',[\theta]})'(\boldsymbol{v})\|_{\mathcal{S}_T^p}$ since they have the same distribution. The rest proof is identical to [44, Lemma 6.7] by using Gronwall's lemma and the

Burkholder-Davis-Gundy inequality a couple times. ■

We now consider the differentiability of the generic process satisfying the linear equation in Lemma 4.4.3. To ease the burden on notation, we omit the $(t, x, [\theta])$ in a_k, and write $a_3\big|_{v=\theta'}$ to denote $a_k(s, x, [\theta], \theta')$ for instance.

Proposition 4.4.4 *Suppose that the process $Y^{x,[\theta]}(\boldsymbol{v})$ is as in Lemma 4.4.3. In addition to the assumptions of Lemma 4.4.3, we introduce the following differentiability assumptions:*

(a) For $k = 1, 2, 3$, all $(s, [\theta], \boldsymbol{v}) \in [0, T] \times \mathcal{P}_2(\mathbb{R}^N) \times (\mathbb{R}^N)^{\#\boldsymbol{v}}$ and each $p \geq 1$, $\mathbb{R}^N \ni x \mapsto a_k(s, x, [\theta], \boldsymbol{v}) \in L^p(\Omega)$ is differentiable.

(b) For $k = 1, 2, 3$, all $(s, [\theta], x) \in [0, T] \times \mathcal{P}_2(\mathbb{R}^N) \times \mathbb{R}^N$ and each $p \geq 1$, $(\mathbb{R}^N)^{\#\boldsymbol{v}} \ni \boldsymbol{v} \mapsto a_k(s, x, [\theta], \boldsymbol{v}) \in L^p(\Omega)$ is differentiable.

(c) For all $(s, x, \boldsymbol{v}) \in [0, T] \times \mathbb{R}^N \times (\mathbb{R}^N)^{\#\boldsymbol{v}}$ the mapping $L^2(\Omega) \ni \theta \mapsto a_2(s, x, [\theta], \boldsymbol{v}) \in L^2(\Omega)$ is Fréchet differentiable.

(d) $a_k(s, x, [\theta], \boldsymbol{v}) \in \mathbb{D}^{1,\infty}$ for $k = 1, 2, 3$ and all $(s, x, [\theta], \boldsymbol{v}) \in [0, T] \times \mathcal{P}_2(\mathbb{R}^N) \times (\mathbb{R}^N)^{\#\boldsymbol{v}}$. Moreover, we assume the following estimates on the Malliavin derivatives hold.

$$\sup_{r\in[0,T]} \mathbb{E}\bigg[\sup_{s\in[0,T]} |\boldsymbol{D}^{\mathcal{H}}_r a_k(s, x, [\theta], \boldsymbol{v})|^p\bigg] < \infty, \quad k = 0, 1, 2, 3.$$

Then, for all $t \in [0, T]$ the following hold:

1. Under assumption (a), $x \mapsto Y^{x,[\theta]}_t(\boldsymbol{v})$ is differentiable in $L^p(\Omega)$ for all $p \geq 1$ and

$$\partial_x Y^{x,[\theta]}_t(\boldsymbol{v}) \overset{L^p}{:=} \lim_{h\to 0} \frac{1}{|h|}\Big(Y^{x+h,[\theta]}_t(\boldsymbol{v}) - Y^{x,[\theta]}_t(\boldsymbol{v})\Big),$$

where the limit is taken in L^p sense, satisfies

$$\partial_x Y_t^{x,[\theta]}(\boldsymbol{v}) = \sum_{i=0}^{d} \int_0^t \Bigg\{ \partial_x a_1^i Y_s^{x,[\theta]}(\boldsymbol{v}) + a_1^i \partial_x Y_s^{x,[\theta]}(\boldsymbol{v}) + \partial_x a_2^i + \mathbb{E}'\bigg[\partial_x a_3^i\big|_{v=\theta'} (Y_s^{\theta',[\theta]})'(\boldsymbol{v}) + \sum_{r=1}^{\#\boldsymbol{v}} \partial_x a_3^i\big|_{v=v_r} (Y_s^{\theta',[\theta]})'(\boldsymbol{v}) \bigg] \Bigg\} \mathrm{d}B_s^i;$$

2. *Under assumption (b), $\boldsymbol{v} \mapsto Y_t^{x,[\theta]}(\boldsymbol{v})$ is differentiable in $L^p(\Omega)$ for all $p \geq 1$ and*

$$\partial_{\boldsymbol{v}} Y_t^{x,[\theta]}(\boldsymbol{v}) \overset{L^p}{:=} \lim_{h \to 0} \frac{1}{|h|} \Big(Y_t^{x,[\theta]}(\boldsymbol{v} + h) - Y_t^{x,[\theta]}(\boldsymbol{v}) \Big)$$

satisfies

$$\partial_{v_j} Y_t^{x,[\theta]}(\boldsymbol{v}) = \sum_{i=0}^{d} \int_0^t \Bigg\{ a_1^i \partial_{v_j} Y_s^{x,[\theta]}(\boldsymbol{v}) + \tilde{a}_1^i \partial_{v_j} \widetilde{Y}_s(\boldsymbol{v}) + \partial_{v_j} a_2^i + \mathbb{E}'\bigg[\partial_v a_3^i\big|_{v=v_j} (Y_s^{v_j,[\theta]})'(\boldsymbol{v}) \bigg] + \mathbb{E}'\bigg[a_3^i\big|_{v=v_j} \partial_x (Y_s^{v_j,[\theta]})'(\boldsymbol{v}) + a_3^i\big|_{v=\theta'} \partial_{v_j} (Y_s^{\theta',[\theta]})'(\boldsymbol{v}) + \sum_{r=1}^{\#\boldsymbol{v}} a_3^i\big|_{v=v_r} \partial_{v_j} (Y_s^{v_r,[\theta]})'(\boldsymbol{v}) \bigg] \Bigg\} \mathrm{d}B_s^i.$$

3. *Under assumption (a), (b) and (c), the maps $\theta \mapsto Y_t^{\theta,[\theta]}(\boldsymbol{v})$ and $\theta \mapsto Y_t^{x,[\theta]}(\boldsymbol{v})$ are Fréchet differentiable for all $(x, \boldsymbol{v}) \in \mathbb{R}^N \times (\mathbb{R}^N)^{\#\boldsymbol{v}}$, so $\partial_\mu Y_t^{x,[\theta]}(\boldsymbol{v})$ exists and it satisfies*

$$\partial_\mu Y_t^{x,[\theta]}(\boldsymbol{v}, \hat{v}) = \sum_{i=0}^{d} \int_0^t \Bigg\{ \partial_\mu a_1^i Y_s^{x,[\theta]}(\boldsymbol{v}) + a_1^i \partial_\mu Y_s^{x,[\theta]}(\boldsymbol{v}, \hat{v}) + \partial_\mu \tilde{a}_1^i \widetilde{Y}_s(\boldsymbol{v}) + \partial_\mu a_2^i + a_1^i \partial_\mu \widetilde{Y}_s(\boldsymbol{v}, \hat{v}) + \mathbb{E}'\bigg[\partial_\mu a_3^i (Y_s^{\theta',[\theta]})'(\boldsymbol{v}) + \partial_v a_3^i (Y_s^{\hat{v},[\theta]})'(\boldsymbol{v}) + a_3^i\big|_{v=\theta'} \partial_\mu (Y_s^{\theta',[\theta]})'(\boldsymbol{v}, \hat{v}) \bigg] + \mathbb{E}'\bigg[a_3^i\big|_{v=\hat{v}} \partial_x (Y_s^{\hat{v},[\theta]})'(\boldsymbol{v}) + \sum_{r=1}^{\#\boldsymbol{v}} a_3^i\big|_{v=v_r} \partial_\mu (Y_s^{v_r,[\theta]})'(\boldsymbol{v}, \hat{v}) \bigg] \Bigg\} \mathrm{d}B_s^i.$$

Moreover, we have the representation, for all $\gamma \in L^2(\Omega)$,

$$\mathcal{D}\Big(Y_t^{\theta,[\theta]}(\boldsymbol{v})\Big)(\gamma) = \Big(\partial_x Y_t^{x,[\theta]}(\boldsymbol{v})\gamma + \mathbb{E}''\big[\partial_\mu Y_t^{x,[\theta]}(\boldsymbol{v},\theta'')\gamma''\big]\Big)\Big|_{x=\theta}.$$

4. *Under assumption (d), $Y_t^{x,[\theta]} \in \mathbb{D}^{1,\infty}$ and $\boldsymbol{D}_r^{\mathcal{H}} Y_t^{x,[\theta]}$ satisfies*

$$\begin{aligned}\boldsymbol{D}_r^{\mathcal{H}} Y_t^{x,[\theta]}(\boldsymbol{v}) = {} & \Big(a_1^j Y_r^{x,[\theta]} + \tilde{a}_1^j \widetilde{Y}_r + a_2^j + \mathbb{E}'\big[a_3^j (Y_s^{x,[\theta]})'(\boldsymbol{v})\big]\Big)_{j=1,\dots,d} \\ & + \sum_{i=0}^{d} \int_0^t \Big\{\boldsymbol{D}_r^{\mathcal{H}} a_1^i Y_s^{x,[\theta]}(\boldsymbol{v}) + \boldsymbol{D}_r^{\mathcal{H}} \tilde{a}_1^i \widetilde{Y}_s + a_1^i \boldsymbol{D}_r^{\mathcal{H}} Y_s^{x,[\theta]}(\boldsymbol{v}) + \tilde{a}_1^i \boldsymbol{D}_r^{\mathcal{H}} \widetilde{Y}_s \\ & + \boldsymbol{D}_r^{\mathcal{H}} a_2^i + \mathbb{E}'\big[\boldsymbol{D}_r^{\mathcal{H}} a_3^i\big|_{v=\theta'} (Y_s^{x,[\theta]})'(\boldsymbol{v})\big]\Big\}\, \mathrm{d}B_s^i.\end{aligned}$$

Moreover, the following bound holds:

$$\sup_{r\le t} \mathbb{E}\Big[|\boldsymbol{D}_r^{\mathcal{H}} Y_t^{x,[\theta]}(\boldsymbol{v})|^p\Big] \le C \sup_{r\le t} \mathbb{E}\Big[\sup_{r\le t\le T} \big(|\boldsymbol{D}_r^{\mathcal{H}} a_1|^p + |\boldsymbol{D}_r^{\mathcal{H}} \tilde{a}_1|^p\big)\Big] \tag{4.33}$$

The limits in the above are taken in L^p sense. When we say $k = 1, 2, 3$ for the assumptions, we also means $\tilde{a}_1$.

Proof: See Proposition 4.4.1 and [44, Proposition 6.8] for the proof. ■

We are now ready to prove Theorem 4.4.2.

Proof: [Proof of Theorem 4.4.2] The proof follows identically the proof of [44, Theorem 3.2], where we apply Lemma 4.4.3 and Proposition 4.4.4. ■

4.4.2 Integration by Parts Formulae

Now we introduce some operators acting on the Kusuoka-Stroock processes. These operators will be used later in the integration by parts formulae. We first make the following common assumption on uniform ellipticity.

Assumption 4.4.5 (Uniform Ellipticity) *Let $\sigma : [0,T]\times\mathbb{R}^N\times\mathcal{P}_2(\mathbb{R}^N)\times\mathbb{R}^N \to \mathbb{R}^{N\times d}$ be given by*

$$\sigma(t,z,\mu,\tilde{z}) := [V_1(t,z,\mu,\tilde{z}),\dots,V_d(t,z,\mu,\tilde{z})].$$

We assume that there exists $\epsilon > 0$ such that, for all $\xi\in\mathbb{R}^N$, $z\in\mathbb{R}^N$ and $\mu\in\mathcal{P}_2(\mathbb{R}^N)$,

$$\xi^\top\sigma(t,z,\mu,\tilde{z})\sigma(t,z,\mu,\tilde{z})^\top\xi \geq \epsilon|\xi|^2.$$

For a function $\Psi : [0,T]\times\mathbb{R}^N\times\mathcal{P}_2(\mathbb{R}^N)\to\mathbb{D}^{n,\infty}$, the following operators acting on Kusuoka-Stroock processes in $\mathbb{K}^q_r(\mathbb{R},n)$ with multi-index $\alpha=(i)$ and $(t,x,[\theta])\in[0,T]\times\mathbb{R}^N\times\mathcal{P}_2(\mathbb{R}^N)$ are given by

$$\begin{aligned}
I^1_{(i)}(\Psi)(t,x,[\theta]) &:= \frac{1}{\sqrt{t}}\delta_{\mathcal{H}}\bigg(r\mapsto\Psi(t,x,[\theta])\big(\sigma^\top\big(\sigma\sigma^\top\big)^{-1}(r,X^{x,[\theta]}_r,[X^\theta_r],\widetilde{X}_r)\partial_x X^{x,[\theta]}_r\big)_i\bigg)\\
I^2_{(i)}(\Psi)(t,x,[\theta]) &:= \sum_{j=1}^N I^1_{(j)}\bigg(\big(\partial_x X^{x,[\theta]}_t\big)^{-1}_{j,i}\Psi(t,x,[\theta])\bigg),\\
I^3_{(i)}(\Psi)(t,x,[\theta]) &:= I^1_{(i)}(\Psi)(t,x,[\theta]) + \sqrt{t}\partial^i\Psi(t,x,[\theta]),\\
\mathcal{I}^1_{(i)}(\Psi)(t,x,[\theta],v_1) &:= \frac{1}{\sqrt{t}}\delta_{\mathcal{H}}\bigg(r\mapsto\big(\sigma^\top\big(\sigma\sigma^\top\big)^{-1}(r,X^{x,[\theta]}_r,[X^\theta_r],\widetilde{X}_r)\\
&\qquad\partial_x X^{x,[\theta]}_r(\partial_x X^{x,[\theta]}_t)^{-1}\partial_\mu X^{x,[\theta]}_t(v_1)\big)_i\Psi(t,x,[\theta])\bigg),\\
\mathcal{I}^3_{(i)}(\Psi)(t,x,[\theta],v_1) &:= \mathcal{I}^1_{(i)}(\Psi)(t,x,[\theta],v_1) + \sqrt{t}(\partial_\mu\Psi)_i(t,x,[\theta],v_1).
\end{aligned}$$

For a general multi-index $\alpha=(\alpha_1,\dots,\alpha_n)$, we inductively define

$$I^1_\alpha := I^1_{\alpha_n}\circ I^1_{\alpha_{n-1}}\circ\cdots\circ I^1_{\alpha_1},$$

the definition of the other operators are analogue to I^1_α. The following Proposition follows directly from our previous discussion and the definition of the Kusuoka-Stroock process.

Proposition 4.4.6 *If $V_0, \dots, V_d \in \mathcal{C}^{k,k,k}_{b,\mathrm{Lip}}([0,T] \times \mathbb{R}^N \times \mathcal{P}_2(\mathbb{R}^N) \times \mathbb{R}^N; \mathbb{R}^N)$, Assumption 4.4.5 holds and $\Psi \in \mathbb{K}^q_r(\mathbb{R}, n)$, then $I^1_\alpha(\Psi)$ and $I^3_\alpha(\Psi)$, are all well-defined for $|\alpha| \leq (k \wedge n)$. $I^2_\alpha(\Psi), \mathcal{I}^1_\alpha(\Psi)$ and $\mathcal{I}^3_\alpha(\Psi)$ are well defined for $|\alpha| \leq n \wedge (k-2)$. Moreover,*

$$\begin{aligned} I^1_\alpha(\Psi), I^3_\alpha(\Psi) &\in \mathbb{K}^{q+2|\alpha|}_r(\mathbb{R}, (k \wedge n) - |\alpha|), \\ I^2_\alpha(\Psi) &\in \mathbb{K}^{q+3|\alpha|}_r(\mathbb{R}, [n \wedge (k-2)] - |\alpha|), \\ \mathcal{I}^1_\alpha(\Psi), \mathcal{I}^3_\alpha(\Psi) &\in \mathbb{K}^{q+4|\alpha|}_r(\mathbb{R}, [n \wedge (k-2)] - |\alpha|). \end{aligned}$$

If $\Psi \in \mathbb{K}^0_r(\mathbb{R}, n)$ and $V_0, \dots, V_d$ are uniformly bounded, then

$$\begin{aligned} I^1_\alpha(\Psi), I^3_\alpha(\Psi) &\in \mathbb{K}^0_r(\mathbb{R}, (k \wedge n) - |\alpha|), \\ I^2_\alpha(\Psi) &\in \mathbb{K}^0_r(\mathbb{R}, [n \wedge (k-2)] - |\alpha|), \\ \mathcal{I}^1_\alpha(\Psi), \mathcal{I}^3_\alpha(\Psi) &\in \mathbb{K}^0_r(\mathbb{R}, [n \wedge (k-2)] - |\alpha|). \end{aligned}$$

From now on, the Integration by Parts Formulae (IBPF) follow in the same way as [44, Sec 4.] by replacing $\boldsymbol{D}, \delta$ by $\boldsymbol{D}^{\mathcal{H}}, \delta_{\mathcal{H}}$ and using integral by parts for this partial Malliavin derivative.

Integration by parts formulae in the space variable are established in the following Proposition.

Proposition 4.4.7 (Proposition 4.1, [44]) *Let $f \in \mathcal{C}^\infty_b(\mathbb{R}^N, \mathbb{R})$ and $\Psi \in \mathbb{K}^q_r(\mathbb{R}, n)$, then*

1. *If $|\alpha| \leq n \wedge k$, then*

$$\mathbb{E}\big[\partial^\alpha_x\big(f\big(X^{x,[\theta]}_t\big)\big)\Psi(t, x, [\theta])\big] = t^{-|\alpha|/2}\mathbb{E}\big[f\big(X^{x,[\theta]}_t\big)I^1_\alpha(\Psi)(t, x, [\theta])\big].$$

2. If $|\alpha| \leq n \wedge (k-2)$, then

$$\mathbb{E}\big[(\partial^\alpha f)\big(X_t^{x,[\theta]}\big)\Psi(t,x,[\theta])\big] = t^{-|\alpha|/2}\mathbb{E}\big[f\big(X_t^{x,[\theta]}\big)I_\alpha^2(\Psi)(t,x,[\theta])\big].$$

3. If $|\alpha| \leq n \wedge k$, then

$$\partial_x^\alpha \mathbb{E}\big[f\big(X_t^{x,[\theta]}\big)\Psi(t,x,[\theta])\big] = t^{-|\alpha|/2}\mathbb{E}\big[f\big(X_t^{x,[\theta]}\big)I_\alpha^3(\Psi)(t,x,[\theta])\big].$$

4. If $|\alpha| + |\beta| \leq n \wedge (k-2)$, then

$$\partial_x^\alpha \mathbb{E}\big[(\partial^\beta f)\big(X_t^{x,[\theta]}\big)\Psi(t,x,[\theta])\big] = t^{-(|\alpha|+|\beta|)/2}\mathbb{E}\big[f\big(X_t^{x,[\theta]}\big)I_\alpha^3\big((I_\beta^2\Psi)\big)(t,x,[\theta])\big].$$

Proof: We will start with proving the first result, and use the it to prove the rest.

1. First, we note that Equation (4.21) satisfied by $\partial_x X_t^{x,[\theta]}$ and Equation (4.26) satisfied by $\boldsymbol{D}_r^{\mathcal{H}} X_t^{x,[\theta]}$ are the same except their initial condition. It therefore follows from our discussion of partial Malliavin derivative that

$$\partial_x X_t^{x,[\theta]} = \boldsymbol{D}_r^{\mathcal{H}} X_t^{x,[\theta]} \sigma^\top \big(\sigma\sigma^\top\big)^{-1}(r, X_r^{x,[\theta]}, [X_r^\theta], \widetilde{X}_r)\partial_x X_r^{x,[\theta]}. \tag{4.34}$$

Let us start with the assumption of $|\alpha| = 1$, and the general desired result can be obtained by repeat the following procedures iteratively. We are then allowed to compute the followings for $f \in \mathcal{C}_b^\infty(\mathbb{R}^N, \mathbb{R})$,

$$\begin{aligned}
\mathbb{E}\big[\partial_x\big(f\big(X_t^{x,[\theta]}\big)\big)\Psi(t,x,[\theta])\big] &= \mathbb{E}\big[\partial f\big(X_t^{x,[\theta]}\big)\partial_x X_t^{x,[\theta]}\Psi(t,x,[\theta])\big] \\
&= \frac{1}{t}\mathbb{E}\bigg[\int_0^t \partial f\big(X_t^{x,[\theta]}\big)\partial_x X_t^{x,[\theta]}\Psi(t,x,[\theta])\,\mathrm{d}r\bigg] \\
&= \frac{1}{t}\mathbb{E}\bigg[\int_0^t \partial f\big(X_t^{x,[\theta]}\big)\boldsymbol{D}_r^{\mathcal{H}} X_t^{x,[\theta]}\sigma^\top\big(\sigma\sigma^\top\big)^{-1}(r, X_r^{x,[\theta]}, [X_r^\theta], \widetilde{X}_r) \\
&\qquad\qquad \times \partial_x X_r^{x,[\theta]}\Psi(t,x,[\theta])\,\mathrm{d}r\bigg] \\
&= \frac{1}{t}\mathbb{E}\bigg[\int_0^t \boldsymbol{D}_r^{\mathcal{H}} f\big(X_t^{x,[\theta]}\big)\sigma^\top\big(\sigma\sigma^\top\big)^{-1}(r, X_r^{x,[\theta]}, [X_r^\theta], \widetilde{X}_r) \\
&\qquad\qquad \times \partial_x X_r^{x,[\theta]}\Psi(t,x,[\theta])\,\mathrm{d}r\bigg] \\
&= \frac{1}{t}\mathbb{E}\bigg[f\big(X_t^{x,[\theta]}\big)\delta_{\mathcal{H}}\bigg(r \mapsto \Psi(t,x,[\theta]) \\
&\qquad\qquad \times \big(\sigma^\top\big(\sigma\sigma^\top\big)^{-1}(r, X_r^{x,[\theta]}, [X_r^\theta], \widetilde{X}_r)\partial_x X_r^{x,[\theta]}\big)\bigg)\bigg],
\end{aligned}$$

where we have applied partial Malliavin calculus integration by parts from Equation (4.20) in the last equality. This proves the result for $|\alpha| = 1$. By Proposition 4.4.6, $I_\alpha^1(\Psi) \in \mathbb{K}_r^{q+2}(\mathbb{R}, (k\wedge n)-1)$ when $|\alpha| = 1$. We can then repeat the above procedures iteratively to get to desired result.

2. By the chain rule,

$$\begin{aligned}
\mathbb{E}\big[(\partial^i f)\big(X_t^{x,[\theta]}\big)\Psi(t,x,[\theta])\big] &= \sum_{j=1}^N \mathbb{E}\bigg[\partial_{x_i}\big(f\big(X_t^{x,[\theta]}\big)\big)\bigg(\big(X_t^{x,[\theta]}\big)^{-1}\bigg)^{j,i}\Psi(t,x,[\theta])\bigg] \\
&= t^{-1/2}\sum_{j=1}^N \mathbb{E}\bigg[f\big(X_t^{x,[\theta]}\big)I_{(j)}^1\bigg(\bigg(\big(X_t^{x,[\theta]}\big)^{-1}\bigg)^{j,i}\Psi(t,x,[\theta])\bigg)\bigg] \\
&= t^{1/2}\mathbb{E}\big[f\big(X_t^{x,[\theta]}\big)I_{(i)}^2(\Psi)(t,x,[\theta])\big],
\end{aligned}$$

where we apply the result in part 1 to the second equality. From Proposition 4.4.6,

$I^2_{(i)}(\Psi) \in \mathbb{K}^{q+3}_r(\mathbb{R}, (n \wedge (k-2)) - 1)$, so since $|\alpha| \leq (n \wedge (k-2))$, the proof follows from applying the same arguments for another $|\alpha| - 1$ times.

3. By part 1 and direct computation,

$$\begin{aligned}\partial^i_x \mathbb{E}\big[f\big(X^{x,[\theta]}_t\big)\Psi(t,x,[\theta])\big] &= \mathbb{E}\big[\partial^i_x f\big(X^{x,[\theta]}_t\big)\Psi(t,x,[\theta]) + f\big(X^{x,[\theta]}_t\big)\partial^i_x \Psi(t,x,[\theta])\big] \\ &= t^{-1/2}\mathbb{E}\Big[f\big(X^{x,[\theta]}_t\big)\big\{I^1_i(\Psi)(t,x,[\theta]) + \sqrt{t}\partial^i_x \Psi(t,x,[\theta])\big\}\Big],\end{aligned}$$

which proves the result for $|\alpha| = 1$. Again, we have $I^3_\alpha(\Psi) \in \mathbb{K}^{q+2}_r(\mathbb{R}, (k \wedge n) - 1)$ when $|\alpha| = 1$. Then the proof follows from iterative implementation of the above procedure.

4. This part follows from parts 2 and 3 directly.

■

Similar to the integration by parts in space variable, we can also derive integration by parts in the measure variable as follows.

Proposition 4.4.8 (Proposition 4.2, [44]) *Let $f \in \mathcal{C}^\infty_b(\mathbb{R}^N, \mathbb{R})$ and $\Psi \in \mathbb{K}^q_r(\mathbb{R}, n)$, then*

1. *If $|\beta| \leq n \wedge (k-2)$, then*

$$\mathbb{E}\big[\partial^\beta_\mu\big(f\big(X^{x,[\theta]}_t\big)\big)(\boldsymbol{v})\Psi(t,x,[\theta])\big] = t^{-|\beta|/2}\mathbb{E}\big[f\big(X^{x,[\theta]}_t\big)\mathcal{I}^1_\beta(\Psi)(t,x,[\theta],\boldsymbol{v})\big].$$

2. *If $|\beta| \leq n \wedge (k-2)$, then*

$$\partial^\beta_\mu \mathbb{E}\big[f\big(X^{x,[\theta]}_t\big)\Psi(t,x,[\theta])\big](\boldsymbol{v}) = t^{-|\beta|/2}\mathbb{E}\big[f\big(X^{x,[\theta]}_t\big)\mathcal{I}^3_\beta(\Psi)(t,x,[\theta],\boldsymbol{v})\big].$$

3. If $|\alpha| + |\beta| \leq n \wedge (k-2)$*, then*

$$\partial_\mu^\beta \mathbb{E}\big[(\partial^\alpha f)\big(X_t^{x,[\theta]}\big)\Psi(t,x,[\theta])\big](\boldsymbol{v}) = t^{-(|\alpha|+|\beta|)/2}\mathbb{E}\Big[f\big(X_t^{x,[\theta]}\big)\mathcal{I}_\beta^3\big(I_\alpha^2(\Psi)\big)(t,x,[\theta],\boldsymbol{v})\Big].$$

Proof: The proofs use the same idea as Proposition 4.4.7 and the Equation (4.34).

■

We now consider the integration by parts formulae for the derivatives of the mapping:

$$x \mapsto \mathbb{E}[f(X_t^{x,\delta_x})].$$

Let us introduce the following operator acting on $\mathbb{K}_r^q(\mathbb{R}, M)$, the set of the Kusuoka-Stroock processes do not depend on measure μ. For $\alpha = (i)$,

$$J_{(i)}(\Phi)(t,x) := I_{(i)}^3(\Phi)(t,x,\delta_x) + \mathcal{I}_{(i)}^3(\Phi)(t,x,\delta_x)$$

and for $\alpha = (\alpha_1, \ldots, \alpha_n)$, $J_\alpha(\Phi) := J_{\alpha_n} \circ \cdots \circ J_{\alpha_1}(\Phi)$.

Theorem 4.4.9 *Let* $f \in \mathcal{C}_b^\infty(\mathbb{R}^N;\mathbb{R})$*. For all multi-indices* α *on* $\{1,\ldots,N\}$ *with* $|\alpha| \leq k-2$*,*

$$\partial_x^\alpha \mathbb{E}\big[f\big(X_t^{x,\delta_x}\big)\big] = t^{-|\alpha|/2}\mathbb{E}\big[f\big(X_t^{x,\delta_x}\big)J_\alpha(1)(t,x)\big].$$

In particular, we get the following bound,

$$\big|\partial_x^\alpha \mathbb{E}\big[f\big(X_t^{x,\delta_x}\big)\big]\big| \leq C\|f\|_\infty t^{-|\alpha|/2}(1+|x|)^{4|\alpha|}$$

Proof: Since δ_x depends on x, we have

$$\partial_x^i \mathbb{E}\big[f\big(X_t^{x,\delta_x}\big)\big] = \partial_z^i \mathbb{E}\big[f\big(X_t^{x,\delta_x}\big)\big]\big|_{z=x} + \partial_\mu^i \mathbb{E}\big[f\big(X_t^{x,[\theta]}\big)\big](v)\big|_{[\theta]=\delta_x, v=x},$$

then for $|\alpha| = 1$ the result yields by Proposition 4.4.7 and 4.4.8. The proof is completed by repeating this procedure for another $|\alpha| - 1$ times. ■

The following Corollary is useful for the smoothness of densities of directed chain SDEs.

Corollary 4.4.10 *Let $f \in \mathcal{C}_b^\infty(\mathbb{R}^N; \mathbb{R})$, α and β are multi-indices on $\{1, \dots, N\}$ with $|\alpha| + |\beta| \leq k - 2$. Then,*

$$\partial_x^\alpha \mathbb{E}\big[(\partial^\beta f)\big(X_t^{x,\delta_x}\big)\big] = t^{-\frac{|\alpha|+|\beta|}{2}} \mathbb{E}\big[f\big(X_t^{x,\delta_x}\big) I_\beta^2(J_\alpha(1))(t,x)\big]$$

and $I_\beta^2(J_\alpha(1)) \in \mathbb{K}_0^{4|\alpha|+3|\beta|}(\mathbb{R}, k - 2 - |\alpha| - |\beta|)$.

Proof: The proof follows from Theorem 4.4.9 and Proposition 4.4.7. ■

4.4.3 Smooth Densities

We are now ready to prove the main theorem of this section.

Theorem 4.4.11 *We assume Assumption 4.4.5 holds and $V_0, \dots, V_d \in \mathcal{C}_{b,\mathrm{Lip}}^{k,k,k}([0,T] \times \mathbb{R}^N \times \mathcal{P}_2(\mathbb{R}^N) \times \mathbb{R}^N; \mathbb{R}^N)$. Let α, β be multi-indices on $\{1, \dots, N\}$ and let $k \geq |\alpha| + |\beta| + N + 2$. Assume also the initial state for directed chain SDE is $\theta \equiv x$, i.e. $[\theta] = \delta_x$. Then the directed chain SDE (4.3) coincides with the alternative SDE (4.13). For all $t \in [0,T]$, X_t^{x,δ_x} has a density $p(t,x,\cdot)$ such that $(x,z) \mapsto \partial_x^\alpha \partial_z^\beta p(t,x,z)$ exists and is continuous. Moreover, there exists a constant C which depends on T, N and bounds on the coefficients, such that for all $t \in (0,T]$*

$$|\partial_x^\alpha \partial_z^\beta p(t,x,z)| \leq C(1+|x|)^{4|\alpha|+3|\beta|+3N} t^{-\frac{1}{2}(N+|\alpha|+|\beta|)}, \quad x \in \mathbb{R}^N,\ z \in \mathbb{R}^N. \qquad (4.35)$$

If $V_0, \dots, V_d$ are bounded, then the following estimate holds, for all $t \in (0, T]$

$$|\partial_x^\alpha \partial_z^\beta p(t,x,z)| \le C t^{-\frac{1}{2}(N+|\alpha|+|\beta|)} \exp\big(-C \frac{|z-x|^2}{t}\big), \quad x \in \mathbb{R}^N,\, z \in \mathbb{R}^N.$$

Proof: The proof is verbatim to Theorem 6.1 of [44] by applying our integration by parts formulae established in Corollary 4.4.10 and Lemma 3.1 in [140]. ■

Theorem 4.4.11 presents the smoothness result for X_t^{x,δ_x} and it can be generalized to X_t^θ with an general initial distribution $[\theta]$.

Corollary 4.4.12 *Suppose Assumption 4.4.5 holds and $V_0, \dots, V_d \in \mathcal{C}_{b,\mathrm{Lip}}^{k,k,k}([0,T] \times \mathbb{R}^N \times \mathcal{P}_2(\mathbb{R}^N) \times \mathbb{R}^N; \mathbb{R}^N)$. Let θ be a random variable in $\mathbb{R}^N$ with finite moments of all orders. For any multi-index β on $\{1, \dots, N\}$ such that $k \ge |\beta| + N + 2$, we have that for all $t \in [0,T]$, X_t^θ has a density $p_\theta(t, \cdot)$ such that $z \mapsto \partial_z^\beta p_\theta(t,z)$ exists and is continuous.*

Proof: The proof is done by taking expectation on both sides of the inequality (4.35) with respect to the initial distribution θ and applying dominated convergence theorem, where we use the assumption that θ has finite moments. ■

The above existence and smoothness results on the marginal density of a single object can be extended to the joint distribution for any number of adjacent particles. Namely, for a fixed integer $m \ge 1$, we may construct the system of stochastic processes $(\widetilde{X}_\cdot^0, \widetilde{X}_\cdot^1, \widetilde{X}_\cdot^2, \dots, \widetilde{X}_\cdot^m)$ such that $(\widetilde{X}_\cdot^{m-1}, \widetilde{X}_\cdot^m) \equiv (X_\cdot^\theta, \widetilde{X}_\cdot)$ in (4.1), and $\widetilde{X}_\cdot^i$ depends on the adjacent process $\widetilde{X}_\cdot^{i+1}$ and Brownian motion $\widetilde{B}_\cdot^i$, independent of $\widetilde{X}^{i+1}$, in the same fashion as of $(X_\cdot^\theta, \widetilde{X}_\cdot)$ in (4.1) for $i = 0, \dots, m-1$.

Corollary 4.4.13 *Suppose Assumption 4.4.5 holds and $V_0, \dots, V_d \in \mathcal{C}_{b,\mathrm{Lip}}^{k,k,k}([0,T] \times \mathbb{R}^N \times \mathcal{P}_2(\mathbb{R}^N) \times \mathbb{R}^N; \mathbb{R}^N)$ and θ has finite moments. Then the joint density of the process $(X_\cdot^\theta, \widetilde{X}_\cdot^1, \widetilde{X}_\cdot^2, \dots, \widetilde{X}_\cdot^m)$ exists and is continuous at any $t \in [0,T]$, where $\widetilde{X}_\cdot^1 \equiv \widetilde{X}_\cdot$ and $\widetilde{X}_\cdot^i$ depends on $\widetilde{X}_\cdot^{i+1}$ in the same fashion as of $(X_\cdot^\theta, \widetilde{X}_\cdot)$ in (4.1).*

Proof: We consider the process evolving in space $\mathbb{R}^{(m+1)N}$ defined by

$$Y_{\cdot} := (\widetilde{X}^0_{\cdot}, \widetilde{X}^1_{\cdot}, \widetilde{X}^2_{\cdot}, \dots, \widetilde{X}^m_{\cdot})$$

and the neighborhood process $\widetilde{Y}_{\cdot} := (\widetilde{X}^{m+1}_{\cdot}, \widetilde{X}^{m+2}_{\cdot}, \dots, \widetilde{X}^{2m+1}_{\cdot})$. Now $(Y_{\cdot}, \widetilde{Y}_{\cdot})$ satisfies the directed chain structure and it can be proved that this new directed chain SDE structure $Y_{\cdot}$ also satisfies Assumption 4.4.5. Hence the existence and continuity follow from Theorem 4.4.11 and Corollary 4.4.12. In particular, if $m = 1$, the coupled process $Y_{\cdot}$ is defined by

$$Y_t = Y_0 + \sum_{i=1}^{2d} \int_0^T V_i^y(s, Y_s, \mathrm{Law}(Y_s), \widetilde{Y}_s)\, \mathrm{d}B_s^{y,i},$$

where the diffusion coefficients V_i^y, $i = 1, \dots, 2d$ are given by

$$V_i^y := \begin{cases} \big(V_i(s, X_s, \mathrm{Law}(X_s), \widetilde{X}_s^1), \mathbf{0}\big)^T \in \mathbb{R}^{2N}, & i = 1, \dots d, \\ \big(\mathbf{0}, V_{i-d}(s, \widetilde{X}_s^1, \mathrm{Law}(\widetilde{X}_s^1), \widetilde{X}_s^2)\big)^T \in \mathbb{R}^{2N}, & i = d+1, \dots 2d, \end{cases}$$

B^y is independent standard Brownian motions in $\mathbb{R}^{2d}$ and $\mathbf{0} \in \mathbb{R}^N$ is a zero vector. ■

4.4.4 Markov Random Fields

The existence of density in Theorem 4.4.11 is closely related to the local Markov property (or Markov random fields) of the directed chain structure. Here, we shall elaborate the relation briefly. A similar topic has been studied by [100] on the undirected graph with locally interactions only on the drift terms. Their approach is to apply a change of measure under which the diffusion coefficients at one vertex of the undirected graph do not depend on the diffusions at the other vertexes of the graph, in order to get the factorization of the probability measure. Usually, this Markov property is only discussed for the undirected graph or directed acyclic graph. The finite particle system

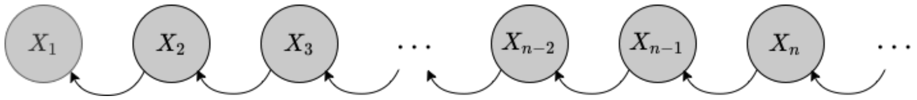


Figure 4.2: This figure shows a finite cut of the infinite directed chain, i.e., X_k is affected by X_{k+1}.

that approximates the directed chain structure discussed in [52] admits a loop structure in the finite graph. More precisely, the finite system of n particles $(X_{1,\cdot}^{(n)}, \ldots, X_{n,\cdot}^{(n)})$ is constructed in a loop of size n so that $X_{1,\cdot}^{(n)}$ depends on $X_{2,\cdot}^{(n)}$, $X_{2,\cdot}^{(n)}$ depends on $X_{3,\cdot}^{(n)}$, $\ldots$, $X_{n-1,\cdot}^{(n)}$ depends on $X_{n,\cdot}^{(n)}$ and $X_{n,\cdot}^{(n)}$ depends on $X_{1,\cdot}$. However, when the size n of this loop is forced to be infinity, i.e., $n \to \infty$, we can then treat the dependence of the system on any finite subgraph as the system on an acyclic graph [52, Section 3], as (4.2) in our paper. An illustration is given in Figure 4.2.

Proposition 4.4.14 *The directed chain SDEs described in* (4.2) *form a first order Markov random fields, or we say it has the local Markov property.*

We follow the notations and terminology in [107]. Given a directed graph $G = (V, E)$ with vertexes V and edges E, for a vertex $\nu \in V$, let $\mathcal{X}_\nu$ denote the generic space of vertex ν and $\mathrm{pa}(\nu) \in V$ denote all of its parents. In the infinite directed chain case, $\mathrm{pa}(X_{k,\cdot}) = X_{k+1,\cdot}$.

Definition 4.4.15 (Recursive Factorization) *Given a directed graph $G = (V, E)$, we say the probability distribution P^G admits a recursive factorization according to G, if there exists non-negative functions, henceforth referred to as kernels, $k^\nu(\cdot,\cdot), \nu \in V$ defined on $\mathcal{X}_\nu \times \mathcal{X}_{\mathrm{pa}(\nu)}$, such that*

$$\int k^\nu(y_\nu, x_{\mathrm{pa}(\nu)})\mu_\nu(\,\mathrm{d}y_\nu) = 1$$

and P^G has density f^G with respect to a product measure μ, which is defined on the *product space* $\prod_{\nu \in V} \mathcal{X}_\nu$ *by* μ_ν *a measure defined on each* $\mathcal{X}_\nu$*, where*

$$f^G(x) = \prod_{\nu \in V} k^\nu(x_\nu, x_{\mathrm{pa}(\nu)}).$$

Proof: [Proof of Proposition 4.4.14] Thanks to the special chain-like structure, it can be shown that the distribution of the chain satisfies the *recursive factorization* property, where the existence and continuity of the kernel functions are given by Theorem 4.4.11 and Corollary 4.4.12. For it, on a filtered probability space, let us consider a system of the directed chain diffusion $X_{i,t}$, $i \in \mathbb{N}$, $t \geq 0$ on the infinitely graph with the vertexes $\mathbb{N} = \{1, 2, \ldots\}$. Firstly, the coupled diffusion $(X_{1,\cdot}, X_{2,\cdot}) \equiv (X^\theta_\cdot, \widetilde{X}_\cdot)$ satisfy the directed chain stochastic equation and have a continuous density by Corollary 4.4.13 and we denote this joint density by $g(\cdot,\cdot) : \mathbb{R}^N \times \mathbb{R}^N \to \mathbb{R}$. We then build the chain recursively by the following rule: given $X_{k,\cdot}$, initial state $X_{k+1,0}$ and Brownian motion $B_{k+1,\cdot}$ independent of $(X_{1,\cdot}, \ldots, X_{k,\cdot}, X_{k+1,0})$, we construct $X_{k+1,\cdot}$ according to the distribution of $(X^\theta_\cdot, \widetilde{X}_\cdot)$.

Defining the kernel functions in the following way

$$k^\nu(x_\nu, x_{\mathrm{pa}(\nu)}) := \begin{cases} g(x_\nu, x_{\mathrm{pa}(\nu)}), & \text{if } \nu = X_1, \\ (g(x_\nu))^{-1} g(x_\nu, x_{\mathrm{pa}(\nu)}), & \text{if } \nu = X_k,\ k \geq 2, \end{cases}$$

i.e., the conditional density of $X_{k+1,t}$ given $X_{k,t}$ for $k \geq 2$, proves the *recursive factorization* property of the chain on any finite cut $(X_{1,t}, X_{2,t}, \ldots, X_{m,t})$, $\forall m \in \mathbb{N}$ of the infinite chain for any $t \in [0,T]$, as well as the local Markov property following from [107, Theorem 3.27], which is also called the first order Markov random field in the context of [100]. This result can also be verified by a filtering problem build upon this directed chains structure that we omit due to the page limitation. ■

4.4.5 Relation to PDE

We have constructed the integration by parts formulae to argue that the density of directed chain SDEs is smooth in section 4.4.2, which is also the tool for constructing solutions to a related PDE problem. To ease notation, we will omit the time dependency in coefficients of SDEs through this section, i.e. we will write $V(X_t^{x,[\theta]}, [X_t^\theta], \widetilde{X}_t) := V(t, X_t^{x,[\theta]}, [X_t^\theta], \widetilde{X}_t)$. In particular, we are interested in the function

$$U(t, x, [\theta]) = \mathbb{E}[g(X_t^{x,[\theta]}, [X_t^\theta])]$$

, $t \in [0, T]$, $x \in \mathbb{R}^N$ for some sufficiently smooth function g. Here $X_\cdot^\theta$ is the solution of (4.3)-(4.4) with random initial θ and $X_t^{x,[\theta]}$ is the solution to (4.13) with deterministic initial x. They depend on a neighborhood process $\widetilde{X}_\cdot$ with an independent initial random vector $\tilde{\theta}$. Recall the flow property (4.16) in section 4.2.3. It follows that for every $0 \le t \le t + h \le T$, $x \in \mathbb{R}^d$,

$$U(t+h, x, [\theta]) = \mathbb{E}[g(X_{t+h}^{x,[\theta]}, [X_{t+h}^\theta])] = \mathbb{E}\big[U(t, X_h^{x,[\theta]}, [X_h^\theta])\big].$$

Hence

$$\begin{aligned} U(t+h, x, [\theta]) - U(t, x, [\theta]) = U(t, x, [X_h^\theta]) - U(t, x, [\theta]) + \\ \mathbb{E}\big[U(t, X_h^{x,[\theta]}, [X_h^\theta]) - U(t, x, [X_h^\theta])\big] \\ = I - \mathbb{E}[J], \end{aligned} \tag{4.36}$$

where we define $I = U(t, x, [X_h^\theta]) - U(t, x, [\theta])$ and $J = U(t, X_h^{x,[\theta]}, [X_h^\theta]) - U(t, x, [X_h^\theta])$. Applying the chain rule introduced in [37] to I and Ito's formula to J, we have

$$\begin{aligned} I = \int_0^h \mathbb{E}\Big[& \sum_{i=1}^N V_0^i(X_r^\theta, [X_r^\theta], \widetilde{X}_r) \partial_\mu U(t, x, [X_r^\theta], X_r^\theta)_i \\ & + \frac{1}{2} \sum_{i,j=1}^N [\sigma\sigma^\top(X_r^\theta, [X_r^\theta], \widetilde{X}_r)]_{i,j} \partial_{v_j} \partial_\mu U(t, x, [X_r^\theta], X_r^\theta)_i \Big] \,\mathrm{d}r, \\ J = \int_0^h & \sum_{i=1}^N V_0^i(X_r^{x,[\theta]}, [X_r^\theta], \widetilde{X}_r) \partial_{x_i} U(t, X_r^{x,[\theta]}, [X_h^\theta]) \,\mathrm{d}r \\ & + \frac{1}{2} \int_0^h \sum_{i,j=1}^N [\sigma\sigma^\top(X_r^{x,[\theta]}, [X_r^\theta], \widetilde{X}_r)]_{i,j} \partial_{x_i} \partial_{x_j} U(t, X_r^{x,[\theta]}, [X_h^\theta]) \,\mathrm{d}r \\ & + \int_0^h \sum_{j=1}^d \sum_{i=1}^N V_j^i(X_r^{x,[\theta]}, [X_r^\theta], \widetilde{X}_r) \partial_{x_i} U(t, X_r^{x,[\theta]}, [X_h^\theta]) \,\mathrm{d}B_r^j. \end{aligned}$$

For the meaning of the differential operator with respect to measure ∂_μ appeared in I, we refer to section 4.2.1. Then let us plug I, J into (4.36) and take expectation, divide by h on both sides, and send h to 0, we will end up with a PDE of the form given below

$$\begin{aligned} (\partial_t - \mathcal{L}) U(t, x, [\theta]) = 0 \quad & \text{for } (t, x, [\theta]) \in (0, T] \times \mathbb{R}^N \times \mathcal{P}_2(\mathbb{R}^N), \\ U(0, x, [\theta]) = g(x, [\theta]) \quad & \text{for } (x, [\theta]) \in \mathbb{R}^N \times \mathcal{P}_2(\mathbb{R}^N), \end{aligned} \tag{4.37}$$

where $g : \mathbb{R}^N \times \mathcal{P}_2(\mathbb{R}^N) \to \mathbb{R}$ and the operator $\mathcal{L}$ acts on smooth enough functions $F : \mathbb{R}^N \times \mathcal{P}_2(\mathbb{R}^N) \to \mathbb{R}^N$ defined by

$$\begin{aligned} \mathcal{L}F(x, [\theta]) = & \mathbb{E}\Big[\sum_{i=1}^N V_0^i(x, [\theta], \tilde{\theta}) \partial_{x_i} F(x, [\theta]) + \frac{1}{2} \sum_{i,j=1}^N [\sigma\sigma^\top(x, [\theta], \tilde{\theta})]_{i,j} \partial_{x_i} \partial_{x_j} F(x, [\theta]) \Big] \\ & + \mathbb{E}\Big[\sum_{i=1}^N V_0^i(\theta, [\theta], \tilde{\theta}) \partial_\mu F(x, [\theta], \theta)_i + \frac{1}{2} \sum_{i,j=1}^N [\sigma\sigma^\top(\theta, [\theta], \tilde{\theta})]_{i,j} \partial_{v_j} \partial_\mu F(x, [\theta], \theta)_i \Big]. \end{aligned} \tag{4.38}$$

The expectation in the first line of (4.38) is taken with respect to the random variable $\tilde{\theta}$ due to the appearance of the neighborhood process in the difference J, while the the expectation in the second line is taken with respect to the joint distribution of $\theta, \tilde{\theta}$, as an application of the chain rule introduced in [37] to the difference I.

It is evidently that a proper condition for the initial g is needed for the existence of the solution to PDE (4.37). Such a directed chain type SDE has not been considered before, the closest work is related to the PDE associated with the McKean-Vlasov type SDE. In [20], g is assumed to have bounded second order derivatives. The smoothness on g is relaxed in [44]. In particular, they assume g belongs to a class of functions that can be approximated by a sequence of functions with polynomial growth, and also satisfy certain growth condition on its derivatives. Hence, they claim that g is not necessarily differentiable. We shall emphasize that detailed discussion on the choice of assumptions in g is beyond the scope of this paper, but we conjecture that some similar results should also hold for our case and will include this in our future research.

4.5 Neural DC-SDEs as Generator for Time Series

Under the general setup, DC-SDEs can be of McKean-Vlasov type where the coefficients have distributions as inputs, corresponding to the n-coupled system having mean-field interaction. In our proposed generator, it is sufficient to use the simple case mentioned above, DC-SDE without the mean-field interaction, as in the following restated definition.

Definition 4.5.1 (DC-SDEs, simple version) *Fix a finite time horizon* $[0, T]$ *and a filtered probability space* $(\Omega, \mathcal{F}, (\mathcal{F}_t)_{t\geq 0}, \mathbb{P})$*. Let* $(X, \tilde{X})$ *with* $X, \tilde{X} \in L^2(\Omega \times [0, T], \mathbb{R}^N)$ *be* *a pair of square-integrable stochastic processes satisfying*

$$X_t = \xi + \int_0^t V_0(s, X_s, \tilde{X}_s)\,\mathrm{d}s + \int_0^t V_1(s, X_s, \tilde{X}_s)\,\mathrm{d}B_s, \tag{4.39}$$

for $t \in [0, T]$, *with the distributional constraint*

$$\mathrm{Law}(X_t, 0 \le t \le T) = \mathrm{Law}(\tilde{X}_t, 0 \le t \le T), \tag{4.40}$$

where Law(·) stands for the distribution, $V_0 \in \mathbb{R}^N$ *and* $V_1 \in \mathbb{R}^{N\times d}$ *are smooth coefficients satisfying Lipschitz and linear growth conditions,* B *is a standard* d*-dimensional Brownian motion, and* $X_0 := \xi$, $\tilde{X}$ *and* B *are assumed to be independent.*

With the smoothness of the solution under certain additional conditions posed on the coefficients (cf. [83]), we can derive a partial differential equation (PDE) for the marginal densities of the solution. Then, the associated PDEs lead to the following theorem: DC-SDEs have at least the same amount of flexibility as Neural SDEs.

Theorem 4.5.2 *Under proper assumptions, for any* Y *that satisfies a system of Markovian SDEs on* $[0, T]$, *there exists a unique solution to the DC-SDE* (4.39) *with constraints* (4.40), *some* V_0 *and non-degenerate coefficients* V_1, *such that they have the same marginal distributions for all* $t \in [0, T]$. *Here by degenerate, we mean that* $V_i(t, x, \tilde{x}) := V_i(t, x)$, $i \in \{0, 1\}$, *i.e., the coefficients have no dependence on neighborhood nodes at all.*

We defer the proof of Theorem 4.5.2 to Appendix B.1.1.

Naturally, if V_0 and V_1 are known (or learned from data), one can take real data paths as $\tilde{X}$ in (4.39) and straightforwardly generate paths of X that have the same distribution as $\tilde{X}$ by the constraint (4.40). However, naively implementing this idea will lead to the following potential problems.

Problem 4.5.3 (Lack of Independence) *The distribution of the generated sequence crucially depends on the real data; Consequently, to avoid dependence, a single real path can only be used once as $\tilde{X}$ to generate one path of X, and thus the number of the generated sequence has to be the same as that of the training data set in one run.*

Note that a qualified generator should also be able to generate *unlimited* independent data that does not depend on the original one. Fortunately, both problems mentioned above can be overcome by the idea behind the following theorem.

Theorem 4.5.4 *Under mild non-degeneracy conditions, the correlation between training data and generated data in DC-SDEs decays exponentially fast, as the distance increases on the chain.*

For reading consistency, we give the formal statement of Theorem 4.5.4 with detailed proof in Appendix B.1.2.

We shall explain how to beat the independence problem during the implementation described in Section 4.5.1. As shown in Appendix B.1.2, the introduction of independent Brownian motions to (4.39) is the key to solving the independence problem. We shall also provide an extreme example (cf. Remark B.1.8) showing that without $\int V_1 \, \mathrm{d}B$, the system (4.39)–(4.40) has only trivial (deterministic) solution.

As the adversarial part of GAN, signature induces another powerful tool to characterize the distribution of random processes: the *expected signatures.* It was proved by [40] that expected signatures characterize the distribution of random processes uniquely, i.e., if $\mathbb{E}[S(x)] = \mathbb{E}[S(y)]$ and $\mathbb{E}[S(x)]$ has an infinite radius of convergence, then x and y have the same distribution.

4.5.1 Proposed Method: DC-GANs

In this section, we describe DC-GANs for generating multimodal distributed time series. Our method builds on the DC-SDEs with a straightforward idea: To find the (sub-) optimal solution of the generator, we implement a GAN model with the Neural DC-SDEs as the generator. For the discriminator, we use Neural CDEs [87] and Sig-Wasserstein GAN [126, 125].

Generator

To overcome the independence issue explained in Problem 4.5.3, we design DC-GANs by two phases: 1) training and 2) decorrelating and branching. The second phase will be utilized during testing. Both V_0 and V_1 in (4.39) will be parameterized by multi-layers fully connected NNs.

Training Phase. We set aside the independence problem and focus on finding the optimal coefficients V_0 and V_1 (together with the discriminator). Denote the training data by $\{\tilde{X}(\omega_i)\}_{i=1}^M$, where each ω_i represents a realization of the randomness in the path space. We treat our training data $\{\tilde{X}(\omega_i)\}_{i=1}^M$ as the neighborhood process $\tilde{X}$ in (4.39). For each training path data $\tilde{X}(\omega_i)$, we generate a DC-SDE path $X(\omega_i)$, according to the Euler scheme of (4.39),

$$\begin{aligned}
&X_{t_{j+1}}(\omega_i)\\
&= X_{t_j}(\omega_i) + V_0(t_j, X_{t_j}(\omega_i), \tilde{X}_{t_j}(\omega_i))(t_{j+1} - t_j)\\
&\quad + V_1(t_j, X_{t_j}(\omega_i), \tilde{X}_{t_j}(\omega_i))(B_{t_{j+1}}(\omega_i) - B_j(\omega_i)),
\end{aligned} \tag{4.41}$$

where $0 = t_0 \leq t_1 \leq \ldots \leq t_J = T$ is a partition on $[0, T]$, $\{B(\omega_i)\}_{i=1}^M$ are independent Brownian paths.

Both the generated paths $\{X(\omega_i)\}_{i=1}^M$ and the training paths $\{\tilde{X}(\omega_i)\}_{i=1}^M$ will be passed into the discriminator, where their Wasserstein distance needs to be minimized. To simplify the notations for later use, we define $G_\theta : (\xi, B, \tilde{X}) \mapsto X$ to represent the overall transformation in (4.41), with θ denoting all network parameters of V_0 and V_1.

Decorrelating and Branching Phase. During testing, we utilize a branching scheme to alleviate the independence problem; see Figure 4.3 for an illustrative example. Let q be the number of steps we "walk" along the directed chain. Here "walking" along the chain means: After we have finished the training (identified V_0 and V_1) phase, we start with the first chain (the grey one in Figure 4.3). We take real data as the first neighborhood X_1 to generate X_2 through the scheme (4.41), where X_1 takes the role of $\tilde{X}$ and X_2 takes the role of X. Then we use X_2 as the neighborhood to generate X_3, and repeat this procedure until we obtain X_q. By Theorem 4.5.4, X_q and X_1 are asymptotically uncorrelated, as $q \to \infty$. We describe the pseudo-code in Algorithm 2 below for this decorrelating step.

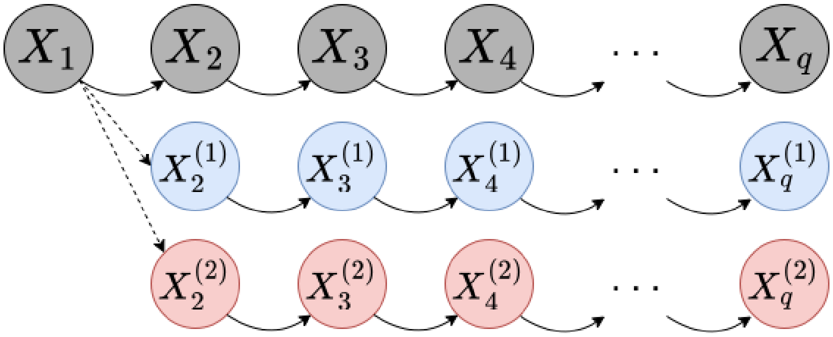


Figure 4.3: Branching Scheme. Let q be the number of steps we "walk" along the directed chain. We take real data as the first neighborhood X_1 to generate X_2 through the scheme (4.41), where X_1 takes the role of $\tilde{X}$ and X_2 takes the role of X. Then we use X_2 as the neighborhood to generate X_3, and repeat this procedure until we obtain X_q.

To generate more fake data, we can initiate more chains with the same starting node X_1 (where the real data are) and independent Brownian paths, and then "walk" along

Algorithm 2 Generator in the Decorrelating and Branching

```
Input: real data $\{\tilde{X}(\omega_i)\}_{i=1}^{M}$, # of steps $q$, generator $G_\theta$;
Set $\{X_1(\omega_i)\}_{i=1}^{M} := \{\tilde{X}(\omega_i)\}_{i=1}^{M}$;
for $k = 2$ to $q$ do
    Generate $M$ independent copies of initials positions and Brownian paths $\{\xi_k(\omega_i), B_k(\omega_i)\}_{i=1}^{M}$;
    Generate $M$ paths $\{X_k(\omega_i)\}_{i=1}^{M}$ by
        $X_k(\omega_i) = G_\theta(\xi_k(\omega_i), B_k(\omega_i), X_{k-1}(\omega_i))$;
end for
Output: $\{X_q(\omega_i)\}_{i=1}^{M}$
```

the chain to get $X_q^{(i)}, i = 2, 3, \ldots$. Again, $X_q^{(i)}$ is asymptotically uncorrelated to X_1. By the definition of DC-SDEs, X_q and $X_q^{(i)}$ are conditionally independent with conditioning on X_1. Therefore, we can claim that X_q and $X_q^{(i)}$ are asymptotically uncorrelated.

Architecture. Note that although the directed chain SDE pair $(X, \tilde{X})$ is Markovian, X itself can be non-Markovian as a standalone stochastic process. All the historical information can be embedded in the neighborhood process and fetched through V_0 and V_1. Such a property leads to one of the key differences between our method and Neural SDEs: there is no need to embed time series into a hidden space. In our implementations, V_0 and V_1 take standard feedforward neural networks; see Appendix B.2 for details.

Discriminator

The purpose of the discriminator is to identify the optimal parameters in the V_0- and V_1- networks. We use the Wasserstein GAN framework [66, 6] to train the generator, and two types of discriminators will be used here.

SigWGAN. Using the idea of expected signature, [126, 125] designed Sig-Wasserstein GAN by directly minimizing the signature Wasserstein-1 distance,

$$\text{Sig-W}_1(\mu, \nu) := |\mathbb{E}_{X\sim\mu}[S(X)] - \mathbb{E}_{X\sim\nu}[S(X)]|,$$

where μ and ν are two distributions of time series corresponding to real data and fake data, S is the signature map, and $|\cdot|$ is the l_2 norm. For practical use, we approximate the infinite sequence S by truncating signatures up to some finite order m, i.e.,

$$\text{Sig-W}_1^m(\mu,\nu) := |\mathbb{E}_\mu[S^m(X)] - \mathbb{E}_\nu[S^m(X)]|. \tag{4.42}$$

The higher the truncation order m, the more information the signature can capture. However, the number of terms in the truncated signature will grow exponentially and become costly when the time series data is high-dimensional.

Neural CDEs. Neural controlled differential equations are the second candidate for the discriminator when the underlying time series is of high dimension. This is also the discriminator used in [87]. Let $D_\phi : X \mapsto R$ be a Neural CDE discriminator where ϕ denotes the network parameters. The training goal is to solve the following optimization problem for the generator

$$\min_\theta \mathbb{E}_{\xi,B}\big[D_\phi\big(G_\theta(\xi,B,\tilde{X})\big)\big],$$

and the following one for the discriminator

$$\max_\phi \big\{\mathbb{E}_{\xi,B}\big[D_\phi\big(G_\theta(\xi,B,\tilde{X})\big)\big] - \mathbb{E}_{\tilde{X}}\big[D_\phi(\tilde{X})\big]\big\}. \tag{4.43}$$

Compared to only using the Neural CDEs as the discriminator, we notice that a combination of Neural CDEs and lower-order signature Wasserstein-1 distance as the discriminator works better for the third numerical example below. That is, the generator is optimized with respect to

$$\min_{\theta} \big\{ \mathbb{E}_{\xi,B} \big[D_{\phi}\big(G_{\theta}(\xi, B, \tilde{X})\big)\big] \\ + \text{Sig-W}_1^m\big(\text{Law}(\tilde{X}), \text{Law}\big(G_{\theta}(\xi, B, \tilde{X})\big)\big)\big\}. \tag{4.44}$$

Remark that DC-GANs can work with different discriminators, and here we choose to use neural CDEs and SigWGAN as the discriminators. The pseudo-algorithm of the overall training strategy is summarized in Algorithm 3.

Algorithm 3 The Training Phase

```
Input: real data $\{\tilde{X}(\omega_i)\}_{i=1}^M$, boolean variable $cde$, total epochs $E$, signature truncation order $m$;
for $e = 1$ to $E$ do
    Generate independent copies of initials and Brownian motions $(\xi(\omega_i), B(\omega_i))_{i=1}^M$;
    Generate fake data $\{X(\omega_i)\}_{i=1}^M$ by
        $X(\omega_i) = G_\theta(\xi(\omega_i), B(\omega_i), \tilde{X}(\omega_i))$;
    if $cde$ is True then
        Compute the loss (4.43) and its gradients w.r.t. $\phi$;
        Compute the loss (4.44) and its gradients w.r.t. $\theta$;
        Update $\theta$ by stochastic gradient descent optimiser;
        Update $\phi$ by stochastic gradient ascent optimiser;
    else
        Compute the loss (4.42) and its gradients w.r.t. $\theta$;
        Update $\theta$ by stochastic gradient descent optimiser;
    end if
end for
Output: Generator $G_\theta$.
```

4.5.2 Experiments

We present the performance of the proposed DC-GANs on four different datasets, including stochastic opinion dynamics, network dynamics from neural science, and real-world stock data and energy consumption data. In all cases, we set $q = 10$, i.e., "walk" along the chain for ten steps during the decorrelating phase. Other hyperparameters for neural network training can be found in Appendix B.2 for details. The codes are submitted as supplementary material and will be made public upon acceptance.

Benchmarks & Evaluation. The first two synthetic datasets are generated by SDEs, the third real-world data set of stock price time series was extracted from Yahoo Finance[1], and the fourth real-world energy consumption data were obtained from Ireland's open data portal[2]. We compare our results by DC-GANs with SigWGAN, CTFP, and Neural SDEs, and DC-GANs give much better accuracy under discriminative, predictive, and maximum mean discrepancy (MMD) metrics detailed below. We also provide independence metrics to show that our decorrelating and branching scheme can resolve the independence problem. We also test over different discriminators, and show the flexibility of choosing the one that brings better performance or has a faster running time.

Metrics

Marginal Distribution & MMD. For the first two examples, we plot histograms to compare their marginal distributions at several time stamps. To measure the goodness of fitting for time series, we use maximum mean discrepancy (MMD) induced by the expected signature given in (4.42).

Discriminative Metric. To quantitively measure the similarity between the fake data generated by DC-GANs and real data, we train a post-hoc time series classifier by optimizing a two-layer LSTM to discriminate original and fake sequential data. The fake data is labeled *nonreal* and the original data is labeled *real*. The worse discriminative

[1] `https://finance.yahoo.com/quote/GOOG?p=GOOG&.tsrc=fin-srch`.

[2] `https://data.gov.ie/dataset?theme=Energy`.

ability of the post-hoc time series classifier implies the better performance of the time series generator. Our discriminative score is calculated as the absolute difference between 0.5 and predicting accuracy on testing data, thus a smaller score indicates a better generator.

Predictive Metric. Typically, a useful time series dataset contains temporal evolution information, and we can predict the future given past data. We expect that DC-GANs can capture this temporal dynamic property accurately from the original data. To this end, we train an auxiliary two-layer LSTM sequential predictor on the generated time series and test this post-hoc predictor on the original time series. The predictive score is calculated as the L^1 distance between predicted sequences and true sequences on testing data (the real data), with smaller scores for better generators.

Independence Metric. It is crucial for success to show that our algorithm can address the independence problem. As an independence metric, we use

$$\rho(x, y) := \sup_{t \in [0,T]} \|\rho(x_t, y_t)\|_1, \tag{4.45}$$

where $x, y \in L^2(\Omega \times [0, T], \mathbb{R}^N)$ and $\rho(x_t, y_t)$ represents the cross-correlation matrix between random vectors x_t, y_t. Smaller $\rho(x, y)$ means less correlation between real data x and generated data y.

All experiments are run over ten different random seeds, and we report the mean and standard deviation (in the parentheses) for all metrics in Tables 4.1–4.4. We give more details on how all these metrics are implemented in Appendix B.2.1.

Example 1: Stochastic Opinion Dynamics

We first consider stochastic opinion dynamics modeled by the following MV-SDE

$$\mathrm{d}Y_t = -\left[\int_{\mathbb{R}} \varphi_\theta(\|Y_t - y\|)(Y_t - y)\ \mu_t(\mathrm{d}y)\right]\mathrm{d}t + \sigma\,\mathrm{d}W_t,$$

where φ_θ is a interaction kernel with $\theta_1, \theta_2 > 0$,

$$\varphi_\theta(r) = \begin{cases} \theta_1 \exp\left(-\frac{0.01}{1-(r-\theta_2)^2}\right), & r > 0, \\ 0, & r \leq 0, \end{cases}$$

and $\mu_t = \mathrm{Law}(Y_t)$ denotes the distribution of Y_t. One can interpret θ_1 as a scale parameter that characterizes the intensity of the attraction between entities, and θ_2 as the range parameter that determines the distance, within which an entity must be of one another in order to interact. This model is widely used in many disciplines, from flocking and swarming behaviors in biology (where Y_t is the position) to public opinion evolution in social science (where Y_t is the opinion towards a topic). We refer to [124] for further details.

We choose $\theta_1 = 6$, $\theta_2 = 0.2$, $\sigma = 0.1$, $T = 1$, $\Delta t = 0.01$, and generate 8192 paths. The distribution μ_t is approximated by the empirical distribution of 8192 samples. These samples are used to produce the blue density in Figure 4.1, where a clear shift in distribution from unimodality to bimodality is observed.

We first compare with the Neural SDEs method [87]. Figure 4.1 gives the comparison of the marginal distributions at $t = 0.1, 0.3, 0.5, 0.7, 0.9, 1.0$. One can see that DC-GANs can accurately capture the bimodal distribution in general, but the Neural SDE method can not. Under the MMD metric (4.42), the discrepancy of DC-GANs is 0.07, while the Neural SDEs give 0.12. More comparisons with SigWGAN, CTFP, and Neural

SDEs under discriminative, MMD, and independence metrics are provided in Table 4.1. Our proposed DC-GANs have a smaller discriminative score, and an independence score comparable with the ones produced by the Neural SDE generator, SigWGAN, and CTFP, all of which generate purely independent samples. Therefore, we conclude that DC-GANs can produce fake data closer to the real data without independence issues.

Table 4.1: Stochastic Opinion Dynamics (Example 1). The scores are computed for SigWGAN, CTFP, Neural SDEs, and DC-GANs under different metrics. The numbers in the parenthesis are the corresponding standard deviations of each score. Note that a smaller value means a better approximation, which indicates the DC-GANs provide more accurate fake data with compared independence and running time.

Method	Discriminative	MMD	Independence	Time (min)
SigWGAN	0.213 (0.01)	0.328 (0.004)	0.009(0.004)	6.55
CTFP	0.131 (0.02)	0.281 (0.005)	0.010(0.003)	5.58
Neural SDEs	0.045 (0.025)	0.122 (0.003)	0.007 (0.005)	7.07
DC-GANs	**0.028 (0.019)**	**0.07 (0.003)**	0.009 (0.004)	6.82

Example 2: Stochastic FitzHugh-Nagumo Model

FitzHugh-Nagumo model is a standard model from neuroscience [8, 134], used to describe the neurons' interacting spiking. Mathematically, for N neurons and P different neuron populations, and $i \in \{1, \dots, N\}$, we denote by $p(i) = \alpha, \alpha \in \{1, \dots, P\}$ the population of i-th particle that belongs to. The state vector of neural i, $(X_t^{i,N})_{t\in[0,T]} =$

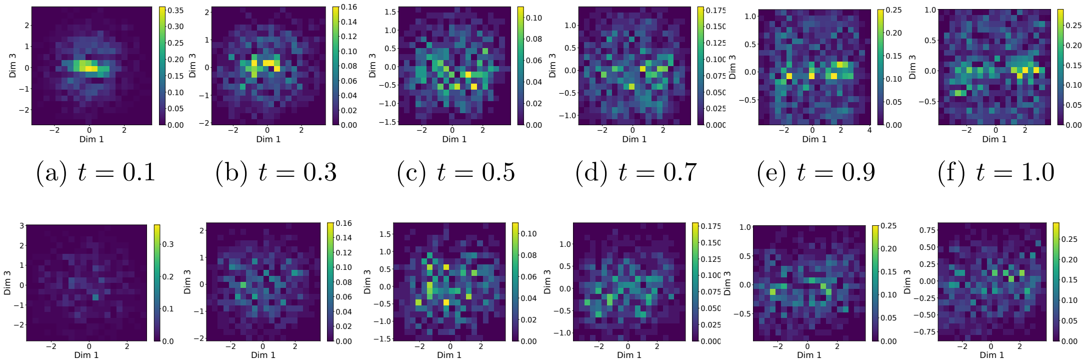
$(V_t^{i,N}, w_t^{i,N}, y_t^{i,N})_{t\in[0,T]}$, satisfies the SDE,

$$\begin{aligned} \mathrm{d}X_t^{i,N} = f_\alpha(t, X_t^{i,N})\,\mathrm{d}t + g_\alpha(t, X_t^{i,N}) &\begin{bmatrix} \mathrm{d}W_t^i \\ \mathrm{d}W_t^{i,y} \end{bmatrix} \\ + \sum_{\gamma=1}^{P} \frac{1}{N_\gamma} \sum_{j,p(j)=\gamma} &\Big(b_{\alpha\gamma}(X_t^{i,N}, X_t^{j,N})\,\mathrm{d}t \\ &+ \beta_{\alpha\gamma}(X_t^{i,N}, X_t^{j,N})\,\mathrm{d}W_t^{i,\gamma} \Big), \end{aligned}$$

where V denotes a short, nonlinear elevation of membrane voltage, w denotes a slower, linear recovery variable, N_γ denotes the number of neurons in the population γ. We defer more details about model description and training data generation to Appendix B.2.2.

The FitzHugh-Nagumo system is an example of a relaxation oscillator, and exhibits a characteristic excursion in phase space, before the variables V and w relax back to their rest values. As a result, their distributions are typically multimodal distributed; see Figure B.1 in Appendix B.2.2.

Figure 4.4 depicts the differences of their joint marginal densities between generated time series and training (real) time series on channels 1 and 3 at $t = 0.1, 0.3, 0.5, 0.7, 0.9, 1.0$. The darker the color the smaller the differences, thus the closer the distribution and indicating a better generator. It can be observed that DC-GANs produce less difference in joint marginal densities at multiple time stamps. Under discriminative, predictive, and MMD metrics, DC-GANs give better samples than SigWGAN, CTFP, and Neural SDEs consistently; see Table 4.2. In particular, fake samples produced by DC-GANs are almost indistinguishable for a two-layer LSTM classifier after exhaustive training. By the comparison using MMD, one can see that DC-GANs generate fake samples with distributions significantly closer to real data than the other three methods. The independence scores given by (4.45) are nearly indistinguishable.



(a) $t = 0.1$ (b) $t = 0.3$ (c) $t = 0.5$ (d) $t = 0.7$ (e) $t = 0.9$ (f) $t = 1.0$

(g) $t = 0.1$ (h) $t = 0.3$ (i) $t = 0.5$ (j) $t = 0.7$ (k) $t = 0.9$ (l) $t = 1.0$

Figure 4.4: Stochastic FitzHugh-Nagumo Model (Example 2). Figures (a)-(f) are generated by Neural SDEs, and Figures (g)-(l) are generated by DC-GANs. They show their joint marginal densities differences between estimated time series and real-time series on channels 1 (Dim 1) and 3 (Dim 3) at $t \in \{0.1, 0.3, 0.5, 0.7, 0.9, 1.0\}$. Darker color means a smaller difference, and thus a better fitting. One can observe that DC-GANs produce less difference in joint marginal densities at multiple time stamps.

Table 4.2: Stochastic FitzHugh-Nagumo Model (Example 2). The scores are computed for SigWGAN, CTFP, Neural SDEs, and DC-GANs under different metrics. Note that a smaller value means a better approximation. Parenthesized numbers are standard deviations.

Method	Discriminative	Predictive	MMD	Independence	Time (min)
SigWGAN	0.126 (0.04)	0.44 (0.001)	0.737 (0.01)	0.0083(0.0024)	9.63
CTFP	0.275 (0.05)	0.501 (0.004)	1.095 (0.02)	0.0088(0.0023)	6.88
Neural SDEs	0.20 (0.003)	0.44 (0.000)	0.97 (0.02)	0.0085 (0.0023)	8.25
DC-GANs	**0.01 (0.009)**	**0.439 (0.000)**	**0.47 (0.02)**	0.0085 (0.0027)	8.13

Example 3: Stock Price Time Series (Real Data)

The third example is Google stock prices from 2004 to 2019, extracted from Yahoo Finance. Sequences of stock prices are known as continuous time series data with unknown distributions, and can even be non-Markovian. Our data have six channels, volume and high, low, opening, closing, and adjusted closing prices. Among all, the first five channels are multimodal. The combined discriminator (4.44) (Neural CDE and Sig-W_1) is used in GAN for this experiment, and we list the comparison results in Table 4.3. One can

Table 4.3: Stocks Price Time Series (Example 3). The scores are computed for SigWGAN, CTFP, TimeGAN, Neural SDEs, and DC-GANs under different metrics. Note that a smaller value means a better approximation. Parenthesized numbers are standard deviations.

Model	Discriminative	Predictive	MMD	Independence	Time (min)
SigWGAN	0.183 (0.03)	0.060 (0.004)	0.121 (0.011)	0.012(0.004)	4.13
CTFP	0.256 (0.05)	0.138 (0.006)	0.187 (0.009)	0.013(0.005)	6.40
TimeGAN	0.102 (0.021)	0.038 (0.001)	0.0220 (0.007)	0.011 (0.005)	>660
Neural SDEs	0.085 (0.028)	0.048 (0.001)	0.0193 (0.008)	0.011 (0.006)	9.93
DC-GANs	**0.045 (0.015)**	**0.036 (0.000)**	**0.0133 (0.005)**	0.013 (0.006)	9.53

see that DC-GANs outperform SigWGAN, CTFP, TimeGAN, and Neural SDEs under all three metrics.

Example 4: Energy Consumption Data (Real Data)

We download the Energy Consumption data from Ireland's open data portal, and choose four electric and gas consumption time series from 02/2011–02/2013, where channels 1,3, and 4 exhibit multimodal features. We list the comparison results in Table 4.4, which shows consistent advantages of DC-GANs compared with other methods under different metrics as in previous examples. Notice that DC-GANs can be used with both Neural CDEs (NCDE) and Signature Wasserstein (SigW) discriminators, and in this example, DC-GANs with SigW as the discriminator present better performance and have a faster running time.

4.5.3 Conclusion

We propose a novel time series generator, DC-GANs, motivated by the study of [53, 83] on directed chain SDEs (DC-SDEs). Compared to more complicated graph systems, we find from numerical examples that the directed chain systems exhibit promising ability in fitting time series of multimodal probability distributions. We prove in theory

Table 4.4: Energy Consumption Data from Ireland's open data portal (Example 4). The scores are computed for SigWGAN, CTFP, Neural SDEs, and DC-GANs under different metrics. Note that a smaller value means a better approximation. Parenthesized numbers are standard deviations.

Method	Discriminative	Predictive	MMD	Independence	Time(min)
SigWGAN	0.368 (0.09)	0.159 (0.002)	0.135 (0.006)	0.022(0.007)	9.47
CTFP	0.487 (0.01)	0.185 (0.001)	0.558 (0.006)	0.021(0.008)	8.52
Neural SDEs	0.413 (0.06)	0.172 (0.004)	0.126 (0.004)	0.022(0.006)	9.73
DC-GANs (w/ NCDE)	0.322 (0.12)	0.155 (0.006)	0.077 (0.003)	0.029(0.007)	23.44
DC-GANs (w/ SigW)	**0.310 (0.09)**	**0.151 (0.008)**	**0.075 (0.003)**	0.033(0.008)	9.38

that DC-GANs have the same flexibility as the Neural SDEs in capturing marginal distributions, and DC-GANs naturally embrace the non-Markovian property in the topological structure, if needed. We also prove that the correlation of the generated path decays exponentially fast as the graph distance of the generated path from the original data becomes large under some mild assumptions, and hence, the lack-of-independence problem can be overcome by walking along the directed chain. We present four numerical examples, two synthetic datasets generated by the SDEs, and two real-world data of stock price and energy consumption, and show that DC-GANs have a better performance than SigWGAN, CTFP, Neural SDEs and TimeGAN, with the comparable independence property. We remark that the DC-GANs algorithm can also work with irregular data (i.e., the sample paths may have data sampled on different time grids), which may happen in healthcare applications.

Appendix A

Signatured DFP for MFG with Common noises

A.1 Proof of Lemma 3.4.1

In this appendix, we shall follow rough path theory and signatures stated in Chapter 2 to give the proof of Lemma 3.4.1 using the factorial decay property of signatures.

Lemma A.1.1 *Suppose $\mu_t = \mathbb{E}[\iota(X_t)|\mathcal{F}_t^B]$ where $\iota : \mathbb{R}^d \to \mathbb{R}$ is a measurable function. View μ_t as $\mu(t, B_{0:t})$ with $\mu : \mathcal{V}^p([0,T], \mathbb{R}^{n_0+1}) \to \mathbb{R}$ continuous for some $p \in (2,3)$, and let $K \subset \mathcal{V}^p([0,T], \mathbb{R}^{n_0+1})$ be a compact set, then for any $\epsilon > 0$, there exist a positive integer M and a linear functional $l \in T((\mathbb{R}^{n_0+1}))^*$, such that*

$$\sup_{t\in[0,T]} \sup_{\hat{B}\in K} |\mu_t - \langle l, S^M(\hat{B}_{0:t})\rangle| < \epsilon. \tag{A.1}$$

Proof: By constructing the iterated integral in Stratonovich sense, $S(\hat{B}_{0:T})$ is the signature of a p-geometric rough path $\forall p \in (2,3)$ [64], and thus it characterizes $B_{0:T}$ uniquely. Therefore, conditional distribution $\mu_t = \mathbb{E}[\iota(X_t)|\mathcal{F}_t^B]$ can be written as $\mu_t :=$

$\mu(t, B_{0,t}) = \mu(\hat{B}_{0,t})$.

By Proposition 2.1.13, for any $\epsilon > 0$ there exits l such that

$$\sup_{\hat{B}\in K} |\mu(\hat{B}_{0:T}) - \langle l, S(\hat{B}_{0:T})\rangle| < \frac{\epsilon}{2}. \tag{A.2}$$

Since $|\langle l, S(\hat{B}_{0:T}) - S^M(\hat{B}_{0:T})\rangle| \leq \|l\| \cdot \|S(\hat{B}_{0:T}) - S^M(\hat{B}_{0:T})\|$ where the first norm is functional norm and second is tensor norm and $\|S(\hat{B}_{0:T}) - S^M(\hat{B}_{0:T})\| = \sum_{i\geq M+1} \|\hat{B}^i_{0:T}\|$. By the compactness of K, and (2.7), (2.8), $\sum_{i\geq M+1} \|\hat{B}^i_{0:T}\|$ admits a convergent uniform norm over $\hat{B} \in K$ and goes to 0 as $M \to \infty$. Then for M large enough,

$$\sup_{\hat{B}\in K} |\mu(\hat{B}_{0:T}) - \langle l, S^M(\hat{B}_{0:T})\rangle| < \frac{\epsilon}{2} + \sup_{\hat{B}\in K} |\langle l, S(\hat{B}_{0:T}) - S^M(\hat{B}_{0:T})\rangle| < \frac{\epsilon}{2} + \frac{\epsilon}{2} = \epsilon. \tag{A.3}$$

For $t < T$, we extend path $\hat{B}_{0:t}$ to space $\mathcal{V}^p([0,T], \mathbb{R}^d)$ by defining

$$\tilde{B}^t_s := \begin{cases} \hat{B}_s, & 0 \leq s \leq t \\ \hat{B}_t, & t < s \leq T. \end{cases}$$

Then $\tilde{B}^t_{0:T} \in \mathcal{V}^p([0,T], \mathbb{R}^d)$, $S(\tilde{B}^t_{0:T}) = S(\hat{B}_{0:t})$ by Chen's identity (2.5), and $\mu(\hat{B}_{0:t}) = \mu(\tilde{B}^t_{0,T})$. Denote $\tilde{K} = \{\tilde{B}^t_{0:T}, \forall t \in [0,T] : \tilde{B}^t_{0:T}$ is constructed by $\hat{B}_{0:t}$ and $\hat{B} \in K\}$. Thus $\tilde{K}$ is also compact.

$$\begin{aligned} \sup_{t\in[0,T]} \sup_{\hat{B}\in K} |\mu(\hat{B}_{0:t}) - \langle l, S^M(\hat{B}_{0:t})\rangle| &= \sup_{t\in[0,T]} \sup_{\hat{B}\in K} |\mu(\tilde{B}^t_{0:T}) - \langle l, S^M(\tilde{B}^t_{0:T})\rangle| \\ &= \sup_{\tilde{B}\in\tilde{K}} |\mu(\tilde{B}_{0:T}) - \langle l, S^M(\tilde{B}_{0:T})\rangle| < \epsilon, \end{aligned} \tag{A.4}$$

where the second equality is due to the construction of $\tilde{B}^t_{0:T}$ and the last inequality is by (A.3). ■

A.2 Details of Implementing the Sig-DFP Algorithm

The simulation of $X^{i,(n)}$ and $J_B(\varphi, \hat{\mu}^{(n-1)})$ follows

$$\begin{aligned} J_B(\varphi, \hat{\mu}^{(n-1)}) &= \frac{1}{B}\sum_{i=1}^{B}\Bigg(\sum_{k=0}^{L-1} f(t_k, X_k^{i,(n)}, \hat{\mu}_k^{(n-1)}(\omega^i), \alpha_\varphi(t_k, X_k^{i,(n)}, \hat{\mu}_k^{(n-1)}(\omega^i))\Delta_k \\ &\quad + g(X_L, \hat{\mu}_L^{(n-1)}(\omega^i))\Bigg), \qquad \text{(A.5)} \\ X_{k+1}^{i,(n)} &= X_k^{i,(n)} + b(t_k, X_k^{i,(n)}, \hat{\mu}_k^{(n-1)}(\omega^i), \alpha_\varphi(t_k, X_k^{i,(n)}, \hat{\mu}_k^{(n-1)}(\omega^i))\Delta_k \\ &\quad + \sigma(t_k, X_k^{i,(n)}, \hat{\mu}_k^{(n-1)}(\omega^i), \alpha_\varphi(t_k, X_k^{i,(n)}, \hat{\mu}_k^{(n-1)}(\omega^i))\Delta W_k^i \\ &\quad + \sigma^0(t_k, X_k^{i,(n)}, \hat{\mu}_k^{(n-1)}(\omega^i), \alpha_\varphi(t_k, X_k^{i,(n)}, \hat{\mu}_k^{(n-1)}(\omega^i))\Delta B_k^i, \quad X_0^{i,(n)} = X_0^i \sim \mu_0, \\ &\qquad \text{(A.6)} \end{aligned}$$

where $\hat{\mu}_k^{(n-1)}(\omega^i)$ is computed by $\hat{\mu}_k^{(n-1)}(\omega^i) = \langle \bar{l}^{(n-1)}, S^M(\hat{B}^i_{0:t_k})\rangle$ with $\bar{l}^{(n-1)}$ obtained from the previous round of fictitious play. Then $l^{(n)}$ is calculated by regressing $\{\iota(X_0^{i,(n)}), \iota(X_{L/2}^{i,(n)}), \iota(X_L^{i,(n)})\}_{i=1}^N$ on $\{S^M(\hat{B}_{0,0}), S^M(\hat{B}_{0,t_{L/2}}), S^M(\hat{B}_{0,t_L})\}_{i=1}^N$, and we update $\bar{l}^{(n)} = \frac{n-1}{n}\bar{l}^{(n-1)} + \frac{1}{n}l^{(n)}$ for $n \geq 1$. The algorithm starts with a random initialization $\bar{l}^{(0)}$ to produce $\hat{\mu}^{(0)}$.

Linear-Quadratic MFGs. We set α_φ to be a feed-forward NN with two hidden layers of width 64. The signature depth is chosen at $M = 2$. This model is trained for $N_{round} = 500$ iterations of fictitious play. Note that fictitious play has a slow convergence speed since our initial guess $m^{(0)}$ is far from the truth. Therefore, we only apply averaging over distributions (or linear functions) during the second half iteration. We set the learning rate as 0.1 for the first half iterations and 0.01 for the second half. The minibatch size is $B = 2^{10}$, and hence $N_{batch} = 2^5$.

Mean-field Portfolio Game. We consider signature depth $M = 2$ and use a fully connected neural network π_φ with four hidden layers to estimate π_t. Since different players are characterized by their type vectors ζ, π_φ takes (ζ, t, X_t, m_t) as inputs. Hidden neurons in each layer are (64, 32, 32, 16). We train our model with $N_{round} = 500$ rounds fictitious play. The learning rate starts at 0.1 and is reduced by a factor of 5 after every 200 rounds. The minibatch size is $B = 2^{10}$, and hence $N_{batch} = 2^5$.

Mean-field Game of Optimal Consumption and Investment. In this example, signature depth is $M = 4$. The optimal controls $(\pi_t, c_t)_{0 \leq t \leq 1}$ are estimated by two neural networks π_φ and c_φ, each with three hidden layers. Due the nature of heterogeneous extended MFG, both α_φ and c_φ take $(\zeta_t, t, X_t, m_t, \Gamma_t)$ as the inputs. Hidden layers in each network have width (64, 64, 64). We will propagate two conditional distribution flows, *i.e.*, two linear functionals $\bar{l}^{(n)}, \bar{l}_c^{(n)}$ during each round fictitious play. Instead of estimating m_t, Γ_t directly, we estimate $\mathbb{E}[\log X_t^* | \mathcal{F}_t^B], \mathbb{E}[\log c_t^* | \mathcal{F}_t^B]$ by $\langle \bar{l}^{(n)}, S^4(\hat{B}_{0:t}) \rangle$, $\langle \bar{l}_c^{(n)}, S^4(\hat{B}_{0:t}) \rangle$, and then take the exponential to get m_t, Γ_t. To ensure the non-negativity condition, we evolve $\log X_t$ according to (A.22), use c_φ to predicted $\log c_t$, and then take exponential to get c_t, X_t. We use $N_{round} = 600$ rounds fictitious play training, learning rate 0.1 decaying by a factor of 5 for every 200 rounds, the minibatch size $B = 2^{11}$, and hence $N_{batch} = 2^4$.

The training time for all three experiments with sample size $N = 2^{13}, 2^{14}, 2^{15}$ is given in Table A.1.

Table A.1: Training time in minutes. Here LQ-MFG = Linear-Quadratic mean-field games, MF Portfolio = Mean-field Portfolio Game, and MFG with Consump. = Mean-field Game of Optimal Consumption and Investment.

	$N = 2^{13}$	$N = 2^{14}$	$N = 2^{15}$
LQ-MFG	12.4	23.7	46.7
MF Portfolio	12.3	23.3	45.5
MFG with Consump.	23.4	40.9	80.1

A.3 Proof of Theorems 3.4.2 and 3.4.3

We first list all main assumptions on $(b, \sigma, \sigma^0, f, g)$ that will be used to prove Theorem 3.4.2. Let $\|\cdot\|$ be the Euclidean norm and K be the same constant for all assumptions below.

Assumption A.3.1 *We make assumptions* **A1-A3** *and* **B1-B3** *as follows.*

A1. *(Lipschitz) $\partial_x f, \partial_\alpha f, \partial_x g$ exist and are K-Lipschitz continuous in (x, α) uniformly in (t, μ), i.e., for any $t \in [0, T]$, $x, x' \in \mathbb{R}^d, \alpha, \alpha' \in \mathbb{R}^m, \mu \in \mathcal{P}^2(\mathbb{R}^d)$,*

$$\|\partial_x g(x, \mu) - \partial_x g(x', \mu)\| \leq K\|x - x'\|,$$

$$\|\partial_x f(t, x, \mu, \alpha) - \partial_x f(t, x', \mu, \alpha')\| \leq K(\|x - x'\| + \|\alpha - \alpha'\|),$$

$$\|\partial_\alpha f(t, x, \mu, \alpha) - \partial_\alpha f(t, x', \mu, \alpha')\| \leq K(\|x - x'\| + \|\alpha - \alpha'\|).$$

The drift coefficient $b(t, x, \mu, \alpha)$ in (3.3) takes the form

$$b(t, x, \mu, \alpha) = b_0(t, \mu) + b_1(t)x + b_2(t)\alpha,$$

where $b_0 \in \mathbb{R}^d$, $b_1 \in \mathbb{R}^{d\times d}$ and $b_2 \in \mathbb{R}^{d\times m}$ are measurable functions and bounded by K. The diffusion coefficients $\sigma(t, x, \mu)$ and $\sigma^0(t, x, \mu)$ are uncontrolled and K-Lipschitz in x uniformly in (t, μ):

$$\|\sigma(t, x, \mu)\| \leq K\|x - x'\|, \quad \|\sigma^0(t, x, \mu)\| \leq K\|x - x'\|.$$

A2. *(Growth) $\partial_x f, \partial_\alpha f, \partial_x g$ satisfy a linear growth condition, i.e., for any $t \in [0, T]$,*

$x \in \mathbb{R}^d, \alpha \in \mathbb{R}^m, \mu \in \mathcal{P}^2(\mathbb{R}^d)$,

$$\begin{aligned}\|\partial_x g(x,\mu)\| &\leq K\bigg(1 + \|x\| + \Big(\int_{\mathbb{R}^d} \|y\|^2 \, \mathrm{d}\mu(y)\Big)^{\frac{1}{2}}\bigg),\\ \|\partial_x f(t,x,\mu,\alpha)\| &\leq K\bigg(1 + \|x\| + \|\alpha\| + \Big(\int_{\mathbb{R}^d} \|y\|^2 \, \mathrm{d}\mu(y)\Big)^{\frac{1}{2}}\bigg),\\ \|\partial_\alpha f(t,x,\mu,\alpha)\| &\leq K\bigg(1 + \|x\| + \|\alpha\| + \Big(\int_{\mathbb{R}^d} \|y\|^2 \, \mathrm{d}\mu(y)\Big)^{\frac{1}{2}}\bigg).\end{aligned}$$

In addition f, g satisfy a quadratic growth condition in μ:

$$\begin{aligned}|g(0,\mu)| &\leq K\bigg(1 + \int_{\mathbb{R}^d} \|y\|^2 \, \mathrm{d}\mu(y)\bigg),\\ |f(t,0,\mu,0)| &\leq K\bigg(1 + \int_{\mathbb{R}^d} \|y\|^2 \, \mathrm{d}\mu(y)\bigg).\end{aligned}$$

A3. *(Convexity) g is convex in x and f is convex jointly in (x,α) with strict convexity in α, i.e., for any $x, x' \in \mathbb{R}^d, \mu \in \mathcal{P}^2(\mathbb{R}^d)$,*

$$(\partial_x g(x,\mu) - \partial_x g(x',\mu))^T(x - x') \geq 0,$$

and there exist a constant $c_f > 0$ such that for any $t \in [0,T]$, $x, x' \in \mathbb{R}^d, \alpha, \alpha' \in \mathbb{R}^m$, $\mu \in \mathcal{P}^2(\mathbb{R}^d)$,

$$f(t,x',\alpha',\mu) \geq f(t,x,\alpha,\mu) + \partial_x f(t,x,\alpha,\mu)^T(x'-x) + \partial_\alpha f(t,x,\alpha,\mu)^T(\alpha'-\alpha) + c_f\|\alpha'-\alpha\|^2.$$

B1. *(Lipschitz in μ) $\partial_x g, \partial_x f, \partial_\alpha f, b_0, \sigma, \sigma^0$ are Lipschitz continuous in μ uniformly in*

(t,x), i.e., there exists a constant K such that

$$\begin{aligned}
\|\partial_x g(x,\mu) - \partial_x g(x,\mu')\| &\leq K\mathcal{W}_2(\mu,\mu'),\\
\|\partial_x f(t,x,\mu,\alpha) - \partial_x f(t,x,\mu',\alpha)\| &\leq K\mathcal{W}_2(\mu,\mu')\\
\|\partial_\alpha f(t,x,\mu,\alpha) - \partial_\alpha f(t,x,\mu',\alpha)\| &\leq K\mathcal{W}_2(\mu,\mu')\\
\|b_0(t,\mu) - b_0(t,\mu')\| &\leq K\mathcal{W}_2(\mu,\mu'),\\
\|\sigma(t,x,\mu) - \sigma(t,x,\mu')\| &\leq K\mathcal{W}_2(\mu,\mu'),\\
\|\sigma^0(t,x,\mu) - \sigma^0(t,x,\mu')\| &\leq K\mathcal{W}_2(\mu,\mu'),
\end{aligned}$$

for all $t \in [0,T], x \in \mathbb{R}^d, \alpha \in \mathbb{R}^m$, $\mu,\mu' \in \mathcal{P}^2(\mathbb{R}^d)$, where $\mathcal{W}_2$ is the 2-Wasserstein distance.

B2. *(Separable in α,μ) f is of the form*

$$f(t,x,\mu,\alpha) = f^0(t,x,\alpha) + f^1(t,x,\mu),$$

where f^0 is assumed to be convex in (x,α) and strictly convex in α, and f^1 is assumed to be convex in x.

B3. *(Weak monotonicity) For all $t \in [0,T]$, $\mu,\mu' \in \mathcal{P}^2(\mathbb{R}^d)$ and $\gamma \in \mathcal{P}^2(\mathbb{R}^d \times \mathbb{R}^d)$ with marginals μ,μ' respectively,*

$$\int_{\mathbb{R}^d\times\mathbb{R}^d} \left[(\partial_x g(x,\mu) - \partial_x g(y,\mu'))^T(x-y)\right]\gamma(\,\mathrm{d}x,\,\mathrm{d}y) \geq 0,$$
$$\int_{\mathbb{R}^d\times\mathbb{R}^d} \left[(\partial_x f(t,x,\mu,\alpha) - \partial_x g(t,y,\mu',\alpha))^T(x-y)\right]\gamma(\,\mathrm{d}x,\,\mathrm{d}y) \geq 0.$$

Note that Assumption A.3.1 extends conditions **A** and **B** in [3] by considering general drift coefficient $b(t,x,\mu,\alpha)$ and non-constant diffusion coefficients $\sigma(t,x,\mu)$ and

$\sigma^0(t, x, \mu)$.

Our proof of Theorem 3.4.2 uses the probabilistic approach. To this end, we define the Hamiltonian by

$$H(t, x, y, \mu, \alpha) = b(t, x, \mu, \alpha) \cdot y + f(t, x, \mu, \alpha).$$

Denote by $\hat{\alpha}$ the minimizer of the Hamiltonian which is unique due to Assumptions **A1** and **A3**:

$$\hat{\alpha}(t, x, y, \mu) = \operatorname*{arg\,min}_{\alpha \in \mathbb{R}^m} H(t, x, y, \mu, \alpha). \tag{A.7}$$

By the Lipschitz property of $\partial_\alpha f$ in (t, μ, α) and the boundedness of $b_2(t)$, $\hat{\alpha}$ is Lipschitz in (x, y, μ). Let $\hat{H}$ be the Hamiltonian, with $\hat{\alpha}$ obtained in (A.7),

$$\hat{H}(t, x, y, \mu) = H(t, x, y, \mu, \hat{\alpha}(t, x, y, \mu)). \tag{A.8}$$

Under Assumptions **A1-A3**, with the stochastic maximum principle, the problem (3.2)-(3.3) is equivalent to solve the following FBSDE, given $\mu \in \mathcal{M}([0, T]; \mathcal{P}^2(\mathbb{R}^d))$,

$$\begin{aligned} &\mathrm{d}X_t = b(t, X_t, \mu_t, \hat{\alpha}(t, X_t, Y_t, \mu_t))\,\mathrm{d}t + \sigma(t, X_t, \mu_t)\,\mathrm{d}W_t + \sigma^0(t, X_t, \mu_t)\,\mathrm{d}B_t, \quad X_0 = x_0 \sim \mu_0, \\ &\mathrm{d}Y_t = -\partial_x \hat{H}(t, X_t, Y_t, \mu_t)\,\mathrm{d}t + Z_t\,\mathrm{d}W_t + Z_t^0\,\mathrm{d}B_t, \quad Y_T = \partial_x g(X_T, \mu_T). \end{aligned} \tag{A.9}$$

Moreover, the optimal control is given by

$$\hat{\alpha}_t = \hat{\alpha}(t, X_t, Y_t, \mu_t), \tag{A.10}$$

for any solution (X_t, Y_t, Z_t, Z_t^0) to FBSDE (A.9).

The next theorem describes the McKean-Vlasov FBSDE for finding the mean-field equilibrium (*cf.* Definition 3.2.1).

Theorem A.3.2 (Theorem 2.2.8, [3]) *Under Assumptions* **A1-A3***, the mean-field equilibrium of* (3.2)-(3.3) *exists if and only if the following McKean-Vlasov FBSDE is solvable:*

$$\begin{aligned} \mathrm{d}X_t &= b(t, X_t, \mathcal{L}(X_t|\mathcal{F}_t^B), \hat{\alpha}(t, X_t, Y_t, \mu_t))\,\mathrm{d}t + \sigma(t, X_t, \mathcal{L}(X_t|\mathcal{F}_t^B))\,\mathrm{d}W_t + \sigma^0(t, X_t, \mathcal{L}(X_t|\mathcal{F}_t^B))\,\mathrm{d}B_t, \\ \mathrm{d}Y_t &= -\partial_x \hat{H}(t, X_t, Y_t, \mathcal{L}(X_t|\mathcal{F}_t^B))\,\mathrm{d}t + Z_t\,\mathrm{d}W_t + Z_t^0\,\mathrm{d}B_t. \end{aligned} \tag{A.11}$$

Moreover, the mean-field control-distribution flow pair is given by

$$\alpha_t^* = \hat{\alpha}(t, X_t, Y_t, \mathcal{L}(X_t|\mathcal{F}_t^B)), \quad \mu_t^* = \mathcal{L}(X_t|\mathcal{F}_t^B), \quad \forall t \in [0, T]. \tag{A.12}$$

Theorem A.3.3 *Under Assumption A.3.1, the FBSDE systems* (A.9) *and* (A.11) *have unique solutions. Moreover, let $\mu_t^1, \mu_t^2 \in \mathcal{M}([0, T]; \mathcal{P}^2(\mathbb{R}^d))$ be different given flow of measures, and denote by $(X_t^i, Y_t^i, Z_t^i, Z_t^{0,i})$ the unique solution to FBSDE* (A.9) *given μ_t^i, then*

$$\mathbb{E}\left[\sup_{t\in[0,T]} \|\Delta X_t\|^2 + \sup_{t\in[0,T]} \|\Delta Y_t\|^2 + \int_0^T \|\Delta Z_t\|^2 + \|\Delta Z_t^0\|^2\,\mathrm{d}t\right] \leq C_{K,T}\mathbb{E}\left[\int_0^T (\Delta\mu_t)^2\,\mathrm{d}t\right], \tag{A.13}$$

where $\Delta X_t = X_t^1 - X_t^2$, $\Delta Y_t, \Delta Z_t, \Delta Z_t^0$ are defined similarly, and $\Delta\mu_t = \mathcal{W}_2(\mu_t^1, \mu_t^2)$.

Proof: The results generalize Theorem 3.1.3, Proposition 3.1.4 and Theorem 3.1.6 in [3] to the multi-dimensional case and with Lipschitz SDE coefficients b, σ, σ^0. The original proofs rely on Theorem 3.1.1 and Theorem 3.1.2 under Assumption **H** in [3]. With the additional conditions on (b, σ, σ^0) in our setting, Assumption **H** of [3] still holds. We omit the details because they essentially parallel the corresponding derivations in [3]. ■

Now we are ready to prove Theorem 3.4.2.

Proof: [Proof of Theorem 3.4.2] The proof uses the estimate (A.13) repeatedly. We first observe that, for $\mu_t = \mathcal{L}(X_t|\mathcal{F}_t^B)$ and $\mu_t' = \mathcal{L}(X_t'|\mathcal{F}_t^B)$, one has

$$\mathbb{E}[\mathcal{W}_2^2(\mu_t, \mu_t')] \leq \mathbb{E}[\|X_t - X_t'\|^2], \quad \forall t \in [0,T]. \tag{A.14}$$

Then we define a map Φ by

$$\mu = \{\mu_t\}_{0 \leq t \leq T} \to \Phi(\mu) := \{\mathcal{L}(X_t^\mu|\mathcal{F}_t^B)\}_{0 \leq t \leq T}, \tag{A.15}$$

where X_t^μ is the optimal controlled process in FBSDE (A.9) given $\mu \in \mathcal{M}([0,T];\mathcal{P}^2(\mathbb{R}^d))$. Combining (A.14) and (A.13) gives

$$\begin{aligned}\sup_{t\in[0,T]} \mathbb{E}[\mathcal{W}_2^2(\Phi(\mu_t), \Phi(\mu_t'))] &\leq \sup_{t\in[0,T]} \mathbb{E}[\|X_t^\mu - X_t^{\mu'}\|^2] \\ &\leq C_{K,T}\mathbb{E}\left[\int_0^T \mathcal{W}_2^2(\mu_t, \mu_t')\,\mathrm{d}t\right] \leq C_{K,T}T \sup_{t\in[0,T]} \mathbb{E}[\mathcal{W}_2^2(\mu_t, \mu_t')].\end{aligned} \tag{A.16}$$

Thus, for sufficiently small T, Φ is a contraction map. By definition, μ_t^* defined in (A.12) is a fixed point of Φ: $\Phi(\mu^*) = \mu^*$. Let $\mu^{(0)}$ be the initial guess of μ^*, and $\mu^{(n)}$ be the resulted flow of measures of X_t given $\tilde{\mu}^{(n-1)}$ which is the approximation of $\mu^{(n-1)}$ by truncated signatures. So the measure flows are generated by

$$\mu^{(0)} \to \mu^{(1)} \rightsquigarrow \tilde{\mu}^{(1)} \to \mu^{(2)} \rightsquigarrow \tilde{\mu}^{(2)} \cdots \to \mu^{(n-1)} \rightsquigarrow \tilde{\mu}^{(n-1)} \to \mu^{(n)} \rightsquigarrow \tilde{\mu}^{(n)} \tag{A.17}$$

where $\to$ corresponds to the map Φ, and $\rightsquigarrow$ corresponds to the truncated signature approximation. Therefore, with (A.16) and the assumption $\sup_{t\in[0,T]} \mathbb{E}[\mathcal{W}_2^2(\tilde{\mu}_t^{(n)}, \mu_t^{(n)})] \leq$

ϵ in Theorem 3.4.2, and denoting by $2C_{K,T}T = q$, we deduce that

$$\begin{aligned}
\sup_{t\in[0,T]} \mathbb{E}[\mathcal{W}_2^2(\tilde{\mu}_t^{(n)}, \mu_t^*)] &\leq 2 \sup_{t\in[0,T]} \mathbb{E}[\mathcal{W}_2^2(\tilde{\mu}_t^{(n)}, \mu_t^{(n)})] + 2 \sup_{t\in[0,T]} \mathbb{E}[\mathcal{W}_2^2(\mu_t^{(n)}, \mu_t^*)] \\
&\leq 2\epsilon + 2C_{K,T}T \sup_{t\in[0,T]} \mathbb{E}[\mathcal{W}_2^2(\tilde{\mu}_t^{(n-1)}, \mu_t^*)] = 2\epsilon + q \sup_{t\in[0,T]} \mathbb{E}[\mathcal{W}_2^2(\tilde{\mu}_t^{(n-1)}, \mu_t^*)] \\
&\leq 2\epsilon + q(2\epsilon + q \sup_{t\in[0,T]} \mathbb{E}[\mathcal{W}_2^2(\tilde{\mu}_t^{(n-2)}, \mu_t^*)]) \\
&\leq \cdots \\
&\leq 2\epsilon(1 + q + q^2 + \ldots q^{n-1}) + q^n \sup_{t\in[0,T]} \mathbb{E}[\mathcal{W}_2^2(\mu_t^{(0)}, \mu_t^*)] \\
&= \frac{2 - 2q^n}{1 - q}\epsilon + q^n \sup_{t\in[0,T]} \mathbb{E}[\mathcal{W}_2^2(\mu_t^{(0)}, \mu_t^*)].
\end{aligned}$$

With sufficiently small T, one has $0 < q < 1$. To estimate $\int_0^T \mathbb{E}|\alpha_t^{(n)} - \alpha_t^*|^2 \,\mathrm{d}t$, we observe that

$$\alpha_t^{(n)} - \alpha_t^* = \hat{\alpha}(t, X_t^{\tilde{\mu}^{(n-1)}}, Y_t^{\tilde{\mu}^{(n-1)}}, \tilde{\mu}_t^{(n-1)}) - \hat{\alpha}(t, X_t^*, Y_t^*, \mu_t^*), \tag{A.18}$$

where $(X_t^{\tilde{\mu}^{(n-1)}}, Y_t^{\tilde{\mu}^{(n-1)}})$ is the solution to FBSDE (A.9) given $\tilde{\mu}^{(n-1)}$, and (X_t^*, Y_t^*) can be viewed as the solution to FBSDE (A.9) given μ^*. Then using the Lipschitz property of $\hat{\alpha}$ in (t, x, μ) and (A.13) again produces

$$\begin{aligned}
\int_0^T \mathbb{E}|\alpha_t^{(n)} - \alpha_t^*|^2 \,\mathrm{d}t &\leq C_{K,T}\mathbb{E}\left[\int_0^T \|X_t^{\tilde{\mu}^{(n-1)}} - X_t^*\|^2 + \|Y_t^{\tilde{\mu}^{(n-1)}} - Y_t^*\|^2 + \mathcal{W}_2^2(\tilde{\mu}_t^{(n-1)}, \mu_t^*) \,\mathrm{d}t\right] \\
&\leq C_{K,T}T \sup_{t\in[0,T]} \mathbb{E}[\mathcal{W}_2^2(\tilde{\mu}_t^{(n-1)}, \mu_t^*)].
\end{aligned}$$

Therefore, we obtain the desired result. ■

Next we give the proof to Theorem 3.4.3. *Proof:* [Proof of Theorem 3.4.3] Consider a partition of $[0, T] : 0 = t_0 < \cdots < t_L = T$, and define $\pi(t) = t_k$ for $t \in [t_k, t_{k+1})$ with $\|\pi\| = \max_{1\leq k<L} |t_k - t_{k-1}|$, then by following the line of the proof to Theorem 3.4.2, one only needs an additional estimate on $\mathbb{E}|X_t^\mu - X_{t_k}^{(n)}|^2$ to complete the proof. Noticing

that X_t solves (3.3) with μ^* and $X_{t_k}^{(n)}$ satisfies (3.10) with $\tilde{\mu}^{(n-1)}$, one can obtain the estimate by following Lemma 14 in [33] with $N = 1$. ■

A.4 Benchmark Solutions

This appendix summarizes the analytical solutions to the three examples in Section 3.5, which are used to benchmark our algorithm's performance.

Linear-Quadratic MFGs. The analytical solution is provided in [32]:

$$m_t := \mathbb{E}[X_t|\mathcal{F}_t^B] = \mathbb{E}[X_0] + \rho\sigma B_t, \quad t \in [0,T], \tag{A.19}$$

$$\alpha_t = (q + \eta_t)(m_t - X_t), \quad t \in [0,T], \tag{A.20}$$

where η_t is a deterministic function solving the Riccati equation:

$$\dot{\eta}_t = 2(a+q)\eta_t + \eta_t^2 - (\epsilon - q^2), \quad \eta_T = c,$$

with the solution given by

$$\eta_t = \frac{-(\epsilon - q^2)(e^{(\delta^+ - \delta^-)(T-t)} - 1) - c(\delta^+ e^{(\delta^+ - \delta^-)(T-t)} - \delta^-)}{(\delta^- e^{(\delta^+ - \delta^-)(T-t)} - \delta^+) - c(e^{(\delta^+ - \delta^-)(T-t)} - 1)}.$$

Here $\delta^{\pm} = -(a+q) \pm \sqrt{R}$, $R = (a+q)^2 + (\epsilon - q^2) > 0$, and the minimized expected cost is $V(0, x_0 - \mathbb{E}[x_0])$ with

$$V(t,x) = \frac{\eta_t}{2}x^2 + \mu_t, \quad \mu_t = \frac{1}{2}\sigma^2(1-\rho^2)\int_t^T \eta_s \,\mathrm{d}s.$$

The benchmark trajectories in Figure 3.2 are simulated according to (3.14) with m_t

and α_t in (A.19) and (A.20).

Mean-field Portfolio Game Given the type vector $\zeta = (\xi, \delta, \theta, \mu, \nu, \sigma)$, the analytical solution provided in [105] is summarized below

$$\begin{aligned} \pi_t^* &= \delta \frac{\mu}{\sigma^2 + \nu^2} + \theta \frac{\sigma}{\sigma^2 + \nu^2} \frac{\phi}{1 - \psi}, \\ m_t &= \mathbb{E}[\xi] + \mathbb{E}[\mu \pi^*] t + \mathbb{E}[\sigma \pi^*] B_t, \end{aligned}$$

where $\phi = \mathbb{E}[\delta \frac{\mu\sigma}{\sigma^2+\nu^2}]$ and $\psi = \mathbb{E}[\theta \frac{\sigma^2}{\sigma^2+\nu^2}]$. Note that, since the type vector ζ is random representing the heterogenuity of agents in this mean-field game, π^* is a random strategy. The maximized expected utility of this game is given by $\mathbb{E}[v(0, \xi - \theta\mathbb{E}[\xi])]$, with

$$v(t, x) = -e^{-x/\delta} e^{-\rho(T-t)},$$

$$\rho = \frac{1}{2(\sigma^2 + \nu^2)} \left(\mu + \frac{\theta}{\delta} \frac{\phi}{1 - \psi} \sigma \right)^2 - \frac{\theta}{\delta} \left(\tilde{\psi} + \frac{\tilde{\phi}\phi}{1 - \psi} \right) - \frac{1}{2} \left(\frac{\theta}{\delta} \frac{\phi}{1 - \psi} \right)^2,$$

$$\tilde{\psi} = \mathbb{E}\left[\delta \frac{\mu^2}{\sigma^2 + \nu^2} \right], \quad \tilde{\phi} = \mathbb{E}\left[\theta \frac{\mu\sigma}{\sigma^2 + \nu^2} \right].$$

Note that Figure 3.3(c) plots the absolute value of $\mathbb{E}[v(0, \xi - \theta\mathbb{E}[\xi])]$.

Mean-field Game of Optimal Consumption and Investment Following [102], the analytical solution is given by

$$\pi_t^* \equiv \pi^* = \frac{\delta\mu}{\sigma^2 + \nu^2} - \frac{\theta(\delta - 1)\sigma}{\sigma^2 + \nu^2} \frac{\phi}{1 + \psi}, \quad c_t^* = \left(\frac{1}{\beta} + (\frac{1}{\lambda} - \frac{1}{\beta}) e^{-\beta(T-t)} \right)^{-1}, \tag{A.21}$$

where

$$\phi = \mathbb{E}\left[\frac{\delta\mu\sigma}{\sigma^2+\nu^2}\right], \quad \psi = \mathbb{E}\left[\frac{\theta(\delta-1)\sigma^2}{\sigma^2+\nu^2}\right], \quad \lambda = \epsilon^{-\delta}\left(e^{\mathbb{E}\left[\log(\epsilon^{-\delta})\right]}\right)^{-\frac{\theta(\delta-1)}{1+\mathbb{E}[\theta(\delta-1)]}},$$
$$\beta = \theta(\delta-1)\frac{\mathbb{E}\left[\delta\rho\right]}{1+\mathbb{E}\left[\theta(\delta-1)\right]} - \delta\rho,$$

and

$$\rho = \left(1-\frac{1}{\delta}\right)\left\{\frac{\delta}{2(\sigma^2+\nu^2)}\left(\mu - \sigma\frac{\phi}{1+\psi}\theta(1-\frac{1}{\delta})\right)^2 + \frac{1}{2}\left(\frac{\phi}{1+\psi}\right)^2\theta^2\left(1-\frac{1}{\delta}\right) \right.$$
$$\left. - \theta\mathbb{E}\left[\frac{\delta\mu^2 - \theta(\delta-1)\sigma\mu\frac{\phi}{1+\psi}}{\sigma^2+\nu^2}\right] + \frac{\theta}{2}\mathbb{E}\left[\frac{(\delta\mu - \theta(\delta-1)\sigma\frac{\phi}{1+\psi})^2}{\sigma^2+\nu^2}\right]\right\}.$$

Note that the expression of m_t, Γ_t and the maximized expected utility are not given in [102]. For completeness, we give their derivations below. Since c_t^* in (A.21) doesn't depend on the common noise B, $\Gamma_t := \exp\mathbb{E}[\log c_t^*|\mathcal{F}_t^B]$ admits a unique formula for all agents

$$\Gamma_t = \exp\mathbb{E}[\log c_t^*].$$

To obtain the formula for $m_t := \exp\mathbb{E}[\log X_t^*|\mathcal{F}_t^B]$, we first deduce by Itô's formula that

$$\mathrm{d}\log X_t^* = \pi_t^*(\mu\,\mathrm{d}t + \nu\,\mathrm{d}W_t + \sigma\,\mathrm{d}B_t) - \frac{1}{2}(2c_t^* + (\pi_t^*)^2\sigma^2 + (\pi_t^*)^2\nu^2)\,\mathrm{d}t, \tag{A.22}$$

from which we easily get

$$\mathbb{E}[\log X_t^*|\mathcal{F}_t^B] = \mathbb{E}[\log\xi] + \mathbb{E}[\pi^*\mu - \frac{1}{2}(\pi^*)^2(\sigma^2+\nu^2)]t - \int_0^t \mathbb{E}[c_s^*]\,\mathrm{d}t + \pi^*\sigma B_t,$$

and $m_t = \exp\mathbb{E}[\log X_t^*|\mathcal{F}_t^B]$. The maximized expected utility of this game is given by

$\mathbb{E}[v(0,\xi,\mathbb{E}[\xi])]$, with

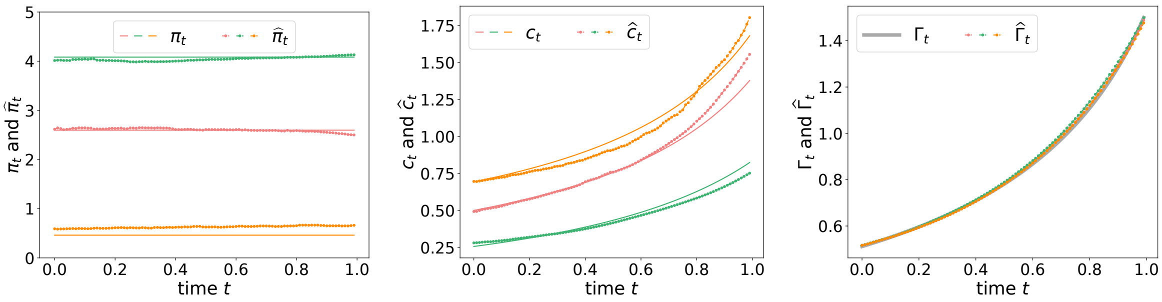
$$v(t,x,y) = \epsilon \left(1 - \frac{1}{\delta}\right)^{-1} x^{1-\frac{1}{\delta}} y^{-\theta(1-\frac{1}{\delta})} f(t),$$

and $f(t)$ is defined by

$$f(t) = \exp\left\{\int_t^T \left(\rho + \frac{1}{\delta}c_s^* + \mathbb{E}[c_s^*]\left(1-\frac{1}{\delta}\right)\theta\right) \mathrm{d}s\right\}.$$

Note that, to ensure the positiveness of X_t required by using the power utility, the trajectories of X_t are obtained by simulating $\log X_t$ via (A.22) then taking the exponential.

A.5 Plots of π_t, c_t, $\Gamma_t = \exp \mathbb{E}(\log c_t|\mathcal{F}_t^B)$ for Mean-Field Game of Optimal Consumption and Investment



(a) π_t (b) c_t (c) $\Gamma_t = \exp \mathbb{E}(\log c_t|\mathcal{F}_t^B)$

Figure A.1: Plots on test data for three different $(X_0^i, W^i, B^i, \zeta^i)$. Solid line is the benchmark solution and dashed line is the numerical approximation using the Sig-DFP algorithm. Each panel presents three trajectories of π_t, c_t, and $\Gamma_t = \exp \mathbb{E}(\log c_t|\mathcal{F}_t^B)$ and their approximations. Parameter choices are: $\delta \sim U(2, 2.5), \mu \sim U(0.25, 0.35), \nu \sim U(0.2, 0.4), \theta, \xi \sim U(0, 1), \sigma \sim U(0.2, 0.4)$, $\epsilon \sim U(0.5, 1)$.

A.6 Experiment setup for the high-dimensional case $n_0 = 5$

To test the performance of Sig-DFP in high dimensions, we implement a toy experiment on the mean-field game of optimal consumption and investment with the common noise of dimension $n_0 = 5$. Specifically, we modify the $\sigma \, \mathrm{d}B_t$ term in (3.18) to be in high dimensions, *i.e.*, X_t now follows

$$\mathrm{d}X_t = \pi_t X_t(\mu \, \mathrm{d}t + \nu \, \mathrm{d}W_t + \boldsymbol{\sigma}^{\mathrm{T}} \, \mathrm{d}\boldsymbol{B}_t) - c_t X_t \, \mathrm{d}t,$$

where $\boldsymbol{\sigma} := (\sigma_1, \ldots, \sigma_5)^{\mathrm{T}}$, $\boldsymbol{B}_t$ is a 5-dimensional Brownian motion, and $X_0 = \xi$. We use the same hyperparameters for training and provide the running time in Table 3.6.

Appendix B

Neural DC-SDEs as Generator for Time Series

B.1 Additional Theorems and Proofs

Given that signature is well-defined and with finite expectation, we call $\mathbb{E}[S(X)]$ the expected signature of X. Intuitively, the expected signature serves the moment-generating function, which can characterize the law induced by a stochastic process under some regularity conditions. More precisely, an immediate consequence of Proposition 6.1 in [41] on the uniqueness of the expected signature is summarized in the below theorem:

Theorem B.1.1 *Let X, Y be two random variables of geometric rough paths such that $\mathbb{E}[S(X)]] = \mathbb{E}[S(Y)]$ and $\mathbb{E}[S(X)]$ has an infinite radius of convergence, then X, Y have the same distribution.*

B.1.1 Proof of Theorem 4.5.2

We first restate Theorem 4.5.2 formally. Without loss of generality, we treat the time-homogeneous case, i.e., μ and σ are independent of t. Our proof relies on constructing the forward equations characterizing marginal distributions of both SDEs and directed chain SDEs, thus can be easily generalized to time-dependence cases. The forward equation associated with directed chain SDEs has been constructed by [83] and will be used directly in our proof.

Theorem B.1.2 *Let $Y \in L^2(\Omega \times [0,T], \mathbb{R}^N)$ be an N-dimensional stochastic process with the following dynamics*

$$\mathrm{d}Y_t = \mu(Y_t)\,\mathrm{d}t + \sigma(Y_t)\,\mathrm{d}B_t^y, \quad Y_0 = \xi^y,$$

where B^y is a standard d-dimensional Brownian motion, and $\mu : \mathbb{R}^N \to \mathbb{R}^N, \sigma : \mathbb{R}^N \to \mathbb{R}^{N\times d}$ are Borel measurable functions with Lipschitz and linear growth conditions. Then, there exist functions V_0 and V_1 such that the process X has the same marginal distribution as Y for all $t \in [0,T]$, where X is described by the following directed chain SDEs with an initial position ξ as an independent copy of ξ^y,

$$\begin{aligned} &\mathrm{d}X_t = V_0(X_t, \tilde{X}_t)\,\mathrm{d}t + V_1(X_t, \tilde{X}_t)\,\mathrm{d}B_t^y, \quad X_0 = \xi, \\ &\textit{subject to: } \mathrm{Law}(X_t, 0 \le t \le T) = \mathrm{Law}(\tilde{X}_t, 0 \le t \le T). \end{aligned}$$

Proof: Let $g \in \mathcal{C}^2(\mathbb{R}^N)$ be a twice continuously differentiable function. To characterize marginal distributions of the SDE solution Y for all $t \in [0,T]$, we use the Kolmogorov forward equations. Define $u(t,x) := \mathbb{E}[g(Y_t)|Y_0 = x]$, it is the solution of the following

Cauchy problem

$$(\partial_t - \mathcal{L})u(t,x) = 0, \tag{B.1}$$

$$u(0,x) = g(x). \tag{B.2}$$

The derivation relies on Itô's formula and can be found in stochastic calculus textbooks, e.g., in [85]. Here the *infinitesimal operator* $\mathcal{L}$ is given by

$$\mathcal{L}g(x) = \mu(x) \cdot \nabla_x g(x) + \frac{1}{2}\mathrm{Tr}(\sigma\sigma^T(x)\mathrm{Hess}_x g(x)),$$

where $\mathrm{Hess}_x(\cdot)$ denotes the Hessian matrix, and $\mathrm{Tr}(\cdot)$ denotes the matrix trace. In [83, Section 4.5], a similar partial differential equation for the directed chain SDEs is derived, and we here summarize a simpler version without the mean-field interaction term. Define $v(t,x) := \mathbb{E}[g(X_t)|X_0 = x]$, then v solves

$$(\partial_t - \mathcal{L}^{dc})v(t,x) = 0, \tag{B.3}$$

$$v(0,x) = g(x). \tag{B.4}$$

Let $\tilde{\xi}$ be an independent copy of ξ, and the differential operator $\mathcal{L}^{dc}$ is given by

$$\mathcal{L}^{dc}g(x) = \mathbb{E}_{\tilde{\xi}}\Big[V_0(x,\tilde{\xi}) \cdot \nabla_x g(x) + \frac{1}{2}\mathrm{Tr}(V_1 V_1^T(x,\tilde{\xi})\mathrm{Hess}_x g(x))\Big], \tag{B.5}$$

where $\mathbb{E}_{\tilde{\xi}}$ is the expectation with respect to the distribution of $\tilde{\xi}$. As long as we can match these two operators $\mathcal{L}$ and $\mathcal{L}^{dc}$ with some non-degenerate choices of V_0, V_1, then (B.1)-(B.2) and (B.3)-(B.4) agree with each other and so do their solutions u and v. To this end, it suffices to choose V_0, V_1 such that

$$\begin{aligned}\mathbb{E}_{\tilde{\xi}}[V_0(x, \tilde{\xi})] &= \mu(x), \\ \mathbb{E}_{\tilde{\xi}}[V_1 V_1^\top(x, \tilde{\xi})] &= \sigma\sigma^\top(x).\end{aligned}$$

A toy example of non-degenerate V_0, V_1 can be $V_0(x, \tilde{\xi}) = \mu(x) + \varphi_1(\tilde{\xi}) - \mathbb{E}_{\tilde{\xi}}[\varphi_1(\tilde{\xi})]$ and $V_1(x, \tilde{\xi})$ such that $V_1 V_1^\top(x, \tilde{\xi}) = \sigma\sigma^\top(x) + \varphi_2(\tilde{\xi}) - \mathbb{E}_{\tilde{\xi}}[\varphi_2(\tilde{\xi})]$ with measurable and integrable functions φ_1, φ_2. ■

B.1.2 Proof of Theorem 4.5.4

From [83, Proposition 2.1], we have the existence and weak uniqueness of directed chain SDEs. Denote this unique measure flow by

$$m := \text{Law}(X_t, 0 \le t \le T) = \text{Law}(\tilde{X}_t, 0 \le t \le T).$$

This measure can also be understood as a probability distribution on $C([0,T], \mathbb{R}^N)$. Given the Brownian motion path and the neighborhood path, we define a map $\Phi : C([0,T], \mathbb{R}^N) \times C([0,T], \mathbb{R}^d) \to C([0,T], \mathbb{R}^N)$ such that

$$X = \Phi(\tilde{X}; B) \in C([0,T], \mathbb{R}^N)$$

and Φ_t as the projection of Φ onto any specific time stamp, i.e. $X_t \equiv \Phi_t(\tilde{X}; B)$. Then, on a chain-like structure depicted in Figure 4.2 or 4.3, we write

$$X_q = \Phi(X_{q-1}; B^q) = \Phi(\Phi(X_{q-2}; B^{q-1}); B^q) = \Phi \circ \Phi(X_{q-2}; B^{q-1}, B^q).$$

Namely, X_q is obtained as an output of the composite map $\Phi \circ \Phi$ from the inputs X_{q-2}, B^{q-1} and B^q. Repeating the above equation until tracing back to the first node produces

$$X_q = \Phi \circ \cdots \circ \Phi(X_1; B^2, \dots, B^q) := \Phi^q(X_1; \mathbf{B}),$$

where $\mathbf{B} = (B^2, \dots, B^q)$ and $B^2, \dots, B^q$ are independent d-dimensional Brownian motions. Such a chain-like structure possesses *local Markov property* as pointed out in Proposition 4.6 in [83]. Let us denote $X_{t,q} = \Phi_t^q(X_1; \mathbf{B})$. In the proof below, we impose Lipschitz and linear growth conditions on coefficients V_0 and V_1.

Assumption B.1.3 *For both coefficients V_0 and V_1, there exists a positive constant C_T such that,*

1. *(Lipschitz conditions) for $i = 0, 1$,*

$$|V_i(x_1, y_1) - V_i(x_2, y_2)| \leq C_T(|x_1 - x_2| + |y_1 - y_2|);$$

2. *(Linear growth conditions) for $i = 0, 1$,*

$$V_i(x, y) \leq C_T(1 + |x| + |y|).$$

The following lemma gives the necessity of having the Brownian motion noises B^j, $j = 2, \dots, q$ in Φ_t^q, in order to have dependence decay properties.

Lemma B.1.4 *Suppose Assumption B.1.3 holds. In the degenerate case, i.e., $V_1 \equiv 0$ and $X_{t,q} = \Phi_t^q(X_1)$, if all the initial conditions $X_{0,1} = X_{0,2} = \cdots = X_{0,q} = \xi$ are identical, then the directed chain SDE satisfy $X_1 = X_2 = \cdots X_q$ in the L^2 sense.*

Proof: We first write our directed chain dynamics in the integral form,

$$X_{t,q} = \xi + \int_0^t V_0(X_{s,q}, X_{s,q-1})\,\mathrm{d}s. \tag{B.6}$$

Note that the current directed chain system with degenerate V_1 also has unique solutions. By the Lipschitz property on V_0, we compute

$$\begin{aligned}
\mathbb{E}[\sup_{0\le s\le t} |X_{s,q} - X_{s,q-1}|^2] &\le \mathbb{E}\Big[\sup_{0\le s\le t} 2C_T \int_0^s (|X_{v,q} - X_{v,q-1}|^2 + |X_{v,q-1} - X_{v,q-2}|^2)\,\mathrm{d}v\Big] \\
&\le C\cdot\mathbb{E}\Big[\int_0^t \sup_{0\le v\le s}\Big(|X_{v,q} - X_{v,q-1}|^2 + |X_{v,q-1} - X_{v,q-2}|^2\Big)\,\mathrm{d}v\Big] \\
&\le C\cdot\int_0^t \mathbb{E}\big[\sup_{0\le v\le s}|X_{v,q} - X_{v,q-1}|^2\big]\,\mathrm{d}v \\
&\qquad + C\cdot\int_0^t \mathbb{E}\big[\sup_{0\le v\le s}|X_{v,q-1} - X_{v,q-2}|^2\big]\,\mathrm{d}v \\
&\le C\cdot e^{CT}\int_0^t \mathbb{E}\big[\sup_{0\le v\le s}|X_{v,q-1} - X_{v,q-2}|^2\big]\,\mathrm{d}v,
\end{aligned}$$

where the third inequality comes from Fubini's theorem and Proposition 2.2 in [83], and the last inequality follows from Gronwall's inequality. Iterating back to the beginning of the chain, we deduce

$$\mathbb{E}[\sup_{0\le s\le T} |X_{s,q} - X_{s,q-1}|^2] \le \frac{TC^{q-1}e^{(q-1)CT}}{(q-1)!}\mathbb{E}[\sup_{0\le s\le T}|X_{s,2} - X_{s,1}|^2].$$

According to the invariance of (joint) distribution (see [53, 83]), we get

$$\mathbb{E}[\sup_{0\le s\le T} |X_{s,2} - X_{s,1}|^2] \le \frac{TC^{q-1}e^{(q-1)CT}}{(q-1)!}\mathbb{E}[\sup_{0\le s\le T}|X_{s,2} - X_{s,1}|^2].$$

The constant q can be arbitrarily large and hence the above inequality forms a contraction, which implies

$$\mathbb{E}[\sup_{0\le s\le T}|X_{s,2}-X_{s,1}|^2]=0.$$

We then conclude $X_1 = X_2 = \cdots = X_q$ in the L^2 sense. ■

Although the assumption of identical initials in Lemma B.1.4 is different from the general setting of directed chain SDEs, where initials should be i.i.d, it is consistent in the case that initials are deterministic. Therefore, the existence of non-degenerate V_1 becomes crucial, and we give the following necessary assumptions for factorial dependence decay property.

Definition B.1.5 **($\mathcal{C}^{k,k}_{b,\mathrm{Lip}}$)** *We have the following definition for $\mathcal{C}^{k,k}_{b,\mathrm{Lip}}$:*

(a) We use ∂_x, ∂_y to denote the derivative with respect to the first and second Euclidean variables in V_0, V_1.

(b) Let $V : \mathbb{R}^N \times \mathbb{R}^N \to \mathbb{R}^N$ with components $V^1, \dots, V^N : \mathbb{R}^N \times \mathbb{R}^N \to \mathbb{R}$. We say $V \in \mathcal{C}^{1,1}_{b,\mathrm{Lip}}(\mathbb{R}^N \times \mathbb{R}^N; \mathbb{R}^N)$ if the following is true: for each $i = 1, \dots, N$, $\partial_x V^i, \partial_y V^i$ exist. Moreover, assume the boundedness of the derivatives for all $(x, y) \in \mathbb{R}^N \times \mathbb{R}^N$,

$$|\partial_x V^i(x,y)| + |\partial_y V(x,y)| \le C.$$

In addition, suppose that $\partial_x V^i, \partial_y V^i$ are all Lipschitz in the sense that for all $(x, y) \in \mathbb{R}^N \times \mathbb{R}^N$,

$$|\partial_x V^i(x,y) - \partial_x V^i(x',y')| \le C(|x-x'| + |y-y'|),$$
$$|\partial_y V^i(x,y) - \partial_y V^i(x',y')| \le C(|x-x'| + |y-y'|),$$

and $V^i, \partial_x V^i, \partial_y V^i$ all have linear growth property,

$$|V^i(x,y)| + |\partial_x V^i(x,y)| + |\partial_y V^i(x,y)| < C_T(1+|x|+|y|),$$

where C_T is a constant depending only on T.

(c) *We write $V \in \mathcal{C}^{k,k}_{b,\mathrm{Lip}}(\mathbb{R}^N \times \mathbb{R}^N; \mathbb{R}^N)$, if the following holds: for each $1, \dots, N$, and all multi-indices α, β on $\{1, \dots, N\}$ satisfying $|\alpha| + |\beta| \le k$, the derivative $\partial^\alpha_x \partial^\beta_y$ exists and is bounded, Lipschitz continuous, and satisfies linear growth condition.*

(d) *We say $V_0 \in \mathcal{C}^{k,k}_{b,\mathrm{Lip}}(\mathbb{R}^N \times \mathbb{R}^N)$ for short if $V_0 : \mathbb{R}^N \times \mathbb{R}^N \to \mathbb{R}^N$ satisfies (c). Let $V_1 : \mathbb{R}^N \times \mathbb{R}^N \to \mathbb{R}^{N\times d}$ with components $V_1^1, \dots, V_1^d : \mathbb{R}^N \times \mathbb{R}^N \to \mathbb{R}^N$. We say $V_1 \in \mathcal{C}^{k,k}_{b,\mathrm{Lip}}(\mathbb{R}^N \times \mathbb{R}^N)$ for short if $V_1^j \in \mathcal{C}^{k,k}_{b,\mathrm{Lip}}(\mathbb{R}^N \times \mathbb{R}^N)$ for every $j = 1, \dots, d$.*

Assumption B.1.6 *We emphasize two assumptions used for the existence and smoothness of the marginal densities of directed chain SDEs:*

1. *(Uniform ellipticity on V_1) Assume that there exists $\epsilon > 0$ such that for all $\eta, x, \tilde{x} \in \mathbb{R}^N$,*

$$\eta^\top V_1(x, \tilde{x}) V_1(x, \tilde{x})^\top \eta \ge \epsilon |\eta|^2.$$

2. *(Smoothness on V_0, V_1) Assume that $V_0, V_1 \in \mathcal{C}^{k,k}_{b,\mathrm{Lip}}(\mathbb{R}^N, \mathbb{R}^N)$ with $k \ge N + 2$, where $V_0, V_1 \in \mathcal{C}^{k,k}_{b,\mathrm{Lip}}(\mathbb{R}^N, \mathbb{R}^N)$ is defined in Definition B.1.5.*

Under Assumption B.1.6, one can prove the existence of the density function of directed chain SDEs [83, Theorem 4.3].

Theorem B.1.7 *Suppose Assumption B.1.6 is satisfied. For every Lipschitz function $\varphi : \mathbb{R}^N \to \mathbb{R}$ with Lipschitz constant K, there exists a constant $c > 0$ such that the difference between the conditional expectation of $\varphi(X_{t,q})$, given X_1 and the unconditional expectation $\varphi(X_{t,q})$ for all $t \in [0, T]$ is bounded, i.e.,*

$$\mathbb{E}\big[\sup_{0 \le t \le T} \big|\mathbb{E}[\varphi(X_{t,q})|X_1] - \mathbb{E}[\varphi(X_{t,q})]\big|^2\big] \le \frac{c^{q-1}}{(q-1)!}. \tag{B.7}$$

We shall first provide some interpretations for Theorem B.1.7 before giving the proof. For random variables in space $C([0,T],\mathbb{R}^N)$, there is no unique choice on how to measure their correlation or covariance. Here, we measure the difference between conditional expectation and unconditional expectation over a family of testing functions φ. Thus, we use the left-hand side in inequality (B.7) to measure the dependence between X_q and X_1.

Proof: Note that Assumption B.1.6 is a stronger version of Assumption B.1.3, and it not only ensures the existence and weak uniqueness of the solution, but also guarantees the existence of a smooth density which excludes the case of deterministic $X_{t,q}$. If X_q and X_1 are independent, the left-hand side is zero for every Lipschitz function φ. The vice versa is also correct because of the exclusion of the deterministic case.

Let us start from the left-hand side in (B.7), the difference between the conditional expectation of φ and unconditional expectation can be bounded by

$$
\begin{aligned}
&\mathbb{E}\big[\sup_{0\le t\le T}\big|\mathbb{E}[\varphi(X_{t,q})|X_1]-\mathbb{E}[\varphi(X_{t,q})]\big|^2\big]\\
&=\int_{C([0,T],\mathbb{R}^N)}\sup_{0\le t\le T}\bigg|\int_{C([0,T],\mathbb{R}^N)}\Big(\mathbb{E}_{\mathbf{B}}\big[\varphi(\Phi_t^q(\omega;\mathbf{B}))\big]-\mathbb{E}_{\mathbf{B}}\big[\varphi(\Phi_t^q(\tilde\omega;\mathbf{B}))\big]\Big)m(\,\mathrm{d}\tilde\omega)\bigg|^2 m(\,\mathrm{d}\omega)\\
&\le\int_{C([0,T],\mathbb{R}^N)^2}\sup_{0\le t\le T}\mathbb{E}_{\mathbf{B}}\big[\big|\varphi(\Phi_t^q(\omega;\mathbf{B}))-\varphi(\Phi_t^q(\tilde\omega;\mathbf{B}))\big|^2\big]m(\,\mathrm{d}\tilde\omega)m(\,\mathrm{d}\omega)\\
&\le K^2\int_{C([0,T],\mathbb{R}^N)^2}\mathbb{E}_{\mathbf{B}}\big[\sup_{0\le t\le T}\big|\Phi_t^q(\omega;\mathbf{B})-\Phi_t^q(\tilde\omega;\mathbf{B})\big|^2\big]m(\,\mathrm{d}\tilde\omega)m(\,\mathrm{d}\omega)\\
&\le\frac{c^{q-1}}{(q-1)!},
\end{aligned}
$$

for some positive constant c, where the proof of the last inequality is verbatim to the procedures in Lemma B.1.4. ■

Remark B.1.8 *We shall emphasize that the assumption $X_{0,1}=X_{0,2}=\cdots=X_{0,q}=\xi$ is not allowed under directed chain framework except for $\xi\equiv x\in\mathbb{R}^N$ (the deterministic* *initial condition). This is quite common in practice, for instance, the investment returns usually start from 1. Given results from Lemma B.1.4 and equation* (B.7)*, we are able to conclude that*

$$\mathbb{E}[\sup_{0\leq t\leq T}|\varphi(X_1)-\mathbb{E}[\varphi(X_1)]|^2]\leq\frac{C^{q-1}}{(q-1)!}.$$

Here q can be arbitrarily large, hence we conclude that $\mathbb{E}[\sup_{0\leq t\leq T}|\varphi(X_1)-\mathbb{E}[\varphi(X_1)]|^2]=0$*. The only possible solution for a directed-chain system is the deterministic case where we have deterministic initial conditions and degenerate V_1 (or we should call it "ODE"). Brownian motion is the key ingredient to enrich the representability of our directed-chain systems.*

B.2 Experimental Details

Both discriminative and predictive metrics involve training tasks, and we shall first list all implementing details of these metrics, which is universal for all experiments. Then, we provide training hyper-parameters and training details used in different experiments.

B.2.1 Metrics

Discriminative Metric. We first generate the same amount of fake data paths as true data paths to avoid imbalance, and choose 80% from both real and fake data as training data, leaving the rest 20% as testing data. We use a two-layer LSTM classifier with `channels/2` as the size of the hidden state, where `channels` is the dimension of generated and real series. We will minimize the cross-entropy loss, and the optimization is done by Adam optimizer with a learning rate of 0.001 for 5000 iterations. The discriminative score is calculated by the difference between 0.5 and the prediction accuracy on testing data.

Predictive Metric. We first generate the same amount of fake data as true data, and use it as training data for the predictive metric, whereas true data is for testing. We use a two-layer LSTM sequential predictor with `channels/2` as the size of the hidden state, where `channels` is the dimension of generated and real series. Our objective function is L^1 distance between predicted sequences and true sequences. The predictor generates one-step future predictions in the last feature with the others as input. Optimization is done by Adam optimizer with a learning rate of 0.001 for 5000 iterations. The predictive score is reported as the L^1 distance (also interpreted as mean absolute error (MAE)) between the predictive sequences and true sequences on testing data.

Independence Metric. The independence score is computed by the maximum of the L^1 distance of cross-correlation matrices over the time period $[0, T]$. In practice, we consider the maximum over the time stamps $t \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$.

B.2.2 Experiments

In all four experiments, we use feed-forward neural networks with two hidden layers of sizes [128,128] to parameterize the drift V_0 and diffusion coefficient V_1. For the purpose of fair comparisons, we use the same GAN structure for both neural SDEs and DC-GANs, i.e., the same Sig-Wasserstein GAN setup (4.42) or the combination of neural CDE and Sig-WGAN scheme (4.44) as discriminators. We remark that DC-GANs can be adapted to `torchsde`[1] framework and use their adjoint method for back-propagation.

Stochastic Opinion Dynamics. In this experiment, we only use Sig-Wasserstein GAN approach for the discriminator, and choose $m = 8$ as the truncation depth in (4.42). We choose $N = 1$ and $d = 3$ dimensional standard Brownian motion in the

[1] See the Python package `https://github.com/google-research/torchsde`.

DC-GANs generator (4.41), a batch size of 1024, a learning rate of 0.001 decaying to one-tenth for every 500 steps, and train a total of 2000 steps. Training data and testing data are sampled by the Euler scheme (4.41) with a sample size of 8192, and their initial distributions ξ are drawn independently from a uniform distribution on $[-2, 2]$.

Stochastic FitzHugh-Nagumo Model. The stochastic FitzHugh-Nagumo model is widely used in neuroscience for describing the neurons' interacting spiking, in particular, to capture the multimodality of neurons' interspike interval distribution. For N neurons and P different neuron populations, we denote by $p(i) = \alpha, \alpha \in \{1, \dots, P\}$, the population of i-th particle belongs to, for $i \in \{1, \dots, N\}$. The state vector $(X_t^{t,N})_{t\in[0,T]} = (V_t^{i,N}, w_t^{i,N}, y_t^{i,N})_{t\in[0,T]}$ of neural i follows a three-dimensional SDE:

$$\begin{aligned}\mathrm{d}X_t^{t,N} &= f_\alpha(t, X_t^{t,N})\,\mathrm{d}t + g_\alpha(t, X_t^{t,N})\begin{bmatrix}\mathrm{d}W_t^i \\ \mathrm{d}W_t^{i,y}\end{bmatrix} \\ &\quad + \sum_{\gamma=1}^{P}\frac{1}{N_\gamma}\sum_{j,p(j)=\gamma}\Big(b_{\alpha\gamma}(X_t^{i,N}, X_t^{j,N})\,\mathrm{d}t + \beta_{\alpha\gamma}(X_t^{i,N}, X_t^{j,N})\,\mathrm{d}W_t^{i,\gamma}\Big),\end{aligned}$$

where N_γ denotes the number of neurons in population γ. For all γ and $\alpha \in \{1, \dots, P\}$, $I^\alpha(t) := I$, $\forall t \in [0, T]$, $\forall\alpha$ for some constant value I, f_α, g_α, $b_{\alpha\gamma}$ and $\beta_{\alpha\gamma}$ are given by

$$f_\alpha(t, X_t^{i,N}) = \begin{bmatrix} V_t^{i,N} - \frac{(V_t^{i,N})^3}{3} - w_t^{i,N} + I^\alpha(t) \\ c_\alpha(V_t^{i,N} + a_\alpha - b_\alpha w_t^{i,N}) \\ a_r^\alpha S_\alpha(V_t^{i,N})(1 - y_t^{i,N}) - a_d^\alpha y_t^{i,N} \end{bmatrix},$$

$$g_\alpha(t, X_t^{i,N}) = \begin{bmatrix} \sigma_{\text{ext}}^\alpha & 0 \\ 0 & 0 \\ 0 & \sigma_\alpha^y(V_t^{i,N}, y_t^{i,N}) \end{bmatrix},$$

and

$$b_{\alpha\gamma}(X_t^{i,N}, X_t^{j,N}) = \begin{bmatrix} -\bar{J}_{\alpha\gamma}(V_t^{i,N} - V_{\text{rev}}^{\alpha\gamma})y_t^{i,N} \\ 0 \\ 0 \end{bmatrix},$$

$$\beta_{\alpha\gamma}(X_t^{i,N}, X_t^{j,N}) = \begin{bmatrix} -\sigma^J_{\alpha\gamma}(V_t^{i,N} - V_{\text{rev}}^{\alpha\gamma})y_t^{i,N} \\ 0 \\ 0 \end{bmatrix}.$$

The functions S_α, $\mathcal{X}$ and σ^y_α are defined as

$$\begin{aligned} S_\alpha(V_t^{i,N}) &= \frac{T^\alpha_{\max}}{1 + e^{-\gamma_\alpha(V_t^{i,N} - V_T^{i,N})}}, \\ \mathcal{X}(y_t^{i,N}) &= \mathbb{1}_{y_t^{i,N} \in (0,1)} \Gamma e^{-\Lambda/(1-(2y_t^{i,N}-1)^2)}, \\ \sigma^y_\alpha(V_t^{i,N}, y_t^{i,N}) &= \sqrt{a_r^\alpha S_\alpha(V_t^{i,N})(1 - y_t^{i,N}) + a_d^\gamma y_t^{i,N}} \times \mathcal{X}(y_t^{i,N}), \end{aligned}$$

where $(W^i, W^{i,y}, W^{i,\gamma}), i = 1, \ldots, N$ are standard three-dimensional Brownian motions that are mutually independent. For sample paths produced by this model, we follow the parameter choices in line with [55],

$$\begin{array}{lllllll} V_0 = 0, & \sigma_{V_0} = 0.4, & a = 0.7, & b = 0.8, & c = 0.08, & I = 0.5, & \sigma_{ext} = 0.5, \\ w_0 = 0.5, & \sigma_{w_0} = 0.4, & V_{rev} = 1, & a_r = 1, & a_d = 1, & T_{max} = 1, & \lambda = 0.2, \\ y_0 = 0.3, & \sigma_{y_0} = 0.05, & J = 1, & \sigma_j = 0.2, & V_T = 2, & \Gamma = 0.1, & \Lambda = 0.5. \end{array}$$

The above choice produces the joint multimodal distribution of V and w; see the figure below.

All training and testing data are generated through the Euler scheme with the above parameters. In the training phase, we choose the Sig-Wasserstein GAN approach again for our discriminator and choose $m = 6$ as the truncation depth in (4.42). We take $N = 3$

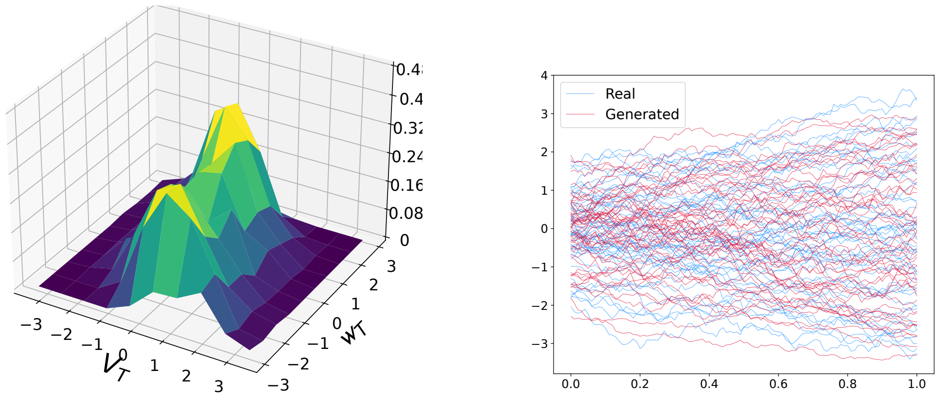


Figure B.1: Stochastic FitzHugh-Nagumo Model (Example 2). Left subfigure shows the multimodal joint density of V_T and w_T, and right subfigure shows the sample paths from the time-dependent model (blue) and from the DC-GANs (red).

and $d = 5$ in our DC-GANs generator, and use a batch size of 1024 for training 2000 steps with a learning rate of 0.001 decaying to one-tenth every 500 steps. Training and testing data are generated by the Euler scheme, where the initial positions ξ are drawn from a 3-dimensional Gaussian random variable with means $(0, 0.5, 0.3)$ and standard deviations $(0.4, 0.4, 0.05)$.

Stock Price Time Series. In this real-world example, we use the six-dimensional stock price data of Google from 2004 to 2019. We segment them into sequences of length 24, which results in 3773 sequences as our time series data set. The combination of Neural CDEs and Signature MMD (4.44) is used as the discriminator. For the purpose of a fair comparison, we use the same noise size d and discriminator setup for both Neural SDEs and DC-GANs generators. In particular, for the Neural CDEs discriminator, we set the dimension of the hidden process to be 16, and their coefficients are approximated by a feed-forward neural network with two hidden layers of size [128, 128]. For both DC-GANs and Neural SDEs generator, the Brownian motion's dimension is set at $d = 10$; and for the Neural SDEs which embed stock prices data into a hidden space, we set its (the hidden space) dimension at 12. The batch size is chosen to be 128. Both generators and discriminators are trained using Adam optimizer. Both learning rates start at 0.0001 and decay to one-tenth after 2000 steps, the signature depth is chosen at $m = 4$ in (4.44) to alleviate the dimensional burden, and training steps are set as 4000. Our CTFP implementation follows the setup in [50], SigWGAN follows from [125] and TimeGAN implementation follows the setup in [149].

Energy Consumption. In the real-world energy consumption example, we choose four electric and gas consumption time series from 02/2011-02/2013 and use daily data as a single time series, bringing 694 sequences with a length of 96. For both neural SDEs and DC-GANs, we use a ten-dimensional Brownian motion and neural nets with two hidden layers of size [128, 128] to estimate drift and diffusion coefficients. The batch size is 128, the training step is 4000, and the learning rate for the generator starts at 0.0001. In the case of using Neural CDEs as the discriminator, we use hidden size 16, [128, 128] as the hidden layers of neural nets estimating coefficients and 0.0001 as the starting learning rate of the discriminator. In the case of using SigWGAN, we consider signature depth 6. All learning rates decay to one-tenth after 2000 steps. Our CTFP implementation follows the setup in [50], SigWGAN follows from [125] and TimeGAN follows from [149].

Bibliography

[1] Yves Achdou, Martino Bardi, and Marco Cirant. Mean field games models of segregation. *Mathematical Models and Methods in Applied Sciences*, 27(01):75–113, 2017.

[2] Yves Achdou and Jean-Michel Lasry. Mean field games for modeling crowd motion. In *Contributions to partial differential equations and applications*, pages 17–42. Springer, 2019.

[3] Saran Ahuja. *Mean Field Games with Common Noise*. PhD thesis, Stanford University, 2015.

[4] Saran Ahuja. Wellposedness of mean field games with common noise under a weak monotonicity condition. *SIAM Journal on Control and Optimization*, 54(1):30–48, 2016.

[5] Andrea Angiuli, Jean-Pierre Fouque, and Mathieu Laurière. Unified reinforcement Q-learning for mean field game and control problems. *arXiv preprint arXiv:2006.13912*, 2020.

[6] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International conference on machine learning*, pages 214–223. PMLR, 2017.

[7] Imanol Perez Arribas. Derivatives pricing using signature payoffs, 2018.

[8] Javier Baladron, Diego Fasoli, Olivier Faugeras, and Jonathan Touboul. Mean-field description and propagation of chaos in networks of hodgkin-huxley and fitzhugh-nagumo neurons. *The Journal of Mathematical Neuroscience*, 2(1):1–50, 2012.

[9] Mustafa Baydogan. *Multivariate Time Series Classification Datasets*, 2015. `http://mustafabaydogan.com`, [Accessed: 2020-07-12].

[10] Erhan Bayraktar, Suman Chakraborty, and Ruoyu Wu. Graphon mean field systems. *arXiv preprint arXiv:2003.13180*, 2020.

[11] Erhan Bayraktar and Ruoyu Wu. Graphon particle system: Uniform-in-time concentration bounds. *arXiv preprint arXiv:2105.11040*, 2021.

[12] Alain Bensoussan, Jens Frehse, and Sheung Chi Phillip Yam. The master equation in mean field theory. *Journal de Mathématiques Pures et Appliquées*, 103(6):1441–1474, 2015.

[13] Horatio Boedihardjo, Xi Geng, Terry Lyons, and Danyu Yang. The signature of a rough path: uniqueness. *Advances in Mathematics*, 293:720–737, 2016.

[14] Tim Bollerslev. Generalized autoregressive conditional heteroskedasticity. *Journal of econometrics*, 31(3):307–327, 1986.

[15] Ariela Briani and Pierre Cardaliaguet. Stable solutions in potential mean field game systems. *Nonlinear Differential Equations and Applications NoDEA*, 25(1):1–26, 2018.

[16] Eoin Brophy, Zhengwei Wang, Qi She, and Tomás Ward. Generative adversarial networks in time series: A systematic literature review. *ACM Computing Surveys (CSUR)*, 2022.

[17] George W Brown. Some notes on computation of games solutions. Technical report, RAND CORP SANTA MONICA CA, 1949.

[18] George W Brown. Iterative solution of games by fictitious play. *Activity analysis of production and allocation*, 13(1):374–376, 1951.

[19] Gerard Brunick and Steven Shreve. Mimicking an itô process by a solution of a stochastic differential equation. *The Annals of Applied Probability*, 23(4):1584–1628, 2013.

[20] Rainer Buckdahn, Juan Li, Shige Peng, and Catherine Rainer. Mean-field stochastic differential equations and associated PDEs. *The Annals of Probability*, 45(2):824 – 878, 2017.

[21] Peter E. Caines and Minyi Huang. Graphon mean field games and the GMFG equations. In *2018 IEEE Conference on Decision and Control (CDC)*, pages 4129–4134, 2018.

[22] Peter E. Caines and Minyi Huang. Graphon mean field games and the GMFG equations: ϵ-nash equilibria. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, pages 286–292, 2019.

[23] Peter E Caines, Minyi Huang, and Roland P Malhamé. Mean field games. In *Handbook of Dynamic Game Theory*, pages 1–28. Springer, 2017.

[24] Pierre Cardaliaguet. *Notes on Mean Field Games. From P.-L.Lions lectures at College de France (2010).* Online Notes Available at, 2010.

[25] Pierre Cardaliaguet, François Delarue, Jean-Michel Lasry, and Pierre-Louis Lions. *The Master Equation and the Convergence Problem in Mean Field Games:(AMS-201)*, volume 201. Princeton University Press, 2019.

[26] Pierre Cardaliaguet and Saeed Hadikhanloo. Learning in mean field games: the fictitious play. *ESAIM: Control, Optimisation and Calculus of Variations*, 23(2):569–591, 2017.

[27] Pierre Cardaliaguet and Charles-Albert Lehalle. Mean field game of controls and an application to trade crowding. *Mathematics and Financial Economics*, 12(3):335–363, 2018.

[28] René Carmona, Daniel B Cooney, Christy V Graves, and Mathieu Lauriere. Stochastic graphon games: I. the static case. *Mathematics of Operations Research*, 47(1):750–778, 2022.

[29] René Carmona and François Delarue. *Probabilistic Theory of Mean Field Games with Applications I*. Springer, 2018.

[30] René Carmona and François Delarue. *Probabilistic Theory of Mean Field Games with Applications II*. Springer, 2018.

[31] René Carmona, François Delarue, and Daniel Lacker. Mean field games with common noise. *Annals of Probability*, 44(6):3740–3803, 2016.

[32] René Carmona, Jean-Pierre Fouque, and Li-Hsien Sun. Mean field games and systemic risk. *Communications in Mathematical Sciences*, 13(4):911–933, 2015.

[33] René Carmona and Mathieu Laurière. Convergence analysis of machine learning algorithms for the numerical solution of mean field control and games: II–the finite horizon case. *arXiv preprint arXiv:1908.01613*, 2019.

[34] René Carmona and Xiuneng Zhu. A probabilistic approach to mean field games with major and minor players. *Annals of Applied Probability*, 26(3):1535–1580, 2016.

[35] René Carmona and François Delarue. Forward-backward stochastic differential equations and controlled McKean-Vlasov dynamics. *The Annals of Probability*, 43(5):2647 – 2700, 2015.

[36] Thomas Cass. Smooth densities for solutions to stochastic differential equations with jumps. *Stochastic Processes and their Applications*, 119(5):1416–1435, 2009.

[37] Jean-François Chassagneux, Dan Crisan, and François Delarue. A probabilistic approach to classical solutions of the master equation for large population equilibria. *arXiv preprint arXiv:1411.3009*, 2014.

[38] Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018.

[39] Ilya Chevyrev and Terry Lyons. Characteristic functions of measures on geometric rough paths. *The Annals of Probability*, 44(6):4049?4082, Nov 2016.

[40] Ilya Chevyrev and Terry Lyons. Characteristic functions of measures on geometric rough paths. *The Annals of Probability*, 44(6):4049–4082, 2016.

[41] Ilya Chevyrev and Harald Oberhauser. Signature moments to characterize laws of stochastic processes. *Journal of Machine Learning Research*, 23(176):1–42, 2022.

[42] Dan Crisan, Paul Dobson, and Michela Ottobre. Uniform in time estimates for the weak error of the euler method for sdes and a pathwise approach to derivative estimates for diffusion semigroups. *Transactions of the American Mathematical Society*, 374(5):3289–3330, 2021.

[43] Dan Crisan, Christian Litterer, and Terry Lyons. Kusuoka–stroock gradient bounds for the solution of the filtering equation. *Journal of Functional Analysis*, 268(7):1928–1971, 2015.

[44] Dan Crisan and Eamon McMurray. Smoothing properties of McKean–Vlasov sdes. *Probability Theory and Related Fields*, 171(1):97–148, 2018.

[45] Dan Crisan and Eamon McMurray. Cubature on Wiener space for McKean–Vlasov SDEs with smooth scalar interaction. *The Annals of Applied Probability*, 29(1):130 – 177, 2019.

[46] Dan Crisan and Salvador Ortiz-Latorre. A high order time discretization of the solution of the non-linear filtering problem. *Stochastics and Partial Differential Equations: Analysis and Computations*, 8(4):693–760, 2020.

[47] Christa Cuchiero, Wahid Khosrawi, and Josef Teichmann. A generative adversarial network approach to calibration of local stochastic volatility models. *Risks*, 8(4):101, 2020.

[48] George Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314, 1989.

[49] Angus Dempster, François Petitjean, and Geoffrey I. Webb. Rocket: exceptionally fast and accurate time series classification using random convolutional kernels. *Data Mining and Knowledge Discovery*, 34(5):1454–1495, 2020.

[50] Ruizhi Deng, Bo Chang, Marcus A Brubaker, Greg Mori, and Andreas Lehrmann. Modeling continuous stochastic processes with dynamic normalizing flows. *Advances in Neural Information Processing Systems*, 33:7805–7815, 2020.

[51] Nils Detering, Jean-Pierre Fouque, and Tomoyuki Ichiba. Directed chain stochastic differential equations, 2018.

[52] Nils Detering, Jean-Pierre Fouque, and Tomoyuki Ichiba. Directed chain stochastic differential equations. *Stochastic Processes and their Applications*, 130(4):2519–2551, 2020.

[53] Nils Detering, Jean-Pierre Fouque, and Tomoyuki Ichiba. Directed chain stochastic differential equations. *Stochastic Processes and their Applications*, 130(4):2519–2551, 2020.

[54] Boualem Djehiche, Alain Tcheukam, and Hamidou Tembine. A mean-field game of evacuation in multilevel building. *IEEE Transactions on Automatic Control*, 62(10):5154–5169, 2017.

[55] Gonçalo dos Reis, Stefan Engelhardt, and Greig Smith. Simulation of McKean–Vlasov SDEs with super-linear growth. *IMA Journal of Numerical Analysis*, 42(1):874–922, 01 2021.

[56] Romuald Elie, Emma Hubert, and Gabriel Turinici. Contact rate epidemic control of COVID-19: an equilibrium view. *Mathematical Modelling of Natural Phenomena*, 15:35, 2020.

[57] Romuald Elie, Julien Pérolat, Mathieu Laurière, Matthieu Geist, and Olivier Pietquin. On the convergence of model free learning in mean field games. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7143–7150, 2020.

[58] Robert F Engle. Autoregressive conditional heteroscedasticity with estimates of the variance of united kingdom inflation. *Econometrica: Journal of the Econometric Society*, pages 987–1007, 1982.

[59] Yichen Feng, Jean-Pierre Fouque, and Tomoyuki Ichiba. Linear-quadratic stochastic differential games on directed chain networks. *Journal of mathematics and statistical science*, 7(2), 2021.

[60] Yichen Feng, Jean-Pierre Fouque, and Tomoyuki Ichiba. Linear-quadratic stochastic differential games on directed chain networks. *Journal of mathematics and statistical science*, 7(2), 2021.

[61] Yichen Feng, Jean-Pierre Fouque, and Tomoyuki Ichiba. Linear-quadratic stochastic differential games on random directed networks. *Journal of mathematics and statistical science*, 7(3), 2021.

[62] Yichen Feng, Jean-Pierre Fouque, and Tomoyuki Ichiba. Linear-quadratic stochastic differential games on random directed networks. *Journal of mathematics and statistical science*, 7(3), 2021.

[63] Yichen Feng, Ming Min, and Jean-Pierre Fouque. Deep learning for systemic risk measures. In *Proceedings of the Third ACM International Conference on AI in Finance*, pages 62–69, 2022.

[64] Peter K. Friz and Nicolas B. Victoir. *Multidimensional Stochastic Processes as Rough Paths: Theory and Applications*. Cambridge Studies in Advanced Mathematics. Cambridge University Press, 2010.

[65] Ken-Ichi Funahashi. On the approximate realization of continuous mappings by neural networks. *Neural Networks*, 2(3):183 – 192, 1989.

[66] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

[67] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems 27 (NIPS)*, 2014.

[68] P Jameson Graber. Linear quadratic mean field type control and mean field games with common noise, with application to production of an exhaustible resource. *Applied Mathematics & Optimization*, 74(3):459–486, 2016.

[69] Xin Guo, Anran Hu, Renyuan Xu, and Junzi Zhang. Learning mean-field games. In *Advances in Neural Information Processing Systems 32 (NeurIPS)*, 2019.

[70] István Gyöngy. Mimicking the one-dimensional marginal distributions of processes having an itô differential. *Probability theory and related fields*, 71(4):501–516, 1986.

[71] Lajos Gergely Gyurkó, Terry Lyons, Mark Kontkowski, and Jonathan Field. Extracting information from the signature of a financial data stream, 2013.

[72] J. Han, R. Hu, and J. Long. Convergence of deep fictitious play for stochastic differential games. *arXiv preprint arXiv:2008.05519*, 2020.

[73] Jiequn Han and Weinan E. Deep learning approximation for stochastic control problems. *Deep Reinforcement Learning Workshop, NIPS*, 2016.

[74] Jiequn Han and Ruimeng Hu. Deep fictitious play for finding Markovian Nash equilibrium in multi-agent games. In *Mathematical and Scientific Machine Learning (MSML)*, volume 107, pages 221–245. PMLR, 2020.

[75] Jiequn Han, Ruimeng Hu, and Jihao Long. A class of dimensionality-free metrics for the convergence of empirical measures. *arXiv preprint arXiv:2104.12036*, 2021.

[76] Jiequn Han, Arnulf Jentzen, and Weinan E. Solving high-dimensional partial differential equations using deep learning. *Proceedings of the National Academy of Sciences*, 115(34):8505–8510, 2018.

[77] YAN Hanshu, DU Jiawei, TAN Vincent, and FENG Jiashi. On robustness of neural ordinary differential equations. In *International Conference on Learning Representations*, 2019.

[78] Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963.

[79] R. Hu. Deep fictitious play for stochastic differential games. *Communications in Mathematical Sciences*, 19(2):325–353, 2021.

[80] Minyi Huang. Large-population LQG games involving a major player: the Nash certainty equivalence principle. *SIAM Journal on Control and Optimization*, 48(5):3318–3353, 2010.

[81] Minyi Huang, Roland P Malhamé, and Peter E Caines. Large population stochastic dynamic games: closed-loop mckean-vlasov systems and the nash certainty equivalence principle. *Communications in Information and Systems*, 6(3):221–252, 2006.

[82] Emma Hubert and Gabriel Turinici. Nash-MFG equilibrium in a SIR model with time dependent newborn vaccination. *Ricerche di matematica*, 67(1):227–246, 2018.

[83] Tomoyuki Ichiba and Ming Min. Smoothness of directed chain stochastic differential equations. *arXiv preprint arXiv:2202.09354*, 2022.

[84] Nobuyuki Ikeda, Ichiro Shigekawa, and Setsuo Taniguchi. The Malliavin calculus and long time asymptotics of certain Wiener integrals. In *Miniconference on Linear Analysis and Functional Spaces*, pages 46–113. Australian National University, Mathematical Sciences Institute, 1985.

[85] Ioannis Karatzas and Steven Shreve. *Brownian motion and stochastic calculus*, volume 113. Springer Science & Business Media, 2012.

[86] Patrick Kidger, Patric Bonnier, Imanol Perez Arribas, Cristopher Salvi, and Terry Lyons. Deep signature transforms. In *Advances in Neural Information Processing Systems 32*, pages 3105–3115. Curran Associates, Inc., 2019.

[87] Patrick Kidger, James Foster, Xuechen Li, and Terry J Lyons. Neural sdes as infinite-dimensional gans. In *International Conference on Machine Learning*, pages 5453–5463. PMLR, 2021.

[88] Patrick Kidger and Terry Lyons. Signatory: differentiable computations of the signature and logsignature transforms, on both CPU and GPU. *arXiv:2001.00706*, 2020.

[89] Patrick Kidger and Terry Lyons. Signatory: differentiable computations of the signature and logsignature transforms, on both CPU and GPU. *arXiv preprint arXiv:2001.00706*, 2020.

[90] Patrick Kidger, James Morrill, James Foster, and Terry Lyons. Neural controlled differential equations for irregular time series. *Advances in Neural Information Processing Systems*, 33:6696–6707, 2020.

[91] Franz J Király and Harald Oberhauser. Kernels for sequentially ordered data. *Journal of Machine Learning Research*, 20(31):1–45, 2019.

[92] Hiroshi Kunita. Stochastic differential equations and stochastic flows of diffeomorphisms. In *Ecole d'été de probabilités de Saint-Flour XII-1982*, pages 143–303. Springer, 1984.

[93] S. Kusuoka and D. Stroock. The partial Malliavin calculus and its application to non-linear filtering. *Stochastics*, 12(2):83–142, 1984.

[94] S Kusuoka and D Stroock. Applications of the Malliavin calculus, part iii. *J. Fac. Sci. Univ. Tokyo Sect IA Math*, 34:391–442, 1987.

[95] Shigeo Kusuoka. Applications of the Malliavin calculus, part iii. *Journal of the Faculty of Science. University of Tokyo. Section IA. Mathematics*, 32:271–306, 1984.

[96] Shigeo Kusuoka. Malliavin calculus revisited. *Journal of Mathematical Sciences-University of Tokyo*, 10(2):261–278, 2003.

[97] Shigeo Kusuoka and Daniel Stroock. Applications of the Malliavin calculus, part i. In Kiyosi Itô, editor, *Stochastic Analysis*, volume 32 of *North-Holland Mathematical Library*, pages 271–306. Elsevier, 1984.

[98] Shigeo Kusuoka and Daniel W. Stroock. Applications of the Malliavin calculus. ii. *Journal of the Faculty of Science, the University of Tokyo. Sect. 1 A, Mathematics*, 32:1–76, 1985.

[99] Aimé Lachapelle, Jean-Michel Lasry, Charles-Albert Lehalle, and Pierre-Louis Lions. Efficiency of the price formation process in presence of high frequency participants: a mean field game analysis. *Mathematics and Financial Economics*, 10(3):223–262, 2016.

[100] Daniel Lacker, Kavita Ramanan, and Ruoyu Wu. Locally interacting diffusions as markov random fields on path space. *Stochastic Processes and their Applications*, 140:81–114, 2021.

[101] Daniel Lacker and Agathe Soret. A case study on stochastic games on large graphs in mean field and sparse regimes. *arXiv preprint arXiv:2005.14102*, 2020.

[102] Daniel Lacker and Agathe Soret. Many-player games of optimal consumption and investment under relative performance criteria. *Mathematics and Financial Economics*, 14(2):263–281, 2020.

[103] Daniel Lacker and Agathe Soret. A case study on stochastic games on large graphs in mean field and sparse regimes. *Mathematics of Operations Research*, 47(2):1530–1565, 2022.

[104] Daniel Lacker and Kevin Webster. Translation invariant mean field games with common noise. *Electronic Communications in Probability*, 20, 2015.

[105] Daniel Lacker and Thaleia Zariphopoulou. Mean field and n-agent games for optimal investment under relative performance criteria. *Mathematical Finance*, 29(4):1003–1038, 2019.

[106] Jean-Michel Lasry and Pierre-Louis Lions. Mean field games. *Japanese Journal of Mathematics*, 2(1):229–260, 2007.

[107] S.L. Lauritzen. *Graphical Models.* Oxford Statistical Science Series. Clarendon Press, 1996.

[108] Daniel A. Levin, Terry Lyons, and Hao Ni. Learning from the past, predicting the statistics for the future, learning an evolving system. *arXiv: Statistical Finance*, 2013.

[109] Xiaomin Li, Vangelis Metsis, Huangyingrui Wang, and Anne Hee Hiong Ngu. Tts-gan: A transformer-based time-series generative adversarial network. In Martin Michalowski, Syed Sibte Raza Abidi, and Samina Abidi, editors, *Artificial Intelligence in Medicine*, pages 133–143, Cham, 2022. Springer International Publishing.

[110] Xuechen Li, Ting-Kam Leonard Wong, Ricky TQ Chen, and David Duvenaud. Scalable gradients for stochastic differential equations. In *International Conference on Artificial Intelligence and Statistics*, pages 3870–3882. PMLR, 2020.

[111] Shujian Liao, Terry Lyons, Weixin Yang, and Hao Ni. Learning stochastic differential equations using rnn with log signature features, 2019.

[112] Alex Tong Lin, Samy Wu Fung, Wuchen Li, Levon Nurbekyan, and Stanley J Osher. APAC-Net: Alternating the population and agent control via two neural networks to solve high-dimensional stochastic mean field games. *arXiv preprint arXiv:2002.10113*, 2020.

[113] Terry Lyons, Sina Nejad, and Imanol Perez Arribas. Nonparametric pricing and hedging of exotic derivatives, 2019.

[114] Terry Lyons and Zhongmin Qian. *System control and rough paths.* Oxford University Press, 2002.

[115] Terry J Lyons, Michael Caruana, and Thierry Lévy. *Differential equations driven by rough paths*. Springer, 2007.

[116] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.

[117] P Malliavin et al. Sur certaines intégrales stochastiques oscillantes. *C.R. SEANCES ACAD. SCI., 295, 295-300*, 1982.

[118] Stefano Massaroli, Michael Poli, Jinkyoo Park, Atsushi Yamashita, and Hajime Asama. Dissecting neural odes. *Advances in Neural Information Processing Systems*, 33:3952–3963, 2020.

[119] Ming Min and Ruimeng Hu. Signatured deep fictitious play for mean field games with common noise. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 7736–7747. PMLR, 18–24 Jul 2021.

[120] Ming Min, Ruimeng Hu, and Tomoyuki Ichiba. Directed chain generative adversarial networks. In *Proceedings of the 40th International Conference on Machine Learning*, 2023.

[121] Ming Min and Tomoyuki Ichiba. Convolutional signature for sequential data. *Digital Finance*, pages 1–26, 2022.

[122] James Morrill, Adeline Fermanian, Patrick Kidger, and Terry Lyons. A generalised signature method for time series, 2020.

[123] James Morrill, Cristopher Salvi, Patrick Kidger, and James Foster. Neural rough differential equations for long time series. In *International Conference on Machine Learning*, pages 7829–7838. PMLR, 2021.

[124] Sebastien Motsch and Eitan Tadmor. Heterophilious dynamics enhances consensus. *SIAM review*, 56(4):577–621, 2014.

[125] Hao Ni, Lukasz Szpruch, Marc Sabate-Vidales, Baoren Xiao, Magnus Wiese, and Shujian Liao. Sig-wasserstein gans for time series generation. In *Proceedings of the Second ACM International Conference on AI in Finance*, pages 1–8, 2021.

[126] Hao Ni, Lukasz Szpruch, Magnus Wiese, Shujian Liao, and Baoren Xiao. Conditional sig-wasserstein gans for time series generation. *arXiv preprint arXiv:2006.05421*, 2020.

[127] Mojtaba Nourian and Peter E Caines. ϵ-Nash mean field game theory for nonlinear stochastic dynamical systems with major and minor agents. *SIAM Journal on Control and Optimization*, 51(4):3302–3331, 2013.

[128] David Nualart. *Malliavin calculus and its applications.* Number 110 in CBMS Regional Conference Series in Mathematics. American Mathematical Soc., 2009.

[129] David Nualart and Moshe Zakai. The partial Malliavin calculus. In *Séminaire de Probabilités XXIII*, pages 362–381. Springer, 1989.

[130] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[131] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

[132] Sarah Perrin, Julien Pérolat, Mathieu Laurière, Matthieu Geist, Romuald Elie, and Olivier Pietquin. Fictitious play for mean field games: Continuous time analysis and applications. In *Advances in Neural Information Processing Systems 33 (NeurIPS)*, 2020.

[133] Alessio Quaglino, Marco Gallieri, Jonathan Masci, and Jan Koutník. Snode: Spectral discretization of neural odes for system identification. In *International Conference on Learning Representations*, 2019.

[134] Christoph Reisinger and Wolfgang Stockinger. An adaptive euler–maruyama scheme for mckean–vlasov sdes with super-linear growth and application to the mean-field fitzhugh–nagumo model. *Journal of Computational and Applied Mathematics*, 400:113725, 2022.

[135] Alejandro Pasos Ruiz, Michael Flynn, James Large, Matthew Middlehurst, and Anthony Bagnall. The great multivariate time series classification bake off: a review and experimental evaluation of recent algorithmic advances. *Data Mining and Knowledge Discovery*, 35(2):401–449, 2021.

[136] Lars Ruthotto, Stanley J Osher, Wuchen Li, Levon Nurbekyan, and Samy Wu Fung. A machine learning framework for solving high-dimensional mean field game and mean field control problems. *Proceedings of the National Academy of Sciences*, 117(17):9183–9193, 2020.

[137] Cristopher Salvi, Maud Lemercier, Chong Liu, Blanka Horvath, Theodoros Damoulas, and Terry Lyons. Higher order kernel mean embeddings to capture

filtrations of stochastic processes. *Advances in Neural Information Processing Systems*, 34:16635–16647, 2021.

[138] Sudheer Kumar Sharma, Sanjeev Kumar, et al. Suppression of multimodality in inter-spike interval distribution: Role of external damped oscillatory input. *IEEE Transactions on NanoBioscience*, 17(3):329–341, 2018.

[139] Lachlan D Smith and Georg A Gottwald. Chaos in networks of coupled oscillators with multimodal natural frequency distributions. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 29(9):093127, 2019.

[140] Setsuo Taniguchi. Applications of Malliavin's calculus to time-dependent systems of heat equations. *Osaka Journal of Mathematics*, 22(2):307 – 320, 1985.

[141] Cristopher Salvi Thomas Cass, Terry Lyons and Weixin Yang. Computing the untruncated signature kernel as the solution of a goursat problem, 2020.

[142] Nilay Tiwari, Arnob Ghosh, and Vaneet Aggarwal. Reinforcement learning for mean field game. *arXiv preprint arXiv:1905.13357*, 2019.

[143] Csaba Toth and Harald Oberhauser. Bayesian learning from sequential data using gaussian processes with signature covariances. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, volume 119, pages 9548–9560. PMLR, 2020.

[144] Alan Tsang and Kate Larson. Opinion dynamics of skeptical agents. In *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems*, pages 277–284, 2014.

[145] Belinda Tzen and Maxim Raginsky. Neural stochastic differential equations: Deep latent gaussian models in the diffusion limit. *arXiv preprint arXiv:1905.09883*, 2019.

[146] Belinda Tzen and Maxim Raginsky. Theoretical guarantees for sampling and inference in generative models with latent diffusions. In *Conference on Learning Theory*, pages 3084–3114. PMLR, 2019.

[147] Ramon van Handel. *Probability in High Dimension.* APC 550 Lecture Notes. Princeton University, 2016.

[148] Yao Xuan, Robert Balkin, Jiequn Han, Ruimeng Hu, and Hector D. Ceniceros. Optimal policies for a pandemic: A stochastic game approach and a deep learning algorithm. In *Mathematical and Scientific Machine Learning (MSML)*, 2021. Accepted. arXiv:2012.06745.

[149] Jinsung Yoon, Daniel Jarrett, and Mihaela Van der Schaar. Time-series generative adversarial networks. *Advances in neural information processing systems*, 32, 2019.

[150] Tianjun Zhang, Zhewei Yao, Amir Gholami, Joseph E Gonzalez, Kurt Keutzer, Michael W Mahoney, and George Biros. Anodev2: A coupled neural ode framework. *Advances in Neural Information Processing Systems*, 32, 2019.