COOPREDICT: COOPERATIVE DIFFERENTIAL GAMES FOR TIME SERIES PREDICTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Modeling time series dynamics with neural differential equations has become a major line of research that opened new ways to handle various real-world scenarios (*e.g.*, missing observations, irregular times). Despite the progress, most existing methods still face challenges in providing an explainable rationale on temporal association, which tells how past observations affect future states. To tackle this challenge, we introduce novel multi-agent based neural stochastic differential equations and analyze the time series prediction through the lens of *cooperative differential games*. Our framework provides an explainable method that can reveal the underlying temporal relevance of the data and fully utilizes this information to systemically solve the prediction problem. We develop the gradient descent based *deep neural fictitious play* to approximate the Nash equilibrium and theoretical results assure the convergence. Throughout the experiments on various datasets, we demonstrate the superiority of our framework over all the benchmarks in modeling time series prediction by capitalizing on the underlying temporal dynamics without any inductive bias. An ablation study shows that neural agents of the proposed framework learn intrinsic temporal relevance to predict accurate time series.

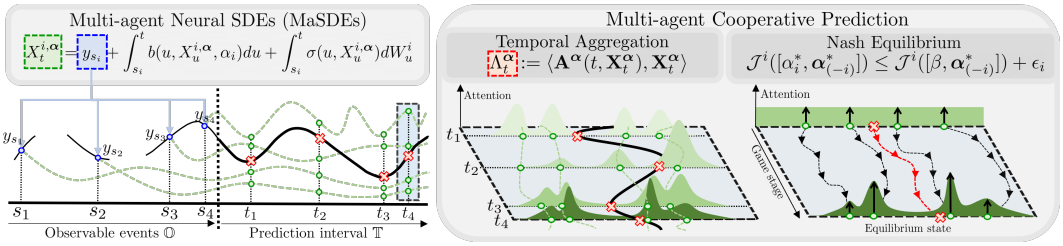
1 INTRODUCTION

The key challenge of time series prediction is understanding how past observations contribute to the future and building a probabilistic model that intrinsically captures the temporal correlation. A model is required to differentiate the relative importance of past observations on the future time series such that the redundant information is well-suppressed during inference. To address this challenge, a series of previous works followed the philosophy of defining information redundancy by adopting a concept known as *temporal decay* (Che et al., 2018; Mei & Eisner, 2017). Such an inductive bias relies on the belief that the influence of past observations exponentially decreases over time. However, due to the strong assumption, any method built upon the fixed inductive bias may fail to properly capture different temporal dynamics in various real-world scenarios.

A promising direction to tackle this challenge is to *learn* the temporal correlation structure from data. One candidate for learning the implicit relation is recurrent neural networks (RNNs) which subsequently encode the past observations into a latent space with a gating mechanism. Due to the implicitness of latent encoding, the model provides ambiguous explanations on how observations are related to the prediction. Furthermore, conventional RNNs are incapable of capturing the irregularly sampled time series, which is common in many real-world applications.

Recently, remarkable advances have been made to model underlying continuous temporal dynamics utilizing *neural differential equations* (Chen et al., 2018; Li et al., 2020; Kidger et al., 2020) (NDEs). For instance, a stream of research (Rubanova et al., 2019; Deng et al., 2021; Schirmer et al., 2022) has focused on overcoming the initial condition problem of conventional differential equation models by encoding temporal dynamics into the latent space. A set of past observations are incorporated into latent representations and fully utilized for time series prediction. However, owing to the inexplicable relation between temporal states in the latent representations, these methods are still incapable of showing the explicit rationale that can reveal the impact of observations on the future.

In this paper, we propose a novel framework, which we call ***CooPredict***, built upon a *game-theoretic* formalism to model temporal dynamics of time series data. More specifically, we extend the conventional differential equation (DE) to the *multi-agent* counterpart for decomposing the observational



(a) Multi-agent Neural SDEs (b) Decision Attention (c) Differential game

Figure 1: Conceptual illustration of the proposed framework: (a) The proposed MaSDE encodes separate initial conditions and propagate individual decisions over the interval $t \in [s_i, T]$. The bold black line indicates the time series on prediction interval $\mathbb{T}$ and dashed green lines are the *decisions* of neural agents. (b) The separate decision is integrated by the temporal aggregation. The red crosses show the predictor $\Lambda_t^{\boldsymbol{\alpha}}$ representing the cooperative prediction. (c) The optimal action profiles $\boldsymbol{\alpha}^*$ are obtained in the Nash equilibrium.

time series. Each agent individually encodes the impact of a partial observation into an underlying stochastic trajectory and interacts with each other to extract meaningful information to predict the future. We formulate the collaboration among agents from the view of *cooperative differential games* (Leitmann, 1974; Staatz, 1983; Sexton, 1986) to integrate the individual information and adaptively balance the importance of each agent. As a result of the differential game, cooperative agents achieve the *Nash equilibrium*, and agree to suppress redundant observations and to highlight the contribution of important observations for accurate prediction. To solve the differential game, we propose a novel deep learning-based algorithm to conduct the *fictitious play*. During the process, our method applies the gradient descent scheme to solve decoupled optimal control problems in a tractable and parallel way. Furthermore, we provide theoretical foundations upon the Feynman-Kac formalism for solving optimal control problems and theoretical analysis to guarantee the convergence of the proposed algorithm.

Unlike the existing methods, our framework explicitly provides a clear explanation of how the past is related to the future in time series. We validate our method through experiments on multiple synthetic and real-world datasets that cover various types of temporal dynamics. Our method outperforms state-of-the-art DE-based methods by capturing the underlying temporal dynamics without any inductive bias via a novel game-theoretic framework.

2 Multi-agent Cooperative Prediction

Problem Setup. Consider a general time series forecasting set-up where each instance of a time series is defined on the entire time interval $[0, T]$ comprising of observable past (*i.e.*, $\mathbb{O}$) and target future intervals (*i.e.*, $\mathbb{T}$). We assume that the irregularly sampled observations are collected at time stamps $\mathcal{O} = \{s_1, \ldots, s_I\}$ where $s_i \in \mathbb{O}$ for $i \in \{1, \ldots, I\}$. Then, we focus on building a probabilistic model that can predict future values at any time $t \in \mathbb{T}$ based on the set of past observations in $\mathbb{O}$.

2.1 Multi-agent Neural SDEs

Suppose we have a set of I number of past observations for a given time series, *i.e.*, $\{y_{s_i}\}_{s_i \in \mathcal{O}}$. The primary object of our interest is the set of stochastic processes $\mathbf{X}_t^{\boldsymbol{\alpha}} = [X_t^{1,\boldsymbol{\alpha}}, \ldots, X_t^{I,\boldsymbol{\alpha}}]$ parameterized by multiple control agents $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_I]$, which is defined as a collection of solutions to the following multi-agent stochastic differential equations (MaSDEs) (Øksendal & Sulem, 2007)[1]:

MaSDEs

$$(\mathbf{X}_t^{\boldsymbol{\alpha}},\ \underbrace{\{y_{s_i}\}_{s_i \in \mathcal{O}}}_{\text{past observations}}) :\ \begin{cases} dX_t^{i,\boldsymbol{\alpha}} = b(t, X_t^{i,\boldsymbol{\alpha}}, \alpha_i)dt + \sigma(t, X_t^{i,\boldsymbol{\alpha}})dW_t^i, \\ X_t^{i,\boldsymbol{\alpha}} = y_{s_i} + \int_{s_i}^t b(u, X_u^{i,\boldsymbol{\alpha}}, \alpha_i)du + \int_{s_i}^t \sigma(u, X_u^{i,\boldsymbol{\alpha}})dW_u^i. \end{cases} \tag{1}$$

Here, $\{W_t^i\}$ are independent d-dimensional Wiener processes, and b and σ are drift and diffusion functions, respectively. The control agent $\alpha_i : [0, T] \times \mathbb{R}^d \times \Theta \to \mathbb{R}^d$, which we refer to as a *neural agent*, is modeled as a neural network $\alpha_i := \alpha_i(t, X_t^{i,\boldsymbol{\alpha}}; \theta_i)$ parameterized by θ_i. Each neural agent takes input a spatio-temporal tuple $(t, X_t^{i,\boldsymbol{\alpha}})$ and produces infinitesimally successive outputs

[1] From this point forward, we use boldface letters to denote a collection of objects obtained from multi-agents, *e.g.*, $\mathbf{X}_t^{\boldsymbol{\alpha}} = [X_t^{1,\boldsymbol{\alpha}}, \ldots, X_t^{I,\boldsymbol{\alpha}}]$, and we omit the dependence on temporal state when clear in the context.

$(t+dt, X^{i,\boldsymbol{\alpha}}_{t+dt})$. We define a neural agent's *decision* as a continuous stochastic trajectory over the future interval $\mathbb{T}$, *i.e.*, $\{X^{i,\boldsymbol{\alpha}}_t\}_{t\in\mathbb{T}}$. By setting the past observation y_{s_i} as the *initial condition*, the decision of a neural agent can be considered as an information propagator that can explicitly represent the impact of the corresponding past observation via the conditional expression $p(X_{t\in\mathbb{T}}|y_{s\in\mathbb{O}})$; see Figure 1-(a) for a pictorial illustration.

Temporal Aggregation. Since each neural agent can only represent the individual impact of partial information, we aggregate the individual decisions made by each neural agent to capitalize on the temporal dynamics available from the entire set of past observations. More specifically, we introduce the *predictor* $\Lambda^{\boldsymbol{\alpha}}_t$ that produces the future value at time t as a weighted sum of the individual decisions as the following:

$$\Lambda^{\boldsymbol{\alpha}}_t \coloneqq \mathbf{A}^{\boldsymbol{\alpha}}(t, \mathbf{X}^{\boldsymbol{\alpha}}_t)^\top \mathbf{X}^{\boldsymbol{\alpha}}_t = A^{\alpha_i}_i(t, X^{i,\boldsymbol{\alpha}}_t) X^{i,\boldsymbol{\alpha}}_t + \sum_{i\neq j} A^{\alpha_j}_j(t, X^{j,\boldsymbol{\alpha}}_t) X^{j,\boldsymbol{\alpha}}_t. \tag{2}$$

Here, we introduce the *decision attention* $\mathbf{A}^{\boldsymbol{\alpha}}$ which provides the relative importance of each decision, *i.e.*, $\mathbf{A}^{\boldsymbol{\alpha}} : \mathbb{T}\times\mathbb{R}^d\times\Theta \to \Delta^I$ where $\Delta^I \coloneqq \{a\in[0,1]^I | \mathbf{1}^\top a = 1\}$ is a I-simplex and $\mathbf{A}^{\boldsymbol{\alpha}} = [A^{\alpha_1}_1, \dots, A^{\alpha_I}_I]$. Given the decision attention, the predictor $\Lambda^{\boldsymbol{\alpha}}_t$ produces a temporally aggregated decision from the decisions made by the individual agents.

The purpose of decision attention is to differentiate the associated impact of stochastic trajectories on predicting the future values in a data-driven fashion. It is worth highlighting that the proposed predictor becomes equivalent to the temporal average suggested by (Park et al., 2022) when the decision attention assigns uniform weights irrespective of spatio-temporal variable, *i.e.*, $A^{\alpha_i}_i = 1/I$.

Stochastic Optimal Control. To train neural agents in MaSDEs, we adopt the *stochastic optimal control* (Yeung & Petrosjan, 2006; Carmona & Delarue, 2018) as our central methodology to formally define an objective functional $\mathcal{J}^i$. Specifically, each neural agent is given a behavioral rule as follows:

Objective Functional

$$\mathcal{J}^i(t, x, [\alpha_i, \boldsymbol{\alpha}_{(-i)}]) = \mathbb{E}\left[\int_{\mathbb{T}} h^i\Big(s, X^{i,\boldsymbol{\alpha}}_s, \mathbf{X}^{(-i),\boldsymbol{\alpha}}_s, [\alpha_i, \boldsymbol{\alpha}_{(-i)}]\Big) ds + \Psi^i(X^{i,\boldsymbol{\alpha}}_T) \,\Big|\, \mathbf{X}_t = x\right], \tag{3}$$

$$\text{where } h^i\left(t, X^{i,\boldsymbol{\alpha}}_t, \mathbf{X}^{(-i),\boldsymbol{\alpha}}_t, [\alpha_i, \boldsymbol{\alpha}_{(-i)}]\right) = \|\Lambda^{\boldsymbol{\alpha}}_t - y_t\|^2 \quad \text{and} \quad \Psi^i(X^{i,\boldsymbol{\alpha}}_T) = \frac{1}{2}||X^{i,\boldsymbol{\alpha}}_T||^2.$$

Here, $\mathbf{X}^{(-i),\boldsymbol{\alpha}}_s \in \mathbb{R}^{d(I-1)}$ and $\boldsymbol{\alpha}_{(-i)} \in \mathbb{A}$ are the sets of other agents' decisions and actions, respectively. The *cost function* h^i is designed to minimize the distance between the prediction (*i.e.*, $\Lambda^{\boldsymbol{\alpha}}_t$) and the target time series (*i.e.*, y_t). This encourages each neural agent to make contributions in the temporal aggregation under the goal of accurate time series prediction. The *terminal cost function* Ψ^i regularizes decisions at the terminal state and plays a central role in the Feynman-Kac formalism to solve the cooperative game. We will further elaborate the details in Section 3.

Given the *environment* (*i.e.*, the entire interacting objects), each neural agent changes its action to find a *value function* $\mathcal{V}^i$ that defines the optimal state of actions:

$$\mathcal{V}^i(t,x) = \min_{\alpha_i\in\mathbb{A}^i} \mathcal{J}^i(t, x, [\alpha_i, \boldsymbol{\alpha}_{(-i)}]) = \min_{\theta_i\in\Theta} \mathcal{J}^i(t, x, [\alpha_i(\cdot,\cdot;\theta_i), \boldsymbol{\alpha}_{(-i)}(\cdot,\cdot;\boldsymbol{\theta}_{(-i)})]), \tag{4}$$

where the optimal control problem in (4) takes arbitrary actions $\boldsymbol{\alpha}_{(-i)}$ made by other agents. To solve (4), we propose a gradient descent-based method motivated from the fictitious play, which we will further elaborate in Section 3.

2.2 Cooperative Differential Games

If the past observation y_s is highly associated with the future event y_t (*i.e.*, $p(y_{t\in\mathbb{T}}|y_{s\in\mathbb{O}}) \uparrow$), it is reasonable to pay high attention (*i.e.*, $\boldsymbol{A}^{\boldsymbol{\alpha}}(t,\cdot)\uparrow$) to capitalize on the observation y_s. Unfortunately, determining the optimal attention is a daunting task as the temporal correlation is intrinsically data-dependent and not given a priori in general. Hence, the method requires a unified structure that (i) can systemically extract information from data to adjust the decision attention $\mathbf{A}^{\boldsymbol{\alpha}}$ based on the relative importance of past observations and (ii) can provide a framework on the coalition of the interacting agents for the proper coordination.

To address the above requirements, we propose a novel framework, which we call CooPredict, to formalize the time series prediction problem as a *cooperative differential games* between multiple agents. Cooperative differential games have been extensively studied in many disciplines (Perelman et al., 2011; Jørgensen et al., 2010; Yeung & Petrosyan, 2012) to investigate the group rationality of agents sharing the same goal. Inspired by these works, we design a cooperative differential game that provides *cooperative prediction* by achieving optimally balanced decision attention among neural agents whose shared goal is an accurate prediction. The critical point is that the neural agent supporting a non-informative observation voluntarily sacrifices its own cost ($i.e., \mathcal{V}^i \uparrow$) by decreasing the influence on temporal aggregation so that the impact of other informative observations ($i.e., \mathcal{V}^{i \neq j} \downarrow$) can be further emphasized. Such a cooperative behavior is mutually advantageous to the interacting agents as it ultimately achieves the performance improvement. Figure 1-(c) shows the conceptual illustration of the proposed cooperative game where the homogeneous decision attentions are gradually transformed into the equilibrium state achieved by enthusiastic agents following the group rationality.

2.3 Cooperative Prediction via Nash Equilibrium

For a mathematical derivation of the optimal decision states, we introduce an important type of equilibrium states called *Nash Equilibrium* as follows:

Definition 1. ***(Markovian ϵ-Nash Equilibrium)*** *Let us consider a set of closed-feedback type Markovian controller parameterized by neural networks $\boldsymbol{\alpha}^* := \{\alpha_i^*\}_{i\in\{1,\dots,I\}}$ that induce value functions. For the arbitrary actions $\boldsymbol{\beta}$ taken in action set $\mathbb{A}$, we say that neural agents are in the ϵ-Nash Equilibrium if the following inequality is satisfied:*

$$\mathcal{J}^i(t, x, [\alpha_i^*, \boldsymbol{\alpha}_{(-i)}^*]) \leq \mathcal{J}^i(t, x, [\beta_i, \boldsymbol{\alpha}_{(-i)}^*]) + \epsilon_i, \quad \forall i \in \{1, \dots, I\}, \quad \epsilon_i > 0, \tag{5}$$

In the state of equilibrium, any non-optimal action $\boldsymbol{\beta}$ produces worse cost (with a margin $\boldsymbol{\epsilon} := \{\epsilon_i\}_{i\in I}$) than taking the optimal actions $\boldsymbol{\alpha}^*$. Here, the marginal constant $\boldsymbol{\epsilon}$ represents a degree of capacity for neural agents to achieve the perfect equilibrium. The inequality in (5) describes an optimal state of interacting agents having no incentive to change their action. While each neural agent mutually finds the best response to the environment made by the optimal colleagues, the entire cost is comprehensively minimized and a more accurate prediction can be obtained. Further, the following relation implies that the equilibrium state achieved in (5) is indeed a result of cooperation:

$$\mathcal{J}_{co}(t, x, \boldsymbol{\alpha}^*) = \sum_i^I \mathcal{J}^i(t, x, [\alpha_i^*, \boldsymbol{\alpha}_{(-i)}^*]) \leq \sum_i^I \mathcal{J}^i(t, x, [\alpha_i^*, \boldsymbol{\beta}_{(-i)}]) + I \sup_i \epsilon_i, \tag{6}$$

where $\mathcal{J}_{co}$ is the objective functional that describes the coalition cost (Yeung & Petrosjan, 2006) of the cooperative neural agents. The inequality in (6) can be easily shown by the algebraic property of cost functions as the following:

$$\mathbb{E}\left[h^i\left(t, X_t^{i,\boldsymbol{\alpha}}, \mathbf{X}_t^{(-i),\boldsymbol{\alpha}}, [\alpha_i^*, \boldsymbol{\alpha}_{(-i)}^*]\right)\right] = \mathbb{E}\left[h^j\left(t, X_t^{j,\boldsymbol{\alpha}}, \mathbf{X}_t^{(-j),\boldsymbol{\alpha}}, [\alpha_j^*, \boldsymbol{\alpha}_{(-j)}^*]\right)\right], \quad i \neq j. \tag{7}$$

Fictitious Play. In order to achieve the Nash equilibrium for multi-agents, the previous interest on considering separate and individual objectives in (4) shifts to an interacting scenario. From this new perspective, we transform the original value function that only considers insensitive feedback into a new one that can consider optimal colleagues' feedback driven by their contribution to the prediction:

$$\mathcal{V}^i(t, x) = \min_{\alpha_i^* \in \mathbb{A}^i} \mathcal{J}^i(t, x, [\alpha_i^*, \boldsymbol{\alpha}_{(-i)}^*]), \quad s_i \in \mathcal{O}. \tag{8}$$

Since finding the Nash equilibrium is PPAD hard (Daskalakis et al., 2009; Goldberg, 2011), it is intractable to directly compute the equilibrium owing to a large number of agents considered in time series prediction problems of our experiments ($i.e., I \approx 50$). Thus, for a tractable computation, we instead rely on an alternative dynamical formulation called *fictitious play* (Cardaliaguet & Hadikhanloo, 2017). The fictitious play is an iterative procedure that decouples interacting agents and separately solves the stochastic optimal control. The central idea is first to share public information about the entire system and then to decouple value functions by solving individual Hamiltonian-Jacobi-Bellman equations (HJBEs) in Section 3.

3 DEEP NEURAL FICTITIOUS PLAY

In this section, we suggest the novel deep learning framework for the fictitious play. To this goal, we start by introducing the set of adjoint equations called the *forward-backward stochastic differential equations* (FBSDEs) (Carmona, 2016) as a probabilistic framework to tackle optimal control problems of obtaining value functions.

Theorem 1. *(Non-linear Feynman Kac theorem Pham (2015)). Let us consider closed-feedback type Markovian controls $\boldsymbol{\alpha}_t$. We define the triplet $(\mathbf{X}_t^{\boldsymbol{\alpha}}, \mathbf{Y}_t^{\boldsymbol{\alpha}}, \mathbf{Z}_t^{\boldsymbol{\alpha}})$ that constitutes the system of SDEs:*

FBSDEs

$$(\mathbf{X}_t^{\boldsymbol{\alpha}}, \mathbf{Y}_t^{\boldsymbol{\alpha}}, \mathbf{Z}_t^{\boldsymbol{\alpha}}): \quad \begin{cases} dX_t^{i,\boldsymbol{\alpha}} = \sigma(t, X_t^{i,\boldsymbol{\alpha}})dB_t^i \\ dY_t^{i,\boldsymbol{\alpha}} = -H^i(t, \mathbf{X}_t^{\boldsymbol{\alpha}}, F_t^{i,\boldsymbol{\alpha}}, [\alpha_i, \boldsymbol{\alpha}_{(-i)}])dt + Z_t^{i,\boldsymbol{\alpha}} \cdot dB_t^i \\ Y_T^{i,\boldsymbol{\alpha}} = \Psi^i(X_T^{i,\boldsymbol{\alpha}}), \quad i \in \{1, \dots, I\} \end{cases}, \tag{9}$$

where H^i is the stochastic Hamiltonian system. Then, auxiliary adjoint variables $Y_t^{i,\boldsymbol{\alpha}}$ and $Z_t^{i,\boldsymbol{\alpha}}$ can be reformulated as follows:

$$\begin{cases} Y_t^{i,\boldsymbol{\alpha}} = \mathcal{J}^i(t, \mathbf{X}_t^{\boldsymbol{\alpha}}, [\alpha_i, \boldsymbol{\alpha}_{(-i)}]), \quad Z_t^{i,\boldsymbol{\alpha}} = \nabla_x \mathcal{J}^i(t, \mathbf{X}_t^{\boldsymbol{\alpha}}, [\alpha_i, \boldsymbol{\alpha}_{(-i)}]), \\ H^i(t, \mathbf{X}_t^{\boldsymbol{\alpha}}, F_t^{i,\boldsymbol{\alpha}}, [\alpha_i, \boldsymbol{\alpha}_{(-i)}]) = \left[\sigma(t, X_t^{i,\boldsymbol{\alpha}})^{-T} Z_t^{i,\boldsymbol{\alpha}}\right] \cdot b(t, X_t^{i,\boldsymbol{\alpha}}, \alpha_i) + h^i(t, \mathbf{X}_t^{\boldsymbol{\alpha}}, \boldsymbol{\alpha}), \end{cases} \tag{10}$$

where we denote $F_t^{i,\boldsymbol{\alpha}} \coloneqq F^{i,\boldsymbol{\alpha}}(t, X_t^{i,\boldsymbol{\alpha}}, \alpha_i) = \sigma(t, X_t^{i,\boldsymbol{\alpha}})^{-T} b(t, X_t^{i,\boldsymbol{\alpha}}, \alpha_i)$ for simplicity.

To solve the optimal control problem, the existing method (Han & Hu, 2020) considers quadratic linear forms in the forward dynamics and its corresponding global convex Hamiltonian system (*i.e.*, $\nabla_{\alpha^i} H^i = 0$). Despite the numerical advantage in obtaining an optimal control, the restricted form of linear SDEs limits the usage of their method on high-level applications such as time series prediction. This shortcoming motivates us to utilize neural networks when modeling control agents for enough expressivity. A critical issue, here, is to approximate the solution to non-convex type HJBEs that support the neural networks structure. To tackle this issue, we reformulate the equivalent adjoint problem to find value functions:

$$\theta_i^* = \arg\min_{\theta_i} \int_{\mathbb{T}} dY_t^{i,\boldsymbol{\alpha}}([\theta_i, \boldsymbol{\theta}_{(-i)}])dt, \quad \alpha_i^* \coloneqq \alpha_i^*(\cdot, \cdot, \theta_i^*). \tag{11}$$

Instead of searching the analytic solution to HJBEs (*i.e.*, $\mathbb{A}^i \approx \mathbf{ker}\nabla_{\alpha^i} H^i$), the newly suggested adjoint equation in (11) enables us to apply the gradient descent scheme to find optimal actions. In this case, our method searches the parameter space to obtain the optimal admissible action (*i.e.*, $\mathbb{A}^i \approx \alpha^i(\cdot, \cdot; \mathbf{ker}\nabla_\theta H^i[\alpha^i(\theta^i)])$) meaning that we focus on the local Nash equilibrium.

Deep Neural Fictitious Play. Let us define an entire set of action profiles as $\boldsymbol{\alpha}^m$ at stage $m \in \{1, \dots, M\}$. Then, our method iteratively conducts the following two-step optimization:

- **Information Distribution.** At the beginning of the m-th stage, each neural agent publicly share the information about the entire system $(\boldsymbol{\alpha}^{m-1}, \mathbf{X}_t^{\boldsymbol{\alpha}, m-1})$ from the $(m-1)$-th stage. In this case, we can define the decoupled value function for individual neural agents as follows:

$$\mathcal{V}^{i,m}(t, x) = \inf_{\alpha^i \in \mathbb{A}^i} \mathbb{E}\left[\mathcal{J}^{i,m}(t, \mathbf{X}_t^{\boldsymbol{\alpha}^{m-1}}, [\alpha_i^{m-1}, \bar{\boldsymbol{\alpha}}_{(-i)}^{m-1}]) \middle| \mathbf{X}_t^{\bar{\boldsymbol{\alpha}}^{m-1}}\right], \tag{12}$$

  where $\mathbf{X}_t^{\bar{\boldsymbol{\alpha}}^{m-1}}$ indicates the set of decisions driven by action profiles $\bar{\boldsymbol{\alpha}}^{m-1}$ of the $(m-1)$-th stage. In this step, each neural agent recognizes response of colleagues towards the environment and makes its relative decision. Note that each agent solves a minimization problem according to the fixed environment of the previous stage.

- **Decoupled Gradient Descent.** After defining the decoupled problems in (12), each neural agent solves its individual adjoint problem in (11) and updates its parameters as

$$\theta_i^{m+1} = \mathbb{E}\left[\theta_i^m - \gamma_m \int_{\mathbb{T}} \nabla_{\theta_i} dY_t^{i,\boldsymbol{\alpha}}([\theta_i^m, \bar{\boldsymbol{\theta}}_{(-i)}^m])dt\right], \quad \bar{\boldsymbol{\alpha}}_{(-i)}^m \coloneqq \alpha_i(\cdot, \cdot, \bar{\boldsymbol{\theta}}_{(-i)}^m), \tag{13}$$

  where γ_m is a learning rate of the optimizer at the m-th stage. In (13), gradient descent is applied to minimize the separate cost $\mathcal{J}^{i,m}$ while fixing the parameters of colleagues $\bar{\boldsymbol{\theta}}_{(-i)}^m$. After solving (13) for I neural agents in a parallel way, we collect the entire action profiles and update the public information for the next stage, *i.e.*, $\boldsymbol{\alpha}_i^{m+1} \leftarrow \boldsymbol{\alpha}_i^m(\cdot, \cdot, \theta + d\theta)$.

Convergence of the Fictitious Play. In the previous contents, we suggest an alternative way to approximate the Nash equilibrium via the gradient descent scheme. In what follows, we will show the existence of an action set $\mathbb{A}$ related to the marginal constant $\boldsymbol{\epsilon} := \{\epsilon_i\}_{i\in\{1,\dots,I\}}$ in the inequality of (5).

Proposition 1. ***(informal)*** *There exists a action set $\boldsymbol{\alpha}(\boldsymbol{\epsilon}) \in \mathbb{A}$ where the Markovian ϵ-Nash equilibrium in (5) can be achieved. Moreover, the obtained optimal actions $\alpha_i^{m+1}{}_{i\in\{1,\cdots,I\}} \in \boldsymbol{\alpha}(\boldsymbol{\epsilon})$ from deep neural fictitious play preserves the stochastic optimality and the solution to HJBEs[2]:*

$$\boldsymbol{\mathcal{V}}_t + \boldsymbol{H}(t, \cdot, F_t, (\boldsymbol{\alpha}^{m+1}, \boldsymbol{\alpha}^m_{(\cdot)})) + \frac{1}{2}\mathbf{Tr}(\Sigma \boldsymbol{Hess}\boldsymbol{\mathcal{V}}_t) = 0, \tag{14}$$

where $\boldsymbol{H}$ is the stochastic Hamiltonian system, $\boldsymbol{F}_t$ is an adjoint variable related to σ and b, and $\boldsymbol{Hess}\boldsymbol{\mathcal{V}}_t$ denotes the Hessian of function $\boldsymbol{\mathcal{V}}_t$.

Proposition 1 reveals that the proposed fictitious play preserves the stochastic optimality during the two-step optimization by showing the existence of local solutions to the decoupled Hamiltonian systems. The crucial point is that the local action set is related to the capacity of neural agents and neural networks with a large capacity assures small marginal values $\boldsymbol{\epsilon}$ for accurate prediction.

Corollary 1. ***(Convergence of Predictor)*** *There exist a constant C that is proportional to I, $\mathbb{T}$ and Lipschitz constants L_b, L_α for $\{b_i, \alpha_i\}_{i\in\{1,\dots,I\}}$ such that the following relation holds:*

$$\mathbb{E}\left[\int_{\mathbb{T}} \left\| \boldsymbol{\Lambda}_s^{\boldsymbol{\alpha}^m} - \boldsymbol{\Lambda}_s^{\boldsymbol{\alpha}^{m^*}} \right\|^2 ds\right] \propto O(||\boldsymbol{\theta}^m - \boldsymbol{\theta}^{m^*}||^3 C\left[I, \mathbb{T}, L_b, L_\alpha\right] \overline{\Sigma}), \tag{15}$$

where $\overline{\Sigma} := \sup_{t\in\mathbb{T}} \mathbb{E}\left[\left\|\boldsymbol{\sigma}^T \boldsymbol{\sigma}(t, \cdot)\right\|_F\right]$ is the maximal bound with respect to quadratic variations of neural agents.

Corollary 1 shows that the convergence of the temporal aggregation made by parameterized neural agents is highly dependent on the dynamics of the gradient descent during the fictitious play. If one reformulates the proposed cooperative game into the zero-sum type counterpart, the opposed direction of gradients induced by conflict agents can cause the divergent learning dynamics and unpleasant results. In other words, the proposed gradient descent based fictitious play fits to the cooperative scenario where neural agents pursue a shared goal. Appendix A.3 provides the numerical analysis on the behavior of neural agents under the cooperation/competition to elucidate how the proposed fictitious play operates in different game scenarios.

4 Related Work

Neural DEs for Time Series. In the pioneering works (Dupont et al., 2019; Rubanova et al., 2019), the general framework of encoding the complex time series into the latent space was first introduced to improve the representational power of conventional Neural ODE. Further, latent SDEs (Li et al., 2020) was proposed to enrich conventional deterministic models by considering a stochastic component (*e.g.*, Wiener process). Neural RDE (Morrill et al., 2021) exploited the representation of log-signatures of successive time series. CLPF (Deng et al., 2021) combined two distinctive ideas, continuous SDE and normalizing flow, to model continuous latent flows. CRU (Schirmer et al., 2022) extended Kalman filters to model-based continuous-discrete filters and showed the relation to neural SDEs. Neural Laplace (Holt et al., 2022) showed novel interpretation on time series modeling by introducing the Laplace representation that generalizes conventional DEs in the frequency domain. CSDE-TP (Park et al., 2022) suggested a different perspective by adopting a control-theoretic interpretation of time series prediction tasks to obtain the optimal paths driven by neural agents. The notable distinction is that our proposed framework generalizes the equiconcentration of temporal attention in CSDE-TP to enhance the aggregated information by balancing the relative importance.

Inductive Bias on Temporal States. In probabilistic modeling for time series analysis, existing methods assumed a strong inductive bias on temporal states. The conditional future state $p(\mathbb{T}|\mathbb{O})$ given past observations $\{y_{s_i}\}_{s_i\in\mathbb{O}}$ is related to the temporal difference between $\mathbb{T}$ and $\mathbb{O}$. For example, GRU-D (Che et al., 2018) suggested a temporal decay module in the recurrent model that can exponentially decrease the influence of past observations. Another strand of works (Mei & Eisner,

[2]We abused notation for clarity. Please refer to Section A.2 for precise information.

Methods	BAQD MSE ↓ / NLL ↑	Speech Commands MSE ↓ / NLL ↑	Physionet MSE ↓ / NLL ↑
RNN-VAE	0.4493 / −1.878	0.5205 / −2.234	0.6494 / −2.878
GRU-D	0.4299 / −1.781	0.4721 / −1.992	0.4403 / −1.832
Latent ODE (RNN-Enc)	0.4058 / −1.660	0.5098 / −2.180	0.5313 / −2.288
Latent ODE (ODE-Enc)	0.3839 / −1.551	0.4950 / −2.106	0.5046 / −2.154
Latent SDE (RNN-Enc)	0.4049 / −1.655	0.5003 / −2.132	0.5301 / −2.282
Latent SDE (ODE-Enc)	0.3806 / −1.534	0.4980 / −2.121	0.4966 / −2.114
NJ-ODE	0.4803 / −2.033	0.5314 / −2.288	0.5167 / −2.214
Res-Flow	0.4379 / −1.821	0.4883/ − 2.073	0.4785 / −2.025
GRU-Flow	0.4853 / −2.054	0.4979 / −2.120	0.4417 / −1.840
Neural Laplace	0.3633 / −1.447	0.4547 / −1.918	0.5550 / −2.392
CRU	0.3613 / −1.438	0.4464 / −1.863	0.5081 / −2.172
CSDE-TP	0.3481 / −1.371	0.4460 / −1.861	0.3958 / −1.610
CooPredict	**0.2886** / **−1.074**	**0.4128** / **−1.695**	**0.3892** / **−1.578**

Table 1: Evaluation of Prediction task on the Air Quality/Speech Commands/Physionet datasets. The best results are highlighted in **bold**. The second best results are colored blue. Evaluations metrics are scaled by 10^{-2} and 10^{-3}, respectively.

2017; Zuo et al., 2020; Chen et al., 2020) directly parameterized temporal point processes (*e.g.*, Hawkes process) that intrinsically assume the temporal decay with exponential intensity kernels. More recently, Neural CDE (Kidger et al., 2020) applied the cubic spline that combines information of temporally adjacent observations to enhance the representational power of the law data. To the best of our knowledge, we propose the first DE-based framework that fully utilizes the temporal correlation without making any inductive bias on the temporal states.

5 Experiments

Benchmarks. We compared CooPredict against a broad range of DE based continuous dynamical models. The dynamic models include GRU-D (Che et al., 2018), Latent ODE (Rubanova et al., 2019), Latent SDE (Li et al., 2020), NJ-ODE (Herrera et al., 2021), Res-Flow (Biloš et al., 2021), GRU-Flow (Biloš et al., 2021), CRU (Schirmer et al., 2022), Neural Laplace (Holt et al., 2022), and CSDE-TP (Park et al., 2022). We implemented the benchmarks using open-source codes published by authors except for Latent SDE whose decoder architecture was replaced from Neural ODE to SDE (Li et al., 2020). Further details on hyper-parameters and implementation can be found in Appendix A.6.

Experimental Settings. We evaluated the time series prediction performance of CooPredict and the benchmarks on multiple real-world datasets: BAQD (Zhang et al., 2017), Speech (Warden, 2018), and Physionet (Silva et al., 2012). More detailed descriptions of these datasets are provided in Appendix A.6.2. We split each time series in the interval $[0, T]$ into two sub-intervals: the first 80% as the observation interval, *i.e.*, $\mathbb{O} = [0, 0.8T]$, and the remaining 20% as the prediction interval, *i.e.*, $\mathbb{T} = [0.8T, T]$. We split time series samples into two halves as training/evaluation sets in Physionet. For BAQD and Speech datasets, we divided time series samples into 80/20 training/testing splits for training and evaluation, respectively. For fair comparisons, we normalized data features by utilizing a min-max normalization to ensure all data features lie within a unit cube $[0, 1]^d$. We used temporally averaged mean square errors (MSEs) and negative log-likelihoods (NLLs) as the performance metrics. To evaluate the performance of the benchmarks, we followed an identical protocol suggested by (Rubanova et al., 2019).

5.1 Time Series Prediction

Table 1 reports the performance comparison of our method and benchmarks. As can be seen in the table, our model significantly outperforms all the benchmarks for the evaluated datasets by a large margin. Notably, most of the existing methods based on latent encoding fail to make accurate predictions. We argue that the reason for such performance degradation is the absence of the calibration process of training data. More specifically, in prior works, the past observations are indiscriminately encoded in the latent space without awareness of temporal uncertainty of irregular time series. For that reason, the model is vulnerable to the temporally noisy environment during

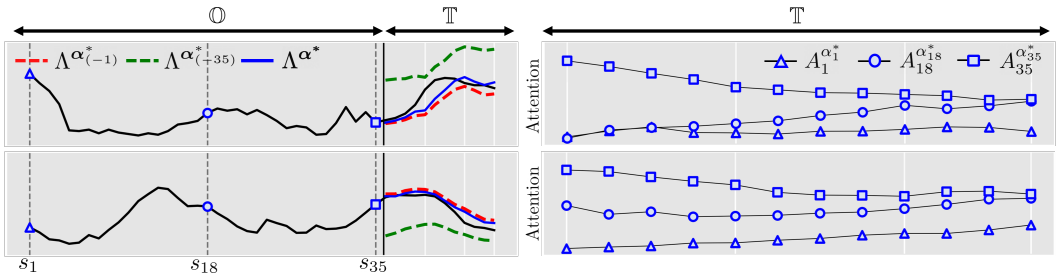


Figure 2: Qualitative result on BAQD dataset. (Left) Sampled trajectories excluding agents α^1 (red) and α^{35} (green), respectively. The prediction trajectory with full information is highlighted as (blue). (Right) The decision attention function on the prediction time interval (*i.e.*, $\boldsymbol{A}_t^{\alpha^*}$).

testing, which may cause a large generalization gap. In contrast, neural agents in our method actively make cooperative decisions to calibrate the vicious effect of unseen data. If the agent detects harmful signals, they voluntarily restrain the decision to follow the group rationality (*i.e.*, accurate prediction). Eventually, the cooperative group robustly suppresses the non-informative temporal dynamics in the inference. Qualitative analysis on the robustness to out-of-distribution noises is provided in the following subsection.

When compared to the baseline model, *i.e.*, CSDE-TP, the key distinction of our method is the ability to discriminate the importance of past observations. More specifically, the baseline model, *i.e.*, CSDE-TP, disregards the effect of temporal association in the model design and simply leverages uniform attention to predict future time series. Contrarily, our method delicately captures the intrinsic temporal association by learning from data such that the cooperative group adaptively filters out the impact of redundant and noisy signals. This results in a significant performance improvement over CSDE-TP ($\approx 20\%$). We further investigate how informative the learned temporal associations is in the next subsection.

5.2 Ablation Study

In this subsection, we provide ablation studies that model four distinct scenarios each of which mimics different types of temporal dynamics in real-world time series: decay, impulse, delay, and noisy observation. For each scenario, we investigate the advantages of the proposed cooperative prediction and the expressivity of decision attention learned without any induction bias.

(A) Temporal Decay. Figure 2 shows that our decision attention infers *temporal decay*. The agent α_1^* voluntarily decreases its own attention so that the observation at the farthest past merely contributes to the inferred prediction. Such a phenomenon can be clearly observed at the prediction times proximate to the beginning of the prediction interval showing that the rational group assigns high attention to s_{35}. Contrarily, the decision attention becomes less distinguishable when predicting the distant future. This is remarkable since the cooperative agents reach the inductive bias (*i.e.*, $p(y_{t\in\mathbb{T}}|y_{s\in\mathbb{O}}) \propto e^{-|t-s|}$) *a posteriori* without any prior knowledge. Furthermore, the performance is drastically worsened after excluding the agent α_{35}^* (green) in the decision-making whereas no meaningful performance drop can be observed after excluding agent α_1^* (red), clearly showing that the decision $X_t^{1,\boldsymbol{\alpha}^*}$ is less informative than others.

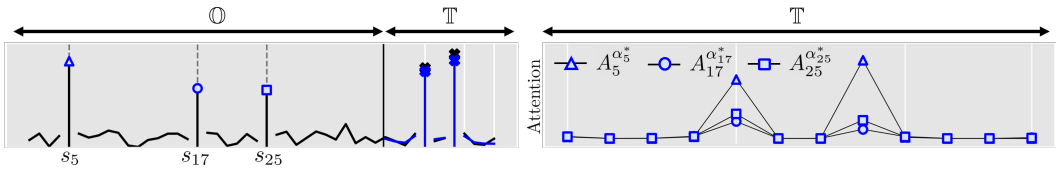


Figure 3: Qualitative Result on Gaussian-Impulse Noises.

(B) Temporal Impulse. Modeling physical phenomenon with point processes (*e.g.*, Hawkes process) is common in many domains including finance for a high-frequency market (Bacry et al., 2015) and geology for earthquake (Ogata, 1998). In this experiment, we utilized Gaussian-impulse noises to mimic such discrete noisy events occurring at random times[3]. Figure 3 illustrates the results on the

[3]Please refer to Appendix A.6.2 for detailed information.

proposed decision attention of neural agents. Given impulse signals in the observation interval $\mathbb{O}$, neural agents accurately restore random peaks. Interestingly, neural agents $\{\alpha_5, \alpha_{17}, \alpha_{25}\}$ that encode the past impulse signals focus on the times at which impulse noise occurs. In other words, neural agents emulate stochastic peaks by learning intrinsic temporal correlations (*i.e.*, $p(y_{t\in\mathbb{T}}|y_{s\in\mathbb{O}}), t, s \sim \texttt{Pois}$) between the random occurrence times of the past and the future impulse signals.

(C) Temporal Delay. In this experiment, we consider a dynamical system based on Mackey–Glass DDE (Mackey & Glass, 1977). The simulated trajectory is highly sensitive to given initial conditions, which influence the future values in a time-delayed manner. Thus, capturing the long-term temporal dependency is the key for accurate time series prediction. In Table 2, we compare our method to benchmarks reported in (Holt et al., 2022) using the same experimental setting. The result shows that our method outperforms all the evaluated benchmarks and effectively captures the delayed effect.

Method	RMSE ↓
CSDE-TP	0.759
Latent ODE*	0.385
Latent SDE	0.366
Coupling-Flow*	0.539
Res-Flow*	0.350
Neural Laplace*	0.282
CRU	0.272
CooPredict	**0.214**

Table 2: Mackey-Glass DDE.

(D) Robustness to OOD Noise. To show the robustness of our method to the out-of-distribution noise unseen during the training time, we set an experiment suggested in (Deng et al., 2021). During the test time, we generate a set of time stamps from a Poisson process $\{\tau\} \sim \texttt{Pois}(\lambda) \in \mathbb{O}$. Then, the original values at the sampled time stamps are replaced by a constant value in the range of $[0.0, 1.0]$. We set the intensity level λ for the Poisson process to 7.0 in all experiments. Figure 4 shows that our method provides the most robust performance that outperforms the prior works, while other methods often fail as the noise level increases. This is because our method can compensate for the information loss by relying on the decisions of the remaining agents. That is, when an observation is regarded as a noisy signal, our decision attention simply filters out the impact of that observation when predicting the future values. Contrarily, existing methods (*e.g.*, Latent ODE) are incapable of dealing with the unseen noises since there is no mechanism that can calibrate the impact of those noisy observations. Figure 4 highlights that the existing methods lose the expressivity due to increased noise levels while our method maintains superior performance even under high noise levels.

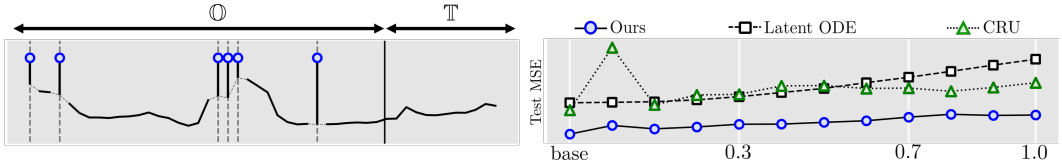


Figure 4: Robustness to Out-of-distribution Noise.

Learning Temporal Relevance from Data. Based on the investigations of the four scenarios, we conclude that the proposed neural agents learn different types of temporal dynamics from the data by cooperatively pursuing the group goal. The rational group filters out the influence of observations that are redundant for predicting future states. This encourages neural agents to focus more on the informative signals for accurate predictions by capitalizing on the data statistics.

6 Conclusion and Future Work

In this paper, we proposed a novel framework for time series prediction as an application of cooperative differential games. The new formulation is built upon the multi-agent dynamics called MaSDEs where each individual agent encodes the partial information from past observations. Under the shared goal, neural agents collaborate to achieve the Nash equilibrium which describes the optimal action states that provide the most accurate future prediction. Theoretical analysis shows the convergence of the proposed novel fictitious play highlighting the effectiveness of the proposed cooperative game.

Future Work. Beyond the basic form of objective functional, the behavioral rule for neural agents can be delicately redesigned for a specified goal of interest. For example, one may extend the proposed framework to the cooperative *mean-field game* (Carmona & Delarue, 2018) where infinitely many agents are considered. In this direction, the system can provide mathematical formalism to understand the individual influences of an infinitely large number of observations. This direction may open a new way to understand the long-term dependency problem of modern recurrent networks. Further, we hope this can bring the application of recent studies on mean-field fictitious play (Perrin et al., 2020; Guo et al., 2019; Min & Hu, 2021) to time series analysis.

REFERENCES

Emmanuel Bacry, Iacopo Mastromatteo, and Jean-François Muzy. Hawkes processes in finance. *Market Microstructure and Liquidity*, 1(01):1550005, 2015.

Marin Biloš, Johanna Sommer, Syama Sundar Rangapuram, Tim Januschowski, and Stephan Günnemann. Neural flows: Efficient alternative to neural ODEs. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, 2021.

Pierre Cardaliaguet and Saeed Hadikhanloo. Learning in mean field games: the fictitious play. *ESAIM: Control, Optimisation and Calculus of Variations*, 2017.

R. Carmona and F. Delarue. *Probabilistic Theory of Mean Field Games with Applications I: Mean Field FBSDEs, Control, and Games*. Probability Theory and Stochastic Modelling. Springer International Publishing, 2018. ISBN 9783319589206.

René Carmona. *Lectures on BSDEs, Stochastic Control, and Stochastic Differential Games with Financial Applications*. Society for Industrial and Applied Mathematics, 2016.

Zhengping Che, Sanjay Purushotham, Kyunghyun Cho, David Sontag, and Yan Liu. Recurrent neural networks for multivariate time series with missing values. *Scientific reports*, 2018.

Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. Neural ordinary differential equations. In *NeurIPS*, 2018.

Ricky TQ Chen, Brandon Amos, and Maximilian Nickel. Neural spatio-temporal point processes. In *International Conference on Learning Representations*, 2020.

Constantinos Daskalakis, Paul W Goldberg, and Christos H Papadimitriou. The complexity of computing a nash equilibrium. *SIAM Journal on Computing*, 39(1):195–259, 2009.

Ruizhi Deng, Marcus A Brubaker, Greg Mori, and Andreas Lehrmann. Continuous latent process flows. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, 2021.

Emilien Dupont, Arnaud Doucet, and Yee Whye Teh. Augmented neural odes. *NeurIPS*, 2019.

Paul W Goldberg. A survey of ppad-completeness for computing nash equilibria. *Cambridge University Press*, pp. 51, 2011.

Xin Guo, Anran Hu, Renyuan Xu, and Junzi Zhang. Learning mean-field games. *Advances in Neural Information Processing Systems*, 32, 2019.

Jiequn Han and Ruimeng Hu. Deep fictitious play for finding Markovian Nash equilibrium in multi-agent games. In *Proceedings of The First Mathematical and Scientific Machine Learning Conference*, 2020.

Calypso Herrera, Florian Krach, and Josef Teichmann. Neural jump ordinary differential equations: Consistent continuous-time prediction and filtering. In *International Conference on Learning Representations*, 2021.

Samuel I Holt, Zhaozhi Qian, and Mihaela van der Schaar. Neural laplace: Learning diverse classes of differential equations in the laplace domain. In *International Conference on Machine Learning*, pp. 8811–8832. PMLR, 2022.

Steffen Jørgensen, Guiomar Martín-Herrán, and Georges Zaccour. Dynamic games in the economics and management of pollution. *Environmental Modeling Assessment*, 15:433–467, 12 2010. doi: 10.1007/s10666-010-9221-7.

Patrick Kidger, James Morrill, James Foster, and Terry Lyons. Neural controlled differential equations for irregular time series. In *NeurIPS*, 2020.

George Leitmann. *Cooperative and non-cooperative many players differential games*. Springer, 1974.

Xuechen Li, Ting-Kam Leonard Wong, Ricky TQ Chen, and David Duvenaud. Scalable gradients for stochastic differential equations. In *AISTATS*, 2020.

Michael C. Mackey and Leon Glass. Oscillation and chaos in physiological control systems. *Science*, 197(4300):287–289, 1977. doi: 10.1126/science.267326.

Hongyuan Mei and Jason M Eisner. The neural hawkes process: A neurally self-modulating multivariate point process. *Advances in neural information processing systems*, 2017.

Ming Min and Ruimeng Hu. Signatured deep fictitious play for mean field games with common noise. In *International Conference on Machine Learning*, pp. 7736–7747. PMLR, 2021.

James Morrill, Cristopher Salvi, Patrick Kidger, James Foster, and Terry Lyons. Neural rough differential equations for long time series. *ICML*, 2021.

Yosihiko Ogata. Space-time point-process models for earthquake occurrences. *Annals of the Institute of Statistical Mathematics*, 50(2):379–402, 1998.

Bernt Karsten Øksendal and Agnes Sulem. *Applied stochastic control of jump diffusions*, volume 498. Springer, 2007.

Sung Woo Park, Kyungjae Lee, and Junseok Kwon. Neural markov controlled SDE: Stochastic optimization for continuous-time data. In *International Conference on Learning Representations*, 2022.

Andrey Perelman, Tal Shima, and Ilan Rusnak. Cooperative differential games strategies for active aircraft protection from a homing missile. *Journal of Guidance, Control, and Dynamics*, 34(3): 761–773, 2011.

Sarah Perrin, Julien Pérolat, Mathieu Laurière, Matthieu Geist, Romuald Elie, and Olivier Pietquin. Fictitious play for mean field games: Continuous time analysis and applications. *Advances in Neural Information Processing Systems*, 33:13199–13213, 2020.

Huyen Pham. Feynman-kac representation of fully nonlinear pdes and applications. *Acta Mathematica Vietnamica*, 2015.

Yulia Rubanova, Ricky TQ Chen, and David K Duvenaud. Latent ordinary differential equations for irregularly-sampled time series. In *NeurIPS*, 2019.

Mona Schirmer, Mazin Eltayeb, Stefan Lessmann, and Maja Rudolph. Modeling irregular time series with continuous recurrent units. In *International Conference on Machine Learning (ICML)*, 2022.

Richard J Sexton. The formation of cooperatives: a game-theoretic approach with implications for cooperative finance, decision making, and stability. *American Journal of Agricultural Economics*, 68(2):214–225, 1986.

Ikaro Silva, George Moody, Daniel Scott, Leo Celi, and Roger Mark. Predicting in-hospital mortality of icu patients: The physionet/computing in cardiology challenge 2012. *Computing in cardiology*, 2012.

John M Staatz. The cooperative as a coalition: a game-theoretic approach. *American Journal of Agricultural Economics*, 65(5):1084–1089, 1983.

Pete Warden. Speech commands: A dataset for limited-vocabulary speech recognition. *arXiv preprint arXiv:1804.03209*, 2018.

David WK Yeung and Leon A Petrosjan. *Cooperative stochastic differential games*. Springer Science & Business Media, 2006.

David WK Yeung and Leon A Petrosyan. *Subgame consistent economic optimization: an advanced cooperative dynamic game analysis*. Springer, 2012.

Shuyi Zhang, Bin Guo, Anlan Dong, Jing He, Ziping Xu, and Song Chen. Cautionary tales on air-quality improvement in beijing. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Science*, 473:20170457, 09 2017.

Simiao Zuo, Haoming Jiang, Zichong Li, Tuo Zhao, and Hongyuan Zha. Transformer hawkes process. In *International conference on machine learning*. PMLR, 2020.

A APPENDIX

A.1 MATHEMATICAL BACKGROUND

In this section, we briefly recall some basic elements of stochastic optimal controls needed in the paper. We start by introducing the *backward stochastic differential equation* used in Section 3. In particular, the adjoint variables $(Y_t^{i,\boldsymbol{\alpha}}, Z_t^{i,\boldsymbol{\alpha}})$ are derived from the classical theory of non-linear Feynman-Kac formula (Carmona & Delarue, 2018):

$$dY_t^{i,\boldsymbol{\alpha}} = -H^i(t, \mathbf{X}_t^{\boldsymbol{\alpha}}, F_t^{i,\boldsymbol{\alpha}}, [\alpha_i, \boldsymbol{\alpha}_{(-i)}])dt + Z_t^{i,\boldsymbol{\alpha}} \cdot dB_t^i, \tag{A.1}$$

where the stochastic Hamiltonian system with uncontrolled volatility for i-th agent is defined as follows:

$$H^i(t, \mathbf{X}_t^{\boldsymbol{\alpha}}, F_t^{i,\boldsymbol{\alpha}}, [\alpha_i, \boldsymbol{\alpha}_{(-i)}]) = \left[\sigma(t, X_t^{i,\boldsymbol{\alpha}})^{-T} Z_t^{i,\boldsymbol{\alpha}}\right] \cdot b(t, X_t^{i,\boldsymbol{\alpha}}, \alpha_i) + h^i(t, \mathbf{X}_t^{\boldsymbol{\alpha}}, \boldsymbol{\alpha}). \tag{A.2}$$

For simplicity, we define the auxiliary function F having following form:

$$F^{i,\boldsymbol{\alpha}}(t, X_t^{i,\boldsymbol{\alpha}}, \alpha_i) \coloneqq \sigma(t, X_t^{i,\boldsymbol{\alpha}})^{-T} b(t, X_t^{i,\boldsymbol{\alpha}}, \alpha_i). \tag{A.3}$$

Next, we introduce the diriftless stochastic differential equation:

$$dX_t^{i,\boldsymbol{\alpha}} = \sigma(t, X_t^{i,\boldsymbol{\alpha}})dB_t^i. \tag{A.4}$$

Then, the Girsanov's theorem gives the following Radon-Nikodym derivative:

$$\frac{d\mathbb{P}}{d\mathbb{Q}} = \varepsilon\left(\int \sigma(t, X_t^{i,\boldsymbol{\alpha}})^{-1} b(t, X_t^{i,\boldsymbol{\alpha}}, \alpha_i) \cdot dB_t^i\right)_T, \tag{A.5}$$

where we denote $M^{-T} \coloneqq [M^{-1}]^T$ for any inevitable matrix M. Note that the regularity conditions **(H1)** are essential to assure the existence of the Doléans-Dade exponential ε which defines the stochastic exponential for the (local) martingale $dM_t^i = F^{i,\boldsymbol{\alpha}}(t, X_t^{i,\boldsymbol{\alpha}}, \alpha_i) \cdot dB_t^i$:

$$\varepsilon(M^i)_T \coloneqq e^{M_t^i - M_0^i - \frac{1}{2}[M^i, M^i]_t} = e^{\int_0^t F^{i,\boldsymbol{\alpha}}(u, X_u^{i,\boldsymbol{\alpha}}, \alpha_i) \cdot dB_u^i - \frac{1}{2}\int_0^t |F^{i,\boldsymbol{\alpha}}|^2(u, X_u^{i,\boldsymbol{\alpha}}, \alpha_i)du}. \tag{A.6}$$

The second equality holds since $M_0^i = 0$ and the quadratic variation is calculated as $[M^i, M^i]_t = \int_0^t |F^{i,\boldsymbol{\alpha}}|^2 ds < \infty$. Then, by definition, the transformed representation W_t^i in following equation is also a Wiener process with respect to the probability measure $\mathbb{P}$.

$$W_t^i = B_t^i - \int_0^t \sigma(s, X_s^{i,\boldsymbol{\alpha}})^{-T} b(s, X_s^{i,\boldsymbol{\alpha}}, \alpha_i)ds = B_t^i - \int_0^t F^{i,\boldsymbol{\alpha}}(t, X_t^{i,\boldsymbol{\alpha}}, \alpha_i)ds. \tag{A.7}$$

Identically, the above relation can be rewritten as following differential form:

$$dW_t^i = -\left[F^{i,\boldsymbol{\alpha}}(t, X_t^{i,\boldsymbol{\alpha}}, \alpha_i)dt - dB_t^i\right]. \tag{A.8}$$

Notably, we can restore the proposed controlled neural SDE from driftless SDE in (A.4) by the relation between W_t^i and B_t^i shown in (A.8).

$$dX_t^{i,\boldsymbol{\alpha}} = b(t, X_t^{i,\boldsymbol{\alpha}}, \alpha_i)dt + \sigma(t, X_t^{i,\boldsymbol{\alpha}})dW_t^i. \tag{A.9}$$

This relation gives the explicit form of adjoint dynamics as follows: $Y_t^{i,\boldsymbol{\alpha}}$:

$$\begin{aligned}
Y_t^{i,\boldsymbol{\alpha}} &= \Psi^i(X_T^{i,\boldsymbol{\alpha}}) + \int_t^T H^i(t, \mathbf{X}_t^{\boldsymbol{\alpha}}, F_t^{i,\boldsymbol{\alpha}}, [\alpha_i, \boldsymbol{\alpha}_{(-i)}])ds - \int_t^T Z_s^{i,\boldsymbol{\alpha}} \cdot dB_s^i \\
&= \Psi^i(X_T^{i,\boldsymbol{\alpha}}) + \int_t^T h^i(t, \mathbf{X}_t^{\boldsymbol{\alpha}}, \boldsymbol{\alpha}) + \left(\sigma(s, X_s^{i,\boldsymbol{\alpha}})^{-T} Z_s^{i,\boldsymbol{\alpha}}\right) \cdot b(s, X_s^{i,\boldsymbol{\alpha}}, \alpha_i)\big)\, ds - \int_t^T Z_s^{i,\boldsymbol{\alpha}} \cdot dB_s^i \\
&= \Psi^i(X_T^{i,\boldsymbol{\alpha}}) + \int_t^T h^i(t, \mathbf{X}_t^{\boldsymbol{\alpha}}, \boldsymbol{\alpha})ds + \int_t^T Z_s^{i,\boldsymbol{\alpha}} \cdot \left[F^{i,\boldsymbol{\alpha}}(s, X_s^{i,\boldsymbol{\alpha}}, \alpha_i)ds - dB_s^i\right] \\
&= \Psi^i(X_T^{i,\boldsymbol{\alpha}}) + \int_t^T h^i(t, \mathbf{X}_t^{\boldsymbol{\alpha}}, \boldsymbol{\alpha})ds - \int_t^T Z_s^{i,\boldsymbol{\alpha}} \cdot dW_s^i.
\end{aligned} \tag{A.10}$$

In the last equality, the Brownian motion is changed from B_s^i to W_s^i, and the generic form of BSDE for tuple $(\mathbb{P}, W_t^i)$ is presented. Since the third term in last line $\int Z_s^{i,\boldsymbol{\alpha}} \cdot dW_s^i$ is a martingale with respect to the measure $\mathbb{P}$, we can identify the conditional expectation with original objective functional J^i in (3) as follows:

$$\mathcal{J}^i = \mathbb{E}[Y_t^{i,\boldsymbol{\alpha}} | \tilde{\mathcal{F}}_t] = \mathbb{E}_{\mathbb{P}} \left[\Psi^i(X_T^{i,\boldsymbol{\alpha}}) + \int_t^T h^i(t, \mathbf{X}_t^{\boldsymbol{\alpha}}, \boldsymbol{\alpha}) ds | \mathcal{F}_t \right], \tag{A.11}$$

where $\tilde{\mathcal{F}}_t$ is the augmented filtration according to the Brownian motion B_t^i.

A.2 Proofs

Assumptions. Throughout the appendix, we make following assumptions for the proof.

- **(H1)** $\sigma^{-T}(t, x, \alpha(\cdot, \cdot; \theta)) b(t, x, \alpha(\cdot, \cdot; \theta))$ is uniformly bounded, twice differentiable, Lipschitz continuous.
- **(H2)** The k-th derivatives of neural agents with respect to both spatial and temporal variables are Lipschitz continuous to the parameter space,

$$||\nabla^{(k)} \alpha_i(\cdot, x; \theta) - \nabla^{(k)} \alpha_i(\cdot, x; \tilde{\theta})|| + ||\partial_t^{(k)} \alpha_i(t, \cdot; \theta) - \partial_t^{(k)} \alpha_i(t, \cdot; \tilde{\theta})|| \leq L_i ||\theta - \tilde{\theta}||. \tag{A.12}$$

  for all $i \in \{1, \ldots, I\}$, $0 \leq k \leq 2$, $\forall \theta, \tilde{\theta} \in \Theta$.

- **(H3)** The expectation of Frobenius norm of adjoint variable $Z_t \in \mathbf{Sym}_+^I$ [4] is bounded, $\mathbb{E} \left\| Z_t \right\|_F < D_2$.

Lemma 1. *(Grönwall's Inequality (Jacod & Shiryaev, 2013)) The left inequality induces the inequality in right-hand side:*

$$B(t) \leq A + \int_{\mathbb{T}} B(s) C(s) ds \quad \longrightarrow \quad B(t) \leq A e^{\int_{\mathbb{T}} C(s) ds}. \tag{A.13}$$

A.2.1 Proof of Proposition 1

The first step to showing the convergence of neural agents towards Nash equilibrium is to obtain the deviation of the adjoint variable Y_t during the fictitious play. For this, we first introduce the HJBE of decoupled SDE system. In particular, the proposed system trains neural agents to solve the individual decoupled HJBE at each stage m:

$$\mathcal{V}_t^{i,m+1} + \inf_{\alpha_i^m \in \mathbb{A}} H^i(t, x, F_t^{i,\boldsymbol{\alpha}^m}, [\alpha_i^m, \boldsymbol{\alpha}_{(-i)}^m]) + \frac{1}{2} \mathbf{Tr}(\Sigma^T \Sigma \boldsymbol{Hess} \mathcal{V}^{i,m+1}) = 0, \tag{A.14}$$

where the cost functional at the next stage $(i.e., m+1)$ is related to the optimal actions in previous stage that minimizes the decoupled Hamiltonian in (A.14). Then, one of our interest is to investigate the deviation of Hamiltonian system:

$$\delta H_t^{i,m} = \underbrace{H^i(t, \mathbf{X}_t^{\boldsymbol{\alpha}^m}, F_t^{i,\boldsymbol{\alpha}^m}, [\alpha_i^{m+1}, \boldsymbol{\alpha}_{(-i)}^m])}_{\text{Fictitious Play}} - \underbrace{H^i(t, \mathbf{X}_t^{\alpha^*}, F_t^{i,\boldsymbol{\alpha}^*}, \boldsymbol{\alpha}^*)}_{\text{Optimal Agents}}, \tag{A.15}$$

where the deviation shows the difference between Hamiltonian systems that are derived by neural agents lying in optimal and sub-optimal regions, respectively. Similarly, we define the deviations for both adjoint variables:

$$\delta Y_t^{i,m} = Y_t^{i,m} - Y_t^{i,*}, \quad \delta Z_t^{i,m} = Z_t^{i,m} - Z_t^{i,*}, \tag{A.16}$$

Following by the notations in (A.16) and the equality (A.10), the deviation for the adjoint variable $(i.e., \delta Y_t^{i,m})$ at stage m can be represented as the following Itô's differential:

$$d\delta Y_t^{i,m+1} = -\delta H^{i,m} dt + \delta Z_t^{i,m+1} \cdot dB_t^i. \tag{A.17}$$

[4] The matrix manifolds $\mathbf{Sym}_+^I$ is the space of semi-positive definite matrices. In this paper, we regard the Euclidean flat norm $(i.e., \|\cdot\|_F)$ is inherited to this space.

Next, we evaluate the squared norm of $\delta \mathbf{Y}_t^{(\cdot)}$ to investigate the convergence of objective functional

$$d\left\|\delta \mathbf{Y}_t^{m+1}\right\|^2 = -2\delta \mathbf{Y}_t^{m+1} \cdot \delta \mathbf{H}_t^m dt + \left\|\delta \mathbf{Z}_t^{m+1}\right\|_F^2 dt + 2(\delta \mathbf{Z}_t^{m+1} \delta \mathbf{Y}_t^{m+1}) \cdot d\mathbf{B}_t. \tag{A.18}$$

Note that the equality in (A.18) follows by Itô's lemma for squared norm of multi-dimensional representation A_t:

$$d\left\|A_t\right\|^2 = 2A_t \cdot \mu_{A_t} dt + \left\|\sigma_{A_t}\right\|_F^2 dt + 2A_t \sigma_{A_t} \cdot dB_t, \tag{A.19}$$

where μ_{A_t} and σ_{A_t} are the terms corresponding to bounded variations and local martingales of process A_t, respectively. By taking the expectation on both sides of (A.18), we obtain the following result:

$$\begin{aligned}\mathbb{E}\left[\left\|\delta \mathbf{Y}_t^{m+1}\right\|^2\right] &= \mathbb{E}\left[\left\|\delta \mathbf{Y}_T^{m+1}\right\|^2\right] + \mathbb{E}\left[\int_{\mathbb{T}} 2\delta \mathbf{Y}_t^{m+1} \cdot \delta \mathbf{H}_t^m - \left\|\delta \mathbf{Z}_t^{m+1}\right\|_F^2 dt\right] \\ &\leq 4\mathbb{E}\left[\left\|\delta \mathbf{X}_T^{m+1}\right\|^2\right] + 2\mathbb{E}\left[\int_{\mathbb{T}} \left\|\delta \mathbf{Y}_t^{m+1}\right\|^2 \left\|\delta \mathbf{H}_t^m\right\|^2 dt\right] - \mathbb{E}\left[\int_{\mathbb{T}} \left\|\delta \mathbf{Z}_t^{m+1}\right\|_F^2 dt\right].\end{aligned} \tag{A.20}$$

Owing to the characteristic of backward SDE (*i.e.*, $\mathbf{Y}_t^m$), the integral sign is reversed in the second term in (A.20). The expectation with respect to the local martingale term vanishes as set of Wiener processes $\mathbf{B}_t$ are related to $\mathbb{Q}$. Since BSDE imposes the terminal constraint $\mathbf{Y}_T^m = \mathbf{\Psi}^m$, $\mathbf{Y}_T^* = \mathbf{\Psi}^*$, the following result is given by Lipschitzness of $\mathbf{\Psi}$ as:

$$\mathbb{E}\left[\left\|\delta \mathbf{Y}_T^{m+1}\right\|^2\right] = \mathbb{E}\left[\left\|\mathbf{\Psi}^{m+1} - \mathbf{\Psi}^*\right\|^2\right] \leq 4\mathbb{E}\left[\left\|\delta \mathbf{X}_T^{m+1}\right\|^2\right]. \tag{A.21}$$

This shows the inequality in (A.20). By rearranging the relation and applying Hölder's inequality we have

$$\mathbb{E}_{\mathbb{Q}}\left[\left\|\delta \mathbf{Y}_t^{m+1}\right\|^2 + \left\|\delta \mathbf{Z}_t^{m+1}\right\|_F^2\right] \leq 4\mathbb{E}_{\mathbb{Q}}\left[\left\|\delta \mathbf{X}_T^{m+1}\right\|^2\right] + 2\mathbb{E}_{\mathbb{Q}}\left[\int_{\mathbb{T}} \left\|\delta \mathbf{Y}_t^{m+1}\right\|^2 dt\right] \mathbb{E}_{\mathbb{Q}}\left[\int_{\mathbb{T}} \left\|\delta \mathbf{H}_t^m\right\|^2 dt\right]. \tag{A.22}$$

Finally, we apply Grönwall's inequality in Lemma 1 to above inequality, we obtain

$$\mathbb{E}_{\mathbb{Q}}\left[\left\|\delta \mathbf{Y}_t^{m+1}\right\|^2\right] \leq \mathbb{E}_{\mathbb{Q}}\left[\left\|\delta \mathbf{X}_T^{m+1}\right\|^2\right] e^{|\mathbb{T}| \ln 16 \mathbb{E}\left[\int_{\mathbb{T}} \|\delta \mathbf{H}_t^m\|^2 dt\right]}. \tag{A.23}$$

Now, our next step is to bound the right-hand side of (A.23). We start by estimating an upper bound of the following time averaged mean-squared Hamiltonian (A.1):

$$\mathbb{E}\left[\int_{\mathbb{T}} \left\|\delta \mathbf{H}_t^m\right\|^2 dt\right] \leq \mathbb{E}_{\mathbb{P}}\left[\int_{\mathbb{T}} \left\|\delta \mathbf{F}_t^m \cdot \delta \mathbf{Z}_t^{m+1} + \delta \mathbf{h}_t^m\right\|^2 dt\right]. \tag{A.24}$$

For this, we rearrange the Hamiltonian deviation $\delta \mathbf{H}_t^m$ by inserting two terms $\delta \mathbf{F}_t^m \cdot \delta \mathbf{Z}_t^m$ and $\delta \mathbf{h}_t^m$. In this case, Hamiltonian deviation is rewritten as

$$\begin{aligned}\delta H_t^{i,m} = {} & F(t, \mathbf{X}_t^\alpha, [\alpha_i^{m+1}, \boldsymbol{\alpha}_{(-i)}^m]) \cdot (Z_t^{i,m+1} - Z_t^{i,*}) \\ & + (F(t, \mathbf{X}_t^\alpha, [\alpha_i^{m+1}, \boldsymbol{\alpha}_{(-i)}^m]) - F(t, \mathbf{X}_t^\alpha, \boldsymbol{\alpha}^m)) \cdot Z_t^{i,*} \\ & + (F(t, \mathbf{X}_t^\alpha, \boldsymbol{\alpha}^m) - F(t, \mathbf{X}_t^\alpha, \boldsymbol{\alpha}^*)) \cdot Z_t^{i,*} \\ & + h^i(t, \mathbf{X}_t^\alpha, [\alpha_i^{m+1}, \boldsymbol{\alpha}_{(-i)}^m]) - h^i(t, \mathbf{X}_t, \boldsymbol{\alpha}^m) + h^i(t, \mathbf{X}_t^\alpha, \boldsymbol{\alpha}^m) - h^i(t, \mathbf{X}_t^\alpha, \boldsymbol{\alpha}^*).\end{aligned} \tag{A.25}$$

Here, we define the Lipschitz constants of three different objects as follows:

$$\mathbf{Lip}[\cdots] = C_n, \qquad \{\cdots\} \in \{F^{i,m}, \sigma^i, h^i\}, \quad n = 1, 2, 3. \tag{A.26}$$

By the Lipschitz continuity defined above (A.26), one can obtain the squared norm of the Hamiltonian deviation:

$$\begin{aligned}\left\|\delta \mathbf{H}_t^m\right\|^2 \leq {} & C_1 \left\|F\right\|^2 \left\|\mathbf{Z}_t^{m+1} - \mathbf{Z}_t^*\right\|_F^2 \\ & + C_1 \left\|\alpha_i^{m+1} - \alpha_i^m\right\|^2 \cdot \left\|\mathbf{Z}_t^*\right\|_F + C_1 \sum_i^I \left\|\boldsymbol{\alpha}^m - \boldsymbol{\alpha}^*\right\|^2 \cdot \left\|Z_t^{i,*}\right\|_F \\ & + C_3 \left\|\alpha_i^{m+1} - \alpha_i^m\right\|^2 + C_3 \left\|\boldsymbol{\alpha}^m - \boldsymbol{\alpha}^*\right\|^2.\end{aligned} \tag{A.27}$$

Consider we have a proper $M = C_1D_1D_2 \vee 2(C_1D_2 \vee C_3) \vee 2I(C_1D_2 \vee C_3)$ from the definitions of pre-determined constants C_1, C_2, C_3, D_1 and D_2. Then, the above inequality can be rewritten in a compact form:

$$\mathbb{E}\left[\|\delta \mathbf{H}_t^m\|^2\right] \leq M\left[\mathbb{E}\left\|\boldsymbol{\alpha}^{m+1} - \boldsymbol{\alpha}^m\right\|^2 + \mathbb{E}\left\|\boldsymbol{\alpha}^m - \boldsymbol{\alpha}^*\right\|^2\right]. \tag{A.28}$$

Note that the equality $\left\|[\beta_i, \boldsymbol{\alpha}^m_{(-i)}] - \boldsymbol{\alpha}^{m,*}\right\| = \left\|\beta_i - \alpha_i^{m,*}\right\|$ holds for an arbitrary action $\beta \in \mathbb{A}$. Finally, we conclude the result (A.28) by showing following inequalities:

$$\begin{cases} \mathbb{E}\left[\sum_i^I \|\delta \mathbf{Z}_t^*\|_F^2\right] \leq I \sup_i \mathbb{E}\left[\left\|\delta Z_t^{i,*}\right\|_F^2\right] \leq 2ID_2, \\ \\ \sup_{t,(\boldsymbol{\alpha}\in\mathbb{A})} \mathbb{E}\left[\|F(t,x,\boldsymbol{\alpha})\|^2\right] \leq C_1\mathbb{E}\left[\left\|F(0,0,\mathbf{0})^2\right\|\right] \leq D_1. \end{cases} \tag{A.29}$$

In previous contents, the detailed convergent states of adjoint variable Y_t^* are not specified. To continue our discussion from (A.28), we develop a gradient descent-based update rule for training neural agents. For any neural parameters $\theta \in \Theta$, we assume that the i-th agent's action $\theta_i \to \alpha_i(\cdot,\cdot;\theta_i) \triangleq \alpha_i(\theta_i)$ lies in the compact subset $A \in \mathbb{A}(\epsilon) \in \mathcal{L}^2([0,T] \times \mathbb{R}^d)$ (*i.e.*, $\int ||\alpha_i(t,x;\theta_i)||^2 dt dp_t(x) dx < \infty$) for some $\epsilon > 0$. Then, we introduce the following notations:

$$\alpha_i^{m,k} \coloneqq \alpha_i^m(t,x;\theta_i^m(k)), \quad \boldsymbol{\alpha}^{m,k} \coloneqq \{\alpha_i^{m,k}\}_{1\leq i\leq d}, \tag{A.30}$$

where the auxiliary notation $u \in \mathbb{R}^+$ is an indicator for gradient descent steps. Similarly, we define the notations $(Y^{i,m,k}, \mathbf{Y}^{m,k})$ and $(H^{i,m,k}, \mathbf{H}^{m,k})$. Let us define the operator $\mathcal{B} : \mathbb{N}^+ \to \mathcal{L}^2(\mathbb{T} \times \mathbb{R}^d)$ as $\mathcal{B}[\theta_i^m(k)] \coloneqq \alpha_i^{m,k+1}$, and if $k > K$, then we denote $\alpha_i^{m+1} \coloneqq \alpha_i^{m+1,k=0}$. By Lipschitz continuity of neural agents, we have

$$\begin{aligned} \|\mathcal{B}[\theta_i^m(k)] - \alpha_i^m\|_{\mathcal{L}^2}^2 &= \mathbb{E}\left[\|\alpha_i^m(t,x;\theta_i^m(k+1)) - \alpha_i^m(t,x;\theta_i^m(k))\|^2\right] \\ &\leq L^2(i)\mathbb{E}\left[\|\theta_i^m(k) - \theta_i^m(k+1)\|^2\right] \\ &\leq \tilde{L}^2\mathbb{E}\left[\left\|\gamma\nabla_{\theta_i}\mathbb{E}_{\mathbb{Q}}[Y_t^{i,m,k}|\tilde{F}_t]\right\|^2\right] \\ &\leq \tilde{L}^2\gamma^2\mathbb{E}\tilde{\mathbb{E}}_{\mathbb{Q}}\left[\left\|\nabla_{\theta_i}Y_t^{i,m,k}\right\|^2|\tilde{\mathcal{F}}_t\right] \\ &\leq \tilde{L}^2\gamma^2\mathbb{E}\tilde{\mathbb{E}}\left[\left\|\nabla_{\theta_i}h^{i,m,k}\right\|^2|\mathcal{F}_t\right] \\ &= \tilde{L}^2\gamma^2\mathbb{E}\left[\left\|\nabla_{\theta_i}h^{i,m,k}\right\|^2\right], \end{aligned} \tag{A.31}$$

where $\tilde{L} = \max_i L_i$. In the fourth inequality, the probability measure for the integration is switched from $\mathbb{Q}$ to $\mathbb{P}$ with the Radon-Nikodym $d\mathbb{P}/d\mathbb{Q}$. Since the cost function is uniformly bounded with the vanishing derivative of terminal cost (*i.e.*, $\partial_{\theta_i}g^i(X_T) = 0$), the expectation is well-defined. It is worth noting that the filtration $\mathcal{F}_t$ contains the information of past observations $\{y_{s_i}\}$ as we impose constraints to satisfy $X_{s_i}^{i,\alpha} = y_{s_i}$ almost surely for neural agents.

As a next step, we evaluate the upper bound of mean-squared evaluation as follows:

$$\begin{aligned} &\mathbb{E}\left[\left\|\nabla_{\theta_i}h^{i,m,k}\right\|^2\right] \\ &\quad\leq 4\mathbb{E}\left[\left\|\langle \mathbf{A}^{\boldsymbol{\alpha}^m}(t), \mathbf{X}_t^{\boldsymbol{\alpha}^m}\rangle - y_t\right\|^2\left\|\nabla_{\theta_i}A_i^{\alpha_i^m(\cdot,\cdot;\theta_i^m)}(t)X_t^{i,\alpha_i^m} + A_i^{\alpha_i^m}(t)\nabla_{\theta_i}X_t^{i,\alpha_i^m(\cdot,\cdot;\theta_i)}\right\|^2\right] \\ &\quad\leq 4\mathbb{E}\left[\left\|h^{i,m,k}\right\|^2\left\|\nabla_{\theta_i}A_i^{\alpha_i^m(\cdot,\cdot;\theta_i^m)}(t)X_t^{i,\alpha_i^m} + A_i^{\alpha_i^m}(t)\nabla_{\theta_i}X_t^{i,\alpha_i^m(\cdot,\cdot;\theta_i^m)}\right\|^2\right]. \end{aligned} \tag{A.32}$$

Let us assume that there exist representations $\hat{b}, \hat{\alpha}$ such that $b(t,X_t,\alpha_i) = \hat{b}(t,X_t)\alpha_i(t,X_t;\theta_i)$ and $A_i(t,X_t;\theta_i) = \hat{A}(t)\alpha_i(t,X_t;\theta_i)$. This brings the evaluations of the following two norm bounds:

$$\begin{aligned} \left\|\mathbb{E}\left[\nabla_{\theta_i}X_t^{i,\alpha_i^m}\right]\right\|^2 \leq \mathbb{E}\left[\left\|\nabla_{\theta_i}X_t^{i,\alpha_i^m}\right\|^2\right] &\leq \int_{s_i}^t \mathbb{E}\left[\left\|\hat{b}(u,X_u)\nabla_{\theta_i}\alpha_i^m(u,X_u;\theta_i^m)\right\|^2\right]du \\ &\leq |\mathbb{T}|\tilde{L}^2\mathbb{E}||\hat{b}||^2. \end{aligned} \tag{A.33}$$

Similarly, we have the gradient norm bound of individual decision attention

$$\mathbb{E}\left[\left\|\nabla_{\theta_i} A_i^{\alpha_i^m}\right\|^2\right] = \mathbb{E}\left[\left\|\hat{A}\nabla_{\theta_i}\alpha_i^m(t, X_t; \theta_i^m)\right\|^2\right] \leq \mathbb{E}\left\|\hat{A}^2\right\| \tilde{L}^2. \tag{A.34}$$

For simplicity, suppose that there exists a constant M_2 such that each function $\hat{b}$, $\hat{A}$, $\partial_t \hat{A}$ has expectation norm bound:

$$\mathbb{E}\left[||\hat{b}||^2 + ||\hat{A}||^2 + \left\|\partial_t \hat{A}\right\|^2\right] < M_2. \tag{A.35}$$

The inequality in (A.32) together with evaluations in (A.33), (A.34) yields:

$$\mathbb{E}\left[\left\|\nabla_{\theta_i} h^{i,m,k}\right\|\right] \leq \mathbb{E}\left[\left\|h^{i,m,k}\right\|^2\right] 4\tilde{L}^2 M_2(1 + |\mathbb{T}|). \tag{A.36}$$

This directly gives the $\mathcal{L}^2$-bound of the operator $\mathbb{B}$ during the fictitious play over stages according to the cost function shown in the right-hand side:

$$\|\mathcal{B}[\theta_i^m(k)] - \alpha_i^m\|_{\mathcal{L}^2}^2 \leq \mathbb{E}\left[\left\|h^{i,m,k}\right\|^2\right] 4\tilde{L}^4\gamma^2 M_2(1 + |\mathbb{T}|). \tag{A.37}$$

Let us say that the gradient descent optimizes the neural parameters to achieve small enough values for cost function for $k \geq K$.

$$K = \min_{k\in\mathbb{N}^+}\left\{k; \mathbb{E}\left[\left\|h^{i,m,k}\right\|^2\right] \leq \frac{\upsilon_{i,m}}{4\tilde{L}^4\gamma^2 M_2(1 + |\mathbb{T}|)}\right\}. \tag{A.38}$$

Following by the definition of constant K, the $\mathcal{L}_2$ deviation of neural agent can be bounded as follows:

$$\mathbb{E}\left[\left\|\alpha_i^{m+1} - \alpha_i^m\right\|^2\right] := \|\mathcal{B}[\theta_i^m(K)] - \alpha_i^m\|_{\mathcal{L}^2}^2 \leq \upsilon_{i,m}. \tag{A.39}$$

If we denote $\alpha_i^* = \alpha_i^{m_i^*}$, $m_i^* \in \mathbb{N}^+$, the following relation holds by triangle inequalities:

$$\mathbb{E}_{\mathbb{P}}\left[\|\alpha_i^* - \alpha_i^m\|^2\right] \leq (\mathbf{m}_i^* - m)\sup_m \upsilon_{i,m}. \tag{A.40}$$

As $\upsilon_{i,m} \vee \sup_m \upsilon_{i,m} = \sup_m \upsilon_{i,m}$ for all $m \leq m^*$, the expectation of Hamiltonian deviation can be bounded as a summation of two terms in (A.28):

$$\mathbb{E}\left[\|\delta\mathbf{H}_t^m\|^2\right] \leq M(\mathbf{m}^* - m + 1)\boldsymbol{\upsilon}_m, \tag{A.41}$$

where $\boldsymbol{\upsilon}_m = \{\upsilon_{i,m+1}\}$ and $\mathbf{m}^* = \{m_i^*\}$. By inserting above inequality into (A.23) and replacing $m + 1 \to m$, we obtain

$$\mathbb{E}\left[\|\delta\mathbf{Y}_t^m\|^2\right] \leq \mathbb{E}\left[\|\delta\mathbf{X}_T^m\|^2\right] e^{|\mathbb{T}|^2 \ln 16M(\mathbf{m}^* - m + 2)\boldsymbol{\upsilon}_m}. \tag{A.42}$$

Next, we evaluate the upper-bound of mean-squared decision deviations in (A.42):

$$\mathbb{E}_{\mathbb{Q}}\left[\|\delta\mathbf{X}_T^m\|^2\right] \leq \tilde{L}^2|\mathbb{T}|\,\|\boldsymbol{\theta}^m - \boldsymbol{\theta}^*\|^2. \tag{A.43}$$

By denoting the deviation of parameters as $\kappa_m = \|\delta\boldsymbol{\theta}^m\|^2$, we select small enough values $\upsilon_{i,m}$ that is related to the marginal constants $\epsilon_{i,m}$:

$$\boldsymbol{\upsilon}_m := \frac{\ln\boldsymbol{\epsilon}_m - \ln(\kappa_{m+1}\tilde{L}^2|\mathbb{T}|)}{|\mathbb{T}|^2 \ln 16M(m^* - m + 2)}, \quad \boldsymbol{\epsilon} := \sup_{\{m\leq m^*, i\in\{1,\cdots,I\}\}} \boldsymbol{\epsilon}_{i,m}. \tag{A.44}$$

If the neural agent have large enough capacity to minimize the via gradient descent, these values assures minimal upper bound of the inequality (A.23). Finally, the non-linear Feynman-Kac theorem, which is related to the equation (A.10), directly gives the :

$$\left\|\mathcal{J}^i([\alpha_i^*, \boldsymbol{\alpha}_{(-i)}^*]) - \mathcal{J}^{i,m\to m^*}([\alpha_i^m, \boldsymbol{\alpha}_{(-i)}^m])\right\|^2 \leq \mathbb{E}\left[\left\|\delta\mathbf{Y}_t^{m\to m^*}\right\|^2\right] \leq \boldsymbol{\epsilon}, \tag{A.45}$$

Since $\mathcal{J}^i(\boldsymbol{\alpha}^*) < \mathcal{J}^i(\boldsymbol{\alpha}^m)$, our next step is to show the inequality between cost functionals given the action α_i^m obtained from the fictitious play and the the other arbitrary action $\beta^i \neq \alpha_i^m$:

$$\mathcal{J}^{i,m\to m^*}([\alpha_i^m, \boldsymbol{\alpha}_{(-i)}^m]) \leq \mathcal{J}^{i,m\to m^*}([\beta_i, \alpha_i^m]), \quad \beta^i \in \mathbb{A}^i. \tag{A.46}$$

Then, by triangle inequality, we obtain inequality that shows the ϵ-Nash equilibrium:

$$\mathcal{J}^i([\alpha_i^*, \boldsymbol{\alpha}_{(-i)}^*]) \leq \mathcal{J}^i([\beta_i, \boldsymbol{\alpha}_{(-i)}^*]) + \epsilon_i, \quad \forall 1 \leq i \leq I. \tag{A.47}$$

The final step of the proof is to show the stochastic optimality according to the defined marginal constants $\boldsymbol{\epsilon}$. Let $\theta_i^m(k) : \mathbb{N}^+ \to \Theta$ be the trajectory of the neural parameters of the i-th neural agents at learning iteration k. Recall the fact that $\alpha_i^m = \alpha_i^{m-1,K} := \mathcal{B}[\theta_i^m(K-1)]$. Let us denote that $\mathbf{Y}_t^m|_\theta$ is the adjoint variable given neural parameters θ. We define the closed metric balls $\{B_{\delta_i^m}^k\}_{k\in\mathbb{N}^+}$ centered at $\theta_i^m(k)$ with the radius $0 < \delta_i^m$ such that

$$B_{\delta_i^m}^k := \{\vartheta \in \Theta; \|\vartheta - \theta_i^m(k)\| \leq \delta_r^k, \theta_i^m(k) \text{ is a local minimum of } \mathbb{E}\mathbf{Y}_t^m|_{\boldsymbol{\theta}}\}. \tag{A.48}$$

Next, we consider the sub-sequence $\{\theta_i^m(\bar{k})\}_{\bar{k}\in\bar{N}} \subseteq \{\theta_i^m(k)\}_{k\in\mathbb{N}}$, which defines the strictly-decreasing sequence $\{\mathbb{E}\mathbf{Y}_t^m|_{\boldsymbol{\theta}^m(k)}\}_{\bar{k}\in\bar{N}}$ an ordered index set $\bar{N}$. Then, the admissible set $\mathbb{A}^i$ is defined as follows:

$$\mathbb{A}^i := \left\{ \bigcup_{(\bar{k},1)}^{(\bar{K},m^*)} \alpha_i^m(\cdot,\cdot,B_{\delta_i^m}^{\bar{k}}); \bar{K} := \max\{\bar{N}\} \right\} \subset \mathcal{L}^2(T\times\mathbb{R}^d), \quad 1 \leq m \leq m^*, \tag{A.49}$$

where $\bar{K}$ is a maximal element in $\bar{N}$. Intuitively saying, the admissible set $\mathbb{A}^i$ is a collection of local metric balls centered at neural parameters of i-th agent updated by gradient descent.

To define the optimal actions of neural agents, let us consider an arbitrary local convex set $\boldsymbol{\beta} \in \mathcal{C} \subset \mathbb{A}$ where $\mathbb{A}$ is the admissible action set for multi-agents defined as follows:

$$\mathbb{A} := \bigotimes_{i=1}^{I} \mathbb{A}^i \subset [\mathcal{L}^2(T\times\mathbb{R}^d)]^{\otimes I}. \tag{A.50}$$

By the convexity, there exist I pairs $\{(\omega_i, \beta_i)\}_{i\in\{1,\dots,I\}}$ such that the following equality holds:

$$\underbrace{(\mathcal{B}\circ\cdots\circ\mathcal{B}[\theta^m(k=0)])}_{K \text{ times}} = \alpha_i^{m+1} + \omega_i\beta_i \tag{A.51}$$

and $\beta_i(\cdot) \neq \mathbf{0}$. Then, the Gâteaux derivative (Carmona, 2016, Theorem 4.12) of adjoint variable $\mathbf{Y}_t^m$ is derived as follows:

$$\begin{aligned} 0 \leq \frac{d}{d\omega} dY_t^{i,\boldsymbol{\alpha}}(\alpha_i^{m+1} + \omega_i\beta_i) &= \mathbb{E}\left[\int_{\mathbb{T}} \nabla_{\alpha_i} H^i(t, \mathbf{X}_t^{\boldsymbol{\alpha}}, F_t^{i,\boldsymbol{\alpha}}, [\alpha_i^{m+1}, \boldsymbol{\alpha}_{(-i)}^m]) \cdot \beta_i dt\right] \\ &\leq \mathbb{E}\left[\int_{\mathbb{T}} \left\| \nabla_{\alpha_i} H^i(t, \mathbf{X}_t^{\boldsymbol{\alpha}}, F_t^{i,\boldsymbol{\alpha}}, [\alpha_i, \boldsymbol{\alpha}_{(-i)}^m])|_{\alpha=\alpha_i^{m+1}+\omega_i\beta_i} \right\|^2 \cdot \|\beta_i\|^2 dt\right]. \end{aligned} \tag{A.52}$$

The first inequality is trivial due to the definition of admissible set $\mathbb{A}^i$. Since $\|\beta\|^2$ is nonzero, one can deduce that $\nabla_{\boldsymbol{\alpha}} \boldsymbol{H}|_{\boldsymbol{\alpha}^m+\boldsymbol{\omega\beta}} \geq 0$. In other words, the action profiles $\boldsymbol{\alpha}^m$ are optimal actions and any arbitrary actions $\beta^i \in \mathbb{C}^i$ are eventually non-optimal. Hence, we can conclude from the above result that the inequality in (A.46) holds in the admissible action set. Finally, we conclude this proof by showing the stochastic optimality is preserved during the fictitious play by the definition of $\mathbb{A}$ that induces the following relation:

$$\boldsymbol{\mathcal{V}}_t + \boldsymbol{H}(t,\cdot,F_t,(\boldsymbol{\alpha}^{m+1}, \boldsymbol{\alpha}_{(\cdot)}^m)) + \frac{1}{2}\mathbf{Tr}(\Sigma \boldsymbol{Hess}\boldsymbol{\mathcal{V}}_t) = 0, \tag{A.53}$$

where $\Sigma = \sigma^T\sigma$. The stochastic optimality is obtained by the Hamiltonian equation (A.53) for every stages .

A.2.2 PROOF OF COROLLARY 1

In the proof of the Proposition 1, we have shown the existence of the local action set that assures the convergence of the fictitious play. In this proof, we show the convergence of predictor $\Lambda_t^{\boldsymbol{\alpha}}$ according to the action set assuring the local Nash equilibrium. We start by calculating the norm deviation for the Itô's differential of the predictor $(i.e., \delta\boldsymbol{\Lambda}_t^{\boldsymbol{\alpha}})$.

$$\begin{aligned}
\mathbb{E}\left[d\left\|\delta\boldsymbol{\Lambda}_t^{\boldsymbol{\alpha}}\right\|^2\right] &= \mathbb{E}\left[2\delta\boldsymbol{\Lambda}_t^{\boldsymbol{\alpha}}\cdot d\left[\delta\boldsymbol{\Lambda}_t^{\boldsymbol{\alpha}}\right] + \left[d\delta\boldsymbol{\Lambda}_t^{\boldsymbol{\alpha}}\right]^T d\delta\boldsymbol{\Lambda}_t^{\boldsymbol{\alpha}}\right] \\
&\leq \sum_j 2\mathbb{E}\Bigg[\left(\delta\mathbf{A}_j\mathbf{X}_t^j\right)\cdot\left(\delta\mathbf{A}_j\mathbf{b}_j + \delta\mathbf{X}_t^j\partial_t\mathbf{A}_j + \delta\mathbf{X}_t^j[\nabla\mathbf{A}_j]^T\mathbf{b}_j \right. \\
&\qquad \left. + \frac{1}{2}\mathbf{X}_t^j\mathbf{Tr}[\delta\Sigma_j\nabla^2\mathbf{A}_j]\right) + \delta(\nabla\mathbf{A}_j)^T\Sigma_j dt\Bigg] \\
&= \sum_j 2\mathbb{E}\Bigg[\underbrace{\delta(\mathbf{A}_j^2\mathbf{b}_j^T\mathbf{X}_t^j)}_{(a)} + ||\delta\mathbf{X}_t^j||^2\Big(\underbrace{\delta(\mathbf{A}_j\partial_t\mathbf{A}_j)}_{(b)} + \underbrace{\delta(\mathbf{A}_j\nabla\mathbf{A}_j\mathbf{b}_j^T)}_{(c)} \\
&\qquad + \underbrace{\frac{1}{2}\mathbf{Tr}[\Sigma_j\nabla^2\delta\mathbf{A}_j]}_{(d)}\Big) + \underbrace{(\nabla\delta\mathbf{A}_j)^T\Sigma_j}_{(e)}\, dt\Bigg].
\end{aligned} \tag{A.54}$$

We followed by the rules for denoting the object deviation in the previous proof to denote object deviations combined with the linear operators $(\partial_t, \nabla, \nabla^2, \mathbf{Tr})$ on $C_2(\mathbb{R}^d)$. From the last equality of (A.54), we estimate the upper bounds of each term.

$$(a)\ :\ \mathbb{E}\left[\left|\delta\mathbf{A}_j^2\mathbf{b}_j^T\mathbf{X}_t^j\right| dt\right] \leq \mathbb{E}\left[||\hat{A}||^2||\hat{b}||^2\left\|\delta\alpha_j^m\right\|^2\left\|\delta\mathbf{X}_t^j\right\|^2\right] \leq \mathbb{T}\tilde{L}^2M_2^3\kappa_m^2. \tag{A.55}$$

The result directly follows from By the assumption in (A.35), the following evaluation is trivial:

$$\mathbb{E}\left[\left\|\delta\mathbf{X}_t^j\right\|^2\right] = \mathbb{E}\left[\left\|\int_{\mathbb{T}}\delta\boldsymbol{b}_s^j ds\right\|^2\right] \leq \mathbb{E}\left[\int_{\mathbb{T}}\left\|\delta\boldsymbol{b}_s^j\right\|^2 ds\right] \leq \mathbb{T}M_2\tilde{L}\kappa_m. \tag{A.56}$$

Now, we estimate the two terms $(b), (c)$ that contain spatial and temporal derivatives of decision attention function in following bracket:

$$\begin{cases}
(b)\ :\ \mathbb{E}\left[\left|\delta\mathbf{A}_j\partial_t\mathbf{A}_j\right| dt\right] \leq \mathbb{E}\left[\left|\hat{A}_j\cdot\partial_t\hat{A}_j\delta\boldsymbol{\alpha}_j\right| + \left|\hat{A}_j\cdot\delta\partial_t\boldsymbol{\alpha}_j\right| dt\right] \leq 2\tilde{L}\kappa_m M_2 dt, \\
(c)\ :\ \mathbb{E}\left[\left|\delta\mathbf{A}_j\nabla\mathbf{A}_j\mathbf{b}_j^T\right| dt\right] = \mathbb{E}\left[\left\|\hat{A}\right\|^2\left|\alpha_j^m\cdot\nabla\alpha_j^m\right| dt\right] \leq M_2\mathbb{E}\left[\left\|\alpha_j^m\right\|^2\left\|\nabla\alpha_j^m\right\|_F^2 dt\right] \\
\qquad\qquad \leq M_2\tilde{L}^2\kappa_m^2 dt,
\end{cases} \tag{A.57}$$

where the basic property of vector gradient $\nabla(v\cdot u) = (\nabla v)\cdot u + v\cdot(\nabla u)$ for all $v, u \in \mathbb{R}^d, \nabla v, \nabla u \in \mathbb{R}^{d\times d}$ is used in the estimation of (c). For the last two terms, *i.e.*, $(d), (e)$, we used the fact that the deviation of the matrix Σ is not defined because our MaSDEs assume uncontrollable volatility.

$$\begin{cases}
(d)\ :\ \frac{1}{2}\mathbb{E}\left[|\mathbf{Tr}[\Sigma_j\nabla^2\delta\mathbf{A}_j]|dt\right] = \frac{1}{2}\mathbb{E}\left[\left|\mathbf{Tr}[\Sigma_j\hat{A}_j(\nabla^2\alpha_j^m - \nabla^2\alpha_j^{m^*})]\right|\right] \\
\qquad\qquad \leq \left\|\hat{A}_j\right\|\frac{1}{2}\mathbb{E}\left[\|\Sigma_j\|_F\left\|\nabla^2\delta\mathbf{A}_j\right\|_F dt\right] \\
\qquad\qquad \leq \frac{M_2\kappa_m}{2}\mathbb{E}\left[\|\Sigma_j\|_F\right] dt, \\
(e)\ :\ \mathbb{E}\left[\left|\delta(\nabla\mathbf{A}_j)^T\Sigma_j\right| dt\right] \leq M_2\kappa_m\mathbb{E}[\|\Sigma_j\|_F]dt.
\end{cases} \tag{A.58}$$

Note that the first line in (A.58) can be obtained by the trace inequality (*i.e.*, $\mathbf{Tr}[AB] \leq \|A\|_F \|B\|_F$). By collecting the evaluated terms above, we conclude the proof by showing the following relation:

$$\mathbb{E}\left[\int_{\mathbb{T}} \|\delta \mathbf{\Lambda}_s^{\boldsymbol{\alpha}}\|^2 \, ds\right] \propto O\left(\left\|\boldsymbol{\theta}^m - \boldsymbol{\theta}^{m^*}\right\|^3 I\mathbb{T}^2 \tilde{L}^3 M_2^3 \sup_{t\in\mathbb{T}} \mathbb{E}\left[\|\mathbf{\Sigma}\|_F\right]\right). \tag{A.59}$$

A.3 Non-cooperative Game

Empirical Study for the Non-cooperation. This section is devoted to answer the following important question: *Why do we consider a cooperative scenario rather than a non-cooperative one?* This question is crucial as the shared goal could have been achieved in a highly competitive and non-cooperative environment (Ye et al., 2018). For an intuitive explanation, we introduce a specific scenario where the neural agent becomes adversarial and determines that interfering other agent's objectives is the best possible strategy to achieve its own goal. Followed by the definition, the behavioral rules for the adversarial agent can be summarized by the following equations:

$$\textit{(Adversarial Agent)} \quad \begin{cases} \mathcal{V}^j(t,x) = \mathcal{J}^j(t,x,[\alpha_j^*, \boldsymbol{\alpha}_{(-j)}]), \\ \mathcal{J}^i(t,x,[\alpha_i^*, \boldsymbol{\alpha}_{(-i)}]) \leq \mathcal{J}^i(t,x,[\alpha_i^*, \alpha_j^*, \boldsymbol{\alpha}_{(-i,-j)}]). \end{cases} \tag{A.60}$$

The second inequality shows that the adversarial agent (here, the j-th agent) can easily ruin other agent's goal although the victim (here, the i-th agent) produces its best response α_i^* to the adversarial environment. Eventually, one can expect that the competitive group will fail to accomplish the original goal (*i.e.*, the accurate prediction). To formalize the aforementioned scenario by estimating the total amount of inefficiency derived from the non-cooperation, we formulate the non-cooperative game (*e.g.*, min-max) by converting the cooperative behavior into an adversarial one:

$$\hat{h}^j = h^j + \left\|A_j^{\alpha_j} - 1\right\|^2, \quad \min_{\alpha_j} \left\|A_j^{\alpha_j} - 1\right\|^2 = \max_{\alpha_j} A_j^{\alpha_j}, \quad j \neq i \in \{1, \ldots, I\}. \tag{A.61}$$

With the adversarial cost $\hat{h}^j$, selfish agents maximize the influence on temporal aggregation by interfering with opponents. As the symmetric relation in (7) fails to be satisfied, the proposed framework with cost function in (A.61) is regarded as a non-cooperative game.

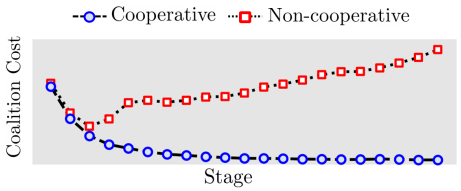


Figure A.1: Coalition cost for two scenarios.

To elucidate the importance of cooperation in our framework of time series prediction, we conduct an experiment on Mackey-Glass dataset by setting a single adversarial agent α_1. The red and blue learning curves in Figure A.1 illustrate the coalition costs during the fictitious play over 20 stages produced by competitive and cooperative groups, respectively. As one might expect, the cooperative group shows the stable learning dynamics until it converges to the agreement on the accurate prediction. This shows that the proposed fictitious play developed in Section 3 enjoys the desirable characteristic for the accurate time series prediction. Contrarily, the non-cooperative group under the competition fails to achieve a shared goal and shows that the adversarial behavior is fatal to the proposed framework regardless of the number of adversaries. In the optimization perspective, the primal reason for the failure is the instability induced by the naive approach for the gradient-descent based min-max type optimization. As a remedy, one may restrict the admissible action sets by considering an additional constraint on neural agents such as PL condition (Nouiehed et al., 2019) to develop a sophisticated algorithm for convergent result. Yet, it is an open question whether this theoretical perspective can lead to the accurate time series prediction.

A.4 Summary of Relevant Concepts

The most relevant strand of research about the multi-agent system in the machine learning community is multi-agent reinforcement learning (MARL) (Foerster et al., 2016). The relevant concepts between MARL and CooPredict are summarized in Table A.1. However, MARL and CooPredict differ in that MARL usually assumes model-free in discrete time, while CooPredict assumes model-based in continuous time.

Observations		$\{y_{s_i}\}_{s_i\in\mathbb{O}}$
Targets		$\{y_t\}_{t\in\mathbb{T}}$
Agents	Agents	$\boldsymbol{\alpha}=[\alpha_1,\ldots,\alpha_I],\quad \alpha_i\coloneqq\alpha_i(\cdot,\cdot;\theta_i)$
Actions	Actions	$\alpha_i(t,X_t^{i,\boldsymbol{\alpha}};\theta_i)$
States	Decisions	$\mathbf{X}_t^{\boldsymbol{\alpha}}=[X_t^{1,\boldsymbol{\alpha}},\ldots,X_t^{I,\boldsymbol{\alpha}}]\coloneqq\textbf{MaSDEs}(\{y_{s_i}\}_{s_i\in\mathbb{O}},\boldsymbol{\alpha})$
Rewards	Negative Errors	$-\lvert\lvert\Lambda_t-y_t\rvert\rvert^2$
Objective functional		$\mathcal{J}^i(t,\mathbf{X}_t^{\boldsymbol{\alpha}},\boldsymbol{\alpha})$
Value function		$\mathcal{V}^i(t,x)=\min_{\alpha_i^*\in\mathbb{A}^i}\mathcal{J}^i(t,x,[\alpha_i^*,\boldsymbol{\alpha}_{(-i)}^*])$
Nash equilibrium		$\mathcal{J}^i(t,x,[\alpha_i^*,\boldsymbol{\alpha}_{(-i)}^*])\le\mathcal{J}^i(t,x,[\beta_i,\boldsymbol{\alpha}_{(-i)}^*])+\epsilon_i$
Continuous Bellman	HJB	$\boldsymbol{\mathcal{V}}_t+\boldsymbol{H}(t,\cdot,F_t,(\boldsymbol{\alpha}^{m+1},\boldsymbol{\alpha}_{(\cdot)}^m))+\frac{1}{2}\mathbf{Tr}(\Sigma\mathbf{Hess}\boldsymbol{\mathcal{V}}_t)=0$

Table A.1: Comparison of relevant concepts. The first column stands for RL terminology; the second column stands for our terminology. In our works, agents and actions are neural networks and infinitesimal output given spatio-temporal variables $(t,X_t^{i,\boldsymbol{\alpha}})$. Decisions are continuous stochastic trajectories, and errors are L_2 loss between aggregated decisions and targets.

A.5 Pseudo-Code of CooPredict

We provide pseudo code for the training procedure of CooPredict via the deep neural fictitious play described in Section 3 in Algorithm 1 and the inference procedure of CooPredict in Algorithm 2.

Algorithm 1 Pseudo-code for Deep Neural Fictitious Play

Input: Time-series data $\boldsymbol{y}=(\{y_{s_i}\}_{s_i\in\mathbb{O}},\{y_t\}_{t\in\mathbb{T}})$, Neural Parameters $\boldsymbol{\theta}^{m=0}=[\theta_1^0,\ldots,\theta_I^0]$

Publicly share the given environment $\boldsymbol{\alpha}^{m=0}=[\alpha_1^0(\cdot,\cdot,\theta_1^0),\ldots,\alpha_I^0(\cdot,\cdot,\theta_I^0)]$
for $m=1$ to M(*i.e.*, The total number of stages) **do**
  Simulate set of decisions with action profiles $\boldsymbol{\alpha}^{m-1}$ at previous stage with equation (1).
  for all $i\in[1,\cdots,I]$ **in parallel do**
    Fix the environment of the previous stage except agent i.

$$\boldsymbol{\theta}^{m-1}=[\theta_i^{m-1},\bar{\boldsymbol{\theta}}_{(i-1)}^{m-1}],\quad \boldsymbol{\alpha}^{m-1}=[\alpha_i^{m-1},\bar{\boldsymbol{\alpha}}_{(i-1)}^{m-1}]$$

    Solve the decoupled adjoint problem for agent i in equation (13).

$$\mathcal{V}_t^{i,m}+\inf_{\alpha_i^{m-1}\in\mathbb{A}}H^i(t,x,F_t^{i,\boldsymbol{\alpha}^{m-1}},[\alpha_i^{m-1},\bar{\boldsymbol{\alpha}}_{(i-1)}^{m-1}])+\frac{1}{2}\mathbf{Tr}(\Sigma^T\Sigma\boldsymbol{Hess}\mathcal{V}^{i,m})=0$$

$$\theta_i^m=\mathbb{E}\left[\theta_i^{m-1}-\gamma_m\int_{\mathbb{T}}\nabla_{\theta_i}dY_t^{i,\boldsymbol{\alpha}}([\theta_i^{m-1},\bar{\boldsymbol{\theta}}_{(-i)}^{m-1}])dt\right]$$

    Collect the updated neural parameters $\boldsymbol{\theta}^m=[\theta_1^m,\ldots,\theta_I^m]$
  end for
  Update the total environment for the next stage $\boldsymbol{\alpha}^m\leftarrow\boldsymbol{\alpha}^{m-1}(\cdot,\cdot,\boldsymbol{\theta}^m)$.
end for
Return Optimal action profiles $\boldsymbol{\alpha}^*=[\alpha_1^*,\ldots,\alpha_I^*]$.

Algorithm 2 Pseudo-code for Inference

Input: Held-out time-series data $\hat{\boldsymbol{y}}=(\{\hat{y}_{s_i}\}_{s_i\in\mathbb{O}},\{\hat{y}_t\}_{t\in\mathbb{T}})$, Optimal action profiles $\boldsymbol{\alpha}^*=[\alpha_1^*,\ldots,\alpha_I^*]$

Simulate optimal decisions over the future interval $\mathbb{T}$ given observations $\{\hat{y}_{s_i}\}_{s_i\in\mathbb{O}}$.

$$[X_t^{1,\boldsymbol{\alpha}^*},\ldots,X_t^{I,\boldsymbol{\alpha}^*}]_{t\in\mathbb{T}}=\textbf{MaSDEs}(\{\hat{y}_{s_i}\}_{s_i\in\mathbb{O}},\boldsymbol{\alpha}^*)$$

Aggregate individual decisions with decision attention $\boldsymbol{A}^{\boldsymbol{\alpha}^*}=[A_1^{\alpha_1^*}(t,X_t^{1,\boldsymbol{\alpha}^*}),\ldots,A_I^{\alpha_I^*}(t,X_t^{I,\boldsymbol{\alpha}^*})]_{t\in\mathbb{T}}$

$$[\boldsymbol{\Lambda}_t^{\boldsymbol{\alpha}^*}]_{t\in\mathbb{T}}=[\boldsymbol{A}^{\alpha^*}(t,\boldsymbol{X}_t^{\boldsymbol{\alpha}^*})^\top\boldsymbol{X}_t^{\boldsymbol{\alpha}^*}]_{t\in\mathbb{T}}$$

Return Prediction on future $[\boldsymbol{\Lambda}_t^{\boldsymbol{\alpha}^*}]_{t\in\mathbb{T}}$ over the interval $\mathbb{T}$.

A.6 Implementation Details

Evaluation. In all experiments with real-world datasets, we train each model for 500 epochs using the Adam optimizer with a learning rate of 10^{-3} and batch size of 128. We reported the MSE and (Gaussian) NLL as suggested in (Rubanova et al., 2019). The performance is evaluated on the predicted parts of the test dataset. For the evaluation of our method, we evaluate MSEs between the predictor $\Lambda_t^{\boldsymbol{\alpha}^*}$ and a *test* data $\hat{y}_t \sim \hat{\nu}_t$ as follows:

$$\textbf{(MSEs)}: \quad \mathbb{E}_{s\sim\mathbb{O},t\sim\mathbb{T},\Lambda_t^{\boldsymbol{\alpha}^*}\sim\mathbb{Q}}\left[\left\|\hat{y}_t - \Lambda_t^{\boldsymbol{\alpha}^*}|_{\hat{y}_{s_1},\ldots,\hat{y}_{s_i},\ldots,\hat{y}_{s_I}}\right\|^2\right], \tag{A.62}$$

where the predictor is conditioned by the past observations $\{\hat{y}_{s_i}\}_{s_i\in\mathbb{O}}$ of the testing time series. As neural agents collaborate to minimize the temporally averaged utilities (*i.e.*, $\mathcal{J}^i \approx \int_{\mathbb{T}} \mathbb{E}||\Lambda_s - y_s||^2 ds$) via cooperative differential game, the goal of the proposed cooperative game becomes identical to forecast the time series $\hat{y}_{t\in\mathbb{T}}$ given observations.

A.6.1 Network Architecture

In this section, we briefly introduce the network architecture of multi-agent neural SDEs as shown in Figure A.2. Given the initial condition $\{y_{(\cdot)}\}_{s_i\in\mathbb{O}}$, each neural agent takes the spatio-temporal variable $(t, X_t^{i,\boldsymbol{\alpha}})$ and produces infinitesimally transformed outputs to propagate its own stochastic trajectory. First, the temporal variable t is embedded into inhomogeneous and non-linear representation as $t' = (t, \sin(t), \cos(t))$. Note that we adopt the temporal privacy function suggested in (Park et al., 2022) for further temporal encoding. After the non-linear time embedding, the intermediate representations $(t', X_t^{i,\boldsymbol{\alpha}})$ are fed into the novel module referred to as *agent identification layer* (AIL), which is defined as follows:

$$\textbf{AIL}(X_t^{i,\boldsymbol{\alpha}}, i) := \zeta_i \left(\frac{X_t^{i,\boldsymbol{\alpha}} - \mathbb{E}(X_t^{i,\boldsymbol{\alpha}})}{\textbf{Std}(X_t^{i,\boldsymbol{\alpha}})}\right) + \zeta_i, \quad \zeta_i = (i+1)/I. \tag{A.63}$$

The AIL is motivated by the adaptive instance normalization (Huang & Belongie, 2017) that adaptively transforms the statistics of intermediate feature representations according to user-guided information (*i.e.*, index for neural agent). This encourages each neural agent to propagate the individual stochastic trajectory and ensures the separated representations. The outputs from AIL are fed into two subsequent linear layers with LipSwish (Chen et al., 2019) and produce values for drift b^i and decision attention functions $A_i^{\alpha_i}$. In the experiments with real-world dataset, we identically set the dimension of hidden layers of each neural agent and the drift network as 128 following the baseline model (Park et al., 2022). We set the number of hidden layers as 2 for neural agents and 1 for the drift network. For the attention network, we used a relatively small dimension (*i.e.*, , 36) with 2 hidden layers across all the experiments. For the simulation of the stochastic trajectory given the architecture, we discretize sampled times by applying the Euler-Maruyama scheme.

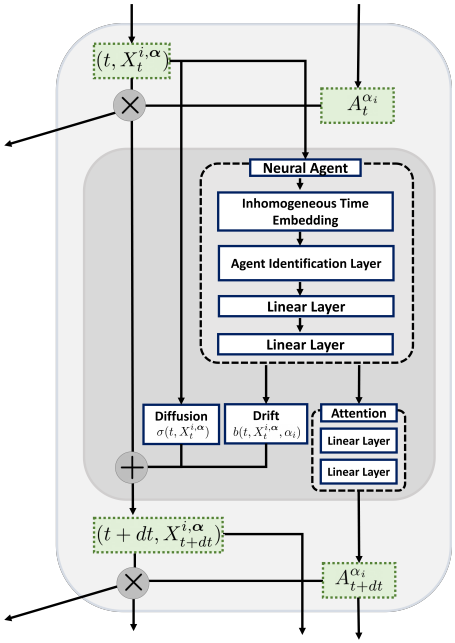


Figure A.2: **The architecture of MaSDEs**.

A.6.2 Dataset

PhysioNet Challenge 2012, (Silva et al., 2012), contains 8000 multivariate clinical time series obtained from the intensive care unit (ICU). Each time series has various clinical features of the patient's first 48 hours after admission to ICU. We processed the dataset to hourly time series and eliminated static features so that each time series has a length of 48 with 35 time varying features (*e.g.*, Albumin, Heart-rate, etc.,). We used half of the time series as the training dataset and the remaining parts as the test dataset.

Speech Commands, (Warden, 2018), consists of one-second audio records of various spoken words such as "Yes", "No", "Up", and "Down". Since there are more than 100,000 record samples, we sub-sampled the dataset on two conflicting classes (*i.e.*, "Right" and "Left"). As a result, 6950 time series records were selected. We pre-processed these time series by computing Mel-frequency cepstrum coefficients so that each time series has a length of 54 with 65 channels. We used 80% of selected data as training dataset and the remaining parts as the test dataset.

Beijing Air-Quality, (Zhang et al., 2017; Dua & Graff, 2017), consists of multi-year air quality recordings across different locations in Beijing. Each sample contains 6-dimensional time series features of $PM_{2.5}$, PM_{10}, SO_2, NO_2, CO, and O_3, which are recorded per hour. We segmented data per two days so that each sample has a length of 48. We combine recordings of 12 different locations into a single dataset and randomly split 80% of data into the training dataset and 20% of data into the test dataset.

Gaussian-Impulse Noise. To generate the Gaussian-impulse noise, we first sample the random time-stamp $\{t_1, t_2, \cdots\}$ from a homogeneous possion process with intensity level of 8.0. With this time-stamp, we generate the data Y_t over the time interval $[0, 48]$ as follows:

$$Y_t = \begin{cases} 0.6 + U_t, & \text{if} \quad \{t\} \sim \texttt{Pois}(\lambda) \\ Z_t, & \text{otherwise.} \end{cases} \tag{A.64}$$

where $U_t \sim \text{Unif}[0, 1]$, $Z_t \sim \mathcal{N}(0, \sigma^2)$ and the standard deviation of Gaussian noise is set to 0.1. Similar to other datasets, we set the observable and the prediction intervals to contain the first 80% and the last 20% of total points, respectively.

Delayed Differential Equation. In this experiment, we simulated deterministic trajectories of *delayed differential equations* (DDEs) as following:

$$\textbf{Mackey-Glass DDE:} \quad \frac{dy_t}{dt} = \frac{\beta y_{(t-\tau)}}{1 + y^n_{(t-\tau)}} - \gamma y_t, \tag{A.65}$$

where we set hyper-parameters as $\tau = 10, n = 10, \beta = 0.25, \gamma = 0.1$. Among the entire set of trajectories, we randomly split 80% of data into training dataset, and remaining 10% and 10% into validation and test dataset, respectively. We follow the identical experiment setting as suggested in (Holt et al., 2022) where the entire interval is split into two halves, *i.e.*, $\mathbb{O} = [0, 10]$ and $\mathbb{T} = [10, 20]$.

A.7 Comparisons with Non-DE Based Model

The most significant advantage of using DE-based methods over non-DE based methods is the capability of handling irregularly sampled or partially observed time-series data which is common in real-world settings such as medicine or business. In such settings, non-DE-based counterparts are incapable of applying without careful consideration of pre-processing steps such as discretization and interpolation. Therefore, we compared CooPredict with the state-of-the-art Transformer-based time-series forecasting methods on the Speech dataset. We chose the Speech dataset since it comprises complete time-series observations with regular time intervals which can be favorable for applying non-DE-based methods and can avoid potential bias due to design choices of pre-processing steps. As can be seen in Table A.2, CooPredict outperforms its non-DE-based counterparts implying that our proposed method can be a promising tool for general time-series modeling under both regular and irregular settings.

Method	Test MSE ↓ ($\times 10^{-2}$)
Transformer (Vaswani et al., 2017)	0.4476 ± 0.069
Informer (Zhou et al., 2021)	0.4258 ± 0.002
Autoformer (Wu et al., 2021)	0.4854 ± 0.022
CooPredict	$\mathbf{0.4136 \pm 0.017}$

Table A.2: Test MSE (mean ± std) over five runs on the Speech datasets.

A.8 Additional Qualitative Results

In this subsection, we illustrate the results of an experiment in Section 5.2 on Mackey-Glass DDE and provide additional qualitative results on BAQD across overall features of different instances to show the effectiveness of the proposed method on time series prediction.

Figure A.3 shows the ground truth trajectories and inferred predictions/attentions results for various sampled trajectories. Notably, the future trajectory is highly sensitive to initial states, so capturing the delayed effect of the distant past on the future is required for successful prediction in this dataset. As can be seen in Figure A.3, our method can adaptively capture the delayed effect and provide accurate predictions regardless of different histories. Compared to our method, the CSDE-TP provides worsened performance. In light of this, we can interpret that adjusting the relative impact of given observations is more effective for generating accurate future predictions than treating them equally.

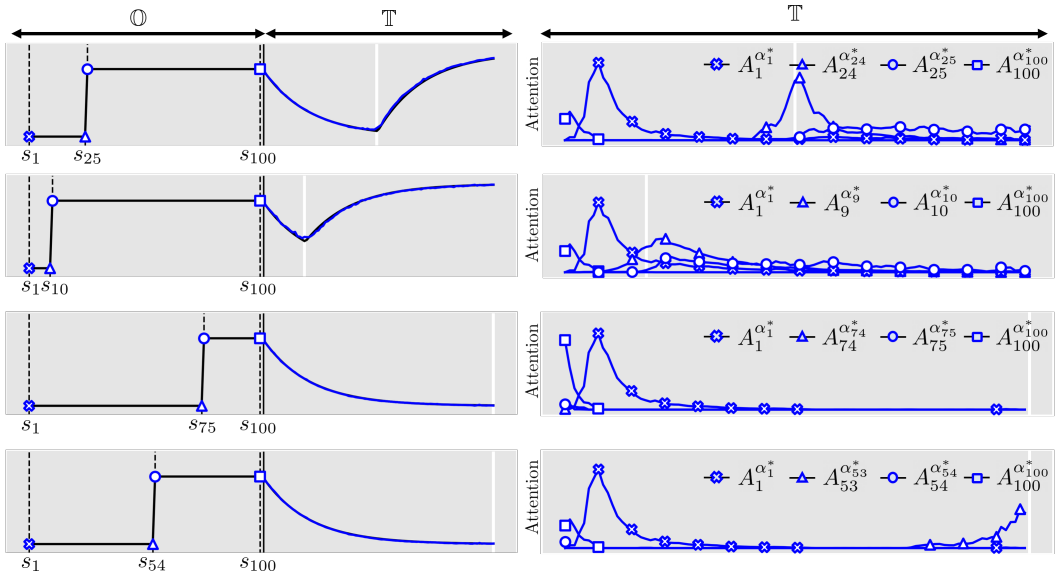


Figure A.3: Qualitative results for randomly selected trajectories on Mackey-Glass DDE. (Left) Black and blue lines correspond to the ground truth trajectory $\hat{y}_t$ and the prediction $\Lambda_t^{\alpha^*}$, respectively. (Right) The decision attention $\boldsymbol{A}_t^{\alpha^*}$ on the prediction time interval $\mathbb{T}$.

Figure A.4 shows prediction results for all features and learned attentions on the BAQD. We can observe that our method exhibits attentions similar (but not strictly restricted) to temporal decay assumption. As can be seen in Figure A.4, the learned decision attention varies depending on the given observation. We claim that neural agents have a variety of cooperation patterns to generate accurate prediction. That is, our method flexibly extracts complex and heterogeneous temporal correlations beyond simple temporal decay assumed by the conventional methods.

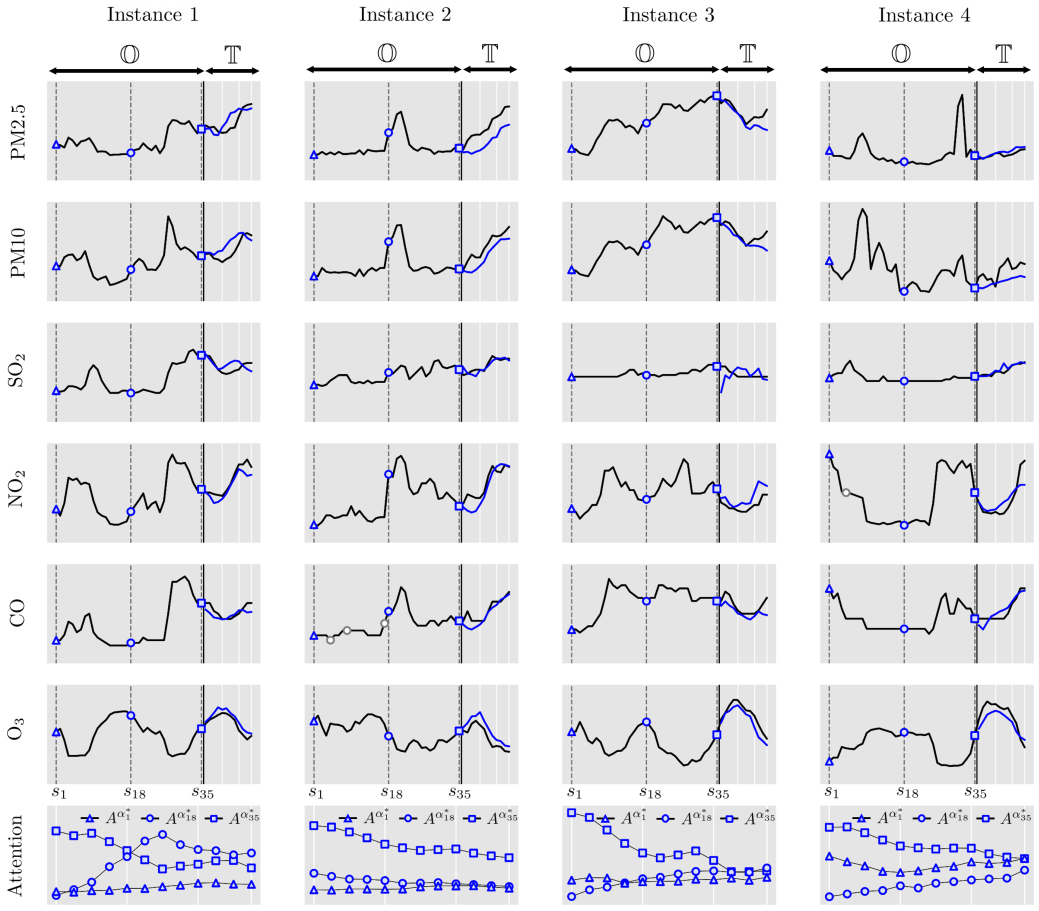


Figure A.4: Additional qualitative results on BAQD dataset. Each row displays the prediction results of corresponding features. The last row displays the learned decision attention function on the prediction interval $\mathbb{T}$.

APPENDIX REFERENCES

Dominique Bakry, Ivan Gentil, and Michel Ledoux. *Analysis and geometry of Markov diffusion operators*, volume 348. Springer Science & Business Media, 2013.

R. Carmona and F. Delarue. *Probabilistic Theory of Mean Field Games with Applications I: Mean Field FBSDEs, Control, and Games*. Probability Theory and Stochastic Modelling. Springer International Publishing, 2018. ISBN 9783319589206.

René Carmona, Christy V Graves, and Zongjun Tan. Price of anarchy for mean field games. *ESAIM: Proceedings and Surveys*, 65:349–383, 2019.

Ricky TQ Chen, Jens Behrmann, David K Duvenaud, and Joern-Henrik Jacobsen. Residual flows for invertible generative modeling. *NeurIPS*, 2019.

Dheeru Dua and Casey Graff. UCI machine learning repository, 2017. URL http://archive.ics.uci.edu/ml.

Jakob Foerster, Ioannis Alexandros Assael, Nando De Freitas, and Shimon Whiteson. Learning to communicate with deep multi-agent reinforcement learning. *Advances in neural information processing systems*, 2016.

Samuel I Holt, Zhaozhi Qian, and Mihaela van der Schaar. Neural laplace: Learning diverse classes of differential equations in the laplace domain. In *International Conference on Machine Learning*, pp. 8811–8832. PMLR, 2022.

Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *ICCV*, 2017.

Jean Jacod and Albert Shiryaev. *Limit theorems for stochastic processes*, volume 288. Springer Science & Business Media, 2013.

Nikita Kitaev, Lukasz Kaiser, and Anselm Levskaya. Reformer: The efficient transformer. In *International Conference on Learning Representations*, 2020.

Maher Nouiehed, Maziar Sanjabi, Tianjian Huang, Jason D Lee, and Meisam Razaviyayn. Solving a class of non-convex min-max games using iterative first order methods. In *Advances in Neural Information Processing Systems*, 2019.

Sung Woo Park, Kyungjae Lee, and Junseok Kwon. Neural markov controlled SDE: Stochastic optimization for continuous-time data. In *International Conference on Learning Representations*, 2022.

Tim Roughgarden. Intrinsic robustness of the price of anarchy. *Journal of the ACM (JACM)*, 62(5): 1–42, 2015.

Yulia Rubanova, Ricky TQ Chen, and David K Duvenaud. Latent ordinary differential equations for irregularly-sampled time series. In *NeurIPS*, 2019.

Ikaro Silva, George Moody, Daniel Scott, Leo Celi, and Roger Mark. Predicting in-hospital mortality of icu patients: The physionet/computing in cardiology challenge 2012. *Computing in cardiology*, 2012.

Daniel W Stroock. *An introduction to Markov processes*, volume 230. Springer Science & Business Media, 2013.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2017.

Pete Warden. Speech commands: A dataset for limited-vocabulary speech recognition. *arXiv preprint arXiv:1804.03209*, 2018.

Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with Auto-Correlation for long-term series forecasting. In *Advances in Neural Information Processing Systems*, 2021.

Maojiao Ye, Guoqiang Hu, and Frank L Lewis. Nash equilibrium seeking for n-coalition noncooperative games. *Automatica*, 95:266–272, 2018.

Shuyi Zhang, Bin Guo, Anlan Dong, Jing He, Ziping Xu, and Song Chen. Cautionary tales on air-quality improvement in beijing. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Science*, 473:20170457, 09 2017.

Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *The Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Virtual Conference*, 2021.