A GENERATIVE MODEL FOR GAME THEORY WITH FLOW EQUILIBRIUM

Anonymous authors
Paper under double-blind review

ABSTRACT

In recent years, generative models have emerged as a groundbreaking development in the field of artificial intelligence, transforming various domains such as image synthesis, natural language processing, and data generation. While recent studies have integrated generative models into multi-agent scenarios, their game-theoretical implications have remained largely unexplored. Specifically, the relationship between solutions derived from generative models and game theoretical equilibrium concepts lacks rigorous investigation. This paper aims to bridge the gap between generative models and game theory by introducing a novel probabilistic framework for modelling multi-agent decision-making problems. This innovative framework reinterprets these problems as generative processes. Furthermore, we introduce a training objective known as "flow equilibrium" and establish a theoretical connection between flow equilibrium and Nash equilibrium. To analyse the theoretical properties of our framework, we present a tabular version algorithm along with a convergence proof. Additionally, we propose an extended algorithm incorporating neural networks to handle more complex environments. Notably, our framework naturally incorporates opponent modelling. Harnessing the capabilities of generative models, our framework excels in capturing the dynamics of strategic interactions among agents. We validate our approach through testing on various multi-agent tasks, including cooperative and general-sum games. The empirical results consistently support our theoretical findings, demonstrating that our framework consistently outperforms existing methods in terms of solution quality.

1 INTRODUCTION

In recent years, the field of artificial intelligence has witnessed remarkable advancements, primarily driven by the emergence of generative models. These models have brought about transformative changes across various domains, ranging from the generation of highly realistic images to the enhancement of natural language understanding (OpenAI, 2023) and the generation of diverse data (Ramesh et al., 2022). Their exceptional capacity to capture complex data distributions has made them a cornerstone of contemporary AI research. Recent studies have been inspired to harness the potent capabilities of generative models for addressing decision-making problems (Janner et al., 2022; Ajay et al., 2023; Lu et al., 2023; Liang et al., 2023). These methods solve the decision making problem by casting it as a generative process. They achieve this by recasting decision-making as a generative process, where the policy is represented by a generative model. These efforts have underscored the effectiveness of generative modelling in tackling sequential decision-making challenges, as evidenced by empirical results. While some researchers have extended the application of generative models to multi-agent decision-making problems (Zhu et al., 2023; Li et al., 2023), their focus has predominantly been on offline settings. Moreover, the game-theoretical analysis of policies derived from generative models has largely remained unexplored. Control as Inference (CAI) (Levine, 2018), rooted in probabilistic inference and the maximisation of likelihood, faces inherent challenges when applied directly to multi-agent decision-making. In such scenarios, agents often pursue multi-dimensional objectives that may not align with each other. Self-interest can lead to conflicts within the objective functions, complicating the optimisation of these probabilistic models.

This paper endeavours to bridge the theoretical gap between generative modelling and game theory by proposing a novel generative framework. Within this framework, we introduce a specialised prob-

abilistic process to model agent interactions, effectively transforming multi-agent decision-making problems into generative processes amenable to online interaction with generative models. We introduce a training objective named "flow equilibrium" for the generative model and establish a theoretical connection between flow equilibrium and Nash equilibrium. Building upon this framework, we present a tabular version algorithm along with a convergence proof. Furthermore, we propose a parameterised algorithm that incorporates neural networks, extending its applicability to complex environments. Additionally, we seamlessly integrate opponent modelling into our framework. Leveraging the capabilities of generative models, our framework excels in capturing the dynamics of strategic interactions among agents, with an analysis of error bounds for opponent modelling. We evaluate the performance of our framework in differential games and non-atomic routing games against strong baseline methods, demonstrating its superior overall performance.

2 Related works

Generative Model A generative model is a type of statistical model that is designed to generate or produce new data samples. Generative models learn the underlying structure or patterns in the training data and then generate new data points by sampling from the learned distribution. Autoregressive models (Larochelle & Murray, 2011; Germain et al., 2015), normalising flow models (Dinh et al., 2015), and variational auto-encoders (VAEs) (Kingma & Welling, 2014; Rezende et al., 2014) directly learn the distribution's probability function via maximum likelihood while generative adversarial networks (GANs) (Goodfellow et al., 2014) represent the probability distribution implicitly by a model of its sampling process. Recent Large Language Models (LLMs) such as GPT-4 (OpenAI, 2023) have demonstrated their remarkable language prowess. In the field of image synthesis, diffusion models have displayed their outperforming abilities (Song et al., 2021). In this paper, we propose a new generative model for solving game theory, extending the application of generative model.

Opponent Modelling In multi-agent reinforcement learning (MARL), learning a robust policy against the uncertainty caused by the unknown opponent policy is crucial. To mitigate this uncertainty, opponent modelling aims to model the opponent's behaviours, goals, or beliefs, thereby reducing the uncertainty. One line of the work is to predict the opponent behaviour using imitation learning (Grover et al., 2018). ToMnet leverages Theory of Mind to infer the agent's actions and goals from past and current observations (Rabinowitz et al., 2018). SOM explicitly model opponent using an agent's own policy to predict an opponent's action based on the opponent's state. The agent then use gradient descent to optimise its belief about the opponent's goal. PR2 (Wen et al., 2019b) and GR2 (Wen et al., 2019a) employ the recursive reasoning based on the joint Q function which requires extra information. We solve the opponent model leveraging the powerful generative model.

Control as Inference Variational inference (VI) is a powerful tool to learn and inference probabilistic models (Jordan et al., 1999; Zhang et al., 2018). VI works by approximating the target distribution through the minimisation of a divergence objective. Casting a control problem into a probability inference problem enables the application of advanced inference tools to the control, and extends the model of control. Applying probabilistic inference to control has a long history (Toussaint, 2009a;b; Rawlik et al., 2010; 2013; Toussaint & Storkey, 2006), (Dvijotham & Todorov, 2012). Casting a control problem into a probability inference problem enables the application of advanced inference tools to the control, and extends the model of control (Levine, 2018; Kappen et al., 2009). However, most of the existing works focus on the single-agent case. There are a few works that try to extend the inference framework to the multi-agent setting, due to interest conflicts among agents. And most of them focus on cooperative games (Tian et al., 2019; Wen et al., 2020), which limits the application of the framework.

3 Preliminaries

3.1 Markov Decision Process

We consider a Markov decision process (MDP) with N agents. An MDP is characterised by a tuple $\mathcal{M} = (\mathcal{S}, \mathcal{N}, \{\mathcal{A}_i, r_i\}_{i\in\mathcal{N}}, P, \mu_0, \gamma)$. $\mathcal{S}$ is a finite state space. $\mathcal{N} = \{1, 2, \ldots, N\}$ is the set of agents. $\mathcal{A}_i$ is the action space of agent i. $\mathcal{A} = \times_i \mathcal{A}_i$ is the space of joint action space. $r_i : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the

reward function of agent i. $P : \mathcal{S} \times \mathcal{A} \to \mathcal{P}(\mathcal{S})$ is transition kernel for the state dynamic. μ_0 is the initial distribution of initial state s_0. $\gamma \in (0, 1)$ is the discount factor for future rewards. The MDPs consider only one self-interested players, which limits the its flexibility in modelling the uncertainty in the external environment.

3.2 STOCHASTIC GAME

The stochastic game (SG) extends MDPs to accommodate scenarios involving multiple self-interested players. We consider a SG (Shapley, 1953; Shoham & Leyton-Brown, 2008) with N players. The horizon of SG is $\mathcal{T} = \{0, 1, \ldots, T\}$. At each time index $t \in \mathcal{T}$, agent $i \in \mathcal{N}$ ($\mathcal{N} = \{1, 2, \ldots, N\}$) at state $s_t \in \mathcal{S}$ will select an action a_t^i from the action space $\mathcal{A}^i$. All the agents take action simultaneously. Let $\boldsymbol{a}_t = (a_t^1, a_t^2, \ldots, a_t^N) \in \mathcal{A}$ denote the joint action. Each agent i will receive a reward $r^i(s_t, \boldsymbol{a}_t)$ and the joint state will change to s_{t+1} according to the transition kernel $P(s_{t+1}|s_t, \boldsymbol{a}_t)$. Agents i take actions according to policy $\pi^i : \mathcal{S} \to \Delta(\mathcal{A}^i)$. Given the joint policy $\boldsymbol{\pi} = (\pi_1, \pi_2, \ldots, \pi_N)$, the cumulative reward of agent i is

$$V^i(s; \boldsymbol{\pi}) = \sum_{t=0}^{\infty} \mathbb{E}\left[\gamma^t r_t^i(s_t, \boldsymbol{a}_t) | s_0 = s, \boldsymbol{\pi}\right], \tag{1}$$

where the expectation is taken with respect to $s_{t+1} \sim P(\cdot|s_t, \boldsymbol{a}_t)$, $\boldsymbol{a}_t \sim \boldsymbol{\pi}(\cdot|\boldsymbol{s}_t)$. The Nash equilibrium is a joint policy $\boldsymbol{\pi}^* = (\pi^{1,*}, \pi^{2,*}, \ldots, \pi^{N,*})$ such that for all agent i, $V^i(s; \boldsymbol{\pi}^*) \geq V^i(s; \pi^i, \boldsymbol{\pi}^*_{-i})$, where $\boldsymbol{\pi}^{-i,*} = (\pi^{1,*}, \ldots, \pi^{i-1,*}, \pi^{i+1,*}, \ldots, \pi^{N,*})$, i.e. $\pi^{i,*}$ is the best response of $\boldsymbol{\pi}^{-i,*}$. Accordingly, $\pi^{i,*} \in \Pi^i := \Delta(\mathcal{A}_i)$, $\boldsymbol{\pi}^* \in \Pi := \Delta(\times_{i \in \mathcal{N}} \mathcal{A}_i)$ and $\boldsymbol{\pi}^{-i,*} \in \Pi := \Delta(\times_{i \neq j \in \mathcal{N}} \mathcal{A}_j)$. Similarly, a joint policy $\boldsymbol{\pi}^*$ is the ϵ-Nash equilibrium if there exists an $\epsilon > 0$ so that for all agent $i \in \mathcal{N}$, $V^i(s; \boldsymbol{\pi}^*) \geq \max_{\pi^i \in \Pi^i} V^i(s; \pi^i, \boldsymbol{\pi}^{-i,*}) - \epsilon$.

4 GRAPHICAL MODEL FOR GAME (GMG)

In this section, we first establish the an abstract graphical model for game from the probabilistic perspective and propose an equilibrium concept named flow equilibrium. Then we connect it with the game theory. We focus on a directed acyclic graph (DAG), represented by $\mathcal{G} = (\mathcal{X}, \mathcal{E})$, where $\mathcal{X}$ denotes a finite set of vertices, and $\mathcal{E} \subset \mathcal{X} \times \mathcal{X}$ refers to a set of directed edges.

We define a parent-child relationship between two vertices x_i and x_{i+1} when the directed edge $x_i \to x_{i+1}$ represents an action. Specifically, x_i is the parent vertex of x_{i+1}, and x_{i+1} is the child vertex of x_i. Furthermore, we define the *initial vertex* x_1 as the unique state with no incoming edges, and we refer to vertices that have no outgoing edges as *terminating vertices*.

Given the current vertex x_t, agent $i \in \mathcal{N} = \{1, 2, \cdots, N\}$ will sample action a_t^i from the policy π_t^i. The joint action and the joint policy are denote as $\boldsymbol{a}_t = \{a_t^1, a_t^2, \cdots, a_t^N\}$ and $\boldsymbol{\pi}_t = \{\pi_t^1, \pi_t^2, \cdots, \pi_t^N\}$, respectively. Then the next vertex x_{t+1} will be sampled from a fixed transition probability function $P(x_{t+1}|x_t, \boldsymbol{a}_t)$. A trajectory can be obtained by sampling states from policy $\boldsymbol{\pi}$ and transition probability P successively. The probability to generate the trajectory τ is denote as $P(\tau; \boldsymbol{\pi})$. The marginal probability of sampling trajectories that ended at x_T is given by $P_T(x_T; \boldsymbol{\pi}) = \sum_{\tau \to x_T} P(\tau; \boldsymbol{\pi})$, where $\tau \to x_T$ is defined as the set of trajectories that reach the terminating vertex x_T. When the terminating vertex x_T is sampled, agent $i \in \mathcal{N}$ will receive a non-negative return function $R^i(x_T; \boldsymbol{\pi})$.

4.1 FLOW EQUILIBRIUM

We denote $\boldsymbol{\pi}^{-i}$ as the joint policy of all agents except i. Given the $\boldsymbol{\pi}^{-i}$, agent i aims to update π^i such that $P_T(x_T; \pi^i, \boldsymbol{\pi}^{-i}) \propto R^i(x_T; \boldsymbol{\pi})$. The goal to optimise the policies is to reaching an equilibrium named flow equilibrium (FE), which is defined as follows.

Definition 4.1. The flow equilibrium is a profile $\boldsymbol{\pi}^\star$ that satisfies the condition $P_T(x_T; \pi^{i,\star}, \boldsymbol{\pi}^{-i,\star}) \propto R^i(x_T; \boldsymbol{\pi})$ for all $i \in \mathcal{N}$ and any policy π^i, where $\boldsymbol{\pi}^{-i,\star}$ denotes the policy profile of all agents except i.

We prove that the FE exists as shown in the Theorem 4.2.

Theorem 4.2. *Given a non-negative function $R(x) = \{R^i(x_T; \boldsymbol{\pi})\}_{i\in\mathcal{N}}$ are continuous with respect to $\boldsymbol{\pi}$ and transition probability P, there exists an FE.*

4.2 Solving Markov Game

In this section, we will introduce how to use GMG to solve Markov game.

In the GMG, the return function $R(x)$ depends solely on the current vertex x, while in a Markov game, the objective is to maximise the long-term return as a function of a trajectory. Consequently, it should be able to determine the long-term return of a trajectory in a GMG using the terminal vertex x_T. A nature way to solve the problem is to choose the full trajectory in the Markov game as the vertex in GMG, but it will make the space complexity grows exponentially.

By utilising state augmentation, we establish a relationship between the vertex in GMG and a trajectory in the Markov game. For a given trajectory $\tau = (s_0, \boldsymbol{a}_0, s_1, \boldsymbol{a}_1, \cdots, s_T, \boldsymbol{a}_T)$, the accumulated reward up to time step t is denoted as $z_t(\tau) = \frac{\sum_{k=0}^{t-1}\gamma^k r^i(s_k,\boldsymbol{a}_k)}{\gamma^{t-1}}$. As this equation is true for each agent i, we omit the i for brevity. We select the vertex in the GMG as $x_t = (s_t, z_t, p_t)$. The transition functions can be expressed as follows:

$$
\begin{aligned}
s_{t+1} &\sim P(\cdot|s_t, \boldsymbol{a}_t)\\
z_{t+1} &= r^i(s_t, \boldsymbol{a}_t) + \frac{z_t}{\gamma}\\
p_{t+1} &= p_t \boldsymbol{\pi}^{-i}(\boldsymbol{a}_t^{-i}|s_t) P(s_{t+1}|s_t, \boldsymbol{a}_t).
\end{aligned}
$$

This vertex choice enable us to compute the long-term return from the terminating vertex, which reduces space complexity in comparison to choosing the entire trajectory as the vertex. The return function of GMG is non-negative, so we choose $R^i(x_T) = \exp(p_T \sum_{t=0}^{\infty}\gamma^t r^i(s_t, a_t)) = \exp(\text{Ret}^i(\tau))$.

The next question to apply GMG to solving Markov game is the connection between FE and NE.

Theorem 4.3. *If $\boldsymbol{\pi}$ is an FE, it is an ϵ-NE with $\epsilon = 2|\mathcal{X}|Ret_{max}e^{-\delta}$, where $|\mathcal{X}| = \max_i |\mathcal{X}^i|$, $Ret_{max} = \max_{\tau,i} Ret^i(\tau)$, and $\delta = \min_i Ret_{max} - \max_{Ret^i(\tau)<Ret_{max}} Ret^i(\tau)$.*

The proof is deferred to Appendix A.2.

4.3 Training Criterion

In this section, we employ variational inference to solve Markov game under the framework of GMG. To achieve flow equilibrium, we want to minimise the KL divergence $\text{KL}(P_T(x_T; \pi^i, \boldsymbol{\pi}^{-i})\|R^i(x_T)/Z)$. From the convexity, we can optimise the upper bound of $\text{KL}(P_T(x_T; \pi^i, \boldsymbol{\pi}^{-i})\|R^i(x_T)/Z)$.

$$
\begin{aligned}
&\text{KL}(P_T(x_T; \pi^i, \boldsymbol{\pi}^{-i})\|\mathbb{E}_{\pi^i,\boldsymbol{\pi}^{-i}}[R^i(x_T)]/Z) \le \text{KL}(P(\tau;\boldsymbol{\pi})\|R^i(x_T)/Z)\\
&=\mathbb{E}_{\boldsymbol{\pi},P}\left[\log\frac{\prod_{t=1}^{\infty}\pi^i(a_t|s_t,\boldsymbol{a}^{-i})\rho(\boldsymbol{a}^{-i}|s_t)}{\prod_{t=1}^{\infty}\boldsymbol{\pi}^{-i}(\boldsymbol{a}_t^{-i}|s_t)} - \sum_{t=0}^{\infty}\gamma^t r^i(s_t,\boldsymbol{a}_t) + \log Z\right]\\
&=\mathbb{E}_{\boldsymbol{\pi},P}\left[-\sum_{t=0}^{\infty}\gamma^t r^i(s_t,\boldsymbol{a}_t) - \sum_{t=0}^{\infty}\gamma^t H(\pi^i(a_t^i|s_t,\boldsymbol{a}_t^{-i}))\right]\\
&\quad+\mathbb{E}_{\boldsymbol{\pi},P}\left[\sum_{t=0}^{\infty}\gamma^t \text{KL}(\rho(\boldsymbol{a}^{-i}|s_t)\|\boldsymbol{\pi}^{-i}(\boldsymbol{a}_t^{-i}|s_t)) + \log Z\right],
\end{aligned}
\tag{2}
$$

where $\pi^i(a_t^i|s_t, \boldsymbol{a}_t^{-i})$ is the policy of the agent i and $\rho(\boldsymbol{a}_t^{-i}|s_t)$ is the opponent model of agent i. It is worth emphasising that the minimisation of Equation 2 necessitates that updates to both the policy and the opponent model can only be performed upon the completion of the entire trajectory. To enhance sample efficiency, we define the action-value function and value function allowing to optimise the policy and opponent model through partial trajectories.

Definition 4.4. Given a joint policy $\boldsymbol{\pi}$, the action-value function is defined as follows.

$$Q^i(s_t, a_t^i, a_t^{-i}; \boldsymbol{\pi}) = r_t^i(s_t, \boldsymbol{a}_t) + \log \hat{\boldsymbol{\pi}}^{-i}(\boldsymbol{a}_t^{-i}|s_t) + \mathbb{E}\Bigg[\sum_{k=t+1}^{\infty} \gamma^{k-t}(r_k^i(s_k, \boldsymbol{a}_k) + H(\pi^i(a_k^i|s_k, \boldsymbol{a}_k^{-i})) - \mathrm{KL}(\rho(a_k^{-i}|s_k)||\hat{\boldsymbol{\pi}}^{-i}(\boldsymbol{a}_k^{-i}|s_k)))\Bigg], \tag{3}$$

where the expectation is taken with respect to $a_k^i \sim \pi^i(\cdot|s_k, \boldsymbol{a}_k^{-i})$, $\boldsymbol{a}_k^{-i} \sim \rho(\cdot|s_k)$, $s_{k+1} \sim P(\cdot|s_k, a_k^i, \boldsymbol{a}_k^{-i})$. And the value function is

$$V^i(s; \boldsymbol{\pi}) = \mathbb{E}[Q^i(s, a^i, \boldsymbol{a}^{-i}; \boldsymbol{\pi}) - \log \pi^i(a^i|s, \boldsymbol{a}^{-i})\rho(\boldsymbol{a}_t^{-i}|s_t)],$$

where the expectation is taken with respect to $a^i \sim \pi^i(\cdot|s, \boldsymbol{a}^{-i})$, $\boldsymbol{a}^{-i} \sim \rho(\cdot|s)$.

Leveraging the notation of action-value function and value function, we derive an equivalent form to minimise the upper bound in the Equation (2).

$$\begin{aligned}
&J^i(\pi^i, s_0; \boldsymbol{\pi}^{-i}) = \mathbb{E}_{s_0 \sim P(s_0)}\big[V^i(s_0; \boldsymbol{\pi})\big] \\
=&\mathbb{E}_{s_0 \sim P(s_0)}\Big[\mathbb{E}[Q^i(s_0, a^i, \boldsymbol{a}^{-i}; \boldsymbol{\pi}) - \log \pi^i(a^i|s_0, \boldsymbol{a}^{-i})] - \log \rho(\boldsymbol{a}^{-i}|s_0)\Big] \\
=&\mathbb{E}_{s_0 \sim P(s_0)}\left[\log Z + H(\rho(\cdot|s_0)) - \mathbb{E}\left[\mathrm{KL}\left(\pi^i(a^i|s_0, \boldsymbol{a}^{-i})\middle\| \frac{\exp(Q^i(s_0, a^i, \boldsymbol{a}^{-i}; \boldsymbol{\pi}))}{Z}\right)\right]\right],
\end{aligned}$$

where $Z = \sum_{a^i \in A_i} \exp(Q^i(s_0, a^i, \boldsymbol{a}^{-i}; \boldsymbol{\pi}))$. Since the KL divergence is non-negative, we have the following proposition.

Proposition 4.5. *The best response policy is in the form of*

$$\pi^{i,*}(a^i|s, \boldsymbol{a}^{-i}) = \frac{\exp(Q^i(s, a^i, \boldsymbol{a}^{-i}; \boldsymbol{\pi}))}{\sum_{a^i \in A_i} \exp(Q^i(s, a^i, \boldsymbol{a}^{-i}; \boldsymbol{\pi}))}. \tag{4}$$

Note that action-value function defined here differs from the Q function in the context of reinforcement learning. The expectation of action-value function is taken with respect to the opponent model while the expectation of Q function in the context of reinforcement learning is taken with respect to the opponent policy $\boldsymbol{\pi}^{-i}$. Therefore, the action value function is not the expected cumulative reward. The following proposition provides the upper bound for this difference.

Proposition 4.6. *Suppose that* $\mathrm{KL}(\rho(\cdot|s)||\boldsymbol{\pi}^{-i}(\cdot|s)) < \epsilon_\rho$ *for all* $s \in \mathcal{S}$. *Without loss of generality, the reward function* $|r^i(s, \boldsymbol{a})| \leq 1$, $\forall s \in \mathcal{S}$, $\boldsymbol{a} \in \mathcal{A}$, $i \in \mathcal{N}$. *Denote the action-value function derived using the opponent model as* $\hat{Q}^i(s, \boldsymbol{a}; \boldsymbol{\pi})$. *Then we have that*

$$\max_{s \in \mathcal{S}, \boldsymbol{a} \in \mathcal{A}, i \in \mathcal{N}} |Q^i(s, \boldsymbol{a}; \boldsymbol{\pi}) - \hat{Q}^i(s, \boldsymbol{a}; \boldsymbol{\pi})| \leq \delta, \tag{5}$$

where $\delta := \frac{2(1+\log|\mathcal{A}_i|)}{(1-\gamma)^2}\sqrt{\frac{1}{2}\epsilon_\rho} + \frac{\epsilon_\rho}{1-\gamma}$.

The proof is deferred to Appendix A.4. Note that the definition of the action-value function requires that we approximate the policy of opponents $\boldsymbol{\pi}^{-i}(\boldsymbol{a}_t^{-i}|s_t)$ with agent i's opponent model $\rho(\boldsymbol{a}_t^{-i}|s_t)$. Here we don't specify the method to update $\rho(\boldsymbol{a}_t^{-i}|s_t)$. The above conclusion applies to any opponent model method.

In order to capture dynamics among agents, we propose an opponent modelling method under our framework. Here we consider the case $|\mathcal{N}| = 2$ for the brevity of notation, but this method can be extended to the cases with more agents. To approximate the behaviour of agent $-i$, we factorise the auxiliary distribution over states and actions $q(\boldsymbol{a}_{0:\infty}, s_{0:\infty})$ in the following way.

$$\begin{aligned}
q(\boldsymbol{a}_{0:\infty}, s_{0:\infty}) &= P(s_0)\prod_{t=0}^{\infty} q(\boldsymbol{a}_t^{-i}|s_t)q(a_t^i|s_t, \boldsymbol{a}_t^{-i})P(s_{t+1}|s_t, \boldsymbol{a}_t^{-i}, a_t^i) \\
&= P(s_0)\prod_{t=0}^{\infty} \rho(\boldsymbol{a}_t^{-i}|s_t)\pi^i(da_t^{-i}|s_t, \boldsymbol{a}_t^{-i})P(s_{t+1}|s_t, \boldsymbol{a}_t^{-i}, a_t^i)
\end{aligned}$$

We denote the solution to this problem as the joint policy $\boldsymbol{\pi}^*$. Denote $Q_\rho^{-i}(s_t, \boldsymbol{a}_t; \rho)$ as the soft action-value function of agent $-i$.

$$\begin{aligned} Q_\rho^{-i}(s_t, \boldsymbol{a}_t; \rho) =& r^{-i}(s_t, \boldsymbol{a}_t) - \mathrm{KL}(\hat{\boldsymbol{\pi}}^{-i}(\cdot|s_t) \| \rho(\cdot|s_t)) \\ &+ \mathbb{E}[\sum_{h=t+1}^{\infty} \gamma^{h-t}(r_h^{-i}(s_h, \boldsymbol{a}_h) - \mathrm{KL}(\hat{\boldsymbol{\pi}}^{-i}(\cdot|s_h) \| \rho(\cdot|s_h)))], \end{aligned}$$

where the expectation is taken with respect to $\boldsymbol{a}_h \sim q(\cdot|s_h)$, $s_h \sim P(\cdot|s_{h-1}, \boldsymbol{a}_{h-1})$. $\hat{\boldsymbol{\pi}}^{-i}$ is the empirical distribution of opponent policy. Then we can derive the optimal opponent model for agent $-i$.

Proposition 4.7. *The optimal opponent model for agent i is*

$$\rho(\boldsymbol{a}^{-i}|s) = \frac{\hat{\boldsymbol{\pi}}^{-i}(\boldsymbol{a}^{-i}|s) \exp(\mathbb{E}_{a^i \sim \pi^i}[Q_\rho^{-i}(s, \boldsymbol{a}; \rho)])}{\mathbb{E}_{\boldsymbol{a}^{-i} \sim \hat{\boldsymbol{\pi}}^{-i}(\cdot|s)}\left[\exp(\mathbb{E}_{a^i \sim \pi^i}[Q_\rho^{-i}(s, \boldsymbol{a}; \rho)])\right]} \tag{6}$$

where $\hat{\boldsymbol{\pi}}^{-i}(\boldsymbol{a}^{-i}|s)$ is the prior of opponent policy $\boldsymbol{\pi}^{-i}(\boldsymbol{a}^{-i}|s)$.

The proof is deferred to Appendix A.8. Proposition 4.7 provides a closed-form of opponent modelling. Note that the result resembles a logit quantal response equilibrium (LQRE) policy when the prior of the opponent's policy is a uniform distribution. This finding suggests that our opponent modelling framework is well-suited for handling uncertainty from the external environment.

5 General Variational Bayesian Opponent Modelling

In this section, we aim to propose an algorithm to solve the Markov game based on our framework and training criterion. Then we prove that this algorithm converges on the Markov Potential Game (MPG). Further, we propose GPI, an actor critic algorithm powered by neural networks to solve complex and continuous problem.

5.1 Variational Policy Gradient

Proposition 4.5 shows that the best response policy is in the form of $\pi^{i,\theta}(a|s, \boldsymbol{a}^{-i}) = \mathrm{softmax}(\theta_{i,s,a,\boldsymbol{a}^{-i}})$, i.e. the best response policy is softmax policy parameterised (Agarwal et al., 2021). We use the natural policy gradient (NPG) method (Kakade, 2001) to derive the best response policy.

Proposition 5.1. *Denote $\theta^{(t)}$ the t-th iterate and $\pi^{(t)} = \mathrm{softmax}(\theta_{s,\boldsymbol{a}})$. For each agent i, state s, and action a, the NPG update rule can be written as*

$$\pi^{i,(t+1)}(a \mid s, \boldsymbol{a}^{-i}) = \frac{1}{Z^{(t)}(s)} \left(\pi^{i,(t)}(a \mid s, \boldsymbol{a}^{-i})\right)^{1-\frac{\eta}{1-\gamma}} \exp\left(\frac{\eta Q^{i,(t)}(s, a, \boldsymbol{a}^{-i}; \boldsymbol{\pi}^{(t)})}{1-\gamma}\right). \tag{7}$$

where η is the learning rate.

The proof is deferred to Appendix A.3. Then we propose the variational policy gradient (VPG) algorithm. The pseudo-code of VPG is listed in the Algorithm 1.

Then we will prove that VPG converges to Nash equilibrium in the Markov potential game.

Definition 5.2. MPG is a Markov decision process that there exists a function $\Phi(s; \pi^i, \boldsymbol{\pi}^{-i}) : \Pi \to \mathbb{R}$, with $s \in \mathcal{S}$, so that

$$\tilde{V}^i\left(s; \pi^i, \boldsymbol{\pi}^{-i}\right) - \tilde{V}^i\left(s; \pi^{i,\prime}, \boldsymbol{\pi}^{-i}\right) = \Phi\left(s; \pi^i, \boldsymbol{\pi}^{-i}\right) - \Phi\left(s; \pi^{i,\prime}, \boldsymbol{\pi}^{-i}\right),$$

for all agents $i \in \mathcal{N}$, states $s \in \mathcal{S}$ and policies $\pi^i, \pi^{i,\prime} \in \Pi^i, \boldsymbol{\pi}^{-i} \in \Pi_{-i}$. Here $\tilde{V}^i\left(s; \pi^i, \boldsymbol{\pi}^{-i}\right)$ is value function with accurate opponent modelling.

The first step is to prove that the estimation error of the opponent is bounded. The proposition 4.6 has provided the upper bound of the estimation error of the opponent.

Algorithm 1 Variational Policy Gradient (VPG)

input Learning rate η
  Initialise opponent model ρ.
  Initialise policy $\pi^{i,(0)}$ for all agent $i \in \mathcal{N}$.
  Initialise the replay buffer M.
  for $k = 1, 2, \ldots$ **do**
    for Each agent $i \in \mathcal{N}$ **do**
      For the current state s_t, $a_t^i \sim \pi^i(\cdot|s_t) = \sum_{\boldsymbol{a}_t^{-i}} \rho(\boldsymbol{a}_t^{-i}|s_t)\pi^i(\cdot|s_t, a_t^i, \boldsymbol{a}_t^{-i})$.
      Observe next state s_{t+1}, opponent action a_t^{-i} and reward r_t^i and save the experience in the reply buffer.
      Update opponent model.
    end for
    for Each agent $i \in \mathcal{N}$ **do**
      Compute the best response policy using Equation (7).
    end for
  end for

The second step is to derive the convergence of VPG with exact opponent modelling. We first show the equivalence between VPG and the global NPG on the potential function. Then we will prove the convergence of VPG using the smoothness of the potential function.

Note that the gradient of the value functions equals the potential function and agents update their policy independently. Hence VPG is equivalent to running Natural Policy Gradient (NPG) on the potential function, which is shown in the following proposition.

Proposition 5.3. *Consider the global NPG dynamic on the potential function:* $\theta_s^{(t+1)} = \theta_s^{(t)} + \eta\mathcal{F}^\dagger(\theta_s^{(t)})\nabla_{\theta_s}\Phi \ \forall s \in \mathcal{S}$, *where* $\mathcal{F}^\dagger(\theta_s) = \mathbb{E}[\nabla_{\theta_s}\log\boldsymbol{\pi}^{\theta_s}(\boldsymbol{a}|s)\nabla_{\theta_s}\log\boldsymbol{\pi}^{\theta_s}(\boldsymbol{a}|s)^T]^\dagger$ *is the pseudo-inverse of the Fischer information matrix.* $\boldsymbol{\pi}^{\theta_s}(\boldsymbol{a}|s) = \prod_{i\in\mathcal{N}}\mathbb{E}_{a^{-i}\sim\rho(\cdot|s)}[\pi^i(a^i|s, a^{-i})]$. *VPG has the same dynamics as global NPG.*

The proof is deferred to Appendix A.5. After showing the connection of VPG and the NPG on the potential function, we next show the smoothness of the potential function in the following lemma.

Lemma 5.4. *The potential function* Φ *is L-smooth with the constant* $L = \frac{2(n+1)^2}{(1-\gamma)^3} + 2(n^2+n+1)\frac{1+\log\max_{i\in\mathcal{N}}|A_i|}{(1-\gamma)^2} + \frac{3n+2}{1-\gamma}$.

The proof is deferred to Appendix A.6. Using Lemma 5.4, the potential function $\Phi(s;\boldsymbol{\pi}^{(t)})$ is non-decreasing if the learning rate is $\frac{1}{L}$ (Bubeck et al., 2015). We finally give the convergence of VPG.

Theorem 5.5. *VPG converges to a fixed point, which is* ϵ*-Nash equilibrium of MPG, where* $\epsilon = \delta + \frac{\log|A|}{1-\gamma}$.

Theorem 5.5 ensures the applicability of VPG for solving MPG.

VPG does not involve a certain opponent modelling method. The next question is how to model the opponent using variational inference.

5.2 Opponent Modelling in General-Sum Game

Since the agent i does not know the reward of the agent $-i$, we have to find a function $\hat{r}^{-i}$ to estimate r^{-i}. The objective of optimising $\hat{r}^{-i}$ is to minimise the KL divergence between the optimal opponent model derived by estimated reward function $\hat{r}^{-i}$ and the history data of agent $-i$. Let $\tau_{-i} = \{s_0, \boldsymbol{a}_0^{-i}, a_0^i, s_1, \boldsymbol{a}_1^{-i}, a_1^i, \ldots\}$ be the historical interaction data. The probability of generating τ_{-i} by the opponent model is

$$\rho(\tau_{-i}) = P(s_0)\prod_{t=1}^{\infty} P(s_t|s_{t-1}, \boldsymbol{a}_{t-1}^{-i}, a_{t-1}^i)\rho(\boldsymbol{a}_{t-1}^{-i}|s_{t-1})\pi^i(a_{t-1}^i|s_{t-1}).$$

Then the objective to optimise $\hat{r}^{-i}$ is

$$\mathrm{KL}(P(\tau_{-i})||\rho(\tau_{-i})) = \mathbb{E}\left[\sum_{t=0}^{\infty} -\gamma^t \hat{r}^{-i}(s_t, \boldsymbol{a}_t^{-i}, a_t^i)\right] + \log \mathbb{E}_{\boldsymbol{a}^{-i}\sim\rho}\left[\mathbb{E}_{a^i\sim\rho_i}[\exp(Q_\rho^{-i}(s, \boldsymbol{a}; \rho))]\right], \quad (8)$$

where the first expectation is taken with respect to $s_t \sim P(s_t|s_{t-1}, \boldsymbol{a}_{t-1}^{-i}, a_{t-1}^i)$. It is difficult to calculate the optimal opponent model because $\mathbb{E}_{\boldsymbol{a}^{-i}\sim\rho}\left[\exp(Q_\rho^{-i}(s, \boldsymbol{a}; \rho))\right]$ is difficult to estimate. We use a sample-based method for estimating $\mathbb{E}_{\boldsymbol{a}^{-i}\sim\rho}\left[\exp(Q_\rho^{-i}(s, \boldsymbol{a}; \rho))\right]$.

$$\mathrm{KL}(P(\tau_{-i})||\rho(\tau_{-i})) = \mathbb{E}\left[\sum_{t=0}^{\infty} -\gamma^t \hat{r}^{-i}(s_t, \boldsymbol{a}_t^{-i}, a_t^i)\right] + \log \mathbb{E}_{\tau_{-i}\sim\rho(\tau_{-i})}\left[\frac{\exp(\sum_{t=0}^{\infty}\gamma^t \hat{r}^{-i}(s_t, \boldsymbol{a}_t))}{\rho(\tau_{-i})}\right] \quad (9)$$

If $\hat{r}_\psi^{-i}$ is parameterised by ψ, the gradient of $\mathrm{KL}(P(\tau_{-i})||\rho(\tau_{-i}))$ with respect to ψ is

$$\frac{\mathrm{dKL}(P(\tau_{-i})||\rho(\tau_{-i}))}{\mathrm{d}\psi} = \mathbb{E}\left[\sum_{t=0}^{\infty} -\gamma^t \frac{\mathrm{d}\hat{r}_\psi^{-i}(s_t, \boldsymbol{a}_t^{-i}, a_t^i)}{\mathrm{d}\psi}\right] + \frac{1}{Z}\mathbb{E}_{\tau_{-i}\sim\rho(\tau_{-i})}\left[w_{-i}\frac{\mathrm{d}\sum_{t=0}^{\infty}\gamma^t \hat{r}_\psi^{-i}(s_t, \boldsymbol{a}_t)}{\mathrm{d}\psi}\right], \quad (10)$$

where $w_{-i} = \frac{\exp(\sum_{t=0}^{\infty}\gamma^t \hat{r}_\psi^{-i}(s_t, \boldsymbol{a}_t))}{\rho(\tau_{-i})}$ and $Z = \mathbb{E}_{\tau_{-i}\sim\rho(\tau_{-i})}[w_{-i}]$.

VPG is for tabular cases and is impractical in problems with high dimensions or continuous action. To handle the problems, we propose the variational actor-critic method, which can be implemented in a complex continuous environment. We use neural-network to parameterise the policy π^θ, opponent model ρ^ϕ, the action-value function Q_ω, and the reward function r_ψ.

The objective to optimise the policy π^θ is to minimise the KL divergence

$$J_\pi(\theta; s) = \mathbb{E}_{a^{-i}\sim\rho(\cdot|s)}\left[\mathrm{KL}\left(\pi_i^\theta(\cdot|s)||\exp(Q_\omega^i(s, \cdot, \boldsymbol{a}^{-i}) - V^i(s))\right)\right]. \quad (11)$$

The objective to optimise the action-value function Q_ω is to minimise:

$$J_Q(\omega) = \mathbb{E}_{\left(s_t, a_t^i, \boldsymbol{a}_t^{-i}\right)\sim\mathcal{D}}\left[\frac{1}{2}\left(Q_\omega^i\left(s_t, a_t^i, \boldsymbol{a}_t^{-i}\right) - r^i\left(s_t, a_t^i, \boldsymbol{a}_t^{-i}\right) - \gamma\mathbb{E}_{s_{t+1}\sim p_s}\left[\bar{V}\left(s_{t+1}\right)\right]\right)^2\right], \quad (12)$$

with $\bar{V}^i\left(s_{t+1}\right) = Q_{\bar{\omega}}^i\left(s_{t+1}, a_{t+1}^i, \hat{\boldsymbol{a}}_{t+1}^{-i}\right) - \log\rho_\phi\left(\hat{\boldsymbol{a}}_{t+1}^{-i} \mid s_{t+1}\right) - \log\pi_\theta\left(a_{t+1}^i \mid s_{t+1}, \hat{\boldsymbol{a}}_{t+1}^{-i}\right) + \log\hat{\boldsymbol{\pi}}\left(\hat{\boldsymbol{a}}_{t+1}^{-i} \mid s_{t+1}\right)$, where $Q_{\bar{\omega}}^i$ is target action-value function.

The gradient of (11) with respect to θ is

$$\nabla_\theta J_\pi(\theta; s) = \mathbb{E}_{\boldsymbol{a}^{-i}\sim\rho(\cdot|s)}[\nabla_\theta \log\pi_i^\theta(a|s) + (\nabla_a\pi_i^\theta(a|a^i, s) - \nabla_a Q^i(s, a, \boldsymbol{a}^{-i}))\nabla_\theta f_\theta(\epsilon; s, \boldsymbol{a}^{-i})] \quad (13)$$

where a is evaluated at $f_\theta(\epsilon; s, \boldsymbol{a}^{-i})$. The gradient of (12) with respect to ω is

$$\nabla_\omega J_Q(\omega) = \nabla_\omega Q_\omega^i\left(s_t, a_t^i, \boldsymbol{a}_t^{-i}\right)\left(Q_\omega^i\left(s_t, a_t^i, \boldsymbol{a}_t^{-i}\right) - r^i\left(s_t, a_t^i, \boldsymbol{a}_t^{-i}\right) - \gamma\mathbb{E}_{s_{t+1}\sim p_s}\left[\bar{V}\left(s_{t+1}\right)\right]\right) \quad (14)$$

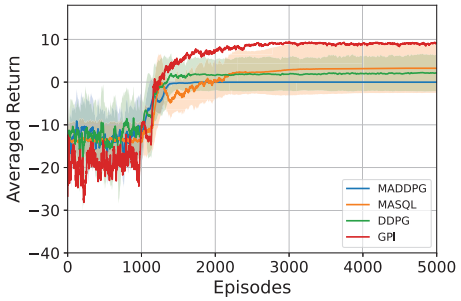
Then the pseudo-code of the variational inference actor-critic method named Generative Policy Inference (GPI) is listed in the Algorithm 2.

6 EXPERIMENTS

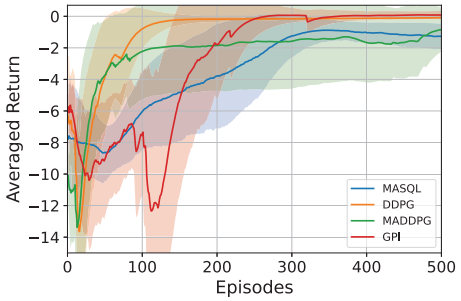
As GPI incorporates entropy regularisation naturally, it enjoys stronger exploration ability. We test its exploration ability on a challenging differential game. Differential game is adopted from (Wei et al., 2018). The two agents in this game have continuous action space. All the agents share the same reward function depending on the joint action (a_1, a_2) following the equations: $r^1\left(a^1, a^2\right) = r^2\left(a^1, a^2\right) = \max\left(f_1, f_2\right)$, where $f_1 = 0.8 \times \left[-\left(\frac{a^1+5}{3}\right)^2 - \left(\frac{a^2+5}{3}\right)^2\right], f_2 = 1.0 \times \left[-\left(\frac{a^1-5}{1}\right)^2 - \left(\frac{a^2-5}{1}\right)^2\right] + 10$. The training process includes 200 episodes with 25 steps per episode. We compare GPI Deep Deterministic Policy Gradient (DDPG) (Lillicrap et al., 2016), and two multi-agent reinforcement learning algorithm: Multi-Agent Deep Deterministic Policy Gradient

Algorithm 2 Generative Policy Inference (GPI)

Initialising replay buffer $\mathcal{D}$.
Initialising parameters θ, ω, ψ and ϕ.
for Each episode $d = 1, 2, \ldots$ **do**
  for $i \in \mathcal{N}$ **do**
    For current state s_t compute $\boldsymbol{a}_t^{-i} \sim \rho(\cdot|s_t)$, $a_t^i \sim \pi^i(\cdot|s_t, a_t^{-i})$
    Observe next state s_{t+1}, opponent action $\boldsymbol{a}_t^{-i}$ and save the new experience in the reply buffer $\mathcal{D}$.
    Update opponent model using Algorithm 3.
    Update π^i using Equation (13).
  end for
end for
Output: policy π^i, $i \in \mathcal{N}$, opponent model ρ



(a) The learning curves of GPI and other baselines in differential game.



(b) The learning curves of GPI and other baselines in non-atomic routing game.

Figure 1: Learning curves in differential game and non-atomic routing game.

(MADDPG) (Lowe et al., 2017), Multi-Agent Soft Q Learning (MASQL) (Wei et al., 2018) in this task. This is a challenging task for most continuous gradient-based RL algorithms because the gradient update often leads the training agent towards a suboptimal point. The reward surface has a local maximum of 0 at (-5, -5) and a global maximum of 10 at (5, 5), with a deep valley in between. If the agents' policies are initialized at (0, 0) (the red starred point), which is within the basin of the left local maximum, gradient-based methods are likely to struggle in reaching the global maximum equilibrium point because the valley blocks the upper right area. The learning curves are shown in Figure 1a. Only GPI shows the capability of converging to the global optimum, while other baselines can only reach the sub-optimal point.

To assess whether GPI can converge in a Markov potential game, we performed GPI on a task referred to as the non-atomic routing game. This game was borrowed from the work of Mguni et al. (Mguni et al., 2021). In this game, the agent and the virtual opponent are playing Markov potential game. Agents in this game are self-interested and learn how to split their commodity to maximise rewards. We compare GPI with MADDPG, MASQL and DDPG in this task. The learning curves are shown in Figure 1b. The learning curve of GPI is smoother and other algorithms suffer from high variance.

7 Conclusion

This paper bridges the gap between generative modelling and game theory in the field of artificial intelligence, recognising the transformative potential of generative models across various domains. By introducing a novel generative framework tailored to multi-agent decision-making scenarios and incorporating the concept of "flow equilibrium," we have addressed critical limitations and established theoretical connections to Nash equilibrium. Our proposed algorithms, including tabular and parameterised versions, combined with seamless opponent modelling, empower the field with versatile tools. Leveraging the expressive power of generative models, our framework excels in capturing dynamics among agents, as validated through empirical evaluations in differential and

non-atomic routing games where it consistently outperforms established baselines. This research not only fills a crucial void but also lays the groundwork for future advancements at the intersection of generative modelling and game theory.

REFERENCES

Alekh Agarwal, Sham M. Kakade, Jason D. Lee, and Gaurav Mahajan. On the theory of policy gradient methods: Optimality, approximation, and distribution shift. *J. Mach. Learn. Res.*, 22: 98:1–98:76, 2021. URL http://jmlr.org/papers/v22/19-736.html.

Anurag Ajay, Yilun Du, Abhi Gupta, Joshua Tenenbaum, Tommi Jaakkola, and Pulkit Agrawal. Is conditional generative modeling all you need for decision-making?, 2023.

Sébastien Bubeck et al. Convex optimization: Algorithms and complexity. *Foundations and Trends® in Machine Learning*, 8(3-4):231–357, 2015.

Shicong Cen, Chen Cheng, Yuxin Chen, Yuting Wei, and Yuejie Chi. Fast global convergence of natural policy gradient methods with entropy regularization. *Operations Research*, 70(4): 2563–2578, 2022.

Laurent Dinh, David Krueger, and Yoshua Bengio. NICE: non-linear independent components estimation. In Yoshua Bengio and Yann LeCun (eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Workshop Track Proceedings*, 2015. URL http://arxiv.org/abs/1410.8516.

Krishnamurthy Dvijotham and Emo Todorov. Linearly solvable markov games. In *American Control Conference, ACC 2012, Montreal, QC, Canada, June 27-29, 2012*, pp. 1845–1850. IEEE, 2012. doi: 10.1109/ACC.2012.6315632. URL https://doi.org/10.1109/ACC.2012.6315632.

Mathieu Germain, Karol Gregor, Iain Murray, and Hugo Larochelle. MADE: masked autoencoder for distribution estimation. *CoRR*, abs/1502.03509, 2015. URL http://arxiv.org/abs/1502.03509.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. Generative adversarial nets. In Zoubin Ghahramani, Max Welling, Corinna Cortes, Neil D. Lawrence, and Kilian Q. Weinberger (eds.), *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pp. 2672–2680, 2014. URL https://proceedings.neurips.cc/paper/2014/hash/5ca3e9b122f61f8f06494c97b1afccf3-Abstract.html.

Aditya Grover, Maruan Al-Shedivat, Jayesh K. Gupta, Yuri Burda, and Harrison Edwards. Learning policy representations in multiagent systems. In Jennifer G. Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1797–1806. PMLR, 2018. URL http://proceedings.mlr.press/v80/grover18a.html.

Xin Guo, Anran Hu, Renyuan Xu, and Junzi Zhang. Learning mean-field games. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 4967–4977, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/030e65da2b1c944090548d36b244b28d-Abstract.html.

Michael Janner, Yilun Du, Joshua B. Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis, 2022.

Michael I Jordan, Zoubin Ghahramani, Tommi S Jaakkola, and Lawrence K Saul. An introduction to variational methods for graphical models. *Machine learning*, 37:183–233, 1999.

Sham M. Kakade. A natural policy gradient. In Thomas G. Dietterich, Suzanna Becker, and Zoubin Ghahramani (eds.), *Advances in Neural Information Processing Systems 14 [Neural Information Processing Systems: Natural and Synthetic, NIPS 2001, December 3-8, 2001, Vancouver, British Columbia, Canada]*, pp. 1531–1538. MIT Press, 2001. URL https://proceedings.neurips.cc/paper/2001/hash/4b86abe48d358ecf194c56c69108433e-Abstract.html.

Bert Kappen, Vicenç Gómez, and Manfred Opper. Optimal control as a graphical model inference problem. *CoRR*, abs/0901.0633, 2009. URL http://arxiv.org/abs/0901.0633.

Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In Yoshua Bengio and Yann LeCun (eds.), *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014. URL http://arxiv.org/abs/1312.6114.

Hugo Larochelle and Iain Murray. The neural autoregressive distribution estimator. In Geoffrey J. Gordon, David B. Dunson, and Miroslav Dudík (eds.), *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2011, Fort Lauderdale, USA, April 11-13, 2011*, volume 15 of *JMLR Proceedings*, pp. 29–37. JMLR.org, 2011. URL http://proceedings.mlr.press/v15/larochelle11a/larochelle11a.pdf.

Sergey Levine. Reinforcement learning and control as probabilistic inference: Tutorial and review. *CoRR*, abs/1805.00909, 2018. URL http://arxiv.org/abs/1805.00909.

Zhuoran Li, Ling Pan, and Longbo Huang. Beyond conservatism: Diffusion policies in offline multi-agent reinforcement learning. *CoRR*, abs/2307.01472, 2023. doi: 10.48550/arXiv.2307.01472. URL https://doi.org/10.48550/arXiv.2307.01472.

Zhixuan Liang, Yao Mu, Mingyu Ding, Fei Ni, Masayoshi Tomizuka, and Ping Luo. Adaptdiffuser: Diffusion models as adaptive self-evolving planners. *arXiv preprint arXiv:2302.01877*, 2023.

Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. In Yoshua Bengio and Yann LeCun (eds.), *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016. URL http://arxiv.org/abs/1509.02971.

Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. *CoRR*, abs/1706.02275, 2017. URL http://arxiv.org/abs/1706.02275.

Cheng Lu, Huayu Chen, Jianfei Chen, Hang Su, Chongxuan Li, and Jun Zhu. Contrastive energy prediction for exact energy-guided diffusion sampling in offline reinforcement learning. *arXiv preprint arXiv:2304.12824*, 2023.

David Henry Mguni, Yutong Wu, Yali Du, Yaodong Yang, Ziyi Wang, Minne Li, Ying Wen, Joel Jennings, and Jun Wang. Learning in nonzero-sum stochastic games with potentials. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pp. 7688–7699. PMLR, 2021. URL http://proceedings.mlr.press/v139/mguni21a.html.

OpenAI. Gpt-4 technical report, 2023.

Neil C. Rabinowitz, Frank Perbet, H. Francis Song, Chiyuan Zhang, S. M. Ali Eslami, and Matthew M. Botvinick. Machine theory of mind. In Jennifer G. Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pp. 4215–4224. PMLR, 2018. URL http://proceedings.mlr.press/v80/rabinowitz18a.html.

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with CLIP latents. *CoRR*, abs/2204.06125, 2022. doi: 10.48550/arXiv.2204.06125. URL https://doi.org/10.48550/arXiv.2204.06125.

Konrad Rawlik, Marc Toussaint, and Sethu Vijayakumar. Approximate inference and stochastic optimal control. *CoRR*, abs/1009.3958, 2010. URL http://arxiv.org/abs/1009.3958.

Konrad Rawlik, Marc Toussaint, and Sethu Vijayakumar. On stochastic optimal control and reinforcement learning by approximate inference (extended abstract). In Francesca Rossi (ed.), *IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, China, August 3-9, 2013*, pp. 3052–3056. IJCAI/AAAI, 2013. URL http://www.aaai.org/ocs/index.php/IJCAI/IJCAI13/paper/view/6658.

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, volume 32 of *JMLR Workshop and Conference Proceedings*, pp. 1278–1286. JMLR.org, 2014. URL http://proceedings.mlr.press/v32/rezende14.html.

Lloyd S Shapley. Stochastic games. *Proceedings of the national academy of sciences*, 39(10): 1095–1100, 1953.

Yoav Shoham and Kevin Leyton-Brown. *Multiagent systems: Algorithmic, game-theoretic, and logical foundations*. Cambridge University Press, 2008.

Yang Song, Conor Durkan, Iain Murray, and Stefano Ermon. Maximum likelihood training of score-based diffusion models. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 1415–1428, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/0a9fdbb17feb6ccb7ec405cfb85222c4-Abstract.html.

Zheng Tian, Ying Wen, Zhichen Gong, Faiz Punakkath, Shihao Zou, and Jun Wang. A regularized opponent model with maximum entropy objective. In Sarit Kraus (ed.), *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pp. 602–608. ijcai.org, 2019. doi: 10.24963/ijcai.2019/85. URL https://doi.org/10.24963/ijcai.2019/85.

Marc Toussaint. Robot trajectory optimization using approximate inference. In *Proceedings of the 26th Annual International Conference on Machine Learning - ICML '09*, pp. 1–8, 2009a. ISBN 9781605585161. doi: 10.1145/1553374.1553508. URL https://homes.cs.washington.edu/{~}todorov/courses/amath579/reading/Toussaint.pdfhttp://portal.acm.org/citation.cfm?doid=1553374.1553508.

Marc Toussaint. Probabilistic inference as a model of planned behavior. *Kunstliche Intelligenz*, 3, 01 2009b.

Marc Toussaint and Amos J. Storkey. Probabilistic inference for solving discrete and continuous state markov decision processes. In William W. Cohen and Andrew W. Moore (eds.), *Machine Learning, Proceedings of the Twenty-Third International Conference (ICML 2006), Pittsburgh, Pennsylvania, USA, June 25-29, 2006*, volume 148 of *ACM International Conference Proceeding Series*, pp. 945–952. ACM, 2006. doi: 10.1145/1143844.1143963. URL https://doi.org/10.1145/1143844.1143963.

Ermo Wei, Drew Wicke, David Freelan, and Sean Luke. Multiagent soft q-learning. In *2018 AAAI Spring Symposia, Stanford University, Palo Alto, California, USA, March 26-28, 2018*. AAAI Press, 2018. URL https://aaai.org/ocs/index.php/SSS/SSS18/paper/view/17508.

Ying Wen, Yaodong Yang, Rui Luo, and Jun Wang. Modelling bounded rationality in multi-agent interactions by generalized recursive reasoning. *arXiv preprint arXiv:1901.09216*, 2019a.

Ying Wen, Yaodong Yang, Rui Luo, Jun Wang, and Wei Pan. Probabilistic recursive reasoning for multi-agent reinforcement learning. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019b. URL https://openreview.net/forum?id=rkl6As0cF7.

Ying Wen, Yaodong Yang, and Jun Wang. Modelling bounded rationality in multi-agent interactions by generalized recursive reasoning. In Christian Bessiere (ed.), *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, pp. 414–421. ijcai.org, 2020. doi: 10.24963/ijcai.2020/58. URL https://doi.org/10.24963/ijcai.2020/58.

Cheng Zhang, Judith Bütepage, Hedvig Kjellström, and Stephan Mandt. Advances in variational inference. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):2008–2026, 2018.

Zhengbang Zhu, Minghuan Liu, Liyuan Mao, Bingyi Kang, Minkai Xu, Yong Yu, Stefano Ermon, and Weinan Zhang. Madiff: Offline multi-agent learning with diffusion models. *CoRR*, abs/2305.17330, 2023. doi: 10.48550/arXiv.2305.17330. URL https://doi.org/10.48550/arXiv.2305.17330.