

Towards Equilibrium Coordination with Stein Variational Game

Zhiyuan Zhang¹ and Panagiotis Tsiotras²

Abstract—Differential Dynamic Games (DDG) often admit non-unique and, in some cases, an infinite number of Nash Equilibria (NE). Miscoordination among agents in selecting a specific Nash Equilibrium can result in cost inefficiencies and compromise any stabilizing properties of these equilibria. In the case of Generalized Nash Equilibria (GNE), such miscoordination can lead to constraint violations, potentially causing catastrophic outcomes, such as collisions in practical applications. To address the equilibrium coordination problem, we propose a novel framework for estimating and selecting equilibria in a DDG. Our method utilizes a diffusion process to model the distribution of Nash Equilibria in a DDG, constructed through a sequence of mappings defined by Stein Variational Gradient Descent (SVGD). This distribution serves as a prior belief over the non-ego agents’ equilibrium choices, which is dynamically refined using Bayesian inference as the game evolves. We validate the proposed approach in an autonomous vehicle traffic trajectory planning problem, demonstrating its effectiveness in environments where agents operate without explicit communication regarding their chosen Nash Equilibria.

I. INTRODUCTION

In many games, particularly Differential Dynamic Games (DDG), there exist multiple, and often infinitely many, Nash Equilibria (NE). To realize their expected outcomes and maintain stability in repeated games, agents must coordinate which NE to adopt and align their policies accordingly. However, misalignment of policies, even in simple games like the “Battle of the Sexes,” leads to a lower social utility than scenarios where explicit coordination is available, such as in a correlated equilibrium [1]. The challenge of equilibrium selection without external coordination remains an open and pressing problem [2].

A. Existing Methods

While solving for NE is a PPAD-complete problem [3], many methods can find approximate or local NE by imposing assumptions on the regularity of DDGs. One line of work discretizes the joint action space of the agents, converting the DDG into a classic matrix game [4]. However, the combinatorial nature of the joint action space poses significant scalability challenges, particularly as the discretization resolution, the length of the horizon, and the number of agents increase. Practical implementations require carefully designed heuristics to keep the algorithm computationally

tractable. Despite this shortcoming, advanced versions of this method, employing tree search for action pair selection, have been adopted by commercial robotaxi companies [5].

Another prominent approach is iLQGame [6], an extension of min-max Differential Dynamic Programming (DDP) [7], which approximates DDGs as multi-stage Linear Quadratic Games (LQGames) that admit analytical solutions under certain regularity conditions. Starting from a reference action sequence pair, iLQGame iteratively solves a series of approximated LQGames, shifting the reference sequence toward the solution.

Another class of methods treats Generalized Nash Equilibrium Problems (GNEP) as constrained optimization problems. ALGame [8] and RD3G [9], for instance, employ variants of the constrained Newton method to find an action sequence that satisfies the necessary conditions for GNE. These Newton-based solvers are generally faster than DDP-based approaches and handle state constraints directly. However, like DDP methods, they can only find local GNE. Moreover, they rely on a good initial guess within the region of convergence of a local solution, and the equilibrium found is sensitive to the choice of this initial guess.

Among the methods discussed, those that convert DDGs to discrete matrix games are unique in that they are not inherently restricted to local solutions, provided that the joint action space is adequately sampled. These sampling-based approaches have the potential to find multiple NE, but the accuracy of this approach is severely hampered by the inherent curse of dimensionality.

More recent works combine sampling and local solvers to find multiple NE in DDGs. For instance, [10] draws samples in the joint strategy space, then uses the samples as initial guesses to solve for local NE using an LQ approximation. Reference [11] recasts a DDG as a moving horizon estimation problem, which is solved using an Unscented Kalman Filter (UKF). The posterior unscented particles are used as coarse candidate solutions that are then refined with a local NE solver.

Our work is inspired by these recent advances, with several important distinctions. First, the particles we draw are updated in a correlated fashion. In [10], each particle independently solves for a local Nash equilibrium from a different initial condition, and is updated without interaction with other particles. Since nearby particles will be drawn to the same local solution, mode collapse is inevitable. The authors of [10] utilize a “CombineDuplicates” step to avoid repeating calculations for collapsed samples, and noted a 6X reduction in execution time from this operation. In our method, the update of the particles is prescribed by the SVGD update law (5), and contains a repelling term between

This work has been sponsored by ONR awards N00014-23-2308 and NSF award IIS-2008686

¹ School of Aerospace Engineering, Institute for Robotics and Intelligent Machines, Georgia Institute of Technology, Atlanta, GA 30332, USA, Email: zzhang615@gatech.edu

² School of Aerospace Engineering, Institute for Robotics and Intelligent Machines, Georgia Institute of Technology, Atlanta, GA 30332, USA, Email: tsiotras@gatech.edu

particles. This avoids mode collapse and improves sampling efficiency. In this aspect, our work is analogous to [11], where the samples in UKF updates are correlated. Second, our method is unified in contrast to the two-stage approach in [11]. There, unscented particles are propagated through a moving horizon estimation process and subsequently used as initialization points for a separate local optimization solver. In our approach, the Nash equilibrium is obtained through a single, cohesive procedure. The distribution represented by the particles is the solution, and there is no need for a standalone local solver.

B. Challenges of Computing Multiple Equilibria and Coordination

The second challenge we address, although to a lesser extent, is how to utilize the potentially multiple NEs. Methods that focus on finding a single, local NE generally assume the existence of a central coordinator who selects the equilibrium and assigns policies to individual agents [6], [8]. With this approach, the game is solved solely by the coordinator, with no coordination responsibility placed on the agents themselves.

Other equilibrium concepts assume less responsibility from the central coordinator. For example, correlated equilibria [12] only require the coordinator to draw a random sample from a problem-specific distribution. A decentralized framework distributes decision-making between agents and a coordinator, as seen in resource allocation or congestion games, where the coordinator sets prices or charges [13]. When resource prices are used as a way to regulate agent behavior, agents implicitly coordinate to jointly reach a solution.

C. Contributions

The proposed method provides a novel approach to equilibrium computation and agent coordination in one-shot constrained Differential Dynamic Games (DDGs). Our contributions are twofold:

Probabilistic Representation of Nash Equilibria: We introduce the use of Stein Variational Gradient Descent (SVGD) to approximate the *distribution* of Nash Equilibria in DDGs. Unlike conventional deterministic solvers that yield a single, local equilibrium, our approach captures the variability of all possible equilibria, enabling a richer and more robust analysis.

Bayesian Belief Updating for Adaptive Decision-Making: We develop a Bayesian framework that treats the estimated Nash Equilibrium distribution as a prior and refines it dynamically based on the observed interactions with other agents. This allows the agents to continuously adapt their strategies in response to their opponents' behavior, leading to more realistic and responsive decision-making compared to traditional static equilibrium concepts.

These contributions collectively provide a robust and adaptable framework for equilibrium computation in dynamic games, making our method particularly valuable for applications in multi-agent robotic systems when inter-agent communication is unavailable. We provide numerical examples to demonstrate the benefits of the proposed approach.

II. PROPOSED METHODOLOGY

A. Problem Formulation

We are interested in an N-player, general sum, fixed horizon differential dynamic game in discrete time with state constraints. Let T denote the horizon of the game and let $f^i(x_k^i, u_k^i)$ denote the discrete-time agent dynamics, where x_k^i and u_k^i denote the state and the control of agent i at time step k , respectively. We will denote by x^i the state trajectory of agent i , by x_k the state of all agents at time step k , and by x the state of all agents at all time steps, and similarly for u . Finally, $[a..b]$ will denote all integers between a, b , including a, b .

The generalized NE (GNE) is defined as follows:

$$\min_{u^i} J^i(x, u^i), \quad \forall i \in [1..N], \quad (1a)$$

$$x_{k+1}^i = f^i(x_k^i, u_k^i), \quad \forall k \in [0..T-1], \forall i \in [1..N], \quad (1b)$$

$$h(x_k) \leq 0, \quad \forall k \in [1..T], \quad (1c)$$

where $h(x_k)$ is the state constraint that couples the actions of all agents, and where J^i, f^i, h are differentiable functions, whose domains are clear from the context. Throughout this paper, we are concerned with a *perfect information game*, where the dynamics and cost function of each agent are common information accessible to all agents. Note that the structure of the game may lead to the existence of a unique NE, multiple distinct NE, or even a manifold of NE [14]. We assume that all *non-ego agents* solve for a NE independently and execute the corresponding control sequences accordingly. Once an agent chooses its control sequence, it commits to this policy regardless of the other agents' behavior. Our proposed method provides a control sequence for the *ego agent* that repeatedly plays the best response to the belief over the other agents' selected NE.

B. Stein Variational Gradient Descent

Our method reformulates the DDG as a variational inference problem. The target distribution is defined using a residual-based likelihood function, which maps the KKT residual of a candidate solution to the likelihood that this candidate satisfies the necessary conditions for a Nash equilibrium.

Stein Variational Gradient Descent (SVGD) [15] is a variational inference method that utilizes Kernelized Stein Discrepancy [16] to find a closed-form solution to the steepest descent direction to minimize the KL divergence between a proposal distribution and the target distribution. In effect, SVGD generates a deterministic flow that transforms any distribution to the target distribution. In our application, this allows us to sample from the intractable distribution of NE in the game. Below, we briefly introduce SVGD, and direct the interested reader to [15] for more details on the method.

Let p, q be two continuously differentiable densities over the parameter $\theta \in \Theta \subseteq \mathbb{R}^d$, and let $\phi(\theta)$ be a smooth vector-valued function. The Kernelized Stein Discrepancy

(KSD) [16] is defined as

$$\mathbb{S}(q, p) = \max_{\phi \in \mathcal{H}^d} \{ [\mathbb{E}_{\theta \sim q} (\text{tr}(\mathcal{A}_p \phi(\theta)))]^2, \text{ s.t. } \|\phi\|_{\mathcal{H}^d} \leq 1 \}, \quad (2)$$

where $\mathcal{A}_p \phi(\theta) = \phi(\theta) \nabla_{\theta} \log p(\theta) + \nabla_{\theta} \phi(\theta)$ is the Stein operator, and $\mathcal{H}^d$ is a Reproducing Kernel Hilbert Space (RKHS). Note that $\mathbb{S}(q, p) = 0$ if and only if $q = p$, and hence KSD can be used as a discrepancy measure between two smooth distributions. A unique feature of KSD is that it admits an analytical solution for the optimal ϕ^* that maximizes KSD, which is generally intractable when ϕ is chosen from function spaces other than an RKHS. The optimal ϕ^* that solves (2) is

$$\phi_{q,p}^*(\cdot) = \mathbb{E}_{\theta \sim q} [\mathcal{A}_p k(\theta, \cdot)], \quad (3)$$

where $k(\cdot, \cdot)$ is the kernel in the RKHS $\mathcal{H}^d$. Without loss of generality, we use the Radial Basis Function (RBF) kernel, $k(x, x') = \exp(-\|x - x'\|^2/h)$, where h is a scaling parameter for all numerical examples in this work.

Consider now a small perturbation of the identity map $T(\theta) = \theta + \epsilon \phi(\theta)$, where $\phi(\theta)$ is a smooth perturbation function and ϵ sufficiently small such that $T : \Theta \rightarrow \Theta$ is a one-to-one map. Denote by q_0 a tractable proposal distribution, let the random variable $z = T(\theta)$ where $\theta \sim q_0$, and let q be the density of z . The directional derivative of the KL-divergence at θ along $\phi(\theta)$ is

$$\nabla_{\epsilon} \text{KL}(q||p) \Big|_{\epsilon=0} = -\mathbb{E}_{\theta \sim q_0} [\text{tr}(\mathcal{A}_p \phi(\theta))]. \quad (4)$$

Notice that this expression is similar to KSD in (2). Therefore, according to (3), the optimal perturbation function that maximizes the directional derivative (4) is given by

$$\phi_{q,p}^*(\cdot) = \mathbb{E}_{\theta \sim q_0} [k(\theta, \cdot) \nabla_{\theta} \log p(\theta) + \nabla_{\theta} k(\theta, \cdot)]. \quad (5)$$

This defines the optimal map $T^*(\theta) = \theta + \epsilon \phi_{q,p}^*(\theta)$ to minimize the KL-divergence between q and p .

Building upon this result, SVGD provides an iterative variational inference method with the goal of finding a particle-defined distribution q that is closest to the target distribution p . SVGD draws samples (particles) from a proposal distribution q_0 , then applies the map T^* , evaluating the expectation $\mathbb{E}_{\theta \sim q_0}$ in (5) as an empirical mean of the particles. Hence, from (5),

$$\phi^*(\theta) = \frac{1}{N_p} \sum_{j=1}^{N_p} [k(\theta^j, \theta) \nabla_{\theta^j} \log p(\theta^j) + \nabla_{\theta^j} k(\theta^j, \theta)], \quad (6)$$

where N_p is the number of particles and θ^j is the j -th particle. The map T^* is applied repeatedly to the particles, effectively updating the proposal distribution q_0 with q after each iteration.

The information about the target distribution enters the update law (6) in the form of $\nabla_{\theta} \log p(\theta)$. Since $\nabla_{\theta} \log p(\theta) = \nabla_{\theta} \log \frac{p(\theta)}{Z}$, (6) is especially useful for working with unnormalized distributions.

C. Residual Descent Differential Dynamic Game

Using the method of multipliers, we can formulate the Lagrangian for the constrained game (1) as follows

$$L^i(x, u^i) = J^i(x, u^i) + \lambda^{iT} F^i(x^i, u^i) + \mu^T H(x), \quad (7)$$

where λ is the Lagrange multiplier for the dynamics constraint $F^i(x^i, u^i)$, which is concatenated from $f^i(x_k^i, u_k^i)$ across $k \in [0..T-1]$ in (1b), and μ is the Lagrange multiplier for the state and control constraint $H(x)$ ¹, concatenated from $h(x_k)$ in (1c). We use $-i$ to denote all agents except agent i . Note that (7) implicitly depends on u^{-i} through the combined state x .

The necessary conditions are given as

$$\frac{\partial L^i}{\partial x^i} = 0, \quad \forall i \in [1..N], \quad (8a)$$

$$\frac{\partial L^i}{\partial u^i} = 0, \quad \forall i \in [1..N], \quad (8b)$$

$$F(x, u) = 0, \quad (8c)$$

$$H(x) \leq 0, \quad (8d)$$

$$\mu \geq 0, \quad (8e)$$

where the inequalities (8d) and (8e) are to be understood componentwise. Next, we use the semicolon to denote the vertical concatenation of vectors, i.e., $[a; b] = [a^T, b^T]^T$. For notational simplicity, we denote $\frac{\partial L}{\partial x} = [\frac{\partial L}{\partial x^1}; \frac{\partial L}{\partial x^2}; \dots; \frac{\partial L}{\partial x^N}]$, $\frac{\partial L}{\partial u} = [\frac{\partial L}{\partial u^1}; \frac{\partial L}{\partial u^2}; \dots; \frac{\partial L}{\partial u^N}]$. We then define the residual vector for the DDG (1) as

$$\mathcal{R}(x, u, \lambda, \mu) = \left[\frac{\partial L}{\partial x}; \frac{\partial L}{\partial u}; F(x, u); \mathbb{H}(H(x, u)) \right], \quad (9)$$

where $\mathbb{H}$ is the Heaviside step function. A solution to (9) that also satisfies (8e) fulfills the necessary condition for a Nash equilibrium². Note that the calculation of (9) requires knowledge of the dynamics and cost function of all agents under our perfect information assumption.

The residual $\mathcal{R}(x, u, \lambda, \mu)$ in (9) has four arguments and has, potentially, a high-dimensional domain. We wish to create an approximation to $\mathcal{R}$ that is defined on a lower-dimensional domain to simplify the ensuing calculations. First, we note that, given the control u , we can propagate the dynamics function $x_{k+1}^i = f^i(x_k^i, u_k^i)$ to obtain the state trajectory x_u^* that ensures that the dynamics residual $F(x_u^*, u)$ is zero. Next, we compute

$$(\lambda_u^*, \mu_u^*) = \arg \min_{\lambda, \mu \geq 0} \left\| \left(\frac{\partial L}{\partial x}, \frac{\partial L}{\partial u} \right) \Big|_{x=x^*} \right\|^2. \quad (10)$$

When the constraints $F(x, u) = 0, H(x) = 0$ are satisfied, the approximation is exact. This allows us to reduce the dimension of the domain by half at the cost of solving a smaller quadratic optimization problem. We denote $\mathcal{R}^*(u) := r(x_u^*, u, \lambda_u^*, \mu_u^*)$

¹There's no need to distinguish constraints involving x^i from those that don't, since only the former contribute to the gradients.

²Since the constraint Lagrange multipliers for all agents are the same, this is a variational GNE (v-GNE). Details on GNE and v-GNE are explored in [17].

Candidate controls that satisfy the necessary conditions (8) may be saddle points. At the end of the SVGD process, we therefore retain only those particles whose Hessian of the Lagrangian $\nabla^2 L$ is positive definite.

Denote the set of admissible controls as $\mathcal{U}$. Let $\pi_\theta : [1..N] \times [0..T-1] \rightarrow \mathcal{U}$ be the joint open-loop policy parameterized by θ . In the simplest case, $u_k^i = \pi_\theta(i, k) = \theta_{i,k}$, i.e., the parameter is the control vector itself. This is the parameterization used in all numerical examples in Section III. However, π_θ may be any differentiable parameterization. Let $r(\pi_\theta)$ denote the NE residual associated with π_θ .

D. Stein Variational Game

Let O denote the event that the control sequence π_θ , parameterized by θ , satisfies the necessary conditions of NE. We compute the conditional likelihood $p(\theta|O)$ from a prior distribution q via Bayesian inference

$$p(\theta|O) = \frac{p(O|\theta)q(\theta)}{\int p(O|\theta)q(\theta)d\theta}, \quad (11)$$

where $q(\theta)$ stands for the prior distribution. To approximate $p(\theta|O)$ with (6), we need to find $\nabla_\theta \log p(\theta|O)$. Using (11), we have

$$\nabla_\theta \log p(\theta|O) = \nabla_\theta \log p(O|\theta) + \nabla_\theta \log q(\theta). \quad (12)$$

We define the residual-likelihood function as

$$p(O|\theta) \propto \exp(-\alpha r(\pi_\theta)), \quad \alpha > 0, \quad (13)$$

which links the residual value of a parameter θ to the probability that θ satisfies the necessary condition of NE. The formulation in (13) is based on the Boltzmann action selection model, which is commonly used to represent agents with bounded rationality [18]. This bounded rationality property may arise from observation noise, limited computational resources, or suboptimal decision-making processes. The parameter α in (13) reflects the degree of rationality of the decision-maker. Algorithm 1 details the Stein Variational Game algorithm to estimate the conditional distribution of the control parameter θ conditioned on θ resulting in a NE.

While we do not provide a convergence proof in this work, we refer interested readers to [15], [19] for theoretical results on the convergence of SVGD.

E. Belief Update and Best Response

The Stein Variational Game (SVG) provides a mechanism for sampling from the target distribution by first drawing sample particles from a tractable proposal distribution, then transforming the samples with the flow defined by SVGD. The posterior particles define the empirical distribution $q(\theta) = \frac{1}{m} \sum_{i=1}^m \delta(\theta - \theta_j)$ and form prior beliefs of the non-ego agents' chosen NE. In the following, we distinguish between the conditional distribution $p(\theta|O)$ from the pre-game SVGD process and the belief of NE updated during the game using a tilde.

During the execution of the game, we update the belief using the observed actions of the non-ego agents using

Bayesian inference. Let $\tilde{q}_k^j$ be the ego agent's belief of agent j 's selected NE at time step k . We update $\tilde{q}_k^j$ using

$$\tilde{q}_{k+1}^j = p(\theta|u_k^j) = \frac{p(u_k^j|\theta)p(\theta|\tilde{q}_k^j)}{\sum_{\theta^l} p(u_k^j|\theta^l)p(\theta^l|\tilde{q}_k^j)}. \quad (14)$$

We further stipulate that $p(u_k^j|\theta) \propto \exp(k(u_k^j, \pi_\theta^j))$, where $k(\cdot)$ is the kernel function previously defined in RKHS used in SVGD.

The control of the ego agent is the best response to the expected control sequence over the current belief of the other agents:

$$u_k^i = \arg \min_{u_k^i} \mathbb{E}_{\theta^j \sim \tilde{q}_k^j, \forall j \neq i} [J^i(x, u^i)|_{u^j = \pi_{\theta^j}}]. \quad (15)$$

Algorithm 2 details the belief update. Before the game starts, Algorithm 1 finds the conditional distribution $p(\theta|O)$. As the game progresses, Algorithm 2 updates the belief online and provides a control for the ego-agent at each time step.

Algorithm 1 Stein Variational Game

```

Input  $x_0, q_0$ 
for each  $i$  in 1 to  $N_p$  do
  sample initial particles  $\theta^i \sim q_0$ 
end for
for each  $\ell$  in 0 to  $\ell_{\text{svgd}} - 1$  do
  for each  $i$  in 1 to  $N_p$  do
    create  $r(\pi_{\theta^i})$  from (9)(10)
     $p(O|\theta^i) \propto \exp(-\alpha r(\pi_{\theta^i}))$ 
     $\nabla_{\theta^i} \log p(\theta^i|O) = \nabla_{\theta^i} \log p(O|\theta^i) + \nabla_{\theta^i} \log q_i(\theta^i)$ 
  end for
  for each  $i$  in 1 to  $N_p$  do
     $\phi^*(\theta^i) = \frac{1}{m} \sum_{j=1}^m [k(\theta^j, \theta^i) \nabla_{\theta^j} \log p(\theta^j) + \nabla_{\theta^j} k(\theta^j, \theta^i)]$ 
     $\theta^i = \theta^i + \eta \phi^*(\theta^i)$ 
     $q_{\ell+1} = \frac{1}{m} \sum_{i=1}^m \delta(\theta - \theta_j)$ 
  end for
end for
Keep only particles whose  $\nabla^2 L(x, u) \geq 0$ 
output  $q_{\ell_{\text{svgd}}}$ 

```

Algorithm 2 Belief Update and Best Response

```

Input  $\tilde{q}_0^j = q_{\ell_{\text{svgd}}}, \forall j \neq i, j \in [1..N]$ 
for each  $k$  in 0 to  $T$  do
   $u_k^i = \arg \min_{u_k^i} \mathbb{E}_{\theta^j \sim \tilde{q}_k^j, \forall j \neq i} [J^i(x, u^i)|_{u^j = \pi_{\theta^j}}]$ 
  Execute ego agent control  $u_k^i$ , observe non-ego agent control  $u_k^j, \forall j \neq i$ 
   $\tilde{q}_{k+1}^j = p(\theta|u_k^j) = \frac{p(u_k^j|\theta)p(\theta|\tilde{q}_k^j)}{\sum_{\theta^l} p(u_k^j|\theta^l)p(\theta^l|\tilde{q}_k^j)}$ 
end for

```

III. NUMERICAL RESULTS

We evaluated SVG using three experiments. First, we demonstrate the representative power of the particle-defined distribution in the SVGD process using two examples of relatively low dimensions and simple NE for an intuitive

interpretation. We use these examples to show that the SVGD framework is capable of representing multi-modal NE distributions. The third example is a more realistic multi-lane traffic merging scenario involving multiple vehicles.

A. Low Dimension Examples

1) *Bimodal Convergence*: For notational simplicity, define the quadratic cost $G_Q(v) = v^T Q v$. Consider the game defined in (1), and specify the following instance:

$$J^i(x, u^i) = \sum_{k=1}^T \left[-e^{-G_Q(x_k^1 - p_1) - G_Q(x_k^2 - p_2)} - e^{-G_Q(x_k^2 - p_1) - G_Q(x_k^1 - p_2)} \right] + \sum_{k=0}^{T-1} u_k^{iT} R u_k^i, \quad (16a)$$

$$f(x_k^i, u_k^i) = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} x_k^i + \begin{bmatrix} 0.5(\Delta t)^2 \\ \Delta t \end{bmatrix} u_k^i, \quad (16b)$$

where Q, R are cost matrices, and p_1, p_2 are target states. This is a two-agent coordination game with double integrator dynamics. To minimize the cost, one agent needs to move to p_1 while the other needs to move to p_2 , but it does not matter which agent goes to which target. Figure 1 shows that SVG indeed converges to a bimodal distribution of terminal states with particle-defined NE distribution, before the observed action of the non-ego agent is used to refine the belief.³

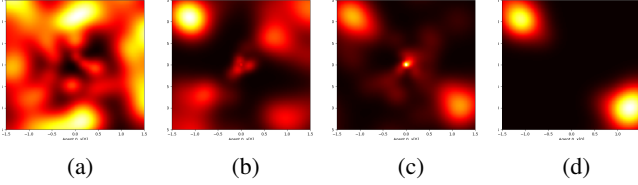


Fig. 1: Terminal state distribution in the particle-defined NE distribution at different stages of the diffusion. (d) excludes particles with negative Hessian. The distribution is smoothed with Radial Basis Function $\delta(\theta) = \sum_i \exp(\theta - \theta_i)$.

2) *1D Manifold Convergence*: Next, consider a similar game as (16b), but with the following cost objective:

$$J^i(x, u^i) = -e^{-(x_T^{iT} Q x_T^i + x_T^{2T} Q x_T^2 - 1)^2} + \sum_{k=0}^{T-1} u_k^{iT} R u_k^i, \quad (17)$$

where Q, R are cost weights. In this problem, the two agents collectively control a point following the double integrator dynamics in 2D Cartesian space. One agent controls the acceleration along the x-axis, and the other controls the acceleration along the y-axis. The goal of the game is to move the point close to the unit circle, but the phase angle is irrelevant. For this problem, the NE lies on a 1D manifold defined by the unit circle. Figure 2 shows the resultant distribution of the terminal states from SVG.⁴

³For this example, the following parameters are used: $p_1 = [1, 0], p_2 = [-1, 0], Q = [1, 0; 0, 0], R = 1 \times 10^{-2} I, dt = 0.2, T = 5, q_0 \sim \mathcal{N}(0, 3I), \ell_{\text{svgd}} = 20$ iterations, and $N_p = 100$ particles.

⁴For this example, we have chosen $Q = [1, 0; 0, 0], R = 1 \times 10^{-2}, dt = 0.2, T = 5, q_0 \sim \mathcal{N}(0, 3I), \ell_{\text{svgd}} = 20$ iterations, and $N_p = 100$ particles.

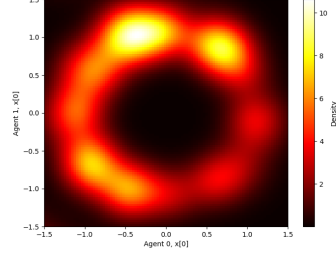


Fig. 2: Terminal state distribution in the particle-defined NE distribution from SVG.

B. Traffic Merging

Next, we consider a more practical N-car merging problem where the full SVG algorithm may be utilized. Each car follows the kinematic bicycle model, where the state variables are represented as $[x, y, v, \theta]$, corresponding to longitudinal position, lateral position, speed, and heading, respectively. The control input for each car is given by $u = [\delta_T, \delta_S]$, where δ_T represents the acceleration and δ_S denotes the steering angle.

The game begins with multiple cars traveling in two adjacent lanes, each moving with different speeds. The cars in the right lane attempt to merge into the left lane. Each agent optimizes a linear quadratic cost function defined as

$$J^i(x, u^i) = \sum_{k=1}^T (x_k^i - x_{\text{ref}}^i)^T Q_r^i (x_k^i - x_{\text{ref}}^i) + x_k^{iT} Q x_k^i + u_k^{iT} R^i u_k^i, \quad (18)$$

which balances the multiple objectives in a lane merging: moving to the target lane, maintaining current speed, minimizing heading deviation, and minimizing acceleration. The empirically chosen cost matrix Q_r^i penalizes deviations from the target lateral position (lane) and reference speed, while Q^i penalizes excessive heading deviations. The reference state x_{ref}^i specifies the target longitudinal velocity for the i^{th} car, which is randomly initialized, and the target lateral position at the center of the left lane⁵. The state constraint functions put a lower bound on the Cartesian distance between cars, namely,

$$h(x_k^i, x_k^j) = d_{\min}^2 - (x_k^i - x_k^j)^T Q_c (x_k^i - x_k^j), \quad (19)$$

for all $i \neq j \in [1..N], k \in [1..T]$, where $d_{\min}$ is the minimum safety distance between cars, and Q_c selects appropriate terms in the state difference. We use 20 particles and a horizon of 20 steps in the following merging simulations.

Figure 3 illustrates two Nash equilibria identified by the SVG solver in the 5-car merging scenario. These equilibria, among others, form the ego agent's prior belief about the NE selected by non-ego agents.

To assess the effectiveness of SVG in preventing catastrophic outcomes due to miscoordination, we compare three approaches to implementing a game-theoretic controller.

⁵ $Q_r = \text{diag}(0, 0.1, 0.01, 0), Q = \text{diag}(0, 0, 0, 1.0), R = 0.3 I_2$

- *Baseline*: Each agent independently employs a Newton solver with a randomly initialized guess.
- *SVG + Best*: The ego agent uses Algorithm 1 to compute the NE distribution, then selects the particle with the highest social utility.
- *SVG + Learning*: The ego agent uses Algorithm 1 and 2.

Each experiment is repeated 100 times with randomized initial states for all agents, including variations in longitudinal position and velocity. To test the robustness of the Algorithm 2, we added Gaussian noise $w_u = \mathcal{N}(0, 0.1)$ to the observed u to simulate observation noise.

Table I reports the percentage of experiments in which a collision occurs under each scenario. In the baseline case without coordination, the collision rate is high, highlighting the peril of miscoordination. The SVG + Best approach achieves a modest reduction in collision rate, likely because SVG is less vulnerable to local equilibria, and its selection of high-social-utility equilibria may offer a slightly larger region of attraction. However, this approach does not utilize the NE distribution provided by SVG. In contrast, SVG + Learning substantially reduces the collision rate. These results demonstrate the proposed method’s effectiveness in enhancing coordination and safety in multi-agent interactions.

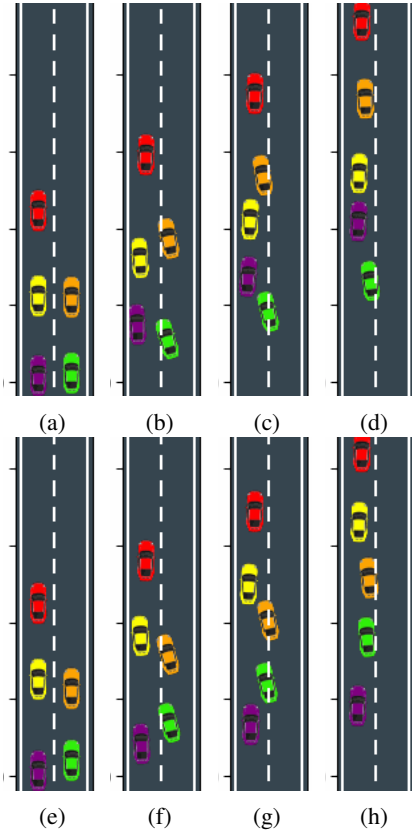


Fig. 3: Time-sequenced snapshots of state trajectories for two representative equilibria in the 5-car merging problem. The top panels (a)–(d) show one equilibrium, and the bottom panels (e)–(h) show another.

TABLE I: Collision ratio of different approaches to coordination.

Cars	Baseline	SVG + Best	SVG + Learning
2	38%	43%	0%
3	52%	43%	0%
4	72%	68%	8%
5	83%	78%	12%
6	88%	72%	19%

TABLE II: Average runtime for SVG.

Cars	Solution Time (s)	Single Particle (s)
2	2.7	0.137
3	4.3	0.215
4	6.9	0.350
5	9.7	0.485
6	12.1	0.605

Figure 4 shows the evolution of the weight associated with each particle (candidate NE) in the belief for a specific non-ego agent under Algorithm 2. Each color represents a different potential NE that the non-ego agent may have chosen. The height of the bar segment represents the relative weight of a potential NE. At the beginning of the game, the weight of each particle θ is determined by the residual $r(\pi_\theta)$. During the interaction with non-ego agents, their chosen NE becomes evident, and the weight converges to the particle associated with the purple bar. Note that the pink bar and the purple bar hold similar weights until step 5. This is likely because their associated NE are similar early in the horizon, and only diverge later on.

Table II presents the average runtime of the proposed algorithm. While the sampling-based SVG algorithm is inherently parallelizable, our current implementation evaluates particles serially, similar to the setup in [10]. To provide a direct comparison in terms of sampling complexity and computational efficiency, we also implemented the multi-agent unicycle planning problem described in [10]. In this benchmark, their implementation takes 285 ms for processing *each particle* for the 5-agent, 100 time step scenario, compared to 386 ms in our case. Despite the longer per-particle runtime, our approach offers two notable advantages: first, its efficient sampling is robust against mode collapse, enabling effective planning with significantly fewer particles—20 in our constrained 5-car merging scenario versus 150 in their unconstrained 5-agent task; and, second, our framework supports hard constraint handling, which is absent in [10].

IV. CONCLUSION

We have introduced a novel framework for discovering and coordinating distributions of NE in non-communicating multi-agent systems. Leveraging Stein Variational Gradient Descent (SVGD), we construct a sequence of deterministic mappings that enable efficient sampling from the distribution of local NE. Our method effectively captures multi-modal NE distributions, as demonstrated in two intuitive two-agent

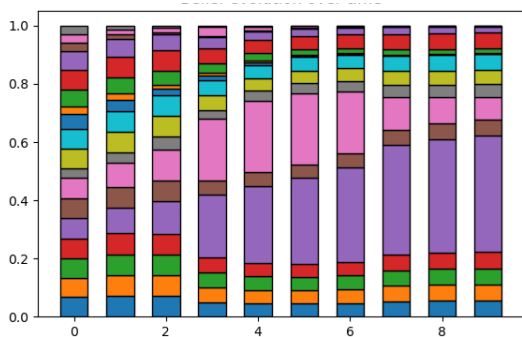


Fig. 4: Belief weight evolution over time in Algorithm 2, each color shows the weight of a specific particle.

coordination games.

Building on this, we utilized the NE distribution as the ego agent’s belief, continuously updating it via Bayesian inference during the game. This adaptive reasoning mechanism allows the agents to dynamically adjust their expectations, mitigating coordination failures. Through empirical validation in a car merging scenario, we showed that our approach significantly reduces collision rates—one of the key challenges arising from miscoordination in multi-agent interactions.

Several promising directions remain for future work. First, this paper adopts a trivial open-loop parameterization for the control policy π_θ . A natural extension is to investigate state-feedback policies, potentially parameterized using neural networks. Second, our formulation assumes a perfect-information game setting—an assumption that may be too strong for many real-world scenarios. Future work should aim to relax this assumption by exploring more realistic information structures, such as categorical or parametric uncertainty in non-ego agents’ cost function and dynamics. Lastly, continued improvements to the algorithm’s implementation and validation on physical robotic platforms will be essential for demonstrating its practical applicability.

This project is open source and available at <https://github.com/Nick-Zhang1996/SteinVariationalGame>.

REFERENCES

- [1] J. Duffy and N. Feltovich, “Correlated Equilibria, Good and Bad: An Experimental Study,” *International Economic Review*, vol. 51, no. 3, pp. 767–796, Aug. 2010.
- [2] W. Güth and H. Otsubo, “When efficiency requires arbitrary discrimination: Theoretical and experimental analysis of equilibrium selection,” *Games*, vol. 14, no. 5, p. 65, 2023.
- [3] X. Chen, X. Deng, and S.-H. Teng, “Settling the complexity of computing two-player Nash equilibria,” *J. ACM*, vol. 56, no. 3, May 2009.
- [4] A. Liniger and J. Lygeros, “A Noncooperative Game Approach to Autonomous Racing,” *IEEE Transactions on Control Systems Technology*, vol. 28, no. 3, pp. 884–897, 2020.
- [5] C. Paxton, V. Raman, G. D. Hager, and M. Kobilarov, “Combining neural networks and tree search for task and motion planning in challenging environments,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Vancouver, Canada, Sep 24–28 2017, pp. 6059–6066.
- [6] D. Fridovich-Keil, V. Rubies-Royo, and C. J. Tomlin, “An Iterative Quadratic Method for General-Sum Differential Games with Feedback Linearizable Dynamics,” in *IEEE International Conf. on Robotics and Automation*, May 31–Aug. 31 2020, pp. 2216–2222.
- [7] J. Morimoto, G. Zeglin, and C. Atkeson, “Minimax Differential Dynamic Programming: Application to a Biped Walking Robot,” in *International Conference on Intelligent Robots and Systems (IROS)*, vol. 2, Las Vegas, NV, 1–5 Oct. 2003, pp. 1927–1932.
- [8] S. L. Cleach, M. Schwager, and Z. Manchester, “ALGAMES: A Fast Solver for Constrained Dynamic Games,” in *Robotics: Science and Systems*, virtual, July 12–16, 2020.
- [9] Z. Zhang and P. Tsotras, “Residual descent differential dynamic game (RD3G) – a fast Newton solver for constrained general sum games,” in *IEEE International Conference on Robotics and Automation (ICRA)*, Atlanta, GA, May 19–23 2025.
- [10] L. Peters, D. Fridovich-Keil, C. J. Tomlin, and Z. N. Sunberg, “Inference-based strategy alignment for general-sum differential games,” in *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*, May 2020, pp. 1037–1045.
- [11] M. Bhatt, I. Askari, Y. Yu, U. Topcu, H. Fang, and N. Mehr, “MultiNash-PF: A Particle Filtering Approach for Computing Multiple Local Generalized Nash Equilibria in Trajectory Games,” *arXiv preprint arXiv:2410.05554*, 2024.
- [12] R. Aumann, “Subjectivity and correlation in randomized strategies,” *Journal of Mathematical Economics*, vol. 1, no. 1, pp. 67–96, 1974.
- [13] B. G. Bakhshayesh and H. Kebriaei, “Decentralized equilibrium seeking of joint routing and destination planning of electric vehicles: A constrained aggregative game approach,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 8, pp. 13 265–13 274, 2022.
- [14] A. Kristály, “Nash-type equilibria on Riemannian manifolds: a variational approach,” *J. Math. Pures Appl. (9)*, vol. 101, no. 5, pp. 660–688, 2014.
- [15] Q. Liu and D. Wang, “Stein variational gradient descent: A general purpose Bayesian inference algorithm,” in *Advances in Neural Information Processing Systems (NeurIPS)*, Barcelona, Spain, Dec 5–10 2016.
- [16] Q. Liu, J. D. Lee, and M. I. Jordan, “A kernelized Stein discrepancy for goodness-of-fit tests and model evaluation,” in *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, New York City, NY, Jun 16–24 2016.
- [17] F. Facchinei, A. Fischer, and V. Piccialli, “On generalized Nash games and variational inequalities,” *Operations Research Letters*, vol. 35, no. 2, pp. 159–164, 2007.
- [18] C. Laidlaw and A. Dragan, “The Boltzmann policy distribution: Accounting for systematic suboptimality in human models,” in *International Conference on Learning Representations*, virtual, Apr 25–29 2022.
- [19] P. D. Moral, *Mean Field Simulation for Monte Carlo Integration*. CRC Press, 2013.