# Diffusion Learning Over Adaptive Competing Networks

Yike Zhao
EPFL
Switzerland

Haoyuan Cai
EPFL
Switzerland

Ali H. Sayed
EPFL
Switzerland

***Abstract*—In this paper, we study a dynamic game between two networks. The networks compete by optimizing two coupled objective functions. Agents within the same network work toward a common goal and are regarded as *cooperative agents*; they exchange their strategies via links with other agents. Additionally, in the assumed model, each agent receives information from some adversary agents following a bipartite cross-network topology. The networks employ a diffusion learning strategy that allows them to learn and pursue the equilibrium state adaptively. We show that the networks converge to the Nash equilibrium in the mean-square-error sense under some reasonable assumptions.**

***Index Terms*—Competing networks, diffusion learning, Nash equilibrium, adaptive learning**

## I. Introduction

Adaptive networks with cooperative agents have a wide range of applicability in distributed optimization and learning problems [1]. However, networks need not be restricted to cooperative scenarios; many real-world applications, such as those in economics [2], multi-GANs [3], and e-sports games [4], are more naturally characterized by competitive network dynamics. Despite these applications, adaptive networks in competing settings have been underexplored compared to their cooperative counterparts. Inspired by this gap, this paper explores a scenario where two adaptive networks compete.

Distributed optimization problems commonly employ a *cooperative* setup, where agents collaborate to achieve a common objective. A vast body of research has focused on this scenario; see, for instance, [5]–[8]. The works [5], [6] establish performance bounds for stochastic gradient algorithms for solving single task problems over graphs, while the work [8] addresses a game problem. All these references, however, focus on solving their formulations through agent cooperation, excluding the possibility that agents may take on different roles.

A graph setting beyond cooperation is studied in [9]–[14]. The work [9] considers a distributed game over the graph where each agent interacts in the game by minimizing its private cost. The work [10] adopts a variational inequality framework to study the network game problem and establishes convergence results for best response dynamics. The work [11] employs a decaying step size strategy to learn the Nash equilibrium in a two-network zero-sum scenario; however, this strategy lacks tracking capabilities. The work [12] proposes an incremental strategy for solving the network competing problem, and its convergence is guaranteed when the stringent assumption of bounded subgradient is satisfied. The work [13] considers a bipartite network topology for cross-network information sharing and carries out convergence analysis in the continuous-time domain. The work [14] considers a two-team competing problem similar to our work; however, the convergence argument is missing.

The contributions of this work are summarized as follows: 1) We establish a convergence guarantee for diffusion learning under a bipartite cross-network setup and without the assumption of bounded gradients [11]–[14]; 2) we provide a variational inequality perspective to analyze the stability performance of the adaptive competing algorithm, enabling it to cover more general scenarios, including nonzero-sum game formulations; and 3) our theoretical results are established in the discrete-time domain, aligning with the nature of iterative algorithms and distinguishing them from the arguments presented in the work [13].

## II. Problem Formulation

### A. Two-Network Game Setting

We consider a collection of $K$ agents, decomposed into two "teams" $\mathcal{N}^{(1)}$ and $\mathcal{N}^{(2)}$ of size $K_1$ and $K_2$, respectively, with index sets $\mathcal{N}^{(1)} = \{1, \cdots, K_1\}$ and $\mathcal{N}^{(2)} = \{K_1+1, \cdots, K\}$, where $K = K_1 + K_2$. Teams 1 and 2 employ the strategies $x \in \mathbb{R}^{M_1}, y \in \mathbb{R}^{M_2}$, respectively, with $M_1 + M_2 = M$. Each team seeks a strategy to minimize its own objective function as follows:

$$\min_x J^{(1)}(x,y), \qquad \min_y J^{(2)}(x,y) \tag{1}$$

where we use superscripts (1) and (2) to denote the team index. Note that the objective for each team depends on the strategy of the other team. Note further that when $J^{(1)}(x,y) = -J^{(2)}(x,y)$, the problem reduces to a min-max problem. Solutions for this type of problems is typically pursued by applying the gradient descent strategy to both problems in (1) — see, e.g., [15]. In the network setting, the cost function for each network is usually expressed as the weighted sum of local risk functions:

$$J^{(1)}(x,y) \triangleq \sum_{k\in\mathcal{N}^{(1)}} p_k J_k(x,y) \tag{2}$$

$$J^{(2)}(x,y) \triangleq \sum_{k\in\mathcal{N}^{(2)}} p_k J_k(x,y) \tag{3}$$

Emails: yike.zhao@epfl.ch, haoyuan.cai@epfl.ch, ali.sayed@epfl.ch

where each $J_k(x, y)$ is the expected value of some loss function:

$$J_k(x, y) \triangleq \mathbb{E} Q_k(x, y; \boldsymbol{\xi}_k) \tag{4}$$

Here, the $\{p_k\}$ are positive coefficients that add up to 1 over each network, i.e., $\sum_{k \in \mathcal{N}^{(t)}} p_k = 1, t \in \{1, 2\}$. The variable $\boldsymbol{\xi}_k$ represents the random data (or observations). For convenience, we introduce the following concatenated vectors:

$$z \triangleq \begin{bmatrix} x \\ y \end{bmatrix} \in \mathbb{R}^M, \quad F(z) \triangleq \begin{bmatrix} \nabla_x J^{(1)}(x, y) \\ \nabla_y J^{(2)}(x, y) \end{bmatrix} \in \mathbb{R}^M \tag{5}$$

To study the problem, we use the following assumption for the gradient operator.

**Assumption 1** (**$\nu$-strong monotonicity**). *For every $z_1, z_2 \in \mathbb{R}^M$, we have*

$$\langle F(z_1) - F(z_2), z_1 - z_2 \rangle \geq \nu \|z_1 - z_2\|^2 \tag{6}$$

□

Assumption 1 is commonly used in the context of variational functions [16], [17]. It is important to note that condition (6) generalizes the traditional setup of strongly-convex strongly-concave problems studied in the context of min-max optimization. Let the Nash equilibrium $z^\star$ be defined as:

$$z^\star \triangleq \begin{bmatrix} x^\star \\ y^\star \end{bmatrix} \in \mathbb{R}^M \tag{7}$$

where $z^\star$ satisfies

$$J^{(1)}(x^\star, y^\star) \leq J^{(1)}(x, y^\star), \; J^{(2)}(x^\star, y^\star) \leq J^{(2)}(x^\star, y) \tag{8}$$

This condition ensures that no team can unilaterally improve their objective by changing their strategy. Based on this definition, we have the following lemma (proofs are omitted due to brevity).

**Lemma 1** (Nash equilibrium)**.** *Under Assumption 1, there exists a unique Nash equilibrium $z^\star$.*

### *B. Network Formulation*

For each team, agents are connected according to combination matrices $A^{(1)} = [a_{lk}^{(1)}] \in \mathbb{R}^{K_1 \times K_1}$ and $A^{(2)} = [a_{lk}^{(2)}] \in \mathbb{R}^{K_2 \times K_2}$, respectively. Additionally, each agent is informed about the strategy of the adversary network by receiving information directly from one or more adversary agents following a bipartite cross-network topology. We use $C^{(12)} \in \mathbb{R}^{K_1 \times K_2}$ and $C^{(21)} \in \mathbb{R}^{K_2 \times K_1}$ to denote the information flowing from Team 1 to Team 2, and from Team 2 to Team 1, respectively. Also, let

$$C \triangleq \begin{bmatrix} 0 & C^{(12)} \\ C^{(21)} & 0 \end{bmatrix} \in \mathbb{R}^{K \times K} \tag{9}$$

We assume the combination matrices $A^{(1)}, A^{(2)}, C^{(12)}, C^{(21)}$ satisfy the following condition.

**Assumption 2** (**Connectivity**)**.** *For $t, t' \in \{1, 2\}, t \neq t'$, the following conditions hold:*

1) *The combination matrix $A^{(t)} \in \mathbb{R}^{K_t \times K_t}$ is primitive and left-stochastic, i.e. $\mathbb{1}^\top A^{(t)} = \mathbb{1}^\top$.*
2) *The matrix $C^{(t't)} \in \mathbb{R}^{K_{t'} \times K_t}$ is left-stochastic, and for each $k \in \mathcal{N}^{(t)}$, there exists at least one $\ell \in \mathcal{N}^{(t')}$ such that $c_{\ell k} > 0$. This condition guarantees that the cross-team topology forms a bipartite graph without isolated nodes.*

□

The second condition is also adopted in two-network zero-sum games [11], [13], ensuring agents in each team connect directly with some agents in the adversary team. The first condition ensures that $A^{(1)}$ and $A^{(2)}$ have Perron eigenvectors $p^{(1)}$ and $p^{(2)}$, respectively, with entries summing to 1 and satisfying $A^{(1)} p^{(1)} = p^{(1)}$ and $A^{(2)} p^{(2)} = p^{(2)}$ [18]. Define

$$p \triangleq \begin{bmatrix} p^{(1)} \\ p^{(2)} \end{bmatrix} \tag{10}$$

where the $\{p_k\}_{k=1}^{K_1}$ and $\{p_k\}_{k=K_1+1}^{K}$ correspond to the coefficients used in (2) and (3), respectively.

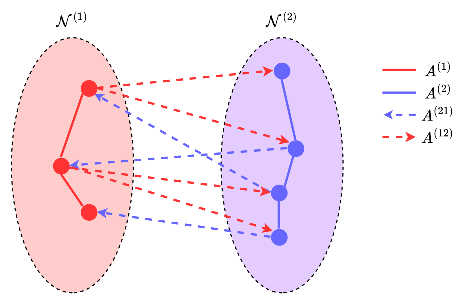


Fig. 1. Within-team and cross-team topologies for the two competing networks $\mathcal{N}^{(1)}, \mathcal{N}^{(2)}$.

## III. Algorithm Development

We describe the diffusion learning algorithm at a single node level. The algorithm is developed in a similar spirit to the Adapt-then-Combine (ATC) strategy for within-team cooperation described in [18]. The difference, however, is the additional step for cross-network information fusion, as each agent requires information from the other team to make decisions. Intuitively, each agent should make decisions based on the latest actions of the opposing network, rather than outdated ones. Therefore, we must first update the information about the adversary team's strategy for an agent; here we consider an agent in Team 1 receiving information from Team 2 for simplicity:

$$\boldsymbol{y}_{k,i-1}^1 = \sum_{\ell \in \mathcal{N}^{(2)}} c_{\ell k} \boldsymbol{y}_{\ell,i-1}^2 \tag{11}$$

where $k \in \mathcal{N}^{(1)}$ belongs to network 1. Therefore, $\boldsymbol{y}_{k,i-1}^1$ denotes the strategy of the adversary team transmitted to agent $k$. Using the latest information $\boldsymbol{y}_{k,i-1}^1$ received from the adversary team, we adopt an ATC strategy to update agent $k$'s strategy in Team 1:

$$\boldsymbol{\phi}_{k,i} = \boldsymbol{x}_{k,i-1}^1 - \mu \widehat{\nabla_x} J_k(\boldsymbol{x}_{k,i-1}^1, \boldsymbol{y}_{k,i-1}^1) \tag{12}$$

$$\boldsymbol{x}_{k,i}^1 = \sum_{\ell \in \mathcal{N}^{(1)}} a_{\ell k}^{(1)} \boldsymbol{\phi}_{\ell,i} \tag{13}$$

where $\widehat{\nabla_x} J_k(\cdot,\cdot)$ denotes a stochastic gradient approximation at agent $k$. Team 2 proceeds in a similar manner.

**Algorithm 1:** Diffusion learning for competing networks (network-level)

```
Initialize: i=0, actions 𝓧¹₋₁, 𝓨²₋₁ ← 0, step size μ
while not done do
    Cross-team learning
    𝓨¹_{i-1} = 𝒞^{(21)⊤} 𝓨²_{i-1}
    𝓧²_{i-1} = 𝒞^{(12)⊤} 𝓧¹_{i-1}
    Within-team adaptation and combination
    𝓧¹_i = 𝒜^{(1)⊤} (𝓧¹_{i-1} − μ𝓖_{x,i})
    𝓨²_i = 𝒜^{(2)⊤} (𝓨²_{i-1} − μ𝓖_{y,i})
    i ⟵ i + 1
end
```

For convenience, we introduce the following notation for the network variables:

$$\boldsymbol{\mathcal{X}}_i^1 \triangleq \text{col}\{\boldsymbol{x}_{1,i}^1, \dots, \boldsymbol{x}_{K_1,i}^1\} \in \mathbb{R}^{K_1 M_1 \times 1} \tag{14}$$

$$\boldsymbol{\mathcal{Y}}_i^1 \triangleq \text{col}\{\boldsymbol{y}_{1,i}^1, \dots, \boldsymbol{y}_{K_1,i}^1\} \in \mathbb{R}^{K_1 M_2 \times 1} \tag{15}$$

Similarly, $\boldsymbol{\mathcal{Y}}_i^2, \boldsymbol{\mathcal{X}}_i^2$ are defined for Team 2. In the above notation, $\boldsymbol{\mathcal{X}}_i^1$ and $\boldsymbol{\mathcal{Y}}_i^2$ correspond to the actions of the same team, while $\boldsymbol{\mathcal{X}}_i^2, \boldsymbol{\mathcal{Y}}_i^1$ correspond to the adversary information received by the agents. At iteration $i$, Team 1 holds the quantities $\boldsymbol{\mathcal{X}}_i^1, \boldsymbol{\mathcal{Y}}_{i-1}^1$, Team 2 holds the quantities $\boldsymbol{\mathcal{Y}}_i^2, \boldsymbol{\mathcal{X}}_{i-1}^2$. In addition, we define the network gradient terms:

$$\boldsymbol{\mathcal{G}}_{x,i} \triangleq \text{col}\{\widehat{\nabla_x} J_k(\boldsymbol{x}_{k,i-1}^1, \boldsymbol{y}_{k,i-1}^1)\}_{k=1}^{K_1} \in \mathbb{R}^{K_1 M_1 \times 1} \tag{16}$$

$$\boldsymbol{\mathcal{G}}_{y,i} \triangleq \text{col}\{\widehat{\nabla_y} J_k(\boldsymbol{x}_{k,i-1}^2, \boldsymbol{y}_{k,i-1}^2)\}_{k=K_1+1}^{K} \in \mathbb{R}^{K_2 M_2 \times 1} \tag{17}$$

The network combination matrices are defined as follows:

$$\begin{aligned} \mathcal{C}^{(21)} &= C^{(21)} \otimes I_{M_2}, \qquad & \mathcal{C}^{(12)} &= C^{(12)} \otimes I_{M_1} \\ \mathcal{A}^{(1)} &= A^{(1)} \otimes I_{M_1}, \qquad & \mathcal{A}^{(2)} &= A^{(2)} \otimes I_{M_2} \end{aligned} \tag{18}$$

where $\otimes$ denotes the Kronecker product operator. With this notation, we summarize the aforementioned learning process in **Algorithm 1**.

To study the behavior of the network centroids, we introduce the following notation:

$$\boldsymbol{x}_{c,i} \triangleq \sum_{k \in \mathcal{N}^{(1)}} p_k \boldsymbol{x}_{k,i}^1, \tag{19}$$

$$\boldsymbol{\mathcal{X}}_{c,i} \triangleq \mathbb{1}_{K_1} \otimes \boldsymbol{x}_{c,i}, \quad \boldsymbol{\mathcal{X}}'_{c,i} \triangleq \mathbb{1}_{K_2} \otimes \boldsymbol{x}_{c,i} \tag{20}$$

The quantities $\boldsymbol{y}_{c,i}, \boldsymbol{\mathcal{Y}}_{c,i}, \boldsymbol{\mathcal{Y}}'_{c,i}$ are defined similarly. According to **Algorithm 1**, we obtain the following recursion for $\boldsymbol{\mathcal{X}}_{c,i}$:

$$\begin{aligned} \boldsymbol{\mathcal{X}}_{c,i} &= \mathbb{1}_{K_1} \otimes \boldsymbol{x}_{c,i} = \left(\mathbb{1}_{K_1} p^{(1)^\top} \otimes I_{M_1}\right) \boldsymbol{\mathcal{X}}_i^1 \\ &= \left(\mathbb{1}_{K_1} p^{(1)^\top} \otimes I_{M_1}\right) \mathcal{A}^{(1)^\top} \left(\boldsymbol{\mathcal{X}}_{i-1}^1 - \mu \boldsymbol{\mathcal{G}}_{x,i}\right) \\ &= \left(\mathbb{1}_{K_1} p^{(1)^\top} \otimes I_{M_1}\right) \left(\boldsymbol{\mathcal{X}}_{i-1}^1 - \mu \boldsymbol{\mathcal{G}}_{x,i}\right) \\ &= \boldsymbol{\mathcal{X}}_{c,i-1} - \mu \left(\mathbb{1}_{K_1} p^{(1)^\top} \otimes I_{M_1}\right) \boldsymbol{\mathcal{G}}_{x,i} \end{aligned} \tag{21}$$

which translates into the following recursion for $\boldsymbol{x}_{c,i}$:

$$\boldsymbol{x}_{c,i} = \boldsymbol{x}_{c,i-1} - \mu \sum_{k \in \mathcal{N}^{(1)}} p_k \widehat{\nabla_x} J_k(\boldsymbol{x}_{k,i-1}^1, \boldsymbol{y}_{k,i-1}^1) \tag{22}$$

In the next section, we show that the network centroids of the two networks asymptotically converge to the Nash equilibrium in the mean-square-error sense.

## IV. Convergence Analysis

To carry out the analysis, we use the following assumptions.

### A. Assumptions

**Assumption 3** (Lipschitz gradients)**.** *For each $t \in \{1,2\}$ and $k \in \mathcal{N}^{(t)}$, we assume the gradients associated with each local risk function $J_k(\cdot,\cdot)$ are $L_f - Lipschitz$, i.e, for any $x_1, x_2 \in \mathbb{R}^{M_1}, y_1, y_2 \in \mathbb{R}^{M_2}$:*

$$\begin{aligned} &\|\nabla_w J_k(x_1, y_1) - \nabla_w J_k(x_2, y_2)\| \\ &\le L_f \left(\|x_1 - x_2\| + \|y_1 - y_2\|\right), \end{aligned} \tag{23}$$

*where $w = x$ or $y$.* □

**Assumption 4** (Bounded gradient disagreement)**.** *For each $t \in \{1,2\}$ and $k \in \mathcal{N}^{(t)}$, the gradient disagreement between the local risk functions and the global risk function is bounded, i.e., for any $x_1 \in \mathbb{R}^{M_1}, y_1 \in \mathbb{R}^{M_2}$:*

$$\|\nabla_w J_k(x_1, y_1) - \nabla_w J^{(t)}(x_1, y_1)\| \le G \tag{24}$$

*where $w = x$ or $y$.* □

**Assumption 5** (Gradient noise process)**.** *We define the filtration generated by the random processes as $\boldsymbol{\mathcal{F}}_i = \{(\boldsymbol{x}_{k,j}^t, \boldsymbol{y}_{k,j}^t) \mid t = \{1,2\}, k \in \mathcal{N}^{(t)}, j = -1, \dots, i\}$. For each $t \in \{1,2\}$ and $k \in \mathcal{N}^{(t)}$, we assume the stochastic gradients are unbiased with bounded variance conditioned on $\boldsymbol{\mathcal{F}}_i$, i.e., for any $\boldsymbol{x}, \boldsymbol{y} \in \boldsymbol{\mathcal{F}}_i$,*

$$\mathbb{E}\{\widehat{\nabla_w} J_k(\boldsymbol{x}, \boldsymbol{y}) \mid \boldsymbol{\mathcal{F}}_i\} = \nabla_w J_k(\boldsymbol{x}, \boldsymbol{y}) \tag{25}$$

$$\mathbb{E}\{\|\widehat{\nabla_w} J_k(\boldsymbol{x}, \boldsymbol{y}) - \nabla_w J_k(\boldsymbol{x}, \boldsymbol{y})\|^2 \mid \boldsymbol{\mathcal{F}}_i\} \le \sigma^2 \tag{26}$$

*where $w = x$ or $y$.* □

### B. Main Results

**Lemma 2** (Within-team consensus)**.** *Under Assumptions 2-5, the iterates of teams 1 and 2, namely $\boldsymbol{\mathcal{X}}_i^1$ and $\boldsymbol{\mathcal{Y}}_i^2$ cluster around their respective team centroids $\boldsymbol{\mathcal{X}}_{c,i}$ and $\boldsymbol{\mathcal{Y}}_{c,i}$ when the step size $\mu$ is sufficiently small and after sufficient iterations $i \ge i_o$. Specifically,*

$$\mathbb{E}\{\|\boldsymbol{\mathcal{X}}_i^1 - \boldsymbol{\mathcal{X}}_{c,i}\|^2 + \|\boldsymbol{\mathcal{Y}}_i^2 - \boldsymbol{\mathcal{Y}}_{c,i}\|^2\} \le O\left(\mu^2\right) \tag{27}$$

*where*

$$i_o = \frac{\log\left(O\left(\mu^2\right)\right)}{\log\left(\alpha\right)} \tag{28}$$

*and $\alpha < 1$ is a constant depending on $A^{(1)}$ and $A^{(2)}$.*

**Lemma 3** (Cross-team consensus)**.** *Under Assumptions 2-5, the adversary strategies $\boldsymbol{\mathcal{X}}_i^2$ known to team 2 and $\boldsymbol{\mathcal{Y}}_i^1$ known to team 1 cluster around the centroids of teams 1 and 2, namely,*

$\boldsymbol{\mathcal{X}}'_{c,i}$ and $\boldsymbol{\mathcal{Y}}'_{c,i}$, *respectively, when the step size* $\mu$ *is sufficiently small and after sufficient iterations* $i \geq i_o$, *i.e.,*

$$\mathbb{E}\{\|\boldsymbol{\mathcal{X}}_i^2 - \boldsymbol{\mathcal{X}}'_{c,i}\|^2 + \|\boldsymbol{\mathcal{Y}}_i^1 - \boldsymbol{\mathcal{Y}}'_{c,i}\|^2\} \leq O\left(\mu^2\right) \tag{29}$$

*Proof.* Using the recursions from Algorithm 1 and the property of $C^{(12)}$ stated in Assumption 2, we obtain the relationship

$$\boldsymbol{\mathcal{X}}_i^2 - \boldsymbol{\mathcal{X}}'_{c,i} = \mathcal{C}^{(12)\top}\left(\boldsymbol{\mathcal{X}}_i^1 - \boldsymbol{\mathcal{X}}_{c,i}\right) \tag{30}$$

Then, we have

$$\mathbb{E}\left\{\|\boldsymbol{\mathcal{X}}_i^2 - \boldsymbol{\mathcal{X}}'_{c,i}\|^2\right\} \leq \left\|\mathcal{C}^{(12)\top}\right\|^2 \mathbb{E}\left\{\|\boldsymbol{\mathcal{X}}_i^1 - \boldsymbol{\mathcal{X}}_{c,i}\|^2\right\} \tag{31}$$

where

$$\begin{aligned}\left\|\mathcal{C}^{(12)\top}\right\|^2 &= \rho\left(C^{(12)}C^{(12)\top}\right) \leq \left\|C^{(12)}C^{(12)\top}\right\|_1 \\ &\leq \mathbb{1}_{K_1}^\top C^{(12)}C^{(12)\top}\mathbb{1}_{K_1} = K_2\end{aligned} \tag{32}$$

A similar result holds for $\mathbb{E}\{\|\boldsymbol{\mathcal{Y}}_i^1 - \boldsymbol{\mathcal{Y}}'_{c,i}\|^2\}$. Using the result from Lemma 2 we complete the proof. ☐

For $\boldsymbol{z}_{c,i}$, we have the following lemma.

**Lemma 4** (Learning dynamics)**.** *Under Assumptions 2-5, the block centroid* $\boldsymbol{z}_{c,i}$ *generated by Algorithm 1 follows the following dynamics when the step size* $\mu$ *is sufficiently small and after sufficient iterations* $i \geq i_o$*:*

$$\boldsymbol{z}_{c,i} = \boldsymbol{z}_{c,i-1} - \mu F(\boldsymbol{z}_{c-1,i}) + \boldsymbol{D}_{c,i} \tag{33}$$

*where* $\mathbb{E}\{\|\boldsymbol{D}_{c,i}\|^2\} \leq O(\mu^2)$.

*Proof.* The result follows directly from Lemmas 2 and 3 along with Assumption 3. ☐

**Theorem 1** (Mean-square-error stability)**.** *Under Assumptions 1-5, the centroid* $\boldsymbol{z}_{c,i}$ *converges to the Nash equilibrium* $z^\star$ *asymptotically in the mean-square-error sense for sufficiently small step size* $\mu$*:*

$$\limsup_{i\to\infty} \mathbb{E}\{\|\boldsymbol{z}_{c,i} - z^\star\|^2\} \leq O(\mu) \tag{34}$$

*Proof.* The proof can be established using the argument from Lemmas 2– 4 and the properties of $\boldsymbol{D}_{c,i}$. ☐

## V. Numerical Results

We consider a regularized stochastic bilinear game similar to [19] to illustrate the performance of the proposed algorithm. In this example, two teams aim to optimize the following objectives:

$$\begin{aligned}\min_x \quad J^{(1)}(x,y) &= \sum_{l\in\mathcal{N}^{(1)}} p_l J_l(x,y) \\ \min_y \quad J^{(2)}(x,y) &= \sum_{r\in\mathcal{N}^{(2)}} p_r J_r(x,y)\end{aligned} \tag{35}$$

where local costs are defined as:

$$\begin{aligned}J_l(x,y) \triangleq \mathbb{E}_{\boldsymbol{\xi}_l}[&x^\top B\left(\boldsymbol{\xi}_l\right) y + \|x\|^2 - \|y\|^2 \\ &+ x^\top g_x(\boldsymbol{\xi}_l) + g_y^\top(\boldsymbol{\xi}_l) y]\end{aligned} \tag{36a}$$

$$\begin{aligned}J_r(x,y) \triangleq -\mathbb{E}_{\boldsymbol{\xi}_r}[&x^\top B\left(\boldsymbol{\xi}_r\right) y + \|x\|^2 - \|y\|^2 \\ &+ x^\top g_x(\boldsymbol{\xi}_r) + g_y^\top(\boldsymbol{\xi}_r) y]\end{aligned} \tag{36b}$$

Here, for $l \in \mathcal{N}^{(1)}$, $B\left(\boldsymbol{\xi}_l\right), g_x(\boldsymbol{\xi}_l), g_y(\boldsymbol{\xi}_l)$ are random quantities depending on the random samples $\boldsymbol{\xi}_l$, and $B\left(\boldsymbol{\xi}_l\right)$ is generated according to the following model:

$$\begin{aligned}&B\left(\boldsymbol{\xi}_l\right) \triangleq B_l + N\left(\boldsymbol{\xi}_l\right) \in \mathbb{R}^{M\times M} \\ &[N\left(\boldsymbol{\xi}_l\right)]_{ij} \sim \mathcal{N}\left(m_{B_l}, \sigma_{B_l}^2\right)\end{aligned} \tag{37}$$

where $B_l$ is a constant matrix, $g_x(\boldsymbol{\xi}_l) \in \mathbb{R}^M$ is a Gaussian random vector with mean $m_{x_l}\mathbb{1}_M$ and covariance $\sigma_{x_l}^2 I_M$, and $g_y(\boldsymbol{\xi}_l)$ follows the same model. All random variables are spatially and temporally independent from each other. We consider a similar model for $r \in \mathcal{N}^{(2)}$.

### A. Experiment Setting and Simulation Results

We consider the following setup for agents $l$ in Team 1, with the same applied to Team 2: Mean values are set to zero, i.e., $m_{B_l} = m_{x_l} = m_{y_l} = 0$, and variances are set to $\sigma_{B_l}^2 = \sigma_{x_l}^2 = \sigma_{y_l}^2 = 1e-4$. The constant matrices are defined as $B_l = I_M$, with $M = 20$. Team 1 consists of $K_1 = 6$ agents while team 2 has $K_2 = 4$ agents. Two strongly-connected networks are generated for the teams using the averaging rule, and a bipartite structure without isolated nodes is implemented for cross-team connections. We run **Algorithm 1** based on this setting.

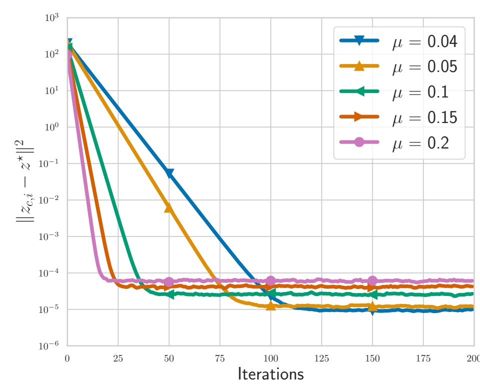


Fig. 2. Performance of **Algorithm 1** for solving regularized stochastic bilinear game

Figure 2 illustrates the mean-square error between the network centroids $\boldsymbol{z}_{c,i}$ obtained by **Algorithm 1** and the Nash equilibrium $z^\star$. The results show that our method converges to the neighborhood of the Nash equilibrium with the size proportional to the magnitude of step size.

## VI. Conclusion

In this work, we proposed a diffusion learning algorithm for finding the Nash equilibrium point in the scenario of an adaptive competing network problem. Convergence of the algorithm is guaranteed under some standard assumptions. We further used a regularized bilinear game simulation to illustrate the theoretical findings. We aim to explore weaker assumptions for the cross-network topology and the gradient operator using the variational inequality approach in the future.

# References


[1] A. H. Sayed, "Adaptive networks," *Proceedings of the IEEE*, vol. 102, no. 4, pp. 460–497, 2014.

[2] H. Chade and J. Eeckhout, "Competing teams," *The Review of Economic Studies*, vol. 87, no. 3, pp. 1134–1173, 2020.

[3] Q. Hoang, T. D. Nguyen, T. Le, and D. Phung, "Mgan: Training generative adversarial nets with multiple generators," in *Proc. International Conference on Learning Representations (ICLR)*, 2018, pp. 1–13.

[4] O. Vinyals *et al.*, "Grandmaster level in starcraft ii using multi-agent reinforcement learning," *Nature*, vol. 575, no. 7782, pp. 350–354, 2019.

[5] J. Chen and A. H. Sayed, "Diffusion adaptation strategies for distributed optimization and learning over networks," *IEEE Transactions on Signal Processing*, vol. 60, no. 8, pp. 4289–4305, 2012.

[6] A. H. Sayed, "Diffusion adaptation over networks," in *Academic Press Library in Signal Processing*, vol. 3, pp. 323–453. Elsevier, 2014.

[7] A. H. Sayed, *Inference and Learning from Data*, Cambridge University Press, 2022.

[8] H. Cai, S. A Alghunaim, and A. H. Sayed, "Diffusion stochastic optimization for min-max problems," *arXiv:2401.14585*, 2024.

[9] C. K. Yu, M. van der Schaar, and A. H. Sayed, "Distributed learning for stochastic generalized nash equilibrium problems," *IEEE Transactions on Signal Processing*, vol. 65, no. 15, pp. 3893–3908, 2017.

[10] F. Parise and A. Ozdaglar, "A variational inequality framework for network games: Existence, uniqueness, convergence and sensitivity analysis," *Games and Economic Behavior*, vol. 114, pp. 47–82, 2019.

[11] Y. Lou, Y. Hong, L. Xie, G. Shi, and K. H. Johansson, "Nash equilibrium computation in subnetwork zero-sum games with switching communications," *IEEE Transactions on Automatic Control*, vol. 61, no. 10, pp. 2920–2935, 2015.

[12] C. X. Shi and G. H. Yang, "Nash equilibrium computation in two-network zero-sum games: An incremental algorithm," *Neurocomputing*, vol. 359, pp. 114–121, 2019.

[13] B. Gharesifard and J. Cortés, "Distributed convergence to Nash equilibria in two-network zero-sum games," *Automatica*, vol. 49, no. 6, pp. 1683–1692, 2013.

[14] S. Vlaski and A. H. Sayed, "Competing adaptive networks," in *Proc. IEEE Statistical Signal Processing Workshop (SSP)*, 2021, pp. 71–75.

[15] F. Schäfer and A. Anandkumar, "Competitive gradient descent," in *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, 2019, pp. 1–11.

[16] S. Komlósi, "On the Stampacchia and Minty variational inequalities," *Generalized Convexity and Optimization for Economic and Financial Decisions*, pp. 231–260, 1999.

[17] E. K. Ryu and S. Boyd, "Primer on monotone operator methods," *Appl. comput. math*, vol. 15, no. 1, pp. 3–43, 2016.

[18] A. H. Sayed, "Adaptation, learning, and optimization over networks," *Foundations and Trends® in Machine Learning*, vol. 7, no. 4-5, pp. 311–801, 2014.

[19] H. Cai, S. A. Alghunaim, and A. H. Sayed, "Diffusion optimistic learning for min-max optimization," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2024, pp. 5535–5539.