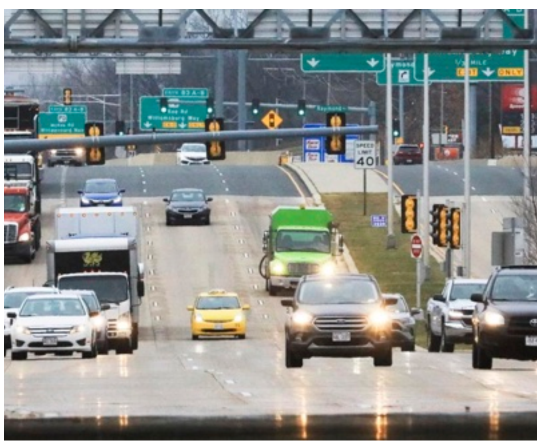
Variations on a Theme:
Information Structure, Equilibria, and Dynamic Games

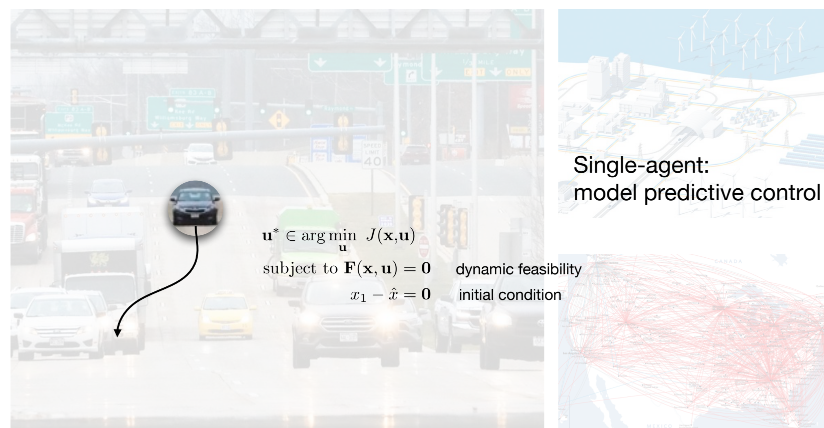
David Fridovich-Keil
Assistant Professor, UT Austin

Game On! Seminar, KTH
February 2026



Single-agent:
model predictive control

Multi-agent:
model predictive gameplay

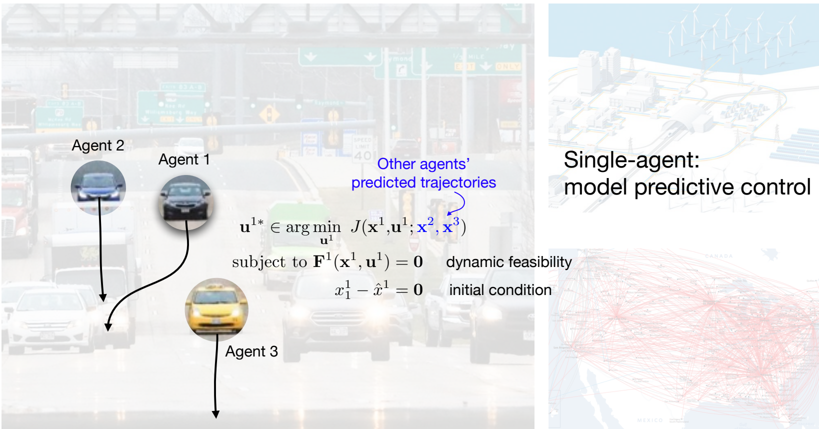


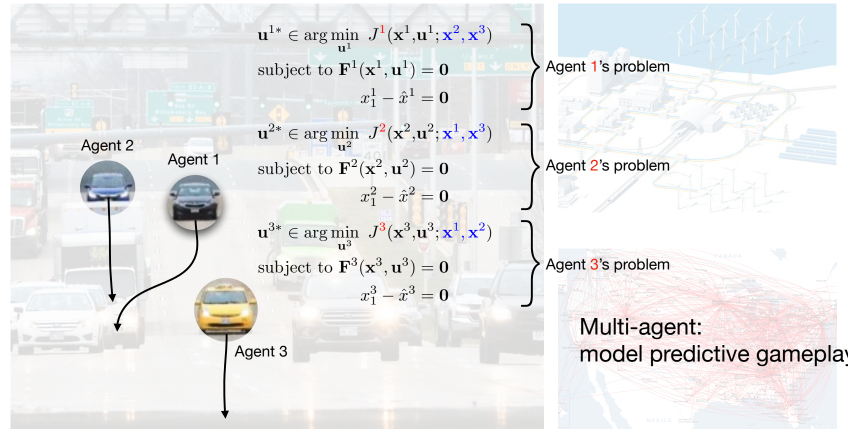
$$\mathbf{u}^* \in \arg\min_{\mathbf{u}} \; J(\mathbf{x},\mathbf{u})$$

$$\text{subject to } \mathbf{F}(\mathbf{x},\mathbf{u}) = \mathbf{0} \quad \text{dynamic feasibility}$$

$$x_1 - \hat{x} = \mathbf{0} \quad \text{initial condition}$$

Single-agent:
model predictive control





Key ingredients of a dynamic game

Objectives

What does each player want?

Player

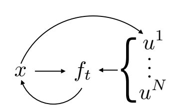
$$J^i(\mathbf{x}, \mathbf{u}^{1:N})$$

Time

$$\text{where } \mathbf{x} := (x_1, \ldots, x_T)$$
$$\mathbf{u}^i := (u_1^i, \ldots, u_T^i)$$

Key ingredients of a dynamic game

Rules



$$x_{t+1} = f_t(x_t, u_t^{1:N})$$

Objectives

What does each player want?

Player

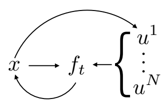
$$J^i(\mathbf{x}, \mathbf{u}^{1:N})$$

Time

$$\text{where } \mathbf{x} := (x_1, \ldots, x_T)$$
$$\mathbf{u}^i := (u_1^i, \ldots, u_T^i)$$

Key ingredients of a dynamic game

Rules



$x_{t+1} = f_t(x_t, u_t^{1:N})$

Objectives

What does each player want?

Player

$J^i(\mathbf{x}, \mathbf{u}^{1:N})$

Time

where $\mathbf{x} := (x_1, \ldots, x_T)$

$\mathbf{u}^i := (u_1^i, \ldots, u_T^i)$

Information

What does each player know?

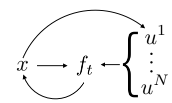
"Strategy"

$u_t^i = \gamma_t^i(?)$

We'll see this more later!

Key ingredients of a dynamic game

Rules



$x_{t+1} = f_t(x_t, u_t^{1:N})$

Objectives

What does each player want?

Player

$J^i(\mathbf{x}, \mathbf{u}^{1:N})$

Time

where $\mathbf{x} := (x_1, \ldots, x_T)$

$\mathbf{u}^i := (u_1^i, \ldots, u_T^i)$

Information

What does each player know?

"Strategy"

$u_t^i = \gamma_t^i(?)$

We'll see this more later!

➡ Looking for a Nash equilibrium where no player can unilaterally improve

The fundamental building blocks: open-loop and feedback solutions to linear-quadratic games

How can we find approximate Nash feedback strategies that minimize inter-agent communication/sensing?

If agents' access to information changes during an interaction, can we still find equilibria efficiently?

Can we extend classical results about the equivalence of open-loop and feedback solutions beyond the linear-quadratic setting?

Linear-quadratic (LQ) games

$$(\text{P}i): \quad \min_{\mathbf{x},\mathbf{u}^i} \quad \frac{1}{2}\sum_{t=1}^{T}\left(x_t^\top Q_t^i x_t + \sum_{j=1}^{N} u_t^{j\top} R_t^{ij} u_t^j\right)$$

$$\text{subject to} \quad x_{t+1} = A_t x_t + \sum_{j=1}^{N} B_t^j u_t^j, \forall t \in \{1, \ldots, T-1\}$$

$$(\mathrm{P}i):\quad \min_{\mathbf{x},\mathbf{u}^i} \quad \frac{1}{2}\sum_{t=1}^{T}\left(x_t^\top Q_t^i x_t + \sum_{j=1}^{N} u_t^{j\top} R_t^{ij} u_t^j\right)$$

$$\text{subject to} \quad x_{t+1} = A_t x_t + \sum_{j=1}^{N} B_t^j u_t^j, \forall t \in \{1,\dots,T-1\}$$

Quadratic cost functions that depend on all agents' actions

Player i

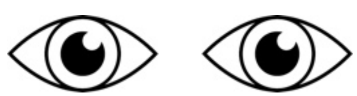
$$(\mathrm{P}i):\quad \min_{\mathbf{x},\mathbf{u}^i} \quad \frac{1}{2}\sum_{t=1}^{T}\left(x_t^\top Q_t^i x_t + \sum_{j=1}^{N} u_t^{j\top} R_t^{ij} u_t^j\right)$$

$$\text{subject to} \quad x_{t+1} = A_t x_t + \sum_{j=1}^{N} B_t^j u_t^j, \forall t \in \{1,\dots,T-1\}$$

Linear state dynamics that also depend on all agents' actions

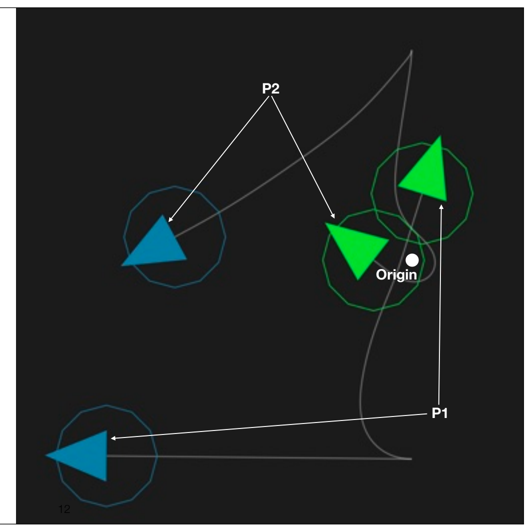
Types of information

Feedback: $u_t^i = \gamma_t^i(x_t)$



Open-loop: $u_t^i = \gamma_t^i(x_1)$

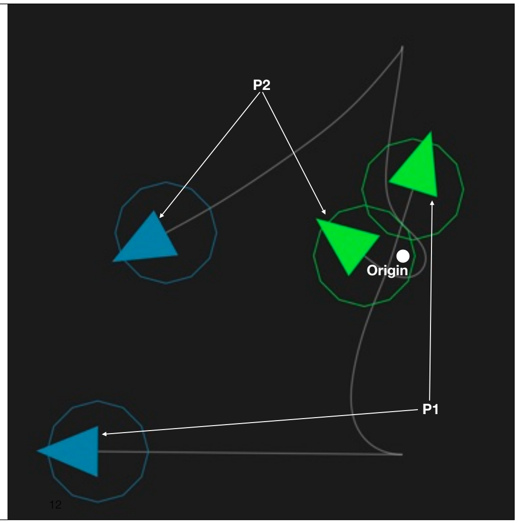
P1 wants P2 => origin
P2 wants => P1
Both want small controls



P1 wants P2 => origin
P2 wants => P1
Both want small controls

Feedback: green

- P1 knows that P2 will see it and go toward it
- So both => origin



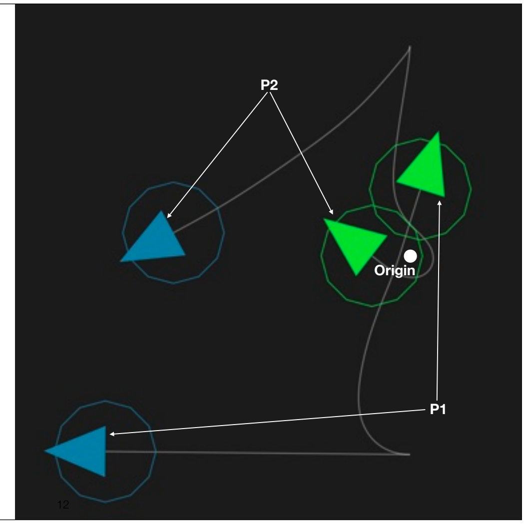
P1 wants P2 => origin
P2 wants => P1
Both want small controls

Feedback: green

- P1 knows that P2 will see it and go toward it
- So both => origin

Open-loop: blue

- P1 knows P2 won't ever see it again after $t = 1$
- So P1 doesn't want to do anything
- But P2 knows this and => P1



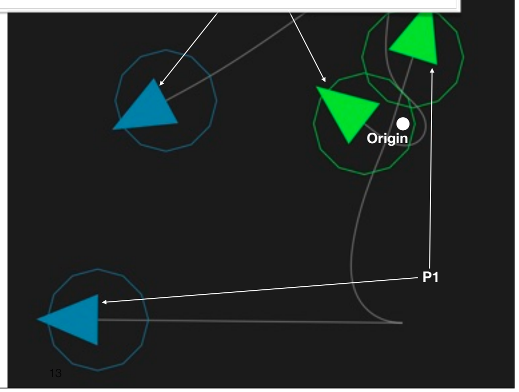
"Closing the loop" in receding horizon is not the answer!

Feedback: green

- P1 knows that P2 will see it and go toward it
- So both => origin

Open-loop: blue

- P1 knows P2 won't ever see it again after $t = 1$
- So P1 doesn't want to do anything
- But P2 knows this and => P1



The open-loop Nash solution

- Open-loop strategy is a time-indexed sequence of control actions
- When *Pi*'s problem is convex, KKT conditions are necessary + sufficient

The open-loop Nash solution

- Open-loop strategy is a time-indexed sequence of control actions
- When *Pi*'s problem is convex, KKT conditions are necessary + sufficient

Pi's Lagrangian

$$\mathcal{L}^i(\mathbf{x},\mathbf{u},\boldsymbol{\lambda}) = \frac{1}{2}\sum_{t=1}^{T}\left(x_t^\top Q_t^i x_t + \sum_{j=1}^{N} u_t^{j\top} R_t^{ij} u_t^j\right) - \sum_{t=1}^{T-1}\lambda_t^{i\top}\left(x_{t+1} - A_t x_t - \sum_{j=1}^{N} B_t^j u_t^j\right)$$

Pi's dynamics
Lagrange multiplier ("costate")

$$\mathcal{L}^i(\mathbf{x},\mathbf{u},\boldsymbol{\lambda}) = \frac{1}{2}\sum_{t=1}^{T}\left(x_t^\top Q_t^i x_t + \sum_{j=1}^{N} u_t^{j\top} R_t^{ij} u_t^j\right) - \sum_{t=1}^{T-1}\lambda_t^{i\top}\left(x_{t+1} - A_t x_t - \sum_{j=1}^{N} B_t^j u_t^j\right)$$

$$0 = \nabla_{u_t^i}\mathcal{L}^i = R_t^{ii}u_t^i + B_t^{i\top}\lambda_t^i, \qquad \forall t \in \{1,2,\ldots,T-1\}$$
$$\implies u_t^i = -(R_t^{ii})^{-1}B_t^{i\top}\lambda_t^i$$
$$0 = \nabla_{u_T^i}\mathcal{L}^i = R_T^{ii}u_T^i \qquad \implies u_T^i = 0$$
$$0 = \nabla_{x_t}\mathcal{L}^i = Q_t^i x_t - \lambda_{t-1}^i + A_t^\top\lambda_t^i, \qquad \forall t \in \{2,3,\ldots,T-1\}$$
$$\implies \lambda_{t-1}^i = Q_t^i x_t + A_t^\top\lambda_t^i$$
$$0 = \nabla_{x_T}\mathcal{L}^i = Q_T^i x_T - \lambda_{T-1}^i \qquad \implies \lambda_{T-1}^i = Q_T^i x_T$$
$$0 = x_{t+1} - Ax_t - \sum_{j=1}^{N} B_t^j u_t^j, \qquad \forall t \in \{1,2,\ldots,T-1\}$$

$$\mathcal{L}^i(\mathbf{x},\mathbf{u},\boldsymbol{\lambda}) = \frac{1}{2}\sum_{t=1}^{T}\left(x_t^\top Q_t^i x_t + \sum_{j=1}^{N} u_t^{j\top} R_t^{ij} u_t^j\right)$$

👌 This is a linear system of equations!

$$0 = \nabla_{u_t^i}\mathcal{L}^i = R_t^{ii}u_t^i + B_t^{i\top}\lambda_t^i, \qquad \forall t \in \{1,2,\ldots,T-1\}$$
$$\implies u_t^i = -(R_t^{ii})^{-1}B_t^{i\top}\lambda_t^i$$
$$0 = \nabla_{u_T^i}\mathcal{L}^i = R_T^{ii}u_T^i \qquad \implies u_T^i = 0$$
$$0 = \nabla_{x_t}\mathcal{L}^i = Q_t^i x_t - \lambda_{t-1}^i + A_t^\top\lambda_t^i, \qquad \forall t \in \{2,3,\ldots,T-1\}$$
$$\implies \lambda_{t-1}^i = Q_t^i x_t + A_t^\top\lambda_t^i$$
$$0 = \nabla_{x_T}\mathcal{L}^i = Q_T^i x_T - \lambda_{T-1}^i \qquad \implies \lambda_{T-1}^i = Q_T^i x_T$$
$$0 = x_{t+1} - Ax_t - \sum_{j=1}^{N} B_t^j u_t^j, \qquad \forall t \in \{1,2,\ldots,T-1\}$$

The open-loop Nash solution

- A very sparse, highly-structured system of equations
- With some work… a recursive (Riccati-type) solution is

The open-loop Nash solution

- A very sparse, highly-structured system of equations
- With some work… a recursive (Riccati-type) solution is

$$x_{t+1} = \boxed{\Lambda_t^{-1}} A_t x_t$$
$$u_t^i = -(R_t^{ii})^{-1} B_t^{i\top} \boxed{M_{t+1}^i} x_{t+1}$$
$$\lambda_t^i = \boxed{M_{t+1}^i} x_{t+1} \,.$$

$$\boxed{M_t^i} = Q_t^i + A_t^\top M_{t+1}^i \Lambda_t^{-1} A_t, \; M_T^i = Q_T^i$$

$$\boxed{\Lambda_t} = I + \sum_{j=1}^N B_t^j (R_t^{jj})^{-1} B_t^{j\top} M_{t+1}^j$$

What changes in the feedback case?

- Feedback strategies $\{u_t^i = \gamma_t^i(x_t)\}$ must be in equilibrium… for the rest of the game, <u>no matter the initial state or time</u>
- Classically, a dynamic programming solution via optimal cost-to-go

What changes in the feedback case?

- Feedback strategies $\{u_t^i = \gamma_t^i(x_t)\}$ must be in equilibrium… for the rest of the game, <u>no matter the initial state or time</u>
- Classically, a dynamic programming solution via optimal cost-to-go

$$\boxed{(\text{Pi}): \quad V_t^i(x_t)} = \min_{x_{t+1}, u_t^i} \quad \frac{1}{2}\left(x_t^\top Q_t^i x_t + \sum_{j=1}^N u_t^{j\top} R_t^{ij} u_t^j\right) + V_{t+1}^i(x_{t+1})$$

Pi's optimal cost-to-go from state x_t at time t

$$\text{s.t.} \qquad x_{t+1} = A_t x_t + \sum_{j=1}^N B_t^j u_t^j \,,$$

Beginning at time *t* = *T*…

$$V_T^i(x_T) = \min_{x_{T+1}, u_T^i} \quad \frac{1}{2}\left(x_T^\top Q_T^i x_T + \sum_{j=1}^N u_T^{j\top} R_T^{ij} u_T^j\right) + \overbrace{V_{T+1}^i(x_{T+1})}^{0}$$

$$= \frac{1}{2} x_T^\top \underbrace{Q_T^i}_{Z_T^i} x_T \,, \text{ with}$$

$$\gamma_T^{i*}(x_T) = 0 \,.$$

Beginning at time t = T…

$$V_T^i(x_T) = \min_{x_{T+1}, u_T^i} \quad \frac{1}{2}\left(x_T^\top Q_T^i x_T + \sum_{j=1}^N \boxed{u_T^{j\top} R_T^{ij} u_T^j}\right) + \overbrace{V_{T+1}^i(x_{T+1})}^{0}$$

$$= \frac{1}{2} x_T^\top \underbrace{Q_T^i}_{Z_T^i} x_T\,, \text{ with}$$

Assuming Pi's problem is convex!

$$\gamma_T^{i*}(x_T) = 0\,.$$

Beginning at time t = T…

$$V_T^i(x_T) = \min_{x_{T+1}, u_T^i} \quad \frac{1}{2}\left(x_T^\top Q_T^i x_T + \sum_{j=1}^N \boxed{u_T^{j\top} R_T^{ij} u_T^j}\right) + \overbrace{V_{T+1}^i(x_{T+1})}^{0}$$

$$= \frac{1}{2} x_T^\top \underbrace{Q_T^i}_{\boxed{Z_T^i}} x_T\,, \text{ with}$$

Assuming Pi's problem is convex!

$$\gamma_T^{i*}(x_T) = 0\,.$$

We will find that $V_t^i(x_t) = \frac{1}{2} x_t^\top Z_t^i x_t$

Moving back to time t = T - 1…

$$V_{T-1}^i(x_{T-1}) =$$

$$\min_{x_T, u_{T-1}^i} \quad \frac{1}{2}\left(x_{T-1}^\top Q_{T-1}^i x_{T-1} + \sum_{j=1}^N u_{T-1}^{j\top} R_{T-1}^{ij} u_{T-1}^j\right) + \overbrace{\frac{1}{2} x_T^\top Z_T^i x_T}^{V_T^i(x_T)}$$

$$\text{s.t.} \quad x_T = A_{T-1} x_{T-1} + \sum_{j=1}^N B_{T-1}^j u_{T-1}^j\,.$$

$$V_{T-1}^i(x_{T-1}) =$$

$$\min_{x_T, u_{T-1}^i} \quad \frac{1}{2}\left(x_{T-1}^\top Q_{T-1}^i x_{T-1} + \sum_{j=1}^N u_{T-1}^{j\top} R_{T-1}^{ij} u_{T-1}^j\right) + \overbrace{\frac{1}{2} x_T^\top Z_T^i x_T}^{V_T^i(x_T)}$$

$$\text{s.t.} \quad x_T = A_{T-1} x_{T-1} + \sum_{j=1}^N B_{T-1}^j u_{T-1}^j\,.$$

The first-order necessary conditions for *Pi* are…

$$0 = R_{T-1}^{ii} u_{T-1}^{i*} + B_{T-1}^{i\top} Z_T^i \left(A_{T-1} x_{T-1} + \sum_{j=1}^N B_{T-1}^j u_{T-1}^{j*}\right)$$

Jointly solving for all players yields…

$$S_{T-1}\begin{bmatrix}u^{1*}_{T-1}\\ u^{2*}_{T-1}\\ \vdots\\ u^{N*}_{T-1}\end{bmatrix} = -\begin{bmatrix}B^{1\top}_{T-1}Z^1_T A_{T-1}x_{T-1}\\ B^{2\top}_{T-1}Z^2_T A_{T-1}x_{T-1}\\ \vdots\\ B^{N\top}_{T-1}Z^N_T A_{T-1}x_{T-1}\end{bmatrix}$$

$$S_{T-1} = \begin{bmatrix} R^{11}_{T-1}+B^{1\top}_{T-1}Z^1_T B^1_{T-1} & \cdot & \cdots & B^{1\top}_{T-1}Z^1_T B^N_{T-1}\\ B^{2\top}_{T-1}Z^2_T B^1_{T-1} & \cdot & \cdots & \cdot\\ \vdots & \cdot & \ddots & \vdots\\ B^{N\top}_{T-1}Z^N_T B^1_{T-1} & \cdot & \cdots & R^{NN}_{T-1}+B^{N\top}_{T-1}Z^N_T B^N_{T-1}\end{bmatrix}$$

$$S_{T-1}\begin{bmatrix}u^{1*}_{T-1}\\ u^{2*}_{T-1}\\ \vdots\\ u^{N*}_{T-1}\end{bmatrix} = -\begin{bmatrix}B^{1\top}_{T-1}Z^1_T A_{T-1}x_{T-1}\\ B^{2\top}_{T-1}Z^2_T A_{T-1}x_{T-1}\\ \vdots\\ B^{N\top}_{T-1}Z^N_T A_{T-1}x_{T-1}\end{bmatrix}$$

$$\gamma^{i*}_{T-1}(x_{T-1}) = u^{i*}_{T-1} = -P^i_{T-1}x_{T-1}$$

$$S_{T-1}\begin{bmatrix}P^1_{T-1}\\ P^2_{T-1}\\ \vdots\\ P^N_{T-1}\end{bmatrix} = \begin{bmatrix}B^{1\top}_{T-1}Z^1_T A_{T-1}\\ B^{2\top}_{T-1}Z^2_T A_{T-1}\\ \vdots\\ B^{N\top}_{T-1}Z^N_T A_{T-1}\end{bmatrix}$$

$$S_{T-1}\begin{bmatrix}u^{1*}_{T-1}\\ u^{2*}_{T-1}\\ \vdots\\ u^{N*}_{T-1}\end{bmatrix} = -\begin{bmatrix}B^{1\top}_{T-1}Z^1_T A_{T-1}x_{T-1}\\ B^{2\top}_{T-1}Z^2_T A_{T-1}x_{T-1}\\ \vdots\\ B^{N\top}_{T-1}Z^N_T A_{T-1}x_{T-1}\end{bmatrix}$$

Rearranging the system of equations…

$$\gamma^{i*}_{T-1}(x_{T-1}) = u^{i*}_{T-1} = -P^i_{T-1}x_{T-1}$$

$$S_{T-1}\begin{bmatrix}P^1_{T-1}\\ P^2_{T-1}\\ \vdots\\ P^N_{T-1}\end{bmatrix} = \begin{bmatrix}B^{1\top}_{T-1}Z^1_T A_{T-1}\\ B^{2\top}_{T-1}Z^2_T A_{T-1}\\ \vdots\\ B^{N\top}_{T-1}Z^N_T A_{T-1}\end{bmatrix}$$

$$S_{T-1}\begin{bmatrix}u^{1*}_{T-1}\\ u^{2*}_{T-1}\\ \vdots\\ u^{N*}_{T-1}\end{bmatrix} = -\begin{bmatrix}B^{1\top}_{T-1}Z^1_T A_{T-1}x_{T-1}\\ B^{2\top}_{T-1}Z^2_T A_{T-1}x_{T-1}\\ \vdots\\ B^{N\top}_{T-1}Z^N_T A_{T-1}x_{T-1}\end{bmatrix}$$

Rearranging the system of equations…

$$\gamma^{i*}_{T-1}(x_{T-1}) = u^{i*}_{T-1} = -P^i_{T-1}x_{T-1}$$

$$S_{T-1}\begin{bmatrix}P^1_{T-1}\\ P^2_{T-1}\\ \vdots\\ P^N_{T-1}\end{bmatrix} = \begin{bmatrix}B^{1\top}_{T-1}Z^1_T A_{T-1}\\ B^{2\top}_{T-1}Z^2_T A_{T-1}\\ \vdots\\ B^{N\top}_{T-1}Z^N_T A_{T-1}\end{bmatrix}$$

This is the one we solve to determine feedback policies!

$$S_{T-1}\begin{bmatrix}u^{1*}_{T-1}\\ u^{2*}_{T-1}\\ \vdots\\ u^{N*}_{T-1}\end{bmatrix} = -\begin{bmatrix}B^{1\top}_{T-1}Z^1_T A_{T-1}x_{T-1}\\ B^{2\top}_{T-1}Z^2_T A_{T-1}x_{T-1}\\ \vdots\\ B^{N\top}_{T-1}Z^N_T A_{T-1}x_{T-1}\end{bmatrix}$$

Rearranging the system of equations…

$$\gamma^{i*}_{T-1}(x_{T-1}) = u^{i*}_{T-1} = -P^i_{T-1}x_{T-1}$$

$$S_{T-1}\begin{bmatrix}P^1_{T-1}\\ P^2_{T-1}\\ \vdots\\ P^N_{T-1}\end{bmatrix} = \begin{bmatrix}B^{1\top}_{T-1}Z^1_T A_{T-1}\\ B^{2\top}_{T-1}Z^2_T A_{T-1}\\ \vdots\\ B^{N\top}_{T-1}Z^N_T A_{T-1}\end{bmatrix}$$

This is the one we solve to determine feedback policies!

…and finally

$$Z^i_{T-1} = Q^i_{T-1} + \sum_{j=1}^{N} P^{j\top}_{T-1}R^{ij}_{T-1}P^j_{T-1} + \Big(A_{T-1} - \sum_{j=1}^{N} B^j_{T-1}P^j_{T-1}\Big)^\top Z^i_T\Big(A_{T-1} - \sum_{j=1}^{N} B^j_{T-1}P^j_{T-1}\Big)$$

So, what changed? Same linear system?

$$Z^i_{T-1} = Q^i_{T-1} + \sum_{j=1}^{N} P^{j\top}_{T-1}R^{ij}_{T-1}P^j_{T-1} + \Big(A_{T-1} - \sum_{j=1}^{N} B^j_{T-1}P^j_{T-1}\Big)^\top Z^i_T\Big(A_{T-1} - \sum_{j=1}^{N} B^j_{T-1}P^j_{T-1}\Big)$$

So, what changed? Same linear system?

Cross-terms R^{ij}_t were not present in the open-loop solution!

$$Z^i_{T-1} = Q^i_{T-1} + \sum_{j=1}^{N} P^{j\top}_{T-1}R^{ij}_{T-1}P^j_{T-1} + \Big(A_{T-1} - \sum_{j=1}^{N} B^j_{T-1}P^j_{T-1}\Big)^\top Z^i_T\Big(A_{T-1} - \sum_{j=1}^{N} B^j_{T-1}P^j_{T-1}\Big)$$

There are other differences too…

Ex. Polite behavior

Ex. Polite behavior

- Politeness ~ caring about others' effort

Ex. Polite behavior

- Politeness ~ caring about others' effort

Suppose $J^1 = \tilde{J}^1(\mathbf{x}, \mathbf{u}^1) + \|\mathbf{u}^2\|$

Ex. Polite behavior

- Politeness ~ caring about others' effort
- In open-loop, this is impossible!

Suppose $J^1 = \tilde{J}^1(\mathbf{x}, \mathbf{u}^1) + \|\mathbf{u}^2\|$

Ex. Polite behavior

- Politeness ~ caring about others' effort
- In open-loop, this is impossible!

Suppose $J^1 = \tilde{J}^1(\mathbf{x}, \mathbf{u}^1) + \|\mathbf{u}^2\|$

dynamics constraint

$$0 = \nabla_{\{\mathbf{x},\mathbf{u}^1\}} J^1 + \lambda^{1\top} \nabla_{\{\mathbf{x},\mathbf{u}^1\}} \boxed{F(\mathbf{x}, \mathbf{u})}$$
$$= \nabla_{\{\mathbf{x},\mathbf{u}^1\}} \tilde{J}^1 + \cancel{\nabla_{\{\mathbf{x},\mathbf{u}^1\}} \|\mathbf{u}^2\|}$$
$$+ \lambda^{1\top} \nabla_{\{\mathbf{x},\mathbf{u}^1\}} F(\mathbf{x}, \mathbf{u})$$

Ex. Polite behavior

- Politeness ~ caring about others' effort
- In open-loop, this is impossible!
- In feedback, possible because of future state observations

dynamics constraint

$$\text{Suppose } J^1 = \tilde{J}^1(\mathbf{x}, \mathbf{u}^1) + \|\mathbf{u}^2\|$$

$$\begin{aligned} 0 &= \nabla_{\{\mathbf{x},\mathbf{u}^1\}} J^1 + \lambda^{1\top} \nabla_{\{\mathbf{x},\mathbf{u}^1\}} \boxed{F(\mathbf{x}, \mathbf{u})} \\ &= \nabla_{\{\mathbf{x},\mathbf{u}^1\}} \tilde{J}^1 + \cancel{\nabla_{\{\mathbf{x},\mathbf{u}^1\}} \|\mathbf{u}^2\|} \\ &\quad + \lambda^{1\top} \nabla_{\{\mathbf{x},\mathbf{u}^1\}} F(\mathbf{x}, \mathbf{u}) \end{aligned}$$

In feedback, this isn't exactly the first-order condition!

The fundamental building blocks: open-loop and feedback solutions to linear-quadratic games

How can we find approximate Nash feedback strategies that minimize inter-agent communication/sensing?

If agents' access to information changes during an interaction, can we still find equilibria efficiently?

Can we extend classical results about the equivalence of open-loop and feedback solutions beyond the linear-quadratic setting?

Minimizing inter-agent dependencies

$$S_t \begin{bmatrix} P_t^1 \\ P_t^2 \\ \vdots \\ P_t^N \end{bmatrix} = \begin{bmatrix} B_t^{1\top} Z_{t+1}^1 A_t \\ B_t^{2\top} Z_{t+1}^2 A_t \\ \vdots \\ B_t^{N\top} Z_{t+1}^N A_t \end{bmatrix}$$

Minimizing inter-agent dependencies

- Remember, feedback Nash strategies satisfied a linear equation at each *t*

$$S_t \begin{bmatrix} P_t^1 \\ P_t^2 \\ \vdots \\ P_t^N \end{bmatrix} = \begin{bmatrix} B_t^{1\top} Z_{t+1}^1 A_t \\ B_t^{2\top} Z_{t+1}^2 A_t \\ \vdots \\ B_t^{N\top} Z_{t+1}^N A_t \end{bmatrix}$$

Minimizing inter-agent dependencies

- Remember, feedback Nash strategies satisfied a linear equation at each *t*
- Inter-agent dependence <=> block structure of agents' policy matrices *P*

$$S_t \begin{bmatrix} P_t^1 \\ P_t^2 \\ \vdots \\ P_t^N \end{bmatrix} = \begin{bmatrix} B_t^{1\top} Z_{t+1}^1 A_t \\ B_t^{2\top} Z_{t+1}^2 A_t \\ \vdots \\ B_t^{N\top} Z_{t+1}^N A_t \end{bmatrix}$$

Minimizing inter-agent dependencies

- Remember, feedback Nash strategies satisfied a linear equation at each *t*
- Inter-agent dependence <=> block structure of agents' policy matrices *P*

$$S_t \begin{bmatrix} P_t^1 \\ P_t^2 \\ \vdots \\ P_t^N \end{bmatrix} = \begin{bmatrix} B_t^{1\top} Z_{t+1}^1 A_t \\ B_t^{2\top} Z_{t+1}^2 A_t \\ \vdots \\ B_t^{N\top} Z_{t+1}^N A_t \end{bmatrix}$$

$$\gamma_t^{i*}(x_t) = u_t^{i*} = -P_t^i x_t$$

$$P_t^i = \begin{bmatrix} P_t^{i1} & P_t^{i2} & \cdots & P_t^{iN} \end{bmatrix}$$

matrix multiplying P2's state variables

Minimizing inter-agent dependencies

- To reduce inter-agent dependencies…
- penalize the group-L1 norm of *P*

$$P_t^i = \begin{bmatrix} P_t^{i1} & P_t^{i2} & \cdots & P_t^{iN} \end{bmatrix}$$

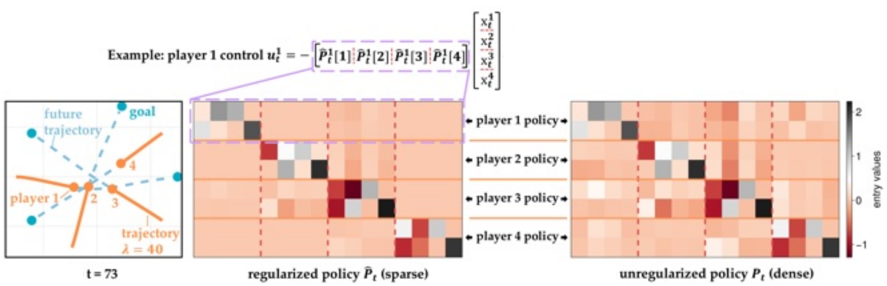
Minimizing inter-agent dependencies

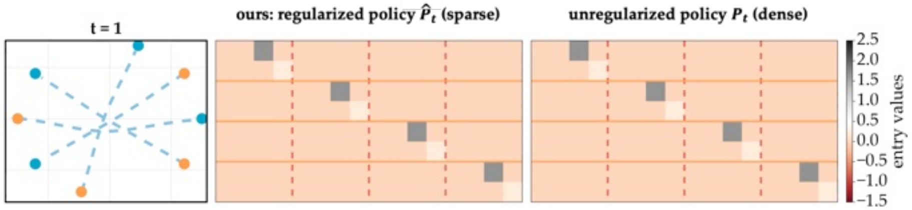
- To reduce inter-agent dependencies…
- penalize the group-L1 norm of *P*

$$P_t^i = \begin{bmatrix} P_t^{i1} & P_t^{i2} & \cdots & P_t^{iN} \end{bmatrix}$$

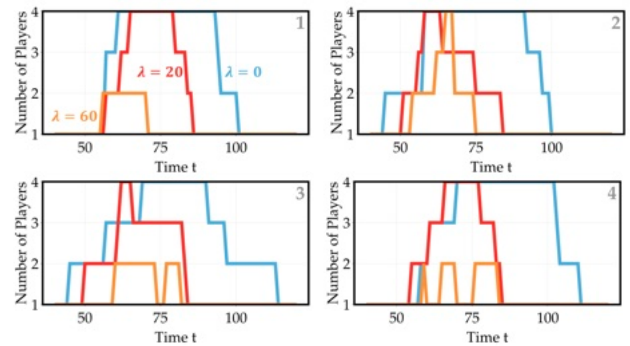
$$\text{minimize } \|S_t P_t - Y_t\|_F^2 + \sum_{i \neq j} \lambda_t^{ij} \|P_t^{ij}\|_F$$

This is a convex program and can be solved very efficiently!





As regularization gets larger, agents' policies get sparser.



Number of nonzero blocks in each player's policy matrix over time

The fundamental building blocks: open-loop and feedback solutions to linear-quadratic games

How can we find approximate Nash feedback strategies that minimize inter-agent communication/sensing?

If agents' access to information changes during an interaction, can we still find equilibria efficiently?

Can we extend classical results about the equivalence of open-loop and feedback solutions beyond the linear-quadratic setting?

In linear-quadratic games…

Both open-loop and feedback Riccati solutions encode a cost-to-go as a quadratic function of state in the neighborhood of the equilibrium trajectory.

In linear-quadratic games…

Both open-loop and feedback Riccati solutions encode a cost-to-go as a quadratic function of state in the neighborhood of the equilibrium trajectory.

open-loop

$$x_{t+1} = \Lambda_t^{-1} A_t x_t$$
$$u_t^i = -(R_t^{ii})^{-1} B_t^{i\top} M_{t+1}^i x_{t+1}$$
$$\lambda_t^i = M_{t+1}^i x_{t+1}\,.$$

$$M_t^i = Q_t^i + A_t^\top M_{t+1}^i \Lambda_t^{-1} A_t,\ M_T^i = Q_T^i$$
$$\Lambda_t = I + \sum_{j=1}^{N} B_t^j (R_t^{jj})^{-1} B_t^{j\top} M_{t+1}^j$$

In linear-quadratic games…

Both open-loop and feedback Riccati solutions encode a cost-to-go as a quadratic function of state in the neighborhood of the equilibrium trajectory.

open-loop

$$x_{t+1} = \Lambda_t^{-1} A_t x_t$$
$$u_t^i = -(R_t^{ii})^{-1} B_t^{i\top} M_{t+1}^i x_{t+1}$$
$$\lambda_t^i = M_{t+1}^i x_{t+1}\,.$$

$$M_t^i = Q_t^i + A_t^\top M_{t+1}^i \Lambda_t^{-1} A_t,\ M_T^i = Q_T^i$$
$$\Lambda_t = I + \sum_{j=1}^{N} B_t^j (R_t^{jj})^{-1} B_t^{j\top} M_{t+1}^j$$

feedback

$$V_t^i(x_t) = \frac{1}{2} x_t^\top Z_t^i x_t$$

$$Z_{T-1}^i = Q_{T-1}^i + \sum_{j=1}^{N} P_{T-1}^{j\top} R_{T-1}^{ij} P_{T-1}^j + \Big(A_{T-1} - \sum_{j=1}^{N} B_{T-1}^j P_{T-1}^j\Big)^\top Z_T^i \Big(A_{T-1} - \sum_{j=1}^{N} B_{T-1}^j P_{T-1}^j\Big)$$

In linear-quadratic games…

Both open-loop and feedback Riccati solutions encode a cost-to-go as a quadratic function of state in the neighborhood of the equilibrium trajectory.

open-loop

$$x_{t+1} = \Lambda_t^{-1} A_t x_t$$
$$u_t^i = -(R_t^{ii})^{-1} B_t^{i\top} \boxed{M_{t+1}^i} x_{t+1}$$
$$\lambda_t^i = \boxed{M_{t+1}^i} x_{t+1}\,.$$

$$\boxed{M_t^i} = Q_t^i + A_t^\top M_{t+1}^i \Lambda_t^{-1} A_t,\ M_T^i = Q_T^i$$
$$\Lambda_t = I + \sum_{j=1}^{N} B_t^j (R_t^{jj})^{-1} B_t^{j\top} M_{t+1}^j$$

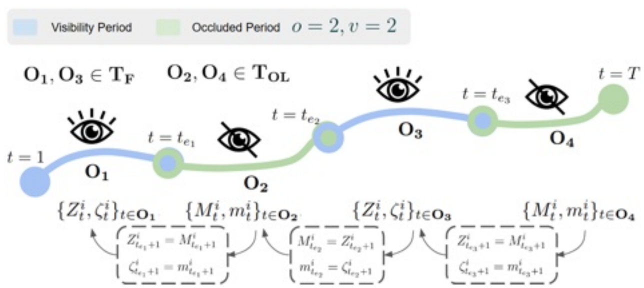
feedback

$$V_t^i(x_t) = \frac{1}{2} x_t^\top \boxed{Z_t^i} x_t$$

$$\boxed{Z_{T-1}^i} = Q_{T-1}^i + \sum_{j=1}^{N} P_{T-1}^{j\top} R_{T-1}^{ij} P_{T-1}^j + \Big(A_{T-1} - \sum_{j=1}^{N} B_{T-1}^j P_{T-1}^j\Big)^\top Z_T^i \Big(A_{T-1} - \sum_{j=1}^{N} B_{T-1}^j P_{T-1}^j\Big)$$

In linear-quadratic games…

…and we can "stitch" these together whenever agents go in or out of view!



The fundamental building blocks: open-loop and feedback solutions to linear-quadratic games

How can we find approximate Nash feedback strategies that minimize inter-agent communication/sensing?

If agents' access to information changes during an interaction, can we still find equilibria efficiently?

Can we extend classical results about the equivalence of open-loop and feedback solutions beyond the linear-quadratic setting?

…actually, everything so far extends beyond LQ.

"The computation of approximate generalized feedback Nash equilibria," SIAM Journal on Optimization, 2023.

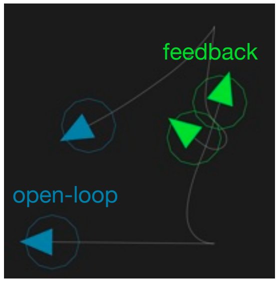
"The computation of approximate generalized feedback Stackelberg equilibria in multiplayer nonlinear constrained dynamic games," SIAM Journal on Optimization, 2024.

In these non-LQ problems…

Open-loop	Feedback
Simpler to compute (in nonconvex settings)	Can encode more "complex" strategies
Players commit to strategy early on	More complicated to compute (in nonconvex settings)
SoTA solvers give *exact* local solutions (with guarantees!)	SoTA solvers give *approximate* first-order local solutions

When to bother with feedback solutions, anyway?

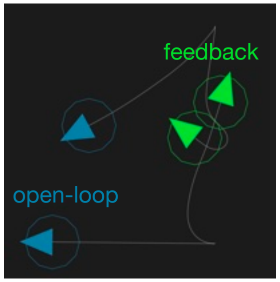
- In general-sum games, solutions can differ
  => so, should compute feedback if possible



When to bother with feedback solutions, anyway?

- In general-sum games, solutions can differ
  => so, should compute feedback if possible

- But, in zero-sum games?



Classical results (LQ)

- For convex-concave zero-sum LQ games, a unique FBNE exists and its open-loop realization is also a (not necessarily unique) OLNE.
- While the existence of a unique OLNE implies the existence of a unique FBNE in a two-agent zero-sum LQ game, the converse is not true.

Classical results (LQ)

- For convex-concave zero-sum LQ games, a unique FBNE exists and its open-loop realization is also a (not necessarily unique) OLNE.
- While the existence of a unique OLNE implies the existence of a unique FBNE in a two-agent zero-sum LQ game, the converse is not true.

Thus, when they both exist, a unique OLNE and a unique FBNE generate the same state trajectory in a two-agent zero-sum LQ game.

Classical results (beyond LQ)

- For a zero-sum dynamic game, if a strongly unique FBNE and an OLNE exist, then the OLNE is unique and generates the same state trajectory as the FBNE.
- On the other hand, if a strongly unique OLNE and a FBNE exist for the game, then the two also give the same state trajectory.

Classical results (beyond LQ)

- For a zero-sum dynamic game, if a strongly unique FBNE and an OLNE exist, then the OLNE is unique and generates the same state trajectory as the FBNE.
- On the other hand, if a strongly unique OLNE and a FBNE exist for the game, then the two also give the same state trajectory.

These assume uniqueness!
In nonconvex settings we never have this…

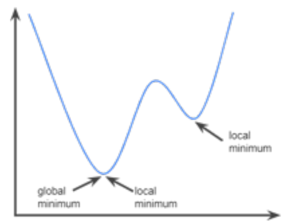
New results for local solutions

First-order necessary condition:

gradient of *Pi*'s Lagrangian wrt
Pi's decision variables is 0

Second-order sufficient condition:

Hessian of *Pi*'s Lagrangian wrt
Pi's decision variables is positive definite
when restricted to *Pi*'s critical cone



Pi's cost at Nash

New results for local solutions

New results for local solutions

- Any local FBNE also satisfies the first-order necessary conditions for a local OLNE of the game, and vice versa.

New results for local solutions

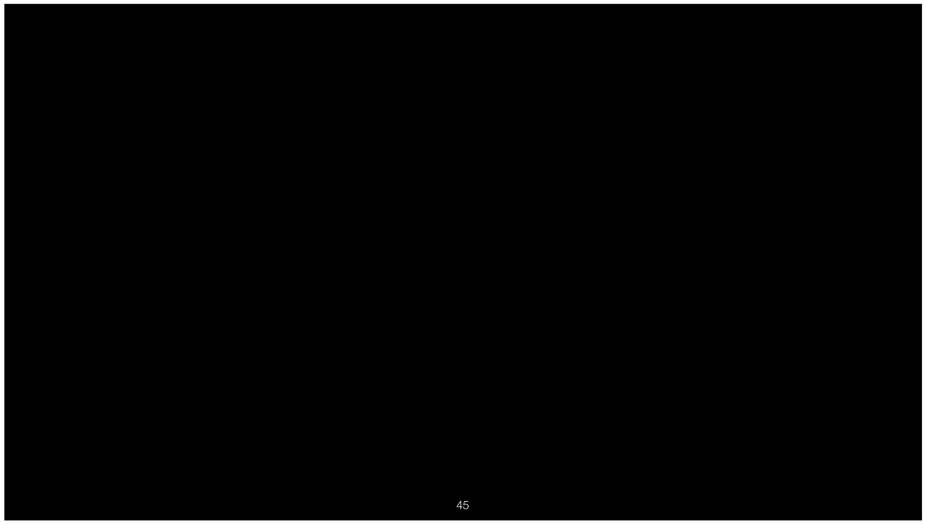
- Any local FBNE also satisfies the first-order necessary conditions for a local OLNE of the game, and vice versa.

- Any local FBNE trajectory also satisfies the second-order necessary conditions for a local OLNE. Further, a local FBNE trajectory satisfying feedback second-order sufficiency conditions is a local OLNE.

New results for local solutions

- Any local FBNE also satisfies the first-order necessary conditions for a local OLNE of the game, and vice versa.
- Any local FBNE trajectory also satisfies the second-order necessary conditions for a local OLNE. Further, a local FBNE trajectory satisfying feedback second-order sufficiency conditions is a local OLNE.

- In the presence of additional constraints on agents’ control variables, any local FBNE satisfying strict complementarity still satisfies the first-order necessary conditions for a local OLNE of the game, and vice versa.



Dynamic games are flexible models of interaction!

Dynamic games are flexible models of interaction!

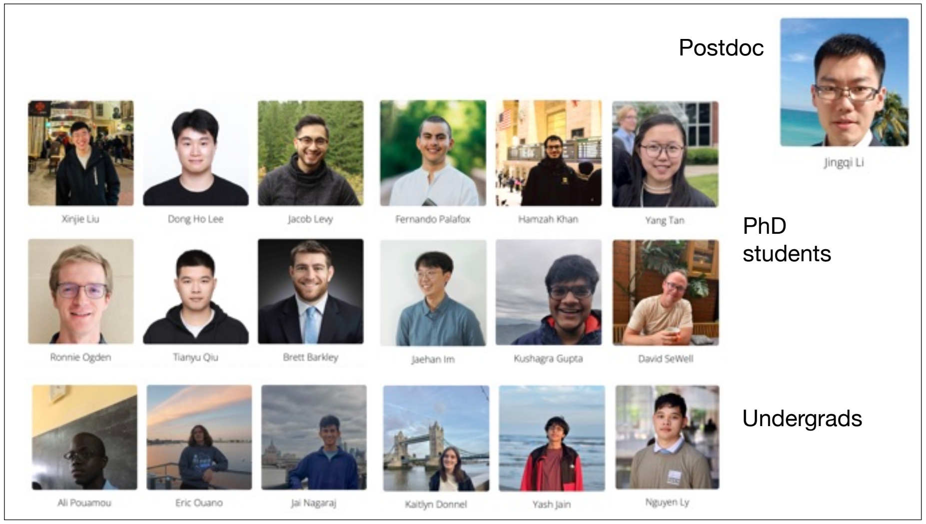
Information structure is (at least to me) the most interesting and underexplored part of game theory.

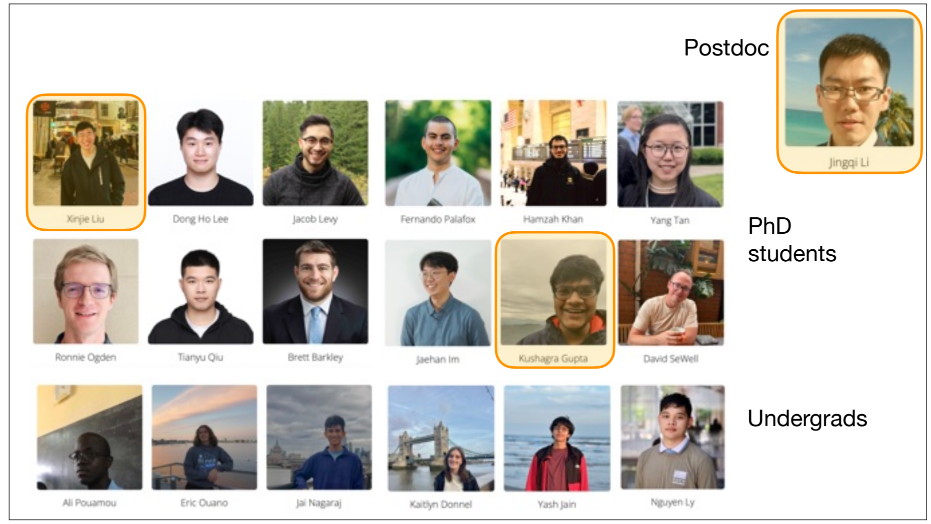
Dynamic games are flexible models of interaction!

Information structure is (at least to me) the most interesting and underexplored part of game theory.

Even though these games have been studied for decades, there are still exciting results out there!

Thank you! Questions?


Postdoc
Jingqi Li
PhD students
Xinjie Liu
Dong Ho Lee
Jacob Levy
Fernando Palafox
Hamzah Khan
Yang Tan
Ronnie Ogden
Tianyu Qiu
Brett Barkley
Jaehan Im
Kushagra Gupta
David SeWell
Undergrads
Ali Pouamou
Eric Ouano
Jai Nagaraj
Kaitlyn Donnel
Yash Jain
Nguyen Ly


Postdoc
Jingqi Li
PhD students
Xinjie Liu
Dong Ho Lee
Jacob Levy
Fernando Palafox
Hamzah Khan
Yang Tan
Ronnie Ogden
Tianyu Qiu
Brett Barkley
Jaehan Im
Kushagra Gupta
David SeWell
Undergrads
Ali Pouamou
Eric Ouano
Jai Nagaraj
Kaitlyn Donnel
Yash Jain
Nguyen Ly