# CSCE 631-D: Intelligent Agents — Computational Game Solving

## Lecture 1: Introduction — Games, Agents, and Strategic Interaction

Alan Kuhnle

Fall 2026

Texas A&M University

# Today's Plan

We start with **game theory** as a discipline, introduce **agents** as players, then build the formal framework: strategies, payoffs, classic examples, pure and mixed strategies, equilibrium existence, and a taxonomy of game types.

# Learning Objectives

By the end of this lecture, you should be able to:

1. Define what makes an interaction a **game** rather than a single-agent optimization problem.
2. Write the **normal-form representation** of a two-player game (players, strategies, payoff matrix).
3. Identify **dominant strategies** and **Pareto-optimal** outcomes in a payoff matrix.
4. Explain why **mixed strategies** are needed and what role they play in equilibrium existence.

# What Is Game Theory?

# What Is Game Theory?

**Informal Definition**

Game theory is the mathematical study of **strategic interaction** among rational decision-makers.

Key ingredients:

- Multiple **agents** (players) whose outcomes depend on each other's choices.
- Each agent ranks outcomes with a scalar **utility** (payoff) and prefers a higher utility.
- Agents reason about what others will do.

**Central question:** What outcomes should we *predict* (or *prescribe*) when rational agents interact?

# Why Game Theory in CS?

- **Internet and networks:** routing, congestion, selfish agents.
- **Auctions & mechanism design:** ad auctions (Google, Meta), spectrum auctions, matching markets.
- **Multi-agent AI:** reinforcement learning, LLM alignment, cooperative/competitive agents.
- **Security:** adversarial models, Stackelberg games.
- **Complexity theory:** PPAD, hardness of equilibrium computation.

Today's multi-agent systems are **LLM agents**: negotiation bots, multi-agent pipelines, agent marketplaces. These are the course's motivating population of players.

# What Is an Agent?

# What Is an Agent?

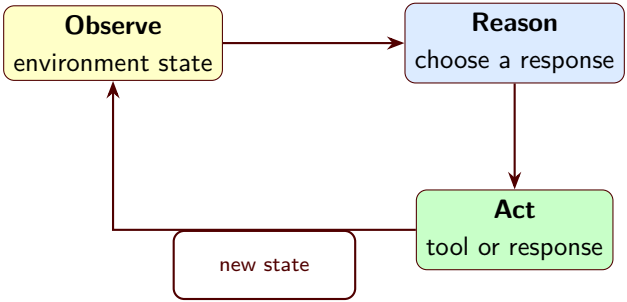
A game needs players. In this course the players are LLM-based agents, so we begin by defining what an agent is.

**Operational definitions** (Anthropic, "Building Effective Agents"):

- **Language model:** a single LLM call; no control flow.
- **Workflow:** LLM calls orchestrated by fixed, human-designed control flow (chains, routers, pipelines).
- **Agent:** the LLM itself directs the control flow, choosing which tools to call and when to stop.

In this course, an **agent** is an LLM with tool access and LLM-directed control flow. The "game" emerges when two or more agents interact strategically.

# The Agent Loop




- **Observe:** read a tool result, user message, or game history.
- **Reason:** select a response using internal reasoning.
- **Act:** invoke a tool or emit a final response.

Tools are the agent's **action set**. The LLM's conditional generation defines the **policy**: a mapping from observations to (distributions over) actions.

# Demo: ReAct Agent on the TAMU API

**Live demo** (recorded for the asynchronous section).

**Demo notebook**

`demos/demo-react-trace.ipynb`

**What to watch for:**

- Raw prompt sent to the LLM.
- Streamed Thought / Action / Observation JSON: one realized trajectory.
- Final action selected by the agent.
- Model, token counts, and estimated cost printed by the notebook.
- **Recorded fallback:** replay the notebook's fixed three-round opponent schedule when live API calls are unavailable.

# The Agent–Environment Tuple

The ReAct trace is one realized trajectory. Three objects describe what an agent can see and do.

- $\mathcal{H}$: **hidden-state space** of the environment (includes game history and opponent's internal state).
- $O: \mathcal{H} \to \mathcal{O}$: **observation function** (what the agent sees after each step).
- $\mathcal{A}$: **action set** (the tools and responses available to the agent).

**Connection to the agent loop**

The agent observes $O(h)$, then chooses an action $a \in \mathcal{A}$. For simplicity $O$ is deterministic here; Module 4 also allows stochastic observations and information sets.

# Dynamics and Objective

Two further components specify what actions do and how the agent is scored.

- $T : \mathcal{H} \times \mathcal{A} \to \Delta(\mathcal{H})$: **transition kernel** (how the hidden state updates after an action).
- $r : \mathcal{H} \times \mathcal{A} \to \mathbb{R}$: **objective** (payoff, reward).

**Mapping to the ReAct trace:** $\mathcal{A} = \{\text{cooperate}, \text{defect}\}$; $O$ maps the hidden game state to the observation the agent receives; $r$ is the payoff.

This describes a single agent interacting with an environment. What is missing before this becomes a *game*?

## What Is Still Missing Before This Is a Game?

The agent–environment tuple describes one player.

*Think for 30 seconds: what does this single-agent tuple lack before it becomes a game?*

A **game** additionally requires:

1. Multiple **players** (and possibly chance).
2. A **history space**: the full tree of possible action sequences, not just one realized trajectory.
3. **Observations** per player: who sees what, and when (information sets).
4. **Counterfactual branches**: what would have happened under alternative actions.
5. **Per-player utilities** at terminal histories.

We will build the full extensive-form game formally in **Module 4**. Today we focus on the simpler **normal-form** representation, where all players choose simultaneously.

**We now have the players; let us define the game.**

# Players, Strategies, and Payoffs

## The Three Ingredients

With multiple players identified as the key missing ingredient, we now define the building blocks of a game.

Every game has three components:

1. **Players:** A finite set $N = \{1, 2, \ldots, n\}$.
2. **Strategies:** For each player $i \in N$, a set $S_i$ of available actions.
3. **Payoffs:** For each player $i$, a utility function

$$u_i : S_1 \times S_2 \times \cdots \times S_n \to \mathbb{R}.$$

**Notation**

A **strategy profile** is a tuple $s = (s_1, s_2, \ldots, s_n) \in S_1 \times \cdots \times S_n$. We write $s_{-i}$ for the profile of all players *except* $i$.

# Normal-Form (Strategic-Form) Game

**Definition**

A **normal-form game** (also called a **strategic-form game**) is a tuple

$$G = (N,\ \{S_i\}_{i \in N},\ \{u_i\}_{i \in N})$$

where $N$ is the set of players, $S_i$ is the strategy set of player $i$, and $u_i : \prod_{j \in N} S_j \to \mathbb{R}$ is the payoff function of player $i$.

Assumption for most of this course: $N$ and every $S_i$ are **finite**.

All players choose **simultaneously** (or equivalently, without knowledge of others' choices).

# Payoff Matrices (Two-Player Case)

For $n = 2$, an $m \times k$ **bimatrix** $(A, B)$ represents a game with $m$ row strategies and $k$ column strategies:

$$A_{ij} = u_1(s_1^{(i)}, s_2^{(j)}), \qquad B_{ij} = u_2(s_1^{(i)}, s_2^{(j)}),$$

Here row $i$ is Player 1's strategy $s_1^{(i)}$, and column $j$ is Player 2's strategy $s_2^{(j)}$.

Convention: Player 1 is the **row** player and Player 2 is the **column** player; cell $(i, j)$ contains $(A_{ij}, B_{ij})$.

| | $L$ | $R$ |
|---|---|---|
| $U$ | $(a_{11},\ b_{11})$ | $(a_{12},\ b_{12})$ |
| $D$ | $(a_{21},\ b_{21})$ | $(a_{22},\ b_{22})$ |

# Nash Equilibrium (Informal Preview)

The classic examples in the next section refer to **Nash equilibria**, the central solution concept in game theory. We formalize Nash equilibria in Lecture 3; for now, the following informal definition suffices.

**Informal Definition**

A **Nash equilibrium (NE)** is a strategy profile in which no player can improve their payoff by unilaterally changing their strategy.

In other words, at a Nash equilibrium every player is already playing a best response to the other players' strategies.

# Two Comparisons for the Examples

The next games use two distinct comparisons. Keep the question being asked separate.

**Dominance: compare one player's actions**

Strategy $s_i$ **strictly dominates** $s_i'$ if, against *every* opponent profile $s_{-i}$, it gives player $i$ a strictly higher payoff:

$$u_i(s_i, s_{-i}) > u_i(s_i', s_{-i}).$$

**Pareto comparison: compare complete outcomes**

Outcome $s'$ **Pareto-dominates** $s$ if every player is at least as well off at $s'$ and at least one player is strictly better off.

# Classic Examples

## Prisoner's Dilemma

The next five games are classic examples. Each isolates a different strategic tension.

Two suspects are interrogated separately. Each can **Cooperate** (stay silent) or **Defect** (betray the other).

| | Cooperate | Defect |
|---|---|---|
| Cooperate | $(-1, -1)$ | $(-3, 0)$ |
| Defect | $(0, -3)$ | $(-2, -2)$ |

**Check (30 seconds):** In each column, which row gives the row player the higher payoff?

## Prisoner's Dilemma: Analysis

- **Answer:** Defect strictly dominates Cooperate: it gives either player a higher payoff against either opponent action.
- Unique outcome: $(\text{Defect}, \text{Defect})$ with payoff $(-2, -2)$.
- $(\text{Defect}, \text{Defect})$ is Pareto-dominated by $(\text{Cooperate}, \text{Cooperate})$.

*Question: Would the outcome change if both players could communicate beforehand?*

# Battle of the Sexes

Two players want to coordinate but have different preferences.

| | Opera | Football |
|---|---|---|
| Opera | $(3, 2)$ | $(0, 0)$ |
| Football | $(0, 0)$ | $(2, 3)$ |

- Two **pure-strategy Nash equilibria**: (Opera, Opera) and (Football, Football).
- Neither player has a dominant strategy.
- Coordination is valuable; mis-coordination is costly.
- Also has a **mixed-strategy** NE (a preview; we define mixing shortly).

# Matching Pennies

A strictly competitive game: payoffs sum to zero in every cell.

| | Heads | Tails |
|---|---|---|
| Heads | $(1, -1)$ | $(-1, 1)$ |
| Tails | $(-1, 1)$ | $(1, -1)$ |

- **No pure-strategy NE:** every pure outcome has a player who can improve their payoff by unilaterally changing their strategy.
- Unique mixed NE (a preview): each player plays Heads with probability $1/2$.
- This is a **zero-sum game**: every cell's payoffs sum to zero.

*Why can neither player commit to a single strategy here?*

## Coordination Game

Both players benefit from choosing the *same* action.

| | $A$ | $B$ |
|---|---|---|
| $A$ | $(2, 2)$ | $(0, 0)$ |
| $B$ | $(0, 0)$ | $(1, 1)$ |

**Quick poll (30 seconds):** Which equilibrium would you predict without a coordinating signal?

- Two pure NE: $(A, A)$ and $(B, B)$.
- $(A, A)$ **Pareto-dominates** $(B, B)$, but $(B, B)$ is still an equilibrium.
- Equilibrium **selection** problem: which NE should we predict?

# Stag Hunt

Two hunters choose independently: pursue a stag (high reward, requires both) or a hare (lower reward, guaranteed).

| | Stag | Hare |
|---|---|---|
| Stag | $(4, 4)$ | $(0, 3)$ |
| Hare | $(3, 0)$ | $(3, 3)$ |

- **Payoff-dominant:** (Stag, Stag) gives every player a higher payoff than any other equilibrium.
- **Risk-dominant:** (Hare, Hare) is safer under uncertainty about the other player's action.
- Models the tension between **cooperation** and **safety**.
- A key example in social contract theory (Rousseau).

# Pure and Mixed Strategies

# Pure Strategies

So far, every example assumed each player picks a single action. We now formalize this idea and then generalize it.

**Definition**
A **pure strategy** for player $i$ is a single element $s_i \in S_i$.

A **pure-strategy profile** is $(s_1, \ldots, s_n) \in S_1 \times \cdots \times S_n$.

In a pure strategy, the player commits deterministically to one action.

Problem: as we saw with Matching Pennies, some games have **no pure-strategy equilibrium**. We need a richer strategy space.

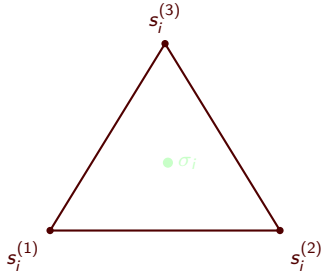
# Mixed Strategies: Distributions

**Definition**

A **mixed strategy** for player $i$ is a probability distribution $\sigma_i \in \Delta(S_i)$ over $S_i$, where

$$\Delta(S_i) = \left\{ p \in \mathbb{R}_{\geq 0}^{|S_i|} \;\middle|\; \sum_{s_i \in S_i} p(s_i) = 1 \right\}.$$

For a three-action strategy set, the triangle at right is the simplex: every point is one probability distribution. Vertices are pure strategies; interior points mix among actions.



# Mixed Strategies: Expected Payoffs

Under a mixed-strategy profile $\sigma = (\sigma_1, \ldots, \sigma_n)$, players randomize independently. Player $i$'s **expected utility** is

$$u_i(\sigma) = \sum_{s \in S} \left( \prod_{j \in N} \sigma_j(s_j) \right) u_i(s).$$

- A pure strategy is a degenerate distribution that assigns probability one to a single action.
- The mixed extension replaces $S_i$ with $\Delta(S_i)$ and extends each payoff by expected value.
- **LLM connection:** temperature-sampled outputs make a policy stochastic. In this course, strategies are prompt, model, and scaffold policies rather than individual utterances.

# Support of a Mixed Strategy

**Definition**

The **support** of a mixed strategy $\sigma_i$ is the set of pure strategies played with positive probability:

$$\text{supp}(\sigma_i) = \{s_i \in S_i \mid \sigma_i(s_i) > 0\}.$$

Preview result (we will prove this after formally defining Nash equilibrium):

Intuitively, if one pure strategy in the support yielded a higher expected payoff, the player would shift all probability to it, contradicting mixing.

**Preview: Indifference Principle**

If $\sigma$ is a Nash equilibrium and $s_i, s_i' \in \text{supp}(\sigma_i)$, then

$$u_i(s_i, \sigma_{-i}) = u_i(s_i', \sigma_{-i})$$

## Example: Mixed Strategy in Matching Pennies

Recall the Matching Pennies payoff matrix for Player 1:

$$A = \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}.$$

**Try it first (30 seconds):** What value of $q$ makes Player 1 indifferent between Heads and Tails?

## Example: Mixed Strategy in Matching Pennies

Recall the Matching Pennies payoff matrix for Player 1:

$$A = \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}.$$

**Try it first (30 seconds):** What value of $q$ makes Player 1 indifferent between Heads and Tails?

Let Player 2 play Heads with probability $q$. Player 1's expected payoffs:

$$u_1(\text{Heads}, q) = q \cdot 1 + (1-q) \cdot (-1) = 2q - 1$$
$$u_1(\text{Tails}, q) = q \cdot (-1) + (1-q) \cdot 1 = 1 - 2q$$

For Player 1 to be willing to mix, the **Indifference Principle** requires equal expected payoffs:

$$2q - 1 = 1 - 2q \quad \Longrightarrow \quad q = \tfrac{1}{2}.$$

# Existence of Equilibrium

**Nash's Theorem (1950)**
Every finite game has at least one **mixed-strategy Nash equilibrium**.

Matching Pennies illustrates why this result requires mixed strategies: no pure-strategy equilibrium exists, yet the uniform mixture $(\frac{1}{2}, \frac{1}{2})$ for each player is a Nash equilibrium. We will study the proof of Nash's theorem in Lecture 3.

# Taxonomy of Games

# Zero-Sum Games

**Definition**

A two-player game $(N, \{S_1, S_2\}, \{u_1, u_2\})$ is **zero-sum** if for every strategy profile $s \in S_1 \times S_2$:

$$u_1(s) + u_2(s) = 0.$$

**Quick check (30 seconds):** Which earlier game has payoffs that sum to zero in every cell?

# Zero-Sum Games

**Definition**

A two-player game $(N, \{S_1, S_2\}, \{u_1, u_2\})$ is **zero-sum** if for every strategy profile $s \in S_1 \times S_2$:

$$u_1(s) + u_2(s) = 0.$$

**Quick check (30 seconds):** Which earlier game has payoffs that sum to zero in every cell?

- Equivalently, $u_2 = -u_1$; the game is fully described by one payoff matrix $A$, with $B = -A$.
- Matching Pennies is zero-sum. Prisoner's Dilemma is **not**.
- Zero-sum games are the "easiest" class: NE = maxmin = minimax (von Neumann, 1928).
- Solvable in polynomial time via linear programming.

# General-Sum and Constant-Sum Games

**General-Sum**

Any game that is not zero-sum. Payoff matrices $A$ and $B$ are independent. Most real-world interactions are general-sum.

**Constant-Sum**

$u_1(s) + u_2(s) = c$ for some constant $c$ and all $s$. Strategically equivalent to zero-sum (shift payoffs by $c/2$).

**Example:** Prisoner's Dilemma is general-sum; matching pennies is the special zero-sum case $c = 0$.

**Symmetric Game**

A two-player game is **symmetric** if $S_1 = S_2$ and $u_1(a, b) = u_2(b, a)$ for all $a, b \in S_1$. Equivalently, $B = A^\top$.

$$A = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \quad \Longrightarrow \quad B = A^\top = \begin{pmatrix} a & c \\ b & d \end{pmatrix}$$

Examples: Prisoner's Dilemma, Stag Hunt, and Coordination Game are symmetric.

## Larger Games and Representations

When $n > 2$ or $|S_i|$ is large, explicit payoff tables become impractical.

Alternative representations (we may see later):

- **Graphical games:** payoff of $i$ depends only on neighbors in a graph.
- **Congestion games:** payoff depends on number of players sharing resources.
- **Succinct games:** compact circuit- or polynomial-based payoff descriptions.
- **Extensive-form games:** sequential structure (preview in Lecture 2; formal treatment in Module 4).

The normal form is the most general but least compact representation. An $n$-player game with $m$ strategies each requires $n \cdot m^n$ payoff values.

**Caution:** succinct representations are exactly where complexity claims need care.

# Connecting Game Theory to LLM Agents

# Two LLM Negotiators as a Normal-Form Game

The earlier examples used abstract players. The same payoff-matrix framework applies when the players are prompted LLMs. Fix two prompt policies per agent (for example, cooperative or aggressive seller/buyer prompts):

| | Cooperative Buyer | Aggressive Buyer |
|---|---|---|
| Cooperative Seller | $(u_{11},\ v_{11})$ | $(u_{12},\ v_{12})$ |
| Aggressive Seller | $(u_{21},\ v_{21})$ | $(u_{22},\ v_{22})$ |

Each cell's payoff is estimated from repeated play. This is the empirical game construction in **Programming Assignment 1 (PA1)**.

# Stochastic Policies and Evaluation

An LLM policy is **stochastic**: the same prompt can produce different outputs across runs (temperature, sampling).

**Key principle**

One sampled interaction is *not* a payoff. Estimate every payoff-matrix cell with repeated calls, then report the mean and confidence interval.

For credible empirical games, also record model/version, token cost, and seeds where possible; use compute-matched baselines and holdout evaluation data.

# Looking Ahead

# Course Roadmap

Nine modules weave together two arcs: game-theory foundations and LLM-agent systems.

**M1** Foundations of strategic games (Weeks 1–2) *today*
**M2** Algorithms for equilibria (Weeks 3–4)
**M3** Regret minimization (Weeks 5–6)
**M4** Extensive-form games (Weeks 7–8)
**M5** CFR and self-play (Week 9)
**M6** Abstraction and agent action spaces (Week 10)
**M7** Multi-agent LLM systems (Week 11)
**M8** Monitoring autonomous agents (Week 12)
**M9** Applications and frontiers (Weeks 13–14)

Each concept resurfaces: NE (M2 algorithms), regret (M3), EFG (M4), empirical

# Summary and Key Takeaways

**Agents:**

- An **agent** is an LLM with tool access and LLM-directed control flow (observe, reason, act).
- A trajectory describes one agent; a game adds multiple players, counterfactual branches, and per-player utilities.
- Stochastic LLM policies require repeated-play payoff estimates.

**Games:**

- A **normal-form game** is $(N, \{S_i\}, \{u_i\})$: players, strategies, payoffs.
- Dominance compares actions; Pareto comparison asks whether an outcome can improve every player's payoff.
- Mixed strategies generalize pure strategies; every finite game has a mixed-strategy equilibrium (Nash, 1950).

# References i


- Y. Shoham and K. Leyton-Brown, *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations*, Cambridge University Press, 2008. Chapter 3.
- S. Yao et al., "ReAct: Synergizing Reasoning and Acting in Language Models," *ICLR*, 2023.
- Anthropic, "Building Effective Agents," blog post, December 2024.
- J. Sun, Z. Wu, Z. Cheng, and X. Chu, "Game Theory Meets Large Language Models," *IJCAI*, 2025.
- L. Wang et al., "A Survey on Large Language Model based Autonomous Agents," arXiv:2308.11432, 2023.
- X. Xi et al., "The Rise and Potential of Large Language Model Based Agents: A Survey," arXiv:2309.07864, 2023.