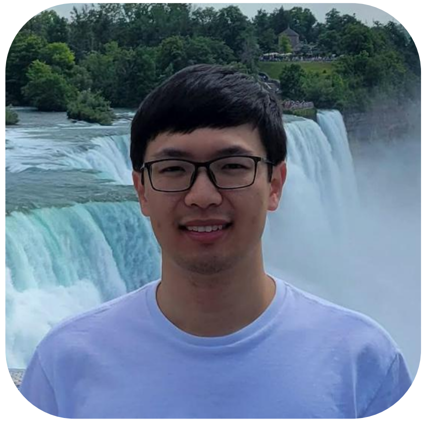
Deviate or Not: Learning Coalition Structures with Multiple-bit Observations in Games

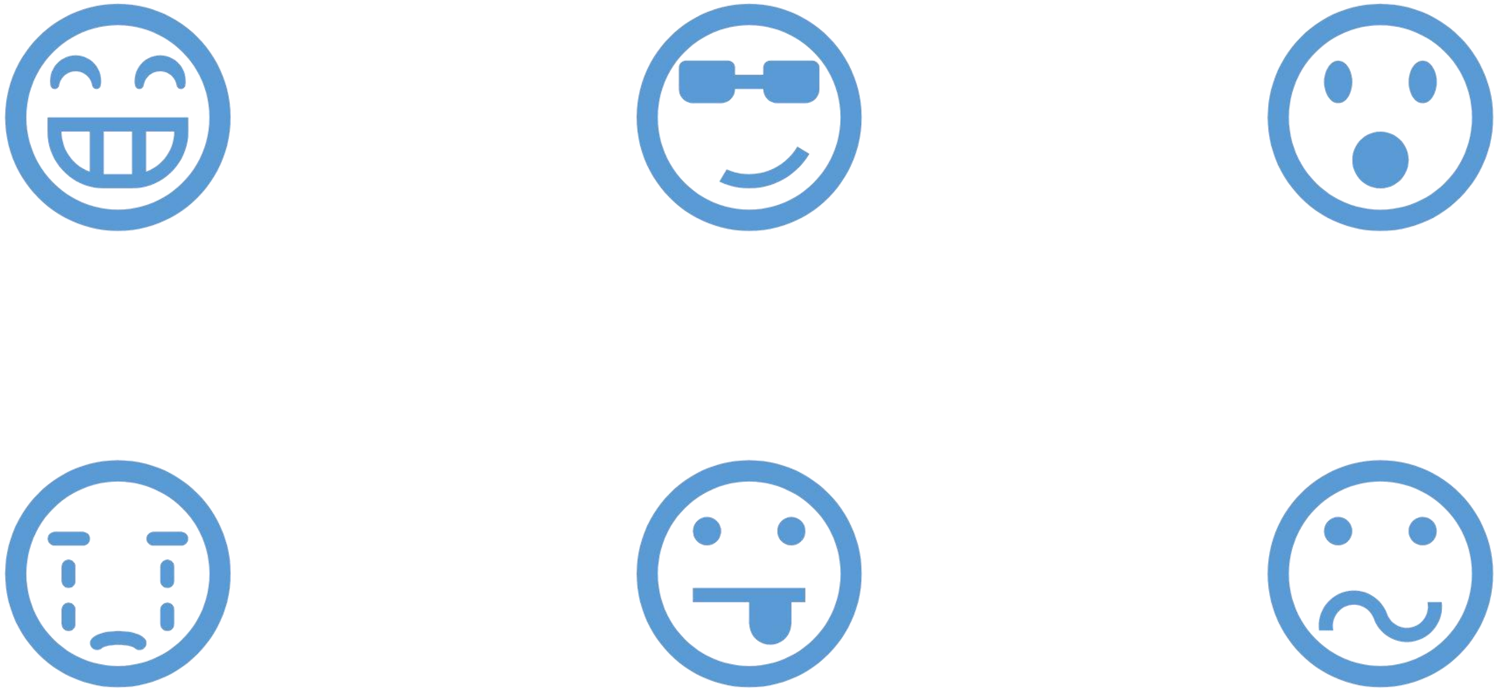
Yixuan (Even) Xu
Carnegie Mellon University



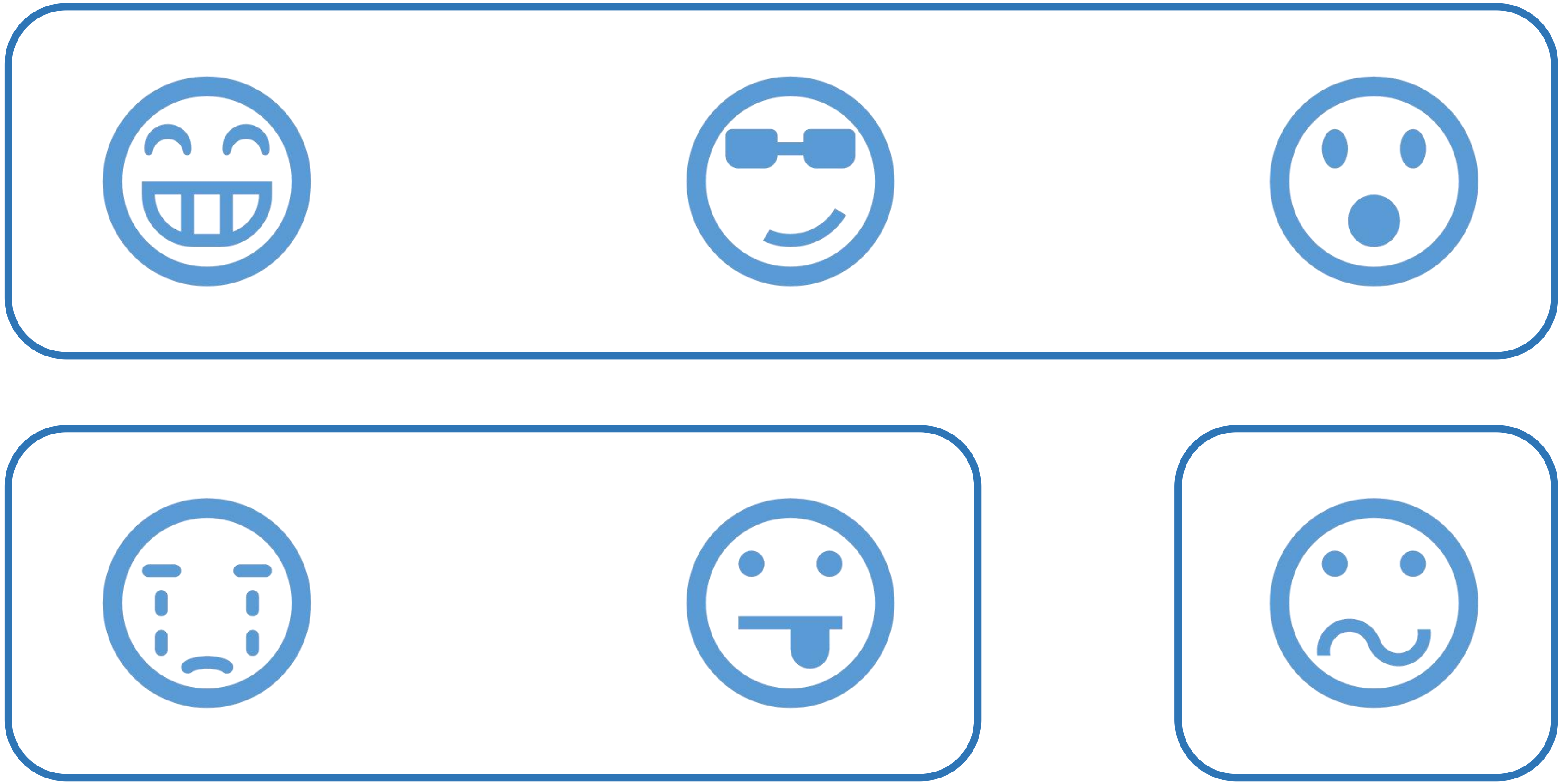
Zhe Feng
Google Research

Fei Fang
Carnegie Mellon University

Coalition Structures

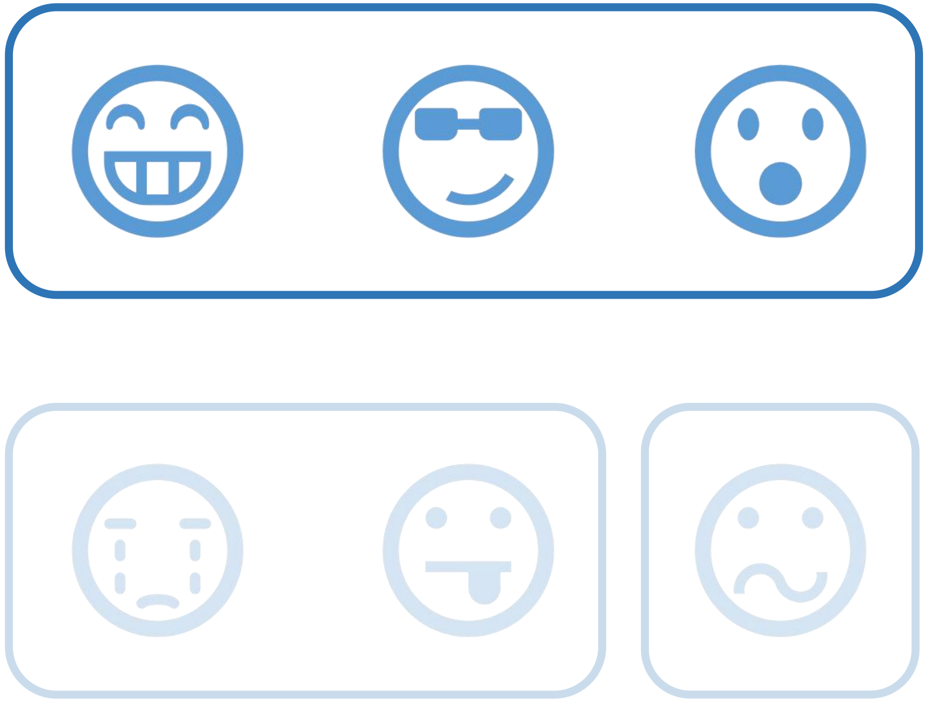


Coalition Structures



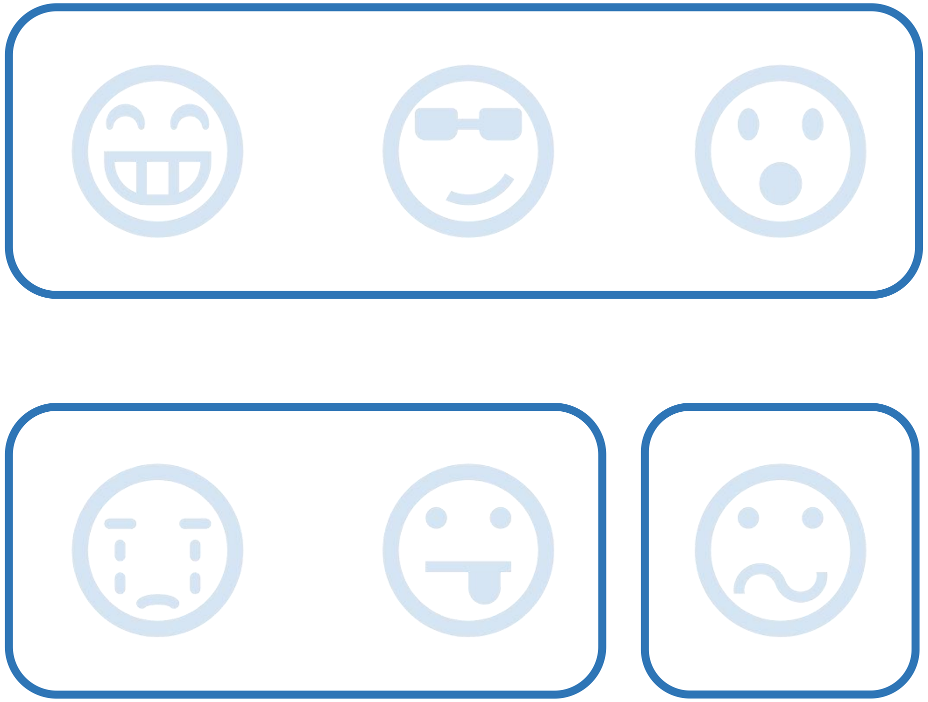
Coalition Structure Learning (CSL)

- **Coalition:** A nonempty subset of the agents, in which
  - The agents **coordinate their actions**
  - The agents **have common interests**



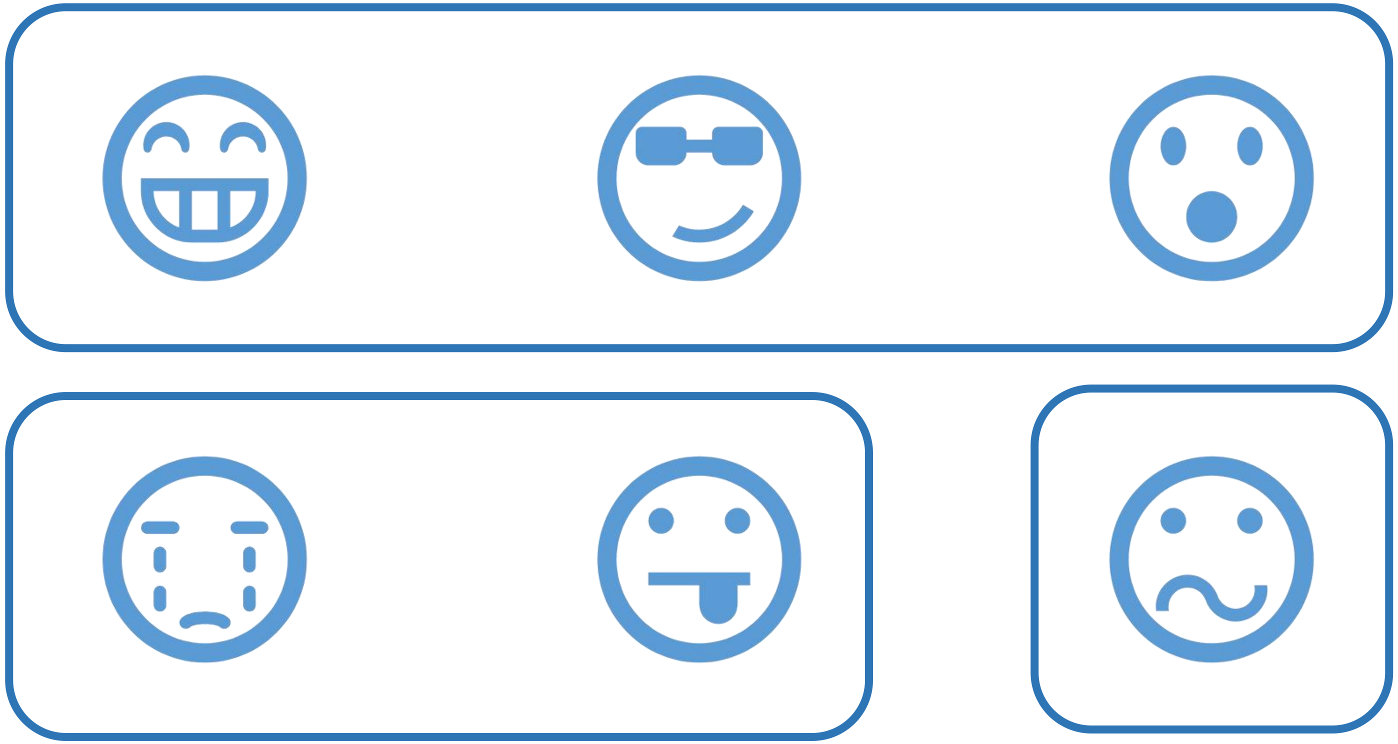
Coalition Structure Learning (CSL)

- **Coalition:** A nonempty subset of the agents, in which
  - The agents **coordinate their actions**
  - The agents **have common interests**
- **Coalition Structure:** A set partition of the agents $\{1,\ 2, \cdots, n\}$
  - Each set is a separate coalition
  - **Behavior Model in a Game:** Coalition **acts as a joint player** whose actual utility equals the **total utilities of its members**

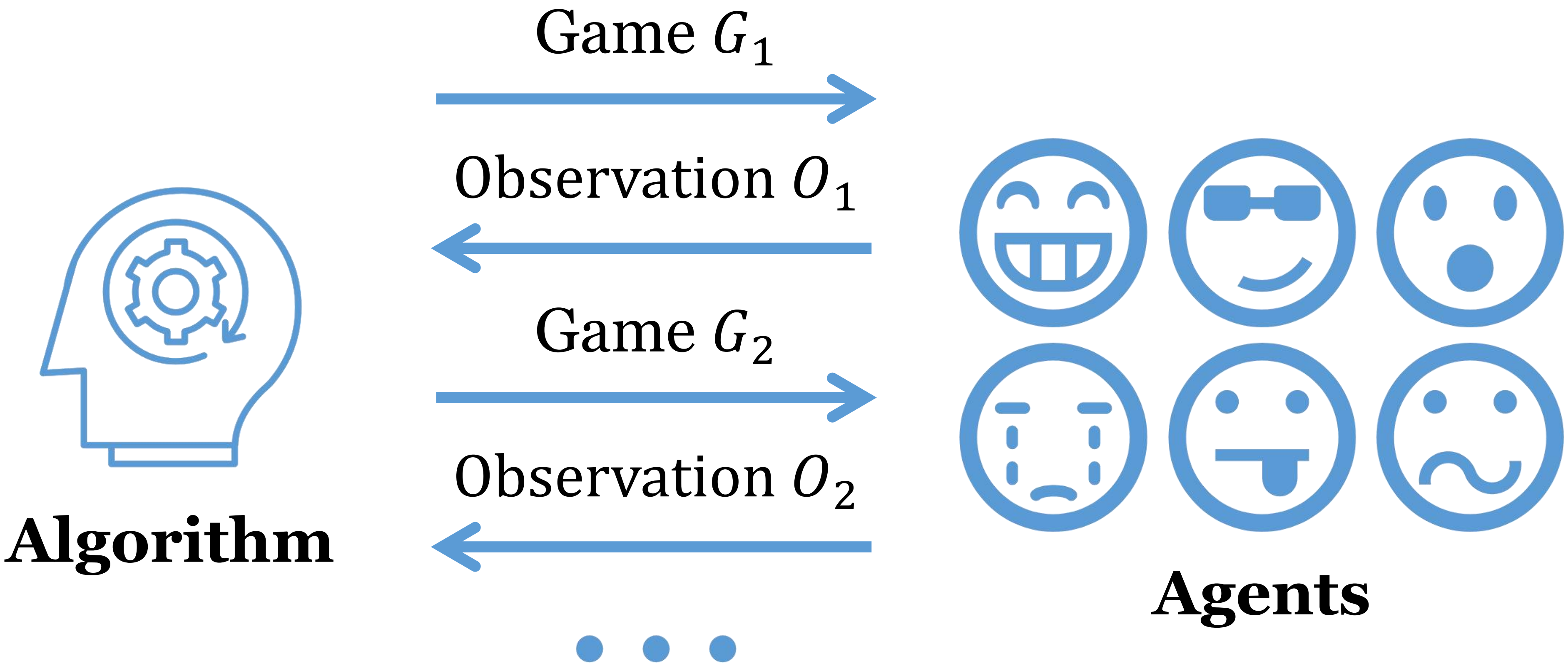


Coalition Structure Learning (CSL)

- **Coalition Structure Learning (CSL):** Recover the unknown coalition structure by observing interactions in designed games



Interactive Model



Single-Bit Observation (Xu et al. AAAI 2024)

- **Model:** The algorithm queries a game G and a strategy profile Σ, and the agents answer whether Σ is a **Nash Equilibrium** in G
  - Easy to compute for the agents
  - **One bit of information** per query

- **Theorem:** **Any algorithm** for CSL must interact **at least** $n \log_2 n - O(n \log_2 \log_2 n)$ rounds with the agents
  - We need this many bits of information to distinguish between answers
  - **Too large** for real-world systems with many agents

Multiple-Bit Observation (This Work)

- **Model:** The algorithm queries a game G and a strategy profile Σ, and each agent indicates **whether they want to deviate**
  - Still Easy to compute for the agents
  - ***n* bits of information** per query

- **Theorem:** Any algorithm for **Multiple-bit CSL** must interact at least $\log_2 n - O(\log_2 \log_2 n)$ rounds with the agents
  - This opens up the possibility of much more efficient algorithms

Types of Games

- What kind of games can the algorithm design?
  - Natural choice: **Normal-form games**
    - The **most general** one, thus the **easiest** for the algorithm
  - Succinct games: **Congestion games**, **graphical games**
  - More related to practice: **Auctions**

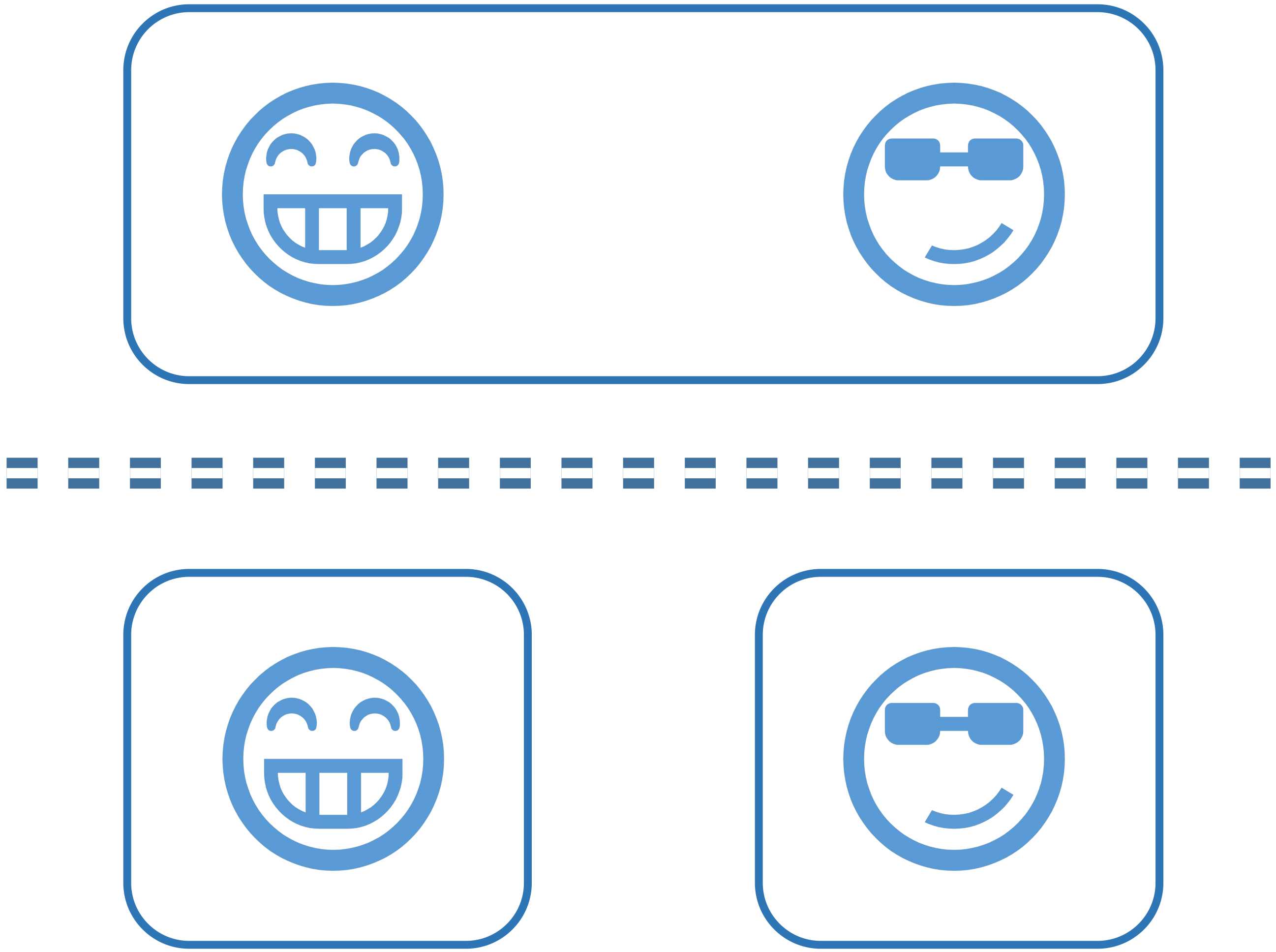
- We study **all** the above settings in this paper
  - And show **asymptotically optimal algorithms** for most of them
  - We **summarize** the results and the important techniques in the slides

Normal-form Games

- **Setting:** The algorithm may design any normal-form games
- **Lower bound:** $\log_2 n - O(\log_2 \log_2 n)$ rounds

- **Algorithm:** $\log_2 n + 2$ rounds in the worst case
  - **Gadget:** Product of **directed prisoner's dilemma**
  - **Key technique:** **Simultaneous binary search**
    - Find a representative in the coalition for each agent simultaneously
  - **Optimal** up to low-order terms

How to Distinguish Between the Two?



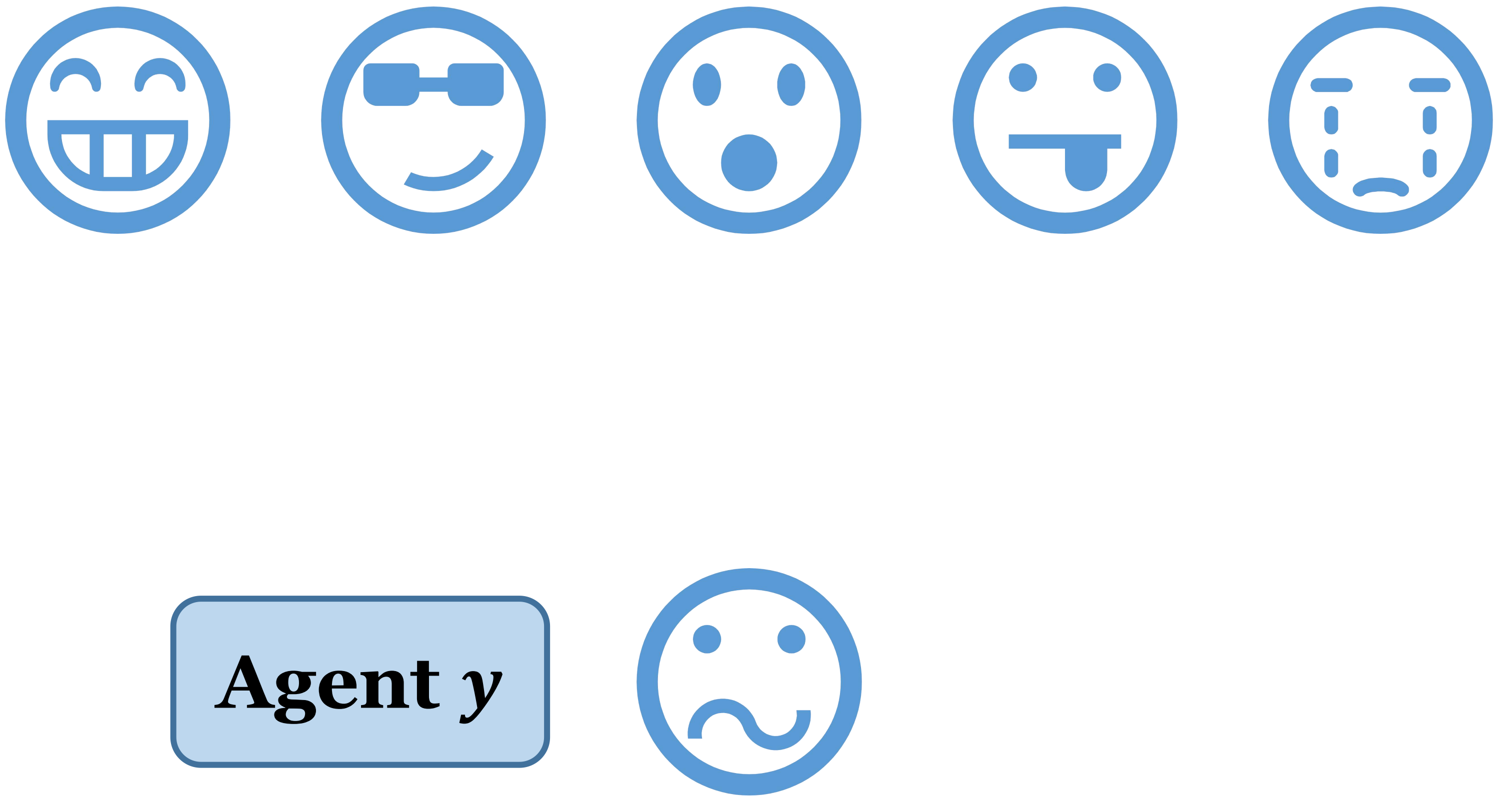
Directed Prisoner's Dilemma

- **Directed Prisoner's Dilemma:** A normal form game between agents (x, y), where agent y can choose cooperate, losing 1 unit of utility and giving agent x 2 units of utility

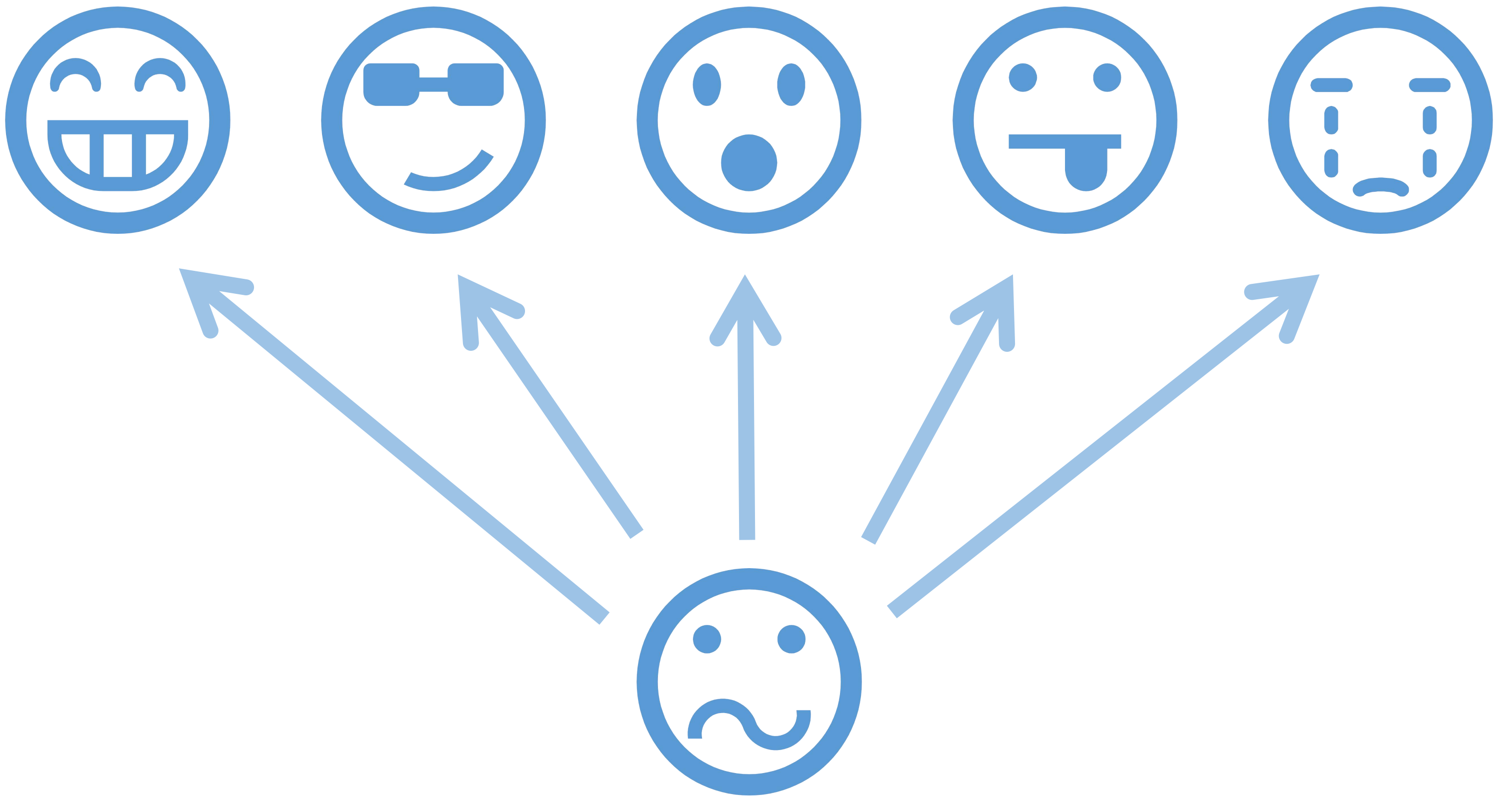
	$\boldsymbol{C_y}$	$\boldsymbol{D_y}$
D_x	(2, -1)	(0, 0)

- **Lemma 3.1:** Agent y **deviates** from D_y if and only if x and y are **in the same coalition**

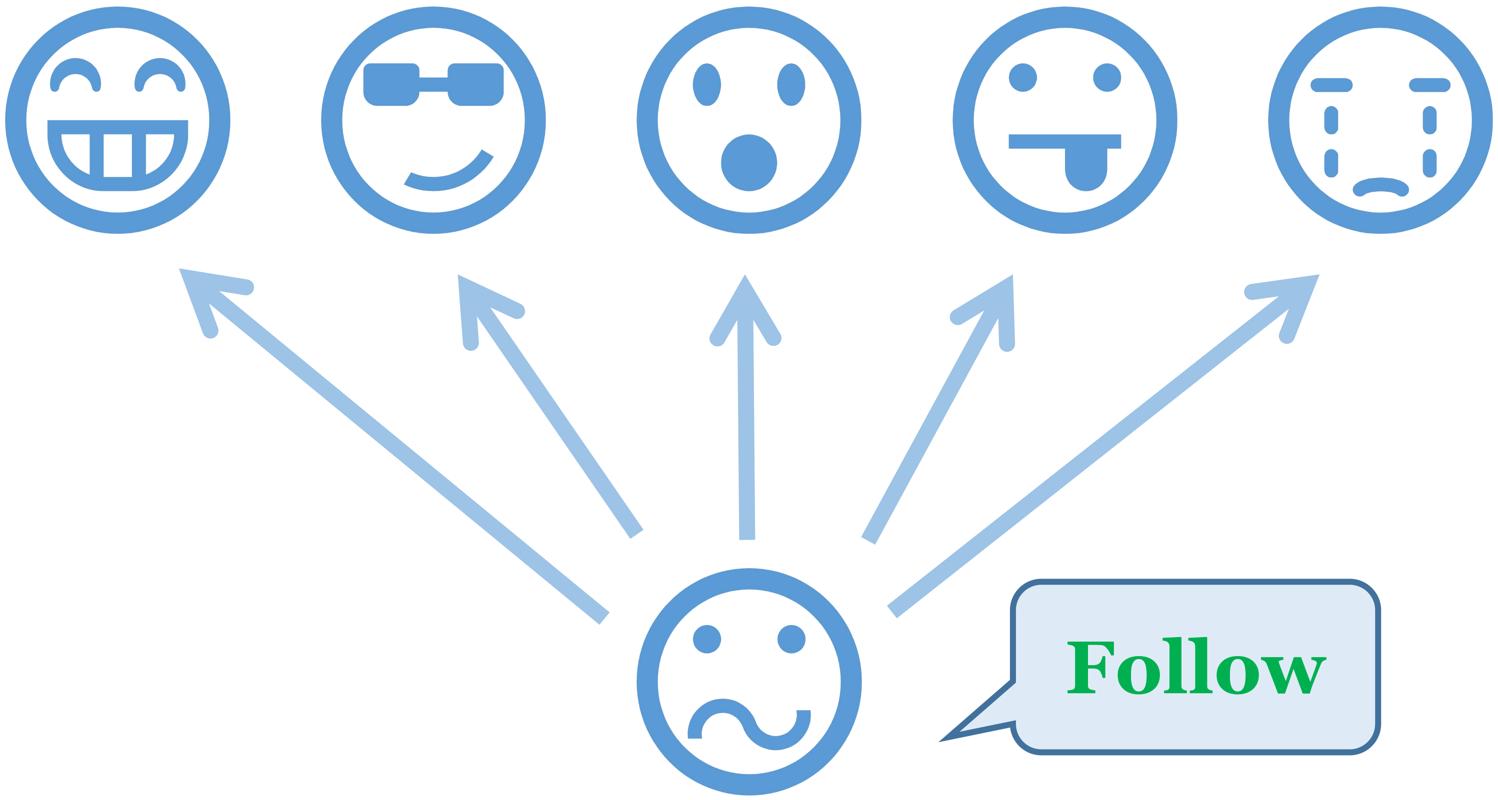
Aggregated Observation



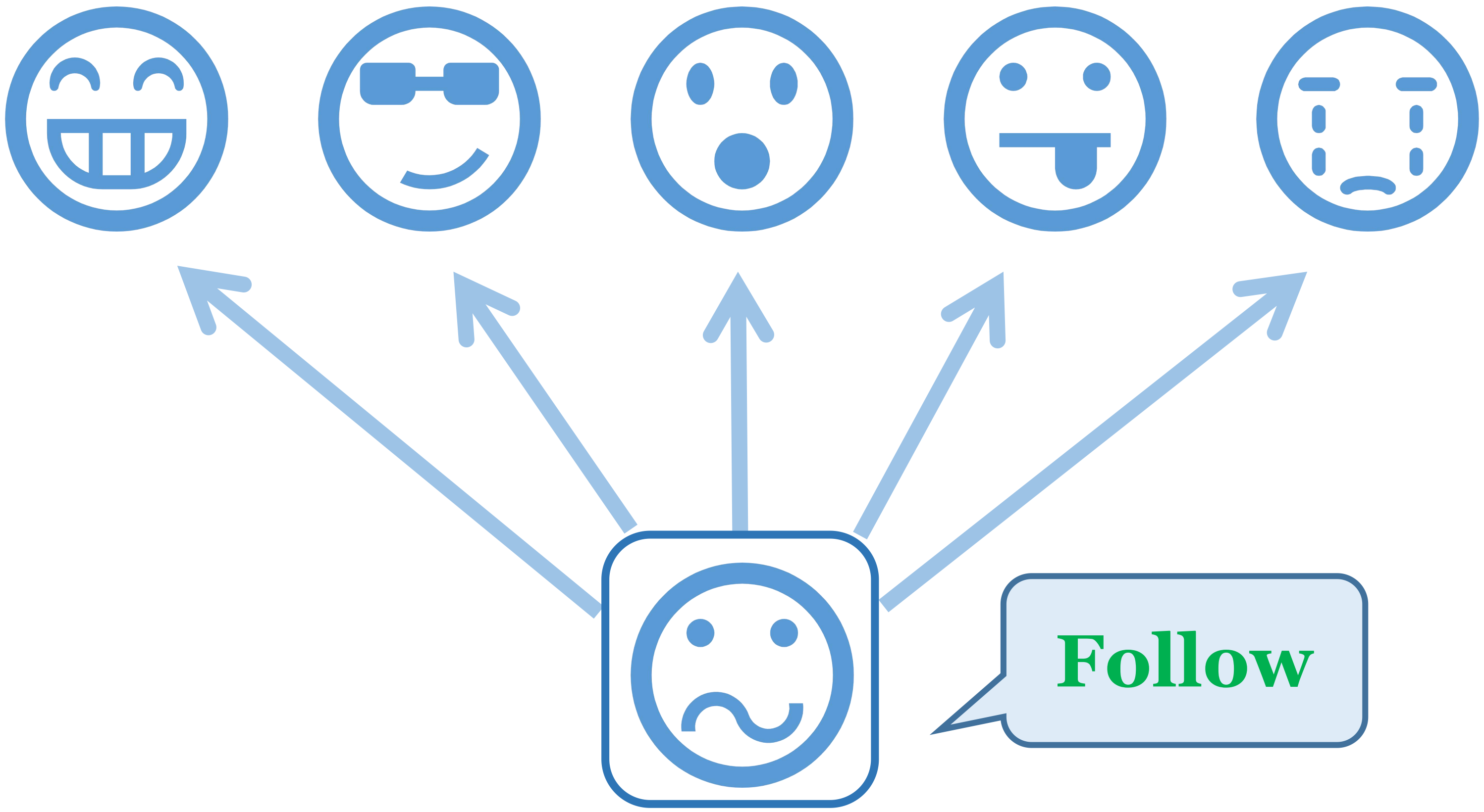
Aggregated Observation



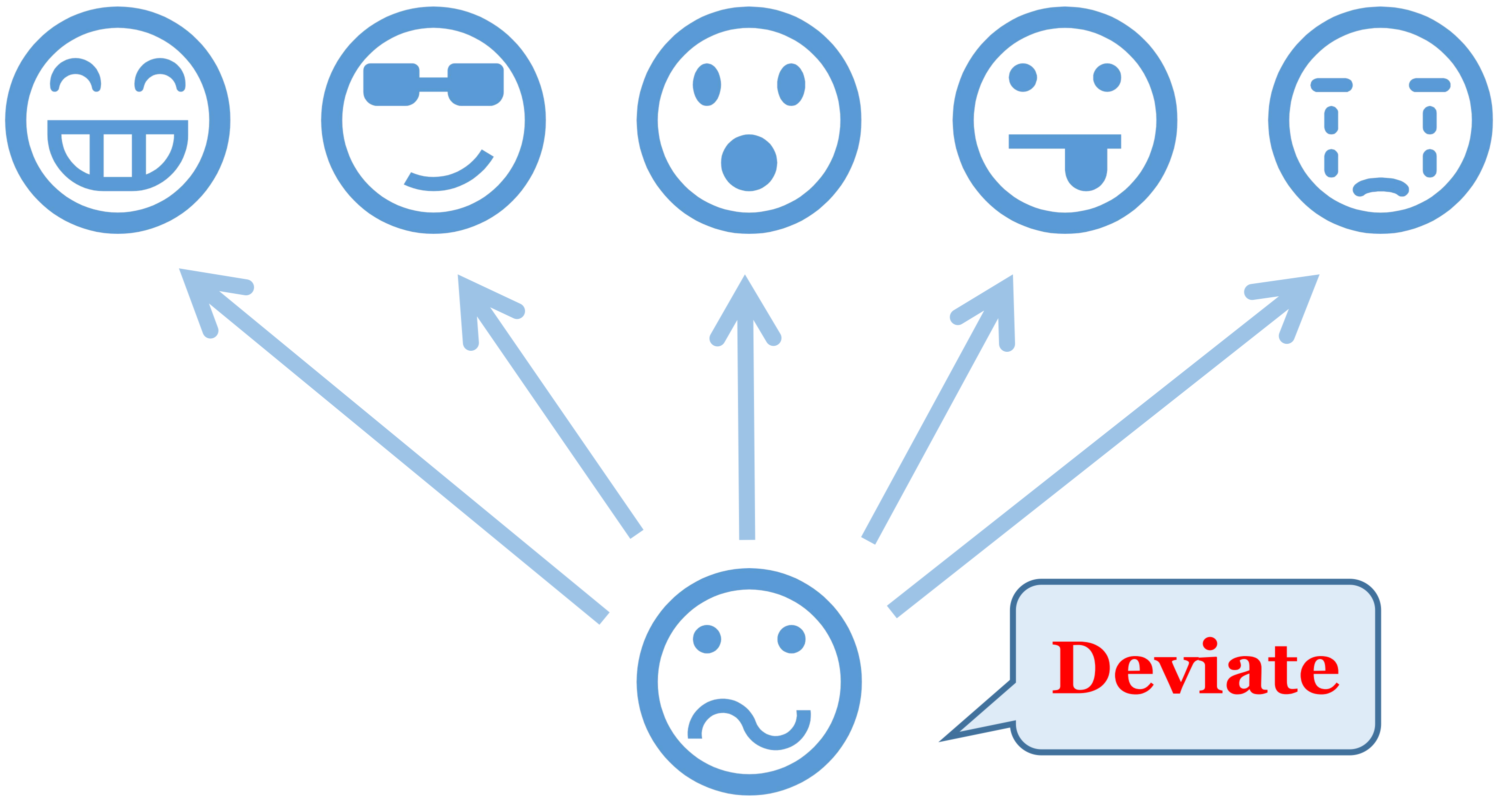
Aggregated Observation



Aggregated Observation

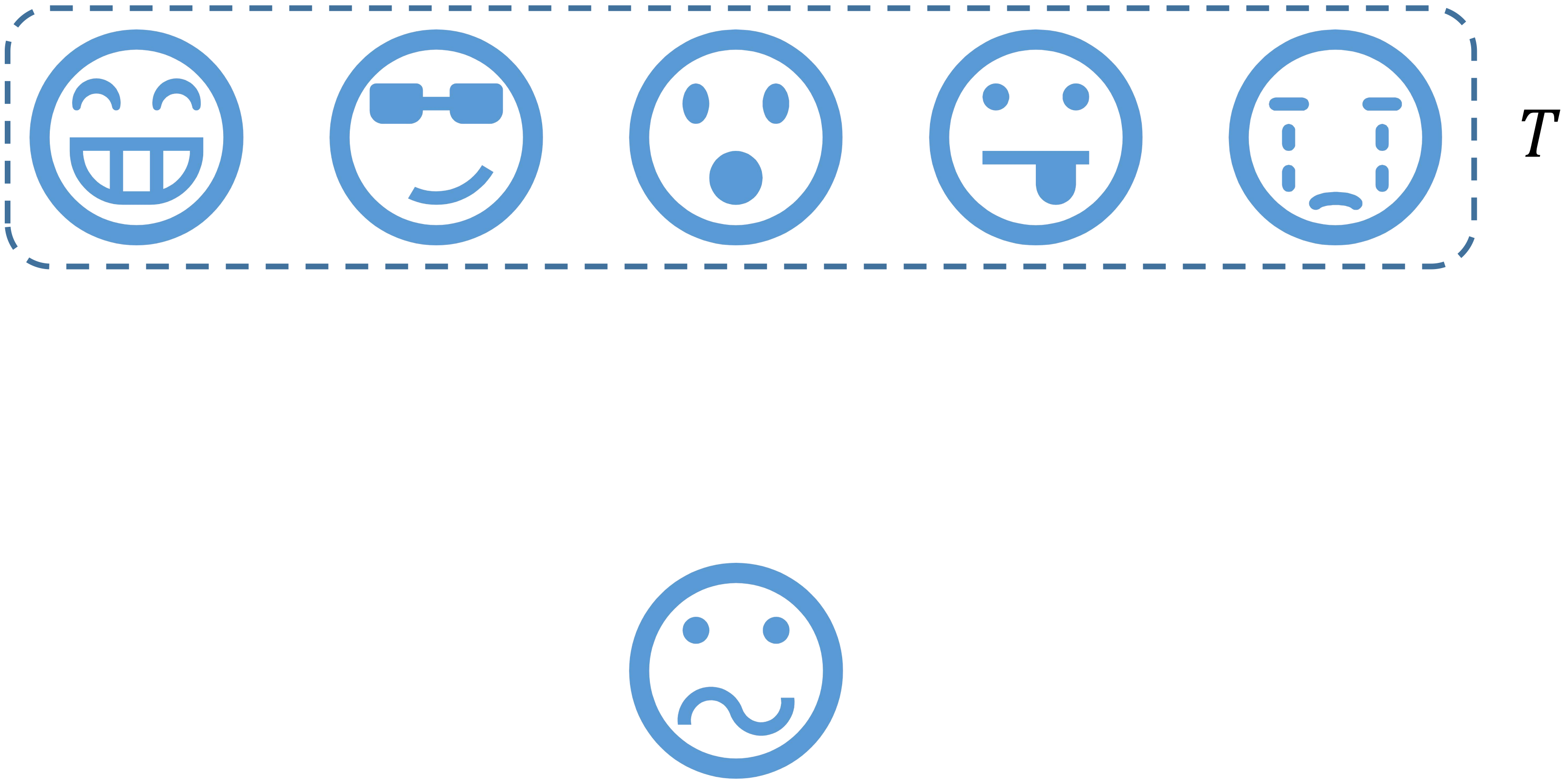


Aggregated Observation

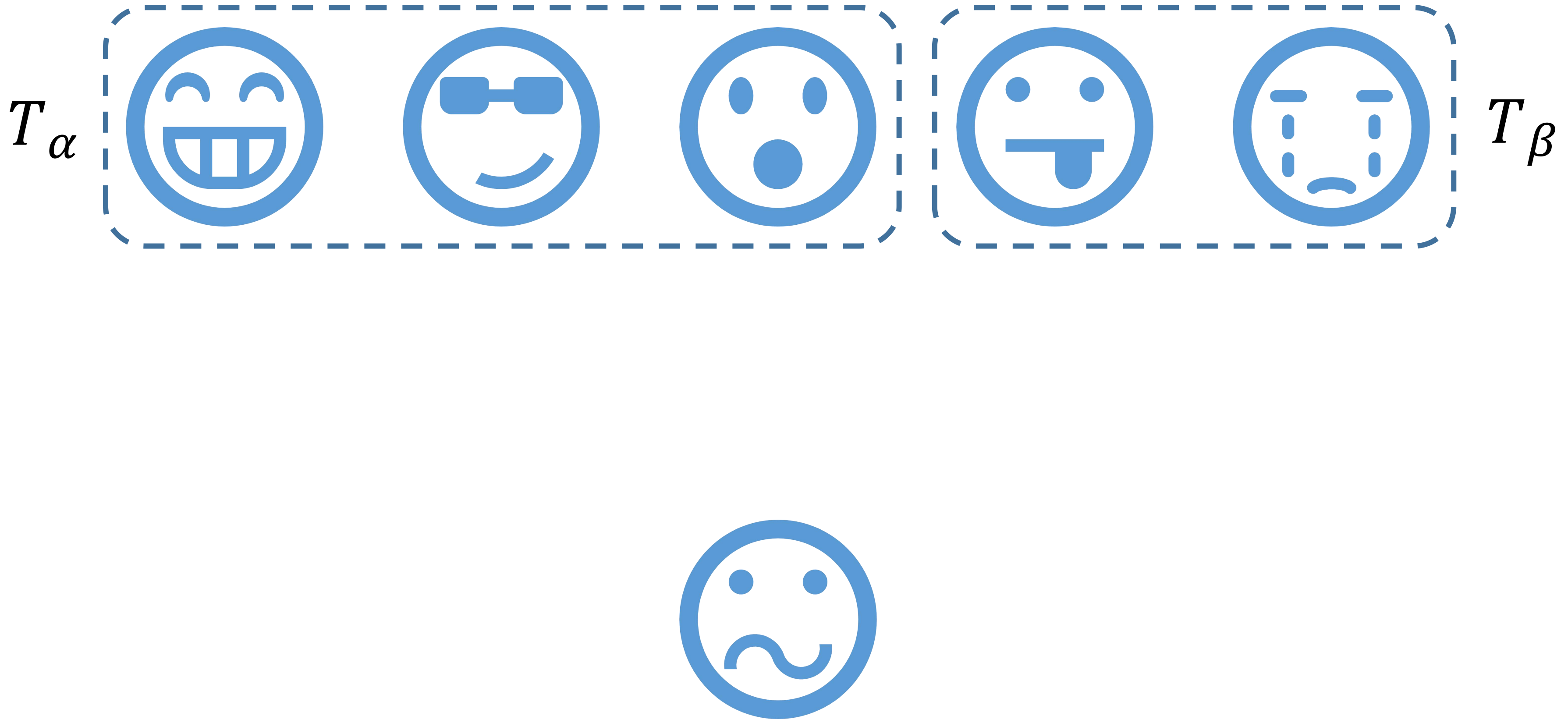


Aggregated Observation

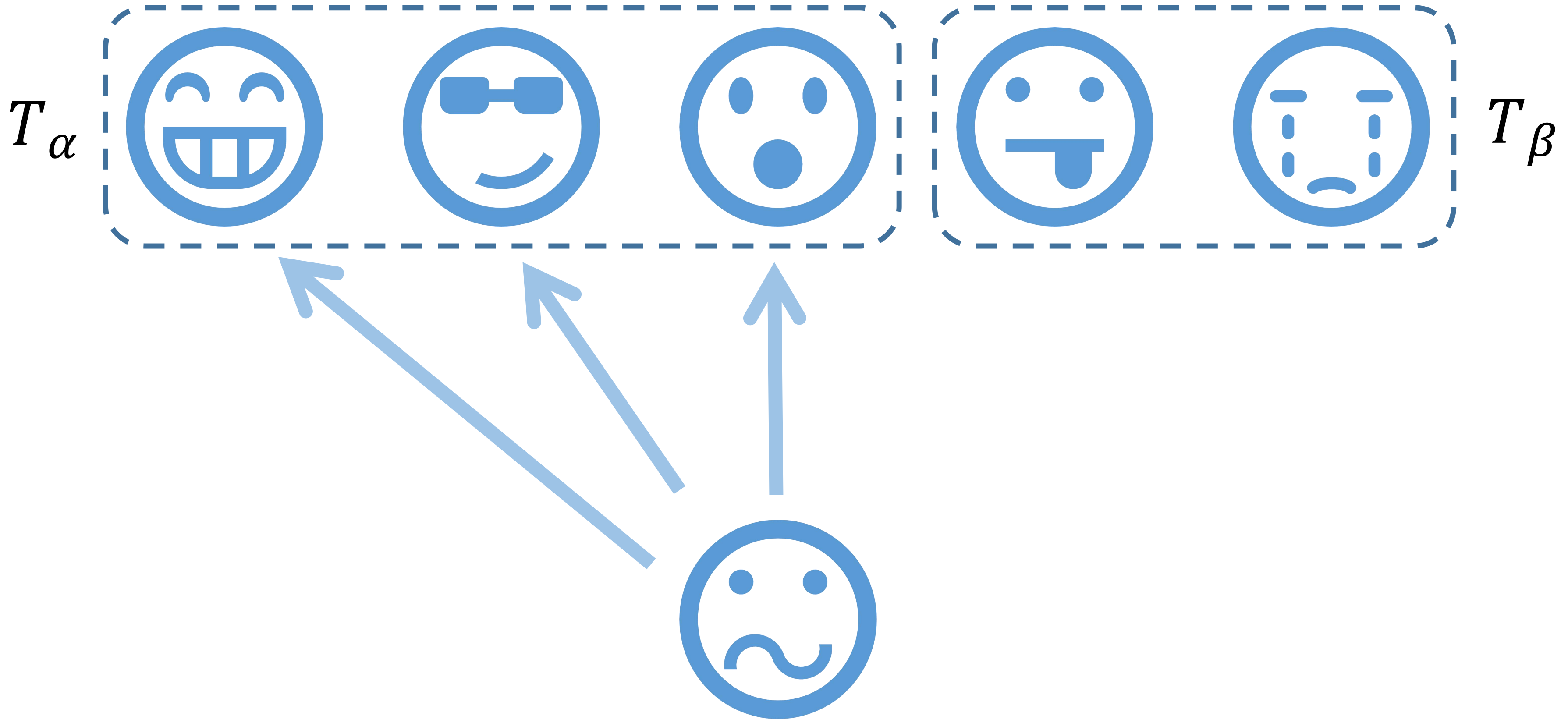
Binary Search



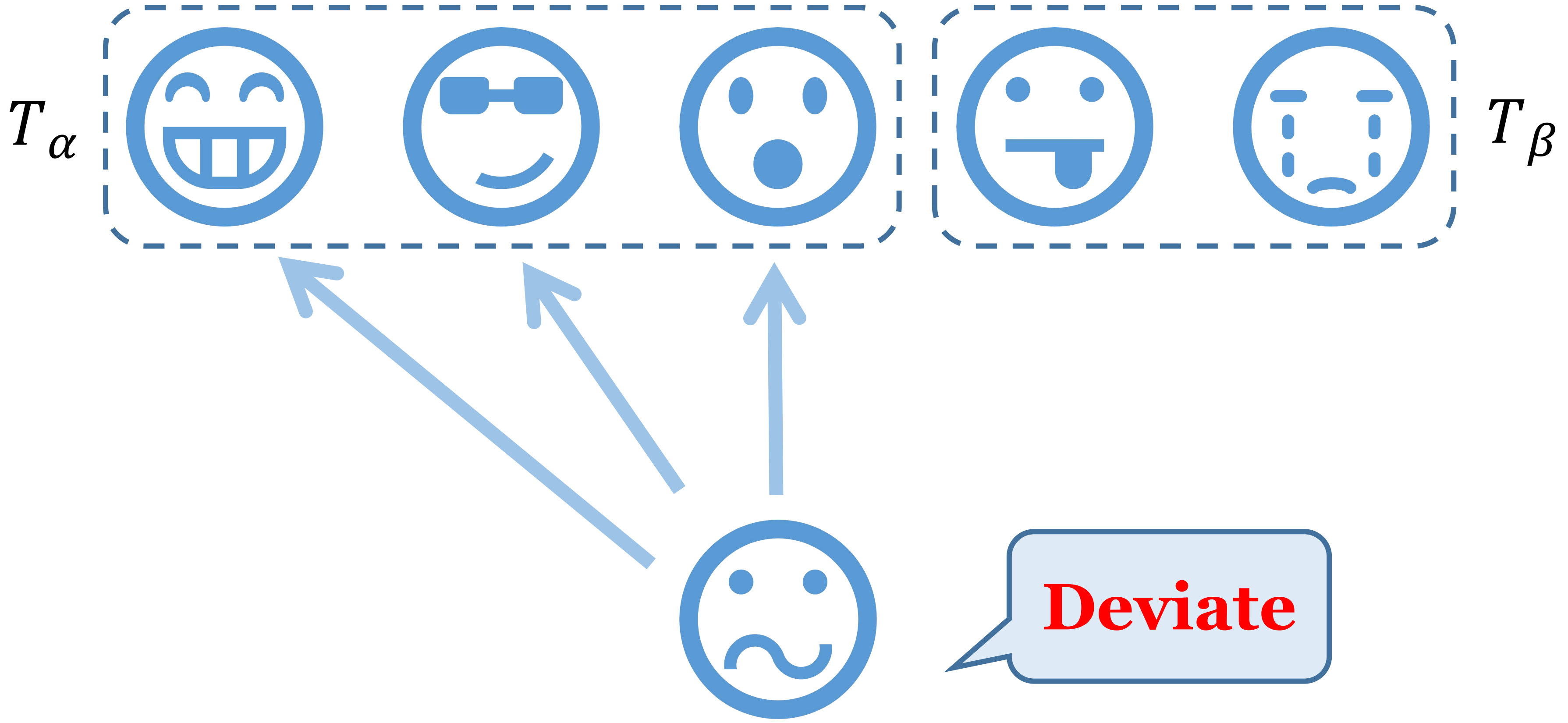
Binary Search



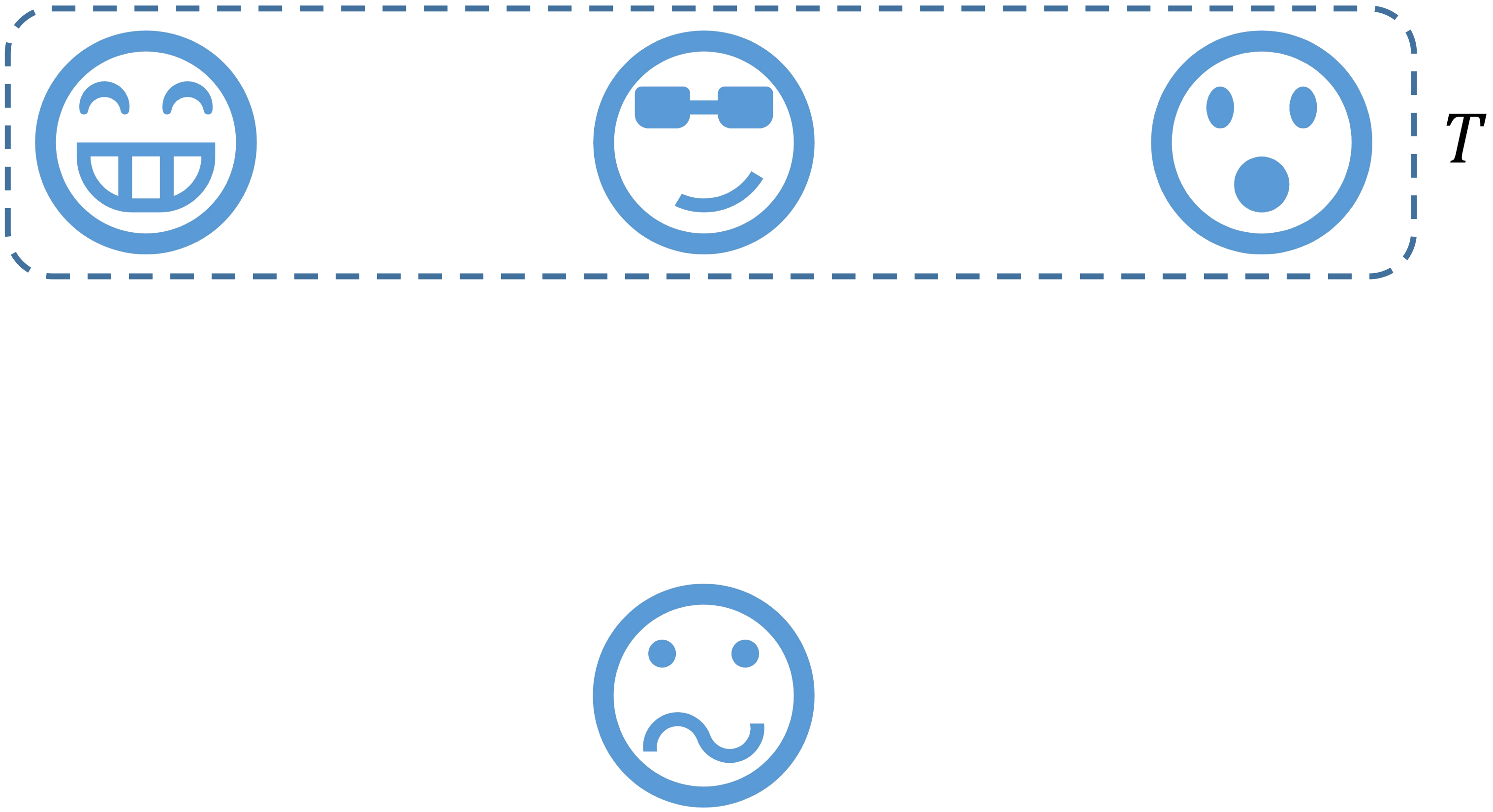
Binary Search



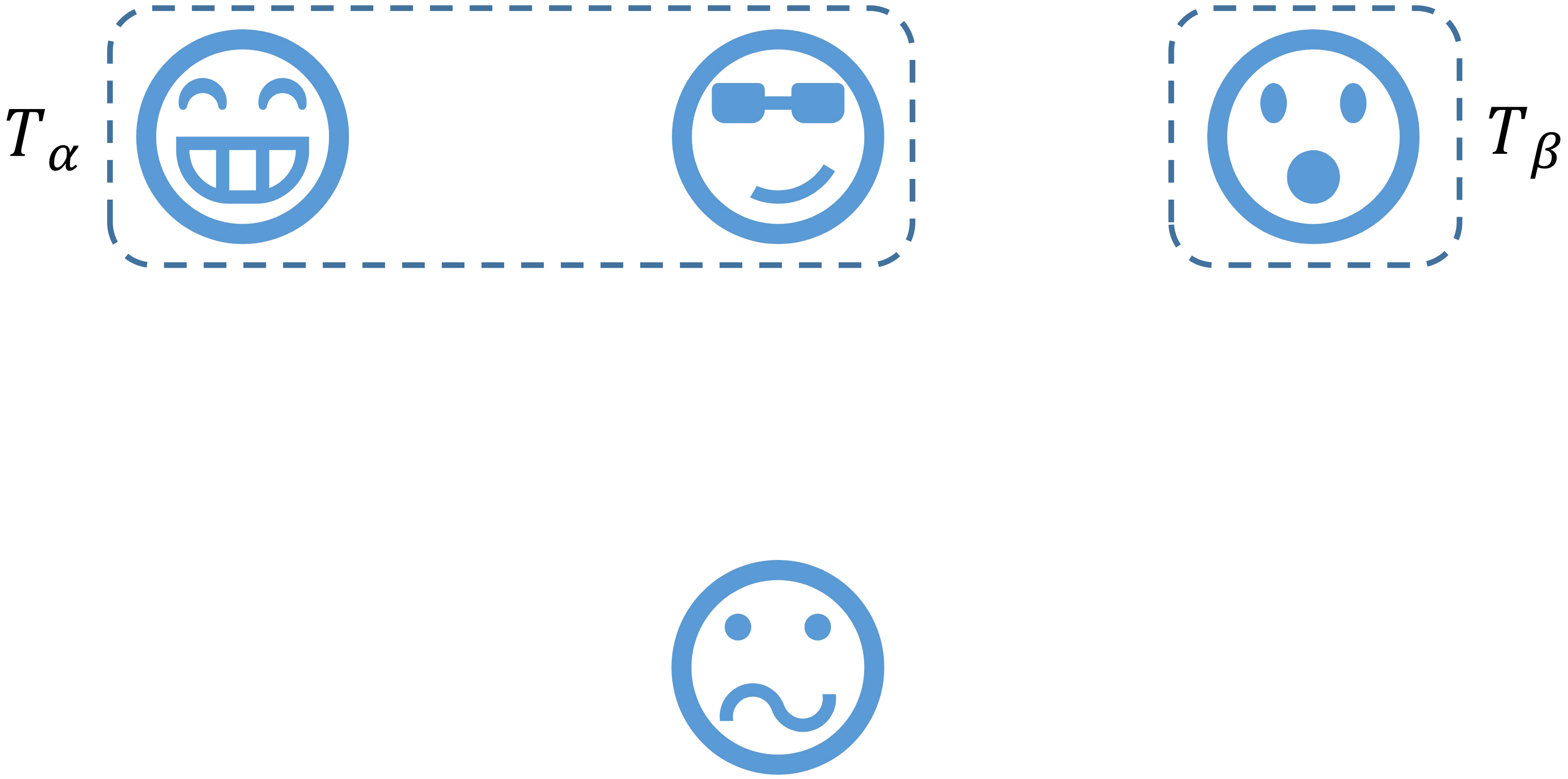
Binary Search



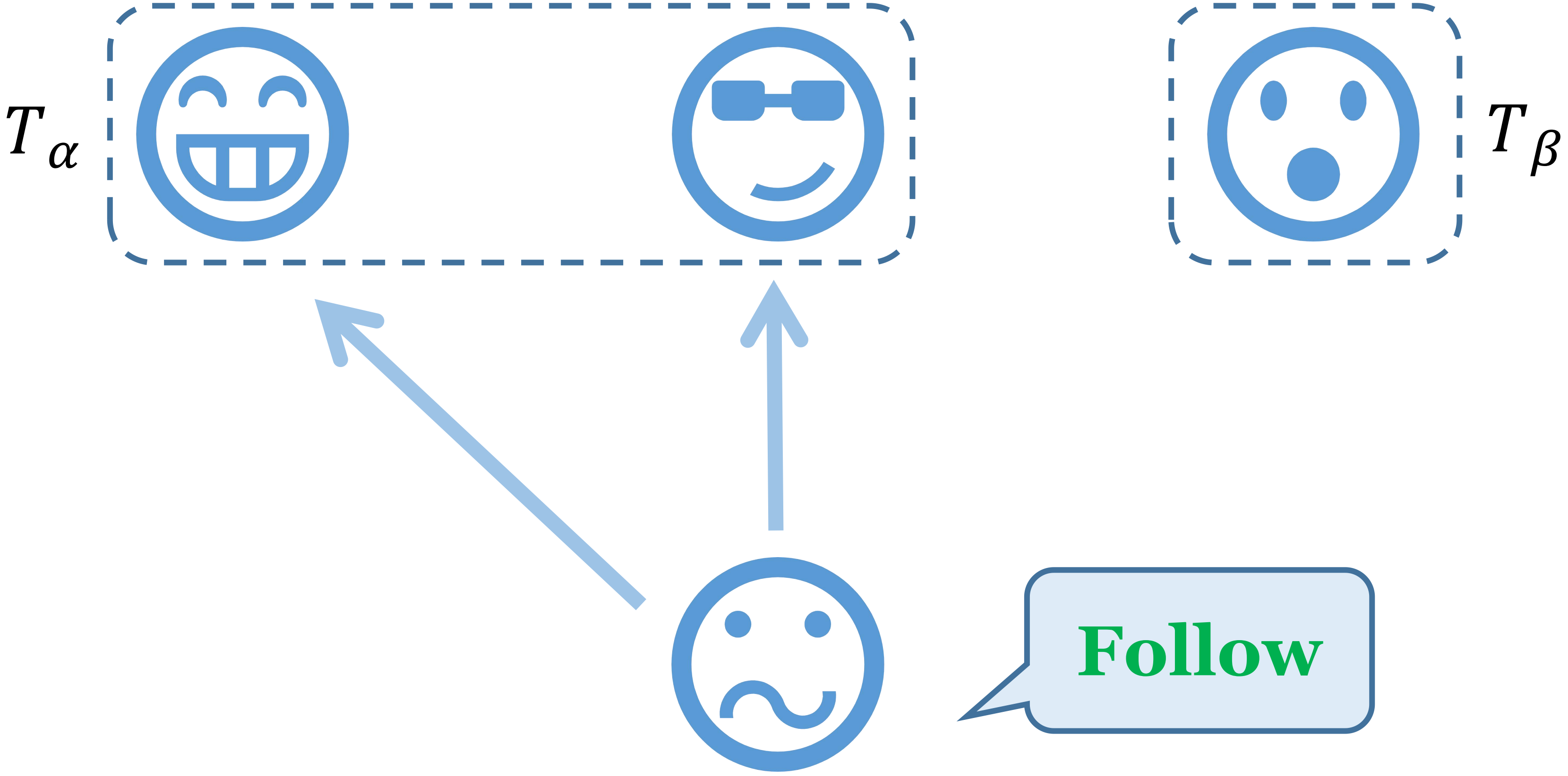
Binary Search



Binary Search

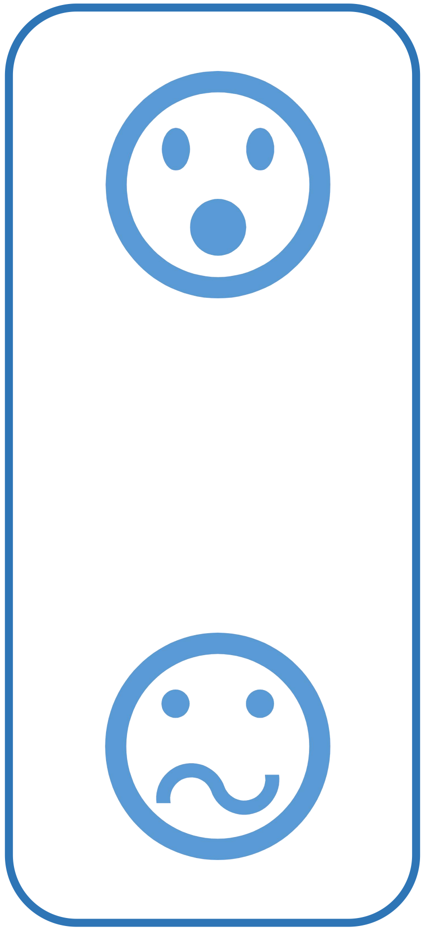


Binary Search

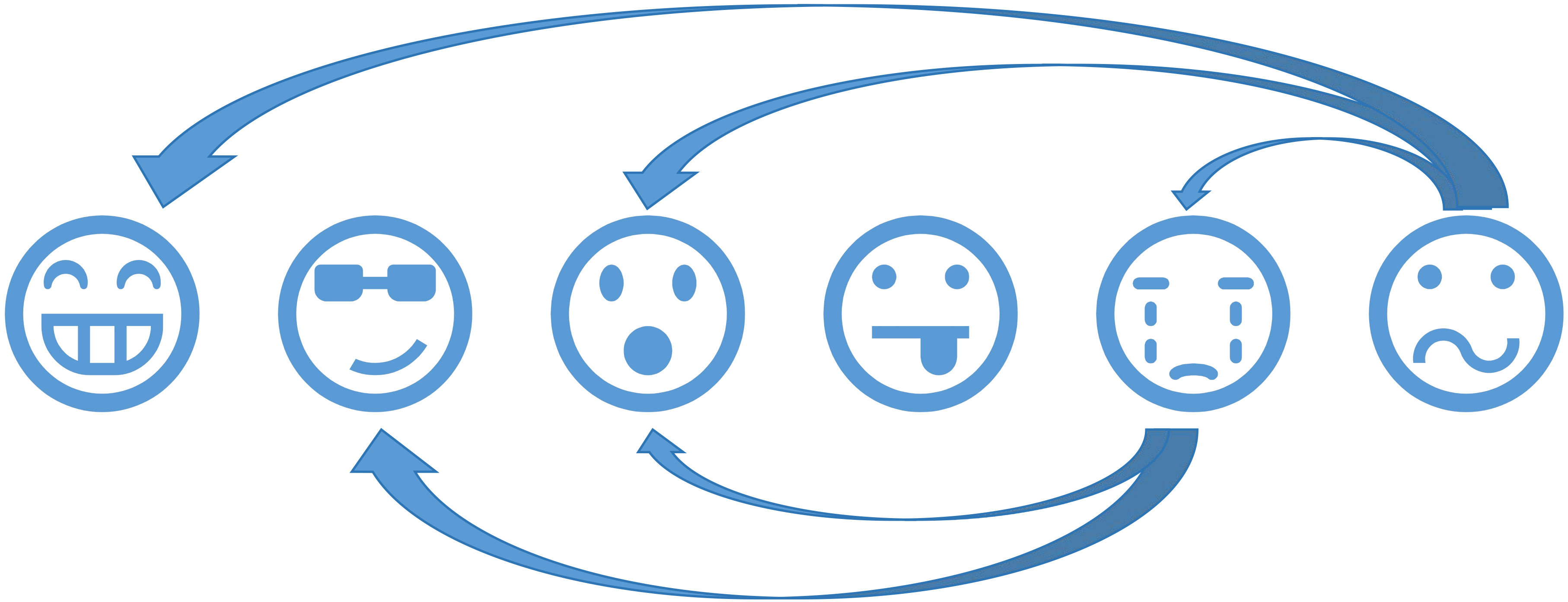


Binary Search

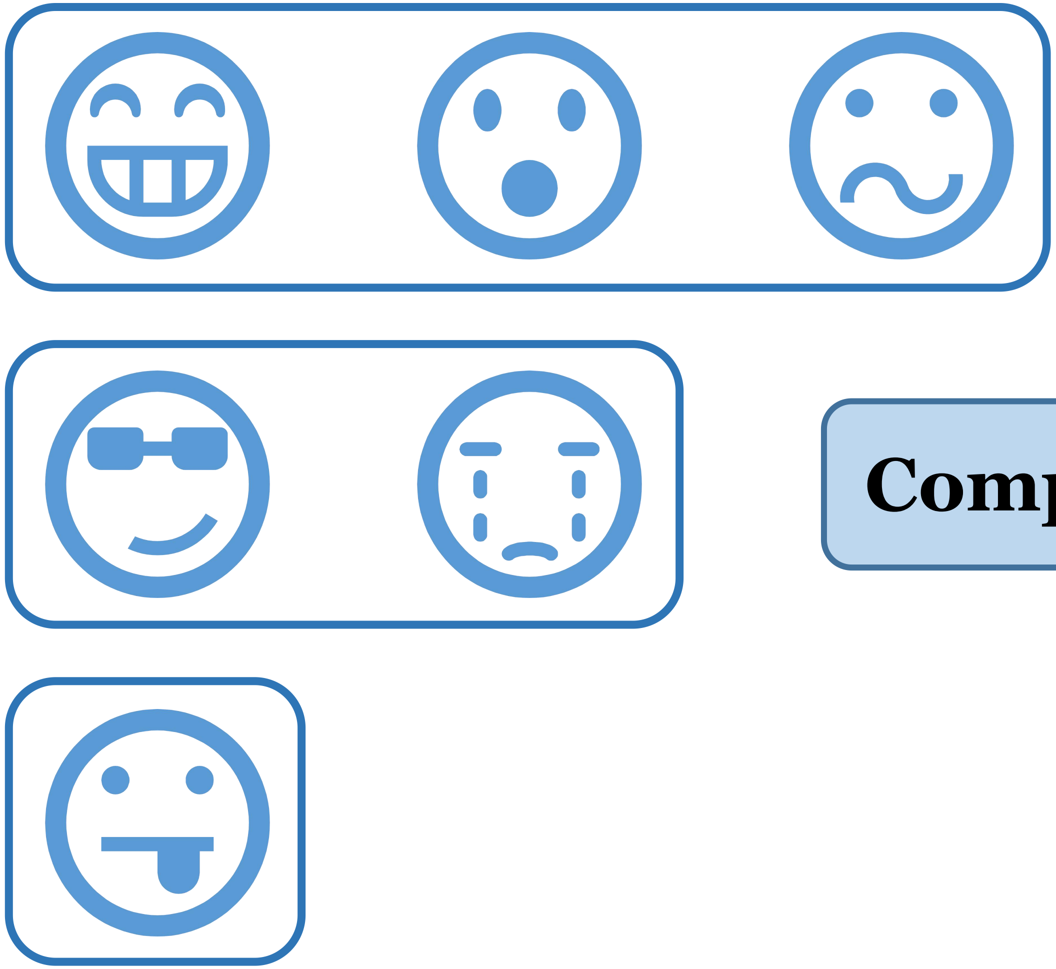
Binary Search



Simultaneous Binary Search



Simultaneous Binary Search



Complexity: $\log_2 n + 2$

Other Types of Games

	Congestion	Graphical	Auctions
Lower Bound*	$\log_2 n$	$\max(\log_2 n, n/d)$	$\log_2 n$
Algorithm	$\log_2 n + 2$	$2n/d + 2\log_2 d + 1$	$(1 + \log_2 n)(1 + c) + 1$
Technique	Directed Brass's paradox	Block decomposition	Bitwise search

- **In the above:**
  - d is the degree of the graph
  - c is the size of the largest coalition
- Mostly, the algorithm's **complexity matches the lower bound**

Our Contributions

- We study the CSL problem with **multiple-bit observations**
  - Since single-bit ones lead to **super-linear** rounds of interaction
- We conduct a thorough theoretical study
  - We present **algorithms** that learn the coalition structure with a **sub-linear** number of rounds using different types of games
  - We complement our algorithmic results with **lower bounds** that shows their **optimality** in most settings we consider

Acknowledgements: *This work was supported in part by NSF grant IIS-2046640 (CAREER) and NSF IIS-2200410.*