Paul J. Galkin
Global Health
Grand Rounds
Lecture Series

# Digitizing Hope: Using AI & Technology to Drive Global Health Equity

**Opening Remarks by Michael Littman**
Associate Provost for Artificial Intelligence

**Kirsten Langdon, PhD**
Director, Substance Abuse Disorder Program
Spring Health

**Hamish Fraser, MD**
Associate Professor of Medical Science
Associate Professor of Health Services, Policy and Practice

**Stephanie Garbern, MD**
Associate Professor of Emergency Medicine
Director of Medical Student Research

**Rochelle Rosen, PhD**
Associate Professor of Psychiatry and Human Behavior
Associate Professor of Behavioral and Social Sciences

## Tuesday, September 30th

The Warren Alpert Medical School | Room 280
5 - 7 pm; Dinner provided.
Zoom option available.

BROWN UNIVERSITY
Center for Digital Health

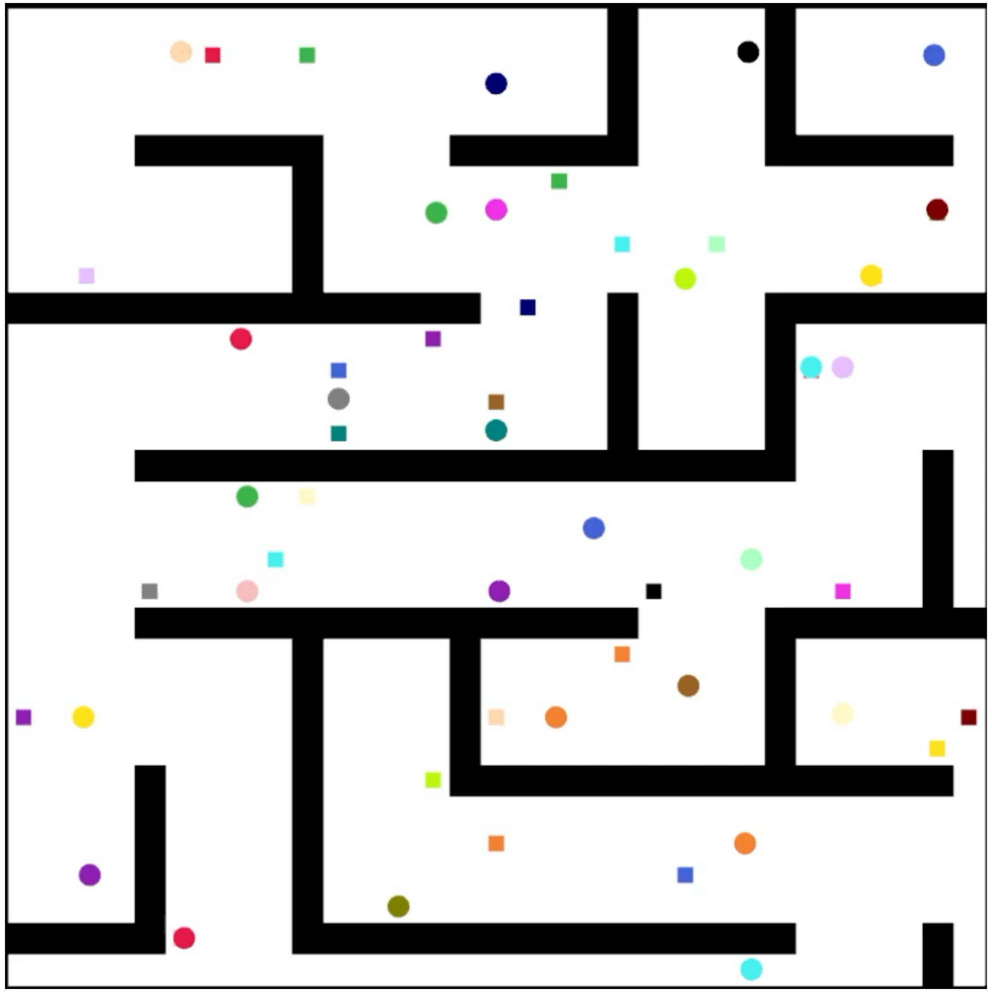
BROWN
Division of Biology and Medicine
CENTER FOR GLOBAL HEALTH EQUITY

Funding support provided by the Arnold T. Galkin Fund in memory of Paul J. Galkin. RSVP with QR code.

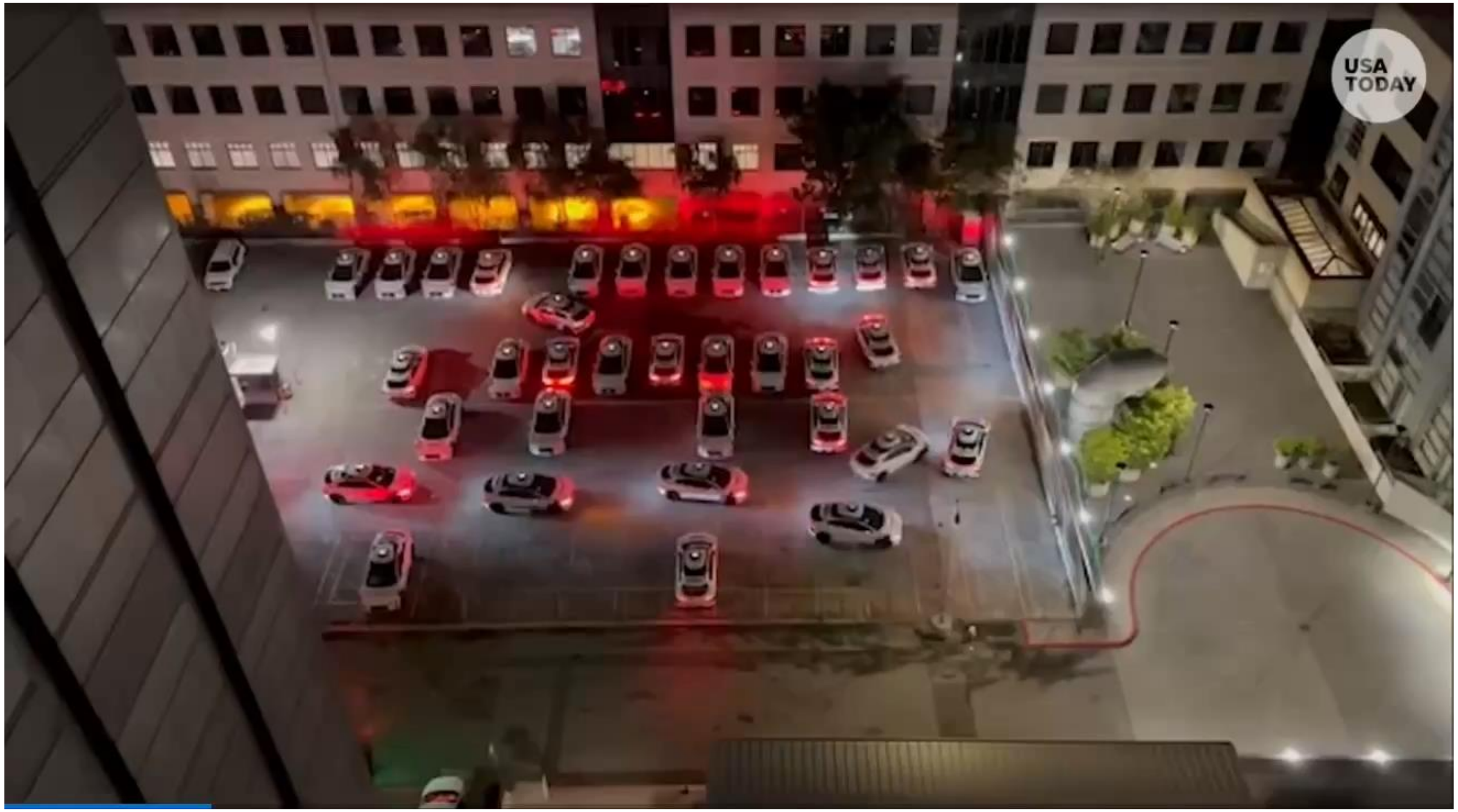
# Multi-Agent Pathfinding (MAPF)

Problem: Find *non-conflicting* paths for n agents on a graph

Objective: Minimize total travel time



# Coordination is hard even in *controlled environments*



Source: https://www.usatoday.com/story/tech/2024/08/15/waymo-driverless-cars-honking-parking-lot-video/74810195007/

# Multi-Robot Coordination

What's currently missing?

- Trust between agents
- Communication between agents
- Centralized authority

*Even with nice assumptions, multi-robot coordination is **hard***

Source: TED, Wanis Kabbaj, 2016

# Methods to Solve MAPF Problems

1. Formulate as search problem

   Action space is all possible combinations of actions agents can take at a certain timestep

2. Formulate as discrete optimization problem

   Use large neighborhood + local search

3. Formulate as Boolean Satisfiability problem
4. Formulate as Mixed-Integer Program

**Algorithm 1** GSAT Algorithm

```
procedure GSAT($\phi, N, M, p$)
    Input: CNF formula $\phi$, number of restarts $N$, number of trials per restart $M$, probability $p$
    Output: satisfying assignment $v$ or FAIL

    for $i = 1$ to $N$ do
        initialize random assignment $m$
        for $j = 1$ to $M$ do
            if $m$ satisfies $\phi$ then
                return $m$
            end if
            generate random number $r \in [0, 1]$
            if $r < p$ then
                flip random variable in $m$
            else
                flip variable that minimizes number of remaining unsatisfied clauses
            end if
        end for
    end for
    return FAIL
end procedure
```

**Algorithm 2** WalkSAT Algorithm

```
procedure WalkSAT($\phi, N, M, p$)
    Input: CNF formula $\phi$, number of restarts $N$, number of trials per restart $M$, probability $p$
    Output: satisfying assignment $v$ or FAIL

    for $i = 1$ to $N$ do
        initialize random assignment $v$
        for $j = 1$ to $M$ do
            if $v$ satisfies $\phi$ then
                return $v$
            end if
            choose unsatisfied clause $C \in \phi$ at random
            generate random number $r \in [0, 1]$
            if $r < p$ then
                choose variable $x \in C$ that minimizes number of remaining unsatisfied clauses
            else
                choose variable $x \in C$ at random
            end if
            $v \leftarrow v$ with bit $x$ flipped
        end for
    end for
    return FAIL
end procedure
```

# N-Queens

Queens can move along rows, columns or diagonals.

Can you fit N Queens on an NxN chess board?