



Sole Agent Syndrome and Machine Learning for Human-like Behavior

Russell Toris, Director of Robotics, Fetch Robotics

Robots in the Workplace has Shifted

Robotics has become a collaborative tool
Robots *themselves* need to be as collaborative



Traditional Robotics Deals With Optimization

Robotic automation in industry is driven by:



Repeatability



Scalability



Increased Throughput

Traditionally achieved with:

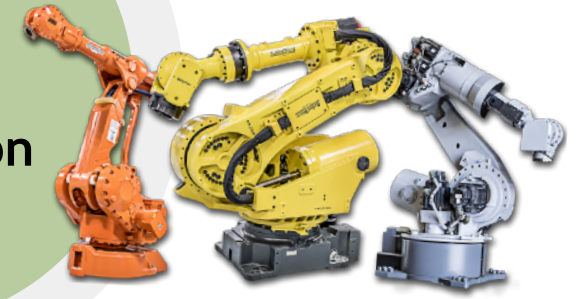
Enclosed Work Cells



Static
World
Assumption



Pre-Defined Motions



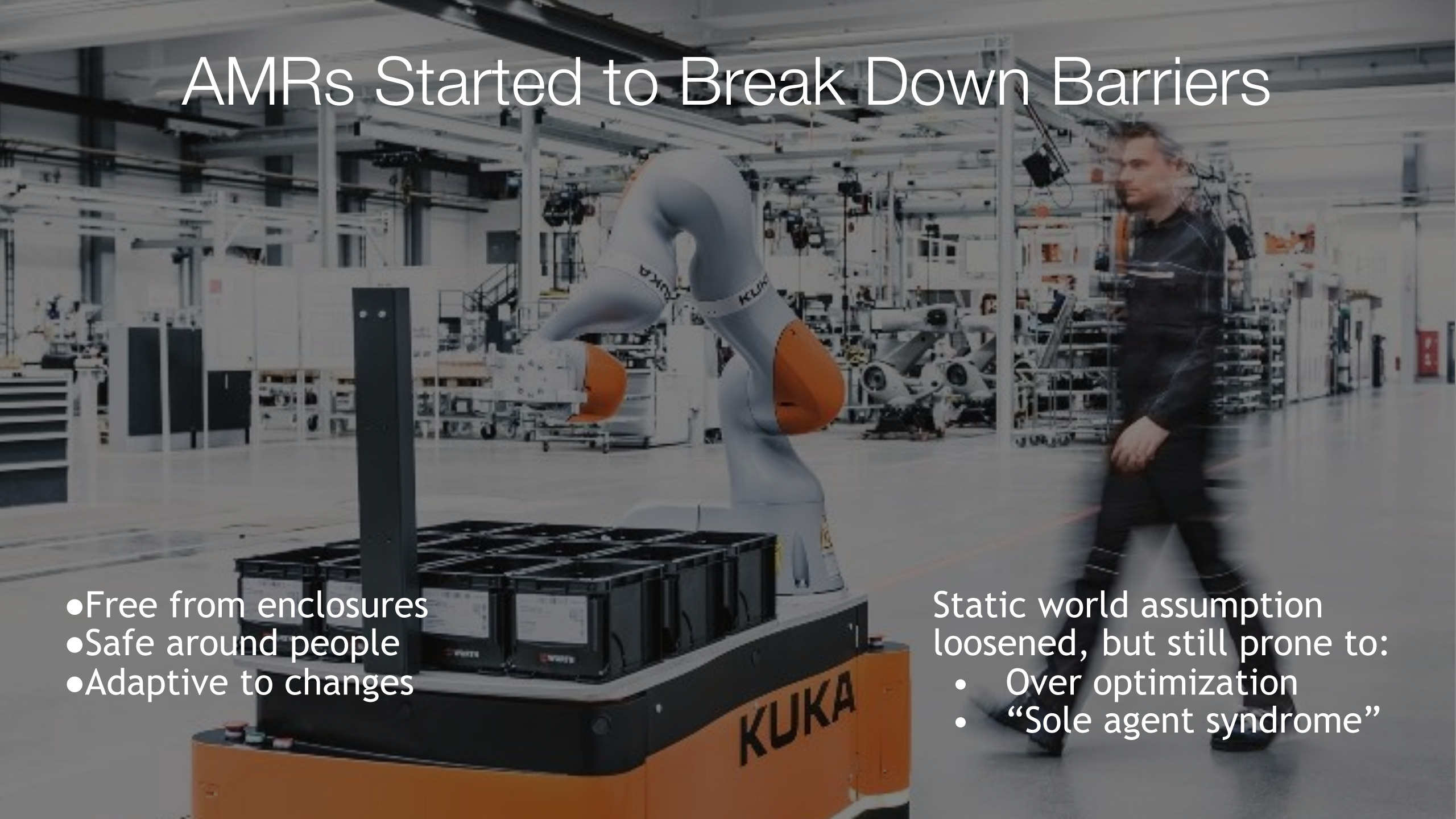
Over Optimization

AMRs Started to Break Down Barriers

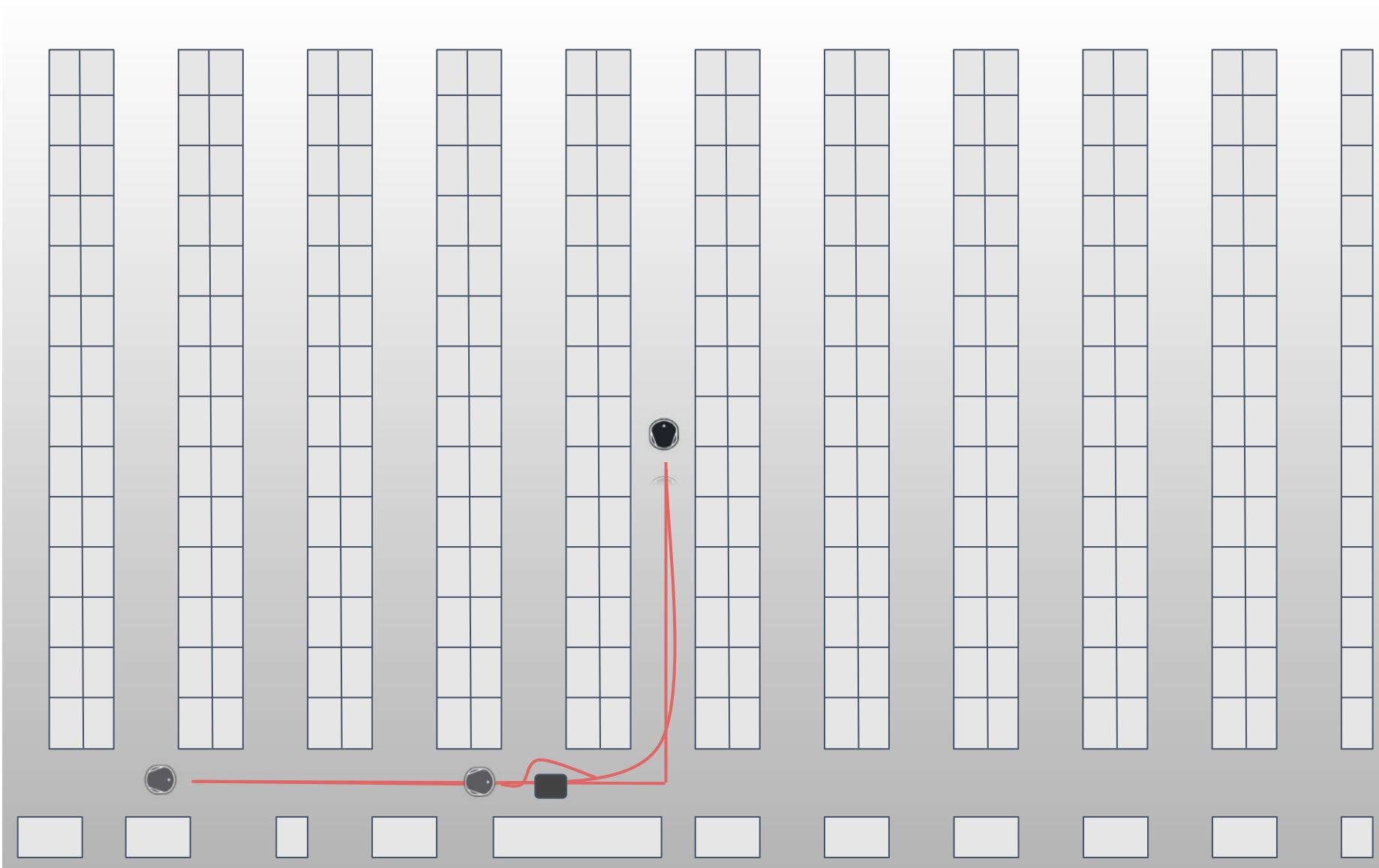
- Free from enclosures
- Safe around people
- Adaptive to changes

Static world assumption loosened, but still prone to:

- Over optimization
- “Sole agent syndrome”



Traditional AMR Workflow



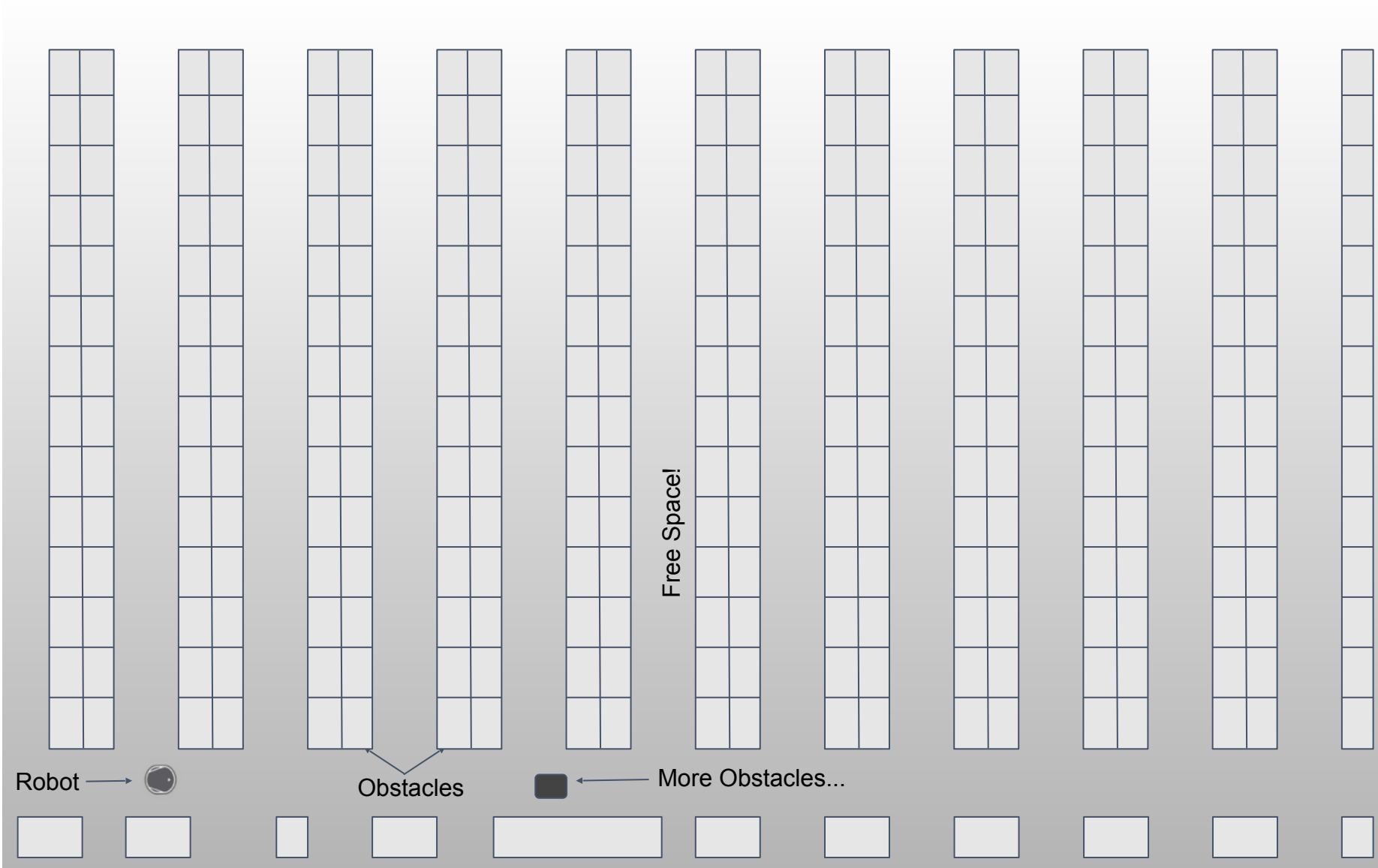
Classic search problem (A*)

Add in optimizations

Execute the motion

Adapt as needed

What is “Sole Agent Syndrome?”

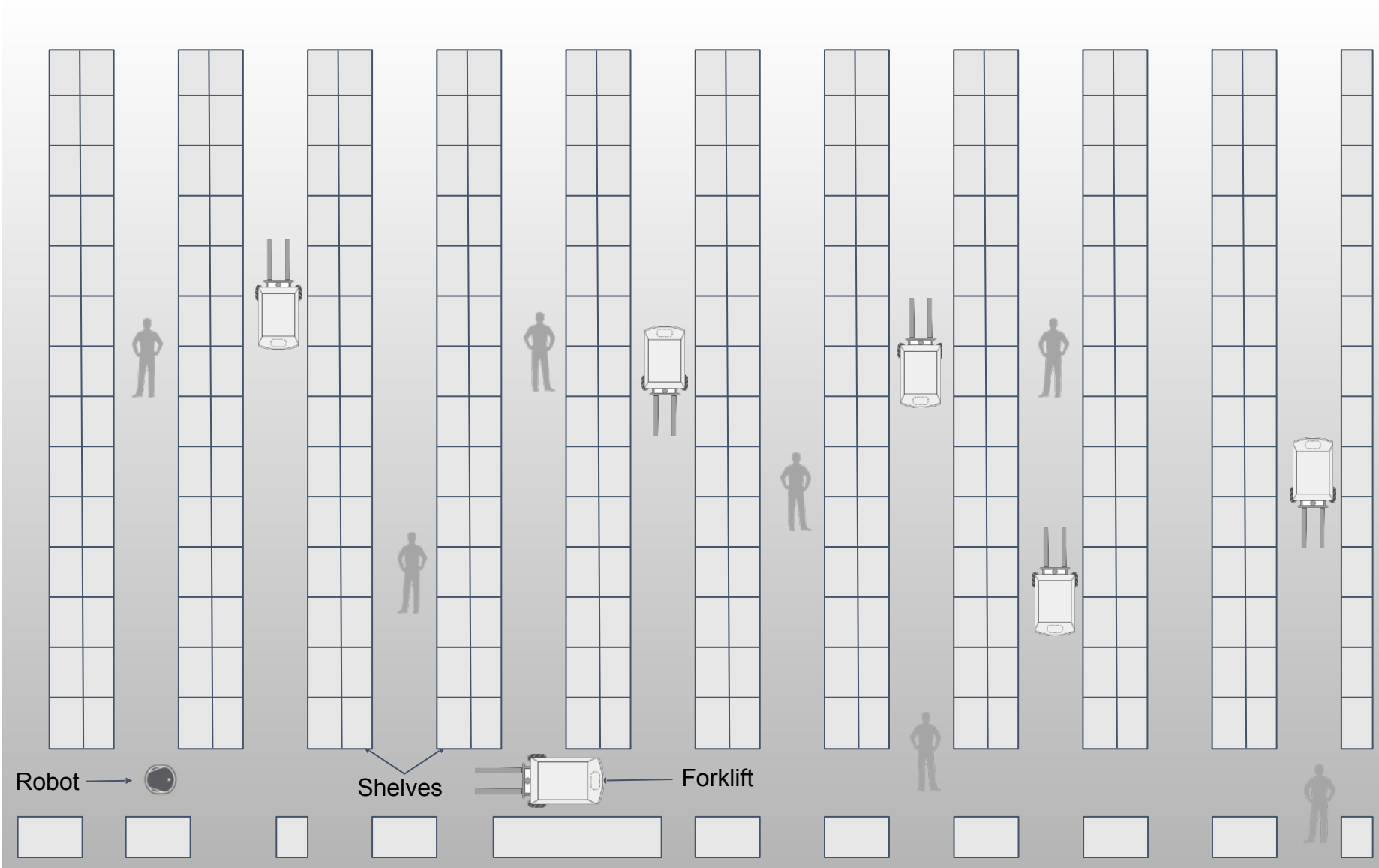


A view from a *robot's* mind

The world is either:

- Myself
- Obstacle
- Free space

What is “Sole Agent Syndrome?”

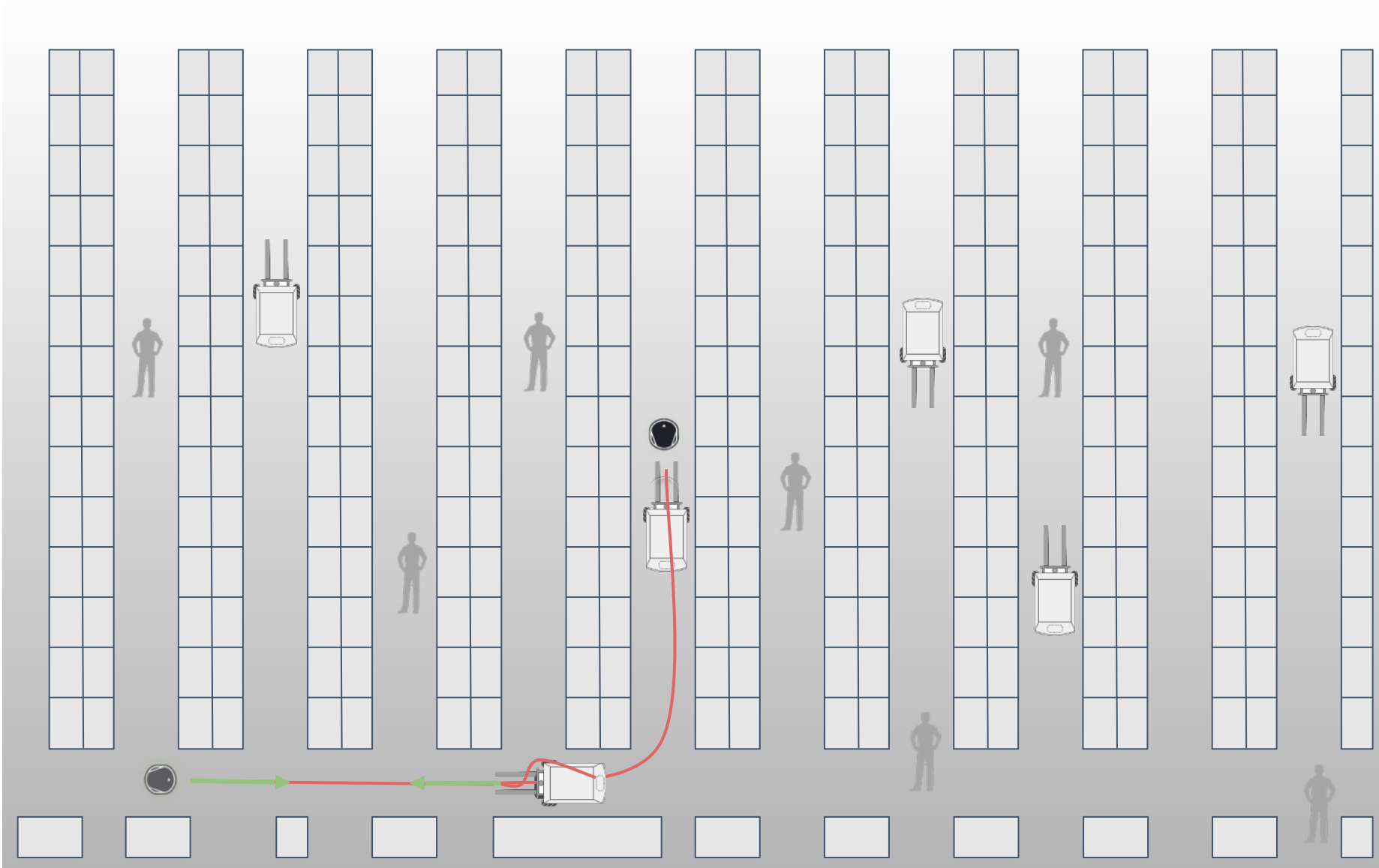


A view from a *human's* mind

The world is:

- Myself
- Automation equipment
- Fixed infrastructure
- Other workers
- Many unknowns

What is “Sole Agent Syndrome?”



A view from a *human's* mind

The robot is:

- “Ignoring me”
- “Getting to close”
- “Too aggressive”
- “Acting drunk”

A composite image showing a man on the left and a robot on the right, separated by a diagonal line. The man, wearing safety glasses and a dark sweater, is in a ready stance holding a baseball bat. The robot is a white, humanoid figure with blue and teal accents, standing in a warehouse setting. The background includes yellow caution tape and industrial equipment.


Sole Agent Syndrome Leads To...

People having feelings about the robot such as:

- Intimidated
- Hesitant
- Lack of Confidence

Ultimately this leads to:

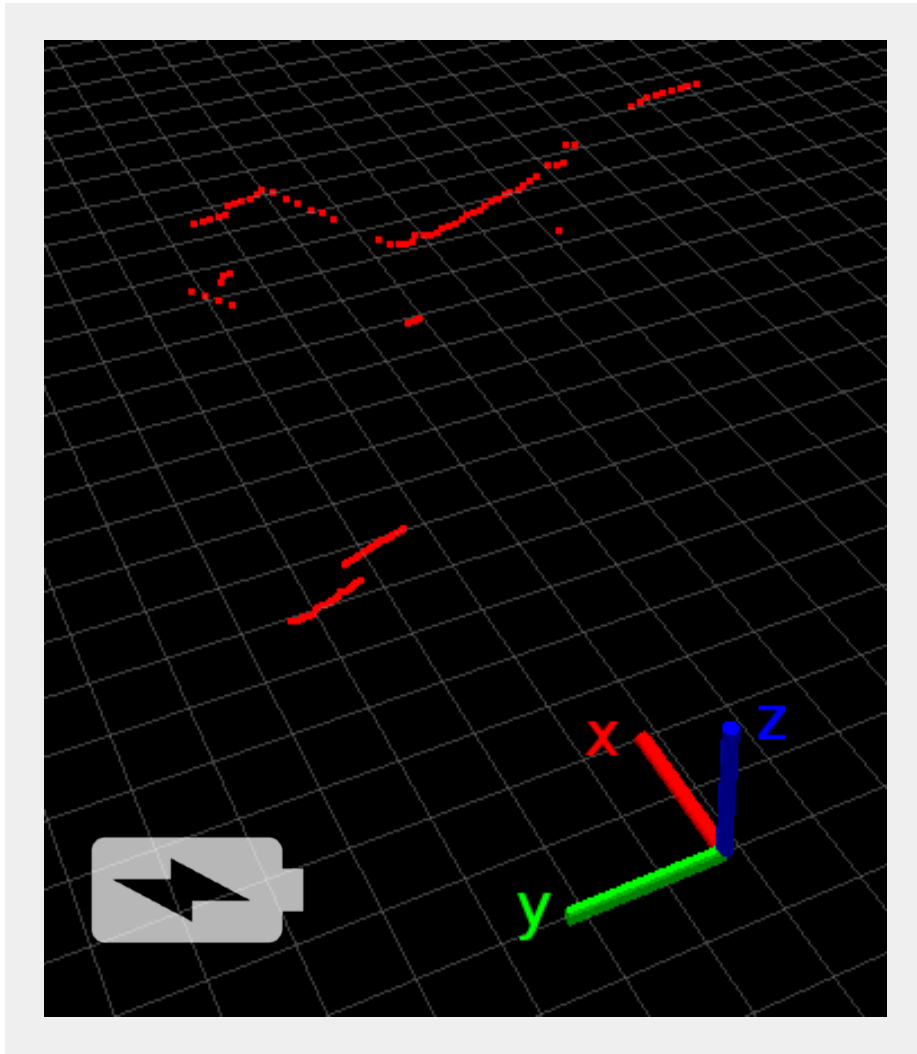
- Bad Preconceptions
- Underutilization
- Slower ROI



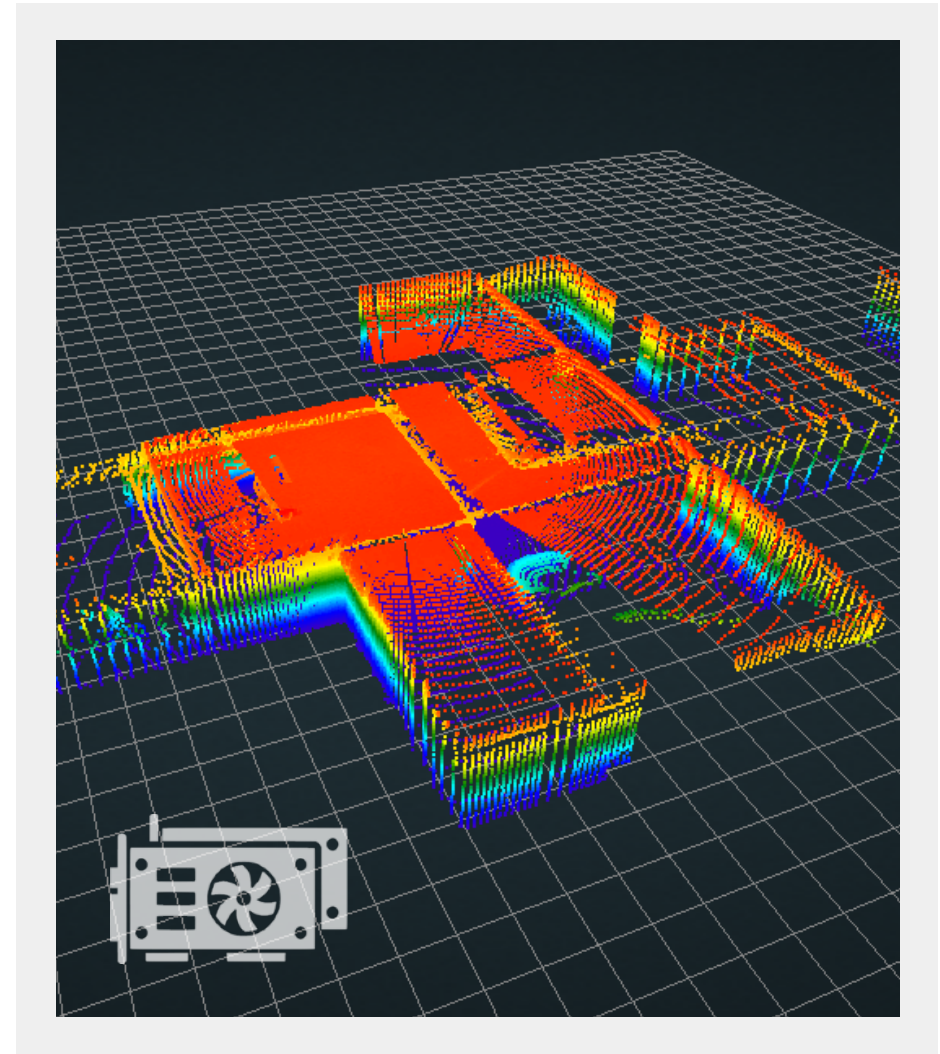
Machine Learning for More Human-like Behavior

Phase 1: Perception

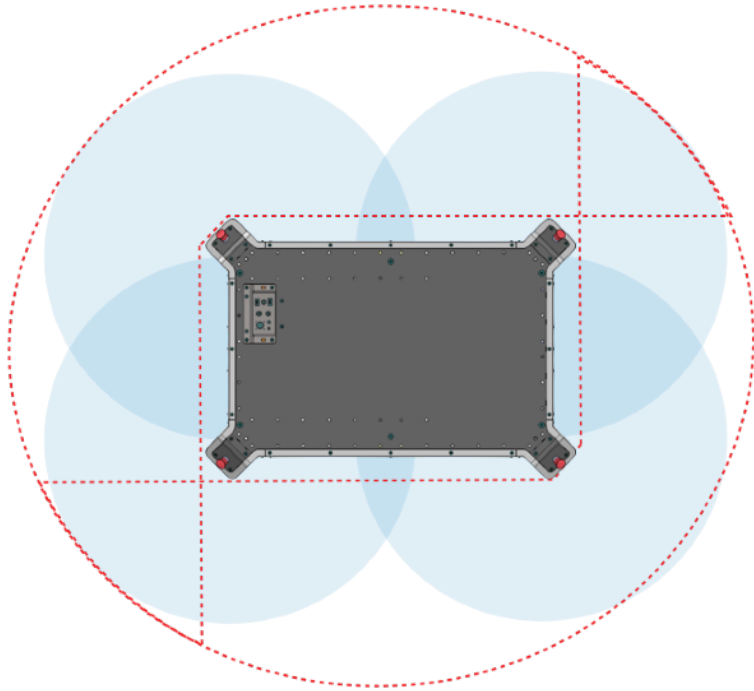
Tradeoffs in Performance



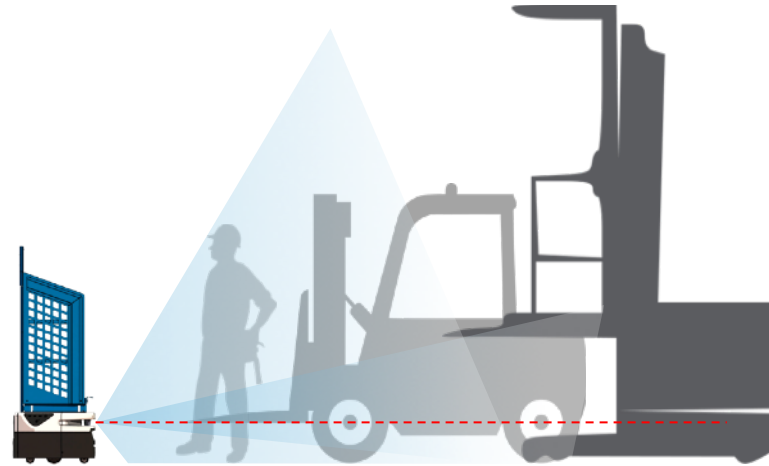
VS



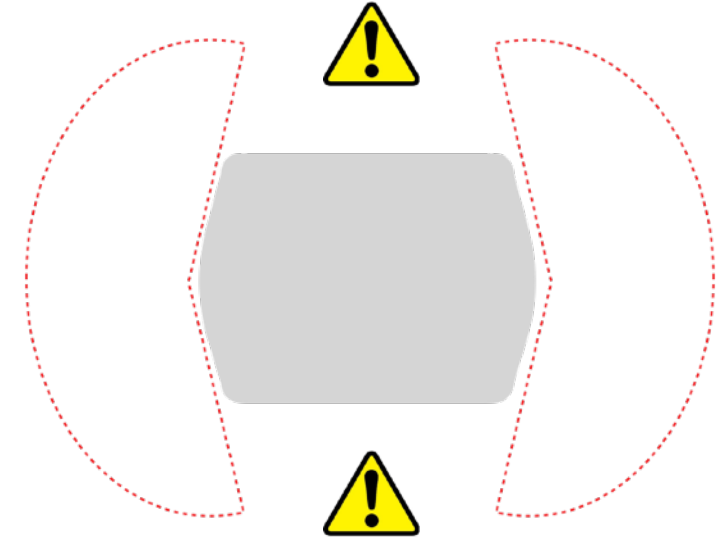
Tradeoffs in Performance



Safety Critical
Processes Need
Priority

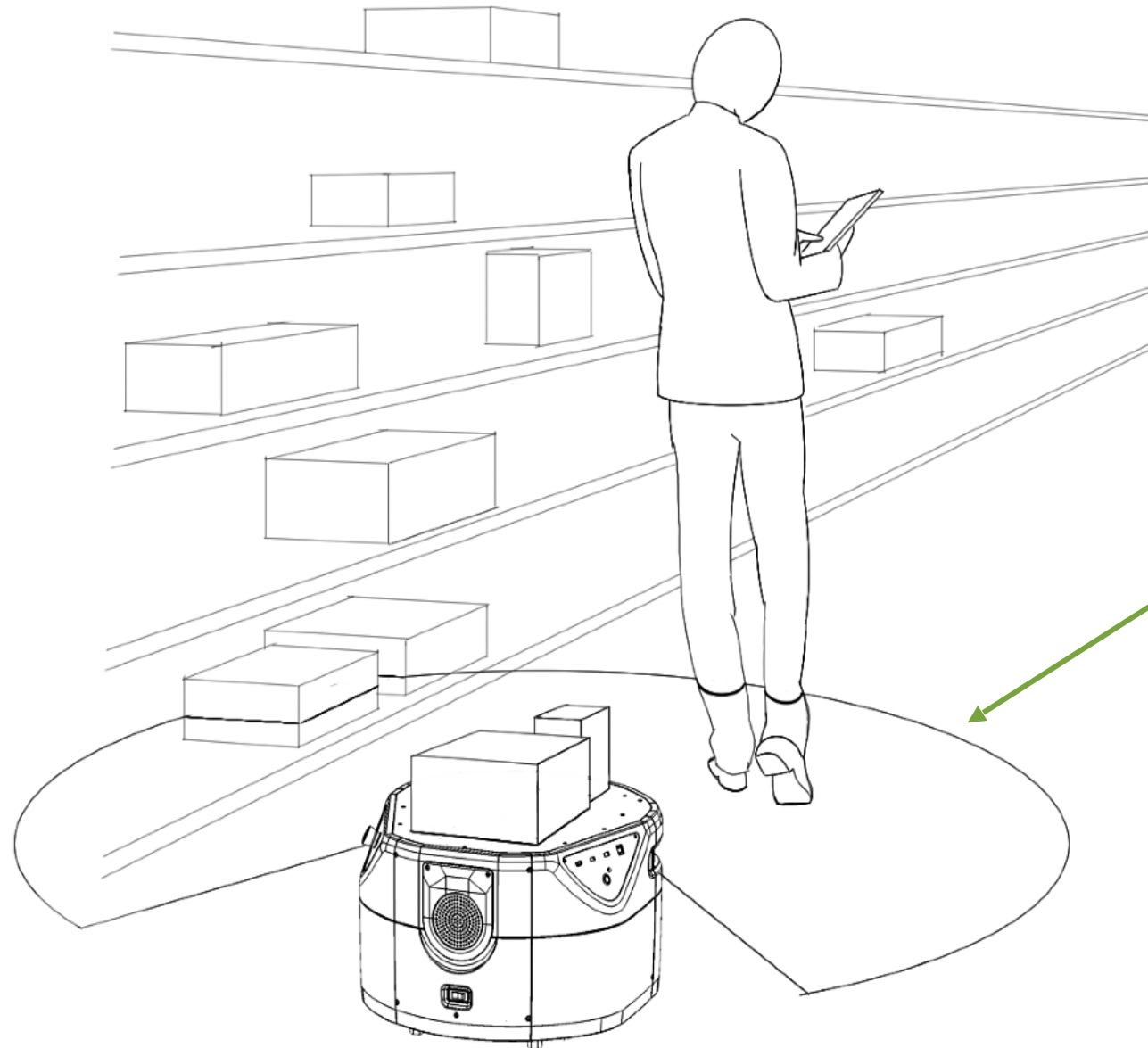


Limited Resources Left
for Perception
Computation



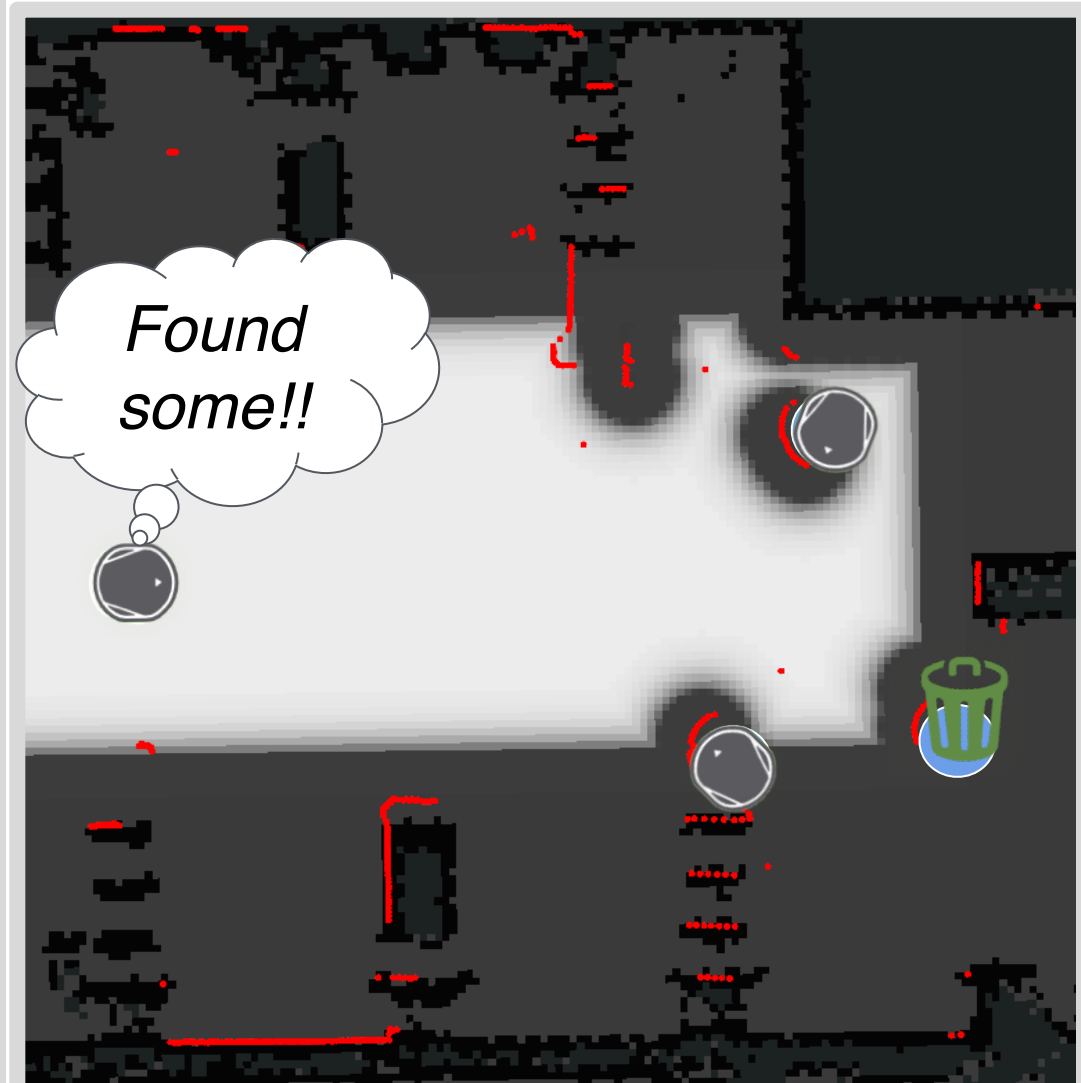
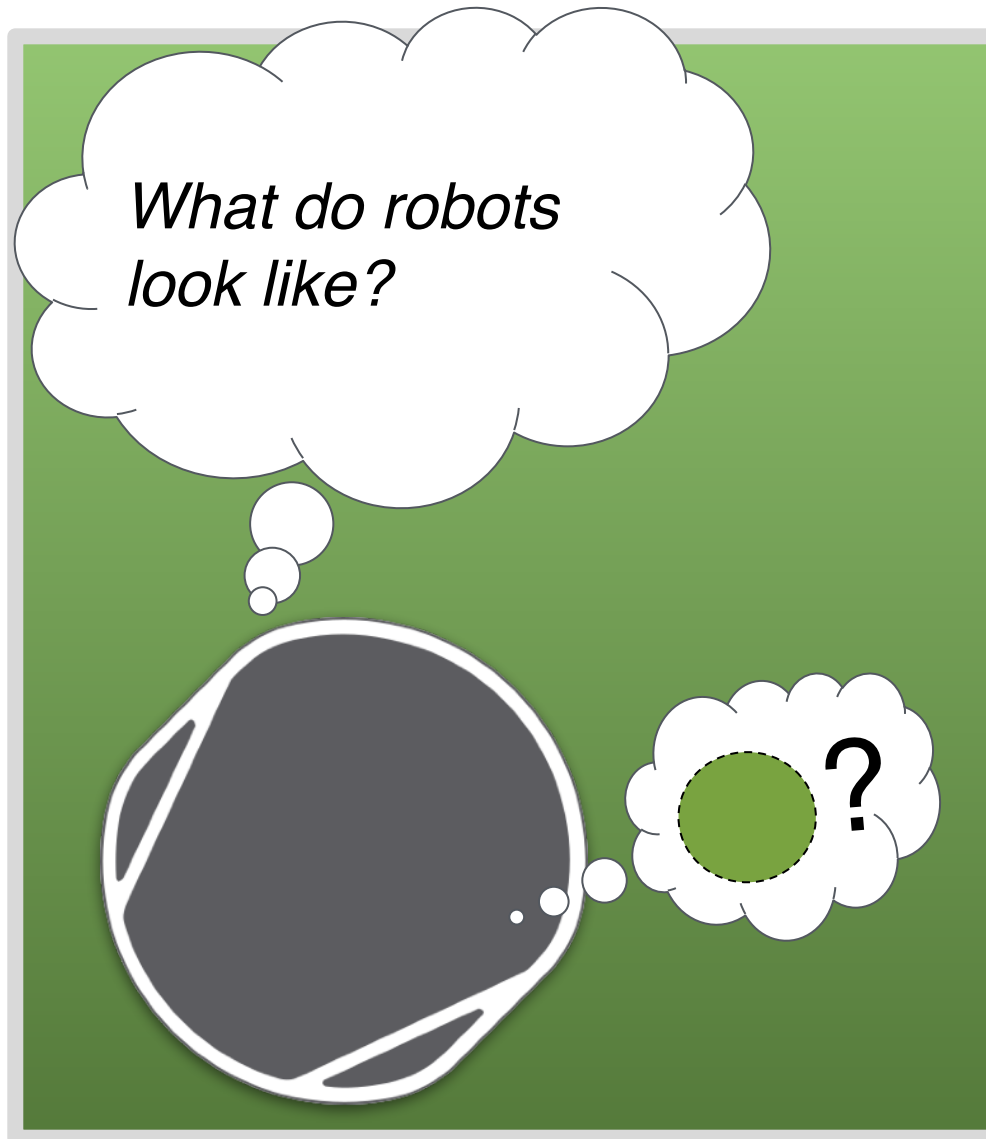
Utilizing 2D
Significantly Faster

What can be done with limited information?



Small Slice =
Limited Information

Traditional Methods (ICP)



Detecting Robots with Convolutional Neural Nets

Well developed open-source tools:

- TensorFlow, Theano, Keras, Caffe, etc
- GPU/CPU interfacing is seamless
- Modeling flexibility, TensorFlow C++ support



theano

Detecting Robots with Convolutional Neural Nets

Segmentation-free

- Classify using local region of full-scan
- No need to cluster before classification step

Train on GPU, reference on CPU

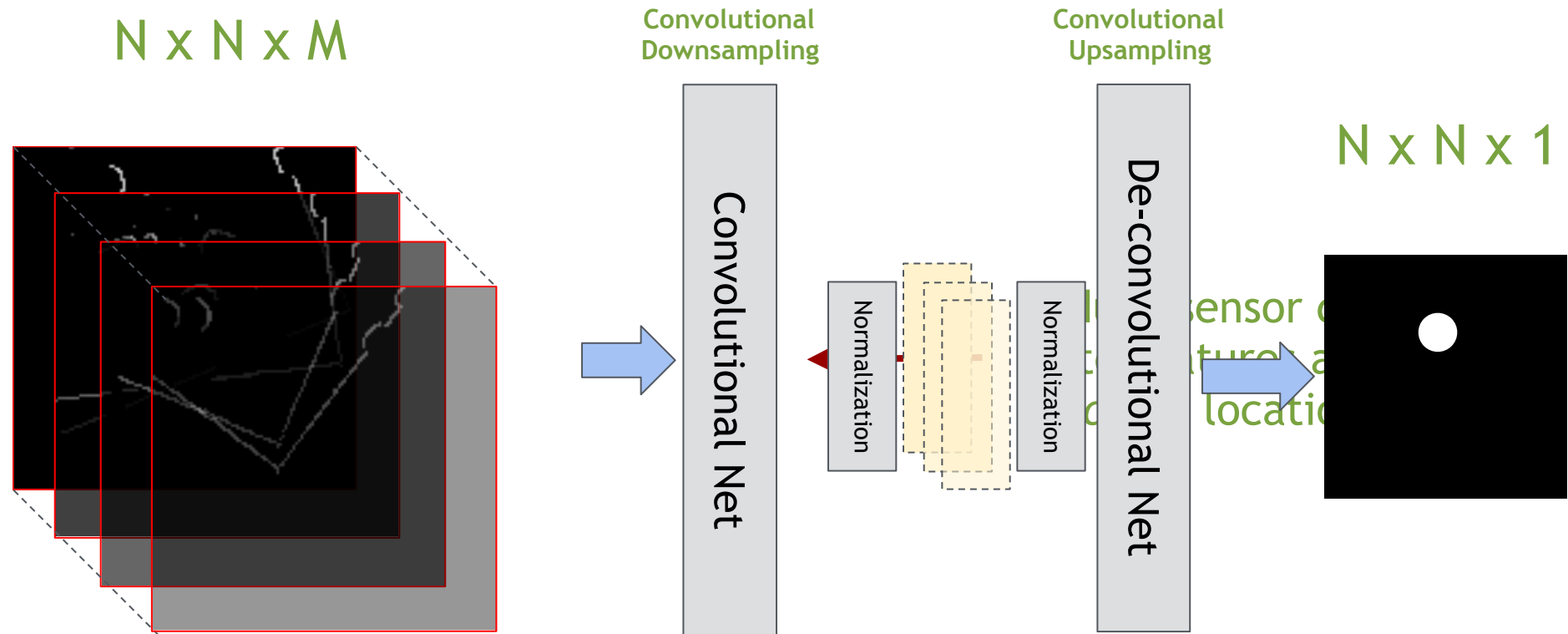
- Do not need to have a GPU on runtime hardware

Fixed computational burden

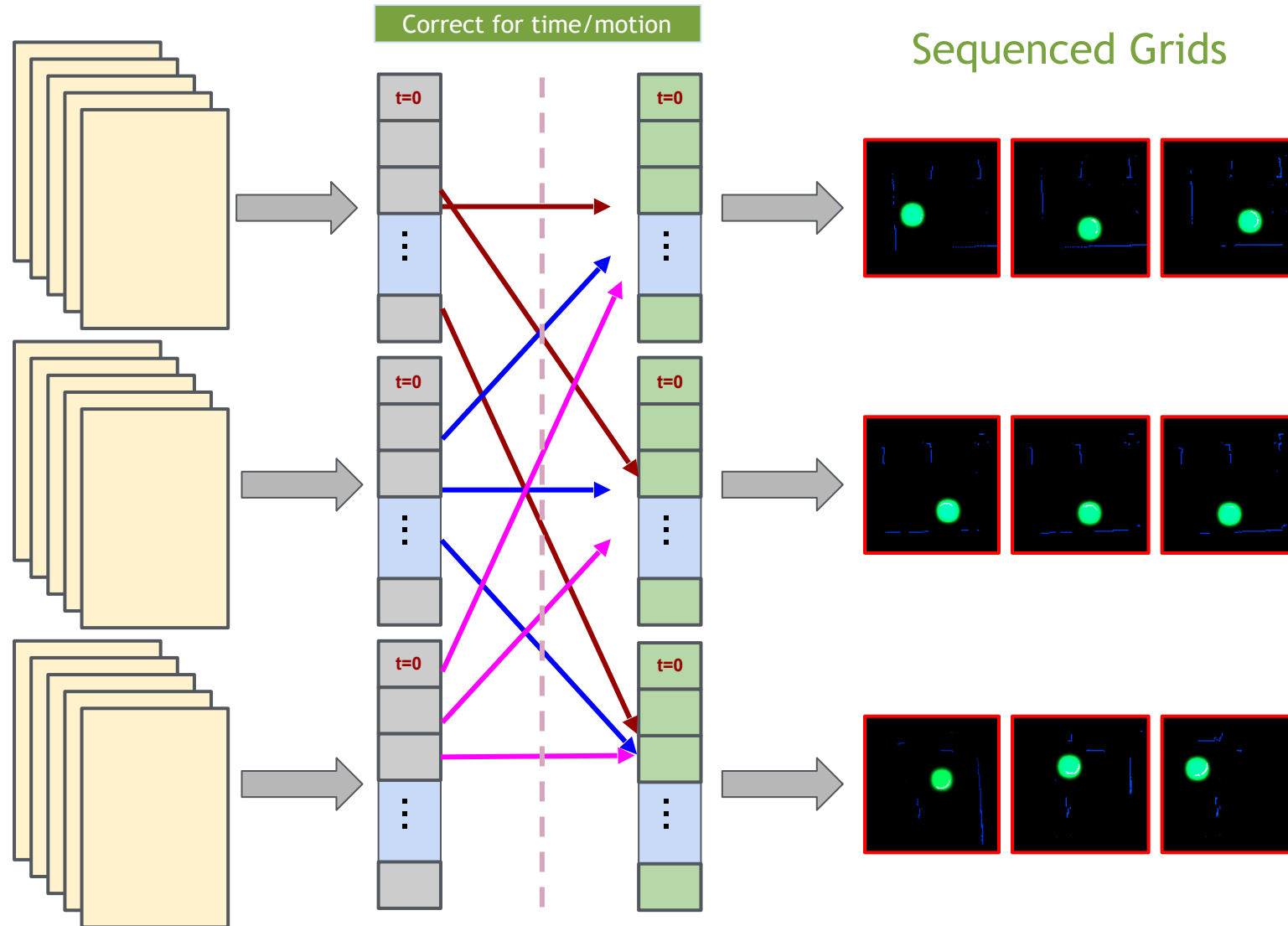
- Fixed sized input/output
- Fixed number of multiplications per prediction

Detecting Robots with Convolutional Neural Nets

Sensor Data *Over Time*




Offline Training with Large Data Sets



Detecting Agents with Convolutional Neural Nets

Same can be used for:





Machine Learning for More Human-like Behavior

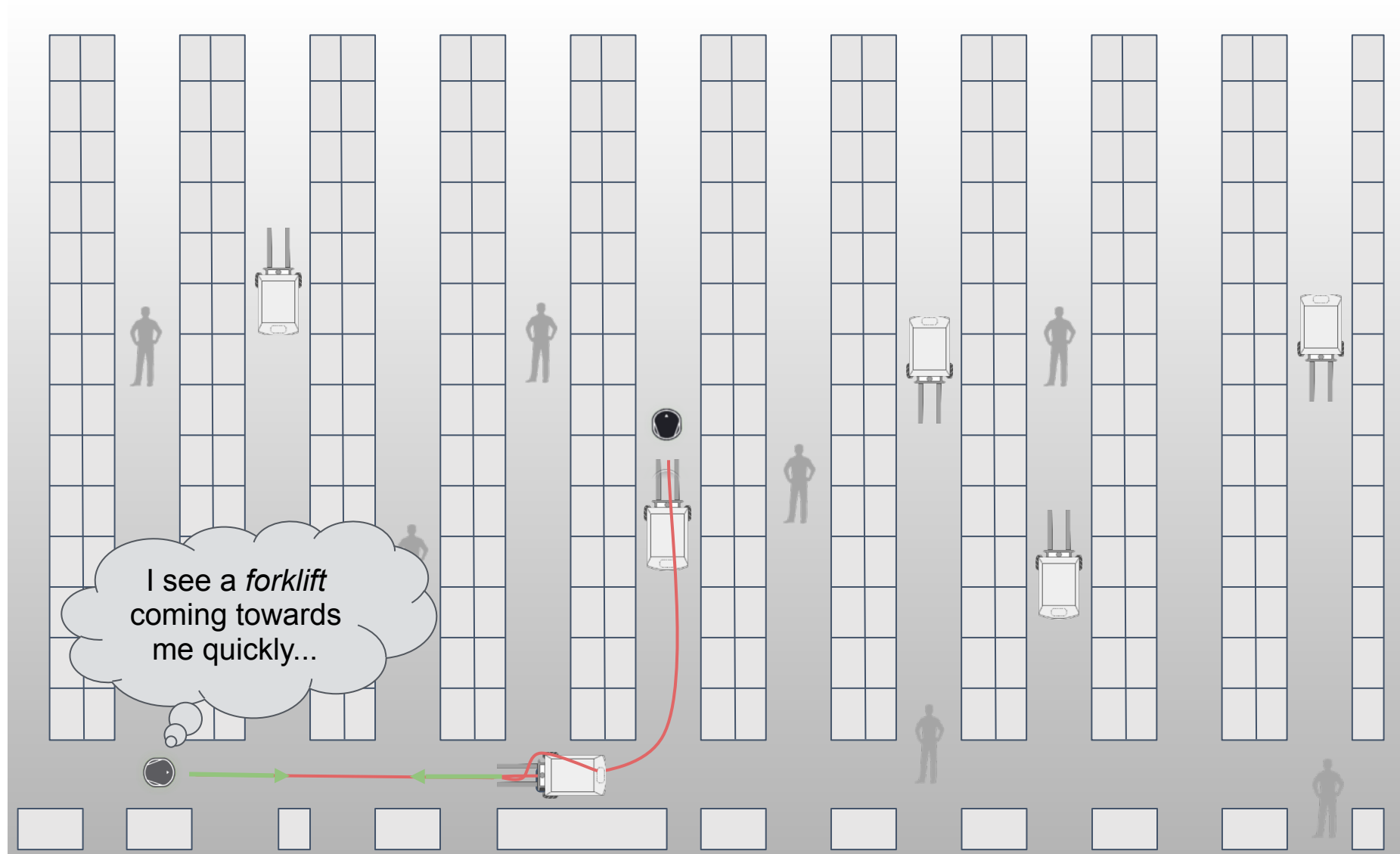
Phase 2: Action



Machine Learning for More Human-like Behavior

Training done on
motion models, not
shape characteristics

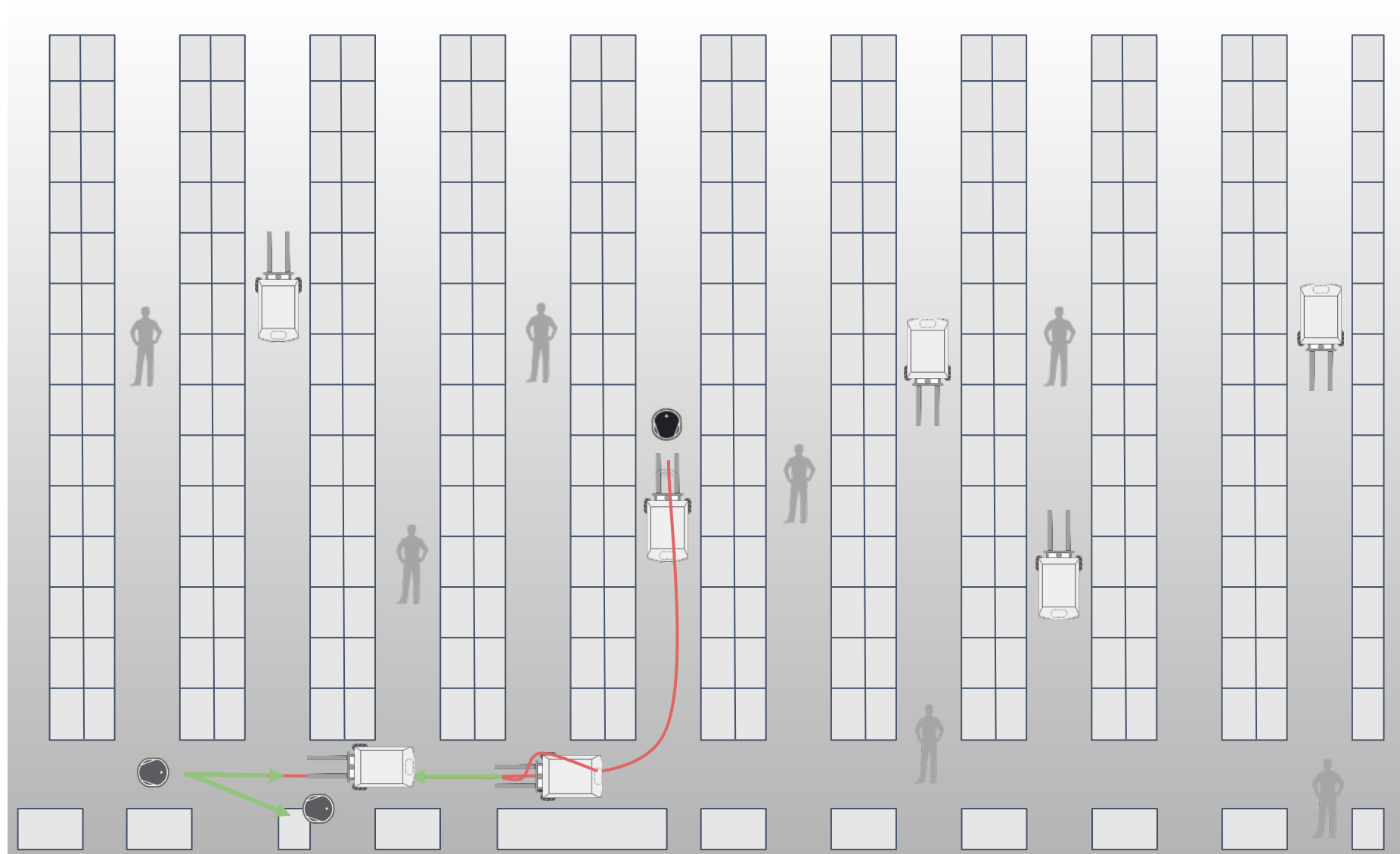
Machine Learning for More Human-like Behavior

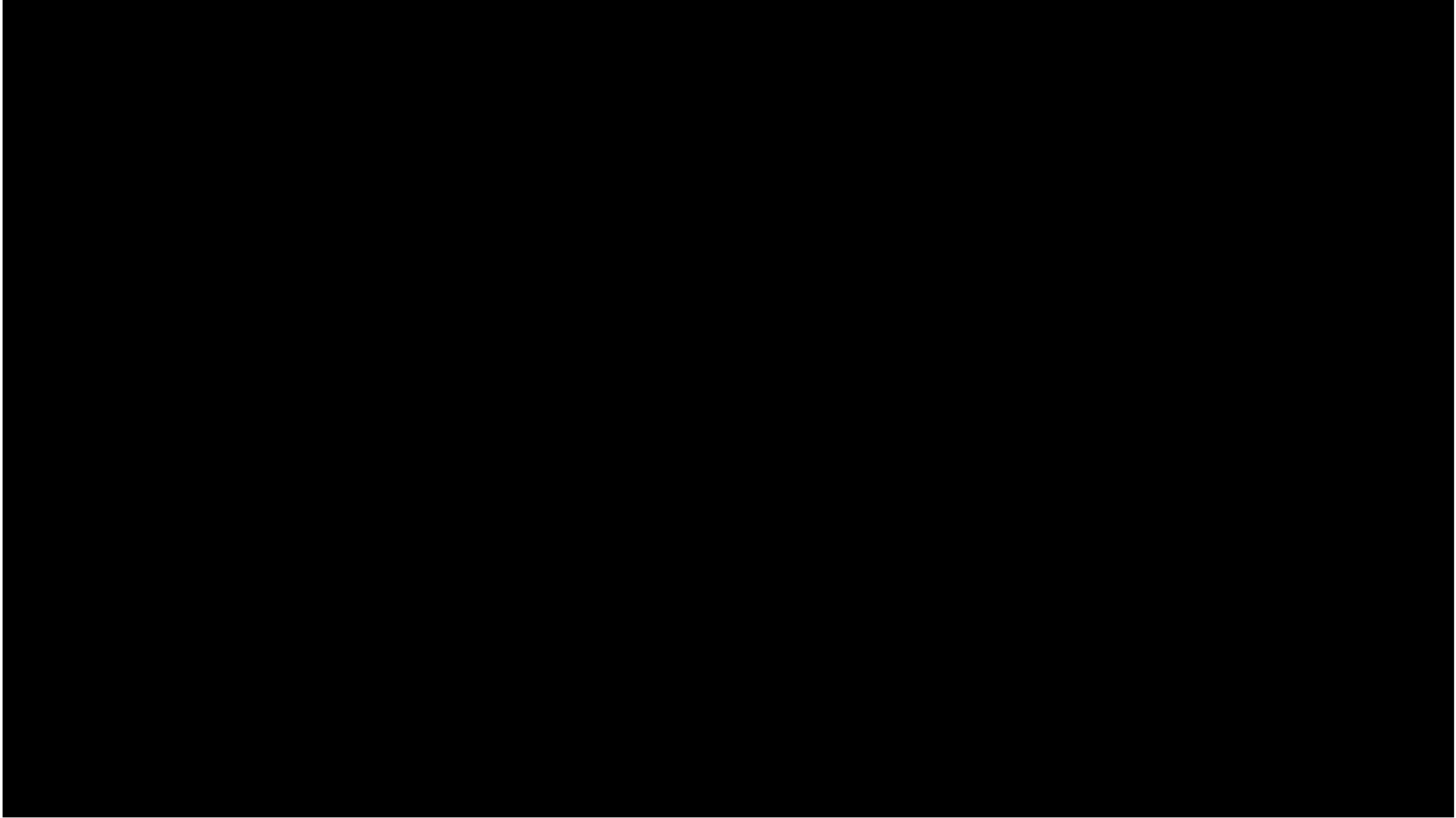


Machine Learning for More Human-like Behavior



Machine Learning for More Human-like Behavior







Thank You!



Russell Toris

Director of Robotics

Fetch Robotics

rtoris@fetchrobotics.com