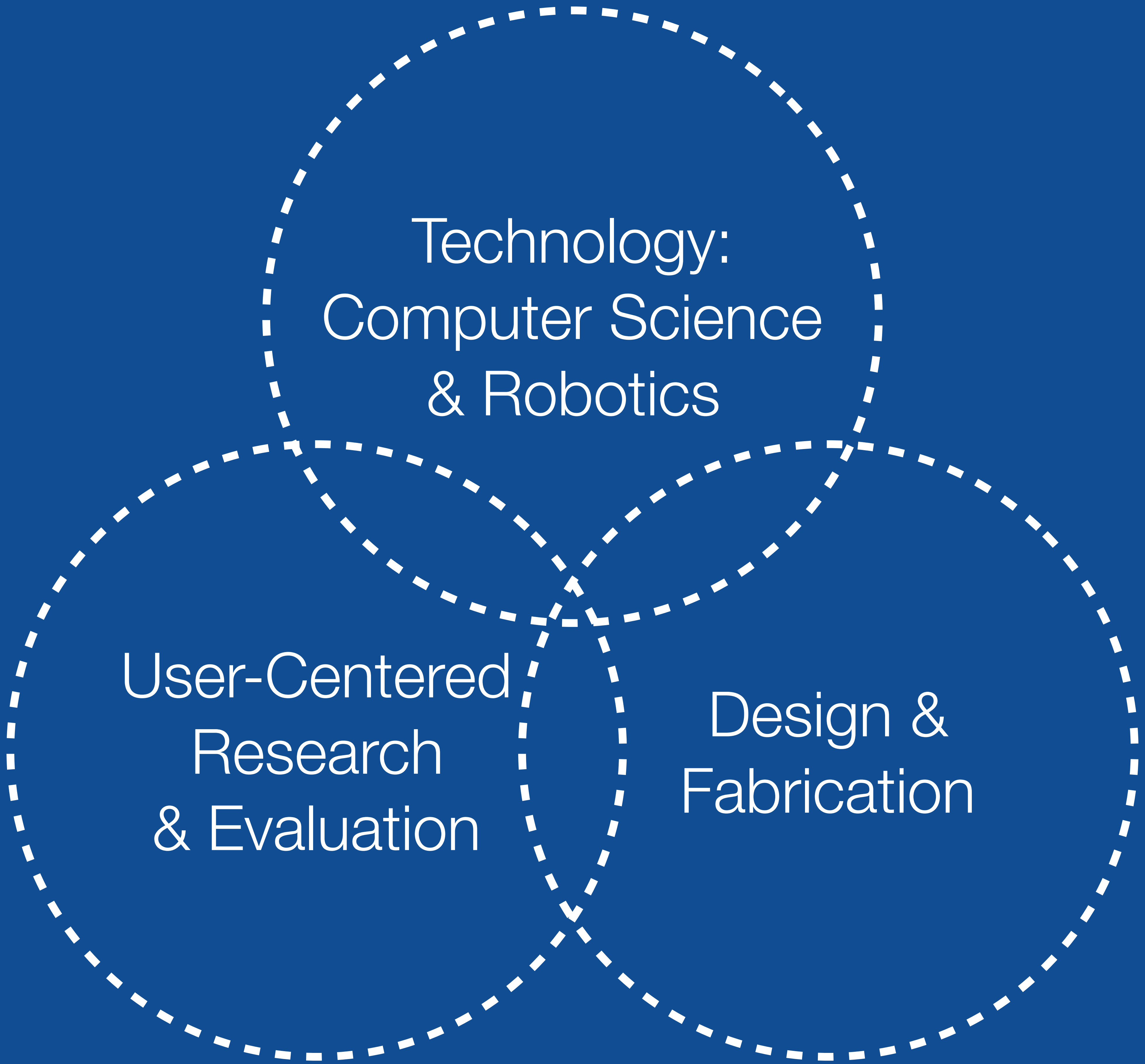
(Deep) Function Approximation in Mobile Robotics

Marynel Vázquez
Assistant Professor, Yale Computer Science
http://www.marynel.net
marynel.vazquez@yale.edu

Yale
INTERACTIVE MACHINES GROUP

We can use **interactivity** to increase the **value** of computing technologies



Leveraging Social Contexts for Multi-Party HRI


FROG (sevilla.abc.es)


MoRo, Ewaybot


STRANDS (europa.eu)

Leveraging Social Contexts for Multi-Party HRI

- Understanding social group phenomena in HRI
- Social Perception
- Decision Making

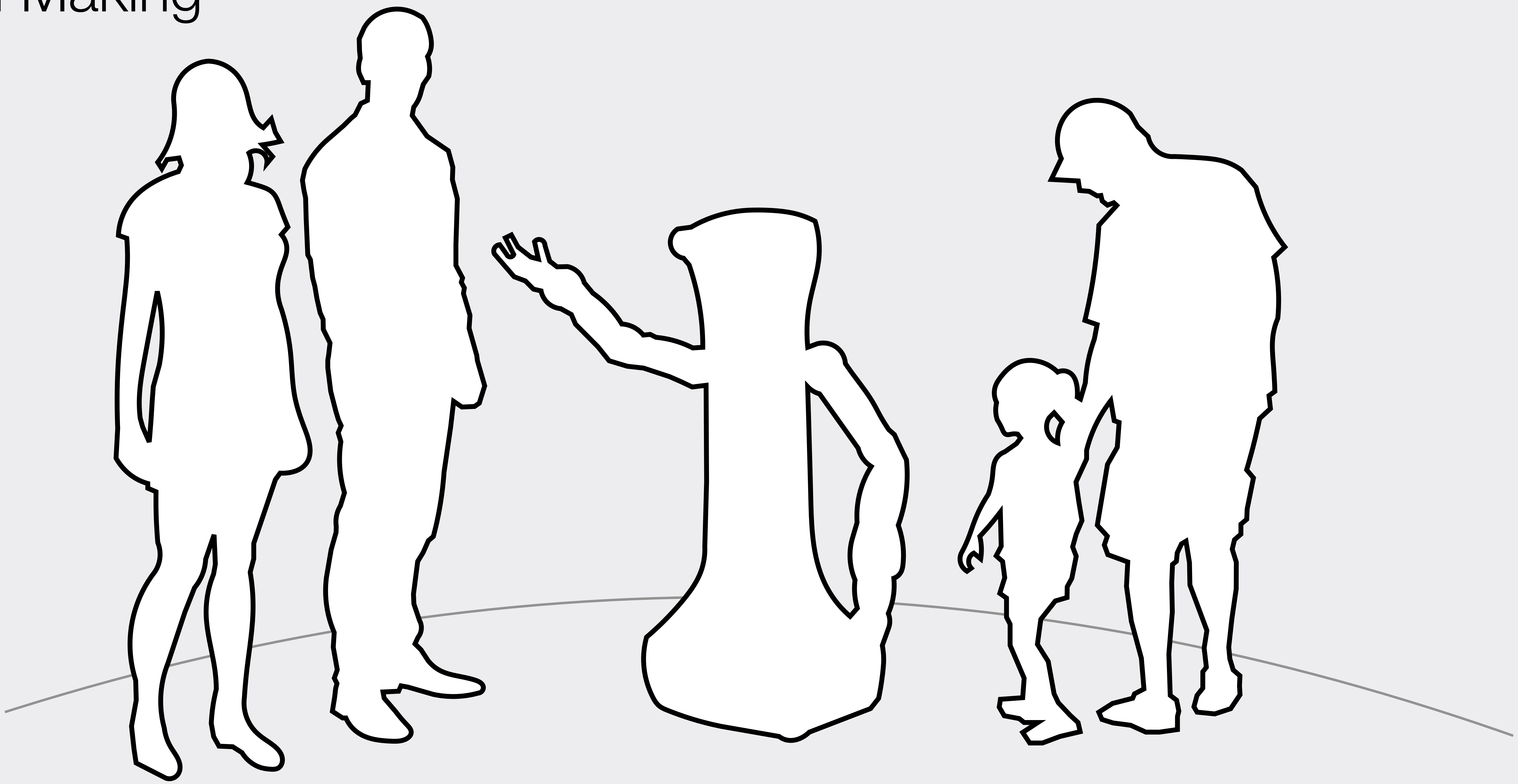

FROG (sevilla.abc.es)


MoRo, Ewaybot


STRANDS (europa.eu)

Leveraging Social Contexts for Multi-Party HRI

- Understanding social group phenomena in HRI
- Social Perception
- Decision Making



On-Going Research Directions

Social Navigation*

* In collaboration with Stanford

Spatial Human Behavior

Social Perception

Robot Communication

Social Influence

Yale
INTERACTIVE MACHINES GROUP

Function Approximation for Mobile Robotics

Social
Navigation*

* In collaboration with Stanford

- Background
- Fundamental technologies for social navigation
  - Traversability Estimation
  - A-B Navigation in Dynamic Environments
- What might come next?

Function Approximation for Mobile Robotics

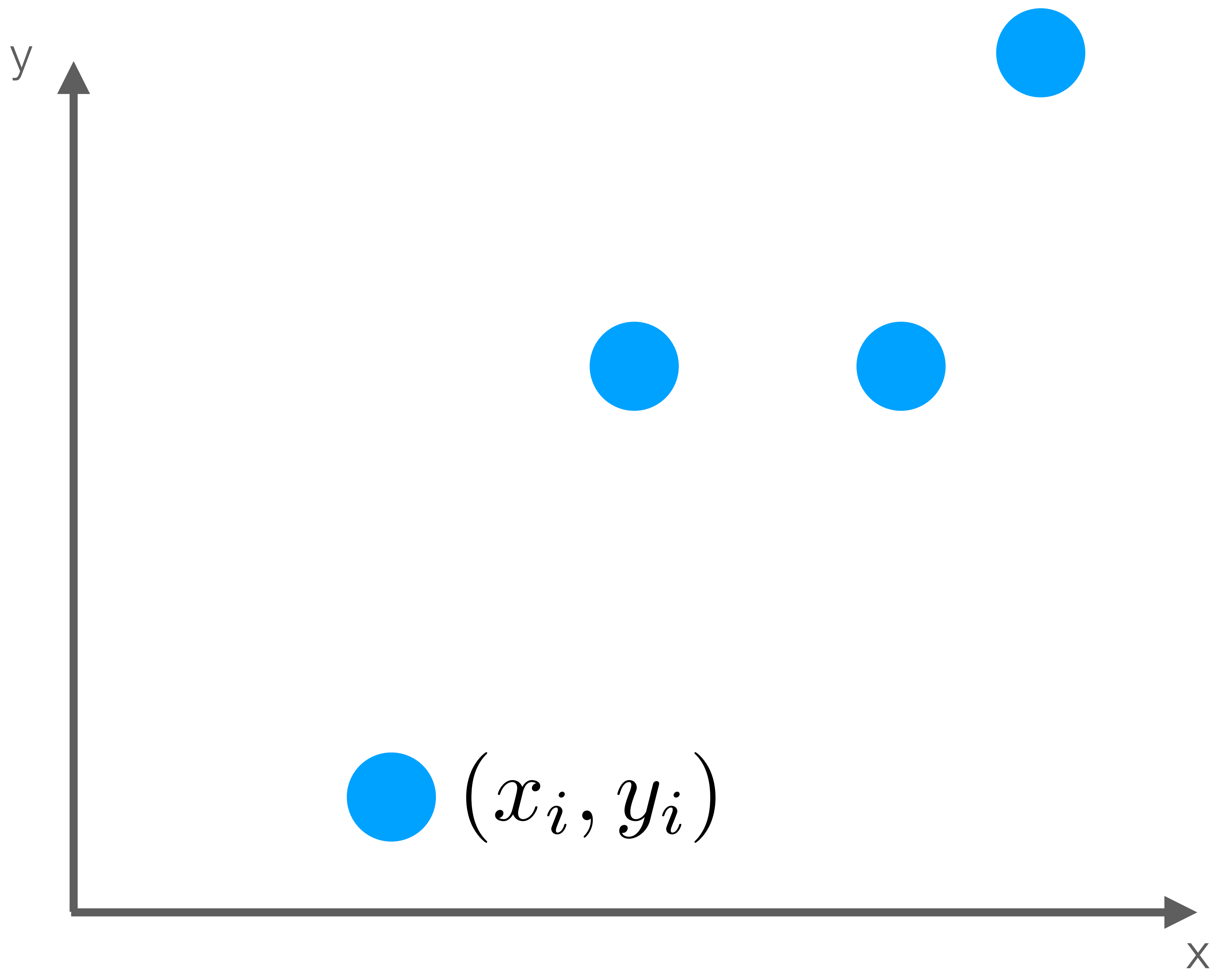
Social Navigation*

* In collaboration with Stanford

- Background
- Fundamental technologies for social navigation
  - Traversability Estimation
  - A-B Navigation in Dynamic Environments
- What might come next?

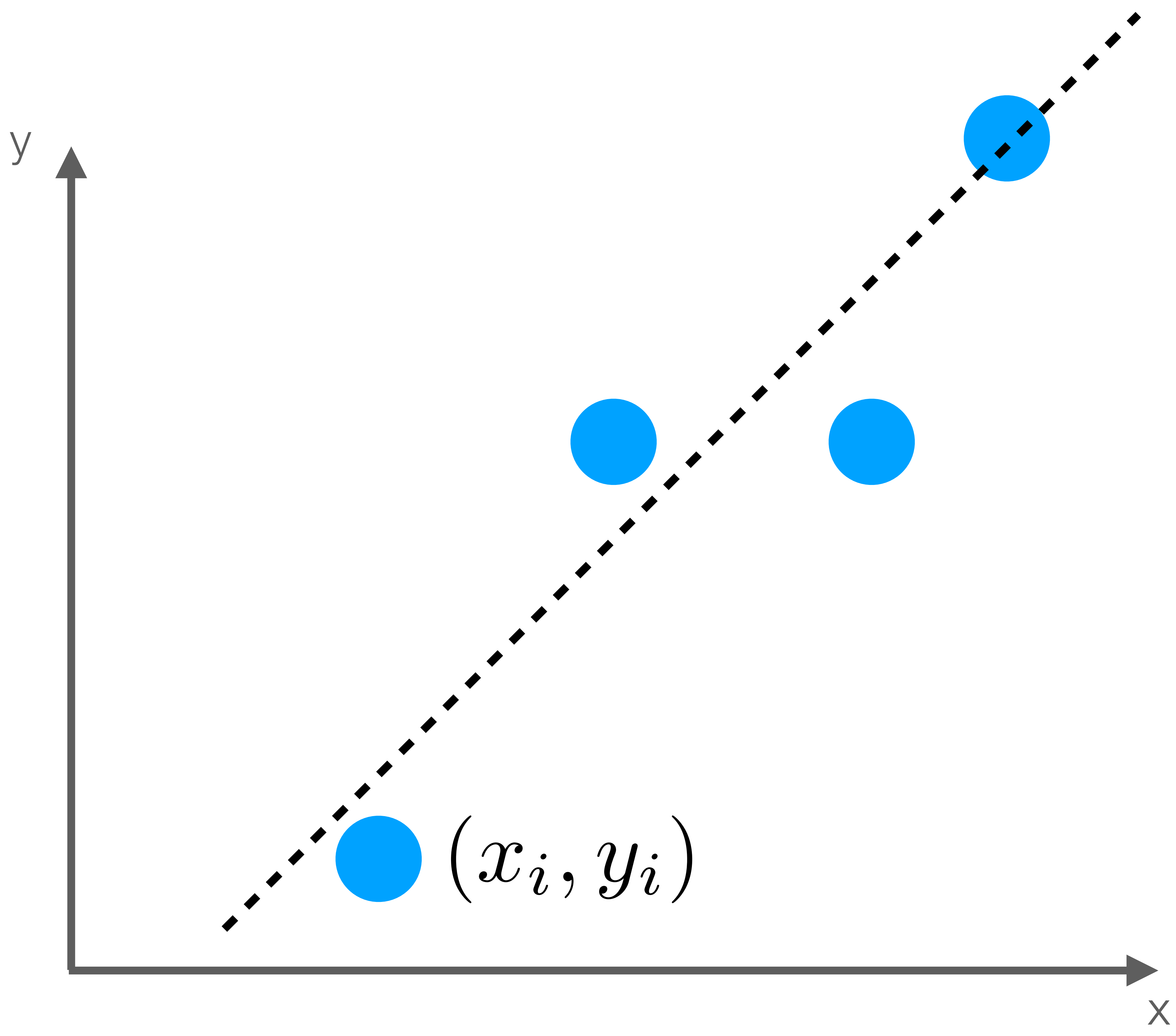
Function Approximation?

A Simple 2D Case



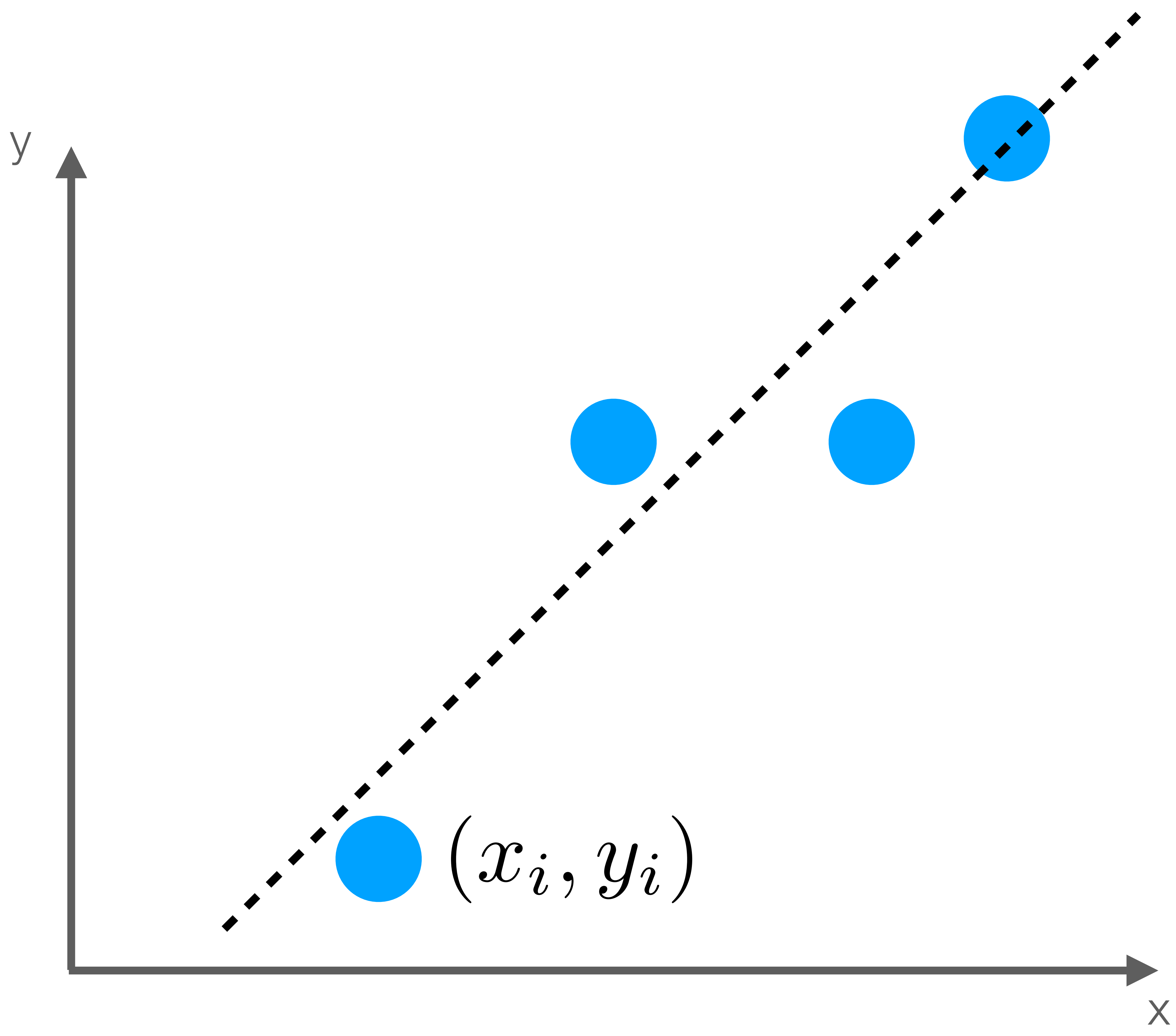
- Some phenomenon F gives you (x, y) samples
- Think of F as a function: $f(x) = y$
- Can we estimate what f is?

A Simple 2D Case



- Some phenomenon F gives you (x, y) samples
- Think of F as a function: $f(x) = y$
- Can we estimate what f is?
  - Assume its a line: $f(x) = mx + b$

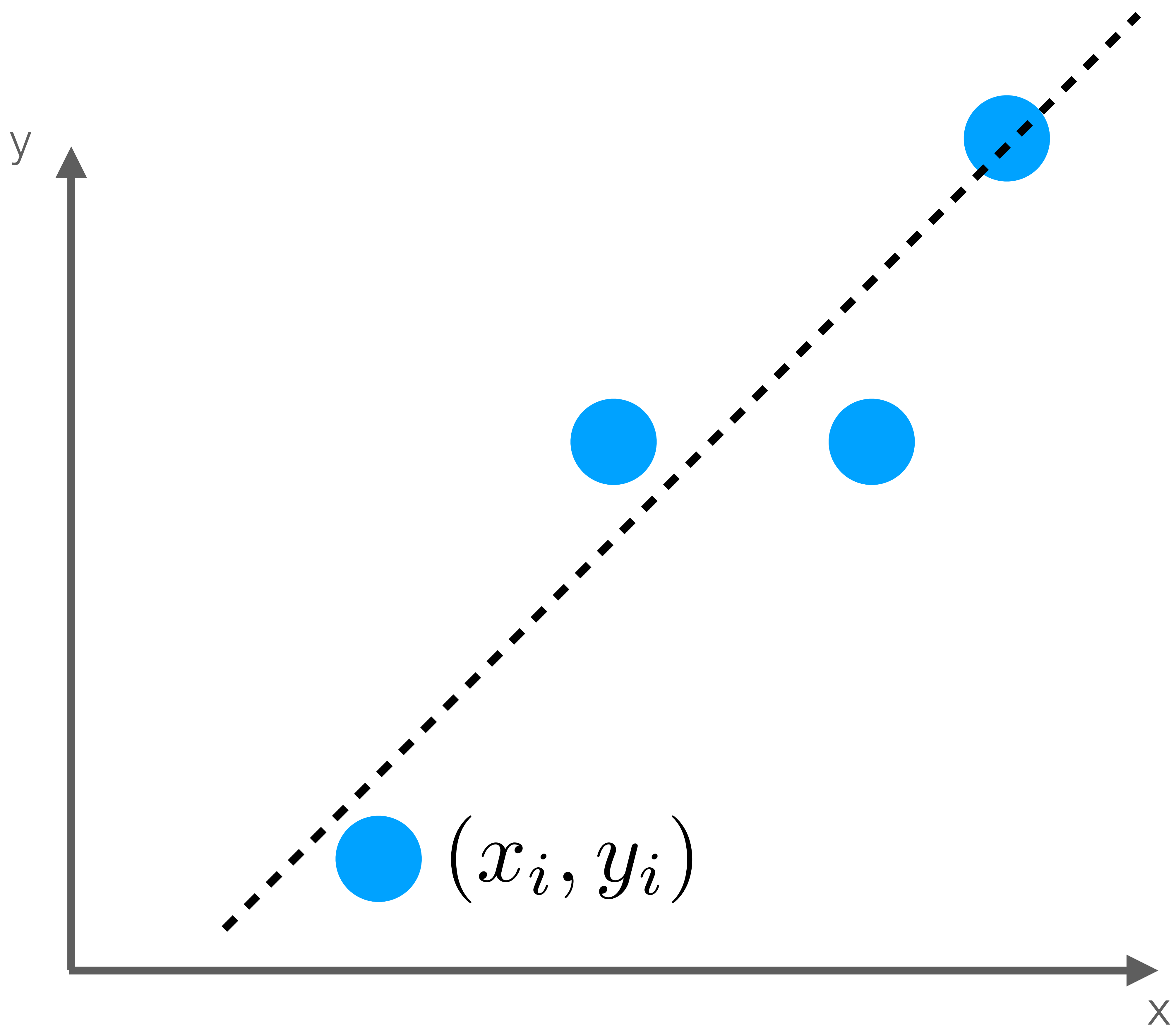
A Simple 2D Case



- Some phenomenon F gives you (x, y) samples
- Think of F as a function: $f(x) = y$
- Can we estimate what f is?
  - Assume its a line: $f(x) = mx + b$

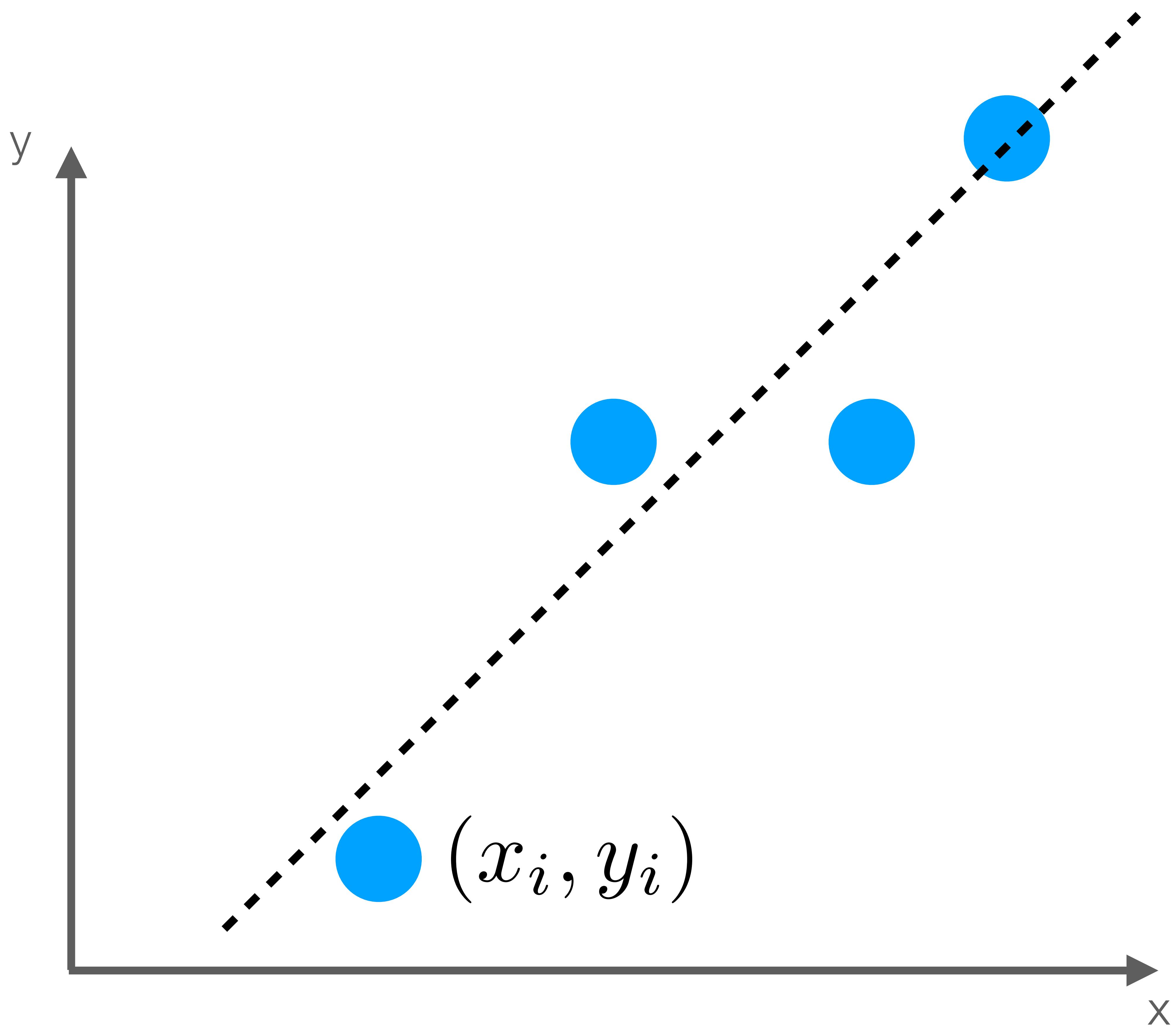
Goal: Estimate *m* and *b*, e.g., by minimizing least squares error to samples: $\sum_i (y_i - f(x_i))^2$

A Simple 2D Case



Goal: $\sum_i (y_i - f_\theta(x_i))^2$ where $\theta = m, b$

A Simple 2D Case

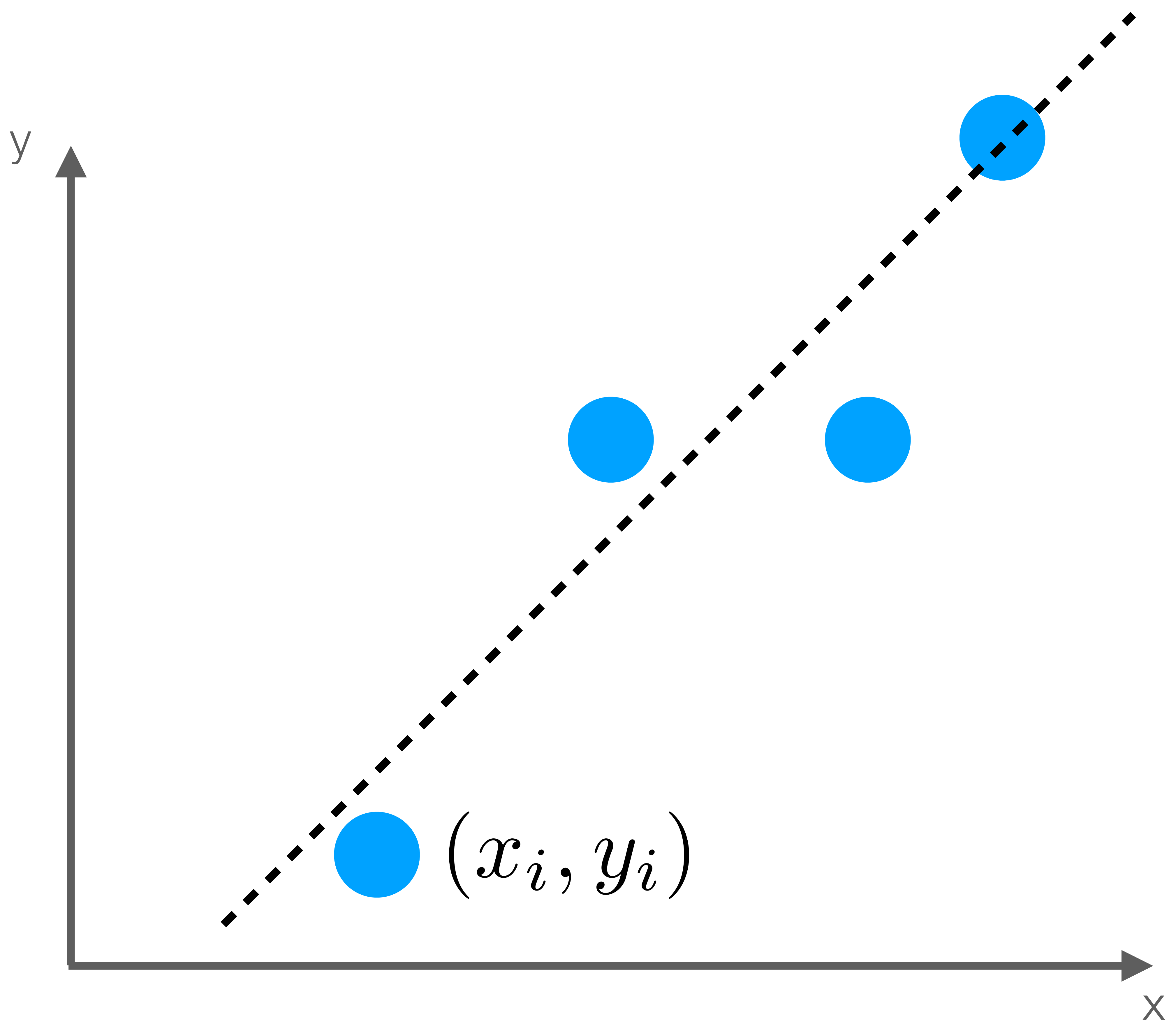


Goal: $\sum_i (y_i - f_\theta(x_i))^2$ where $\theta = m, b$

Closed-Form Solution:

$$\hat{\theta} = (X^T X)^{-1} X^T \mathbf{y}$$

A Simple 2D Case



Goal: $\sum_i (y_i - f_\theta(x_i))^2$ where $\theta = m, b$

Closed-Form Solution:

$$\hat{\theta} = (X^T X)^{-1} X^T \mathbf{y}$$

Iterative Search: Iteratively search for better parameters.

More Generic Case

$$y = \theta_0 + \theta_1 x_1 + ... + \theta_d x_d = \sum_{j=0}^{d} \theta_j x_j \; = h_\theta(\mathbf{x})$$

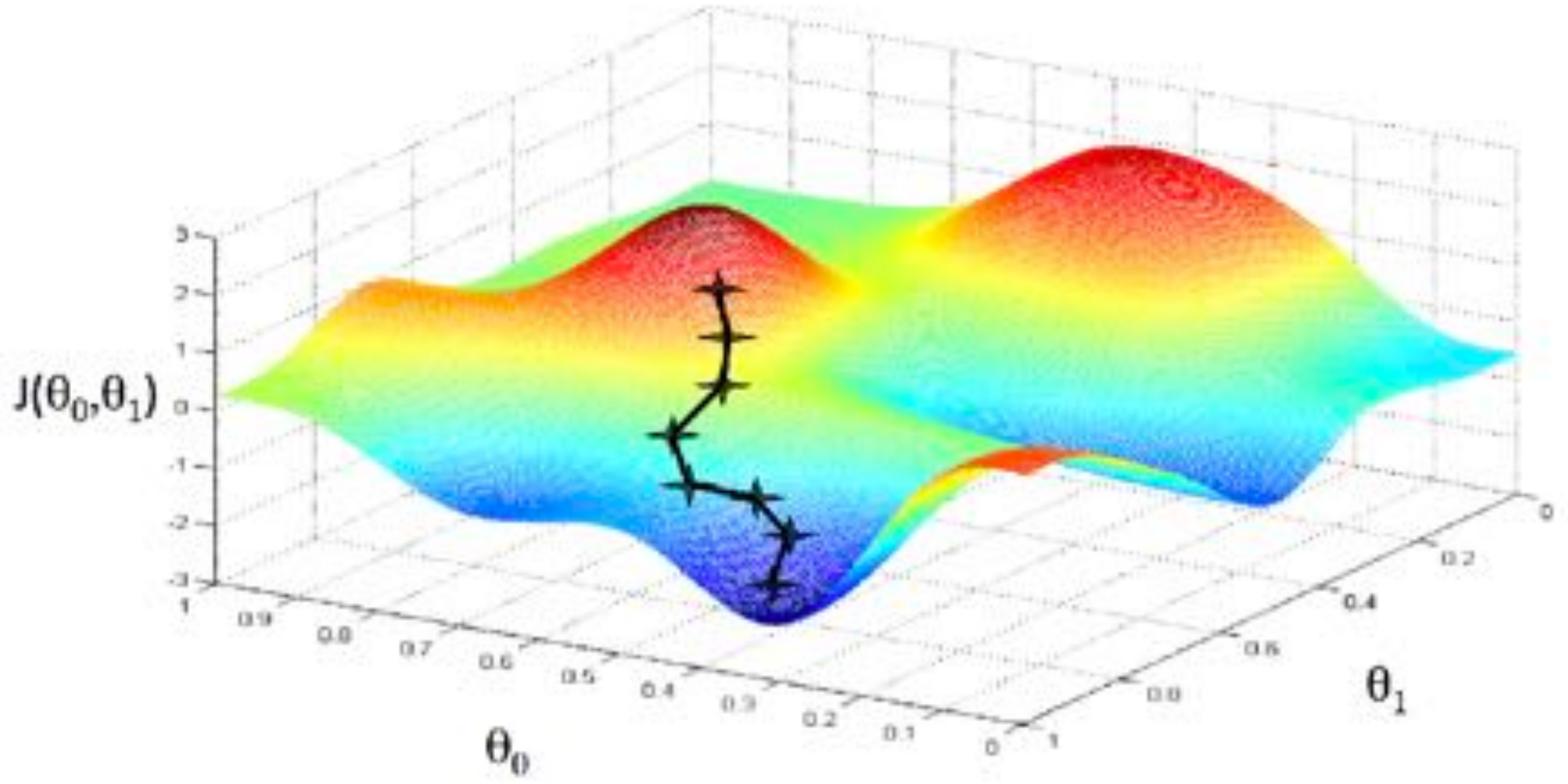
assume $x_0 = 1$

- Fit model by minimizing sum of squared errors.

$$J(\theta) = \frac{1}{2n} \sum_{i=1}^{n} \left(h_\theta(\mathbf{x}^{(i)}) - y^{(i)} \right)^2$$

Basic Search Procedure

- Choose initial values for θ
- Until we reach a minimum:
  - Choose a new parameters θ to reduce $J(\theta)$



Gradient Descent

- Choose initial values for θ
- Until we reach a minimum:
  - Choose a new parameters θ to reduce $J(\theta)$

$$\theta_j \leftarrow \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$

(simultaneously update all parameters j=0, …, d)

Gradient Descent

- Choose initial values for θ
- Until we reach a minimum:
  - Choose a new parameters θ to reduce $J(\theta)$

$$\theta_j \leftarrow \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$

Learning Rate
(small value)

(simultaneously update all parameters j=0, …, d)

Illustrative Example: GD for Linear Regression

$$\frac{\partial}{\partial \theta_j} J(\theta) = \frac{\partial}{\partial \theta_j} \frac{1}{2n} \sum_{i=1}^{n} \left(h_\theta(\mathbf{x}^{(i)}) - y^{(i)} \right)^2$$

Illustrative Example: GD for Linear Regression

$$\frac{\partial}{\partial \theta_j} J(\theta) = \frac{\partial}{\partial \theta_j} \frac{1}{2n} \sum_{i=1}^{n} \left(h_\theta(\mathbf{x}^{(i)}) - y^{(i)} \right)^2$$

$$= \frac{\partial}{\partial \theta_j} \frac{1}{2n} \sum_{I=1}^{n} \left(\sum_{k=0}^{d} \theta_k x_k^{(i)} - y^{(i)} \right)^2$$

Illustrative Example: GD for Linear Regression

$$
\begin{aligned}
\frac{\partial}{\partial \theta_j} J(\theta) &= \frac{\partial}{\partial \theta_j} \frac{1}{2n} \sum_{i=1}^{n} \left(h_\theta(\mathbf{x}^{(i)}) - y^{(i)} \right)^2 \\
&= \frac{\partial}{\partial \theta_j} \frac{1}{2n} \sum_{I=1}^{n} \left(\sum_{k=0}^{d} \theta_k x_k^{(i)} - y^{(i)} \right)^2 \\
&= \left[\frac{1}{n} \sum_{I=1}^{n} \left(\sum_{k=0}^{d} \theta_k x_k^{(i)} - y^{(i)} \right) \right] \frac{\partial}{\partial \theta_j} \left(\sum_{k=0}^{d} \theta_k x_k^{(i)} - y^{(i)} \right)
\end{aligned}
$$

Illustrative Example: GD for Linear Regression

$$
\begin{aligned}
\frac{\partial}{\partial \theta_j} J(\theta) &= \frac{\partial}{\partial \theta_j} \frac{1}{2n} \sum_{i=1}^{n} \Big(h_\theta(\mathbf{x}^{(i)}) - y^{(i)} \Big)^2 \\
&= \frac{\partial}{\partial \theta_j} \frac{1}{2n} \sum_{I=1}^{n} \Big(\sum_{k=0}^{d} \theta_k x_k^{(i)} - y^{(i)} \Big)^2 \\
&= \Big[\frac{1}{n} \sum_{I=1}^{n} \Big(\sum_{k=0}^{d} \theta_k x_k^{(i)} - y^{(i)} \Big) \Big] \frac{\partial}{\partial \theta_j} \Big(\sum_{k=0}^{d} \theta_k x_k^{(i)} - y^{(i)} \Big) \\
&= \frac{1}{n} \sum_{I=1}^{n} \Big(\sum_{k=0}^{d} \theta_k x_k^{(i)} - y^{(i)} \Big) x_j^{(i)}
\end{aligned}
$$

Illustrative Example: GD for Linear Regression

- Choose initial values for θ
- Until we reach a minimum:
  - Compute gradient for all θ_j:

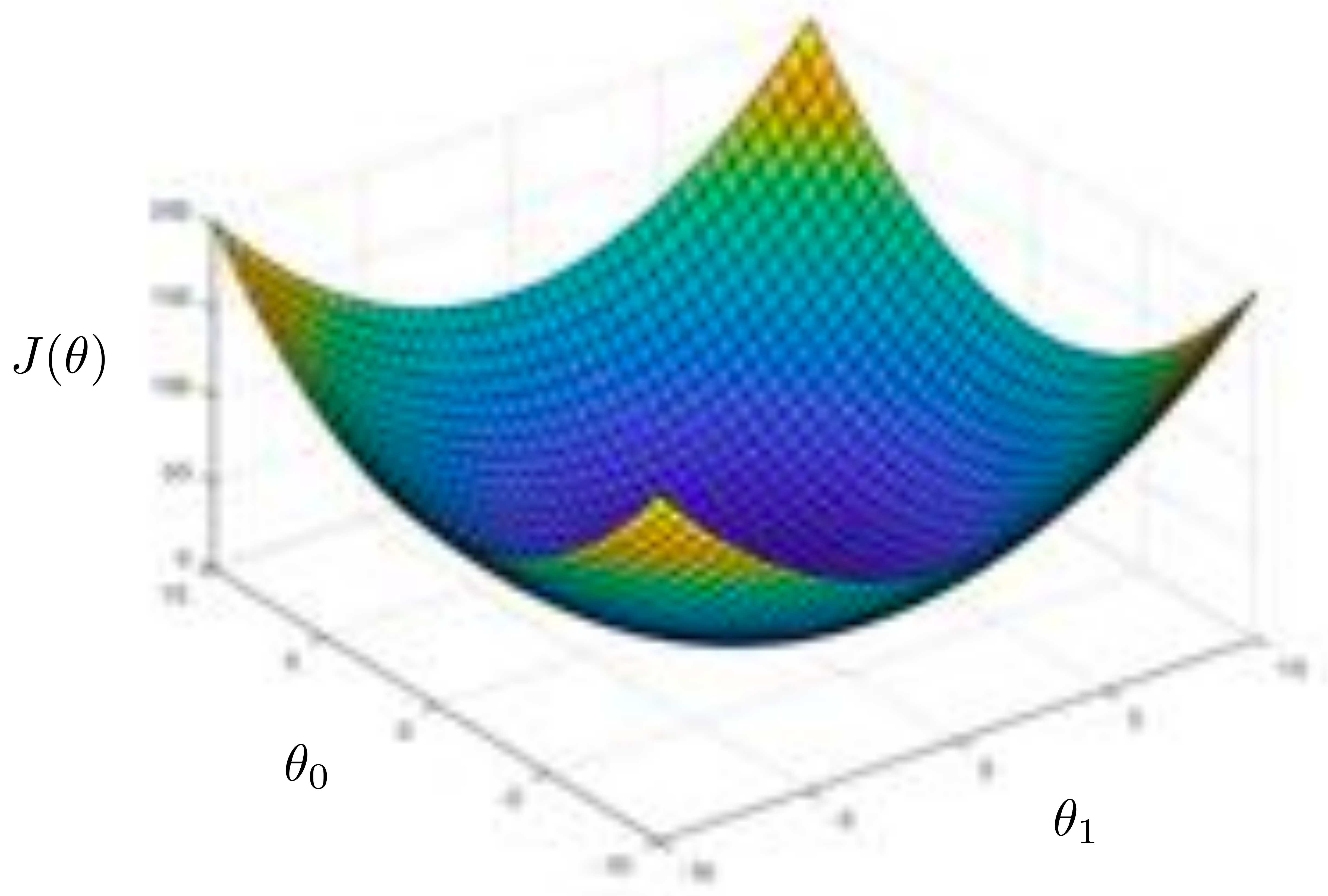
$$\frac{\partial}{\partial \theta_j} J(\theta) = \frac{1}{n} \sum_{I=1}^{n} \Big(\sum_{k=0}^{d} \theta_k x_k^{(i)} - y^{(i)} \Big) x_j^{(i)}$$

  - Choose new parameters for all θ_j: $\theta_j \leftarrow \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$

L2 error (1D input)

- Assume d=1:

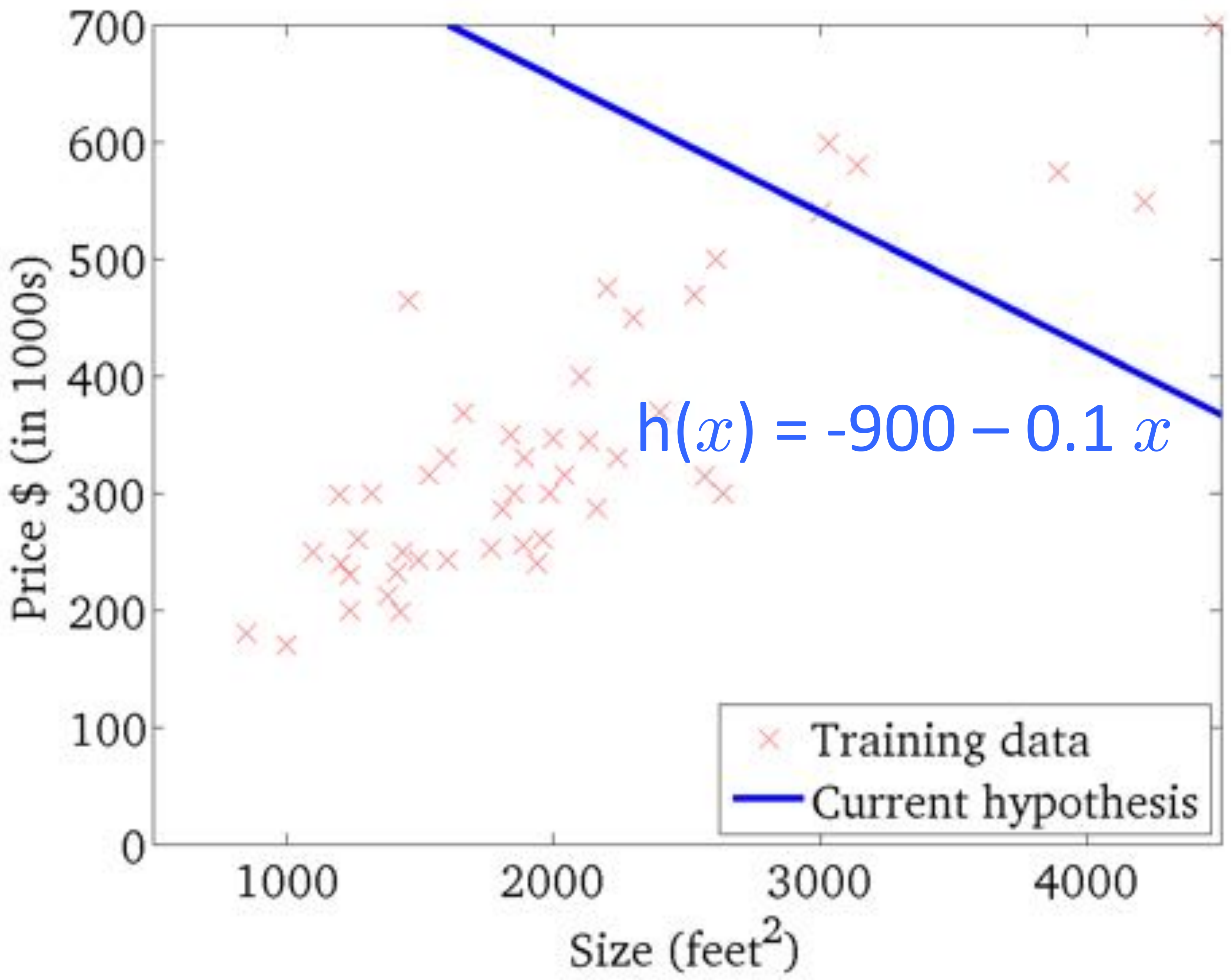
$$y = \theta_0 + \theta_1 x_1 = h_\theta(\mathbf{x})$$



Illustrative Example: GD for Linear Regression

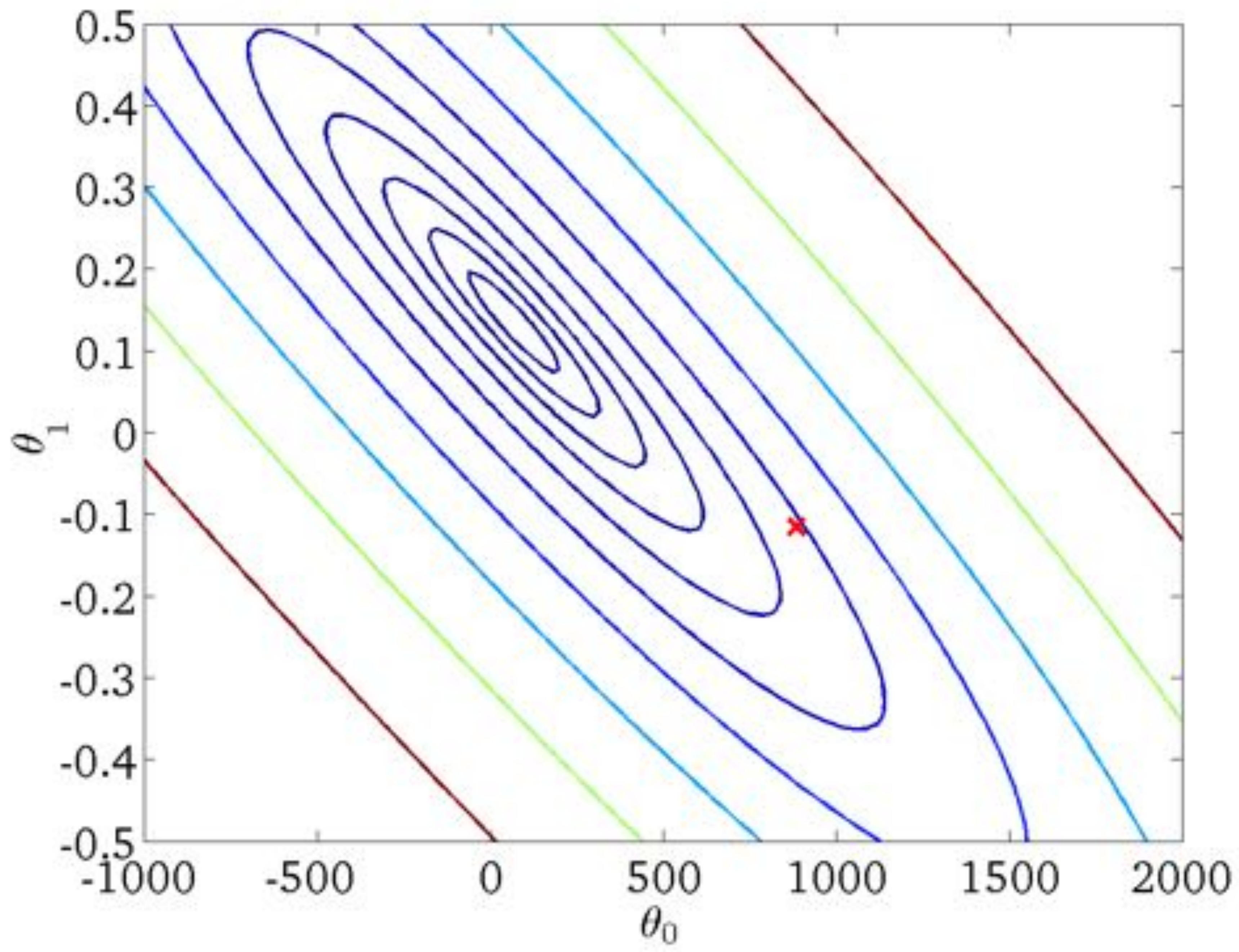
$$h_\theta(x)$$

(for fixed θ_0, θ_1, this is a function of x)



$$J(\theta_0, \theta_1)$$

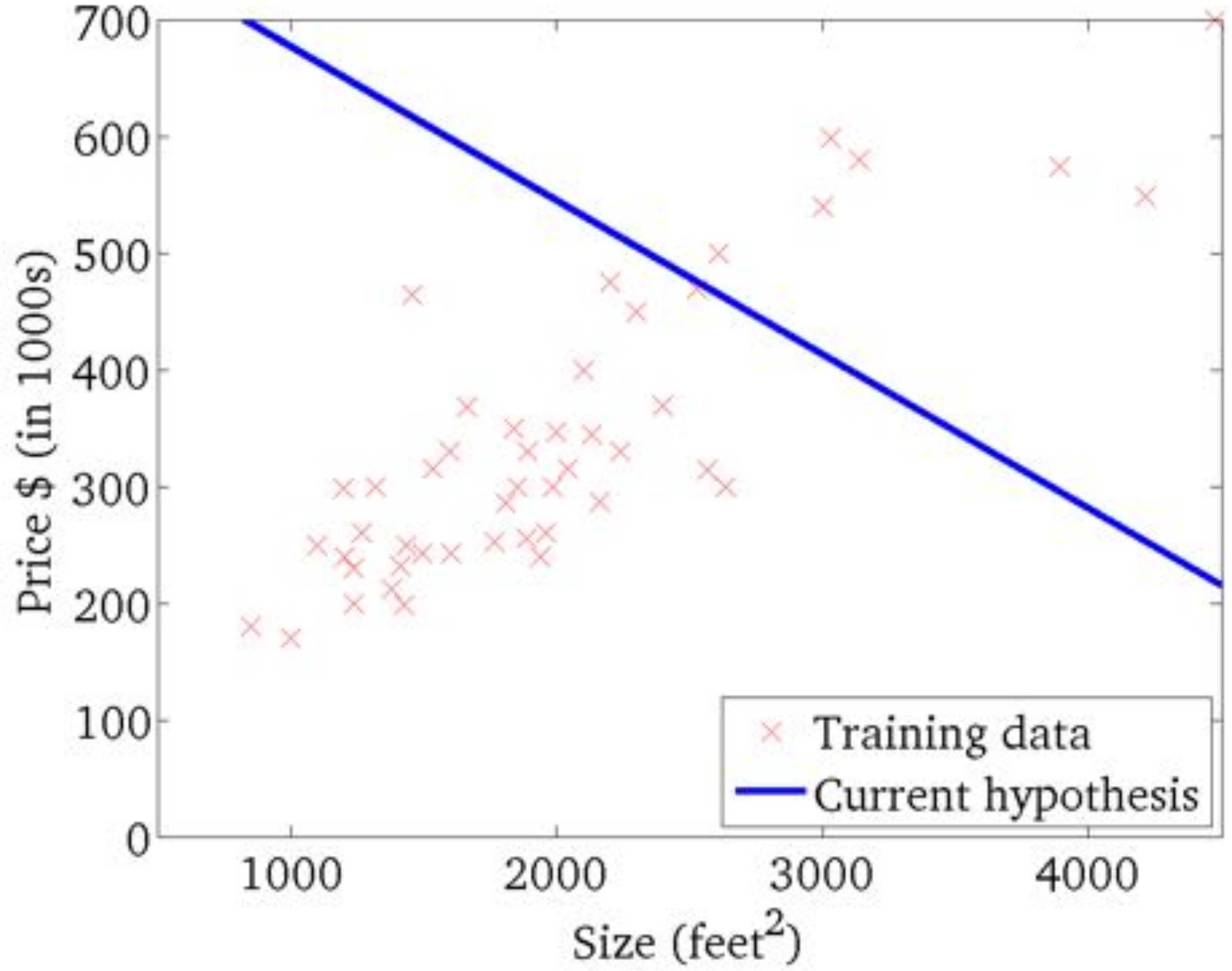
(function of the parameters θ_0, θ_1)



Illustrative Example: GD for Linear Regression

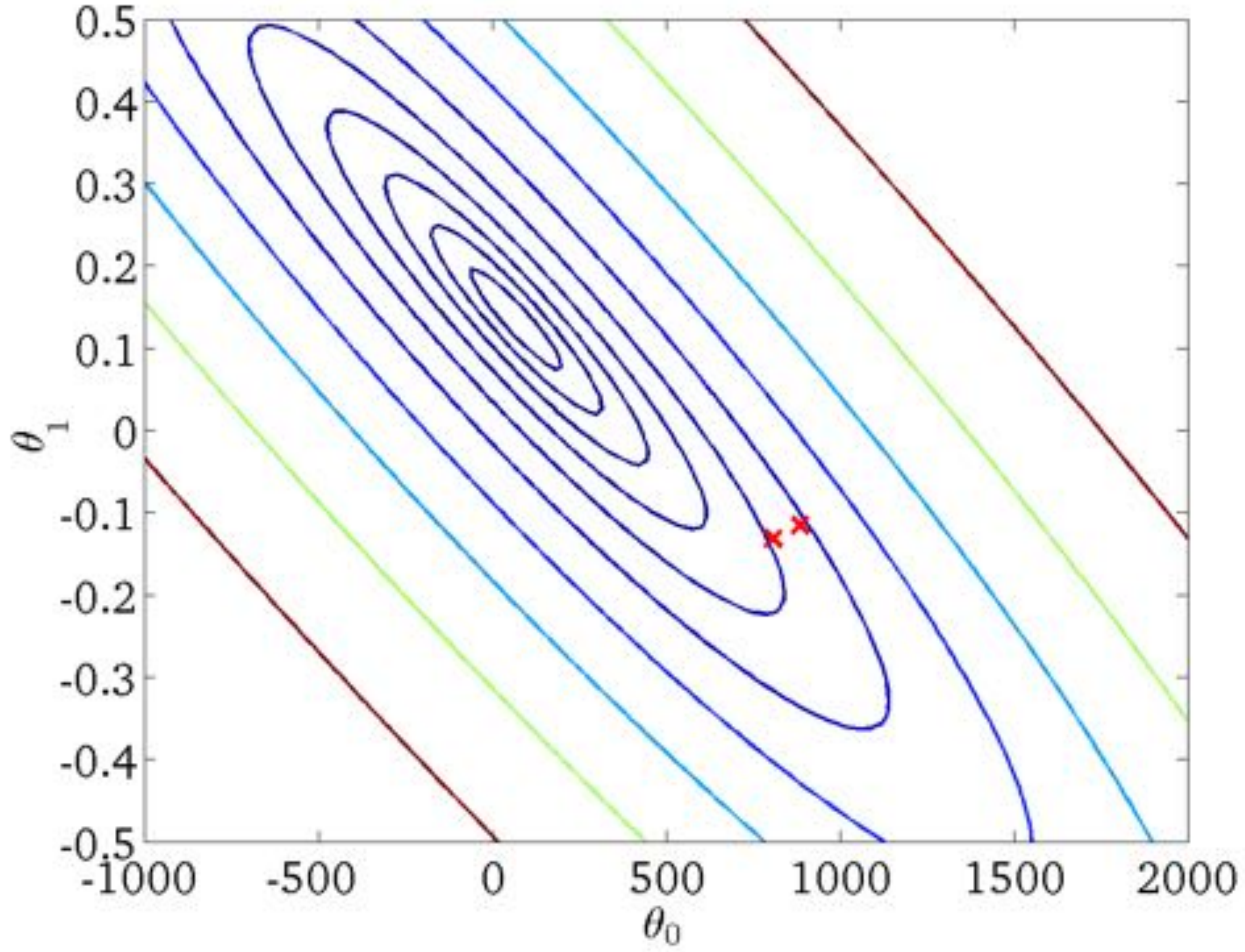
$h_\theta(x)$

(for fixed θ_0, θ_1, this is a function of x)



$J(\theta_0, \theta_1)$

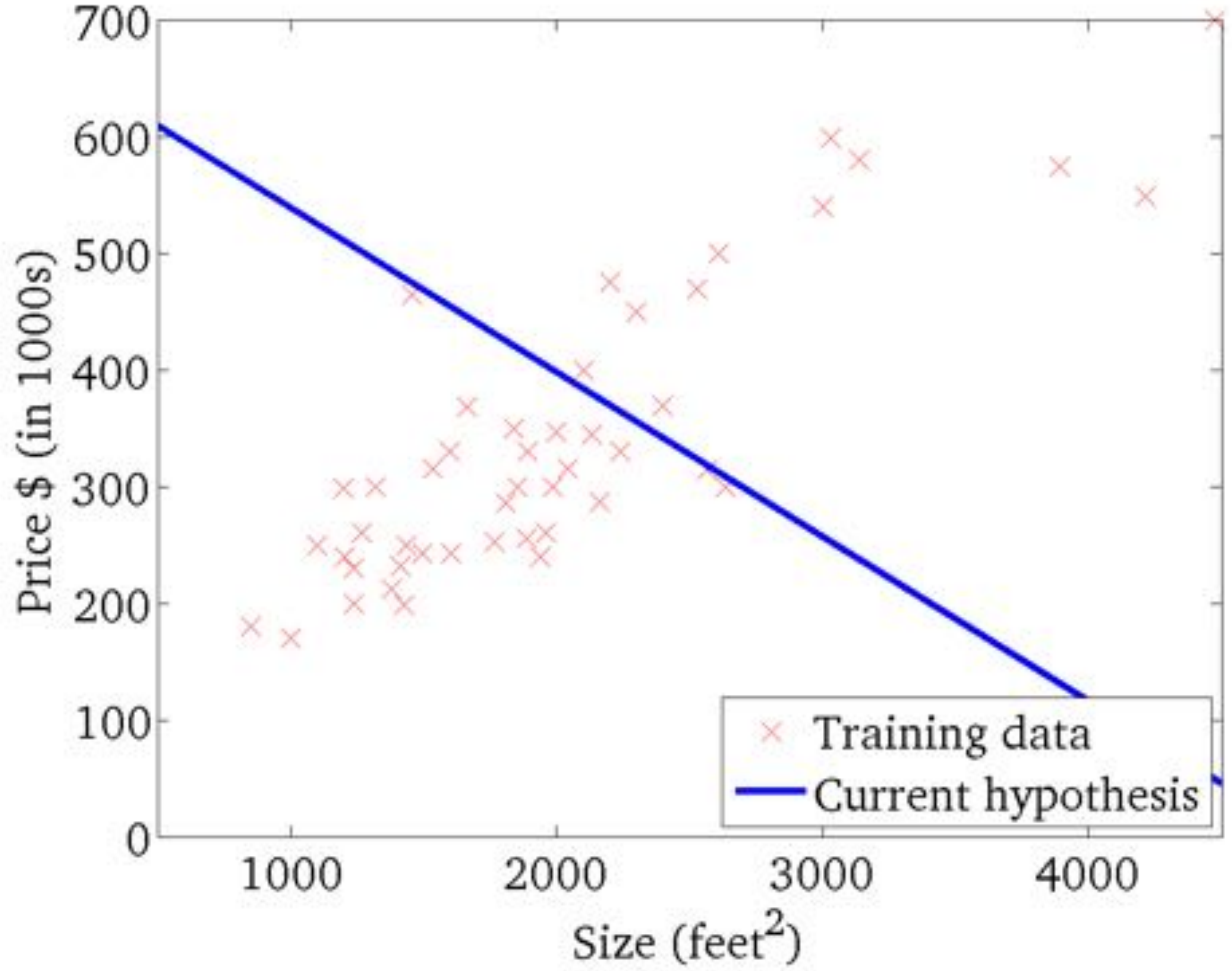
(function of the parameters θ_0, θ_1)



Illustrative Example: GD for Linear Regression

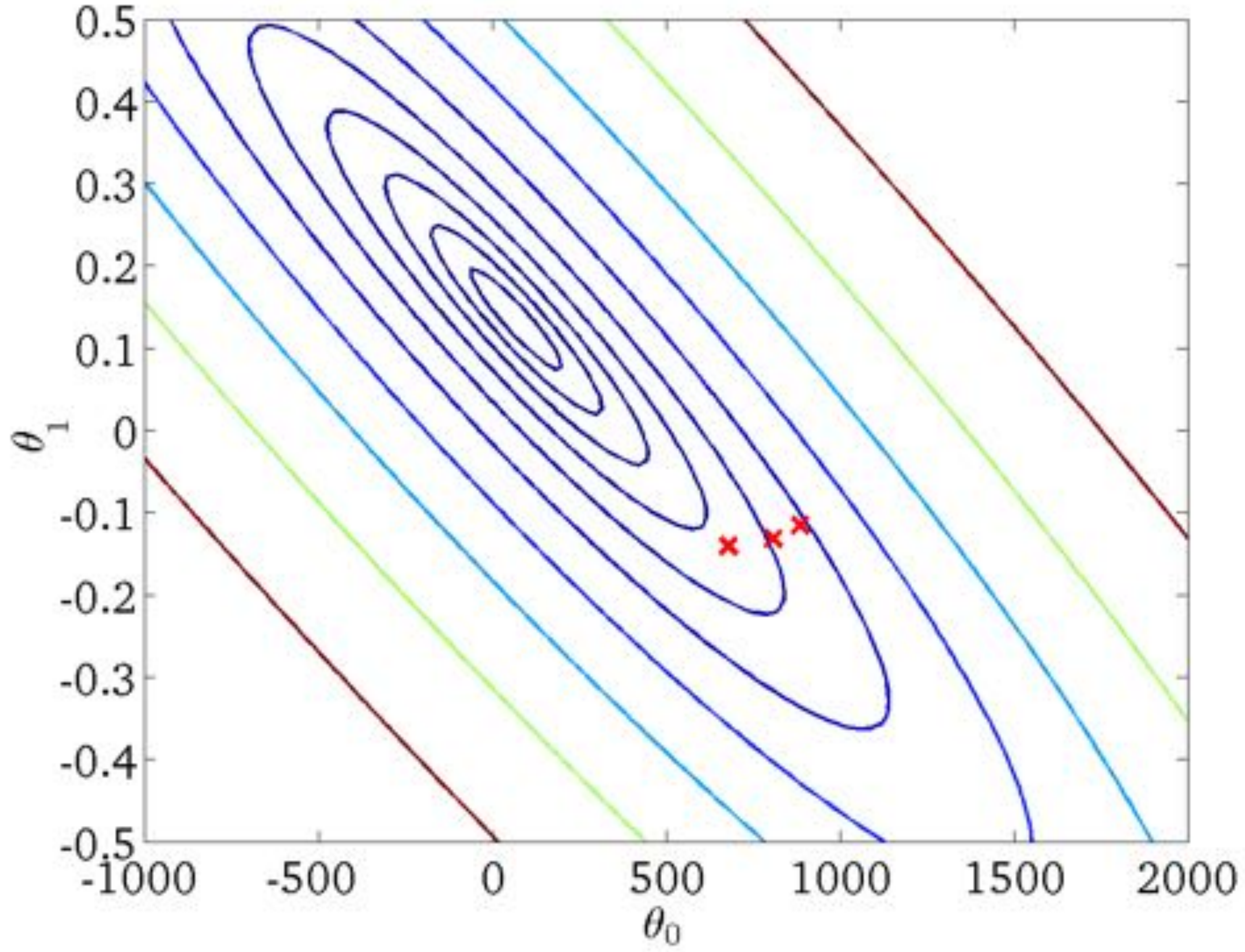
$h_\theta(x)$

(for fixed θ_0, θ_1, this is a function of x)



$J(\theta_0, \theta_1)$

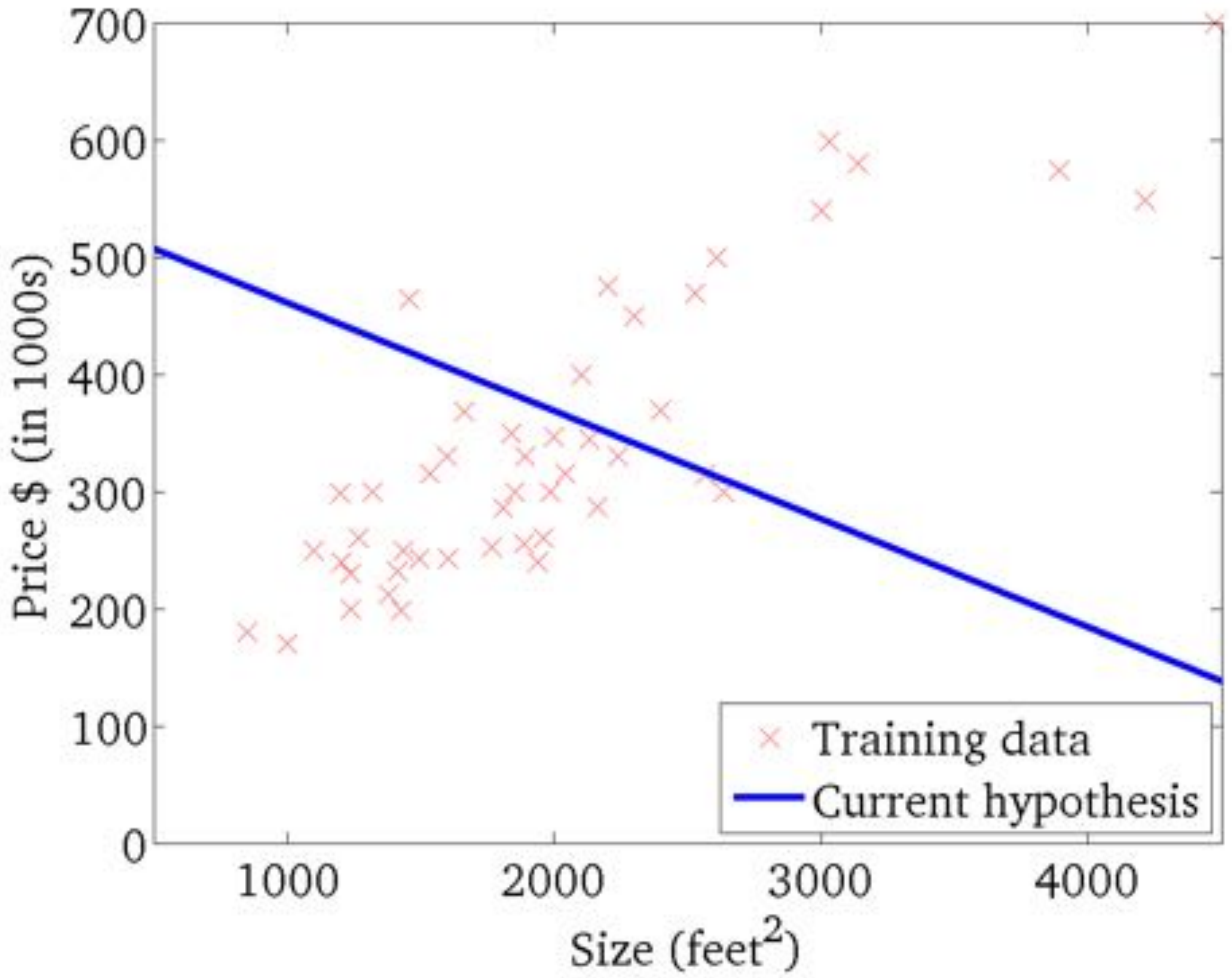
(function of the parameters θ_0, θ_1)



Illustrative Example: GD for Linear Regression

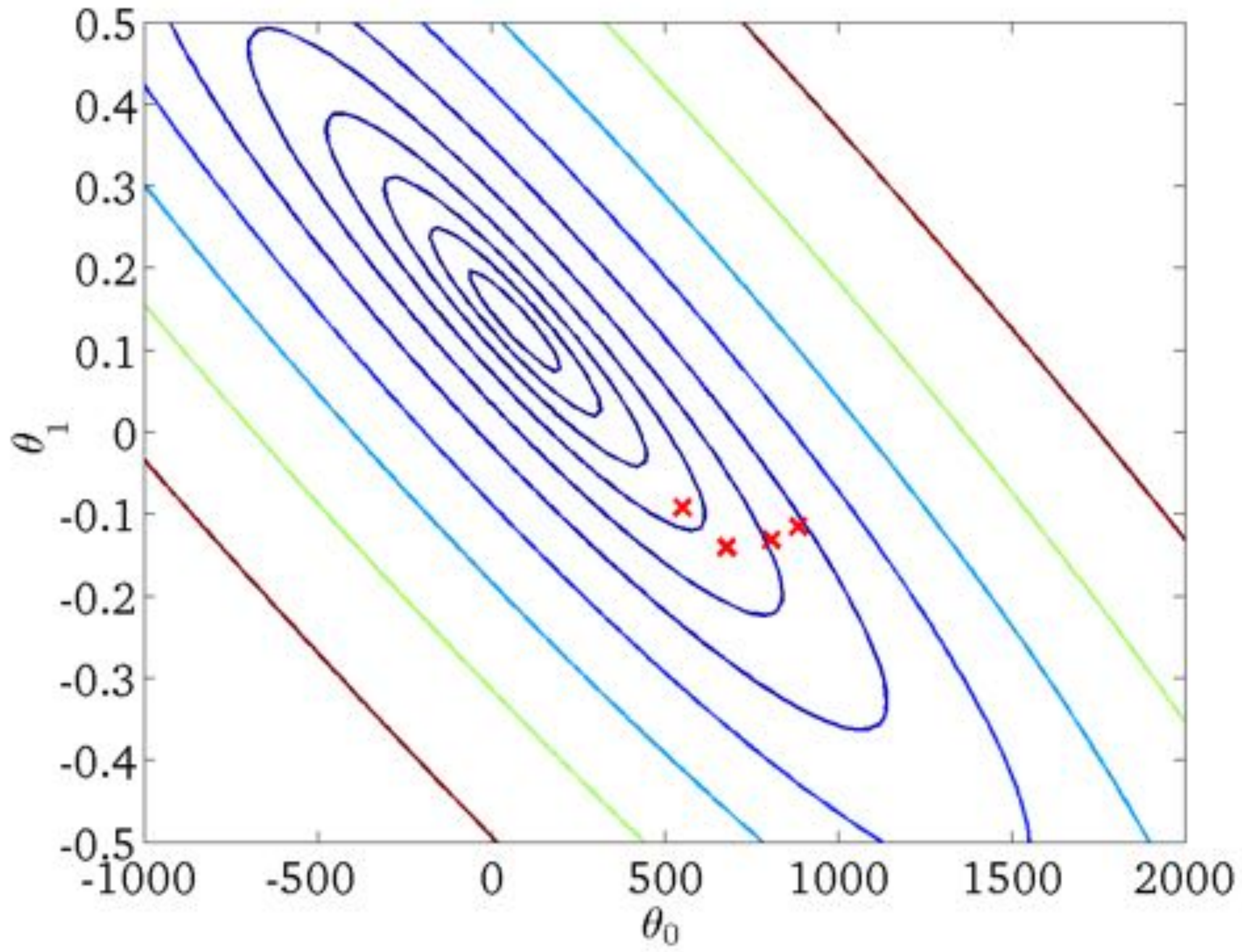
$h_\theta(x)$

(for fixed θ_0, θ_1, this is a function of x)



$J(\theta_0, \theta_1)$

(function of the parameters θ_0, θ_1)

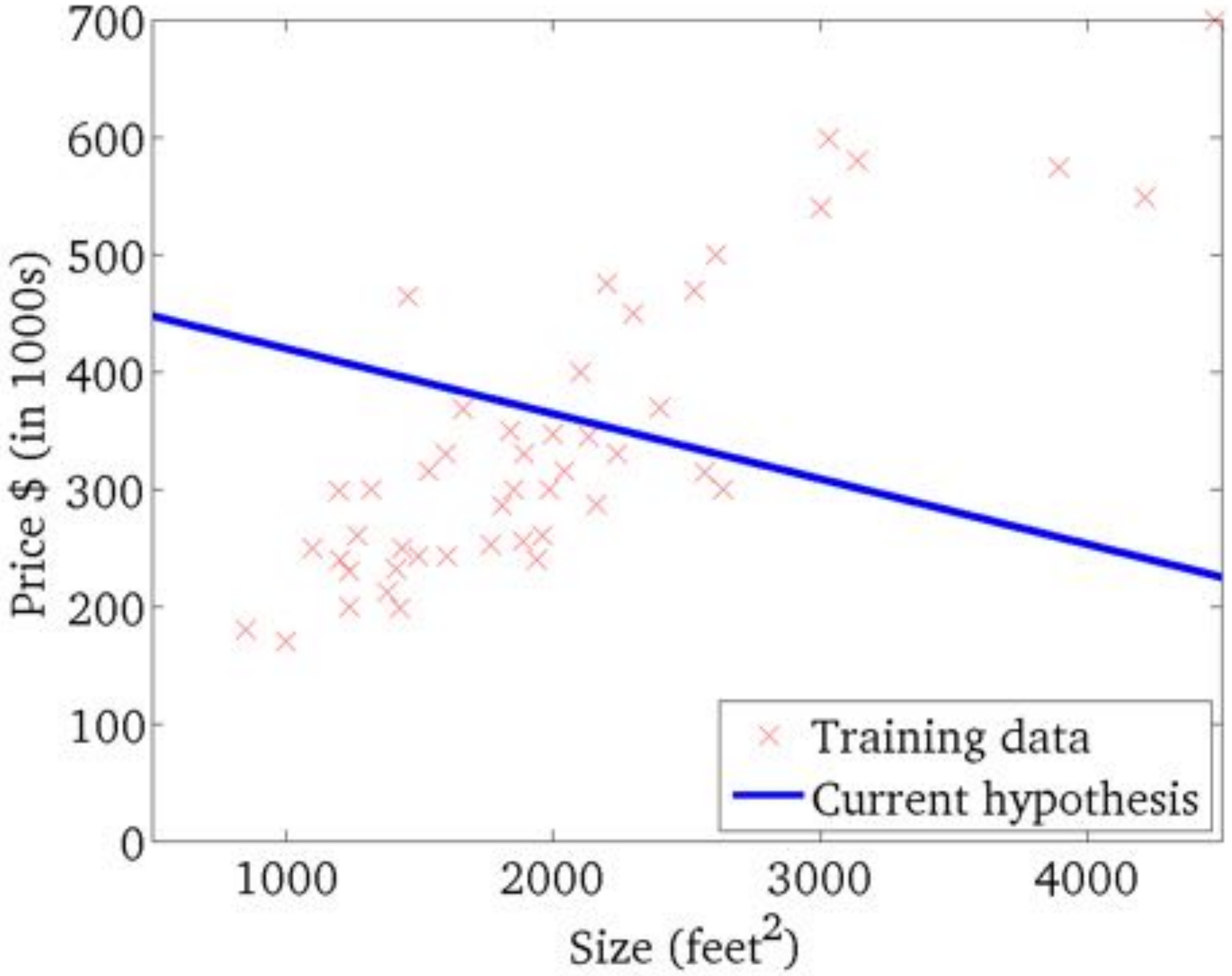


Slide by A. Ng

Illustrative Example: GD for Linear Regression

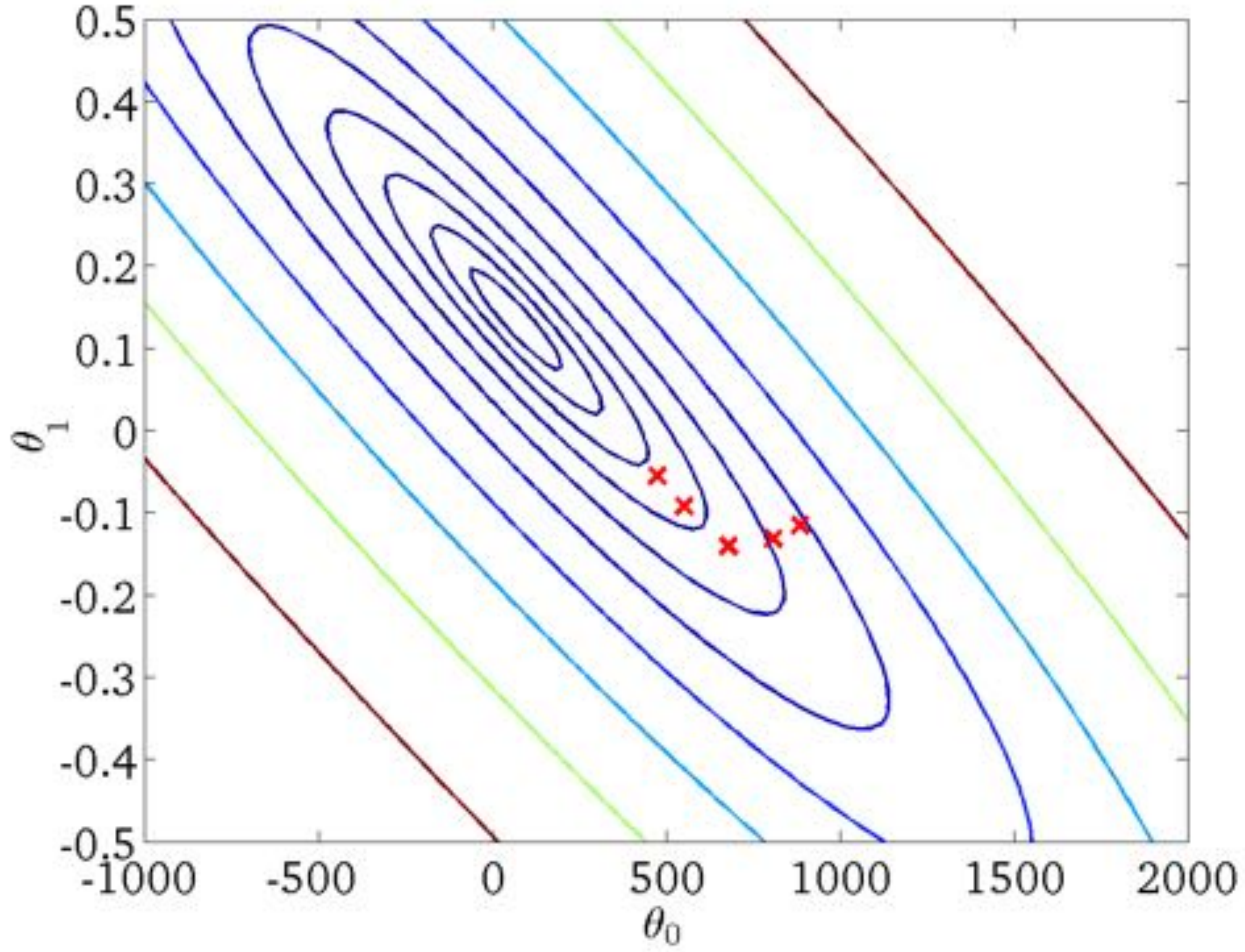
$h_\theta(x)$

(for fixed θ_0, θ_1, this is a function of x)



$J(\theta_0, \theta_1)$

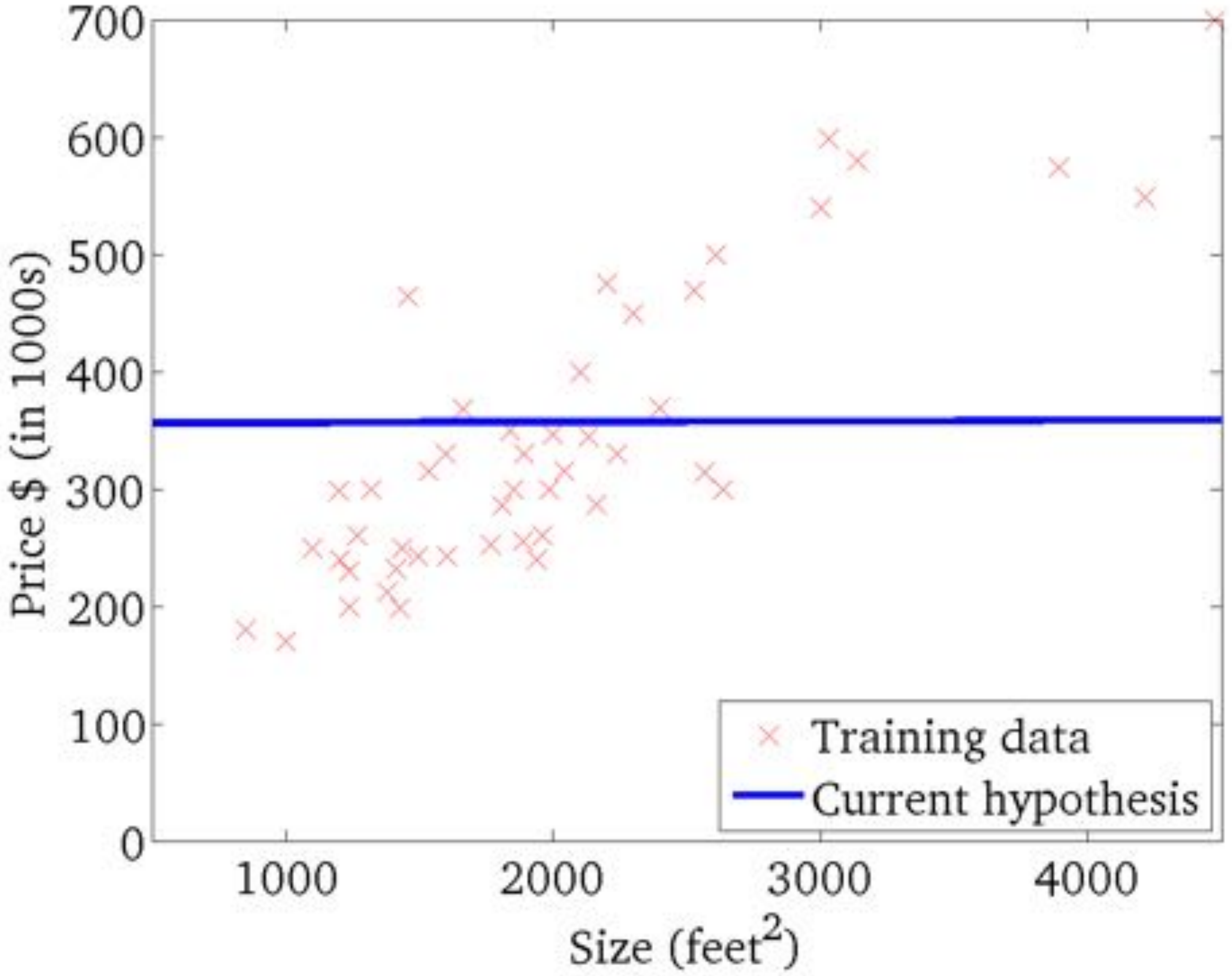
(function of the parameters θ_0, θ_1)



Illustrative Example: GD for Linear Regression

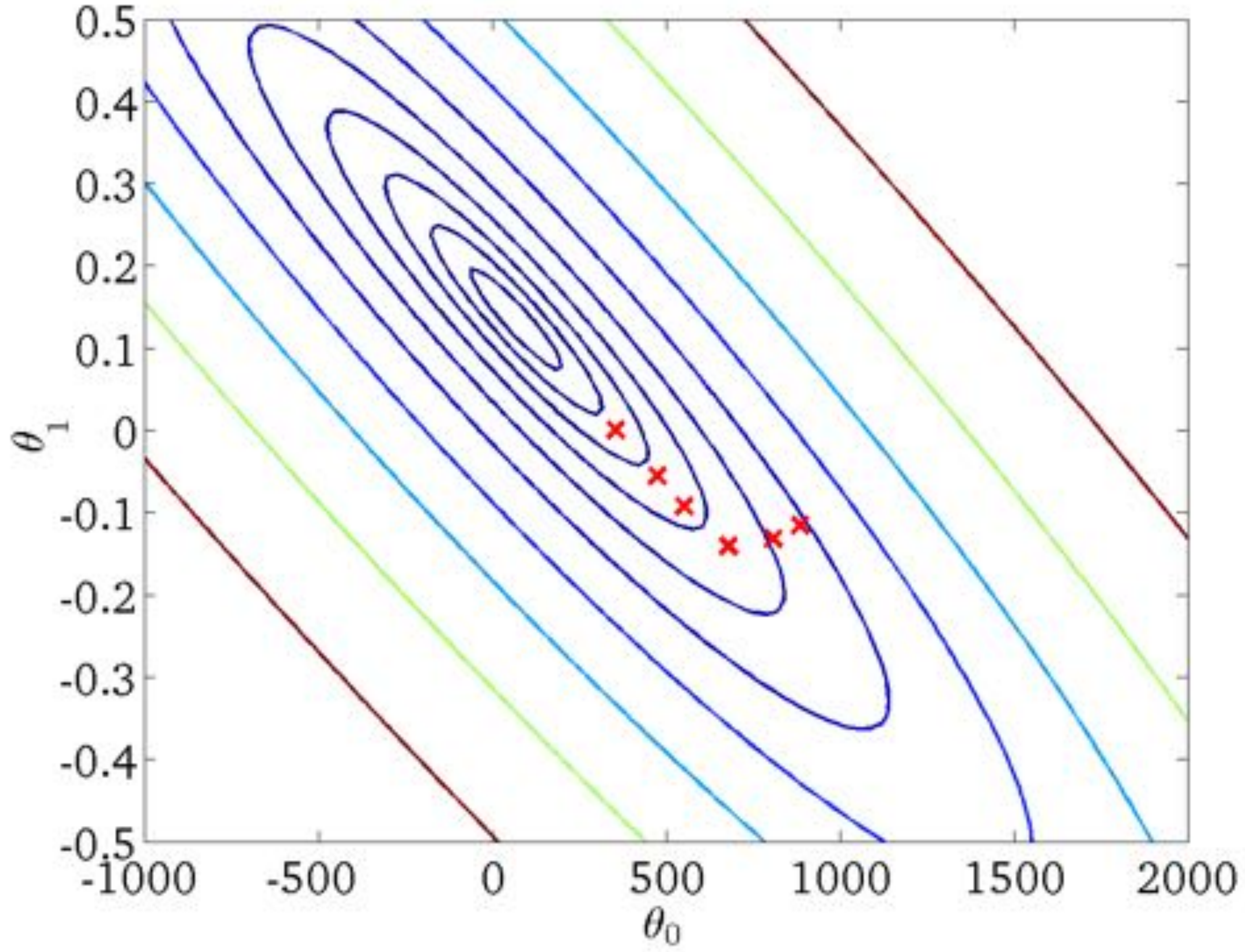
$h_\theta(x)$

(for fixed θ_0, θ_1, this is a function of x)



$J(\theta_0, \theta_1)$

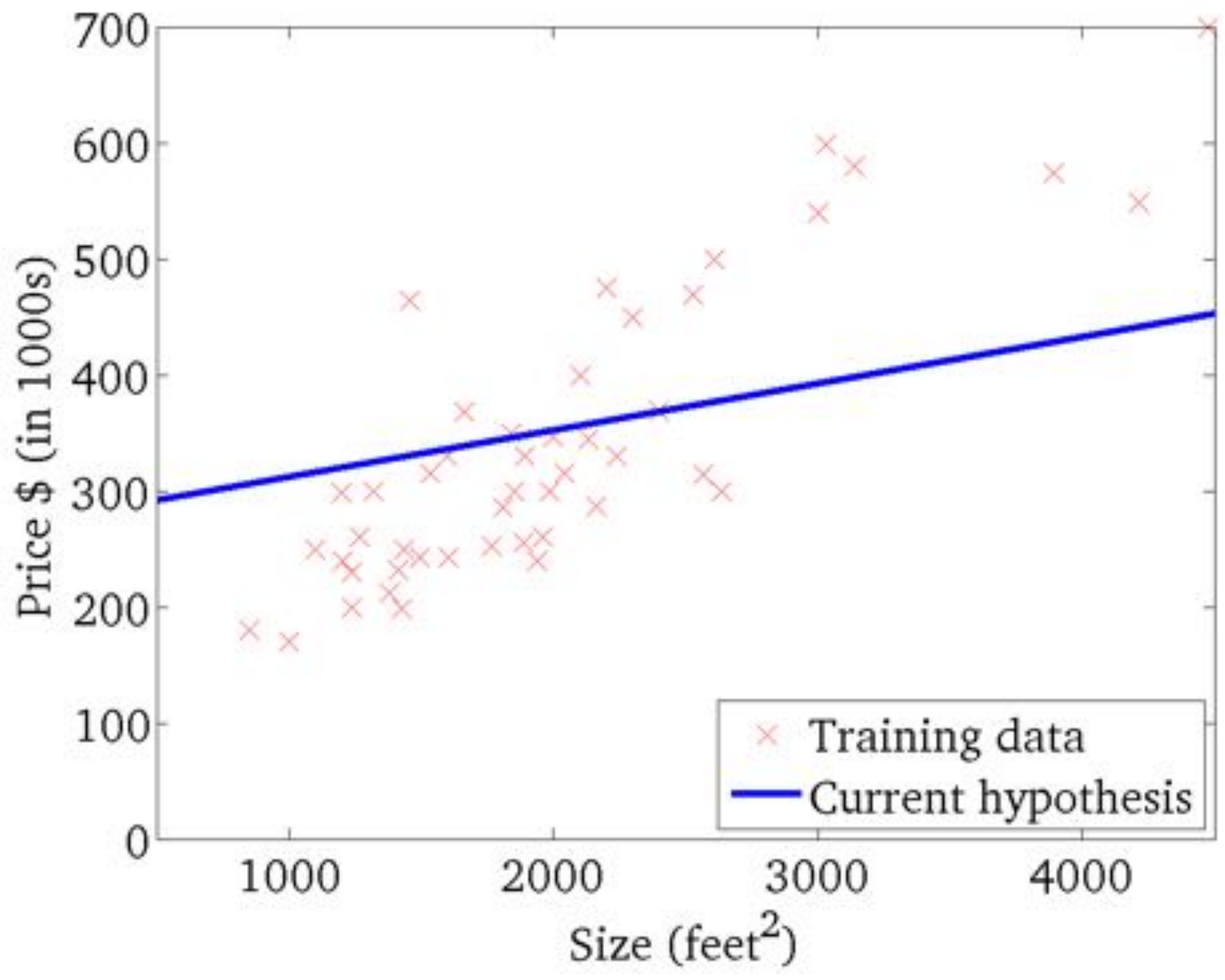
(function of the parameters θ_0, θ_1)



Illustrative Example: GD for Linear Regression

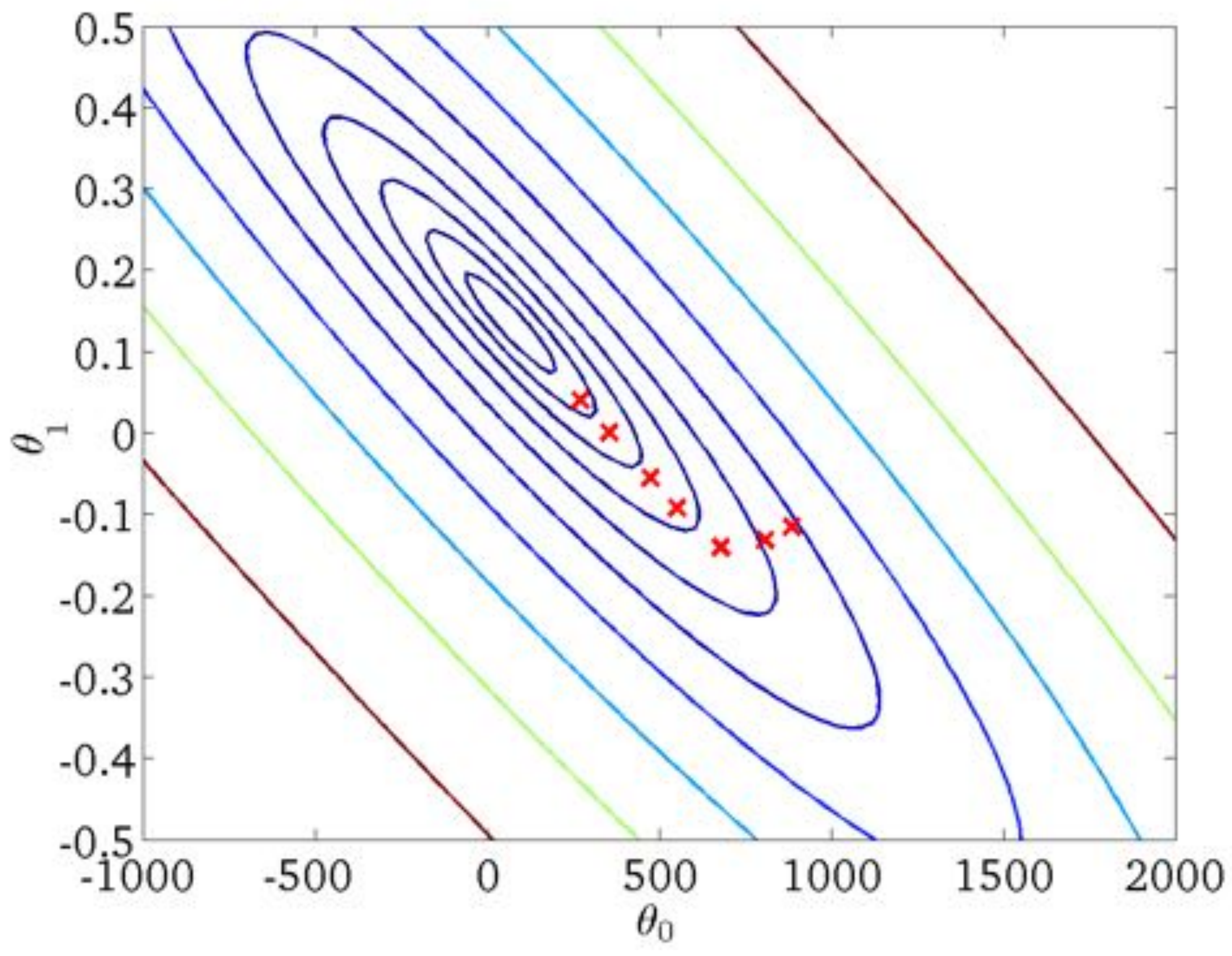
$h_\theta(x)$

(for fixed θ_0, θ_1, this is a function of x)



$J(\theta_0, \theta_1)$

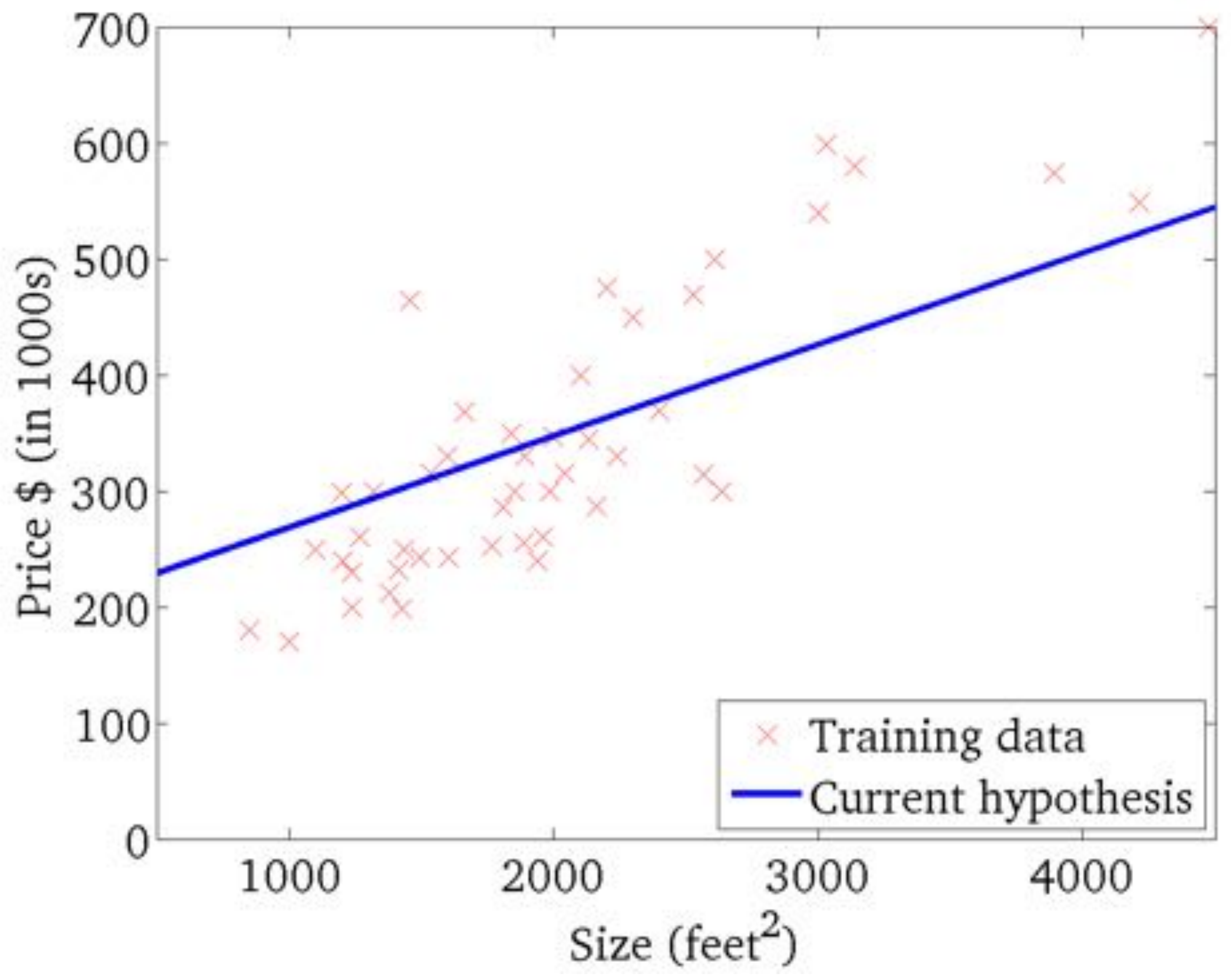
(function of the parameters θ_0, θ_1)



Illustrative Example: GD for Linear Regression

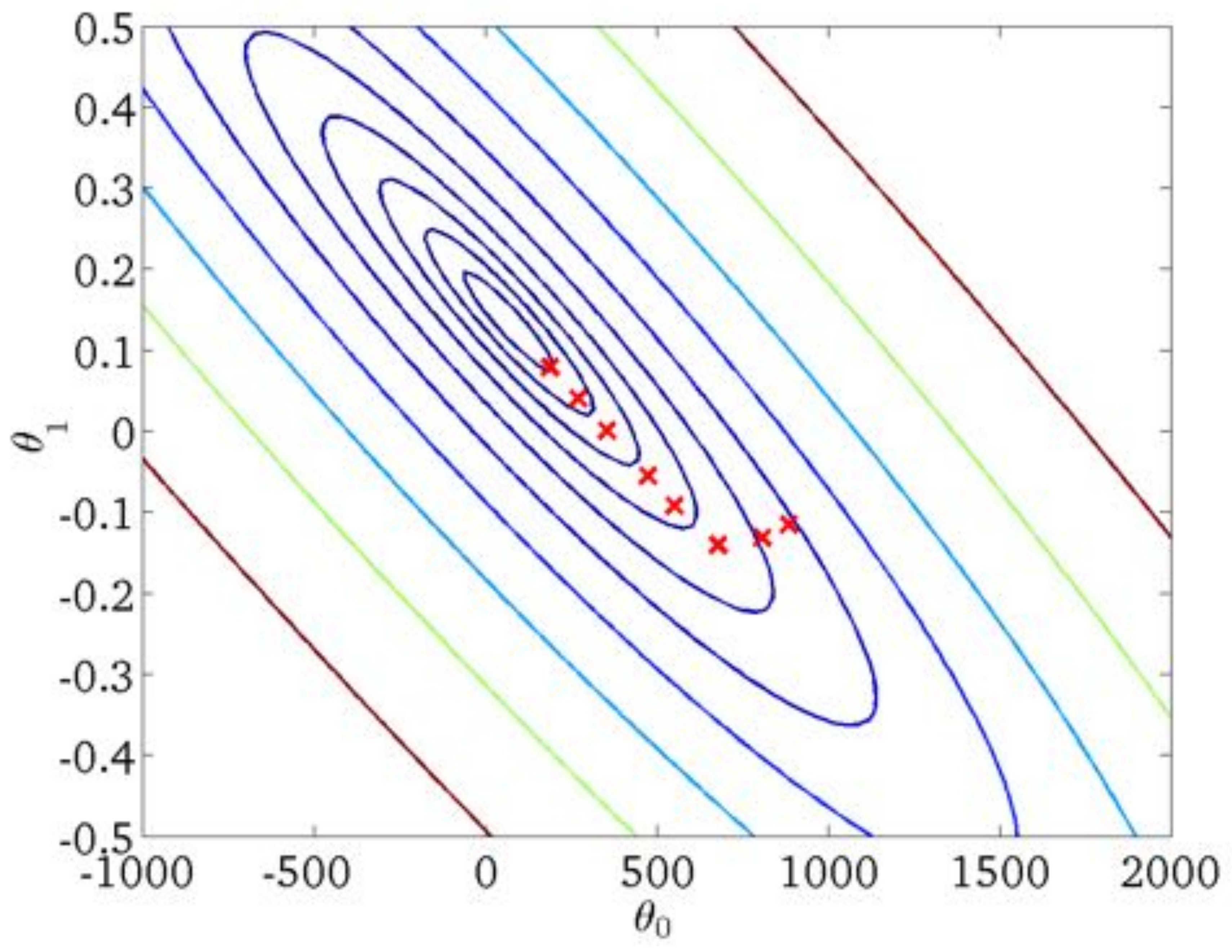
$h_\theta(x)$

(for fixed θ_0, θ_1, this is a function of x)



$J(\theta_0, \theta_1)$

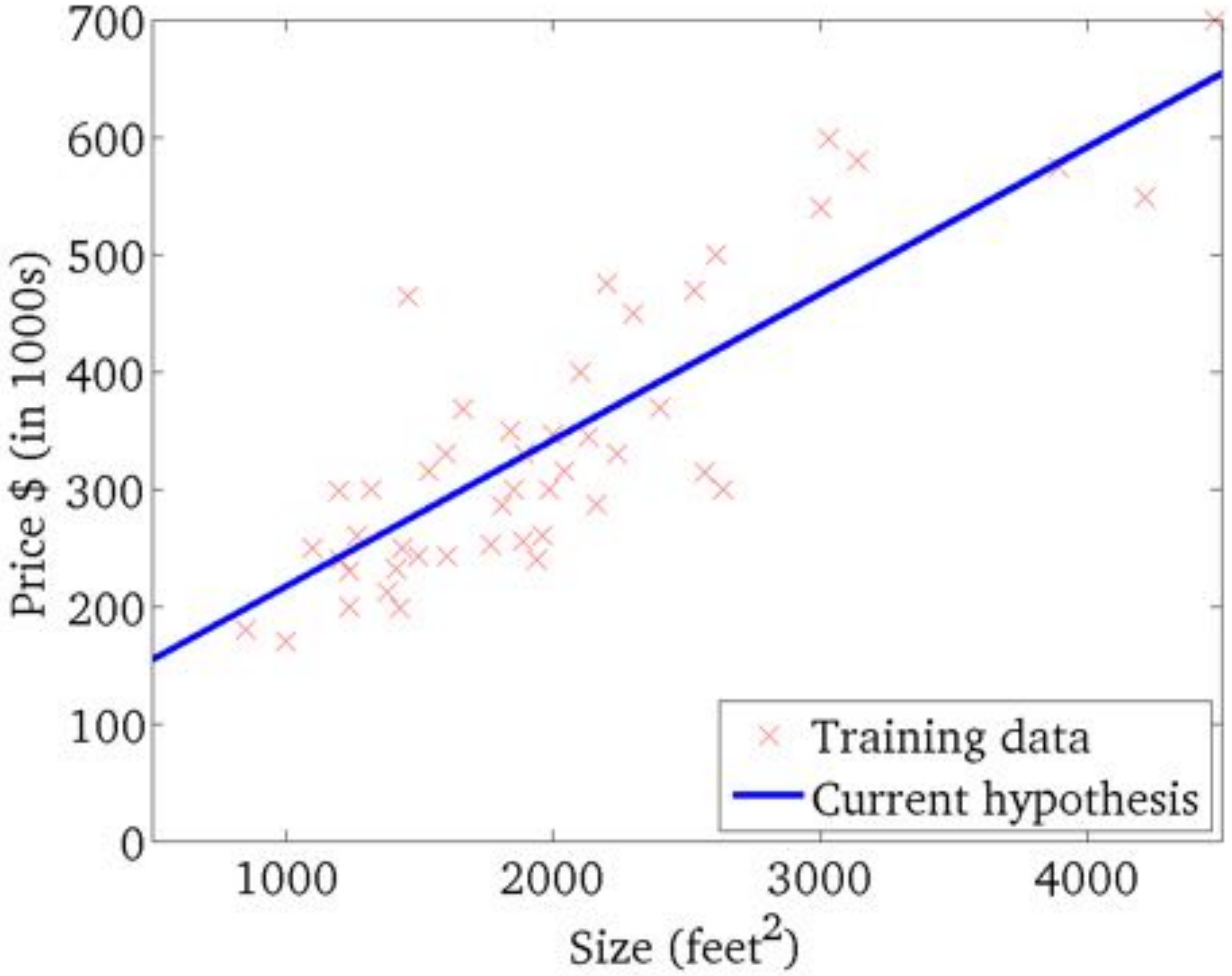
(function of the parameters θ_0, θ_1)



Illustrative Example: GD for Linear Regression

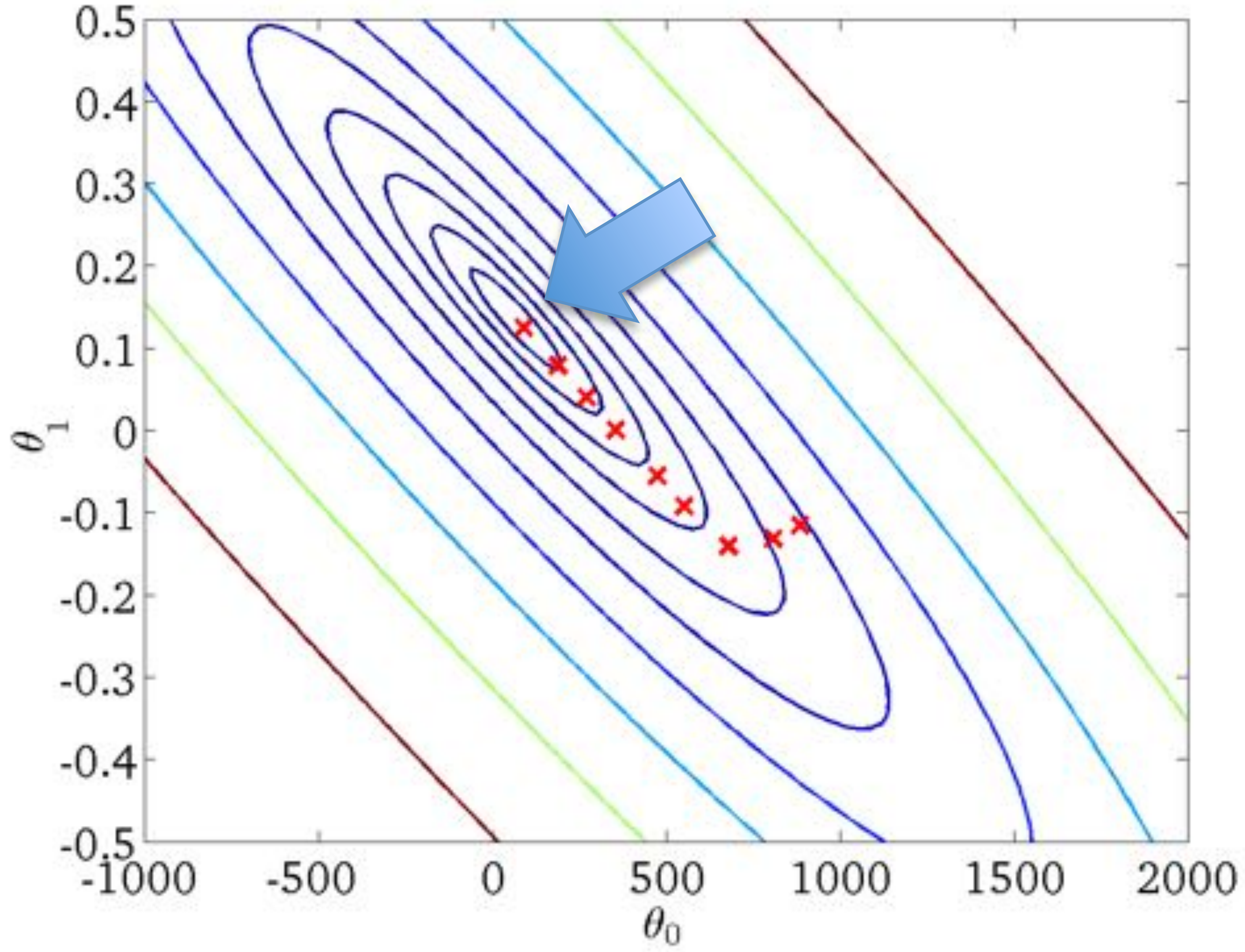
$$h_\theta(x)$$

(for fixed θ_0, θ_1, this is a function of x)



$$J(\theta_0, \theta_1)$$

(function of the parameters θ_0, θ_1)



Feed-Forward Neural Networks

- The information flows through the function being evaluated from the input x, through the intermediate computations used to define f, and finally to the output y.

$$f_\theta(\mathbf{x}) = \mathbf{y}$$

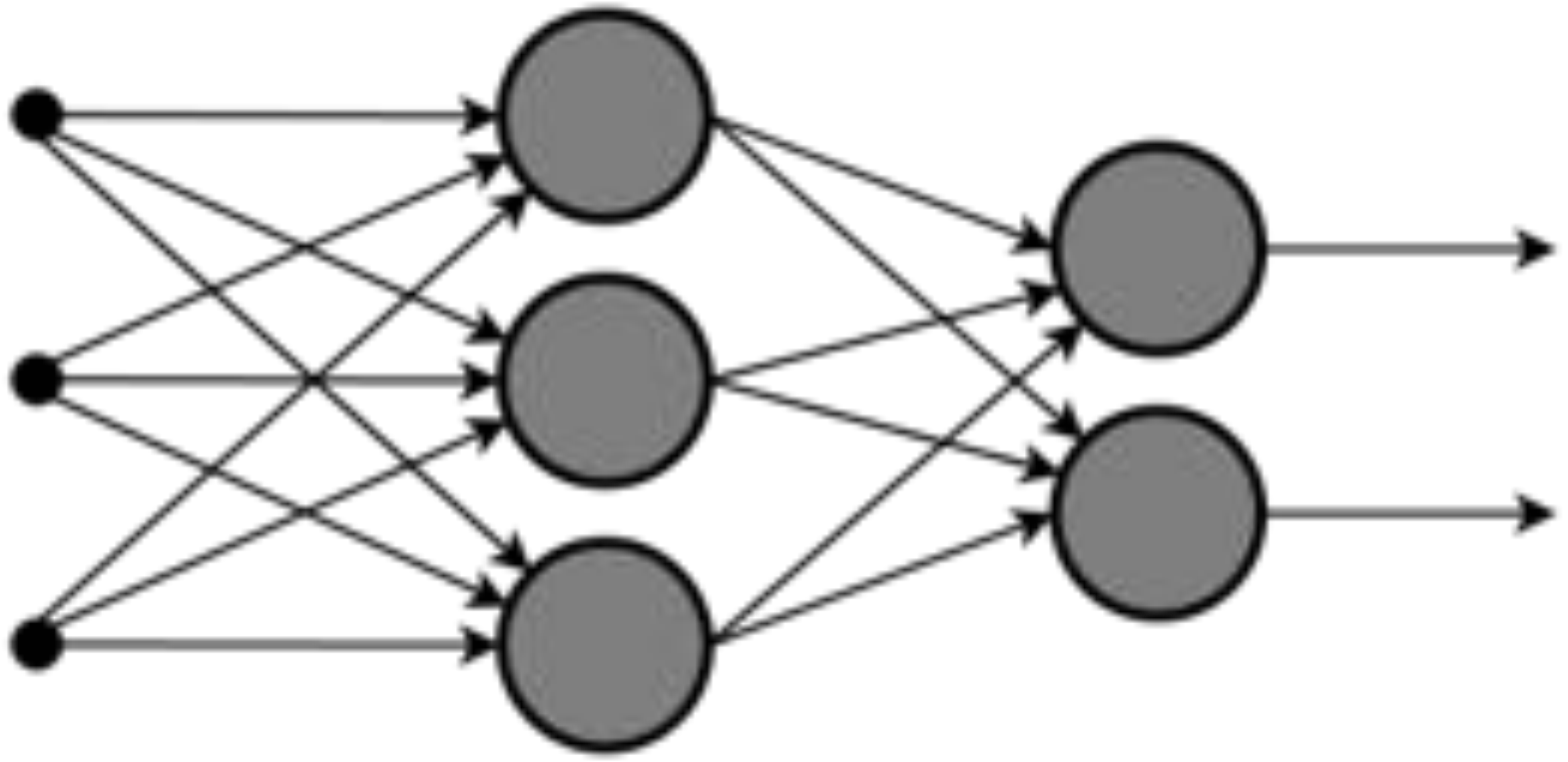
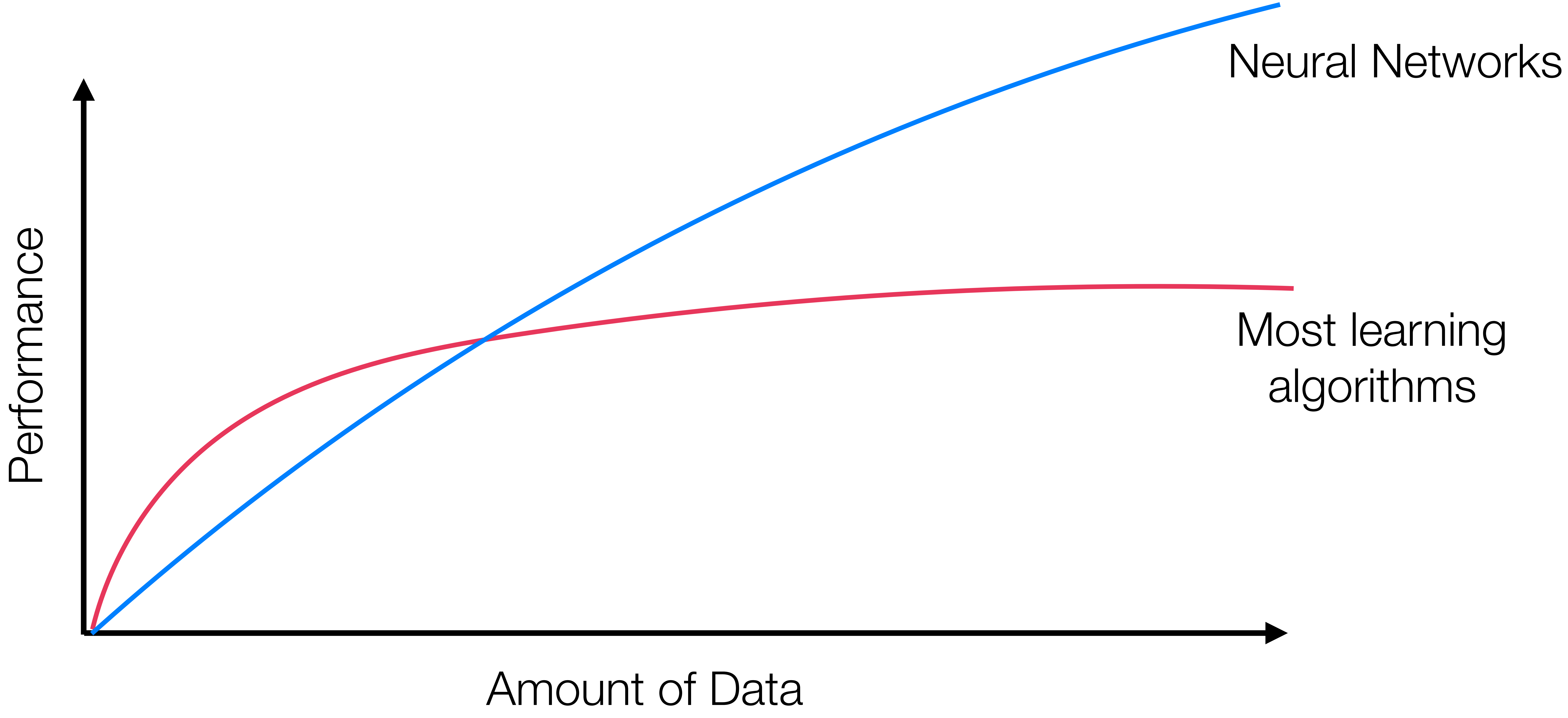
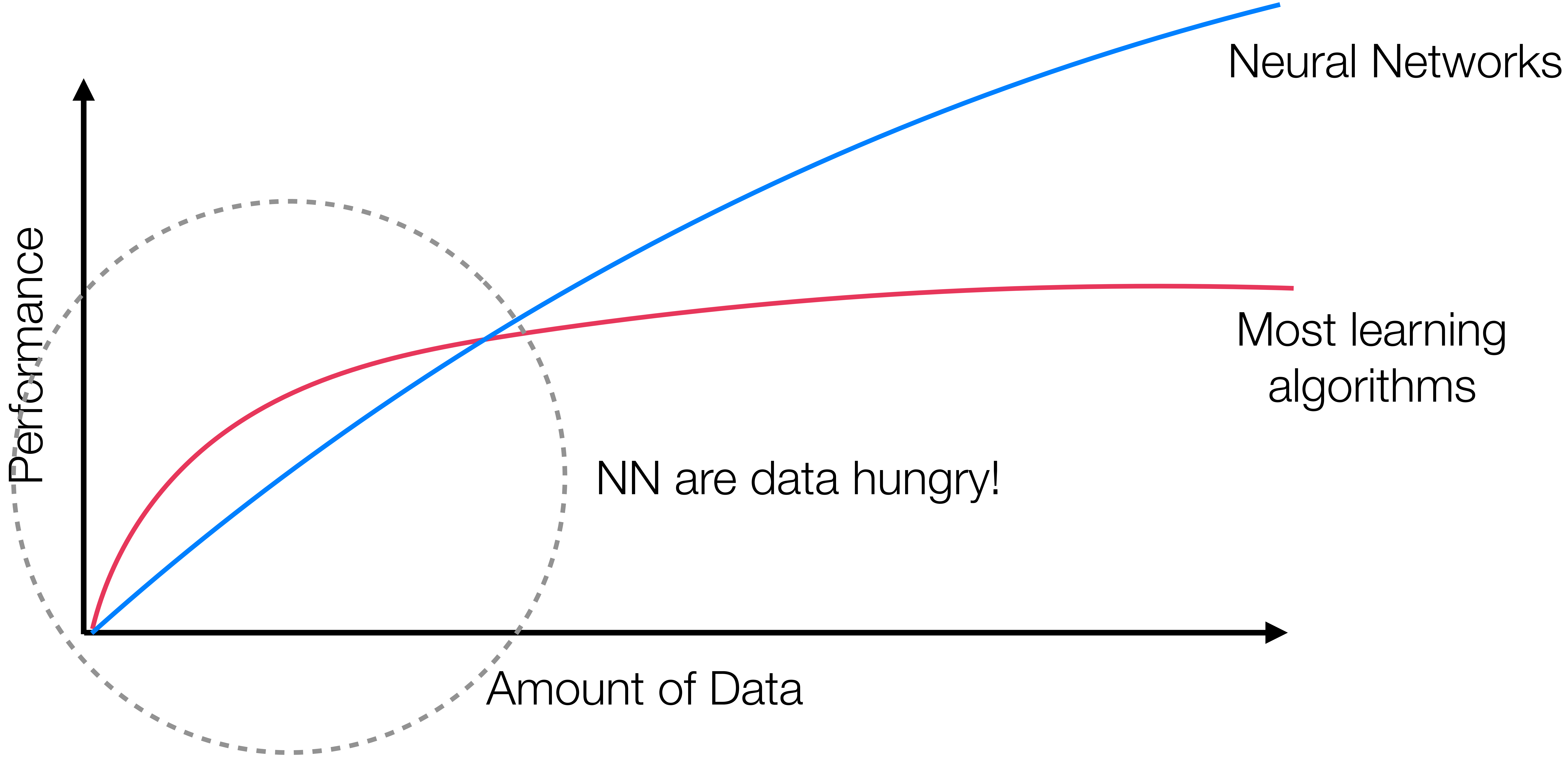


Image from wikibooks

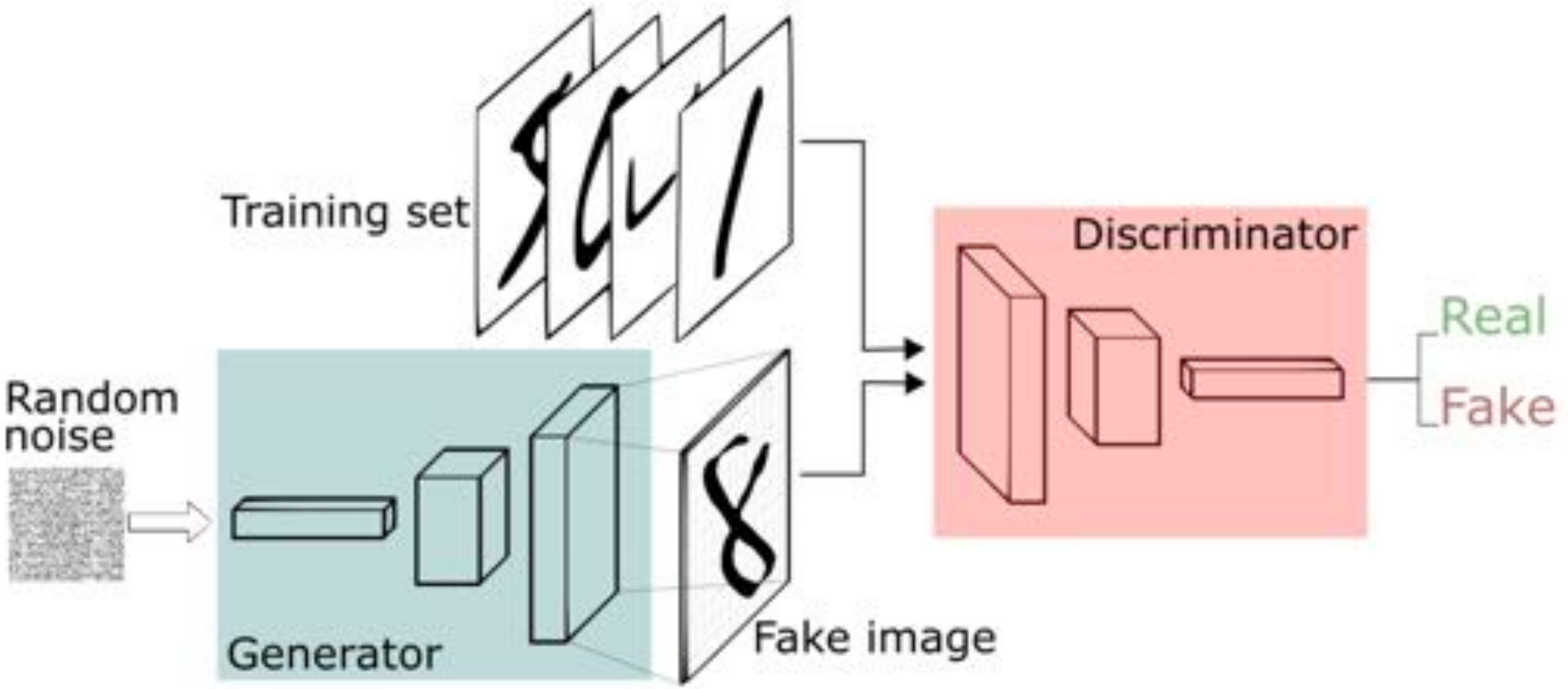
Why Neural Networks?



Why Neural Networks?

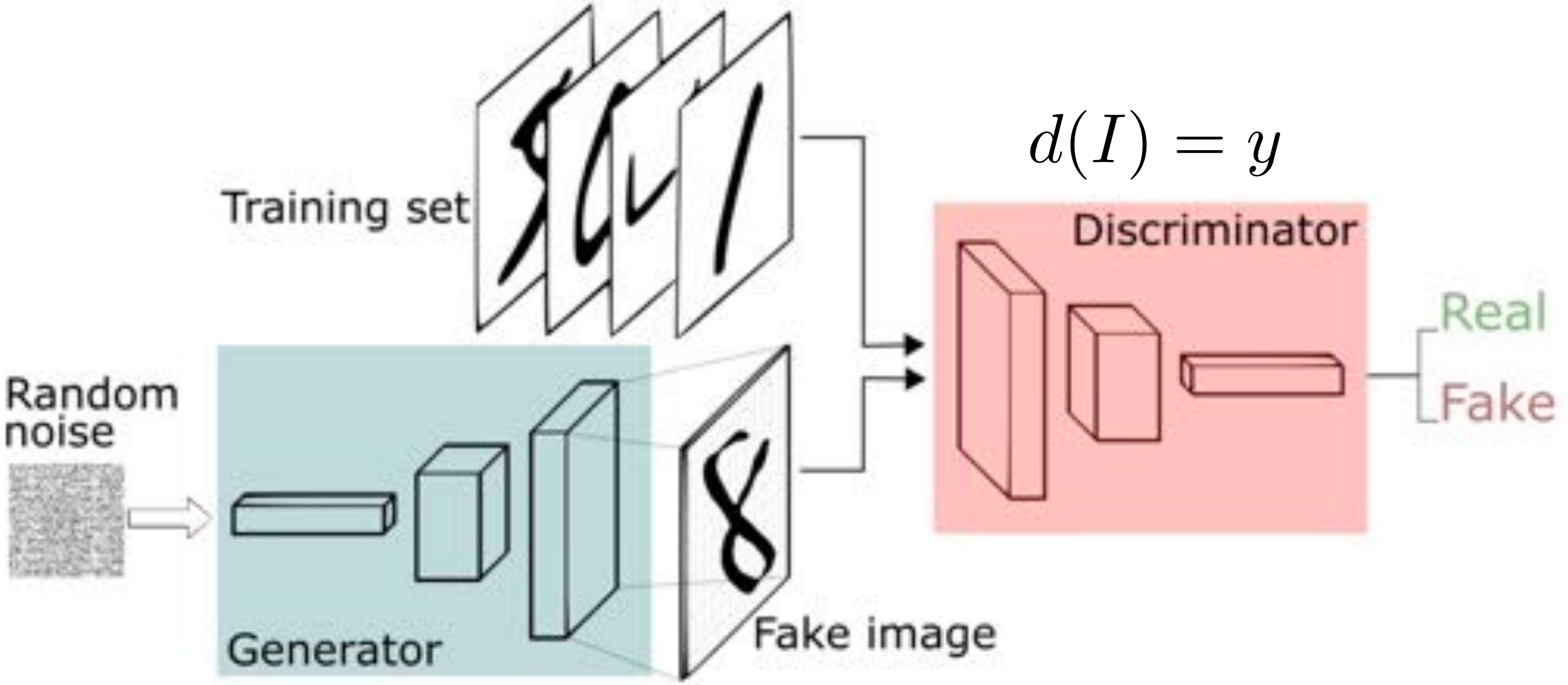


Generative Adversarial Networks



https://skymind.ai/wiki/generative-adversarial-network-gan

Generative Adversarial Networks



$$d(I) = y$$

$$g(\mathbf{z}) = I'$$

https://skymind.ai/wiki/generative-adversarial-network-gan

Function Approximation for Mobile Robotics

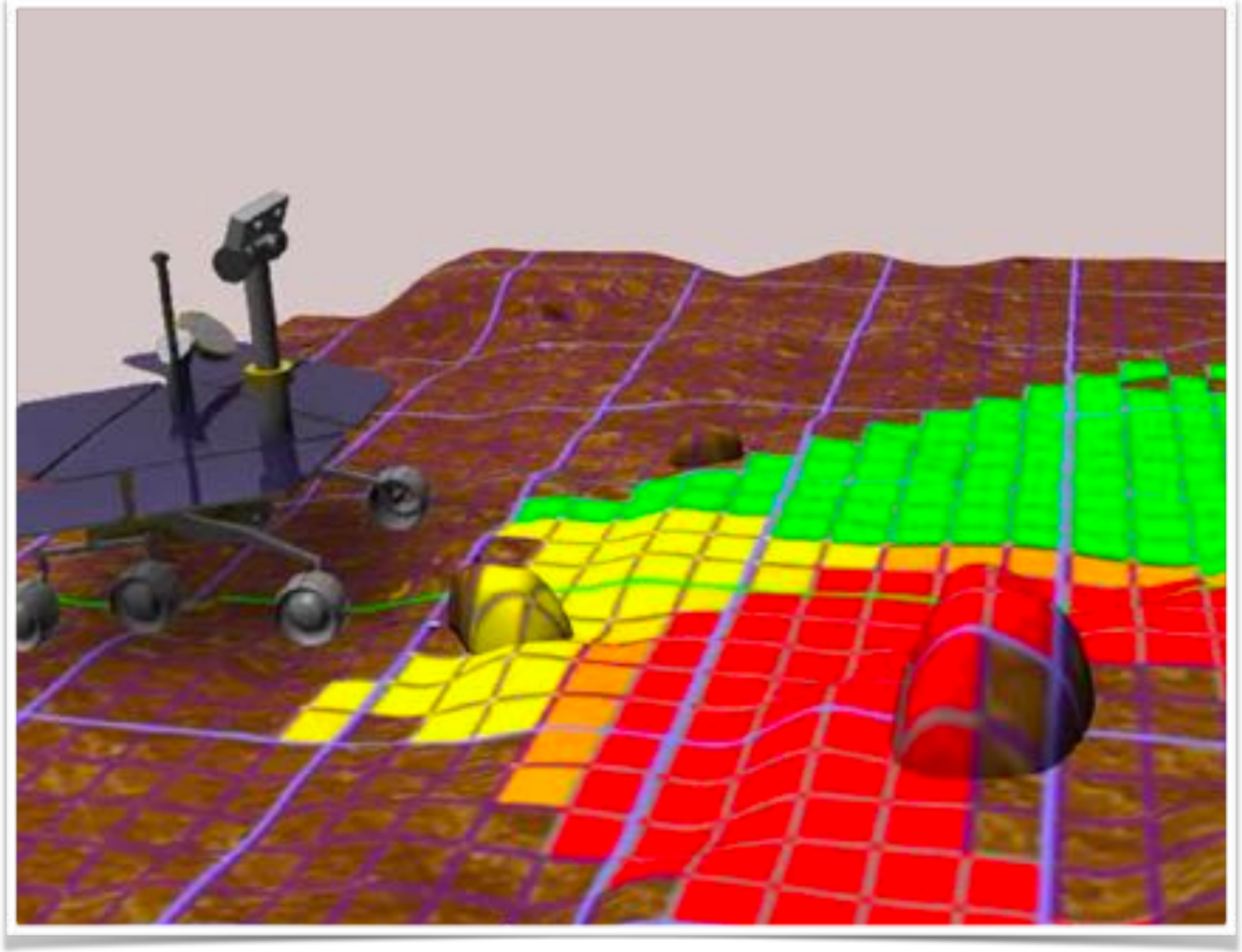
Social Navigation*

* In collaboration with Stanford

- Background
- Fundamental technologies for social navigation
  - Traversability Estimation
  - A-B Navigation in Dynamic Environments
- What might come next?

Traversability Estimation

- Is the space in front of a robot safe to traverse or not?
- Software-based signal for emergency stop.



https://www-robotics.jpl.nasa.gov

THE VERGE

TECH \ ROBOT

TL;DR

DC security robot quits job by drowning itself in a fountain

I'm shook

By Natt Garun | @nattgarun | Jul 17, 2017, 5:14pm EDT

SHARE

Twitter.com/greggpinelo

I'm sorry to inform you that perhaps some robots are taking this whole "be more human" thing a bit too far. I've seen people make jokes about jumping into a river or out of a window when they feel distressed, but one DC-based security robot appears to have interpreted this on a literal level.

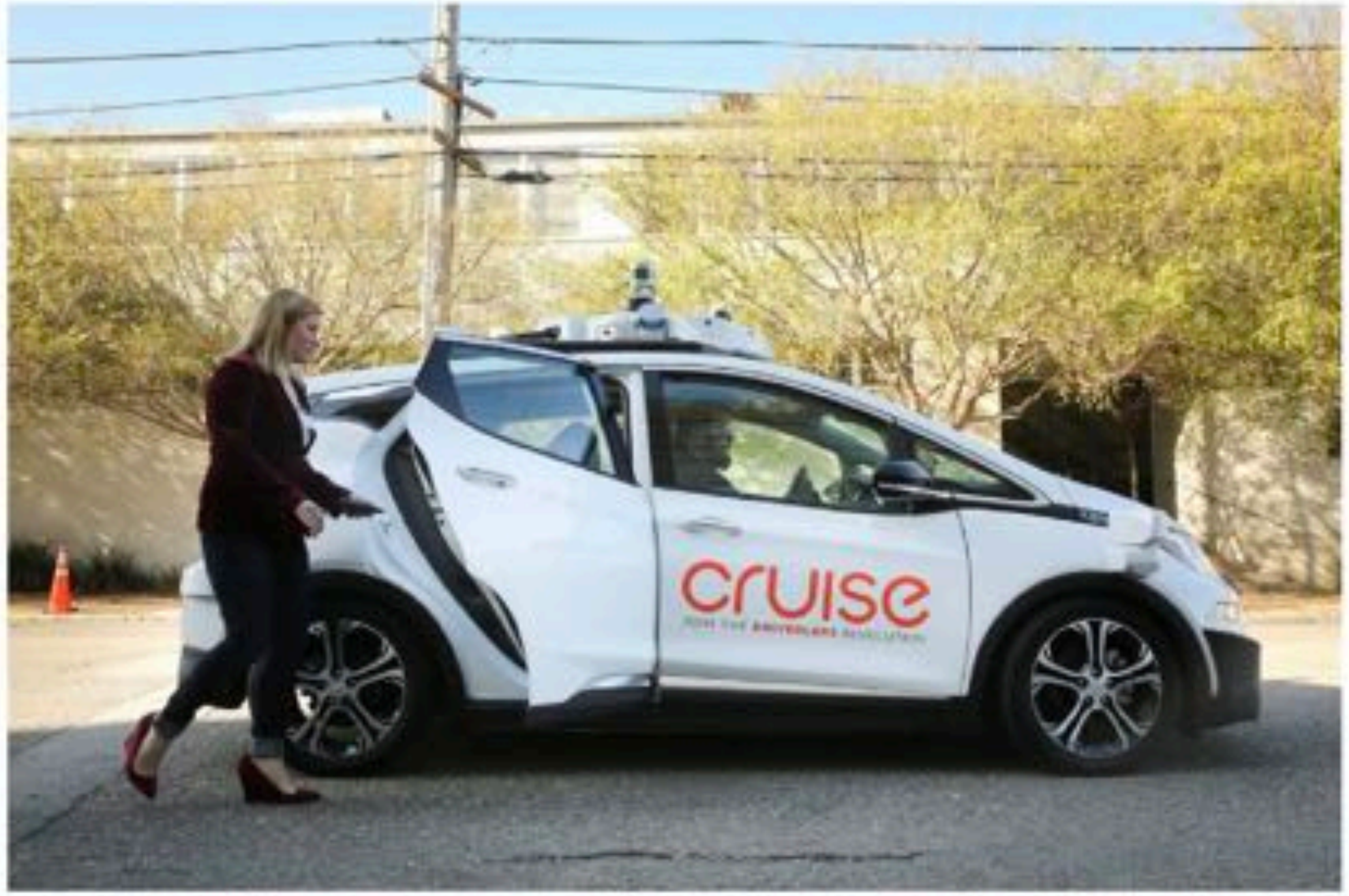
The Washington Post

Democracy Dies in Darkness

Innovations

After crash, injured motorcyclist accuses robot-driven vehicle of 'negligent driving'

By Peter Holley | Email the author



A woman gets in a self-driving Chevy Bolt EV car during a media event by Cruise, GM's autonomous car unit, in San Francisco on Nov. 28, 2017. (Elijah Nouvelage/Reuters)

A California motorcyclist has filed a lawsuit against General Motors, accusing one of the manufacturer's robot-operated vehicles of "negligent driving."

Oscar Willhelm Nilsson claims he was traveling down a San Francisco street last month when a Cruise AV aborted a lane change and swerved into his lane. The car struck him, "knocking him to the ground," in a

Best Paper Award Finalist
at IROS'18

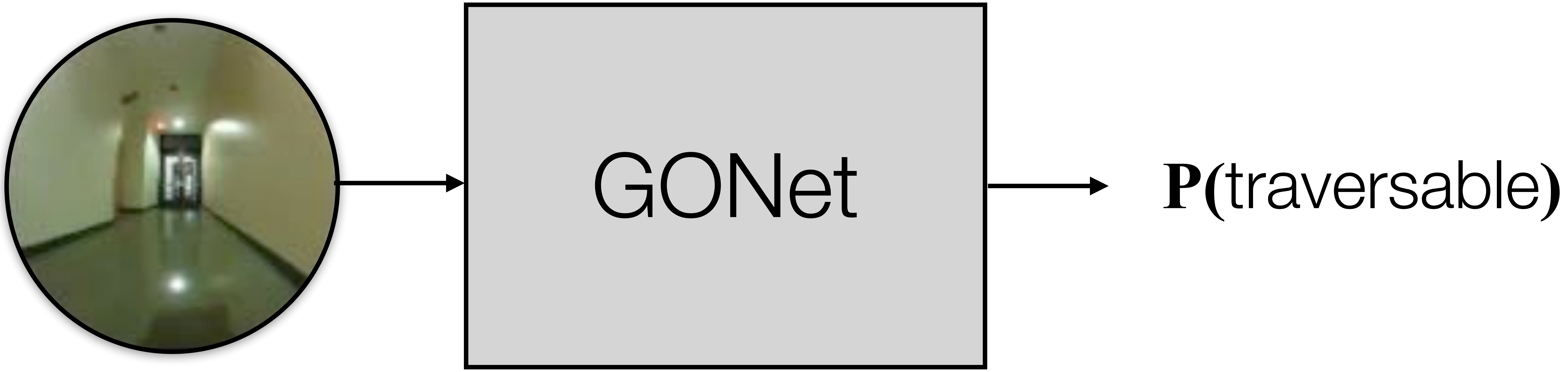
GONet:
Semi-Supervised Traversability Estimation

Fisheye Camera

[Hirose et al., 2018]

Best Paper Award Finalist
at IROS'18

GONet:
Semi-Supervised Traversability Estimation



[Hirose et al., 2018]

Challenging Scenarios

Stairs

Fences

Windows

[Hirose et al., 2018]

Challenging Scenarios

Stairs

Fences

Windows

[Hirose et al., 2018]

Challenging Scenarios

Stairs

Fences

Windows

[Hirose et al., 2018]

Challenging Scenarios

Stairs

Fences

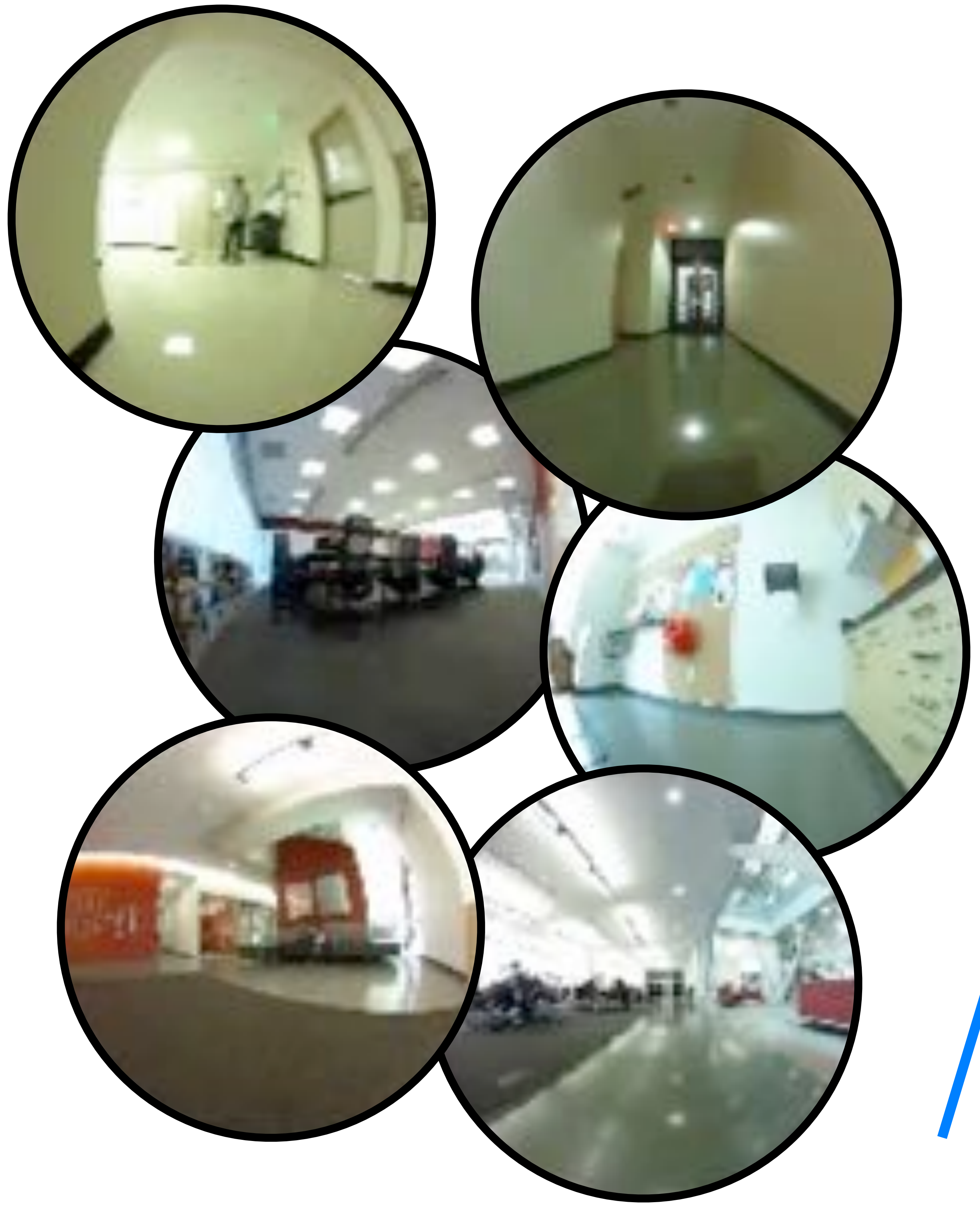
Windows

[Hirose et al., 2018]

Traversability Estimation as Anomaly Detection

Expected Input

Traversable

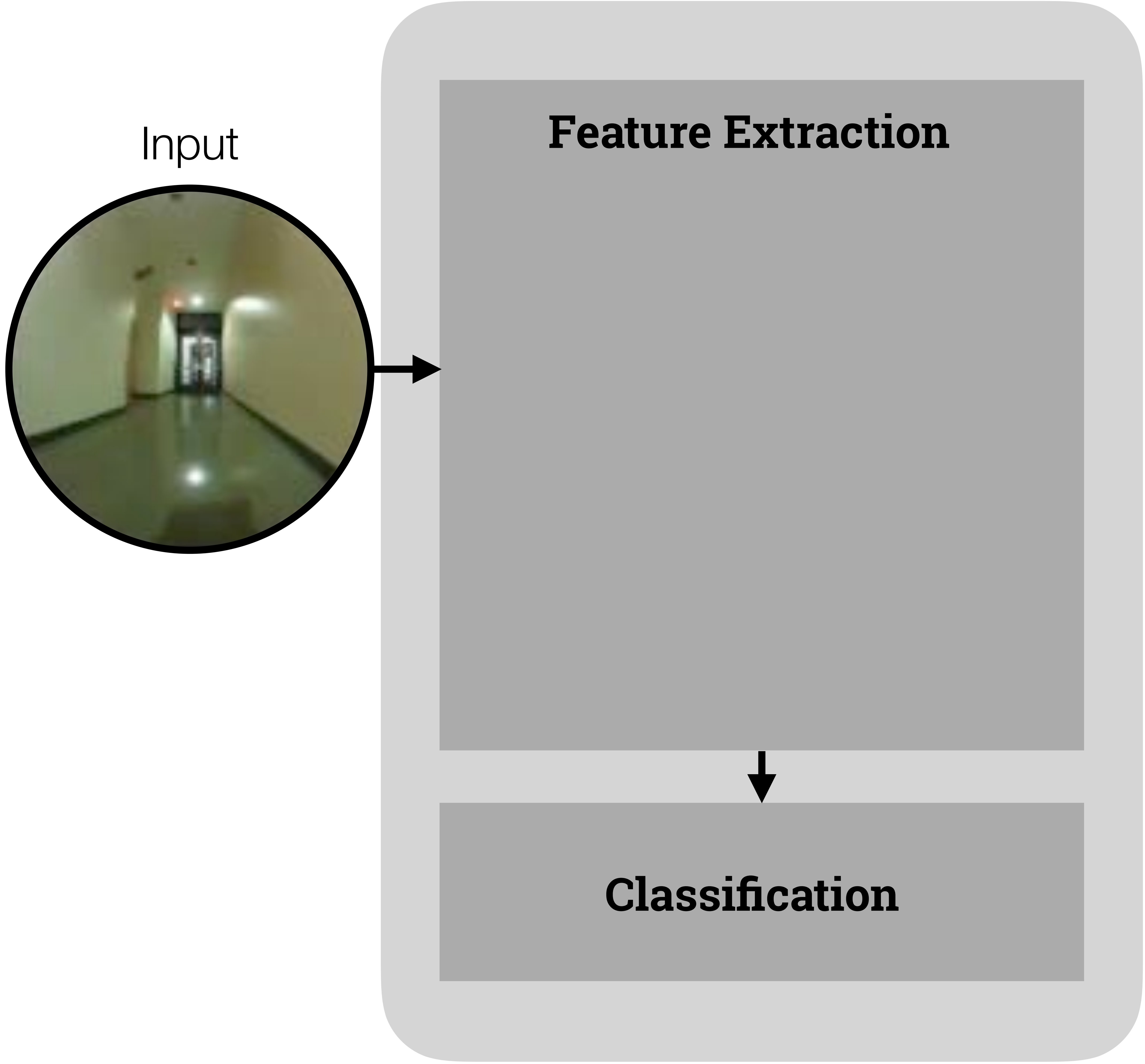


Outliers

Non-traversable

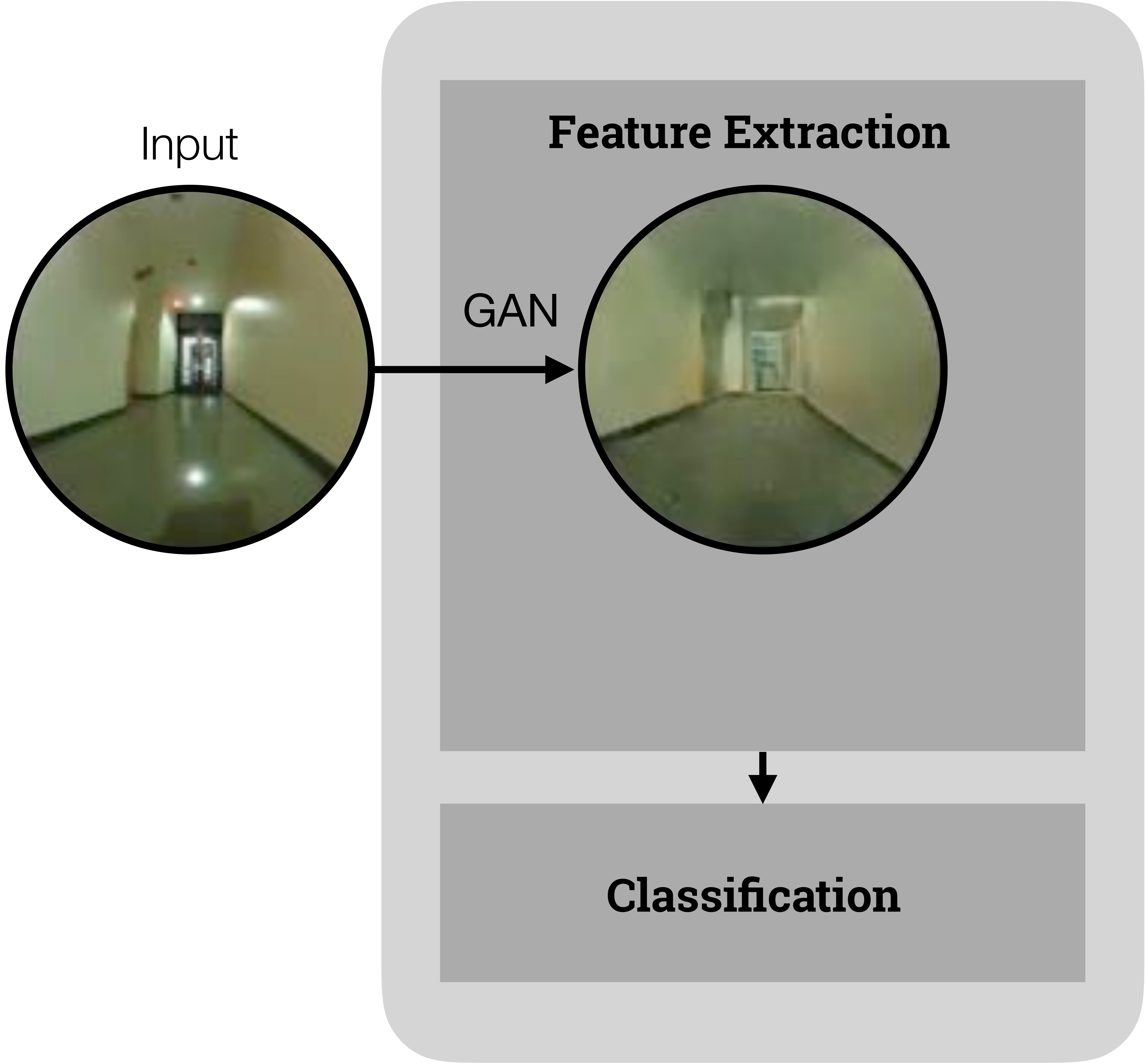
[Hirose et al., 2018]

GONet



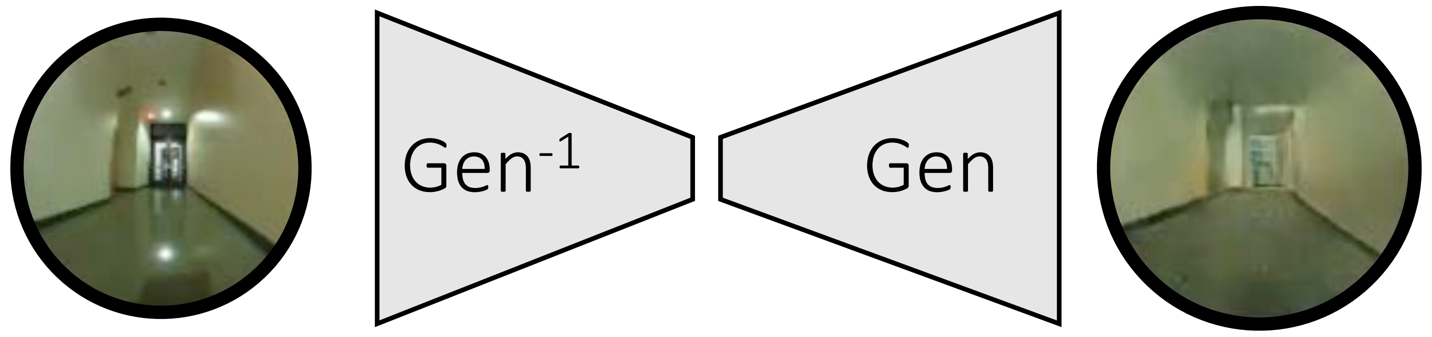
[Hirose et al., 2018]

GONet



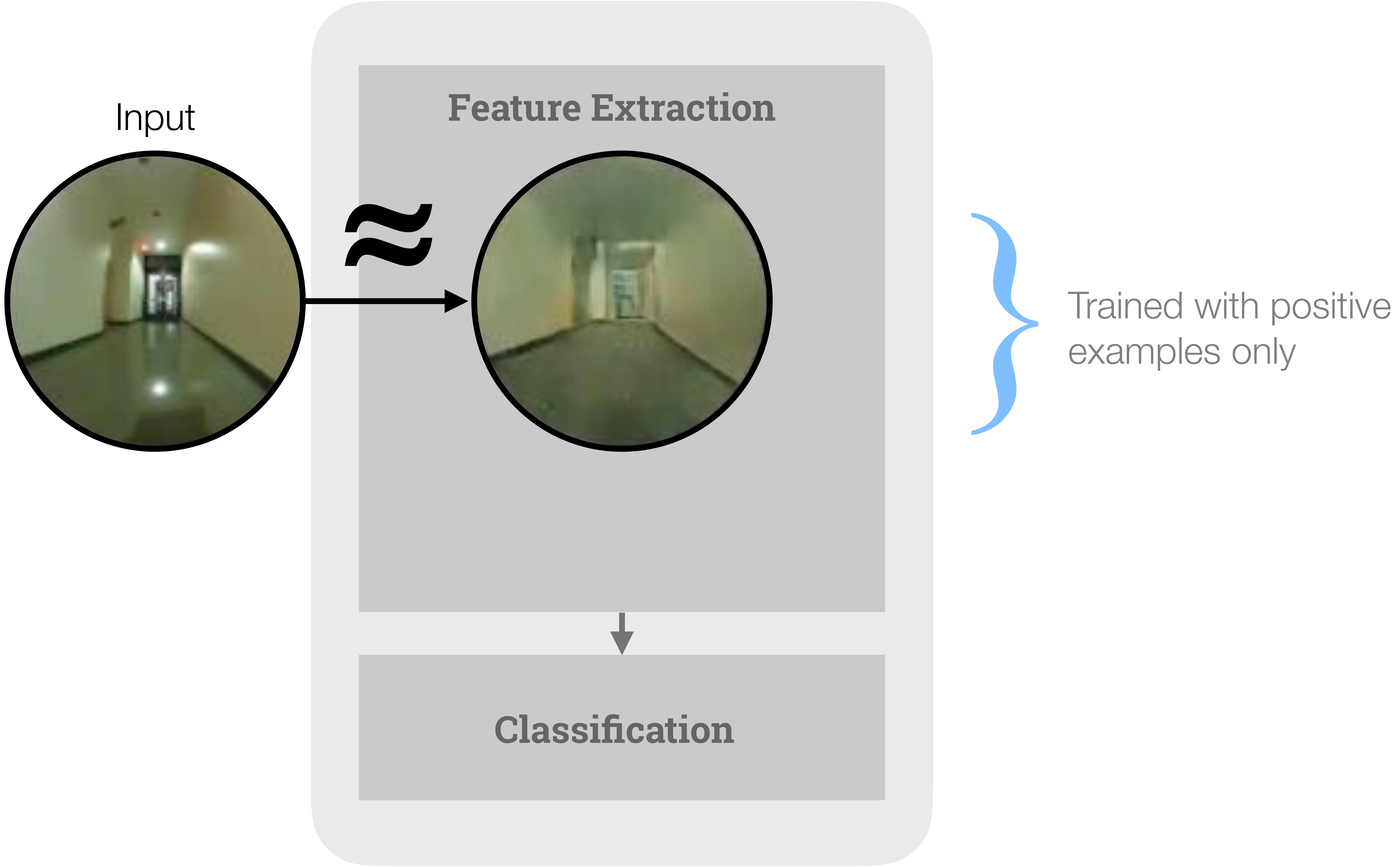
Trained with positive examples only

We induce the generated image to look like the input [Zhu et al., 2016].

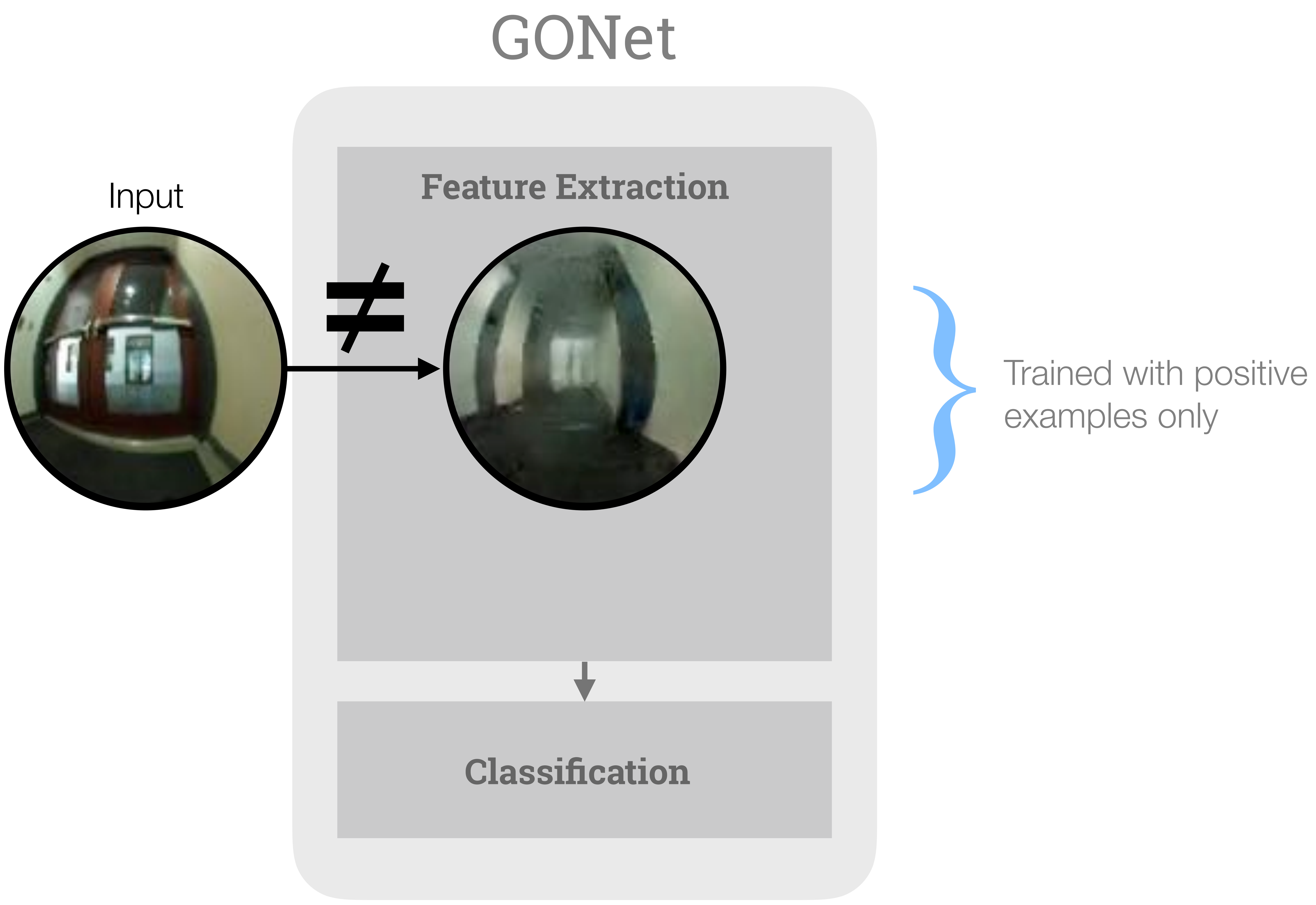


[Hirose et al., 2018]

GONet

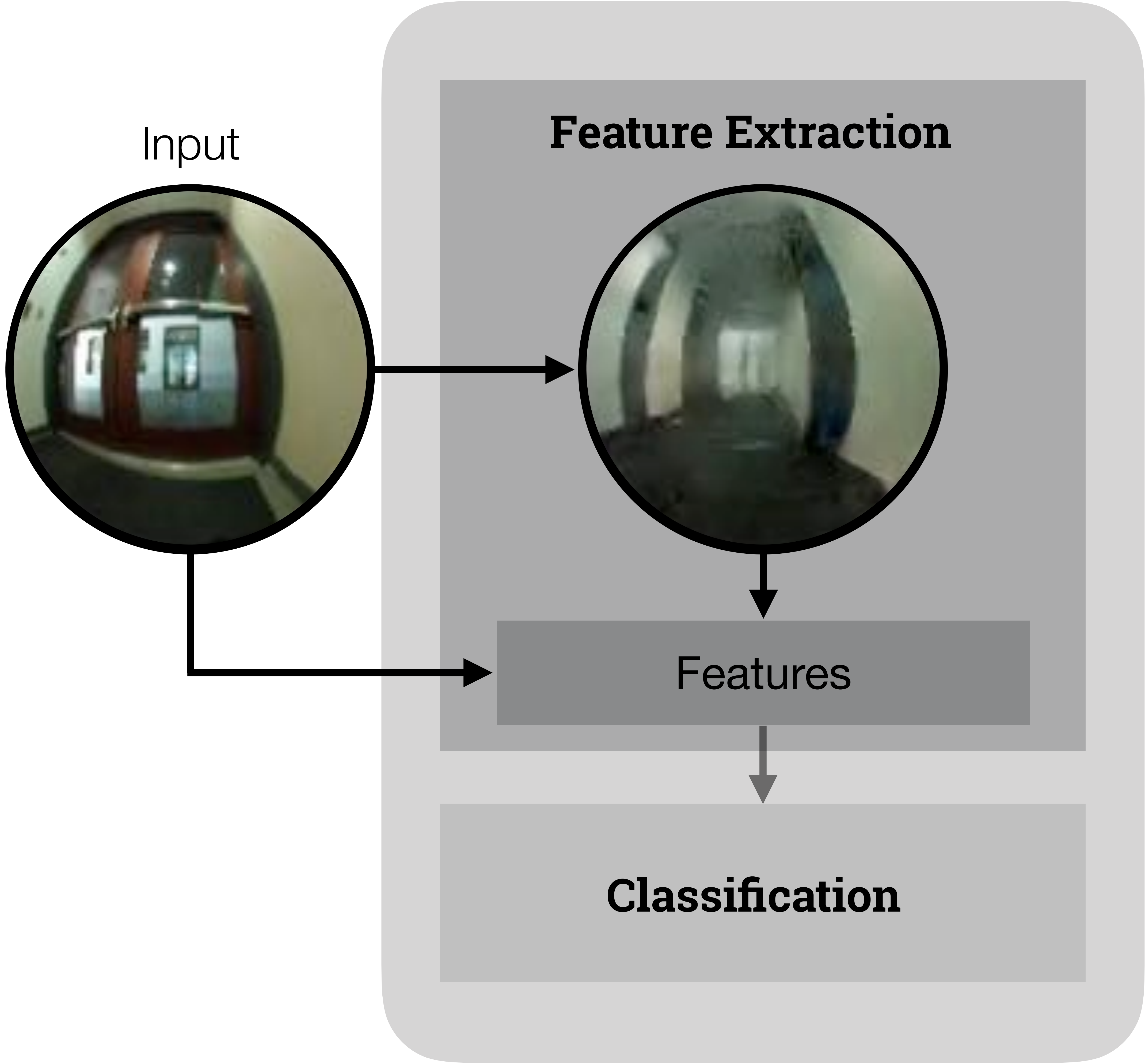


[Hirose et al., 2018]



[Hirose et al., 2018]

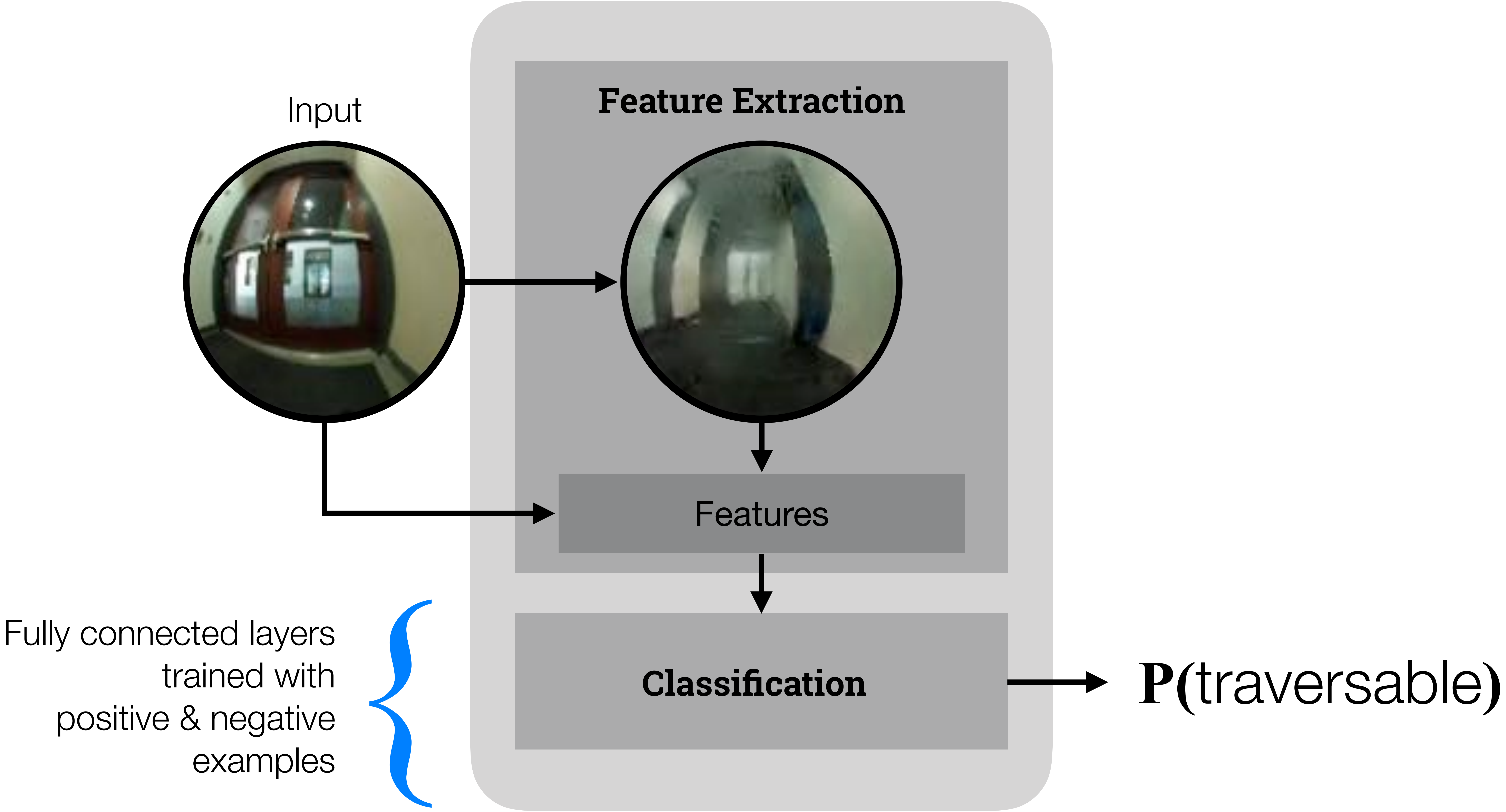
GONet



Simple features, e.g.,
difference of images,
discriminator features

[Hirose et al., 2018]

GONet



[Hirose et al., 2018]

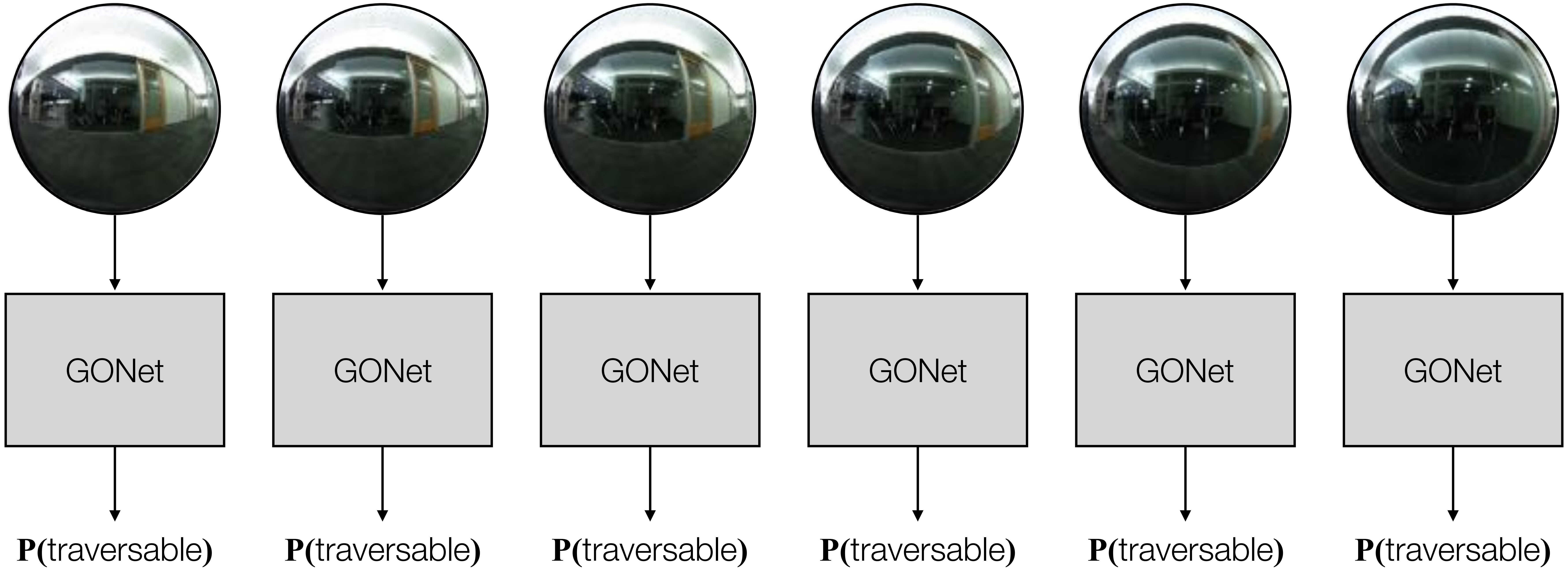
Predicted Images

Input Image

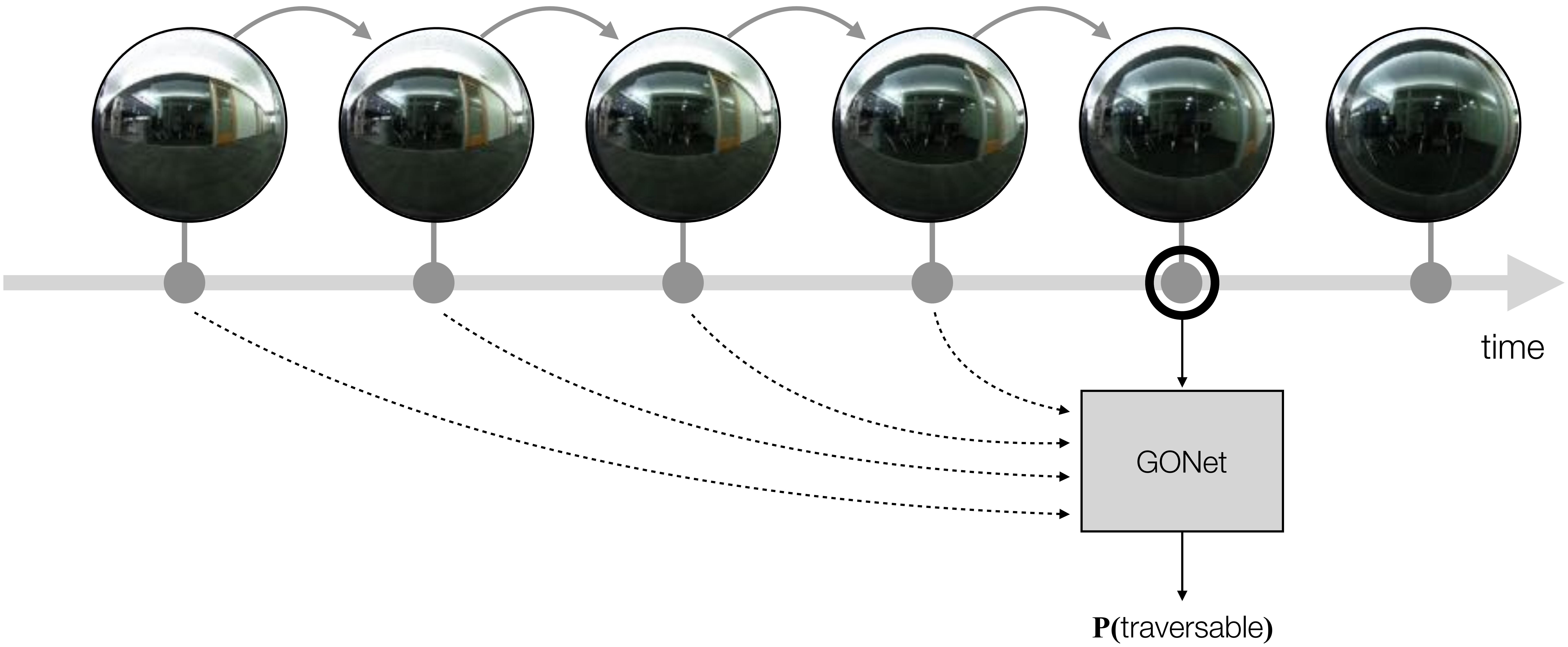
Predicted Image

GONet Extensions

GONet assumed i.i.d. input images

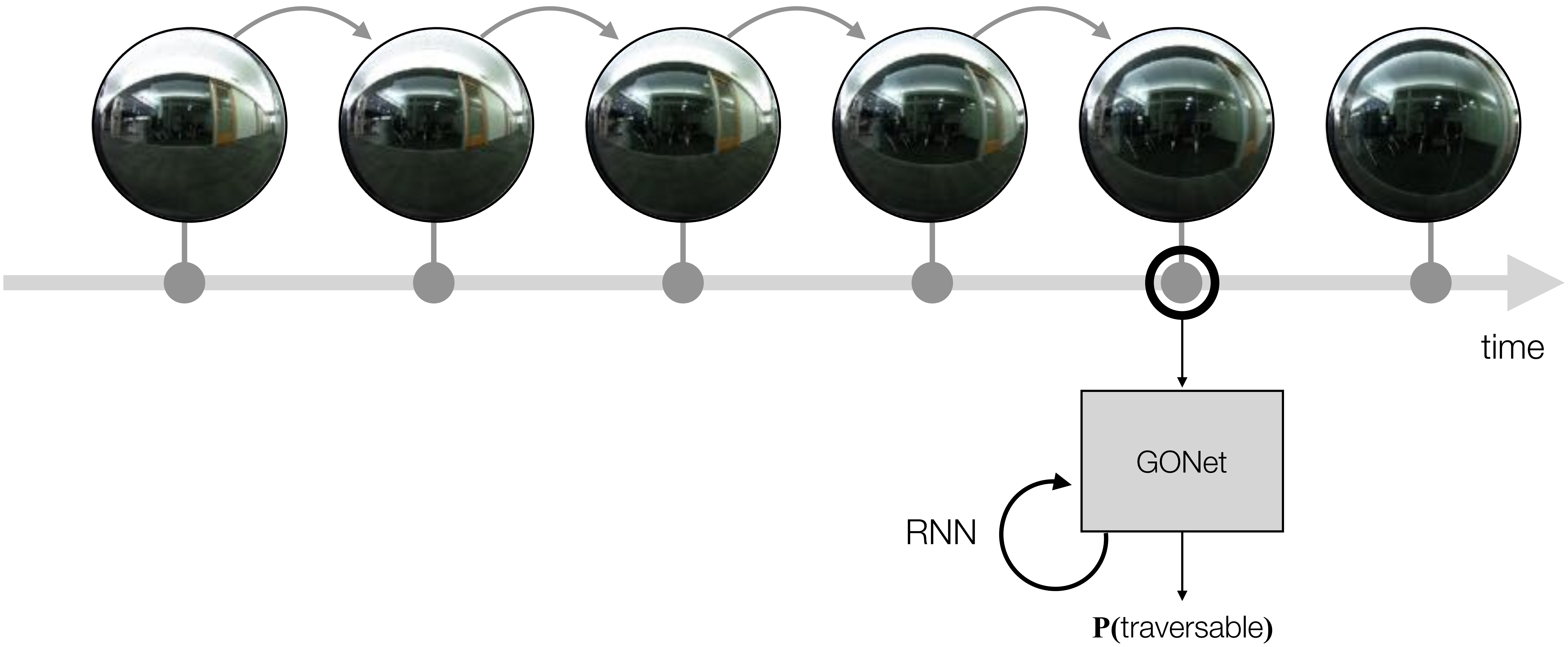


[Hirose et al., 2018]



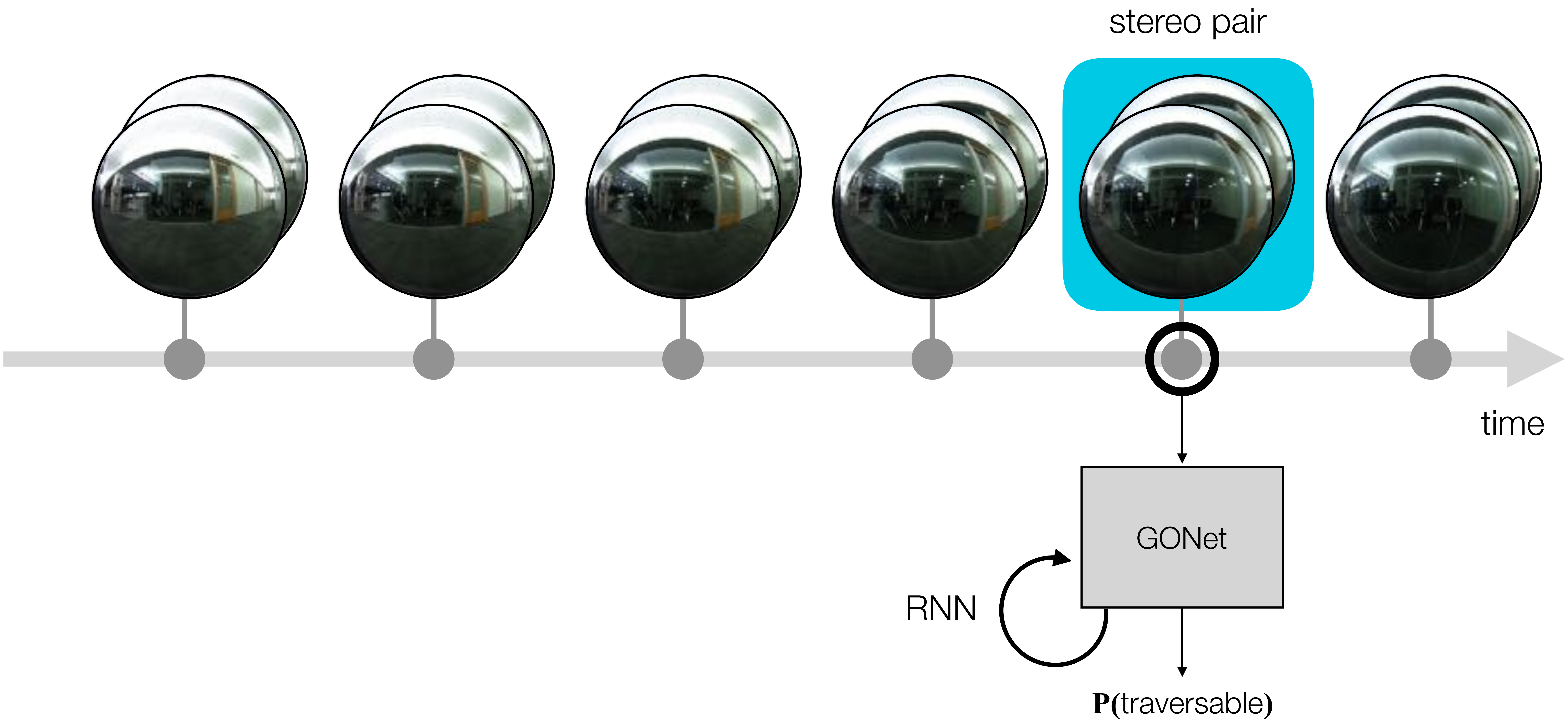
[Hirose et al., 2018]

GONet+T: Enforcing Temporal Consistency



[Hirose et al., 2018]

GONet+TS: Traversability From Stereo Views



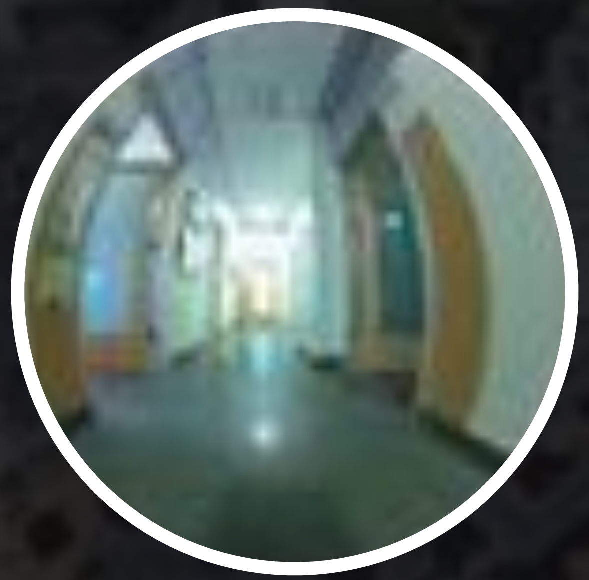
[Hirose et al., 2018]

Datasets


Stanford University

GO Stanford 1

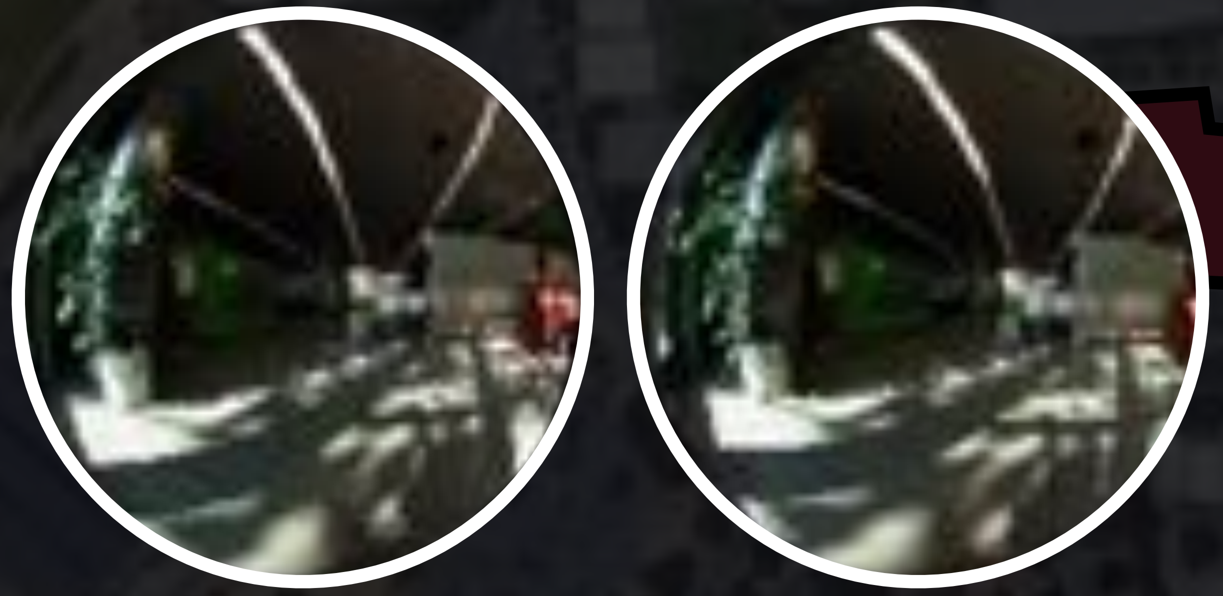
- Single fisheye camera



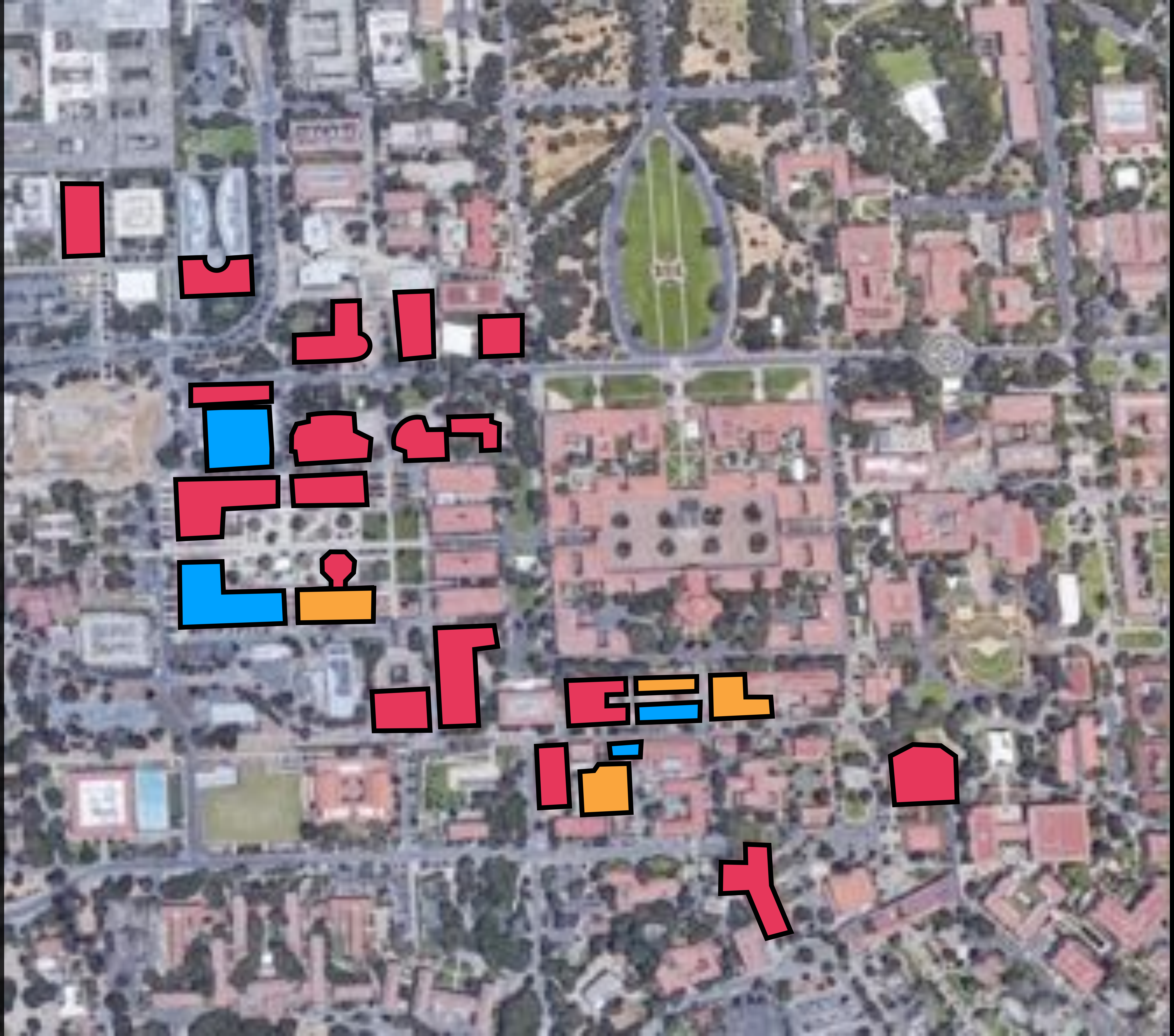
- 15 buildings
  - 9 for training
  - 3 for validation
  - 3 for testing
- ~36K images
- 2400 labeled images

GO Stanford 2

- Stereo fisheye pair

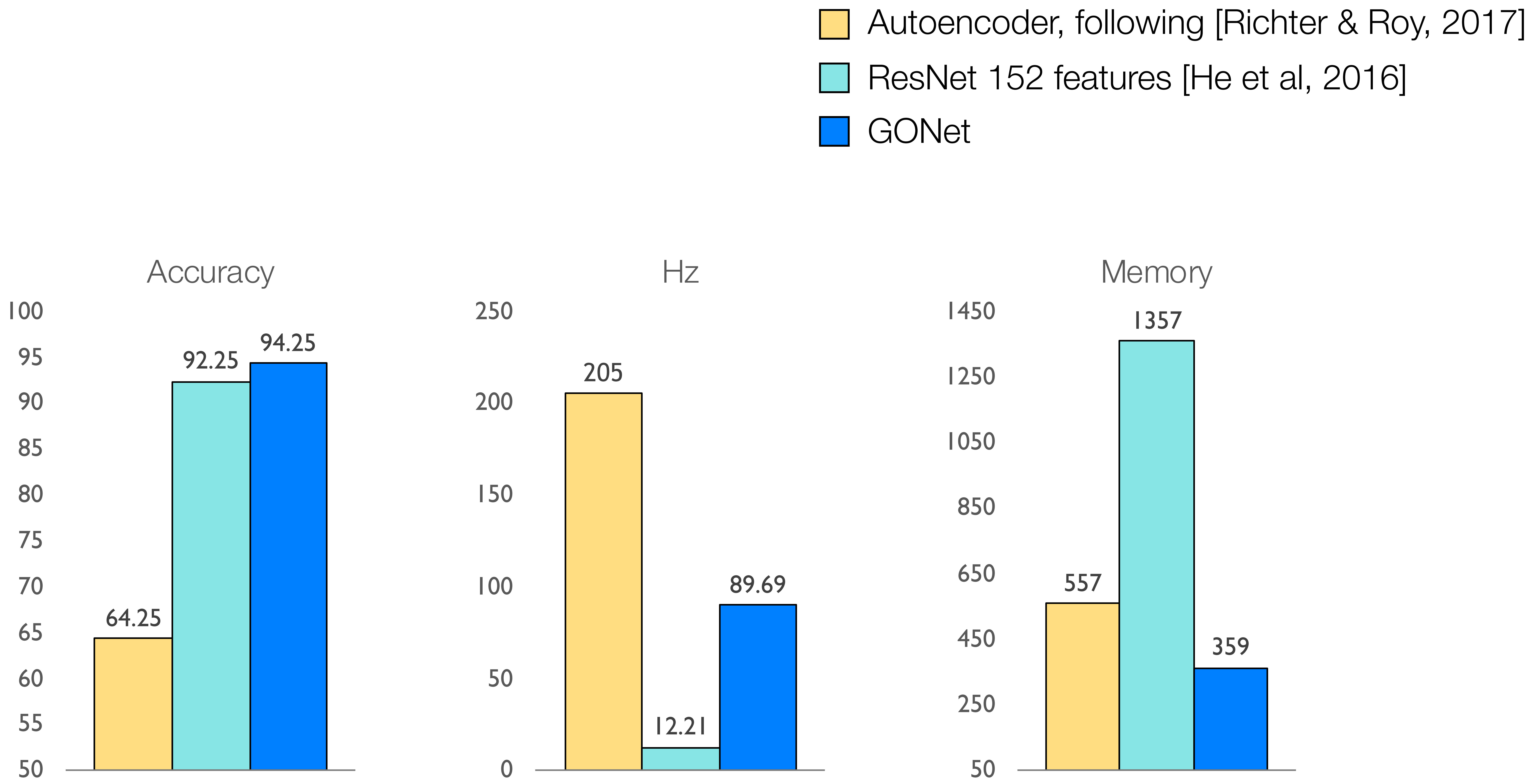


- 27 buildings
  - 19 for training
  - 4 for validation
  - 4 for testing
- ~177K images
- 2400 labeled images



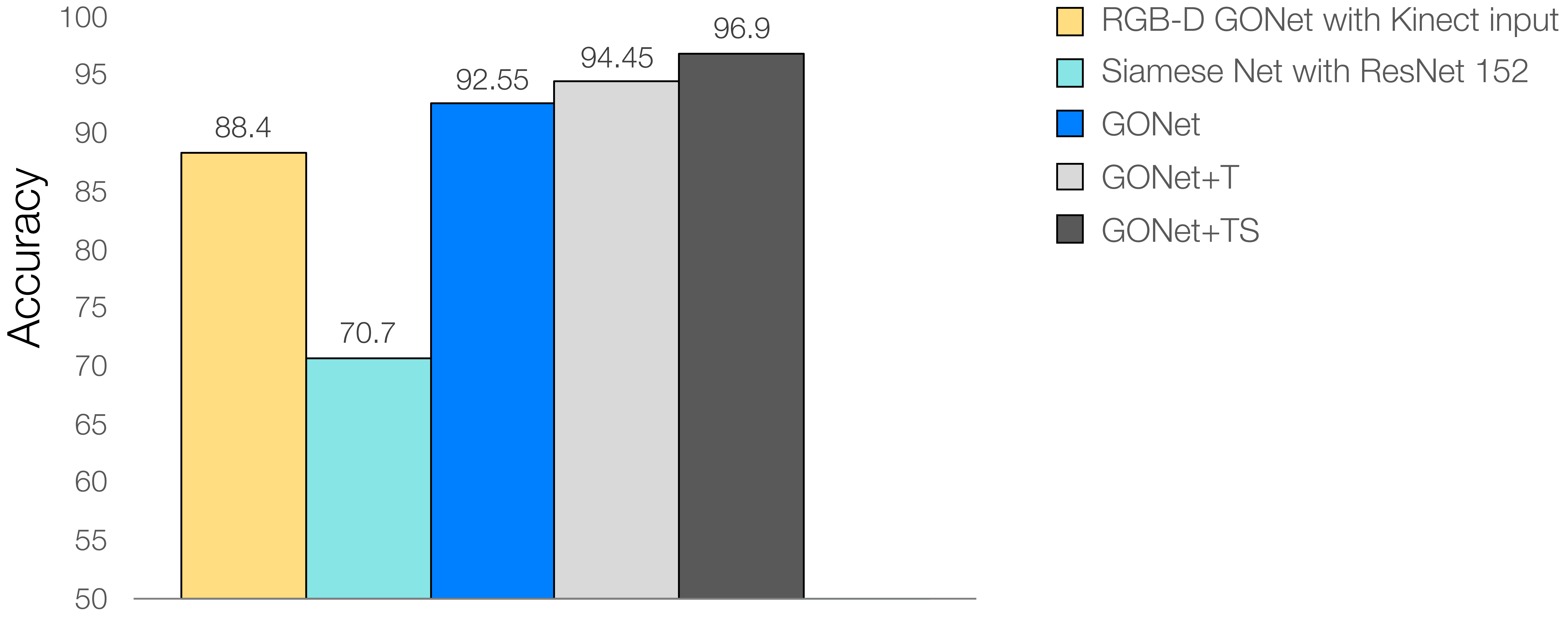
Results

Quantitative Evaluation on GO Stanford 1



[Hirose et al., 2018]

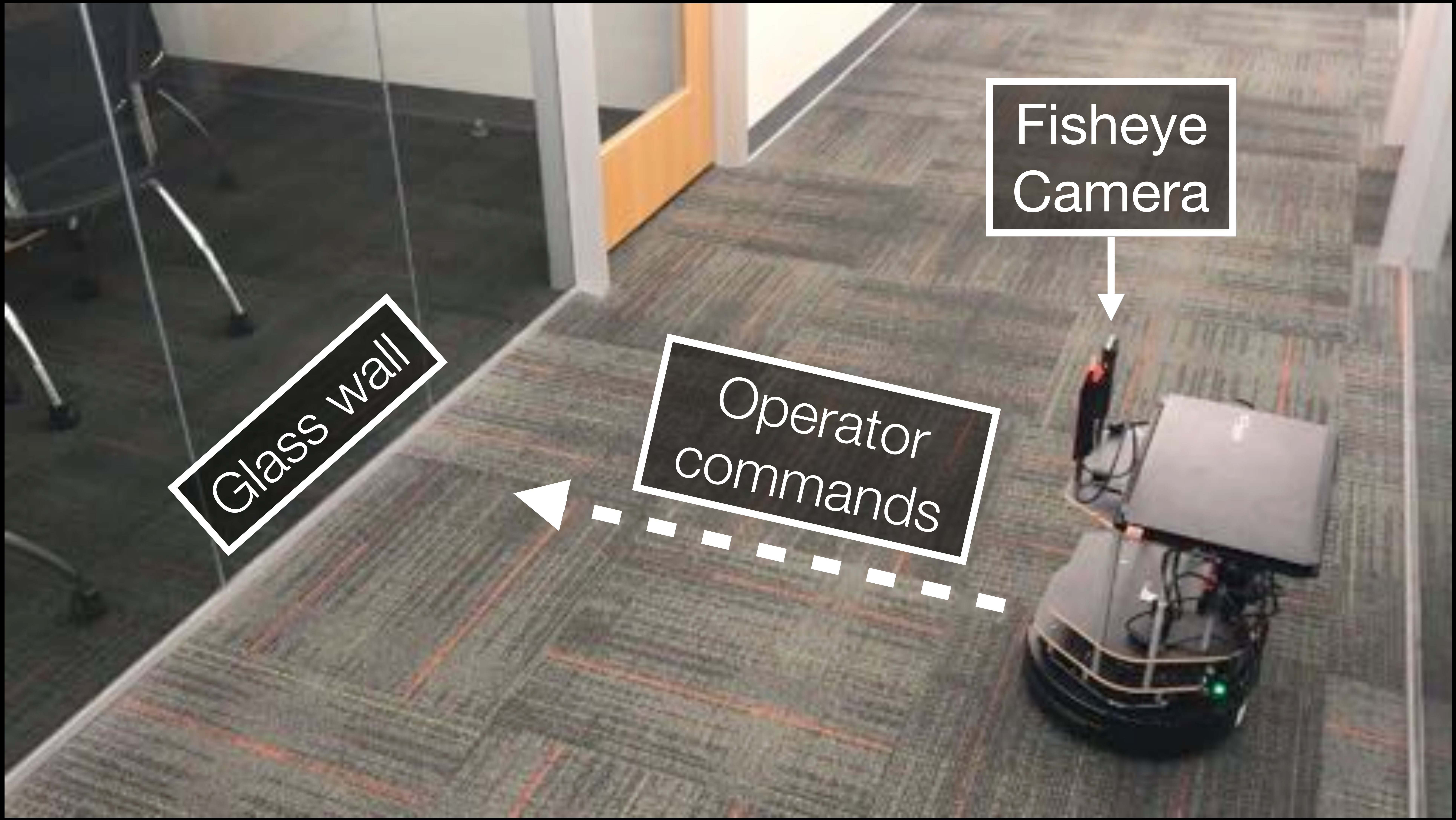
Quantitative Evaluation on GO Stanford 2



[Hirose et al., 2018]

Example Results in Real World

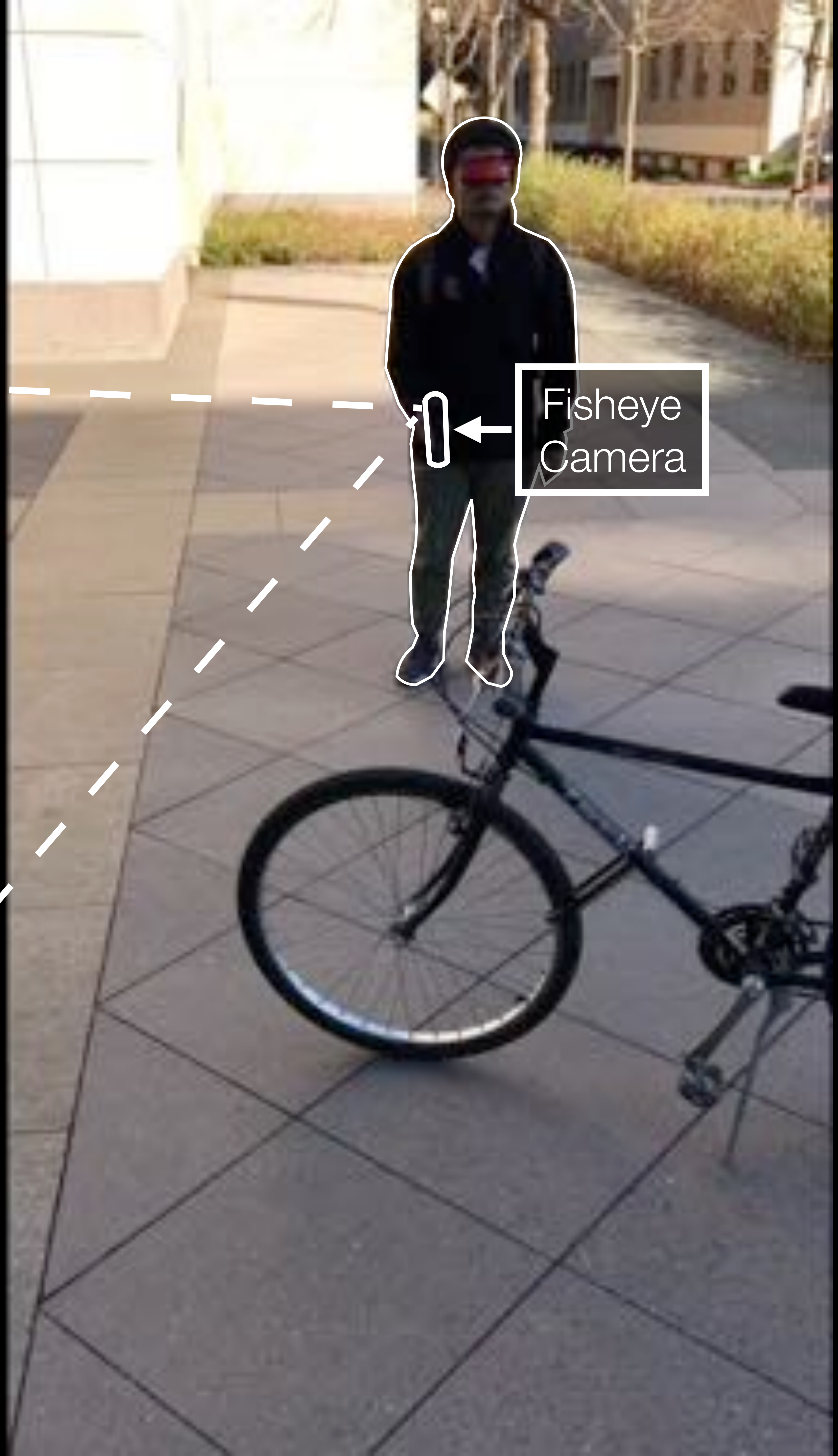
Visual Emergency Stop Switch (GONet+T)



Visual Emergency Stop Switch (GONet+TS)

Failure Case

Traversability Estimation as an Assistive Device



More at http://cvgl.stanford.edu/gonet/

Function Approximation for Mobile Robotics

Social
Navigation*

* In collaboration with Stanford

- Background
- Fundamental technologies for social navigation
  - Traversability Estimation
  - A-B Navigation in Dynamic Environments
- What might come next?

A-B Navigation

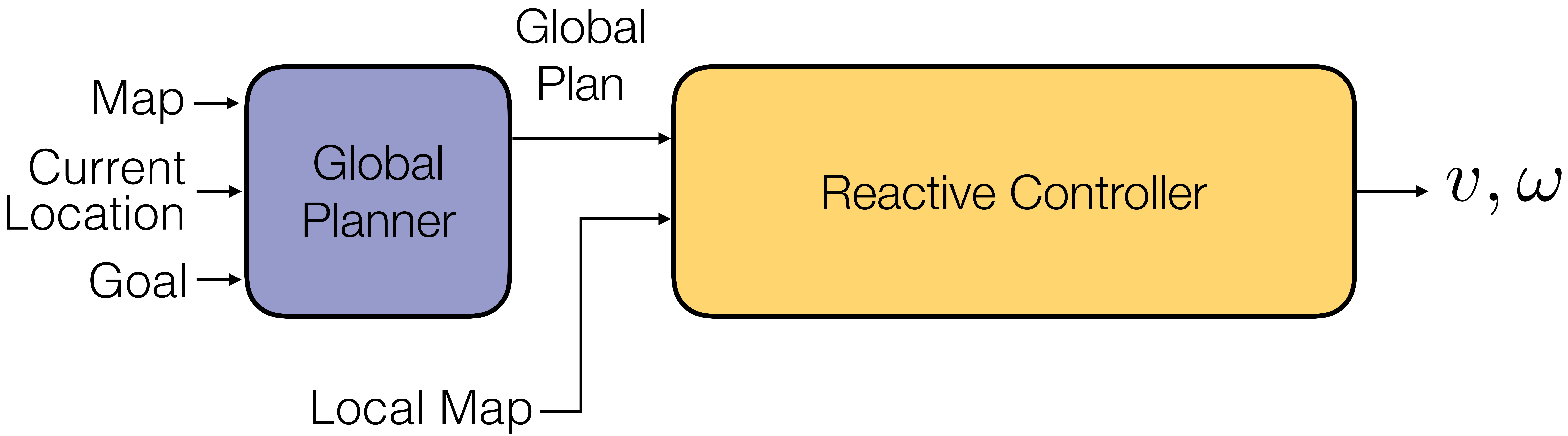
- Enable a robot to navigate to a location B from a starting location A.
- Many different approaches.
- We focus on cases where a map of the environment is given and the robot can localize itself relative to the map.

R.O.B.O.T. Comics

"HIS PATH-PLANNING MAY BE SUB-OPTIMAL, BUT IT'S GOT FLAIR."

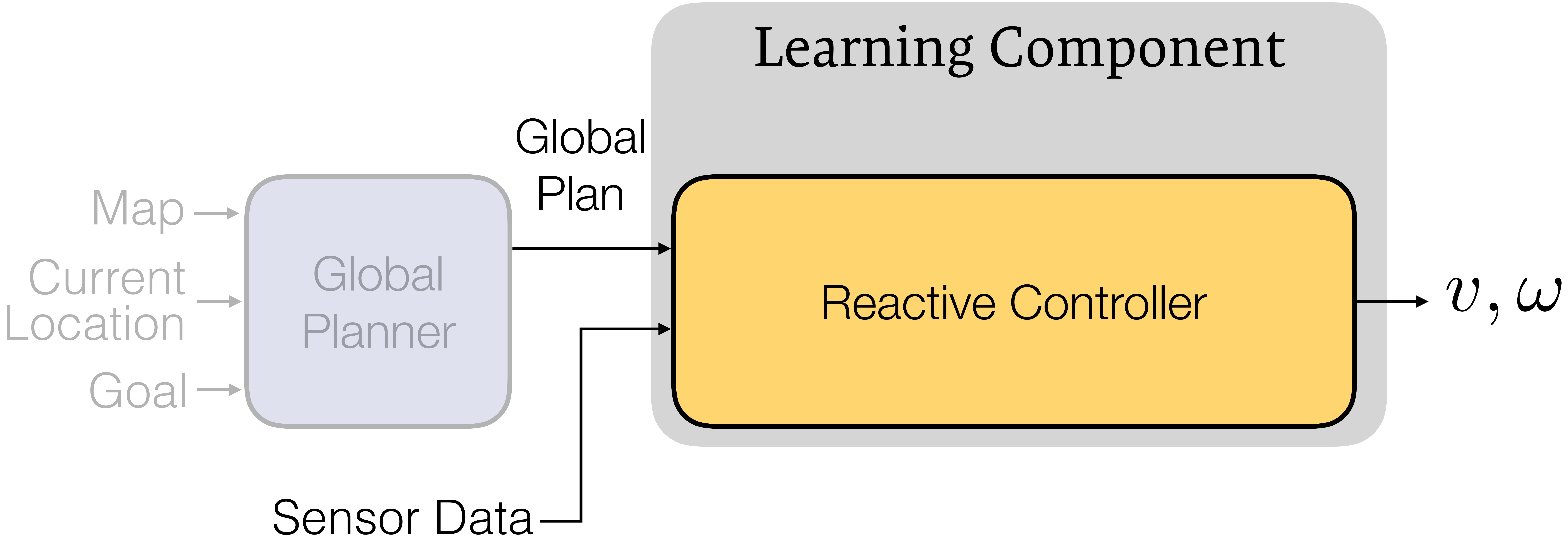
https://www.youtube.com/watch?v=8BtDuzu2Wel

Navigation Systems



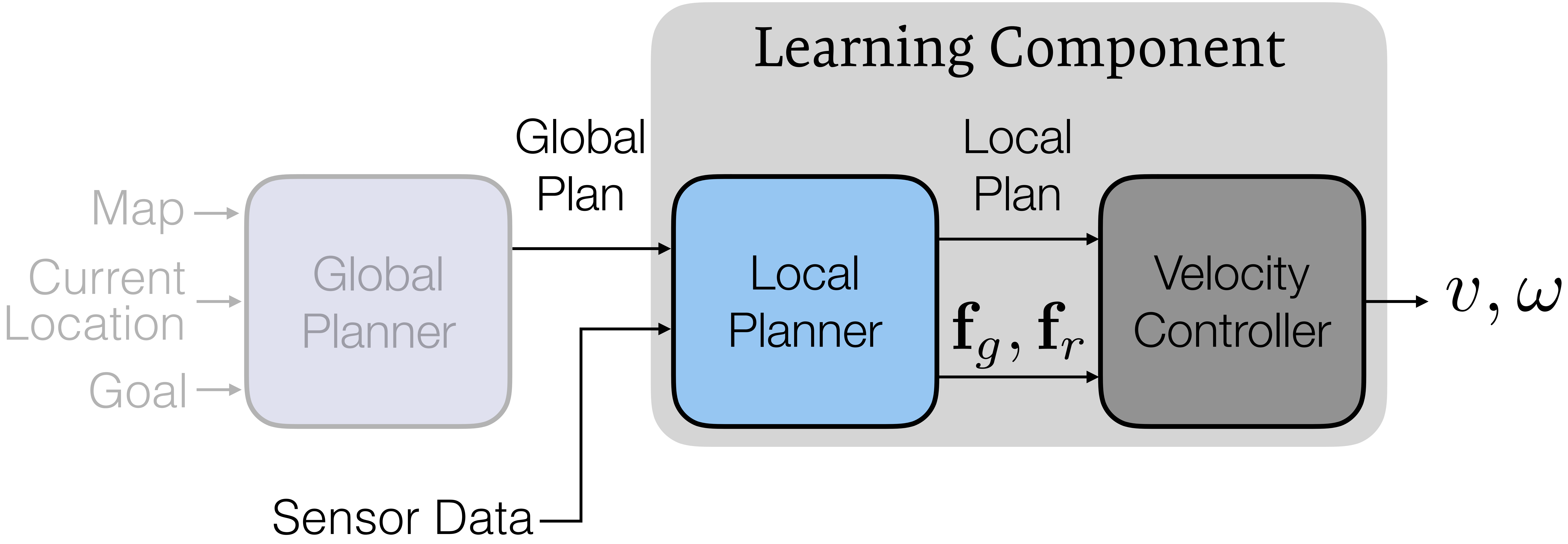
[La Valle, 2006; Fox, Burgard, Thrun, 1997; Gerkey and Konolige, 2008; Lu, Hershberger, Smart, 2014]

Navigation Systems



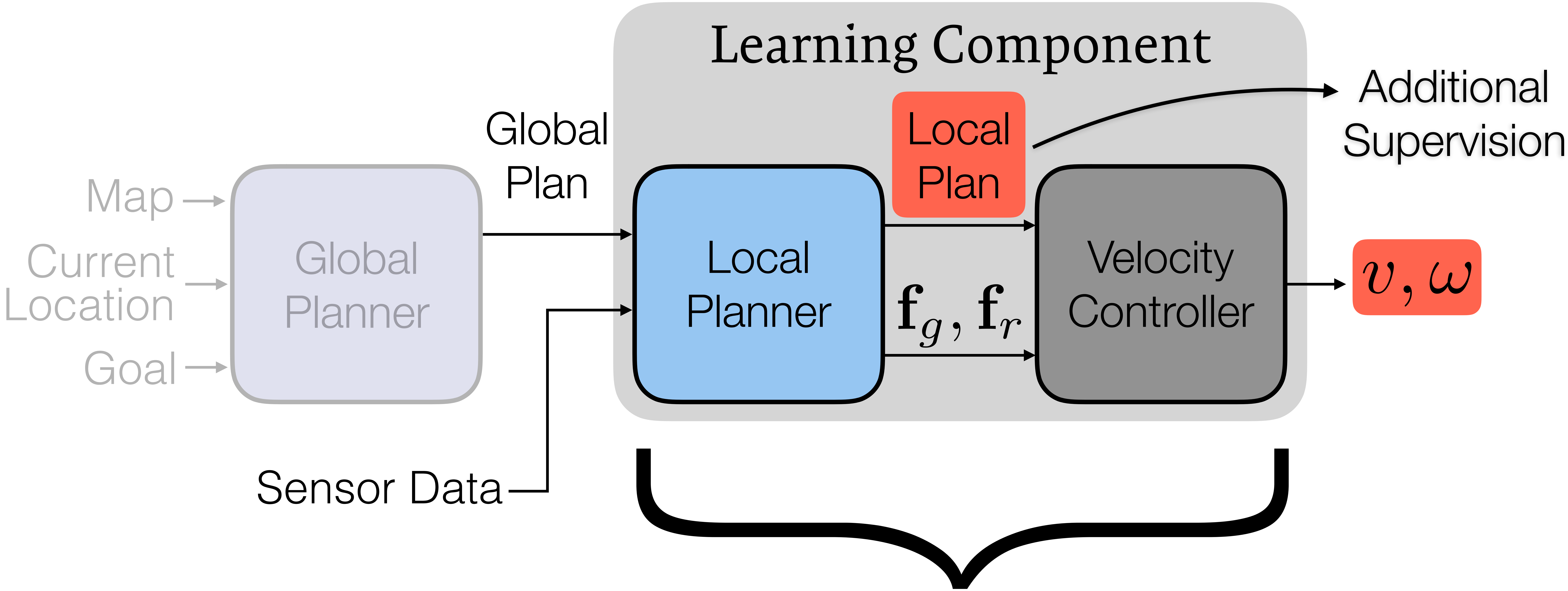
[Gao et al., 2017; Pfeiffer et al., 2017]

Deep Local Re-Planning and Control



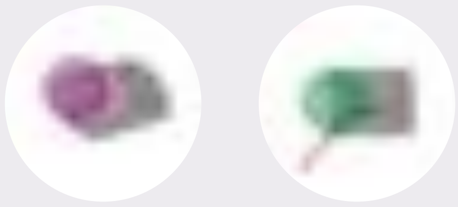
[Pokle et al., 2019]

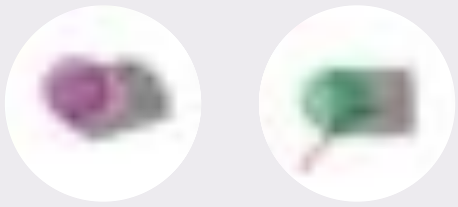
Deep Local Re-Planning and Control



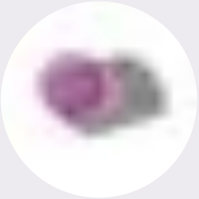
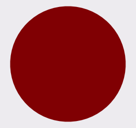
Trained with Imitation Learning given expert navigation examples

[Pokle et al., 2019]

 Pedestrians

 Pedestrians

 Static Obstacles

  Pedestrians  Goal

 Static Obstacles

Pedestrians

Goal

Static Obstacles

Global Plan


Pedestrians
Goal
Robot
Static Obstacles
Global Plan


Pedestrians
Goal
Robot
Static Obstacles
Global Plan
Predicted Local Plan

Pedestrians

Goal

Robot

Static Obstacles

Global Plan

Predicted Local Plan

Transfer to Real World?


2X
Internal World Representation

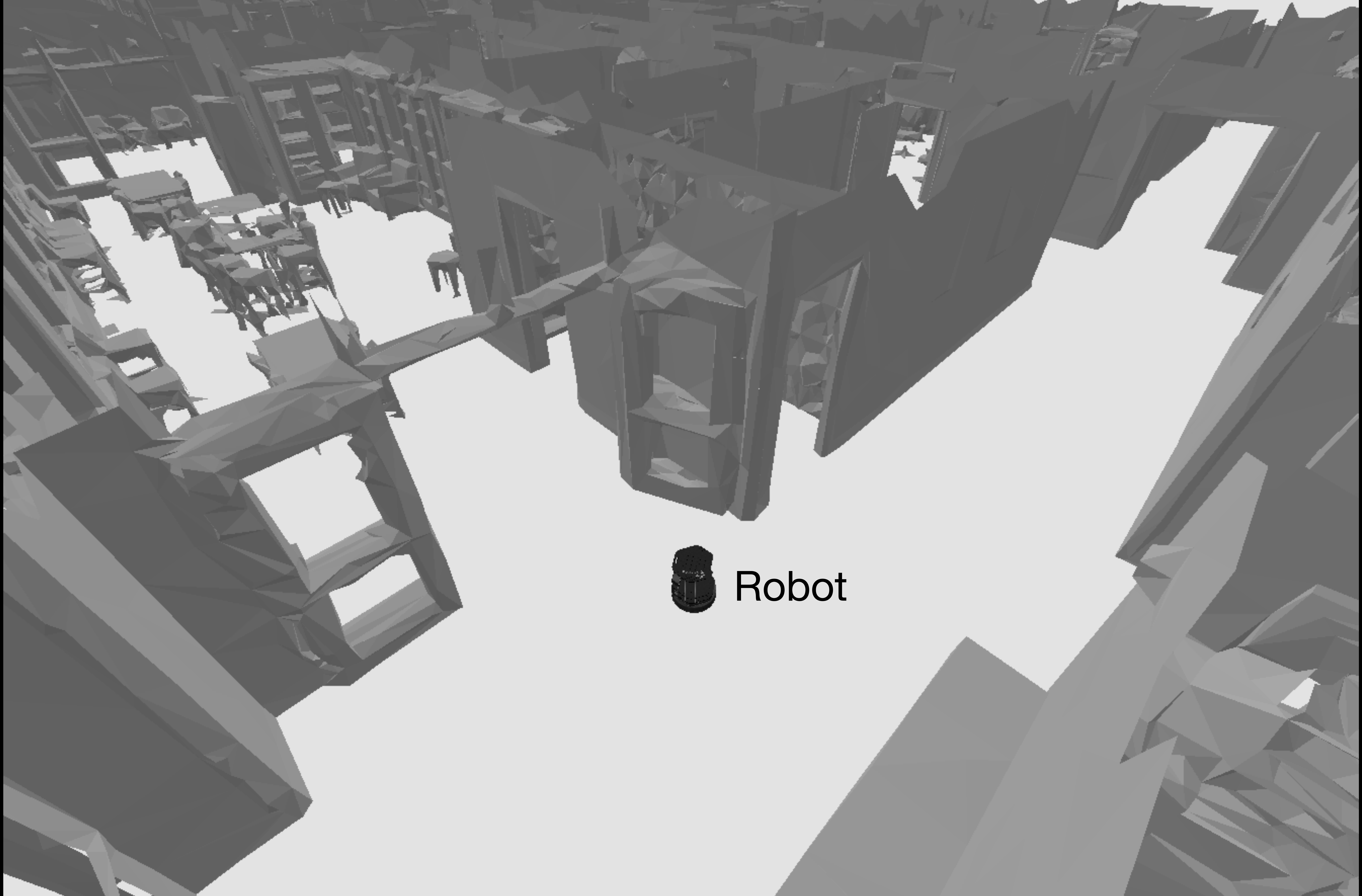
Function Approximation for Mobile Robotics

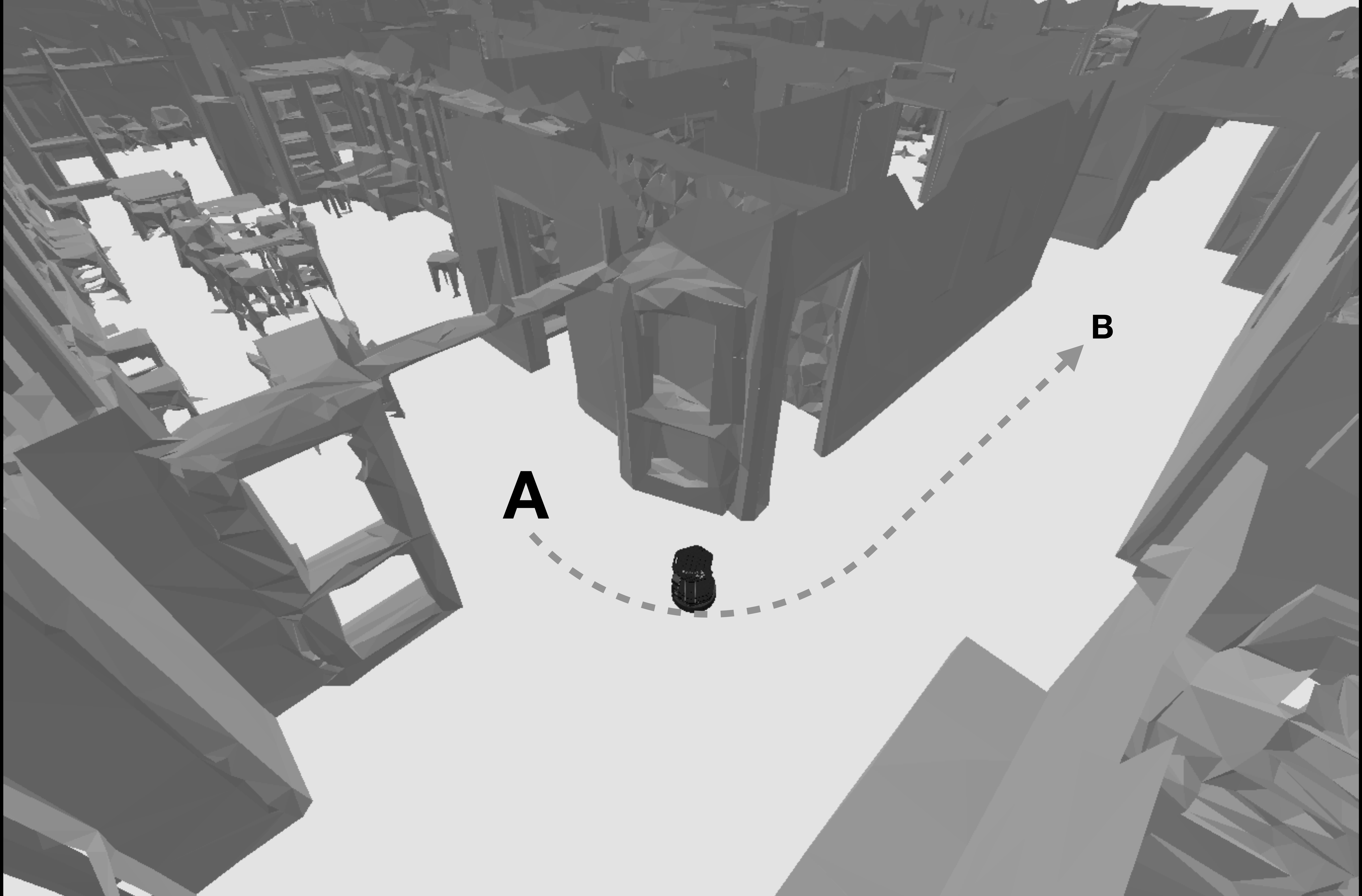
Social
Navigation*

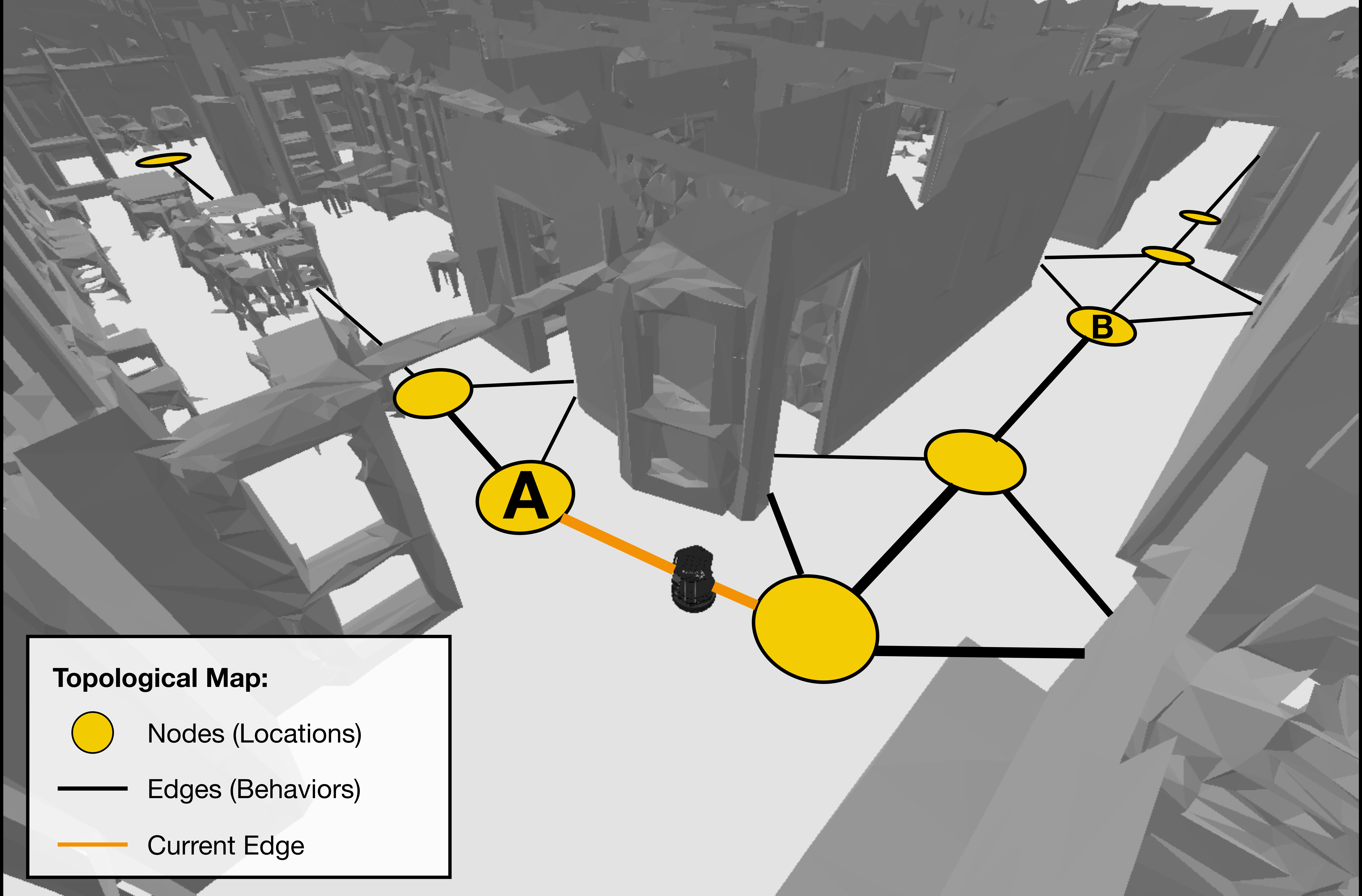
* In collaboration with Stanford

- Background
- Fundamental technologies for social navigation
  - Traversability Estimation
  - A-B Navigation in Dynamic Environments
- What might come next?

2D-3D-S Dataset


Robot


A
B



A
B
Topological Map:
Nodes (Locations)
Edges (Behaviors)
Current Edge

GraphNav

Localization

Behavior Selection

Behavior Execution

[Chen et al., 2019]

GraphNav

Localization

Behavior Selection

Behavior Execution

Topological Map

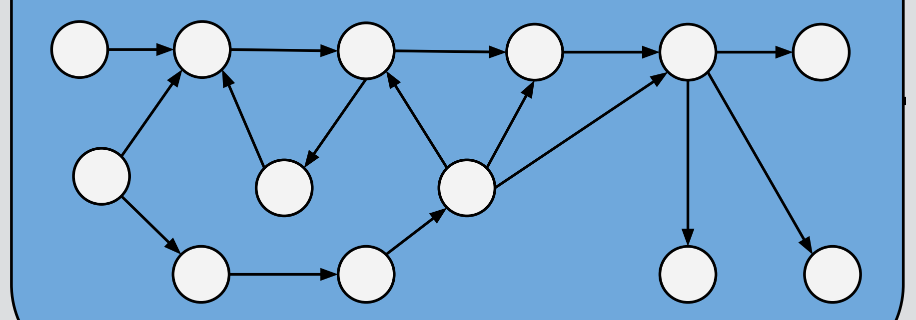
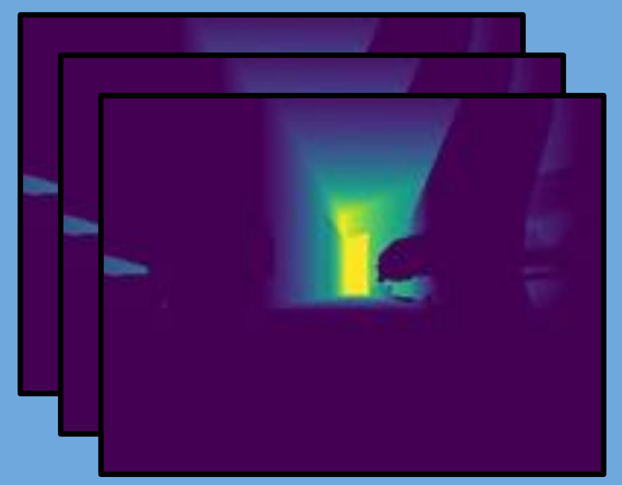


Image Stack



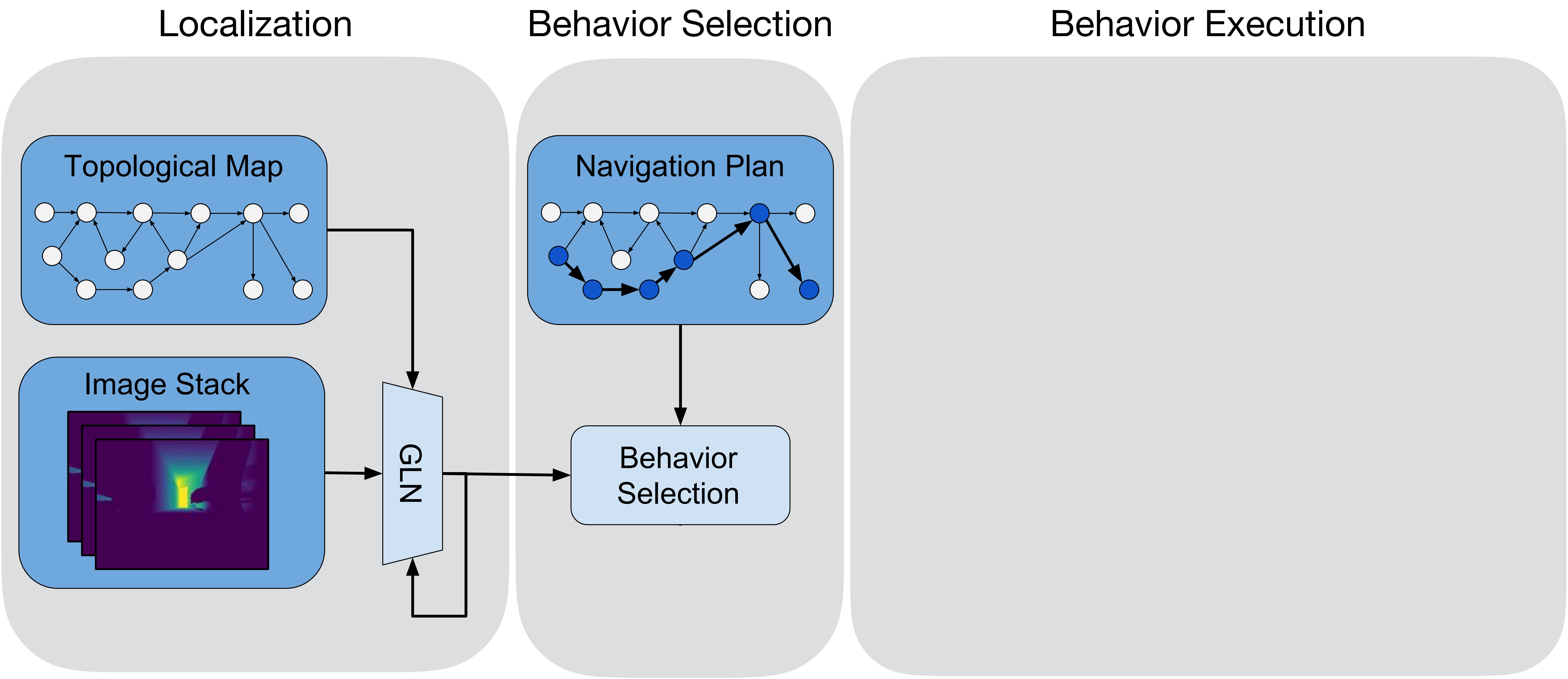
[Chen et al., 2019]

GraphNav

Localization | Behavior Selection | Behavior Execution

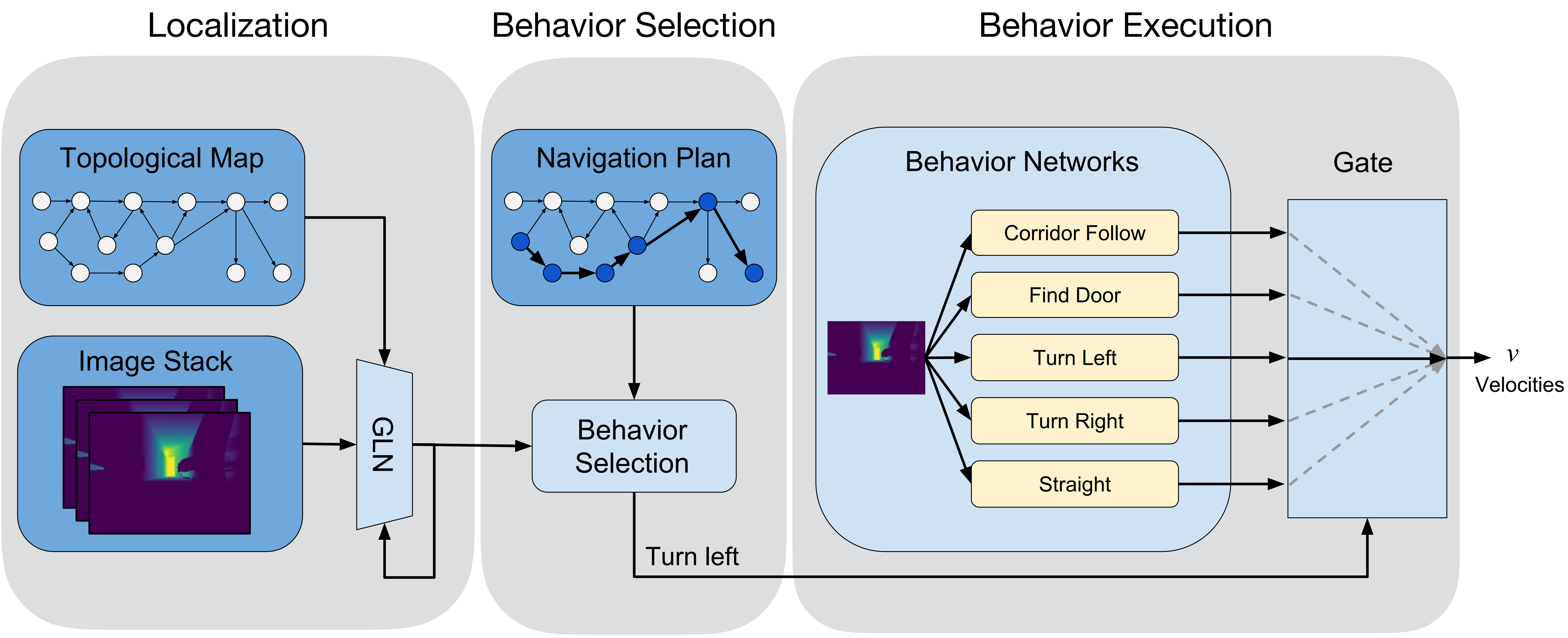
[Chen et al., 2019]

GraphNav



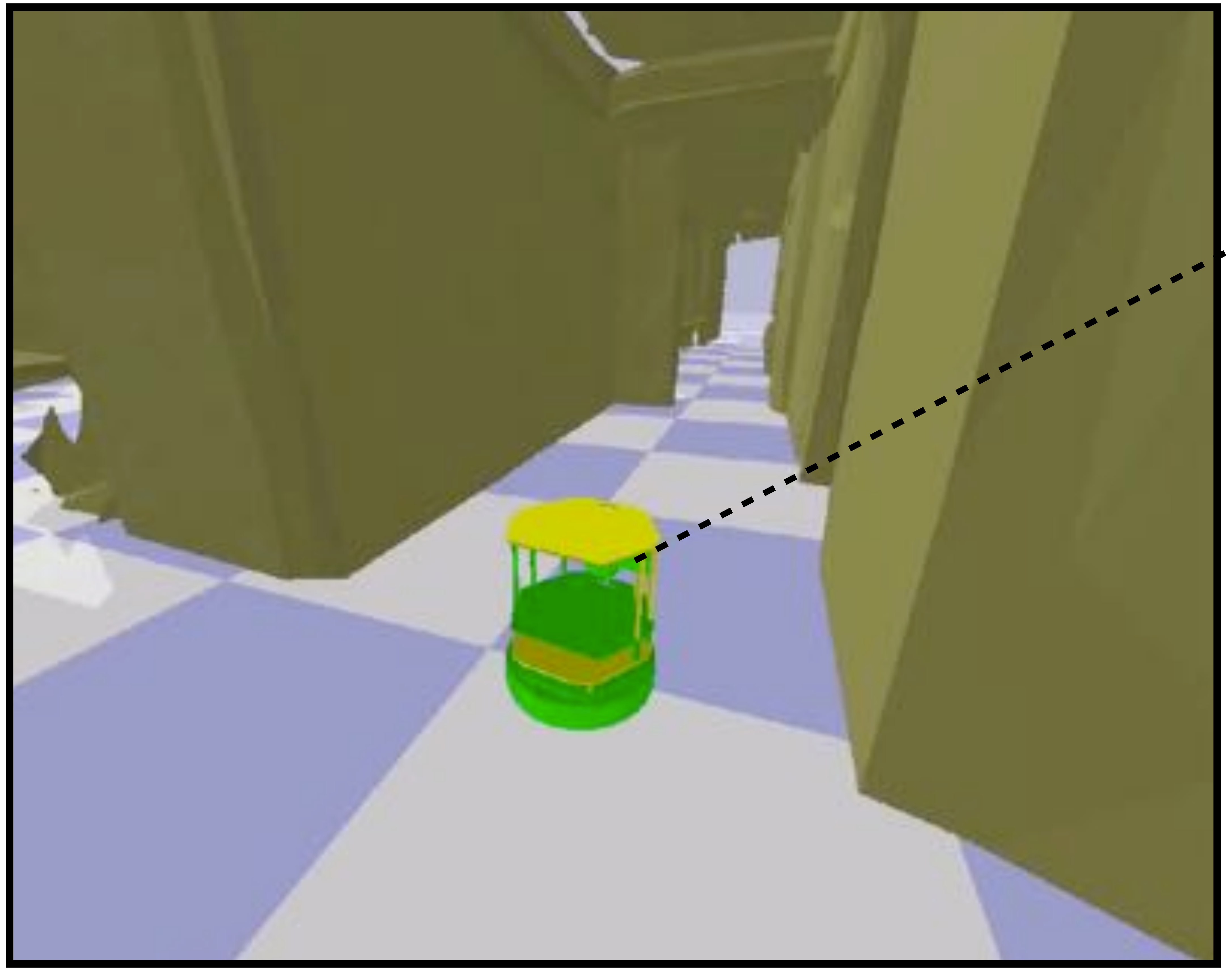
[Chen et al., 2019]

GraphNav



[Chen et al., 2019]

Corridor follow behavior



Robot in the Gibson physics-based simulator.

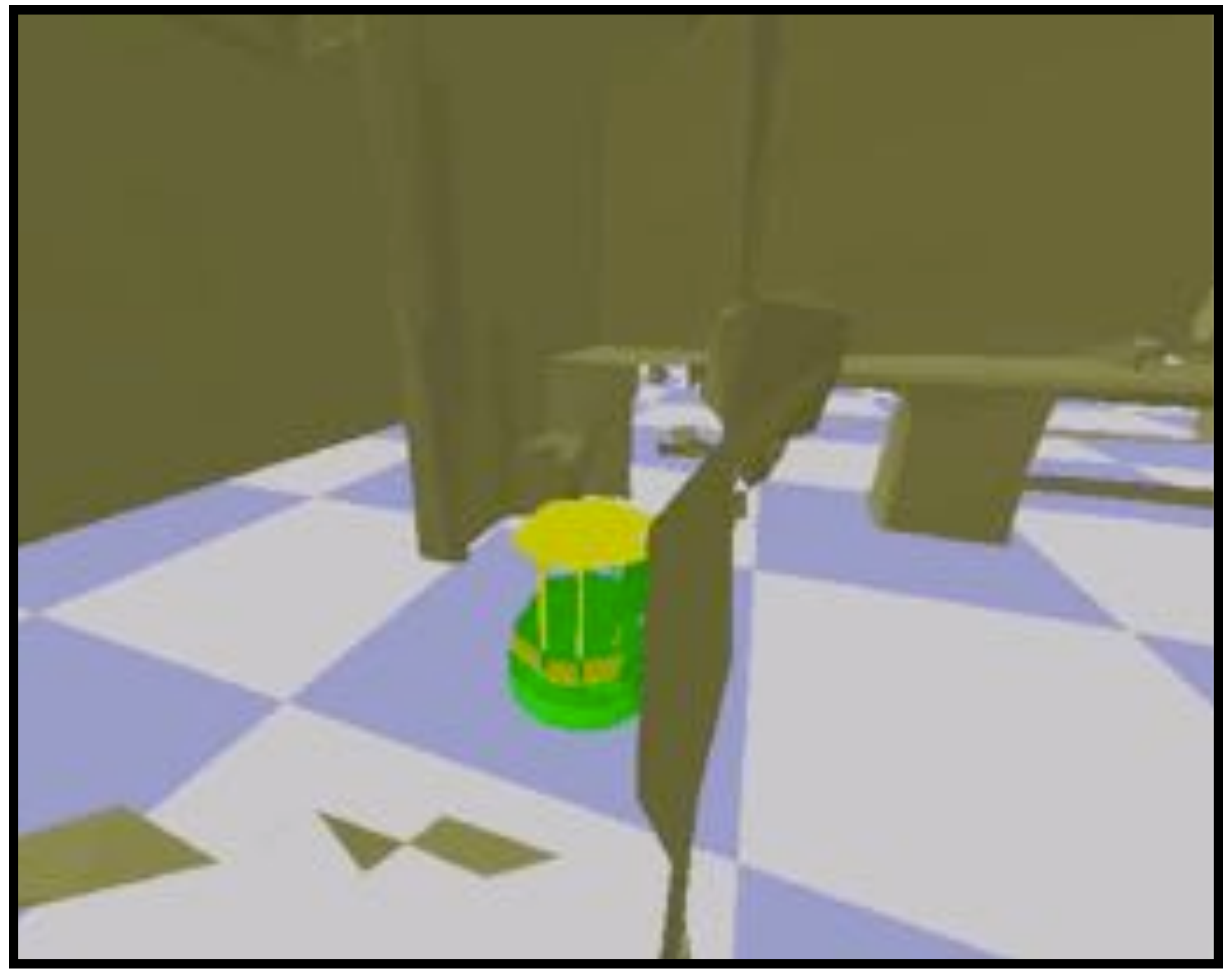
Robot Observations

Depth

We only use depth in this work,
but contribute a dataset with all 3 types of observations.

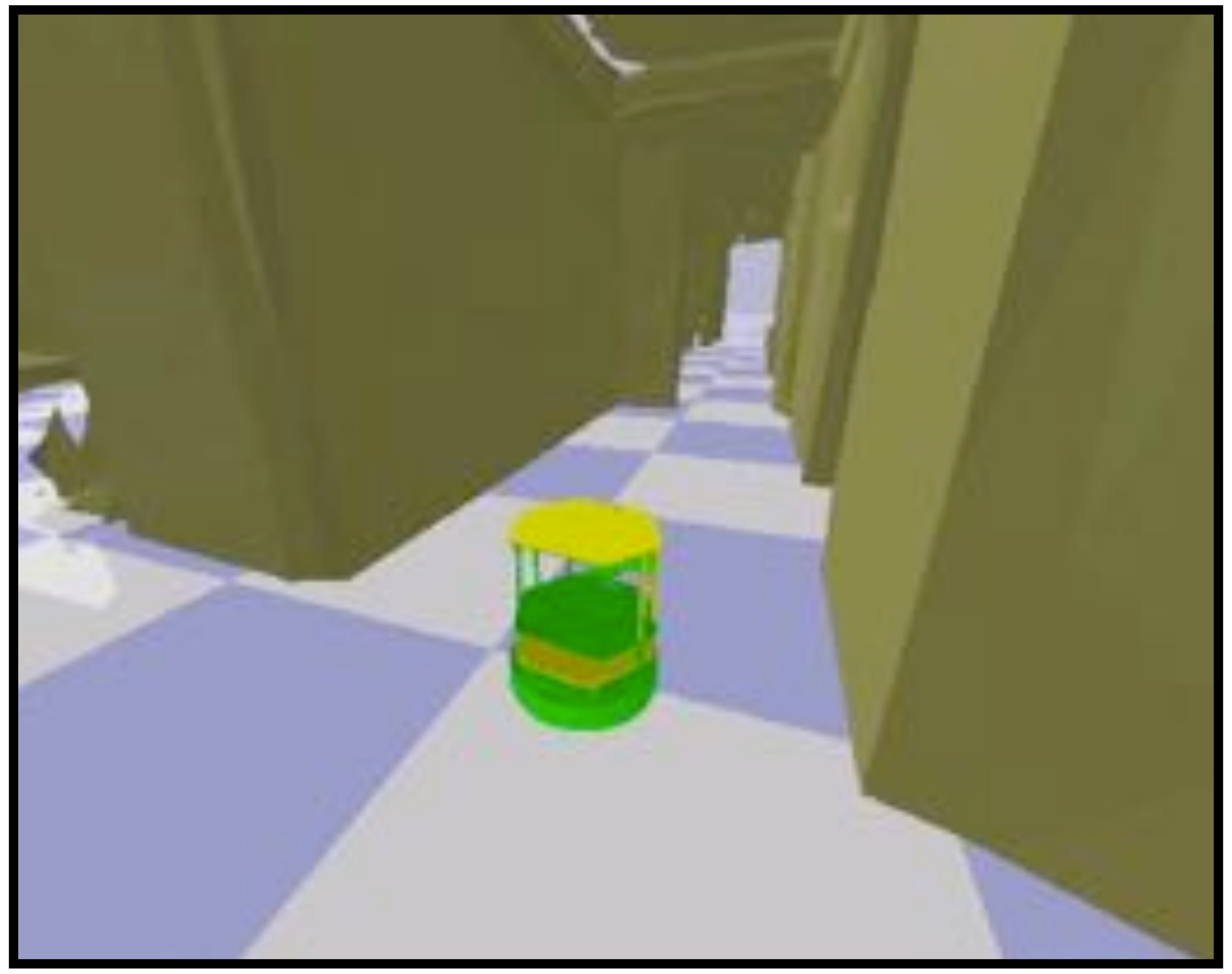
5x speed

turn left behavior

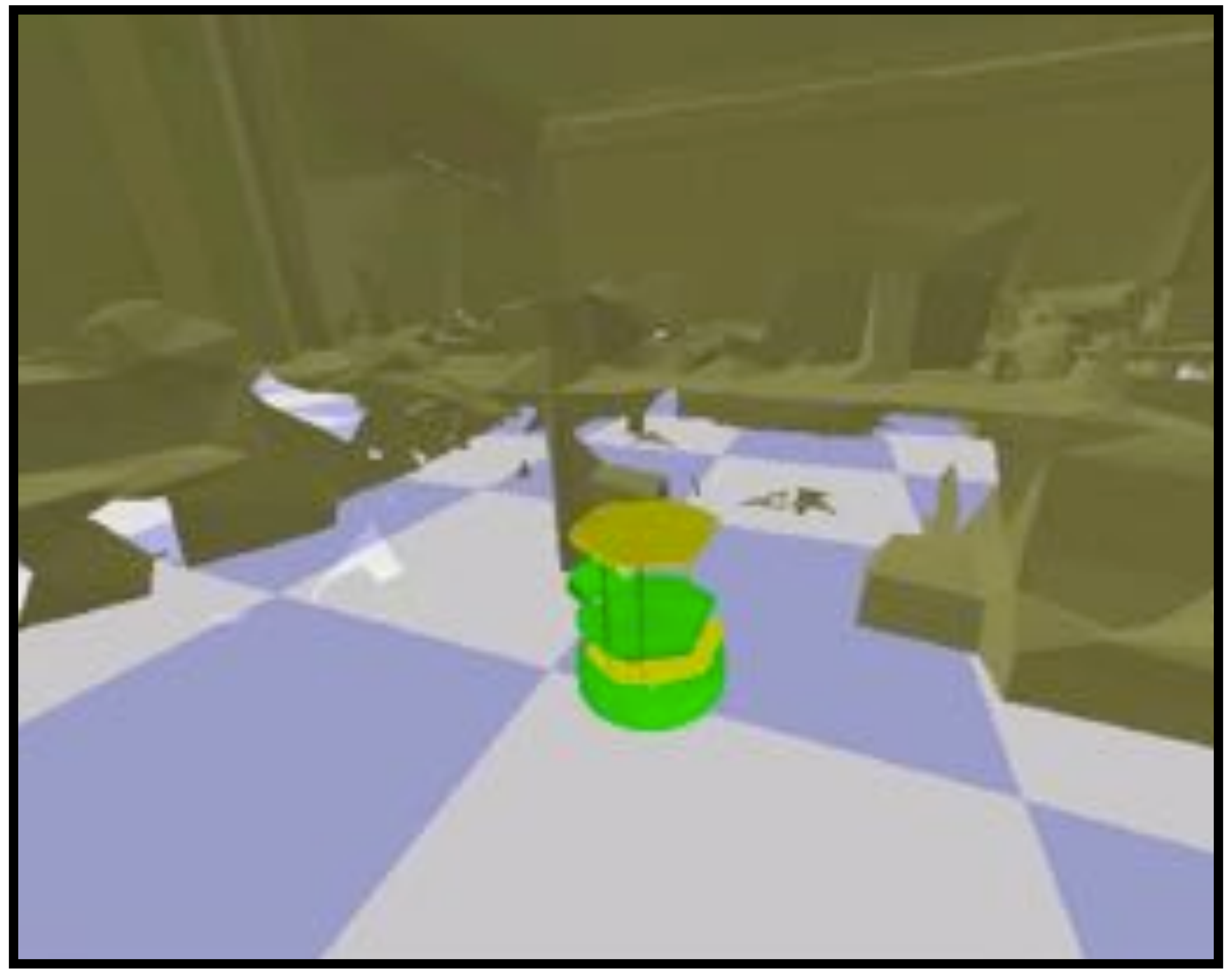


turn right behavior

corridor follow behavior



find door behavior

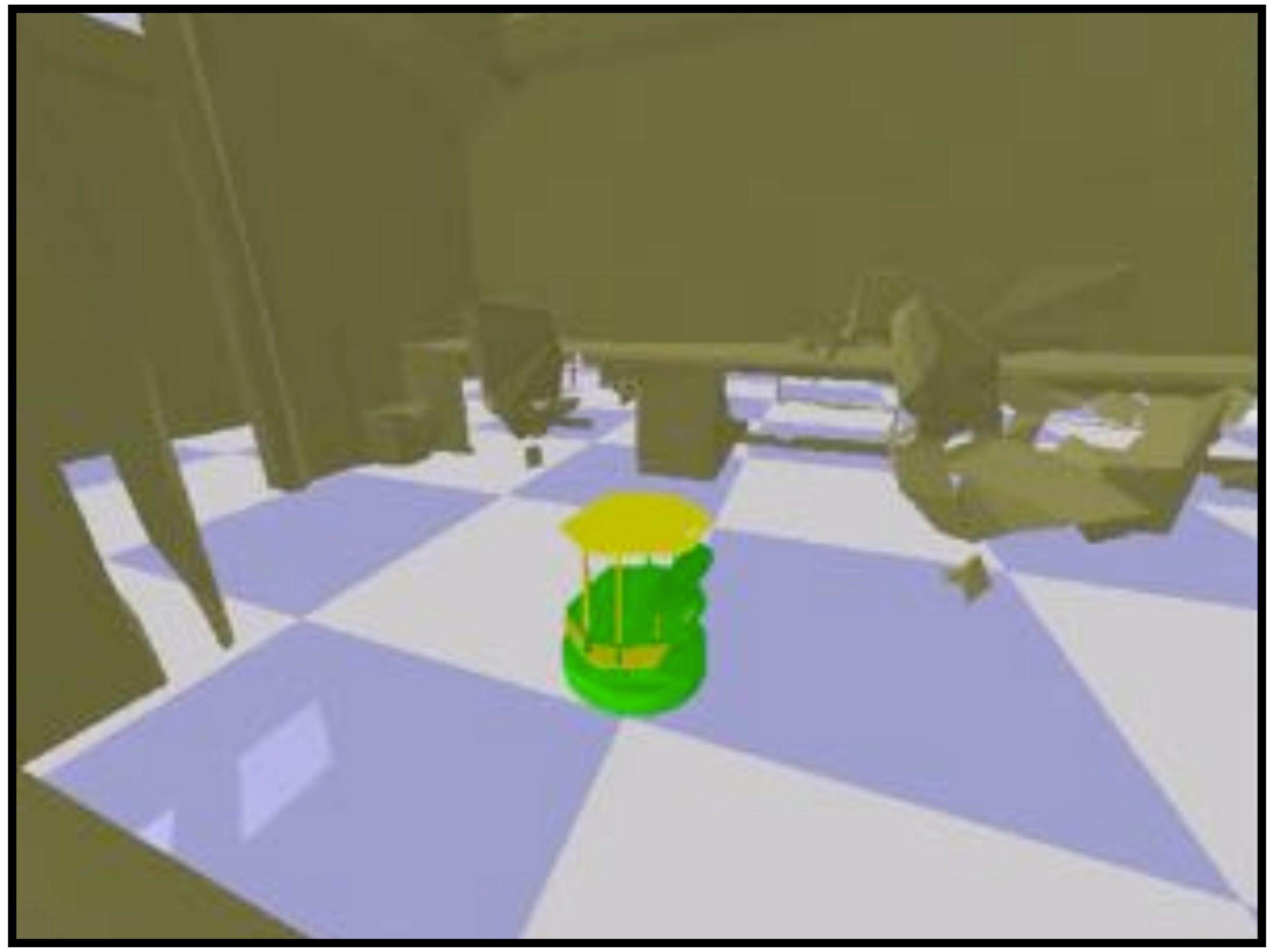


straight behavior

3x speed

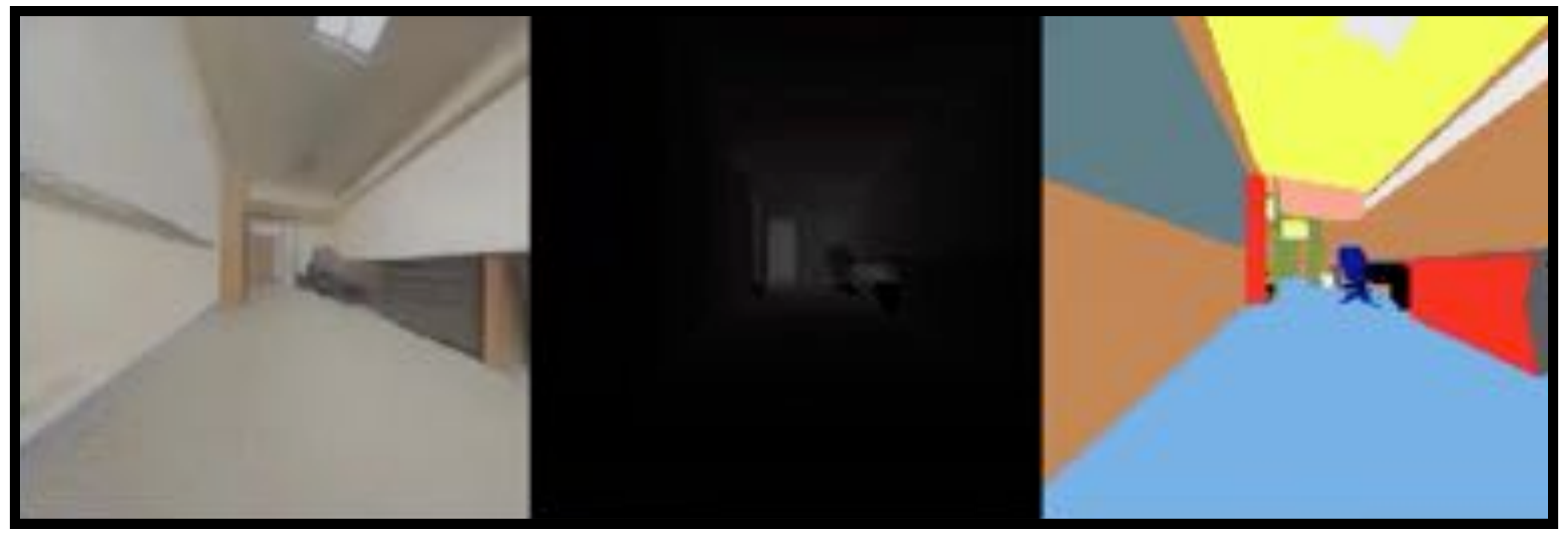
Navigation Example 1

Robot *in Gibson*



Observations

(we only use depth)



– Robot

– Destination

5x speed

More at https://graphnav.stanford.edu/

Function Approximation in Mobile Robotics

Marynel Vázquez
Assistant Professor, Yale Computer Science
http://www.marynel.net
marynel.vazquez@yale.edu

Yale
INTERACTIVE MACHINES GROUP