Carnegie Mellon University

Active Neural Localization

Devendra Singh Chaplot
with Emilio Parisotto, Ruslan Salakhutdinov

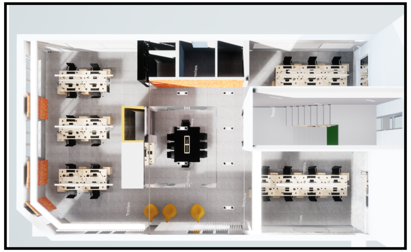
Localization

Estimating the location of an autonomous agent given:

Localization

Estimating the location of an autonomous agent given:

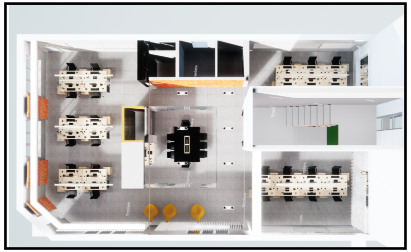
- a map of the environment



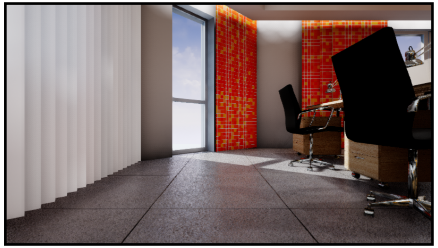
Localization

Estimating the location of an autonomous agent given:

- a map of the environment



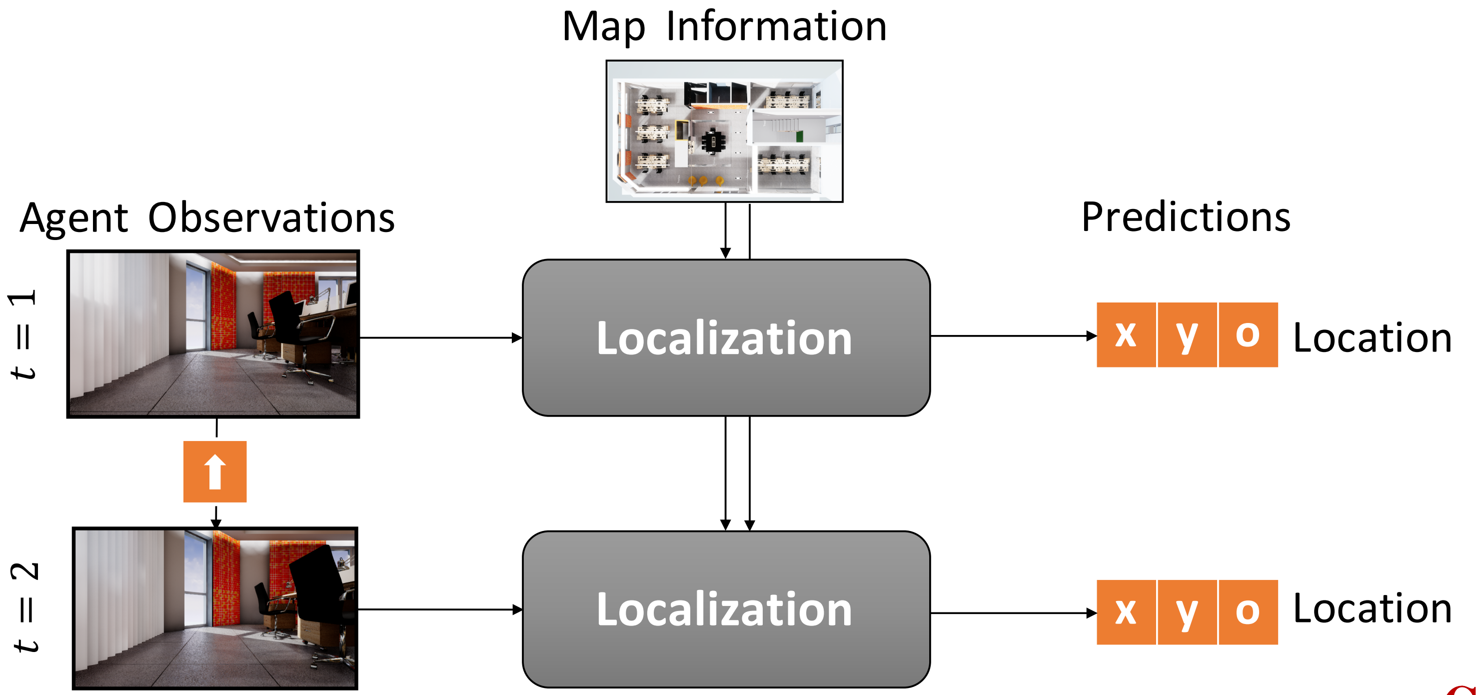
- Agent observations



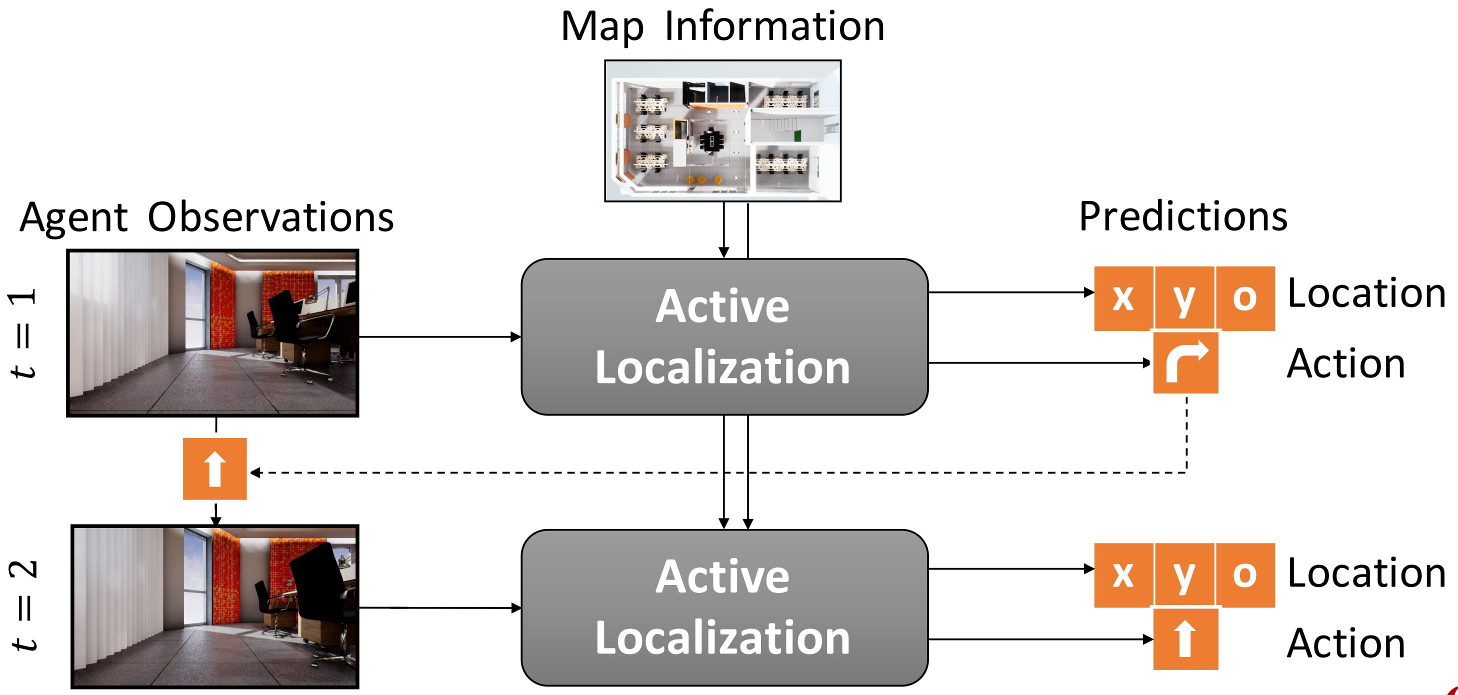
Motivation

- Localization is considered as the **basic precondition for truly autonomous agents** by Burgard et al. (1998)
- Downstream tasks: exploration, target-navigation, planning
- Applications: autonomous vehicles, factory robots, housekeeping robots, delivery drones

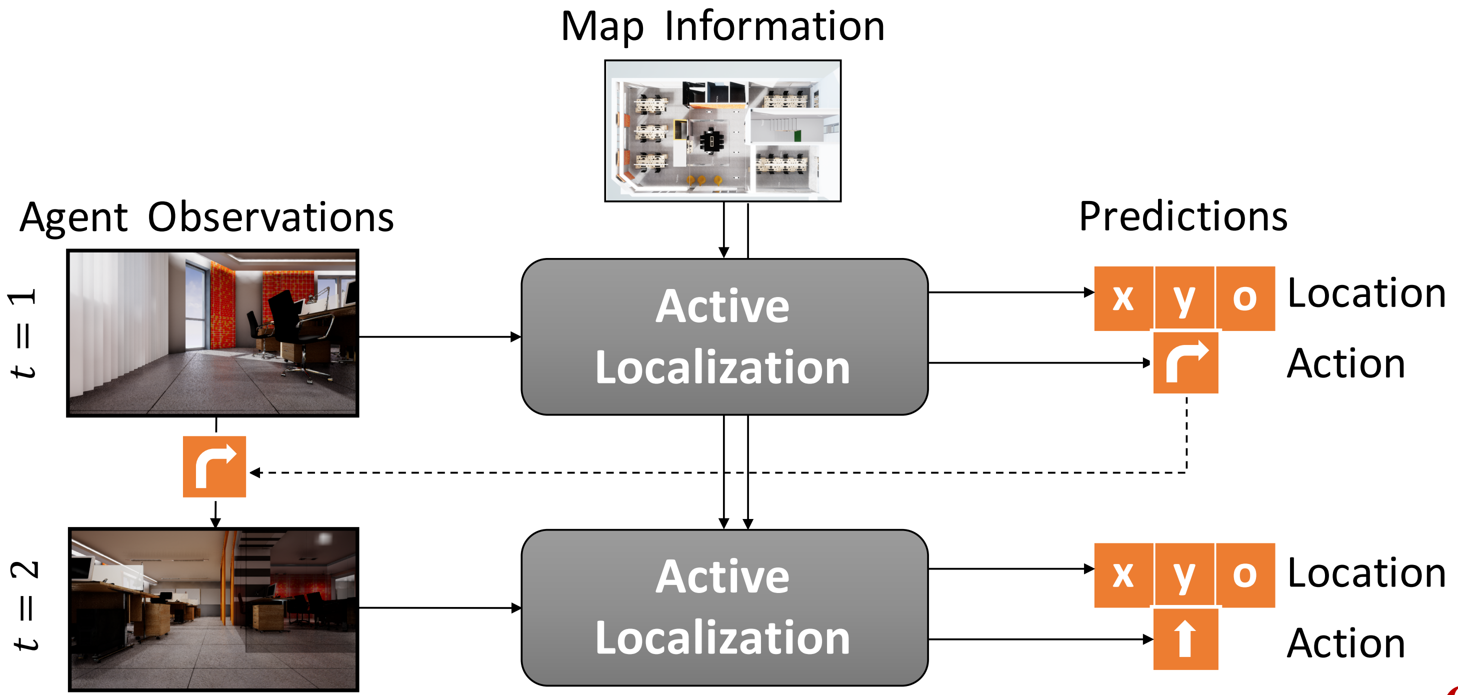
Passive Localization



Active Localization



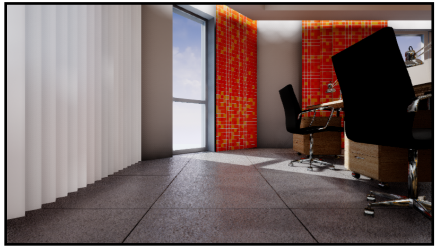
Active Localization



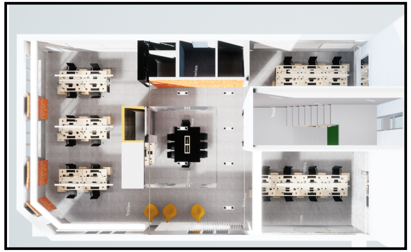
Related Work

- Local Localization:
  - Kalman Filters (Smith et al., 1990)
  - Geometry-based visual odometry methods (Nister et al., 2006)
  - DeepVO (Wang et al., 2017), VINet (Clark et al., 2017)
- Global Localization:
  - Markov Localization (Fox, 1998)
  - Multi-hypothesis Kalman filters (Cox & Leonard, 1994; Roumeliotis & Bekey, 2000)
  - Monte Carlo Localization (Thrun et al., 2001)
  - Active Markov Localization (Fox et al., 1998)
- Learning policy:
  - Navigation: (Mirowski et al. 2017)
  - Planning: Value Iteration Networks (Tamar et al., 2016)
  - Planning under uncertainty: QMDP-Net (Karkus et al., 2017)
  - Mapping and Planning: Cognitive Mapper and Planner (Gupta et al.,2017)
- End-to-end Localization on known maps:
  - PoseNet (Kendall et al., 2015), VidLoc (Clark et al., 2017)

Problem Formulation



s_t: Agent observation at time t



M: Information about the map

a_t: Action taken by the agent at time t

x	y	o

y_t: Position of the agent at time t

Problem Formulation

s_t: Agent observation at time t

a_t: Action taken by the agent at time t

y_t: Position of the agent at time t

M: Information about the map

$P(y_t|s_{1:t}, a_{1:t-1}, M)$: Belief

$\pi(a_t|s_{1:t}, a_{1:t-1}, M)$: Policy

Bayesian Filtering

s_t: Agent observation at time t
a_t: Action taken by the agent at time t
y_t: Position of the agent at time t
M: Information about the map

(Fox et al., 2003)

Bayesian Filtering

s_t: Agent observation at time t
a_t: Action taken by the agent at time t
y_t: Position of the agent at time t
M: Information about the map

Belief: Probability distribution over y_t conditioned over past observations $s_{1:t}$ and actions $a_{1:t-1}$:

$$Bel(y_t) = P(y_t | s_{1:t}, a_{1:t-1}, M)$$

Likelihood: Probability of observing s_t given that the location of the agent is y_t:

$$Lik(s_t) = P(s_t | y_t)$$

(Fox et al., 2003)

Bayesian Filtering

s_t: Agent observation at time t
a_t: Action taken by the agent at time t
y_t: Position of the agent at time t
M: Information about the map

Belief: Probability distribution over y_t conditioned over past observations $s_{1:t}$ and actions $a_{1:t-1}$:

$$Bel(y_t) = P(y_t | s_{1:t}, a_{1:t-1}, M)$$

Likelihood: Probability of observing s_t given that the location of the agent is y_t:

$$Lik(s_t) = P(s_t | y_t)$$

Under the Markov assumption:

$$\underbrace{\overline{Bel}(y_t)}_{\text{Belief before observing } s_t} = \sum_{y_{t-1}} \underbrace{P(y_t | y_{t-1}, a_{t-1})}_{\text{Transition function}} \underbrace{Bel(y_{t-1})}_{\text{Belief after observing } s_{t-1}}$$

$$\underbrace{Bel(y_t)}_{\text{Belief after observing } s_t} = \frac{1}{Z} \underbrace{Lik(s_t)}_{\text{Prob. of observing } s_t} \underbrace{\overline{Bel}(y_t)}_{\text{Belief before observing } s_t}$$

Transition function: Probability of landing in a state y_t from y_{t-1}, based on the action, a_{t-1}:

$$f_T = P(y_t | y_{t-1}, a_{t-1})$$

(Fox et al., 2003)

Bayesian Filtering

s_t: Agent observation at time t
a_t: Action taken by the agent at time t
y_t: Position of the agent at time t
M: Information about the map

Belief: Probability distribution over y_t conditioned over past observations $s_{1:t}$ and actions $a_{1:t-1}$:

$$Bel(y_t) = P(y_t | s_{1:t}, a_{1:t-1}, M)$$

Likelihood: Probability of observing s_t given that the location of the agent is y_t:

$$Lik(s_t) = P(s_t | y_t)$$

Under the Markov assumption:

$$\underbrace{\overline{Bel}(y_t)}_{\text{Belief before observing } s_t} = \sum_{y_{t-1}} \underbrace{P(y_t | y_{t-1}, a_{t-1})}_{\text{Transition function}} \underbrace{Bel(y_{t-1})}_{\text{Belief after observing } s_{t-1}}$$

$$\underbrace{Bel(y_t)}_{\text{Belief after observing } s_t} = \frac{1}{Z} \underbrace{Lik(s_t)}_{\text{Prob. of observing } s_t} \underbrace{\overline{Bel}(y_t)}_{\text{Belief before observing } s_t}$$

Posterior Belief

Prior Belief

(Fox et al., 2003)

Carnegie Mellon University

Bayesian Filtering

s_t: Agent observation at time t
a_t: Action taken by the agent at time t
y_t: Position of the agent at time t
M: Information about the map

Belief: Probability distribution over y_t conditioned over past observations $s_{1:t}$ and actions $a_{1:t-1}$:

$$Bel(y_t) = P(y_t | s_{1:t}, a_{1:t-1}, M)$$

Likelihood: Probability of observing s_t given that the location of the agent is y_t:

$$Lik(s_t) = P(s_t | y_t)$$

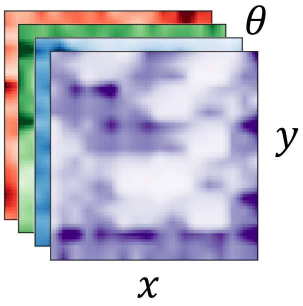
Under the Markov assumption:

$$\underbrace{\overline{Bel}(y_t)}_{\text{Belief before observing } s_t} = \sum_{y_{t-1}} \underbrace{P(y_t | y_{t-1}, a_{t-1})}_{\text{Transition function}} \underbrace{Bel(y_{t-1})}_{\text{Belief after observing } s_{t-1}}$$

$$\underbrace{Bel(y_t)}_{\text{Belief after observing } s_t} = \frac{1}{Z} \underbrace{Lik(s_t)}_{\text{Prob. of observing } s_t} \underbrace{\overline{Bel}(y_t)}_{\text{Belief before observing } s_t}$$

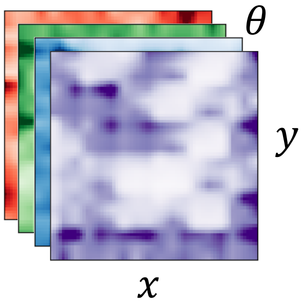
(Fox et al., 2003)

Representation of Belief and Likelihood



$$\underset{\text{Number of orientations}}{O} \times \underbrace{M \times N}_{\text{Map size}}$$

Representation of Belief and Likelihood



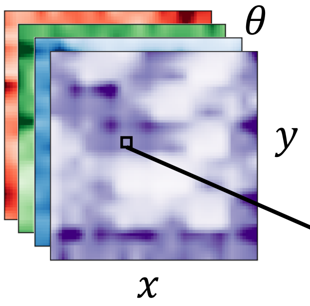
3-dimensional tensor representing
x-coordinate, y-coordinate and orientation

$$O \times \underbrace{M \times N}$$

Map size

Number of orientations

Representation of Belief and Likelihood



3-dimensional tensor representing x-coordinate, y-coordinate and orientation

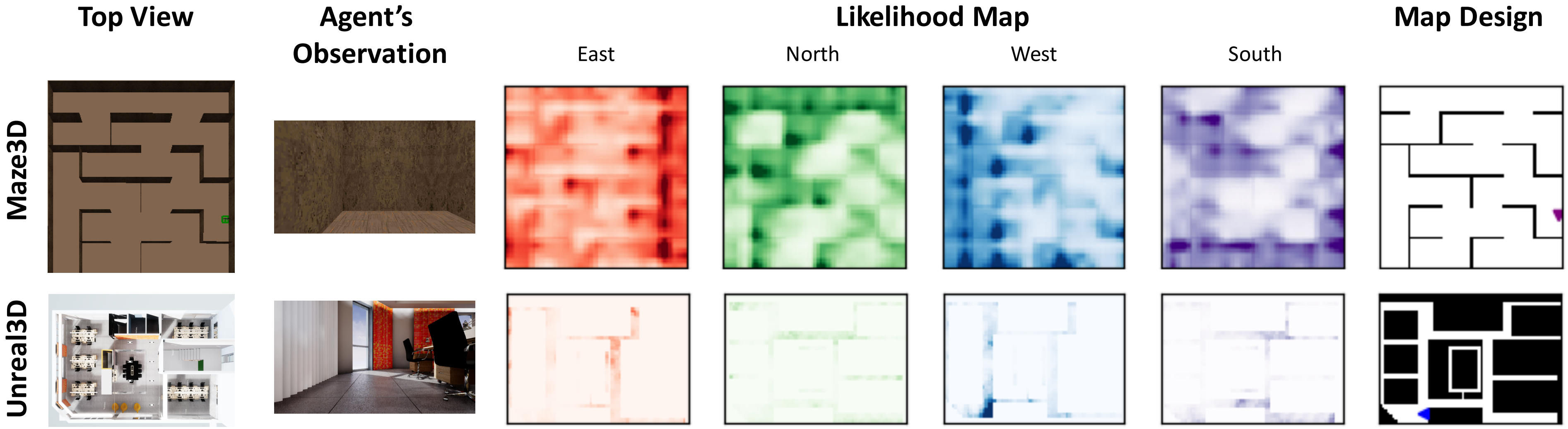
Each element represents the probability of the agent being present in the corresponding location

$O \times \underbrace{M \times N}$

Map size

Number of orientations

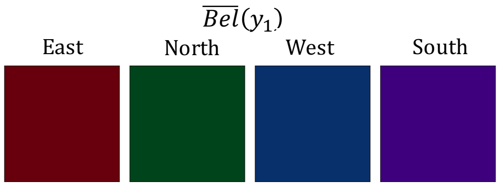
Simulation Environments



Active Neural Localization

Active Neural Localization

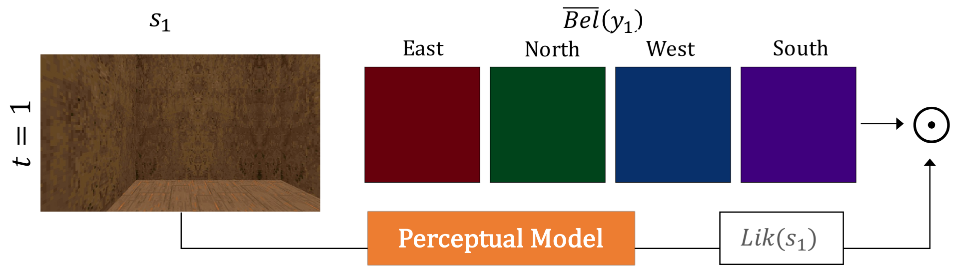
Belief before observing s_t ($\overline{Bel}(y_t)$)



Active Neural Localization

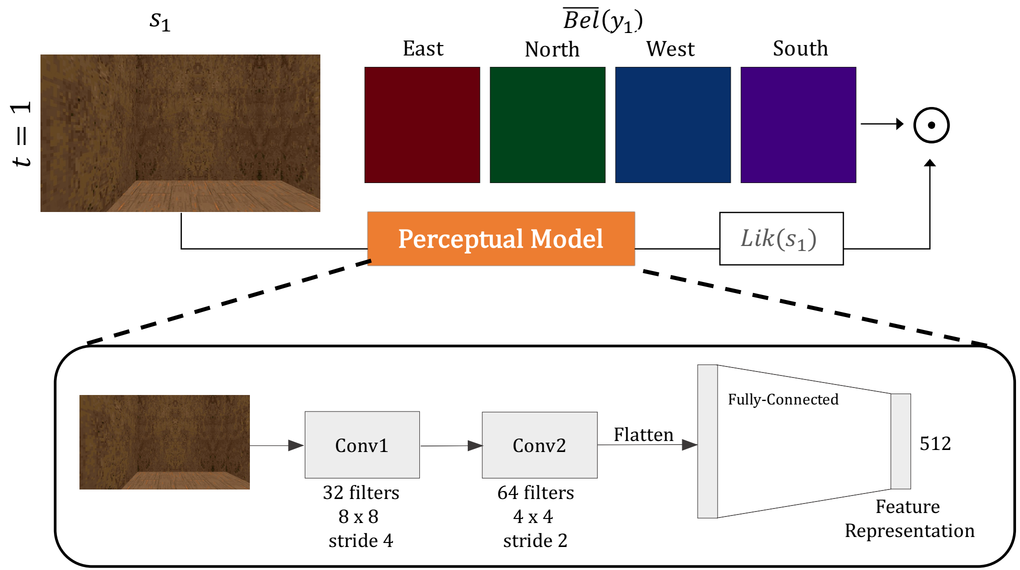
Agent's observation (s_t)

Belief before observing s_t ($\overline{Bel}(y_t)$)



Active Neural Localization

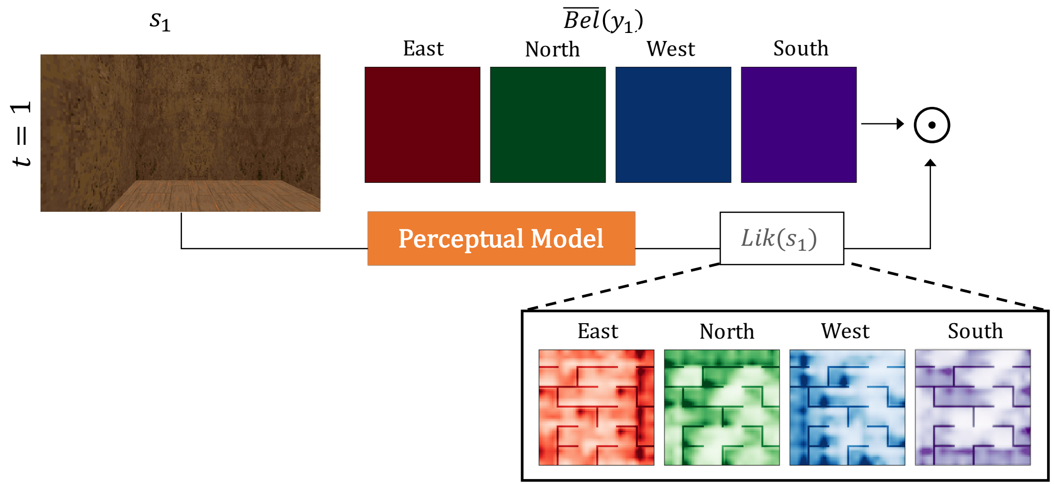
Agent's observation (s_t)   Belief before observing s_t ($\overline{Bel}(y_t)$)



Active Neural Localization

Agent's observation (s_t)

Belief before observing s_t ($\overline{Bel}(y_t)$)

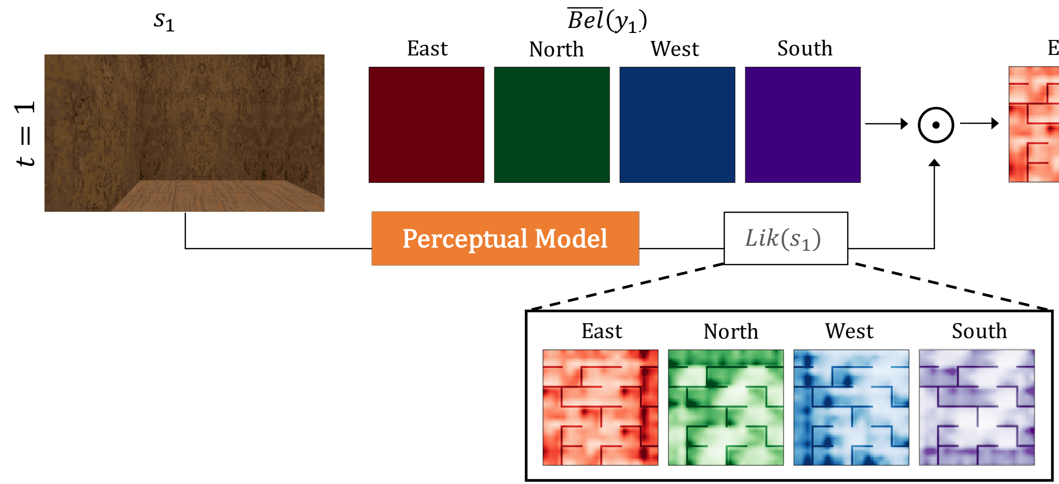


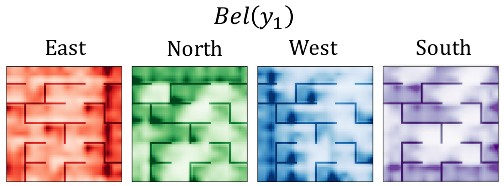
Active Neural Localization

Agent's observation (s_t)

Belief before observing s_t ($\overline{Bel}(y_t)$)

Belief after observing s_t ($Bel(y_t)$)





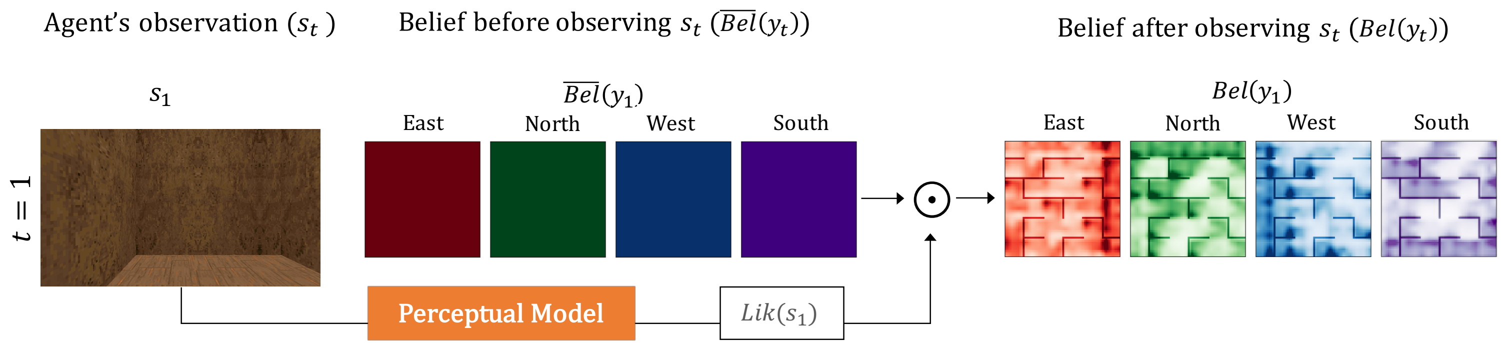
$$Bel(y_t) = \frac{1}{Z} Lik(s_t)\, \overline{Bel}(y_t)$$

Belief after observing s_t

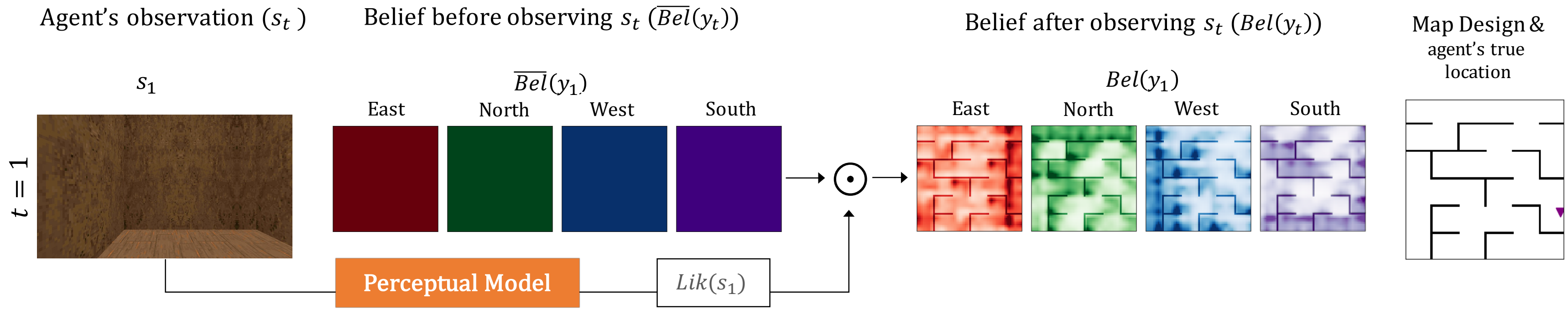
Prob. of observing s_t

Belief before observing s_t

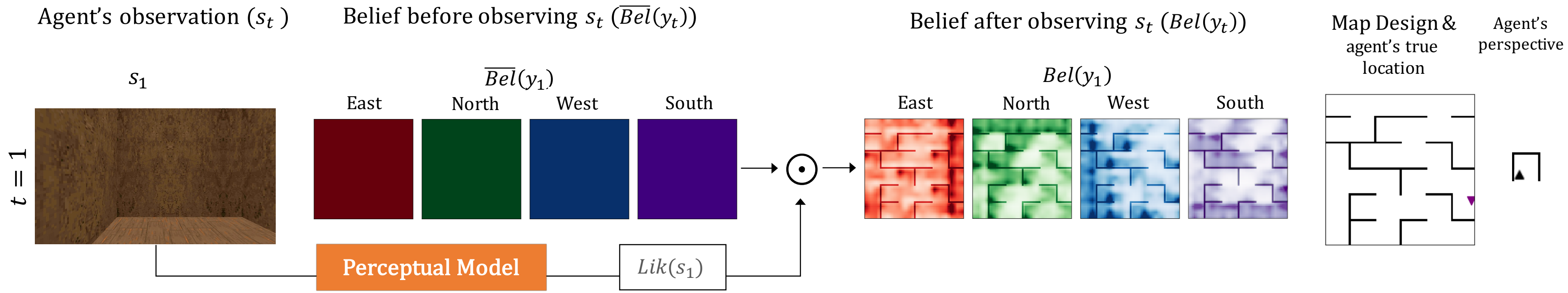
Active Neural Localization



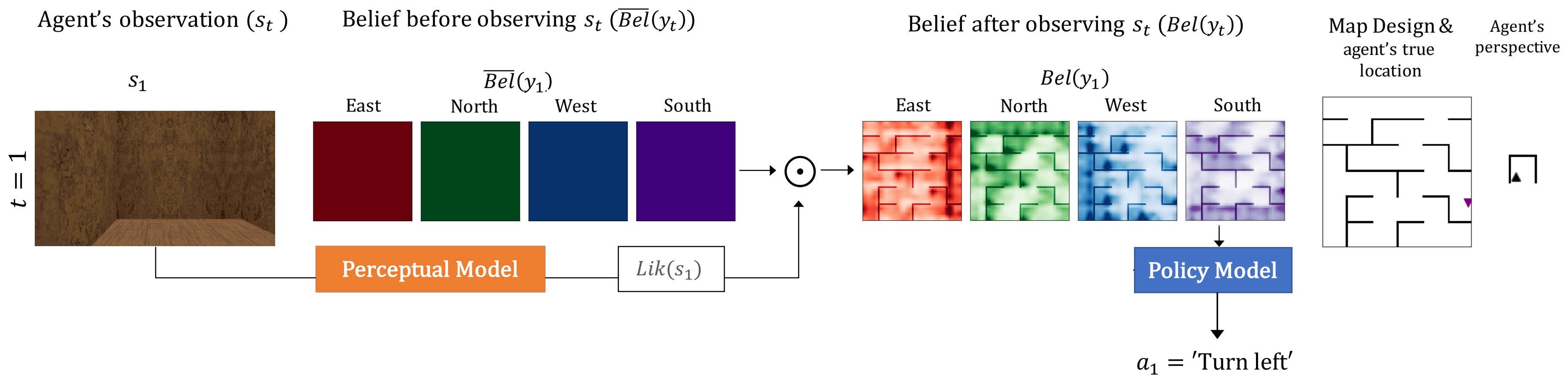
Active Neural Localization



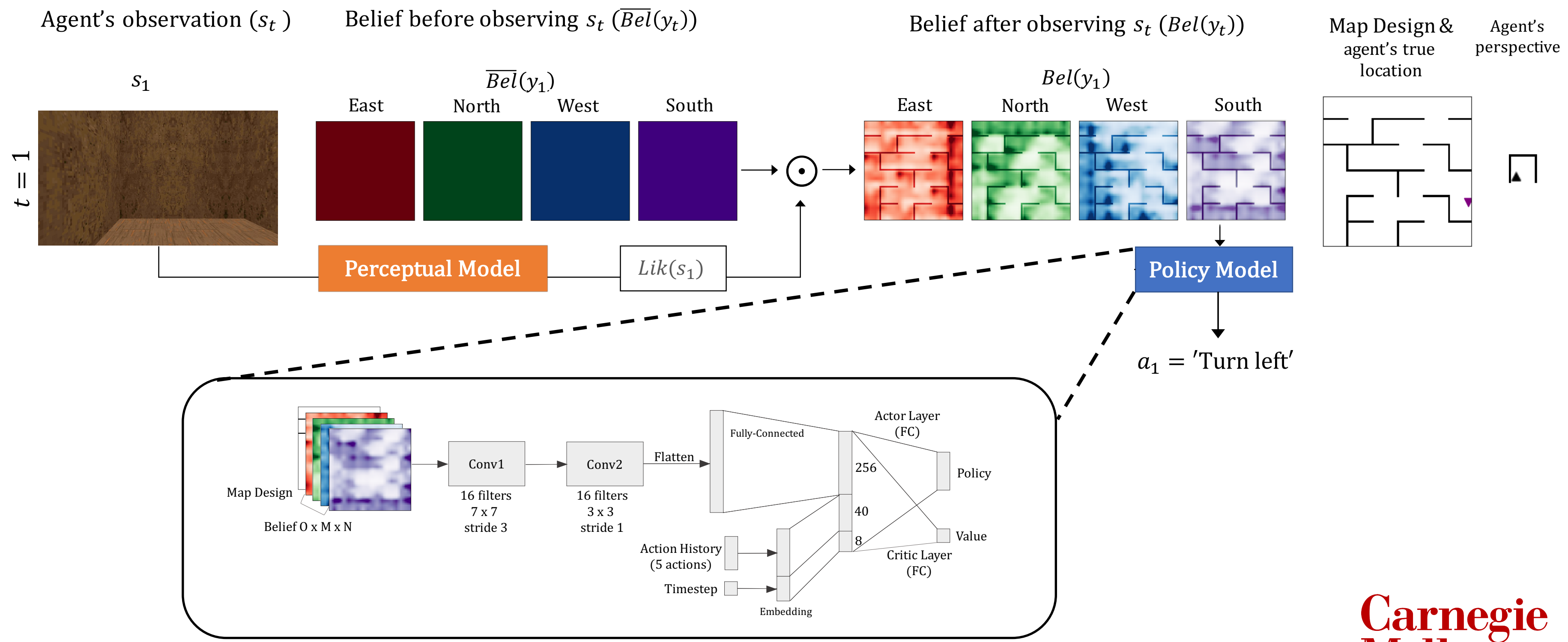
Active Neural Localization



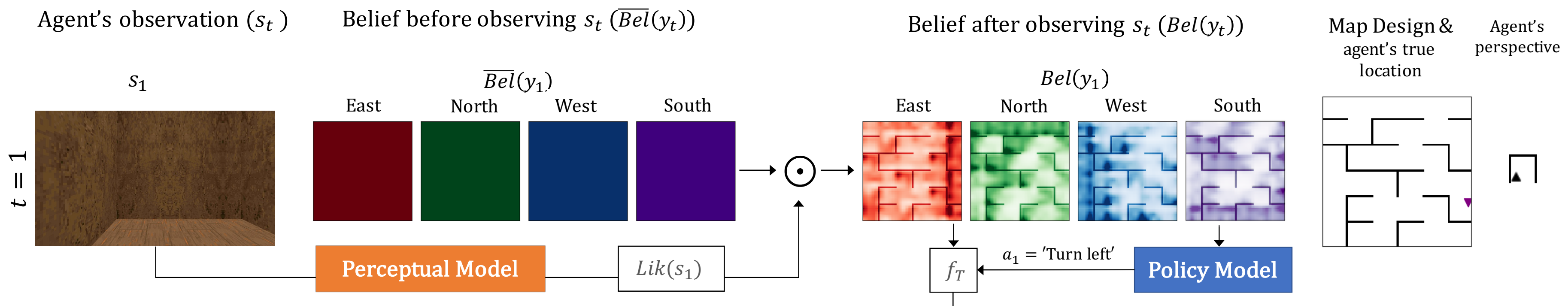
Active Neural Localization



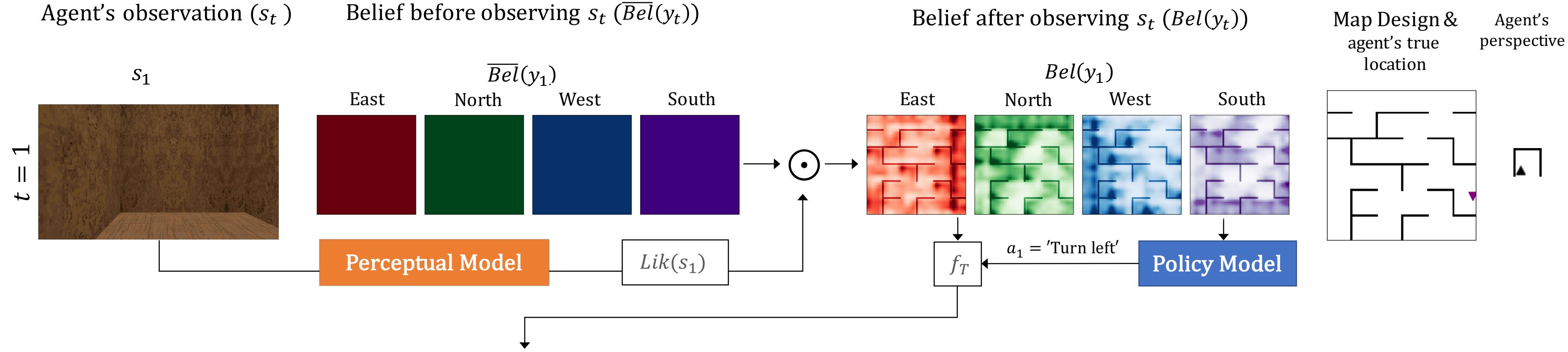
Active Neural Localization

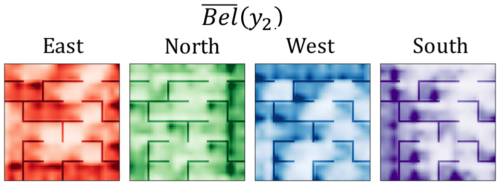


Active Neural Localization



Active Neural Localization



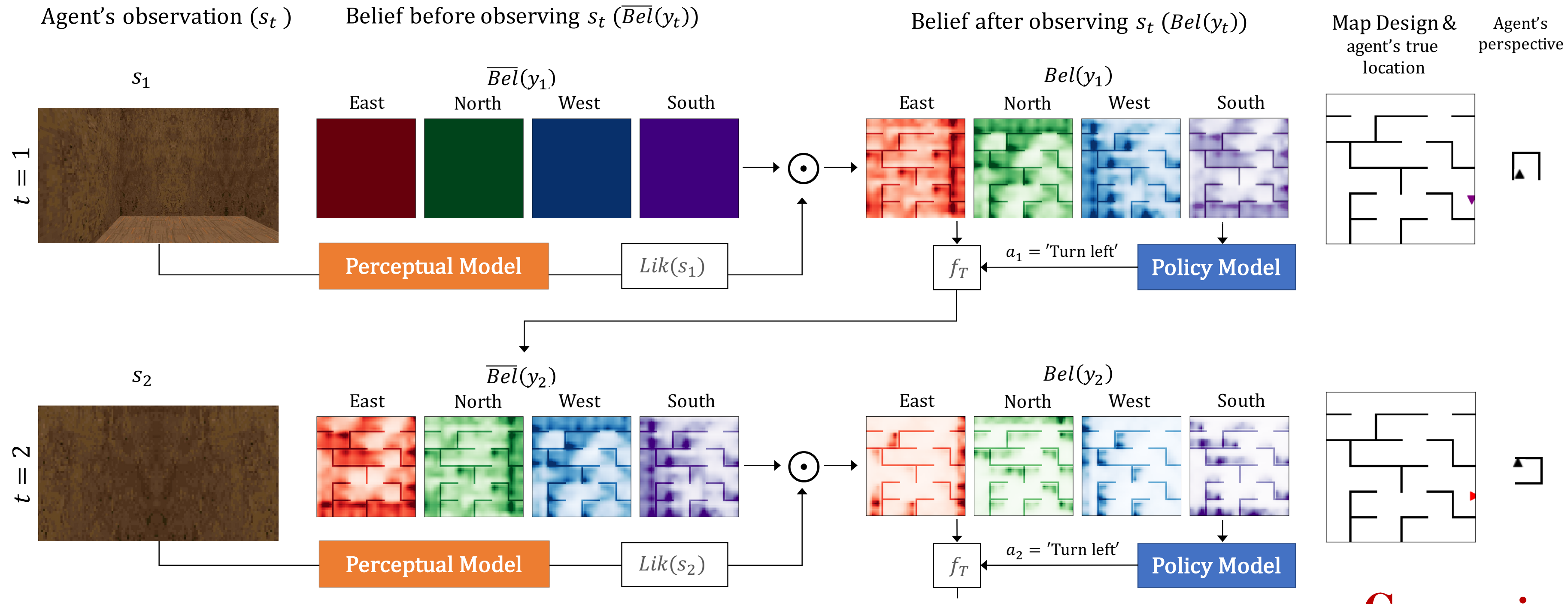


$$\overline{Bel}(y_t) = \sum_{y_{t-1}} P(y_t | y_{t-1}, a_{t-1})\, Bel(y_{t-1})$$

Belief before observing s_t — Transition function — Belief after observing s_{t-1}

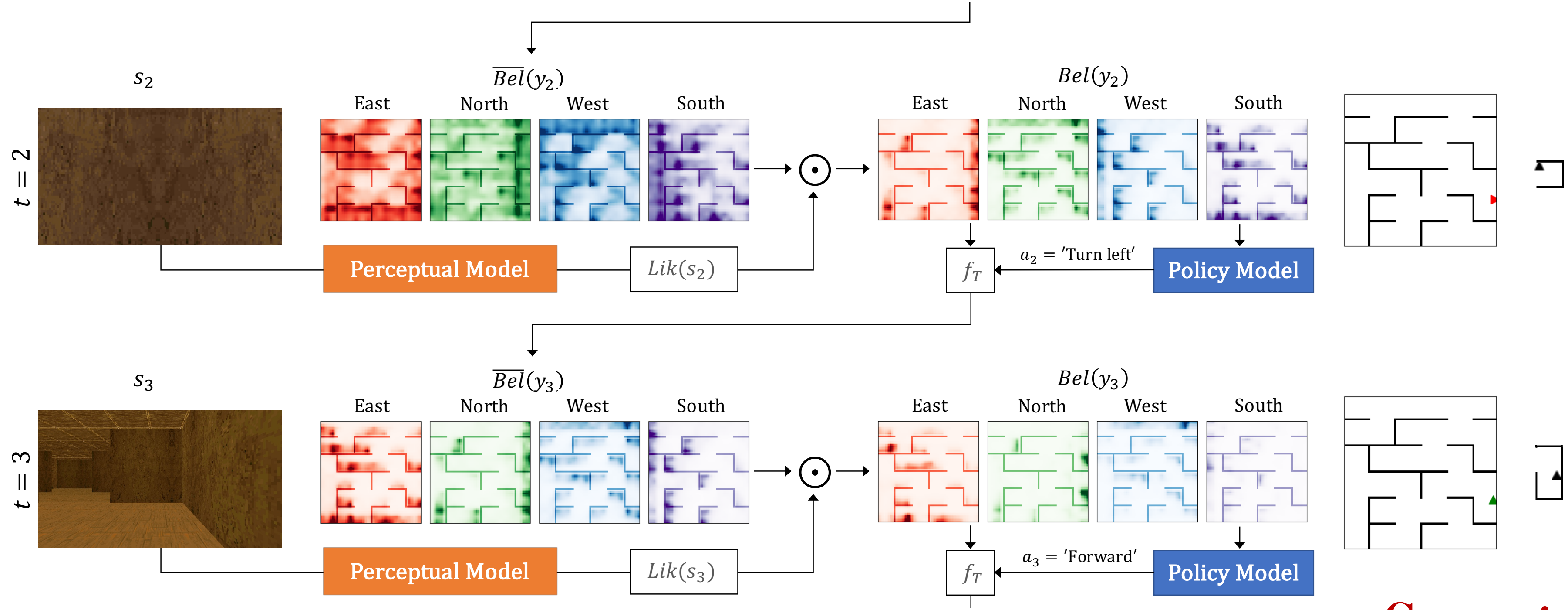
Carnegie Mellon University

Active Neural Localization

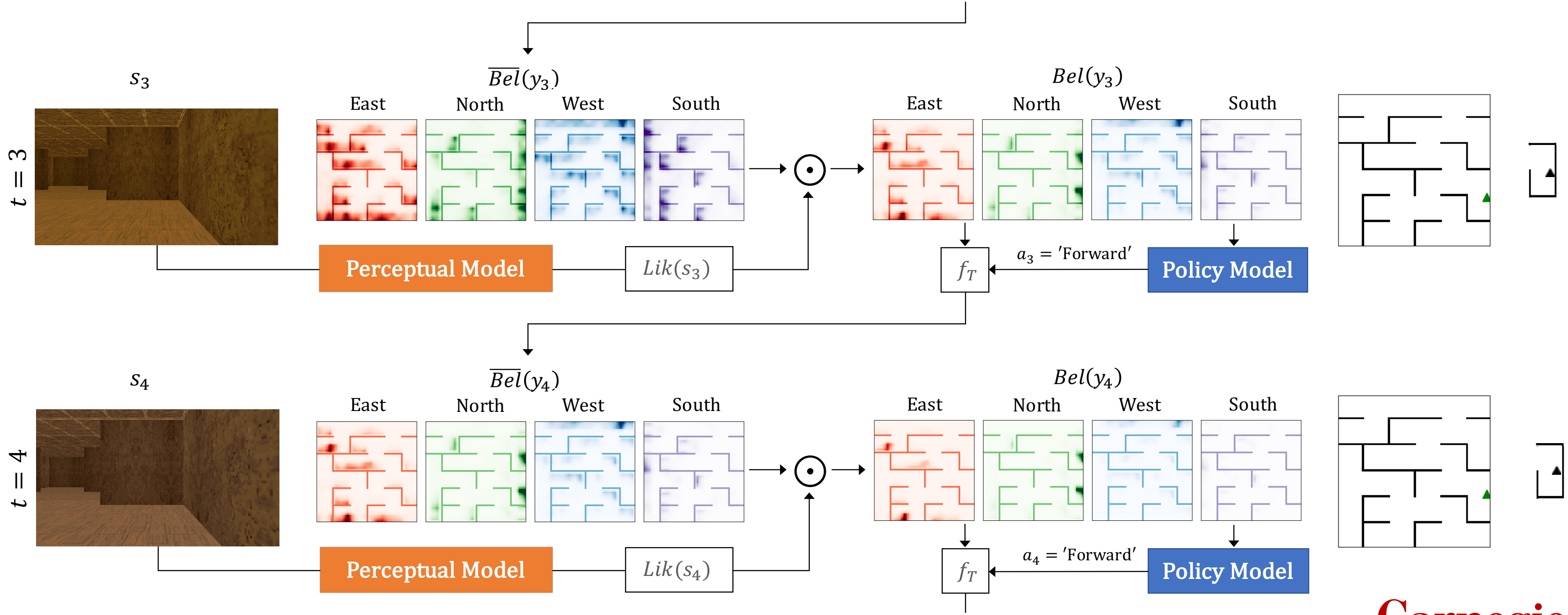


Carnegie Mellon University

Active Neural Localization



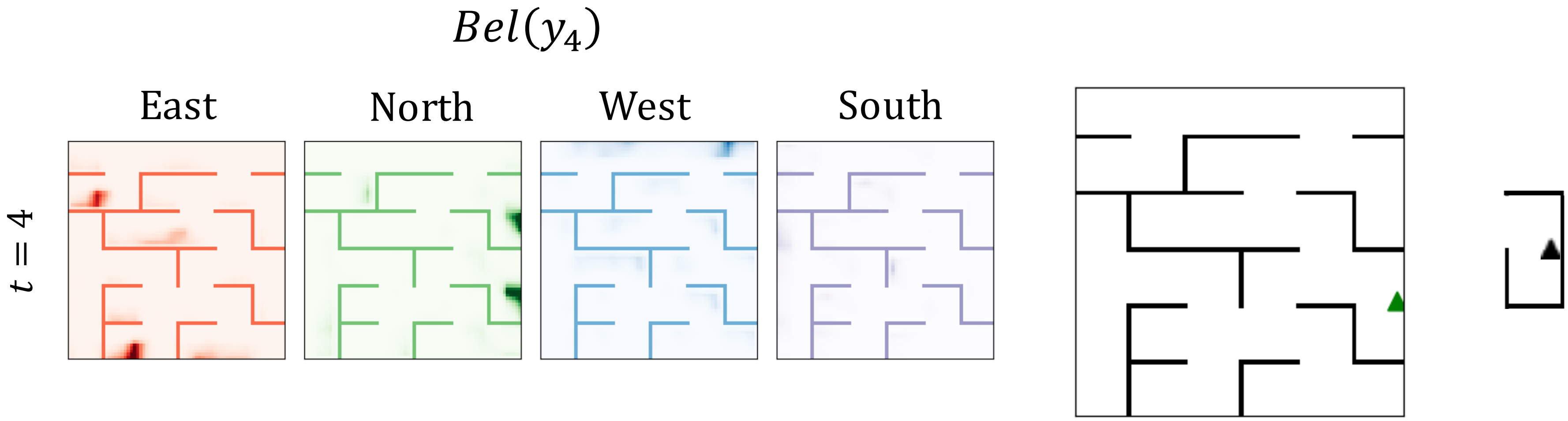
Active Neural Localization



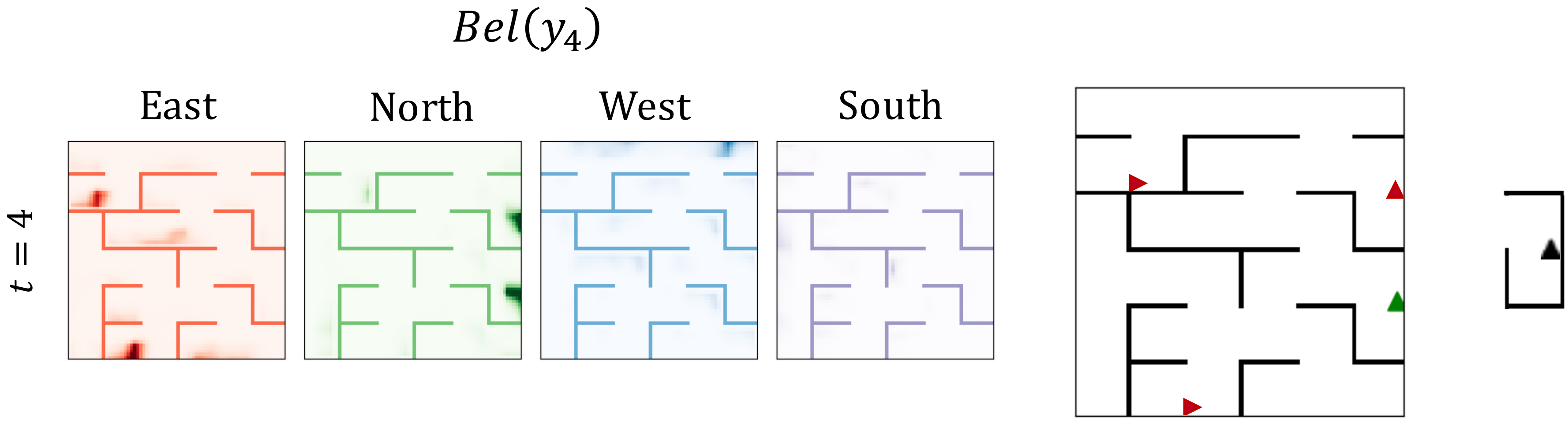
Optimization

- At the end of the episode, the location prediction is the element with the maximum probability in the belief tensor.
- The agent receives a positive reward (+1) for correct prediction.
- The entire model is trained end-to-end with reinforcement learning, specifically Asynchronous Advantage Actor-Critic (A3C).

Example

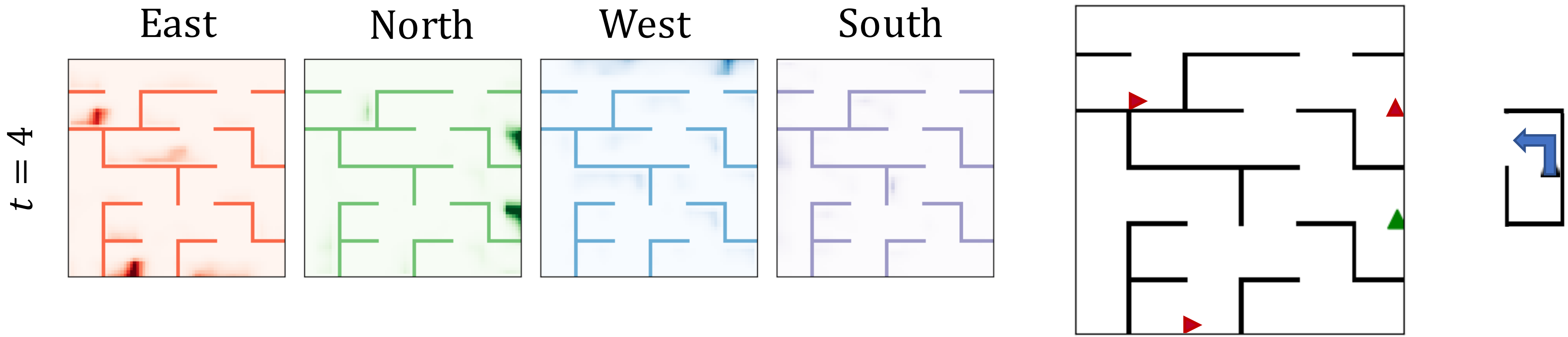


Example

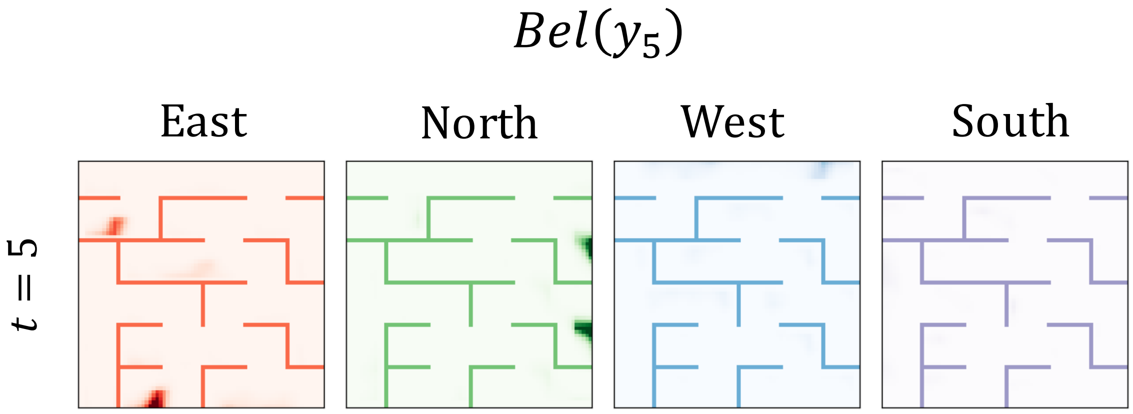


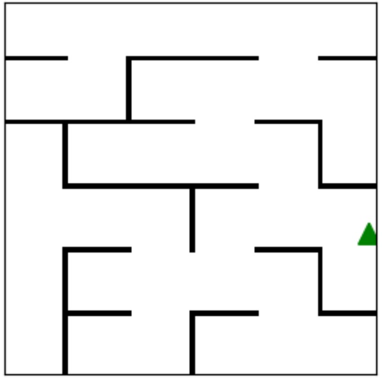
Example

$Bel(y_4)$

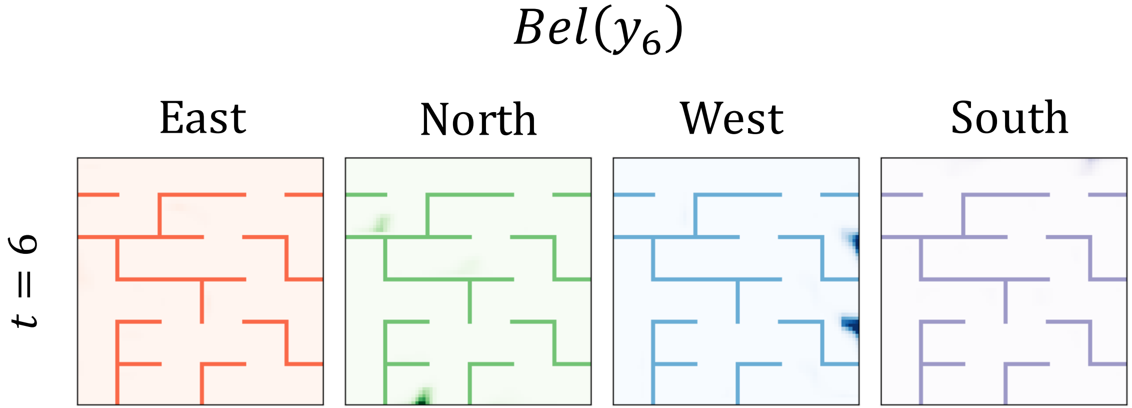


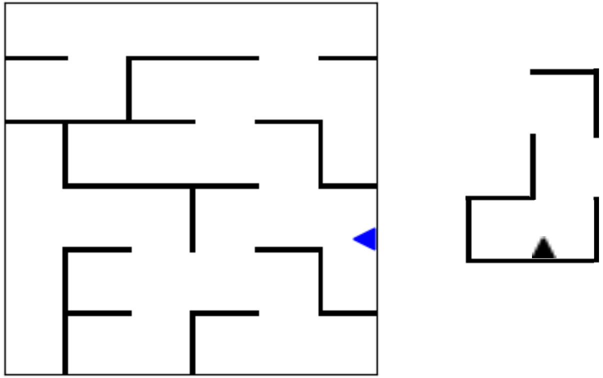
Example





Example





Experiments

Train

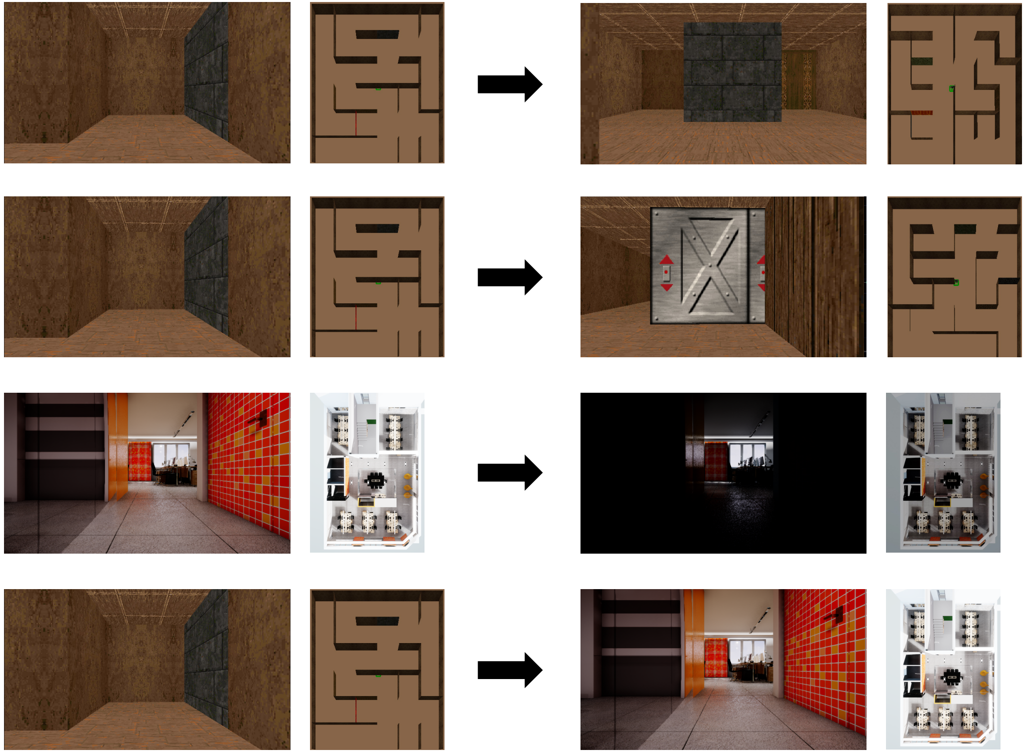
Test

Unseen Mazes

Unseen Mazes
Unseen Textures

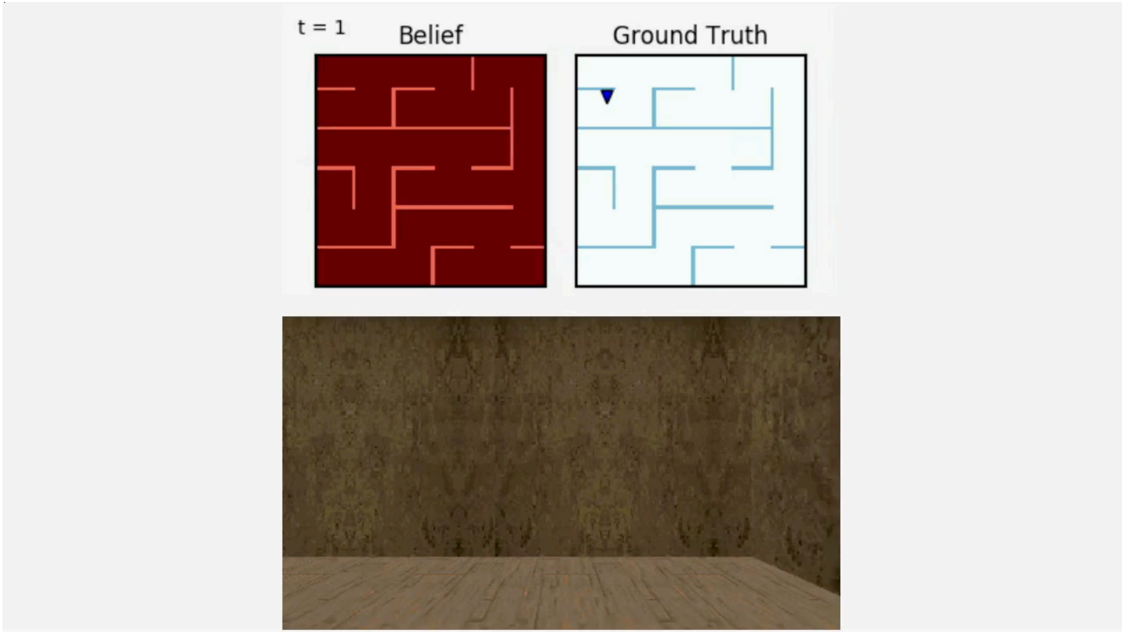
Dynamic
Lighting

Domain
Adaptation



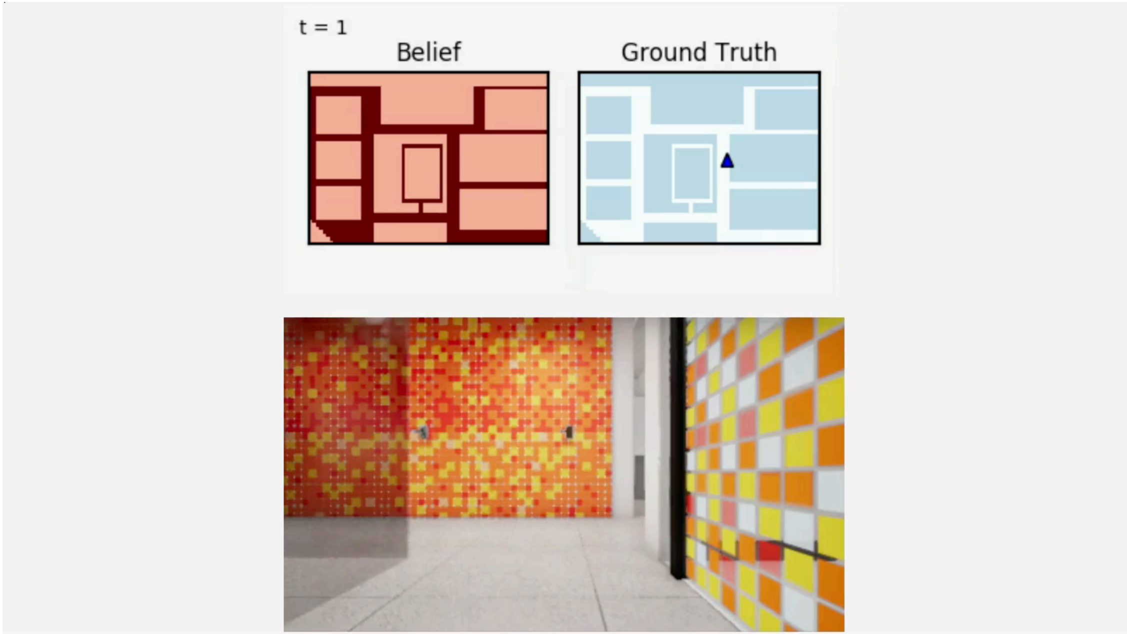
Demo video: Doom

https://www.youtube.com/watch?v=rdhKu8GqVLw

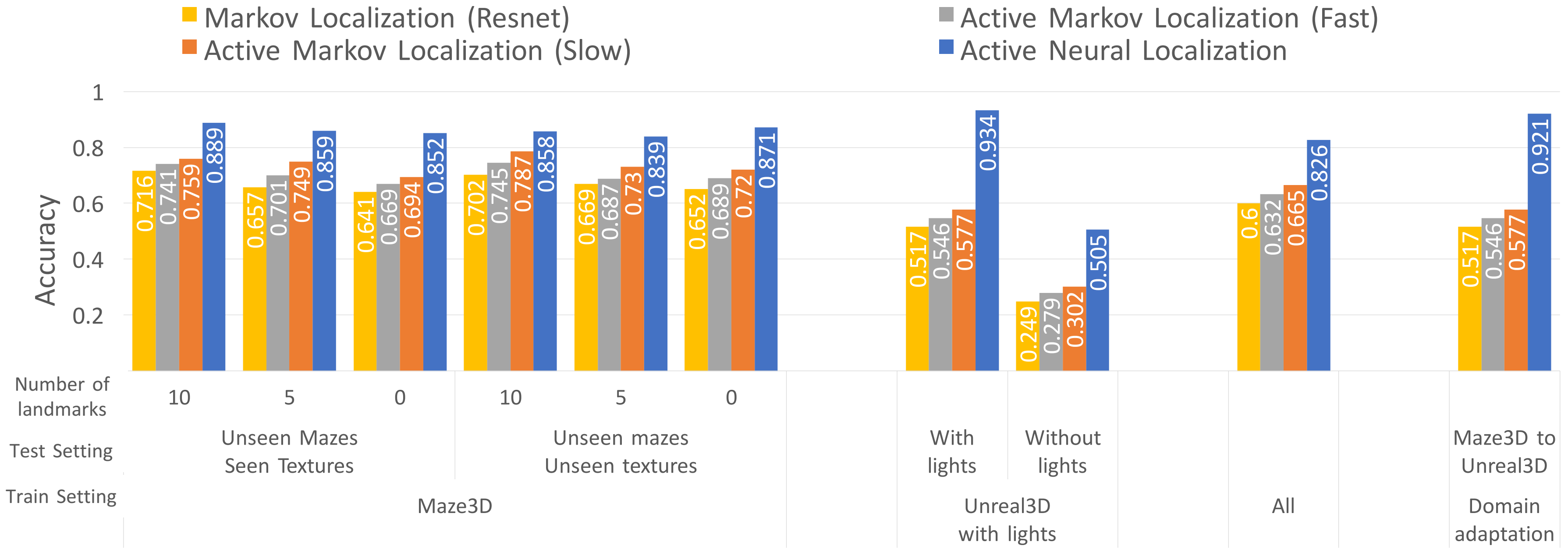


Demo video: Unreal

https://www.youtube.com/watch?v=T5Ezx-_QfU0

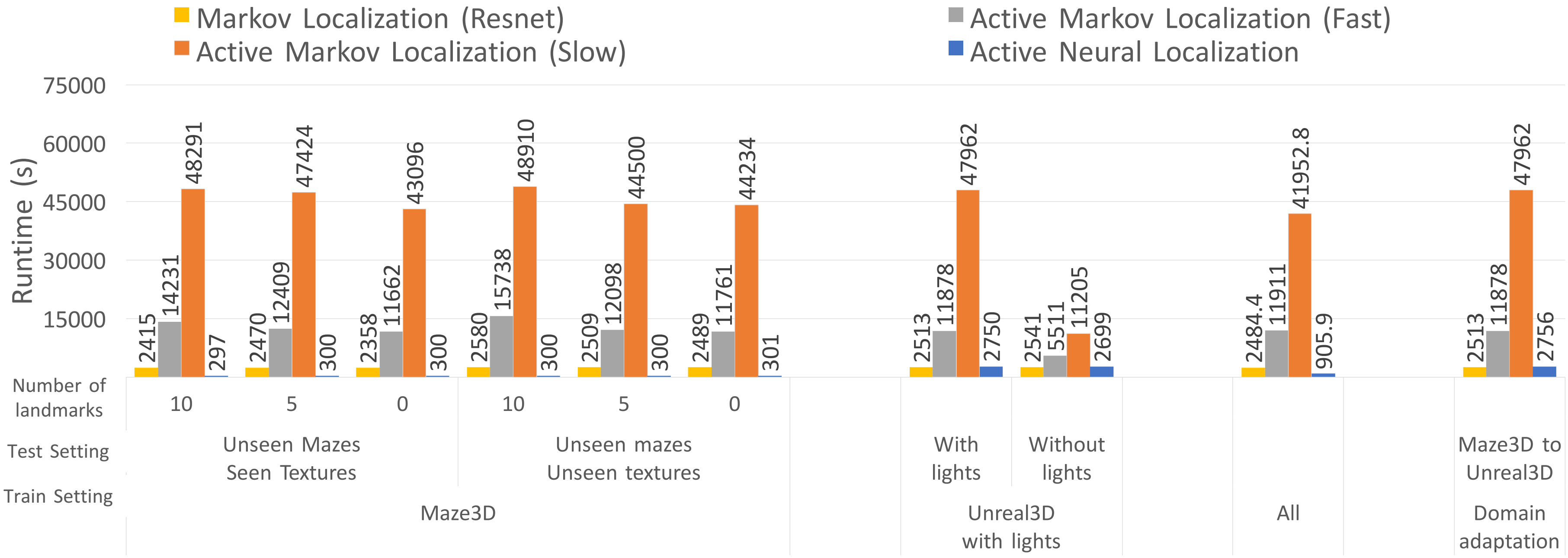


Results



Carnegie Mellon University

Results



Markov Localization (Resnet)
Active Markov Localization (Fast)
Active Markov Localization (Slow)
Active Neural Localization
Runtime (s)
75000
60000
45000
30000
15000
2415
14231
48291
297
2470
12409
47424
300
2358
11662
43096
300
2580
15738
48910
300
2509
12098
44500
300
2489
11761
44234
301
2513
11878
47962
2750
2541
5511
11205
2699
2484.4
11911
41952.8
905.9
2513
11878
47962
2756
Number of landmarks
10
5
0
10
5
0
Test Setting
Unseen Mazes Seen Textures
Unseen mazes Unseen textures
With lights
Without lights
Maze3D to Unreal3D
Train Setting
Maze3D
Unreal3D with lights
All
Domain adaptation

Contributions

- Fully-differentiable model for active localization
  - Incorporates ideas of traditional filtering-based localization methods by using a structured belief
  - Capable to deciding actions for accurate and efficient localization
  - Entire model trained end-to-end using reinforcement learning
    - Allows perceptual model and policy model to be trained jointly
    - Doesn’t require labels, needs only a reward at the end of the episode
- Generalization to not only unseen maps in the same domain but also across domains.

Active Neural Localization

Devendra Singh Chaplot, Emilio Parisotto, Ruslan Salakhutdinov

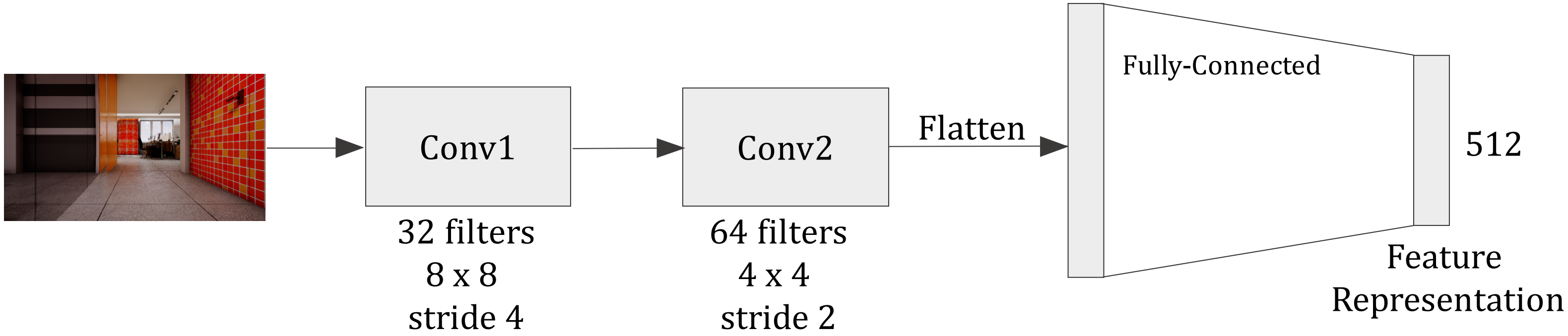
Questions?

Appendix

Results (2D)

Env		Maze2D						All
Map Size		7x7		15x15		21x21		
Episode Length		15	30	20	40	30	60	
Markov Localization	Time	12	15	29	31	49	51	31.2
	Acc	0.334	0.529	0.351	0.606	0.414	0.661	0.483
Active Markov Localization (Fast)	Time	29	53	72	165	159	303	130.2
	Acc	0.436	0.619	0.468	0.657	0.512	0.735	0.571
Active Markov Localization (Slow)	Time	1698	3066	3791	8649	8409	13554	6527.8
	Acc	0.854	0.938	0.846	**0.984**	0.845	0.958	0.904
Active Neural Localization	Time	22	34	44	66	82	124	62.0
	Acc	**0.936**	**0.939**	**0.905**	0.939	**0.899**	**0.984**	**0.934**

Perceptual Model



Policy Model