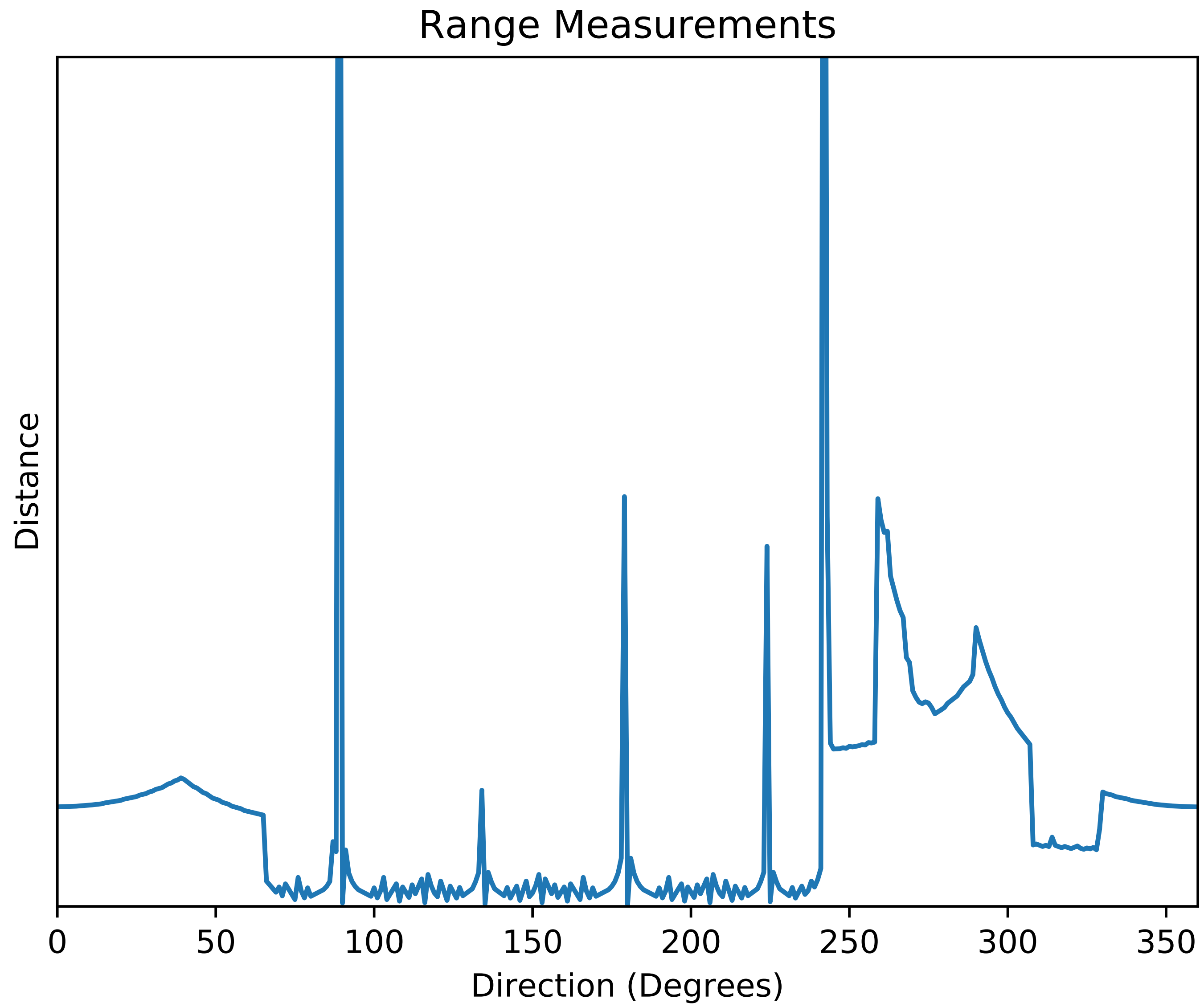
Machine learning approaches for optimization problems involving lines-of-sight

Louis Ly and Richard Tsai

The University of Texas at Austin
and
KTH Royal Institute of Technology, Sweden

Overview

- **Range data** from a sensor, sampling the environment at designated locations (lines-of-sight)
- Locations of the sensor

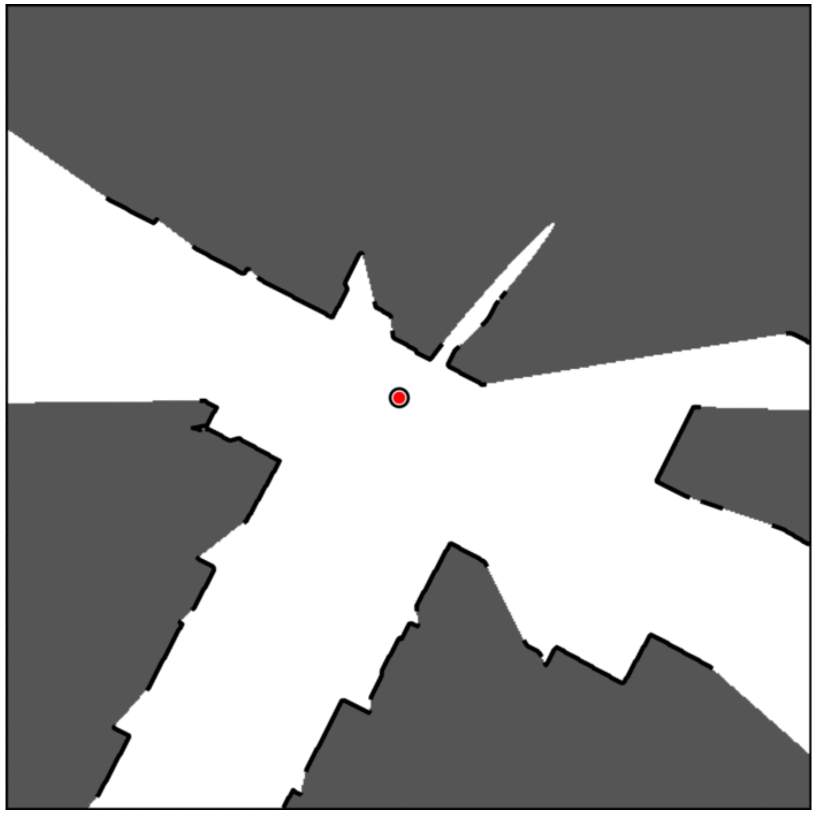


Overview

- **Objectives**:

  Determine and use as few observing locations as possible to

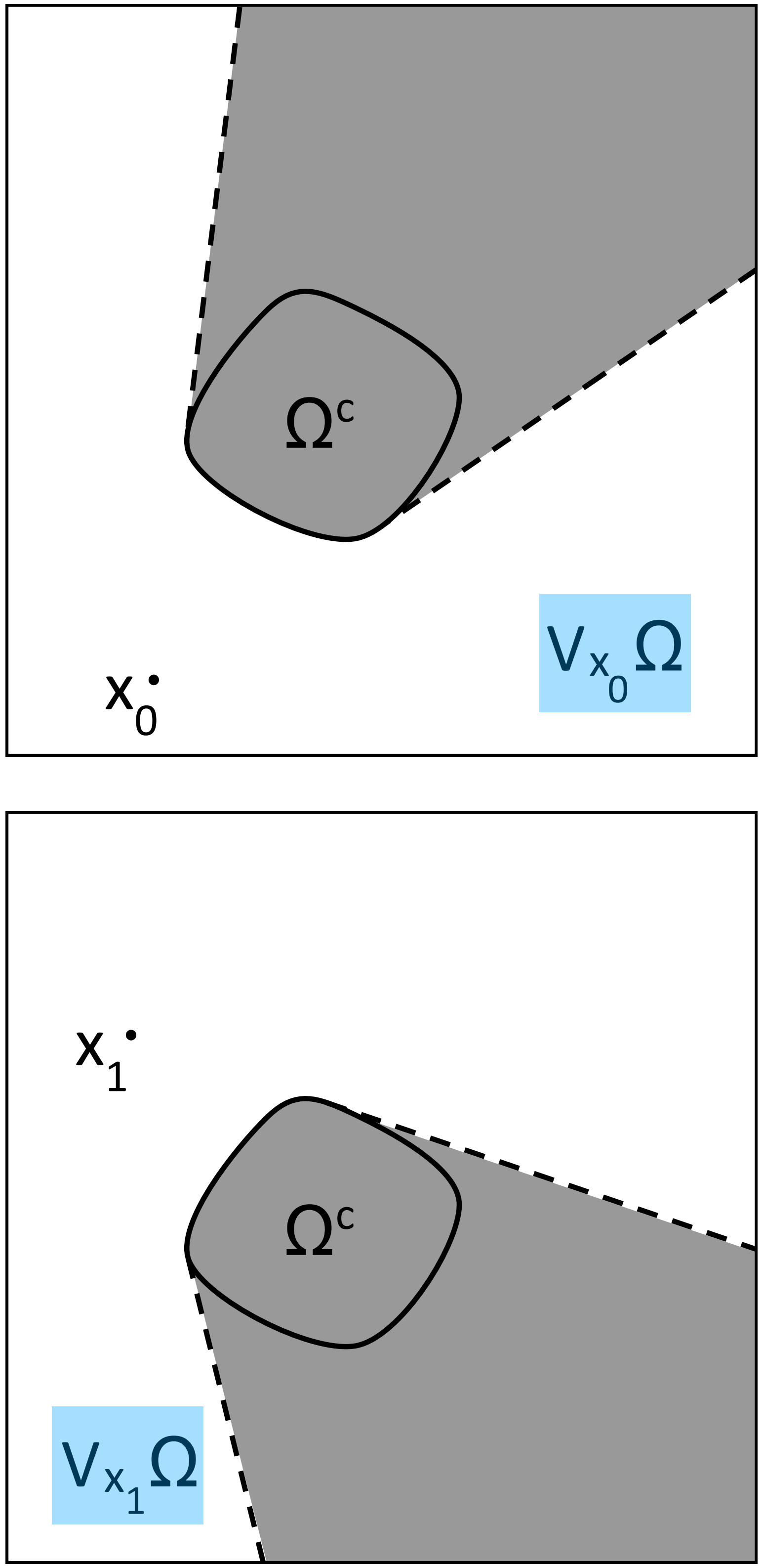
  - **learn** (map out) the environment
  - put an a priori unknown environment under **surveillance**



Simulation using the exploration algorithm to be discussed

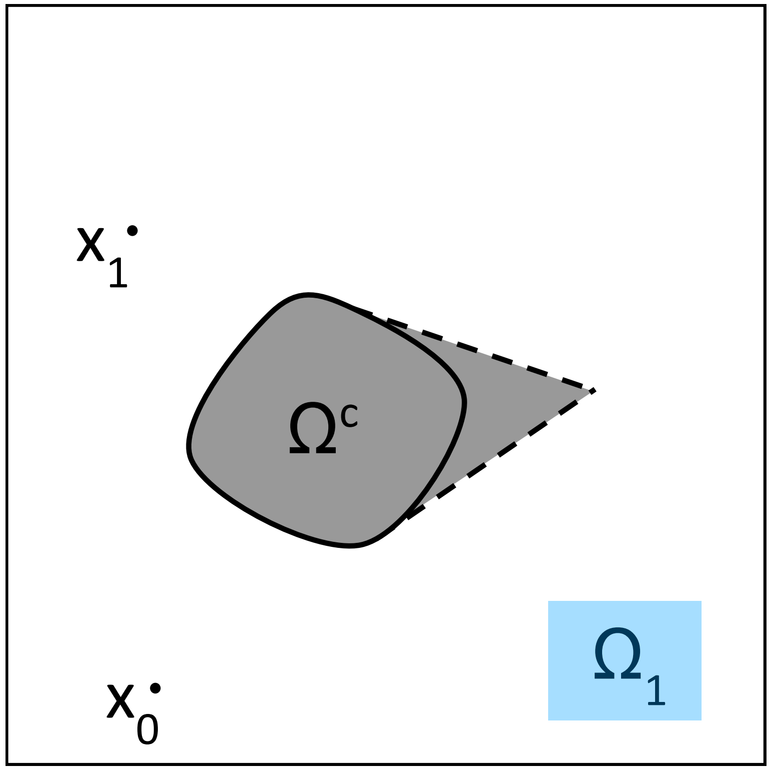
Problem formulation

Notation	Description
$D \subset \mathbb{R}^d$	Domain
$\Omega \subset D$	Free space
$x_i \in D$	Vantage points
$\mathcal{V}_{x_i}\Omega$	Visibility set from x_i
$\Omega_k = \bigcup_{i=0}^{k} \mathcal{V}_{x_i}\Omega$	Cumulatively visible set



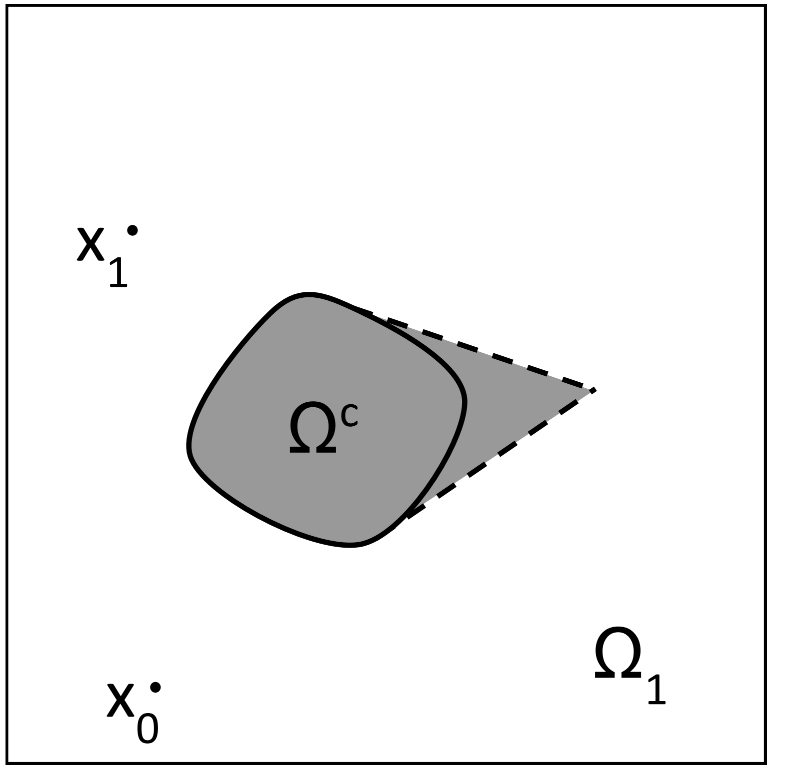
Problem formulation

Notation	Description
$D \subset \mathbb{R}^d$	Domain
$\Omega \subset D$	Free space
$x_i \in D$	Vantage points
$\mathcal{V}_{x_i}\Omega$	Visibility set from x_i
$\Omega_k = \bigcup_{i=0}^{k} \mathcal{V}_{x_i}\Omega$	Cumulatively visible set



“Sparse solution”

Notation	Description
$D \subset \mathbb{R}^d$	Domain
$\Omega \subset D$	Free space
$x_i \in D$	Vantage points
$\mathcal{V}_{x_i}\Omega$	Visibility set from x_i
$\Omega_k = \bigcup_{i=0}^{k} \mathcal{V}_{x_i}\Omega$	Cumulatively visible set
$I : \mathbb{R}^{m^d} \to \{0,1\}$	Indicator function

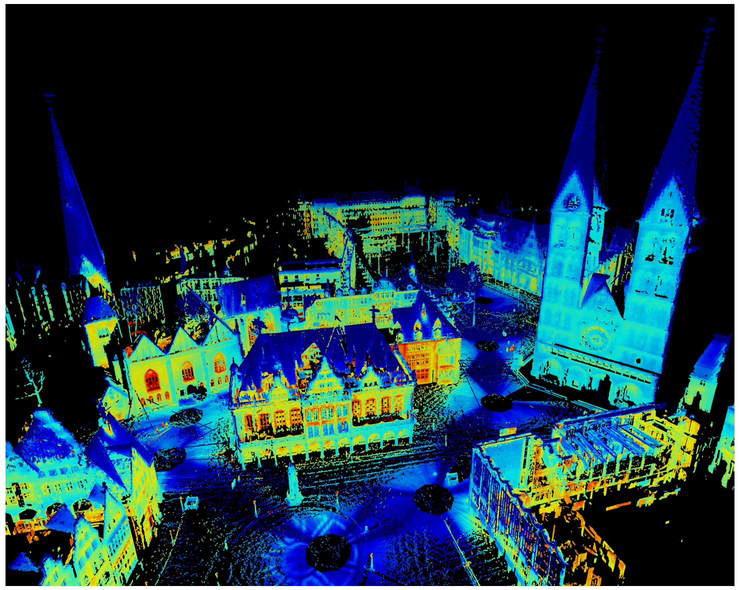


$$\min_I ||I||_0 \quad \text{subject to} \quad \bigcup_{\{x_j | I(x_j)=1\}} \mathcal{V}_{x_j}\Omega = \Omega$$

(formulated for surveillance of a known environment)

Applications

- Surveillance: if Ω is known.
- Exploration: if Ω is unknown.
- Reconstruction: high resolution scans of scene or structure
- Compression of scan data



Supervised learning

$$\min_{I} \|I\|_0 \quad \text{subject to} \quad \bigcup_{\{x_i | I(x_i)=1\}} \mathcal{V}_{x_i}\Omega = \Omega$$

- Can we use ML to speed up computation and learn priors for large class of obstacles?
- Given data X, and labels Y, learn the mapping $X \mapsto Y$.
- Learn function approximator f_θ parametrized by θ.

$$\arg\min_{\theta} \|Y - f_\theta(X)\|$$

- What should X, Y be for our problem?

Our approach

- Greedy algorithm
- “Reward” computed by a convolutional neural network
- Visibility level set method

Terms

- **Cumulative Visibility**: the space that is visible to at least one observing location. = estimate of the terrain

- **Frontiers (shadow boundaries)**: the boundary between the occluded and visible region, excluding obstacles.
existence of un-explored region

- **Gain**: the increase of visible region resulting from a move to each location.

- **Next Step**: the location at which gain is maximized.

Greedy approach

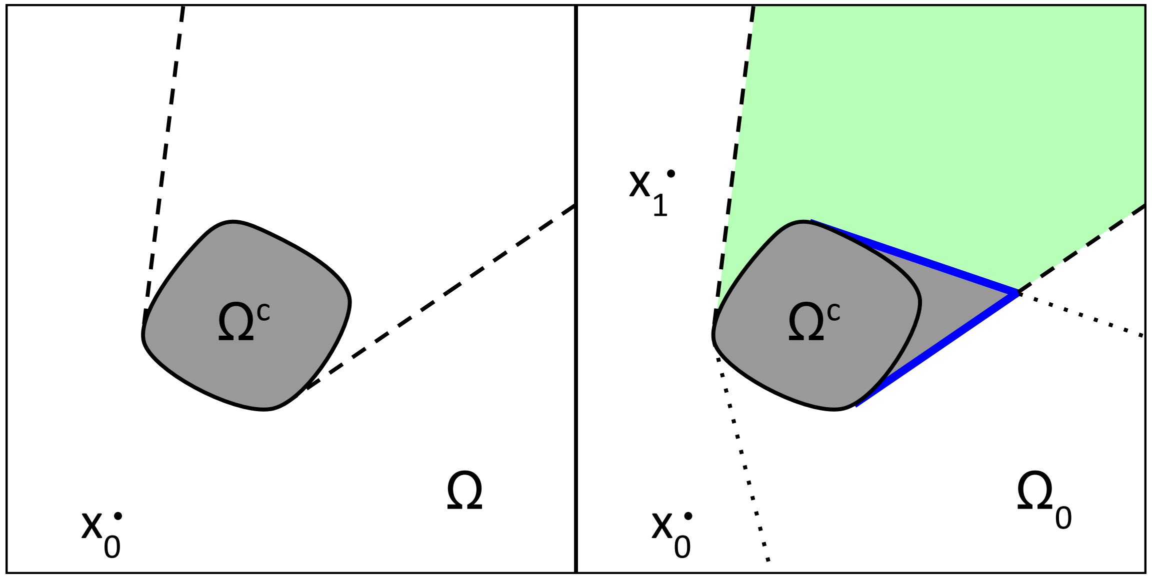
$$\min_{I} \|I\|_0 \quad \text{subject to} \quad \bigcup_{\{x_i | I(x_i)=1\}} \mathcal{V}_{x_i}\Omega = \Omega$$

Assume environment Ω is known. Define gain function

$$g(x; \Omega_k, \Omega) := |\mathcal{V}_x\Omega \cup \Omega_k| - |\Omega_k|$$

Algorithm:

$$x^{k+1} = \arg\max_x g(x; \Omega_k, \Omega)$$



- Gain function ***g*** captures the volume of region (green) revealed by moving to ***x***.

Complexity of 1-step gain function

Fix the current cumulative visibility Ψ_k.
$\Psi_k(x) > 0 \implies x$ is visible from one of $\{x_i\}_{i=0}^{k}$.
Let $D = \mathbb{R}^{m \times m}$. Let $n \geq m$ be # grid pts for visibility computation.

```
for x ∈ D do
    Compute visibility ψ(x, ·)                              O(n^2)
    g(x; ·) = Area({ξ | ψ(x, ξ) > 0, Ψ_k(ξ) < 0})           O(n^2)
end for
```

Total $O(m^2 n^2)$ flops.

Greedy approach

- Does using the “gain” function in a greedy algorithm give good solutions?
  - Different notions of “gains”
- For known Ω, need an efficient way to compute it
- For unknown Ω, need a way to approximate it, using priors
- What features from the data does it depend on?

Submodularity

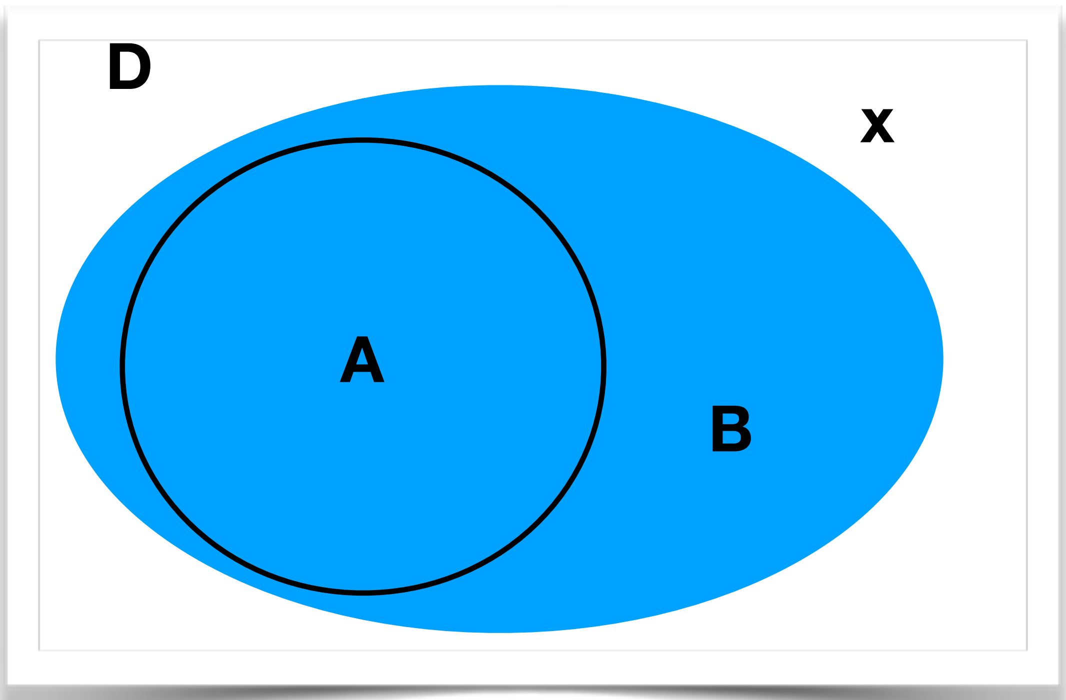
- Let $O = \{x_i\}$ be sets of vantage points.
- Let $f(O)$ be the area of region visible from points in O.
- The function f is monotone (placing more vantage points increases visibility):

$$f(A) \leq f(B)$$

- The function f is submodular (diminishing returns):
  suppose $A \subseteq B \subseteq D$ and $x \in D \setminus B$,

$$f(A \cup \{x\}) - f(A) \geq f(B \cup \{x\}) - f(B)$$

Our gain function for the set A



- A common coverage problem:
  - Maximization of the submodular function with fixed number of sensors
- Our problem setup:
  - Minimize the number of sensors subject to a coverage constraint

Greedy maximization

- Let S^* the optimal set of k sensors.
- Let $O = \{x_i\}_{i=1}^{n}$ be set of n sensors placed using greedy approach.
- Theorem [1]:

$$f(O) \geq \left(1 - e^{-n/k}\right) f(S^*)$$

- The greedy algorithm covers at least $1 - e^{-1} \approx 0.63$ of the map in $n = k$ steps.
- The greedy algorithm covers at least $1 - e^{-5} \approx 0.99$ of the map in $n = 5k$ steps.
- Any algorithm that only evaluates f at a polynomial number of sets will not be able to obtain a better guarantee. [2]

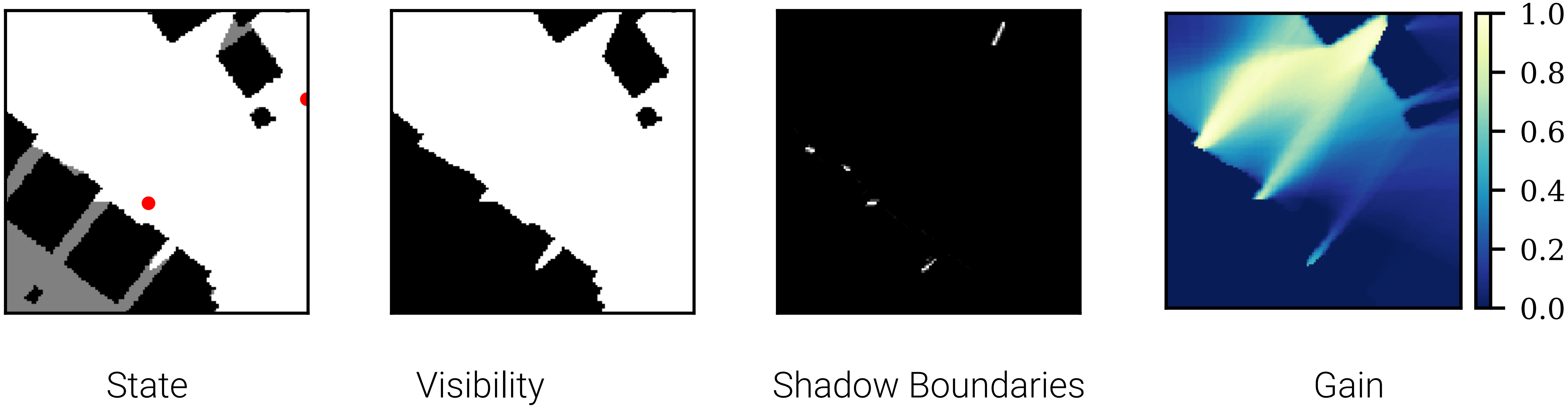
---

[1] Nemhauser, George L., Wolsey, Laurence A., and Fisher, Marshall L. 1978. An analysis of approximations for maximizing submodular set functions - I. Mathematical Programming, 14(1), 265–294.

[2] Nemhauser, G. L., and Wolsey, L. A. 1978. Best algorithms for approximating the maximum of a submodular set function. Math. Oper. Research, 3(3), 177–188.

Inference of the gain function

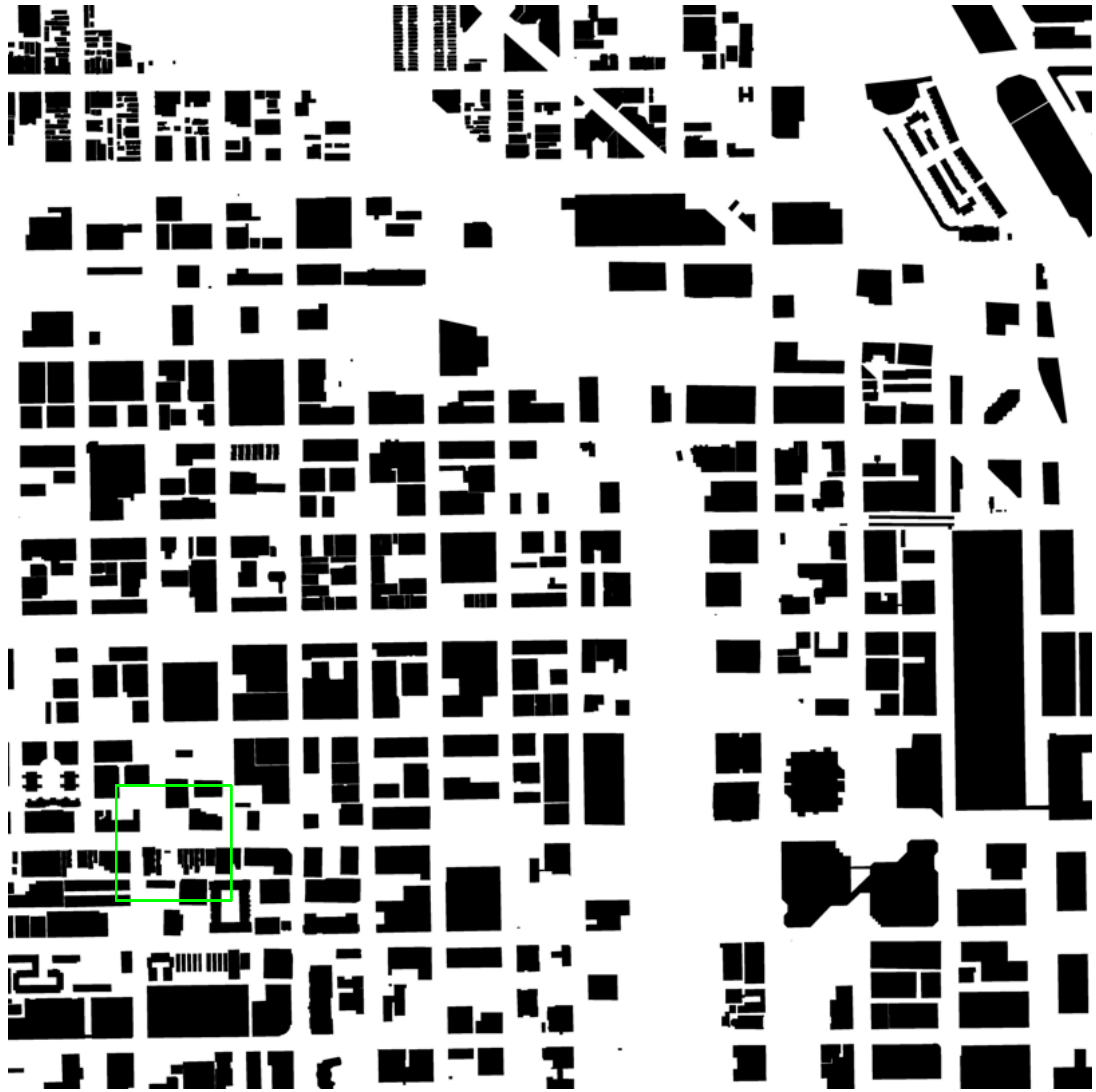
- Seek $g_\theta(x; \Omega_k; B_k) \approx g(x; \Omega_k, \Omega)$
- Use Convolutional Neural Networks as function approximator
- Input: Cumulative visibility and shadow boundaries
- Output: Gain function



Training data

- ◇ Environments Ω^i sampled from database.
- ◇ For each Ω^i, sample sequence of vantage points $\boldsymbol{x}_k$.
- ◇ Generate data pair $\left\{(\Omega^i_k, B^i_k), g(\cdot)\right\}$

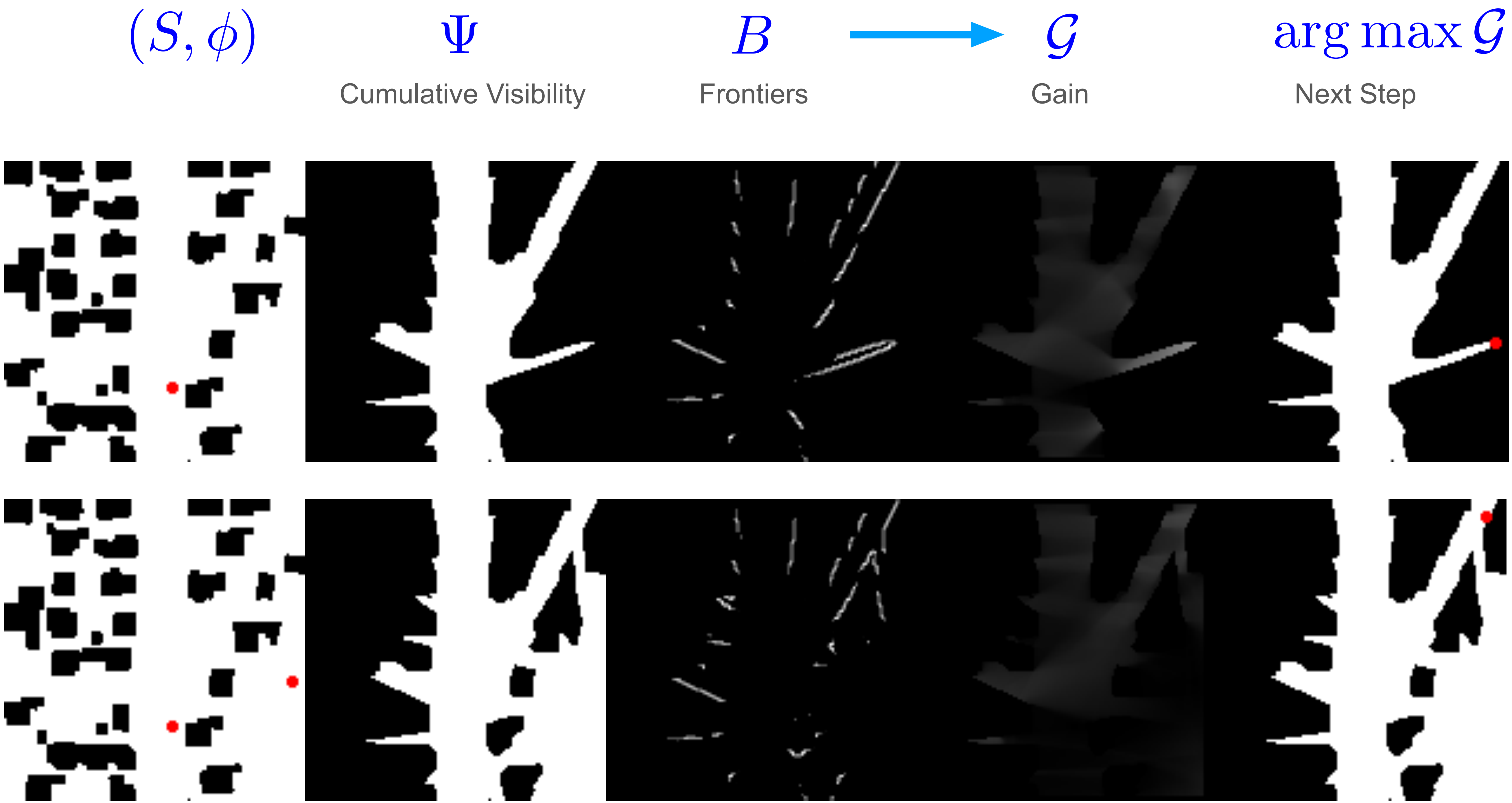
Chicago

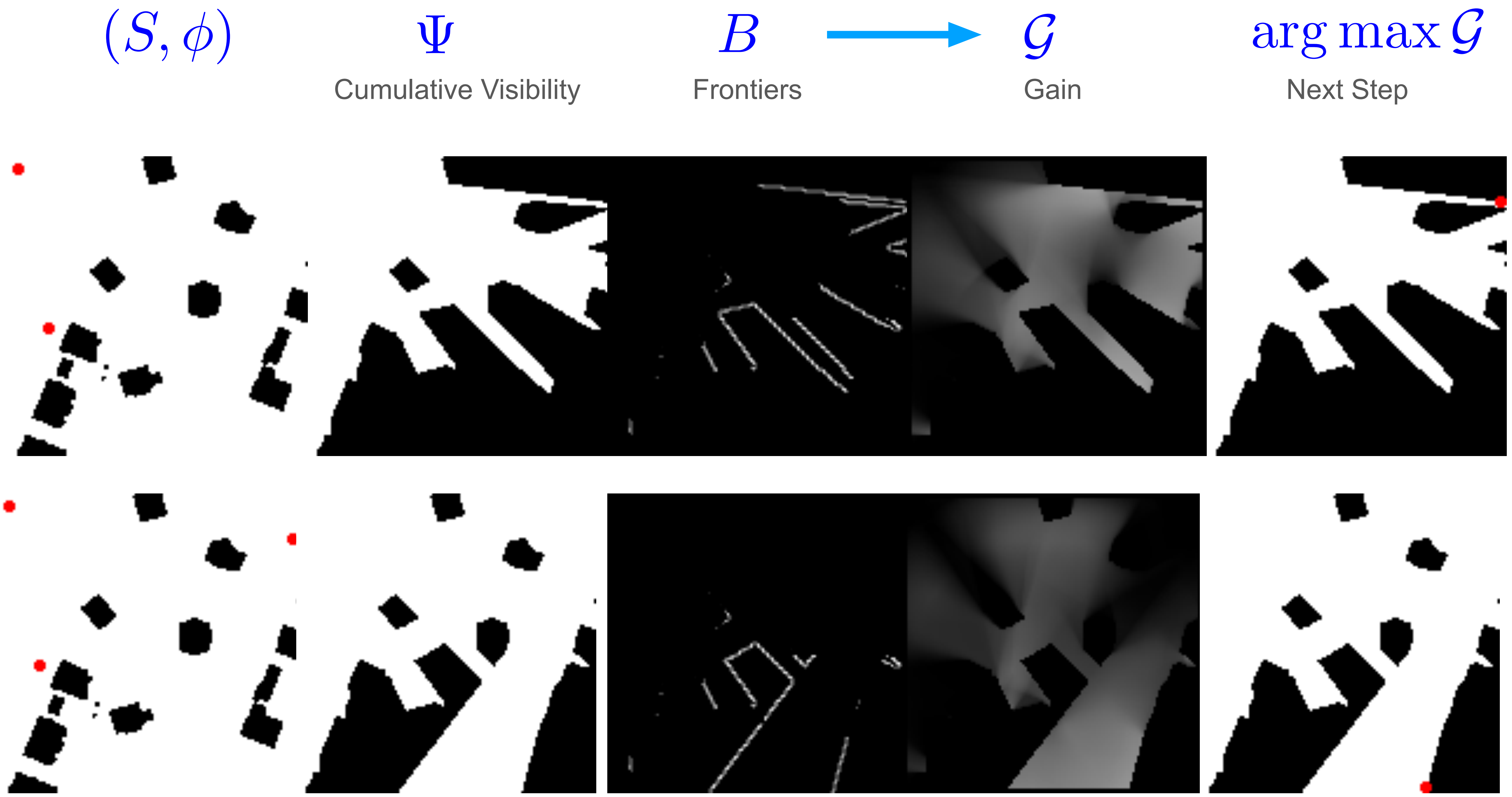


Tyrol

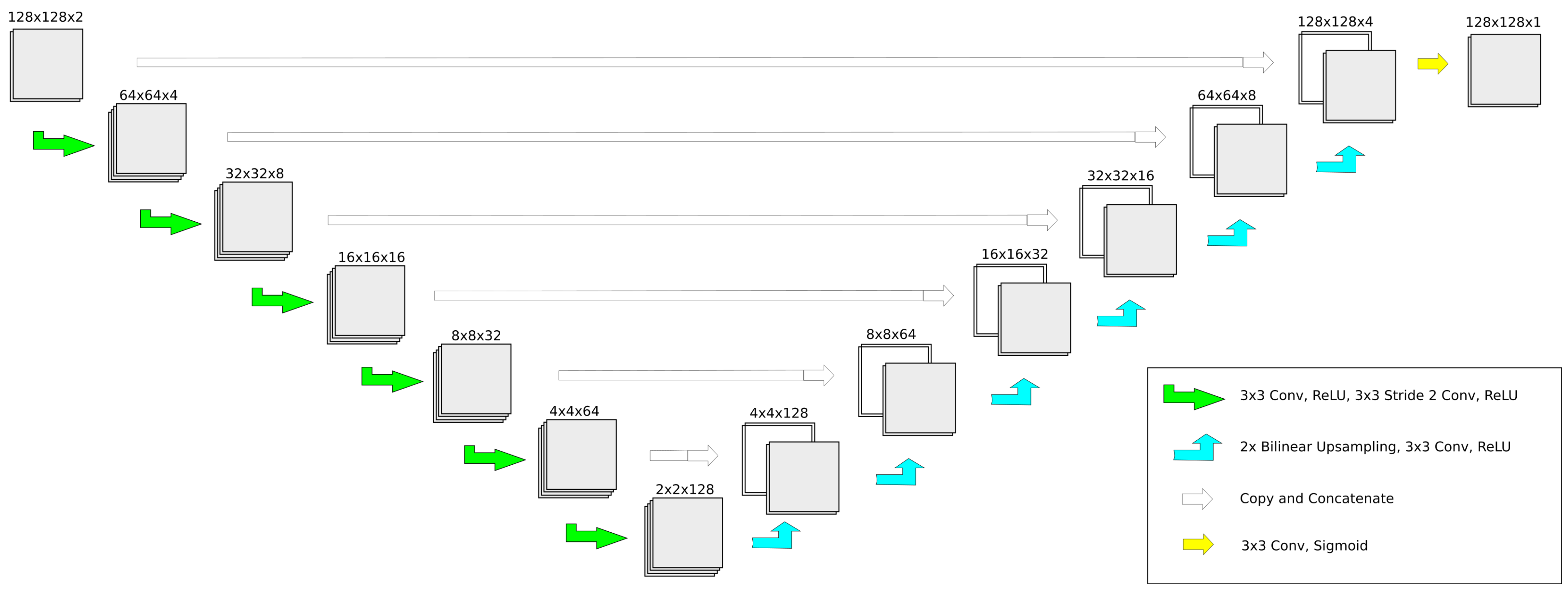
Vienna

Kitsap

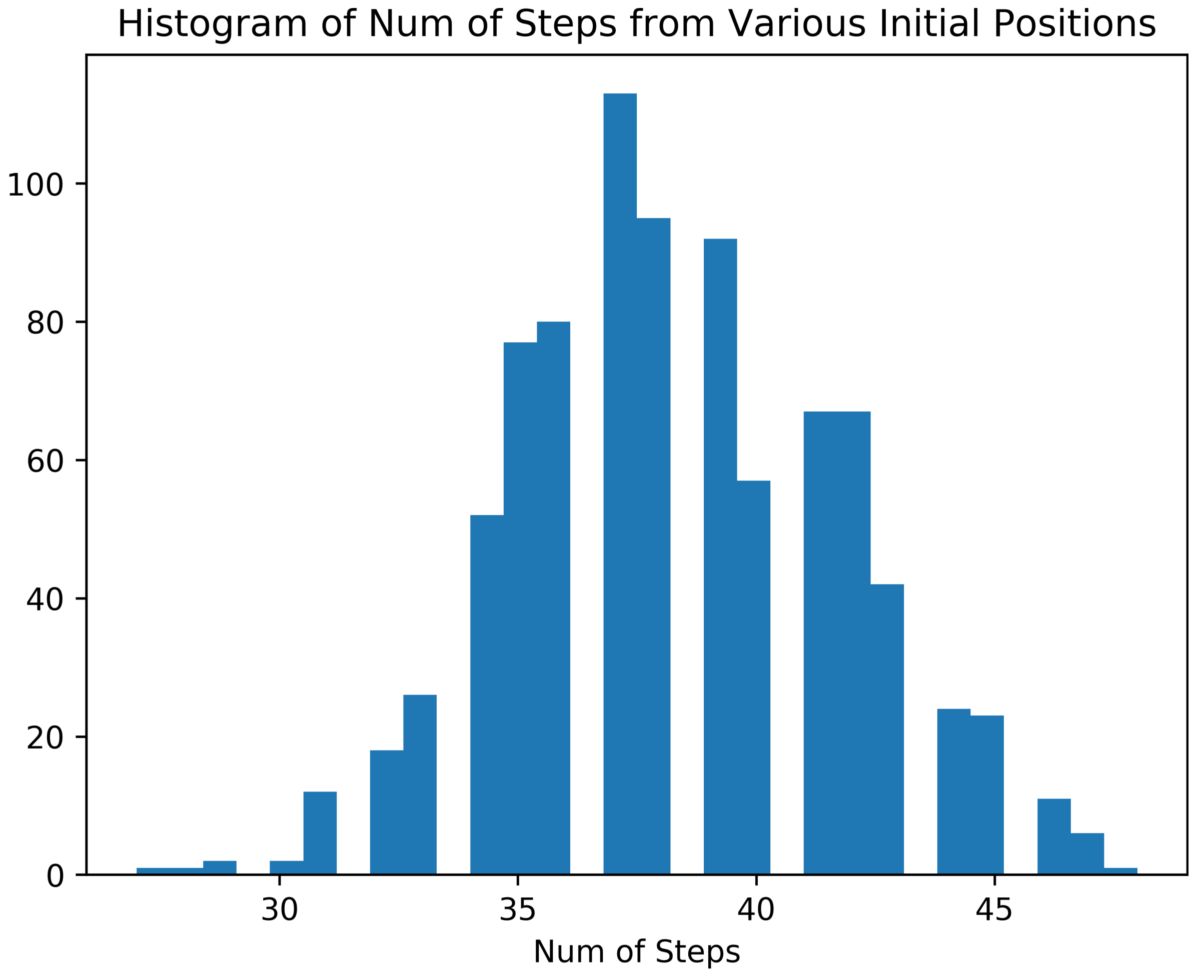

(S, ϕ)
Ψ
Cumulative Visibility
B
Frontiers
$\mathcal{G}$
Gain
$\arg\max \mathcal{G}$
Next Step

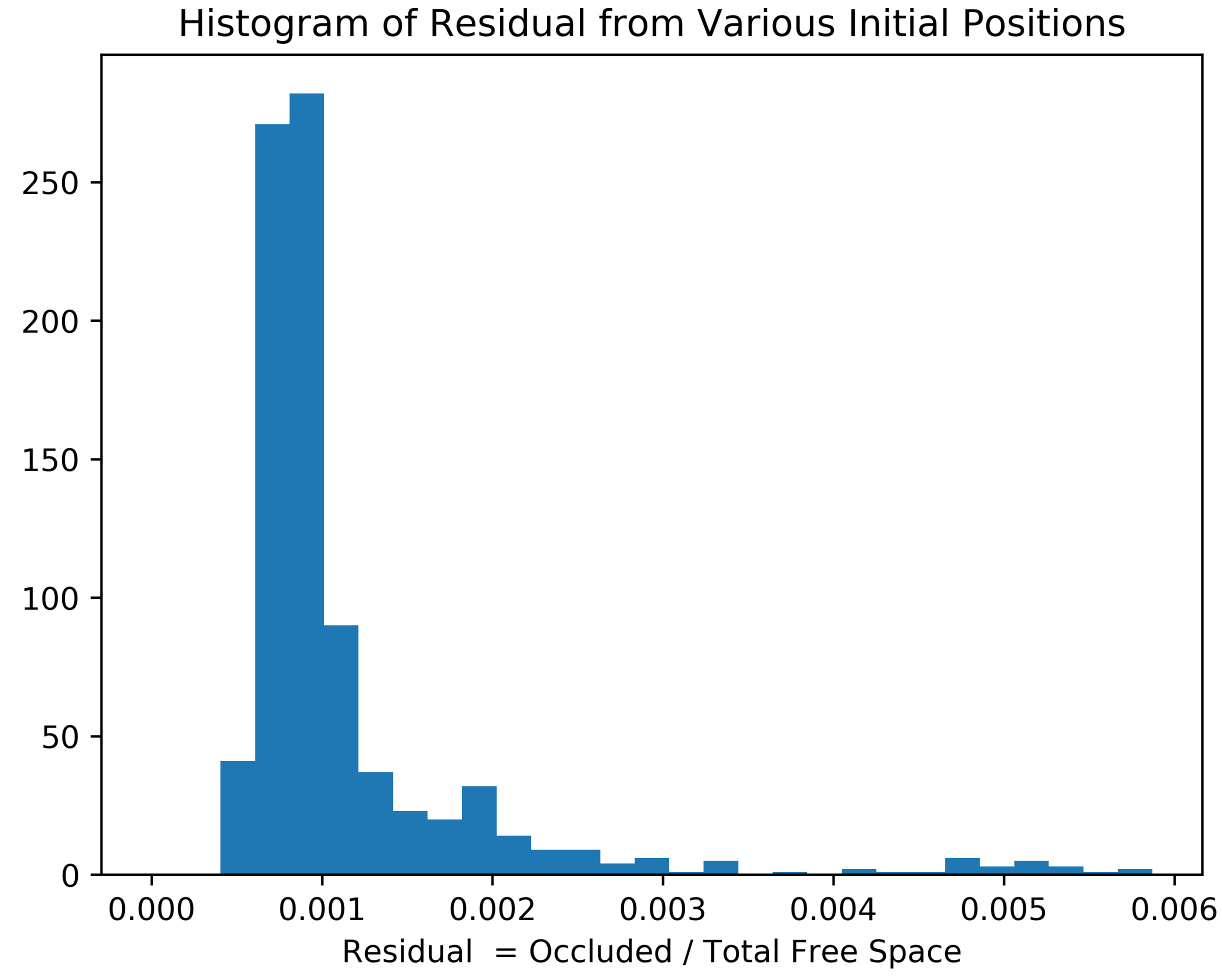

(S, ϕ)
Ψ
Cumulative Visibility
B
Frontiers
$\mathcal{G}$
Gain
$\arg\max \mathcal{G}$
Next Step

The network architecture



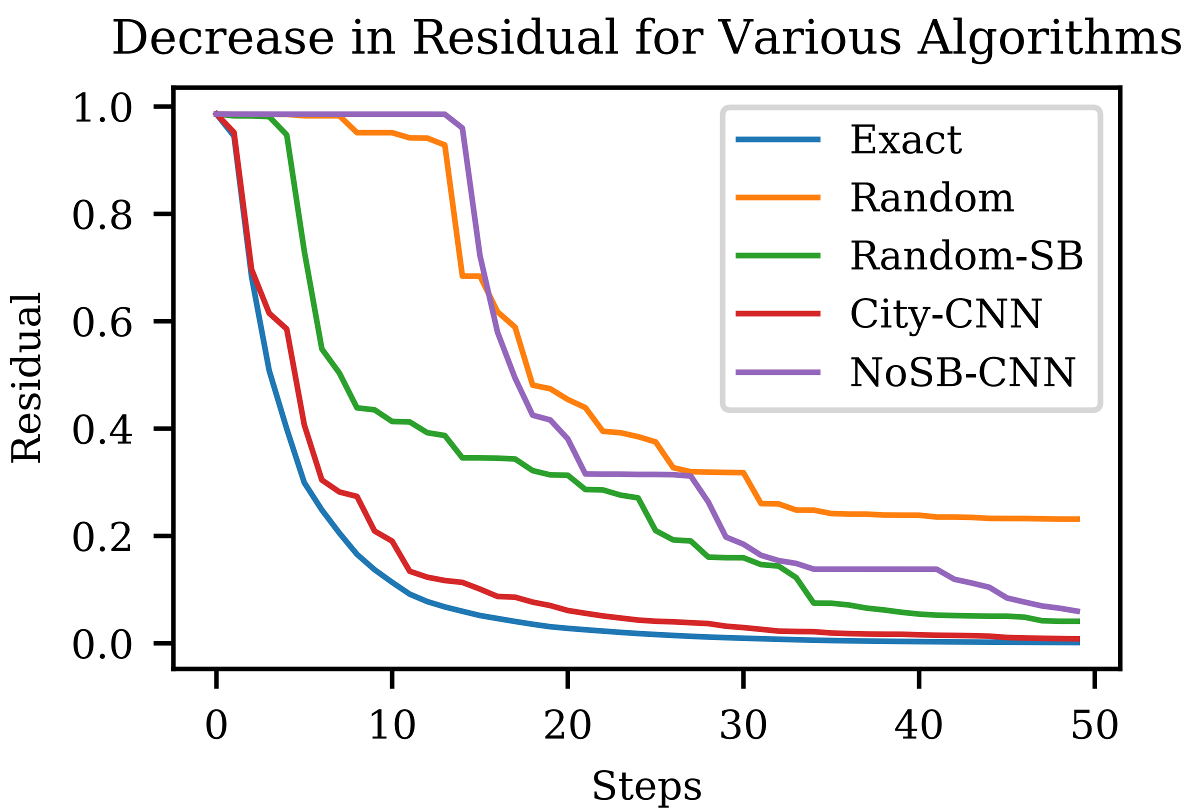
Performance statistics

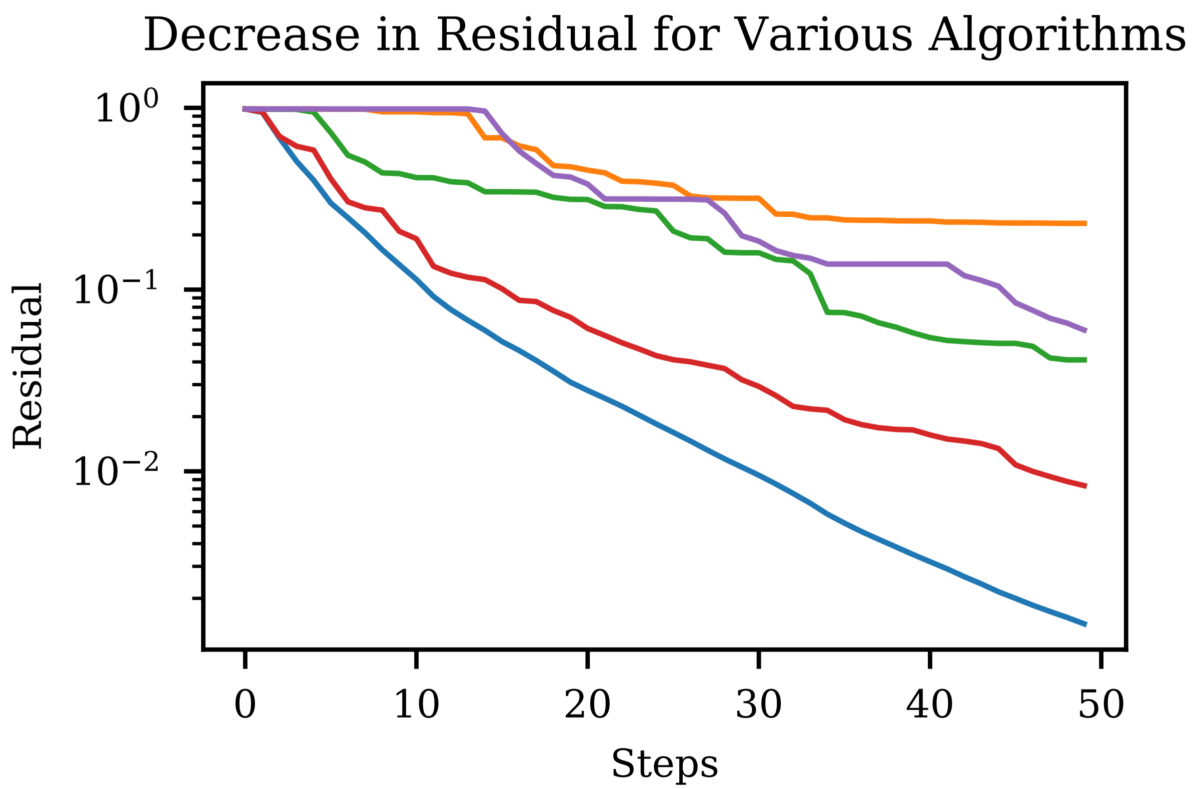




~39 buildings of different sizes

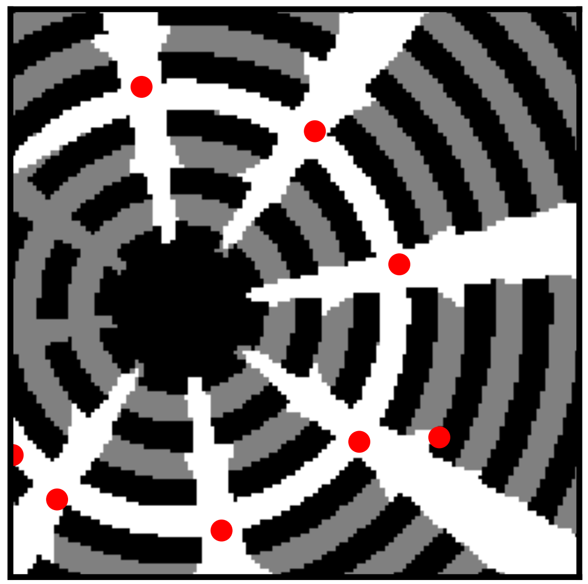
Performance



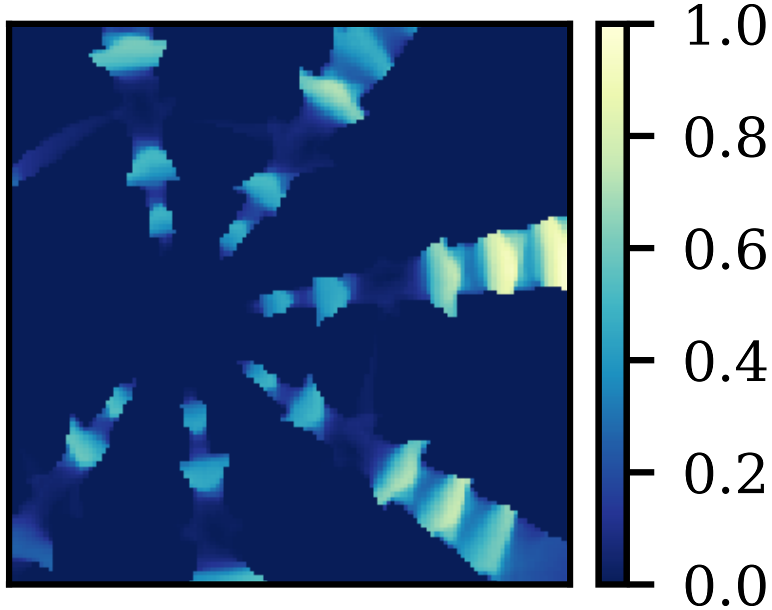


- Residual: Fraction of free space that is occluded
- CNN approach is comparable to Exact computation but significantly more efficient.
- 140 secs with 1024x1024 map on CPU vs 16+ hours
- Shadow boundary is essential for CNN (City-CNN vs NoSB-CNN)
- Random walk (Random) and randomly sampling shadow boundary (Random-SB)

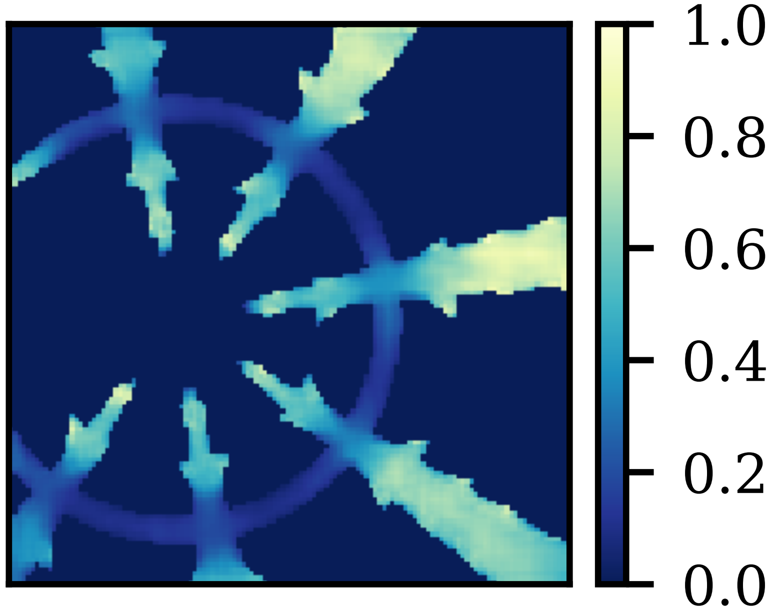
Different priors



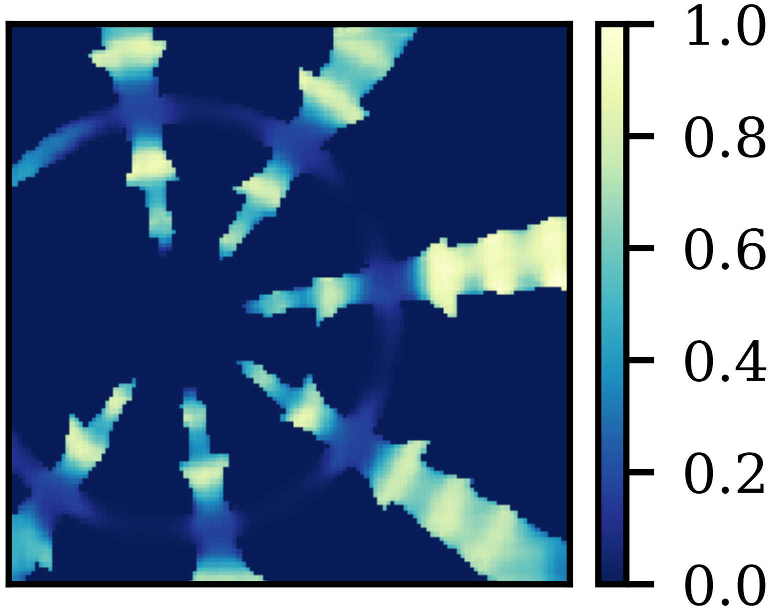
(a)



(b)



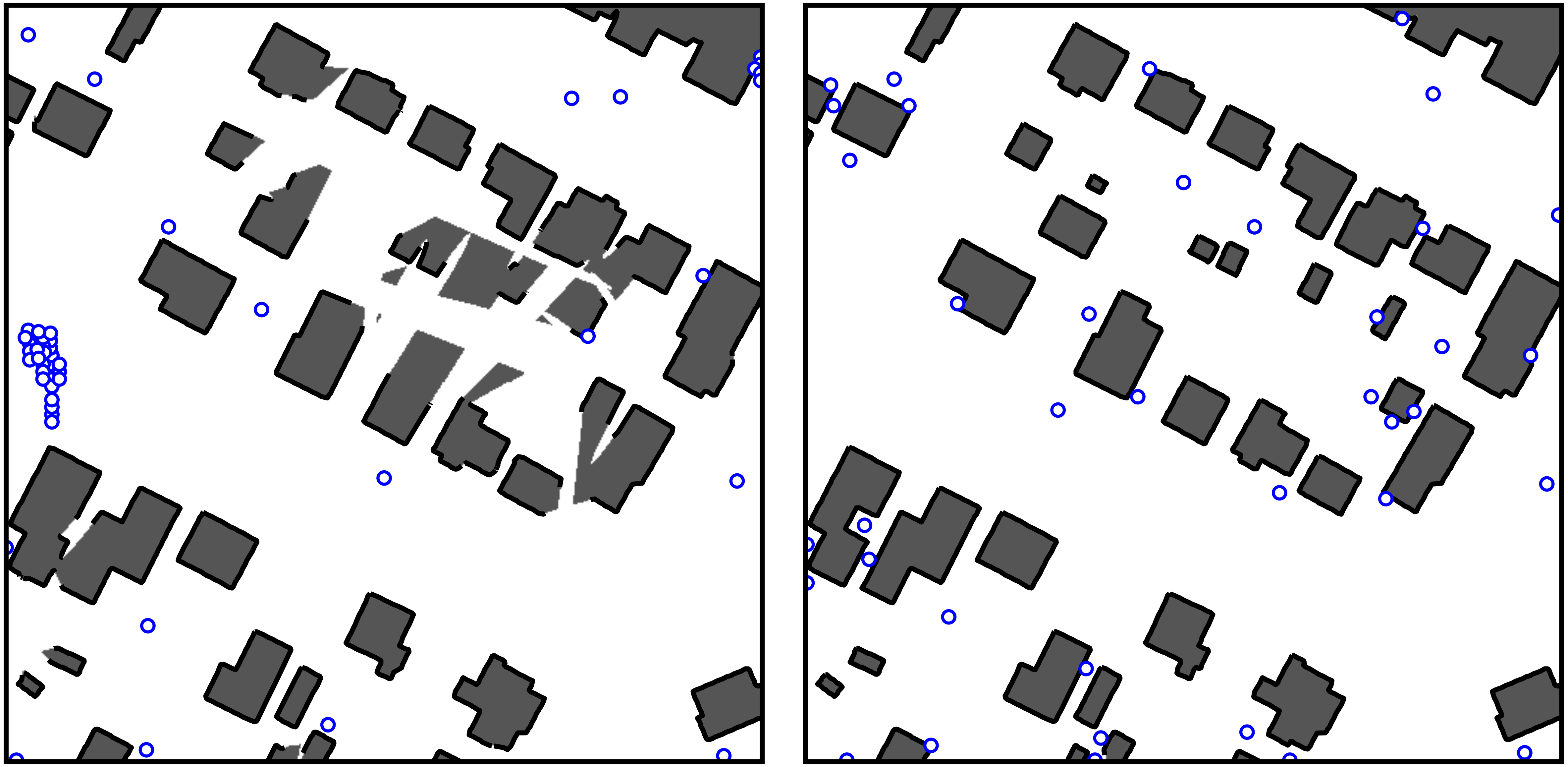
(c)



(d)

Improperly formulated approach

$$\mathcal{G}(\Psi, \cancel{B}) \in \mathbb{R}^{n \times n}$$

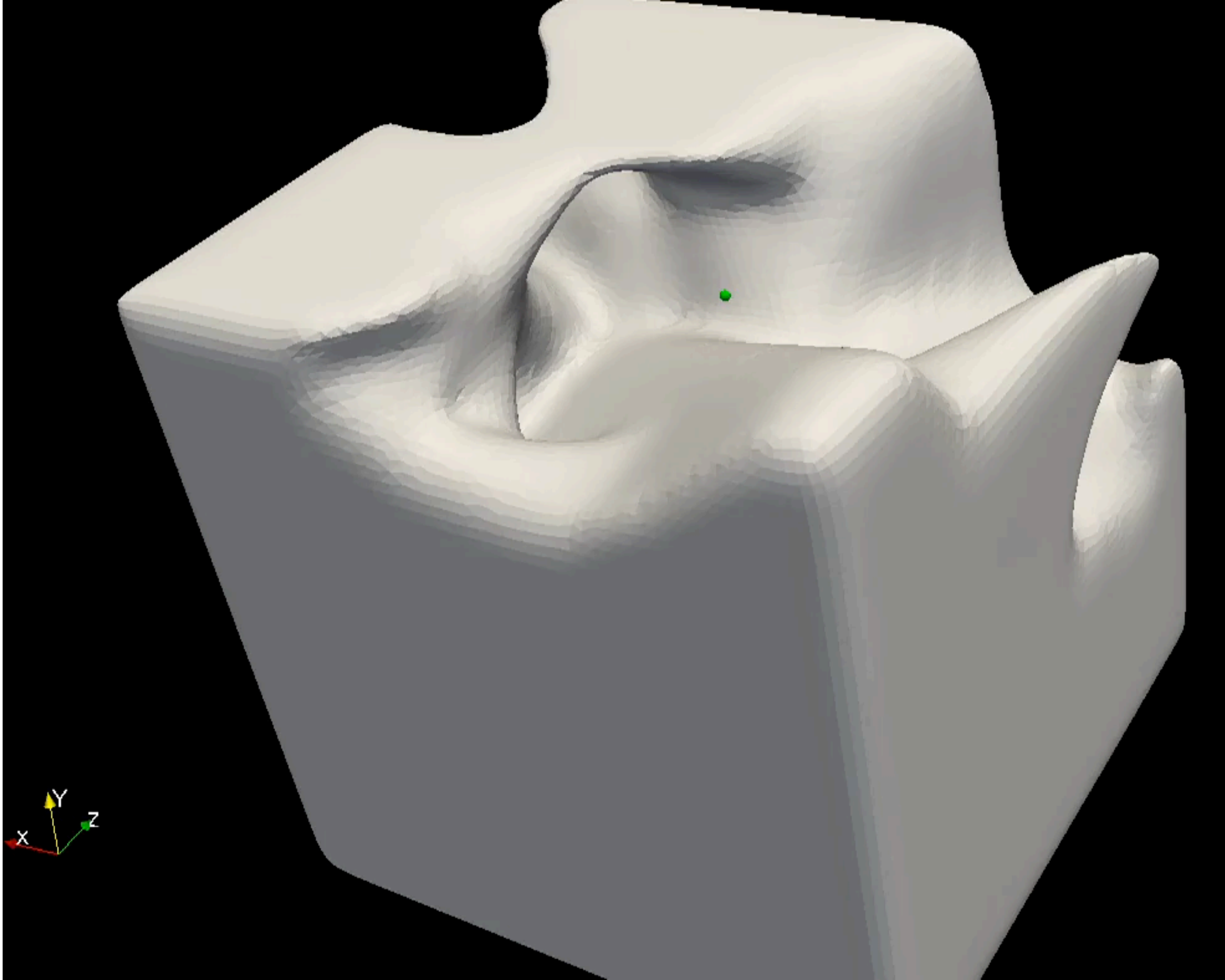


W/o shadow boundaries, CNN can’t distinguish b/w flat obstacles and unexplored regions.

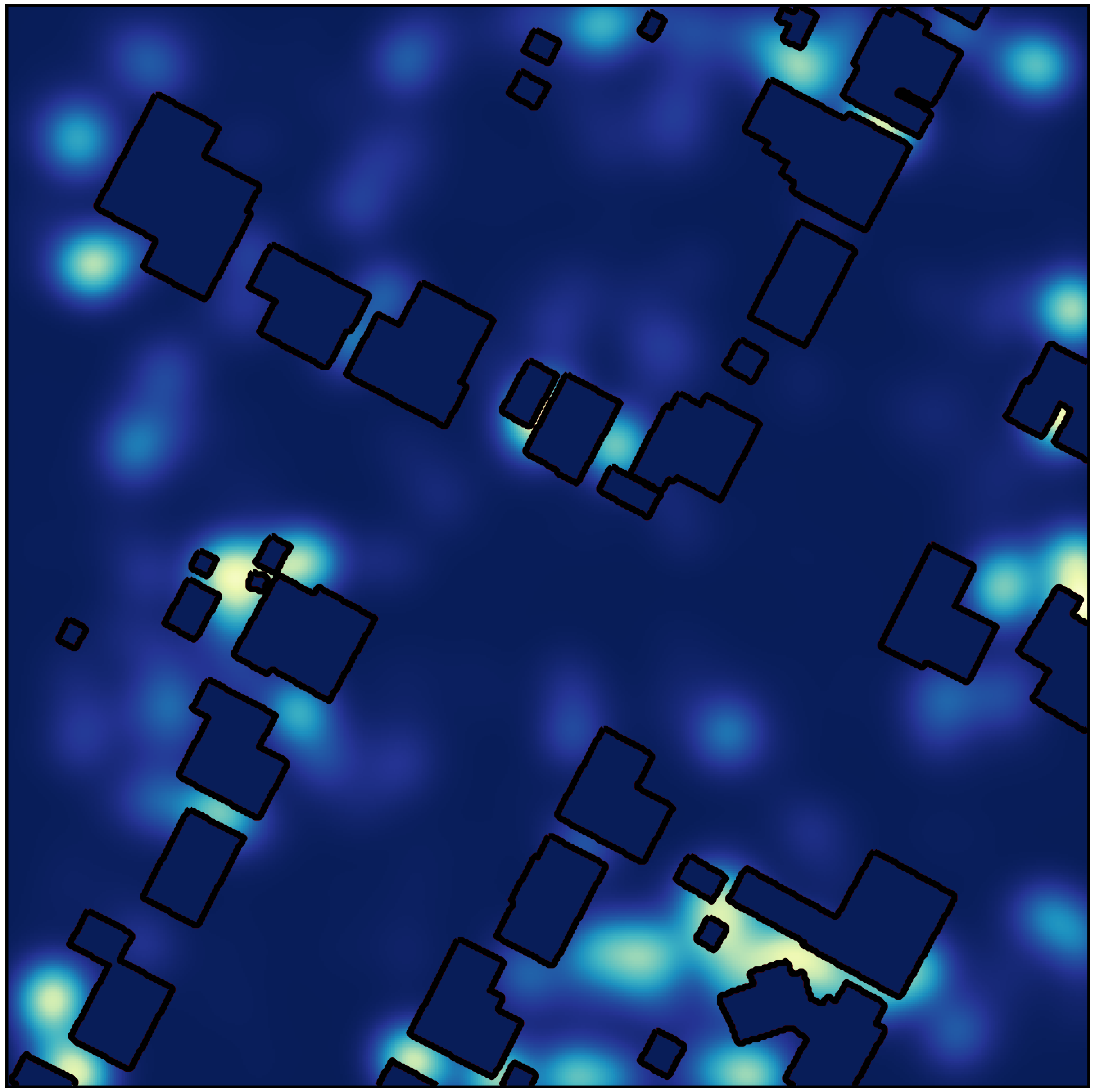
Autonomous Exploration, Reconstruction, and Surveillance of 3D Environments Aided By Deep Learning

Louis Ly and Yen-Hsi Richard Tsai

Oden Institute for Computational Engineering and Sciences, The University of Texas at Austin


Y
Z
X

“Importance” of observing locations

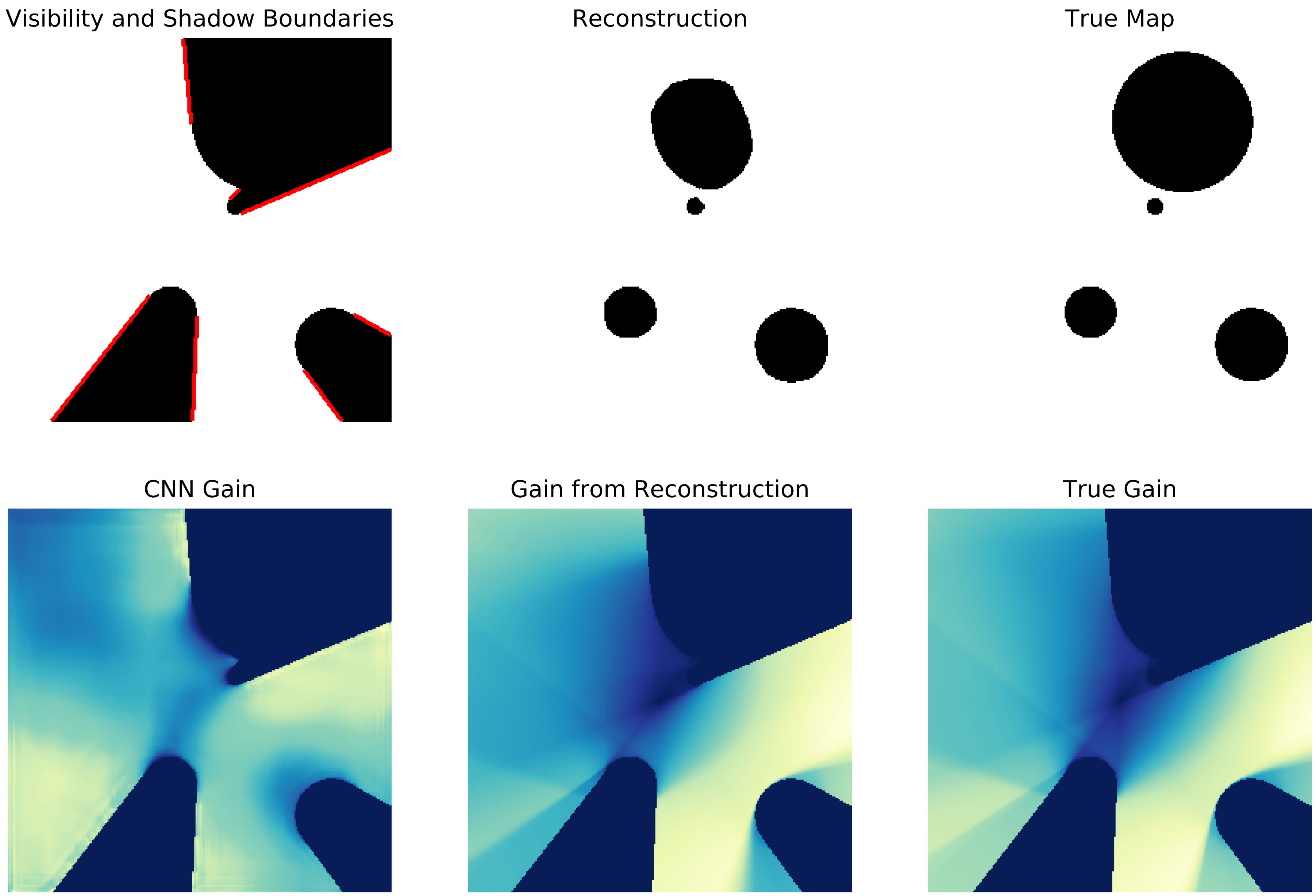


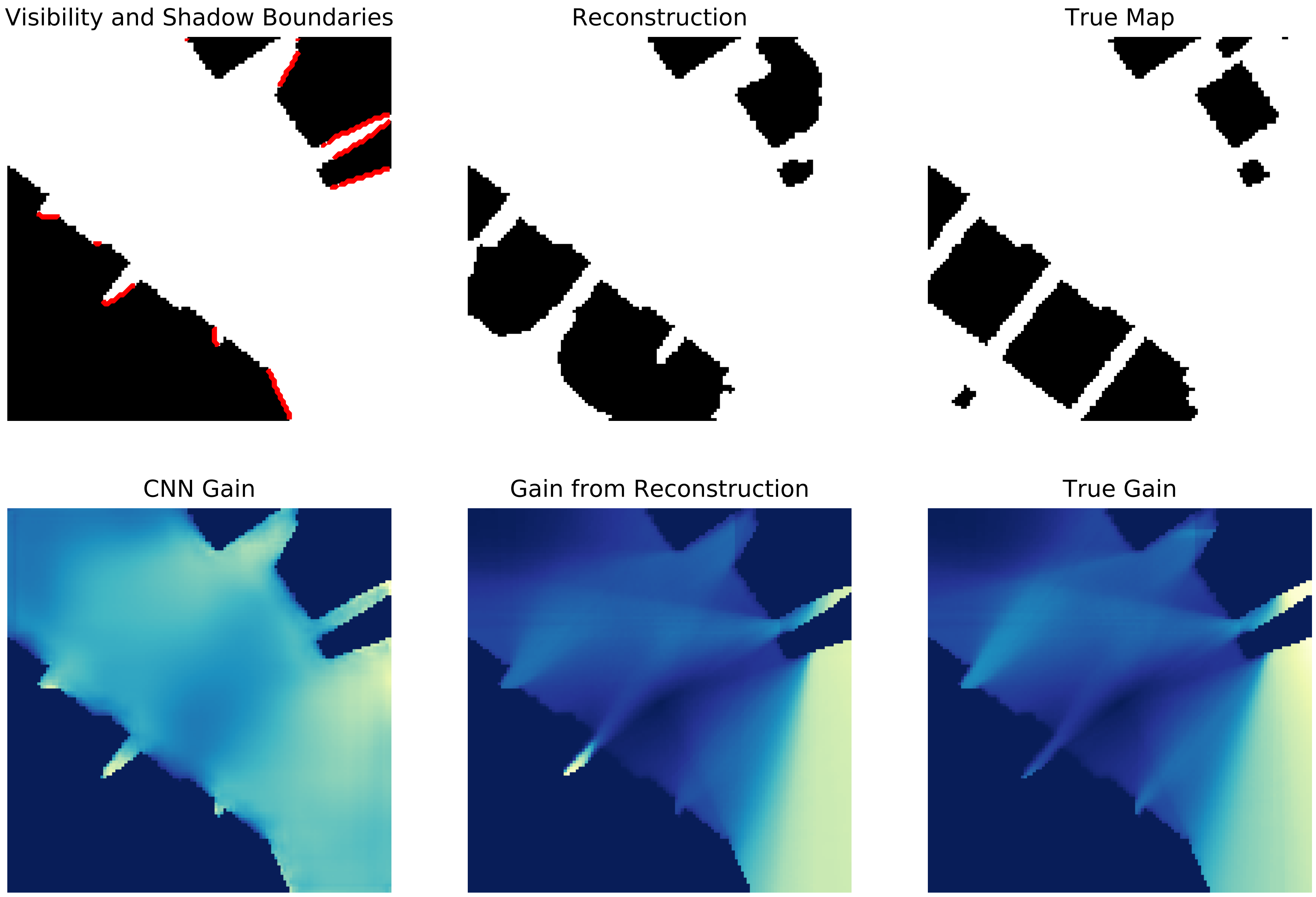
From N random initial locations, the brighter spots have more observing locations nearby than the dimmer.

It seems that some locations are more important than others

Prediction of the obstacles

An in-painting problem





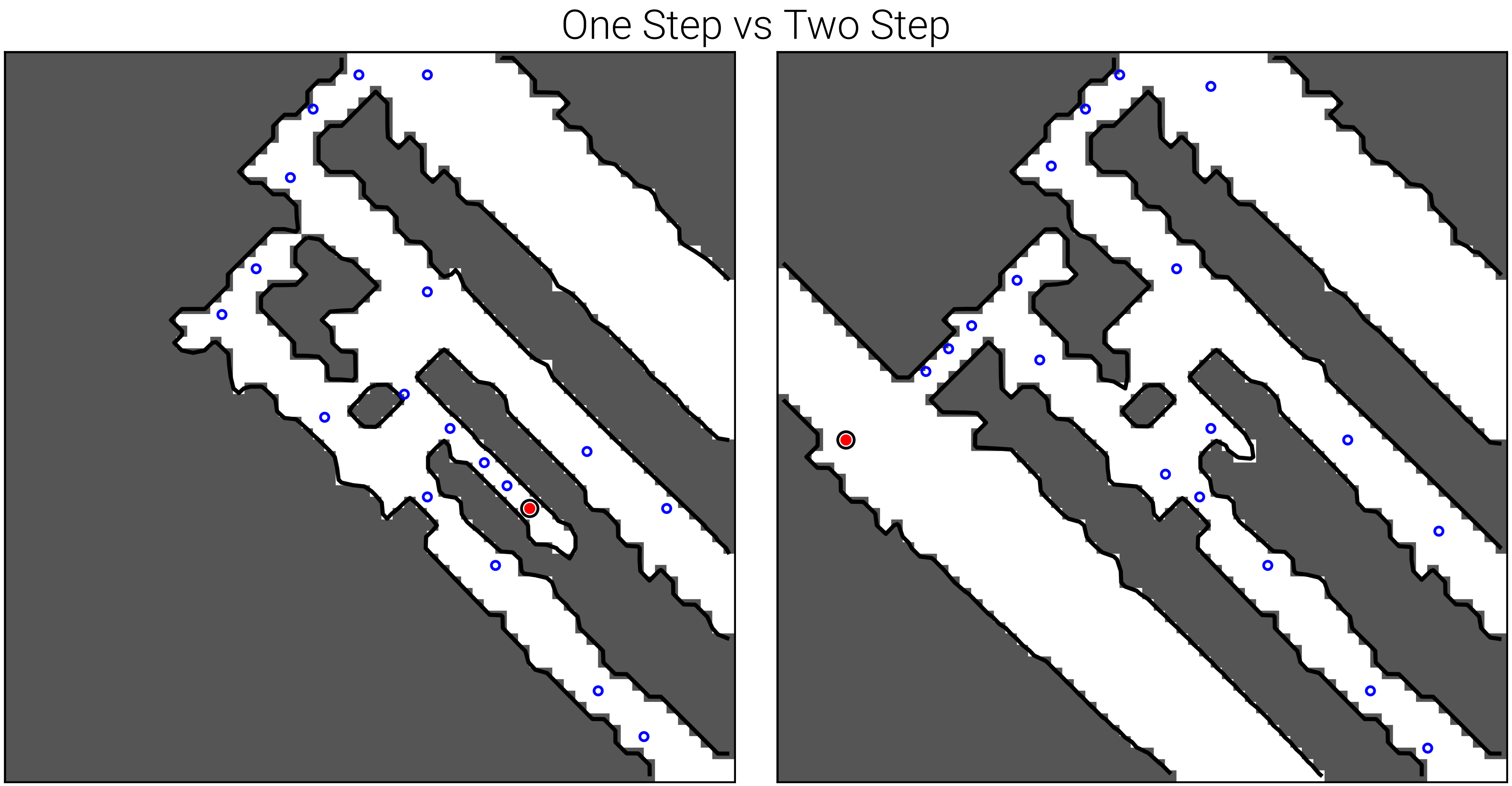
Different information is picked up by the “gain-net” and the “obstacle-net”.

How to best use the two pieces of information?

Summary

- Greedy approach using the “gain” function as reward
- Train on simulated data set
- Level set methods enable efficient data generation
- The CNN learns shape priors for a large class of obstacles
- On-going:
  - improvement via a GAN setup
  - multi-step greedy algorithms
  - surveillance-evasion games

Multi-step greedy algorithm



Will a multi-step greedy algorithm (always) yield more efficient learning of the environment?

$$\min_{I} ||I||_0 \qquad \text{subject to} \qquad \bigcup_{\{x_j | I(x_j)=1\}} \mathcal{V}_{x_j}\Omega = \Omega$$

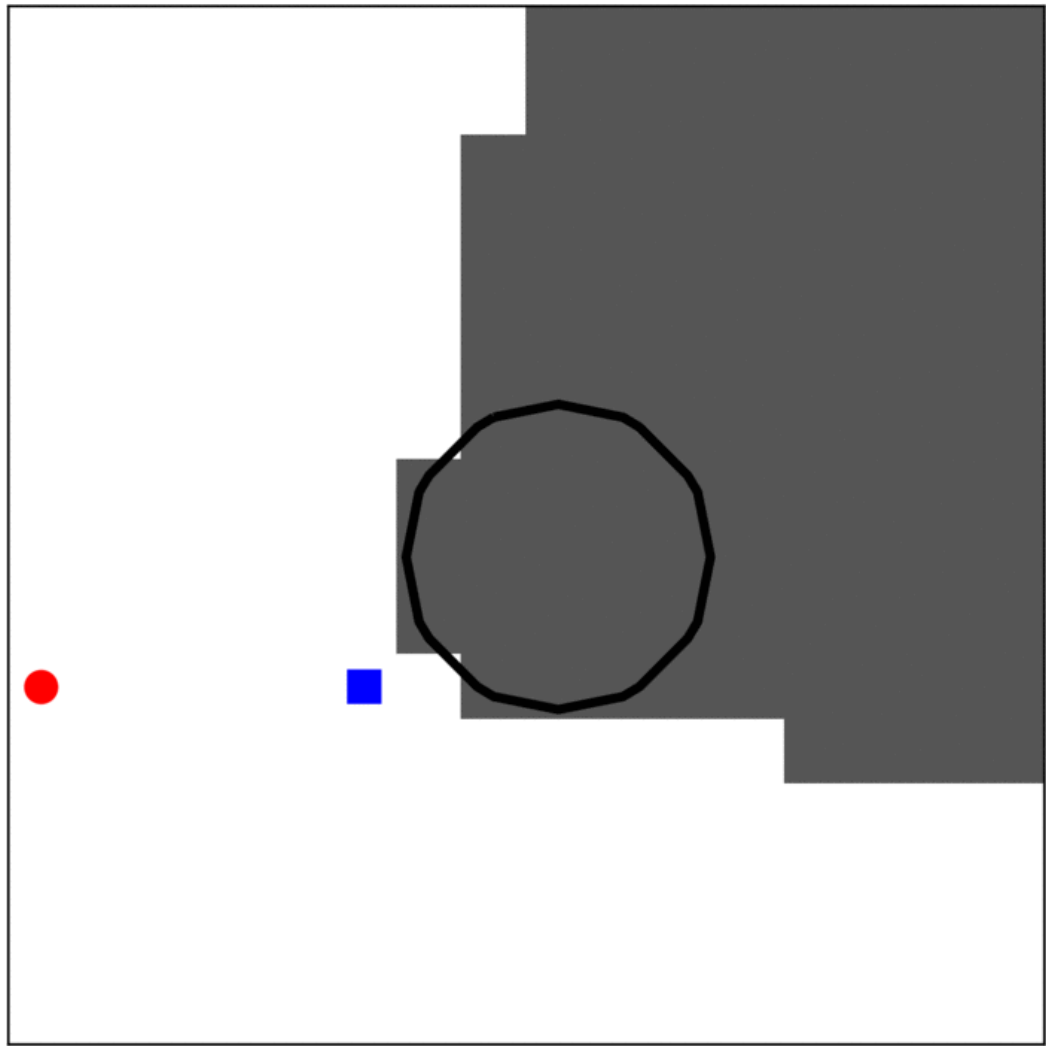
Multi-step greedy algorithm

- The previous way of training for the gain function is too costly for multiple steps.
- Natural to consider reinforcement learning type strategies
- We consider AlphaGo Zero style Monte-Carlo-Tree-Search algorithms

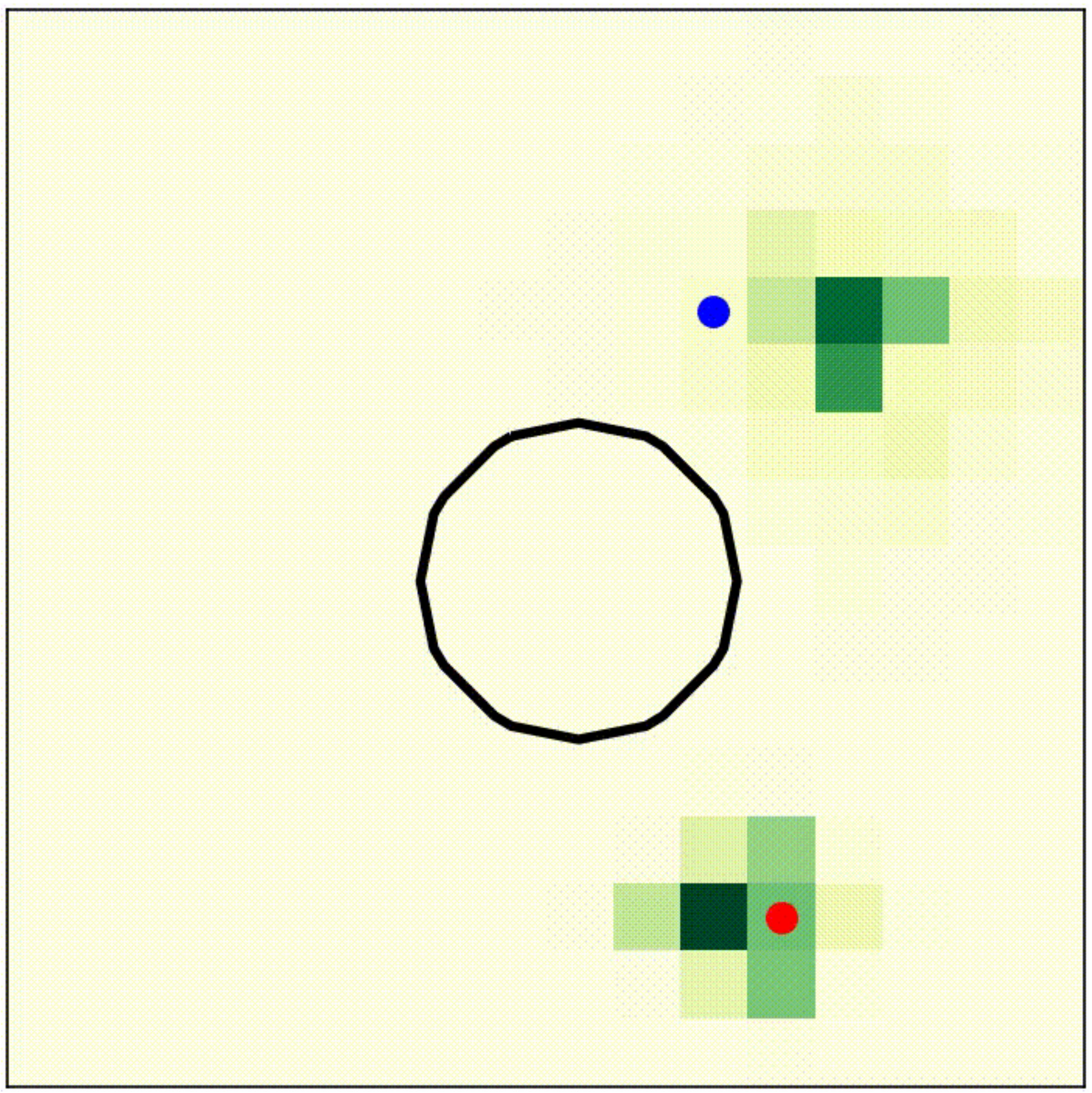
Learning to play the surveillance-evasion game

Pursuer must maintain the evader under line-of-sight.

Evader steered towards the nearest boundary of pursuer's visibility.



Value and policy networks



Policy network:
Predicts the probability of the next move

Value network:
Estimates how good the current state is

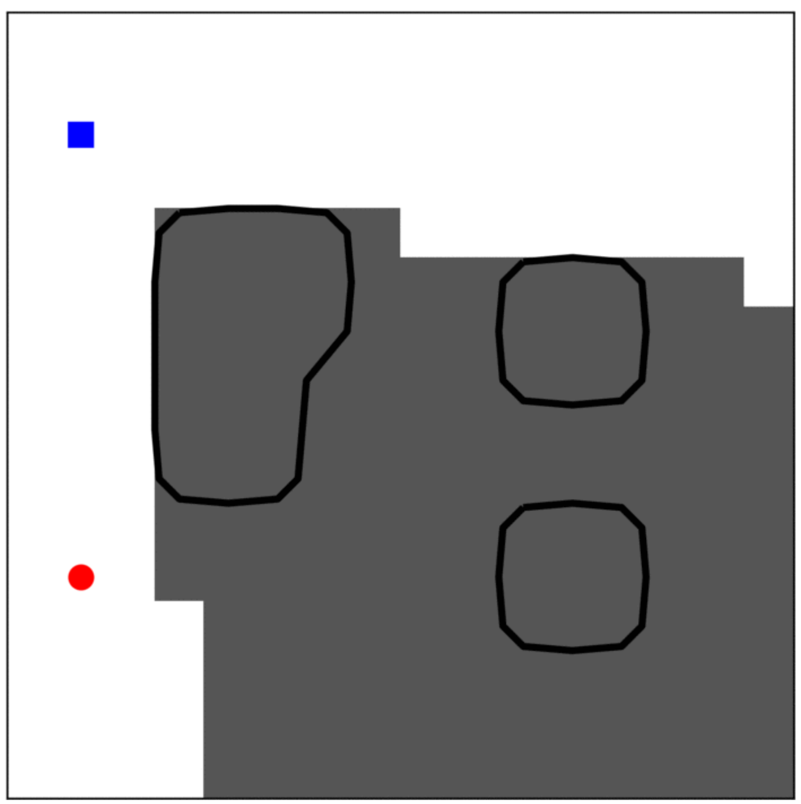
Both networks trained with various obstacles: reusable for different environments

Training procedure

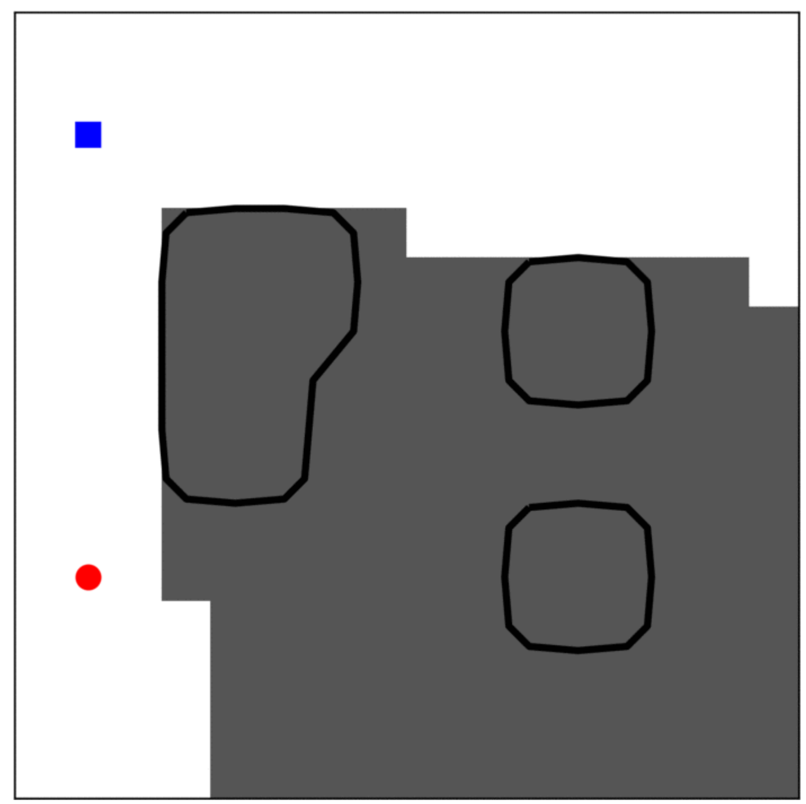
Learning to play the surveillance-evasion game

Pursuer must maintain the evader under line-of-sight.

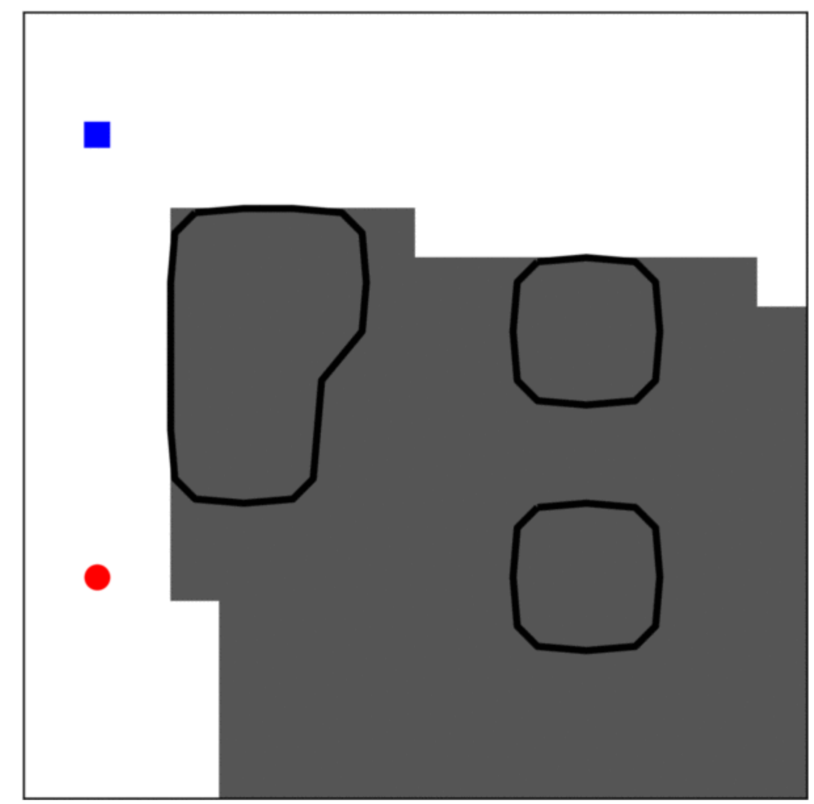
Evader steered towards the nearest boundary of pursuer's visibility.



A MCTS algorithm (1000 search steps)



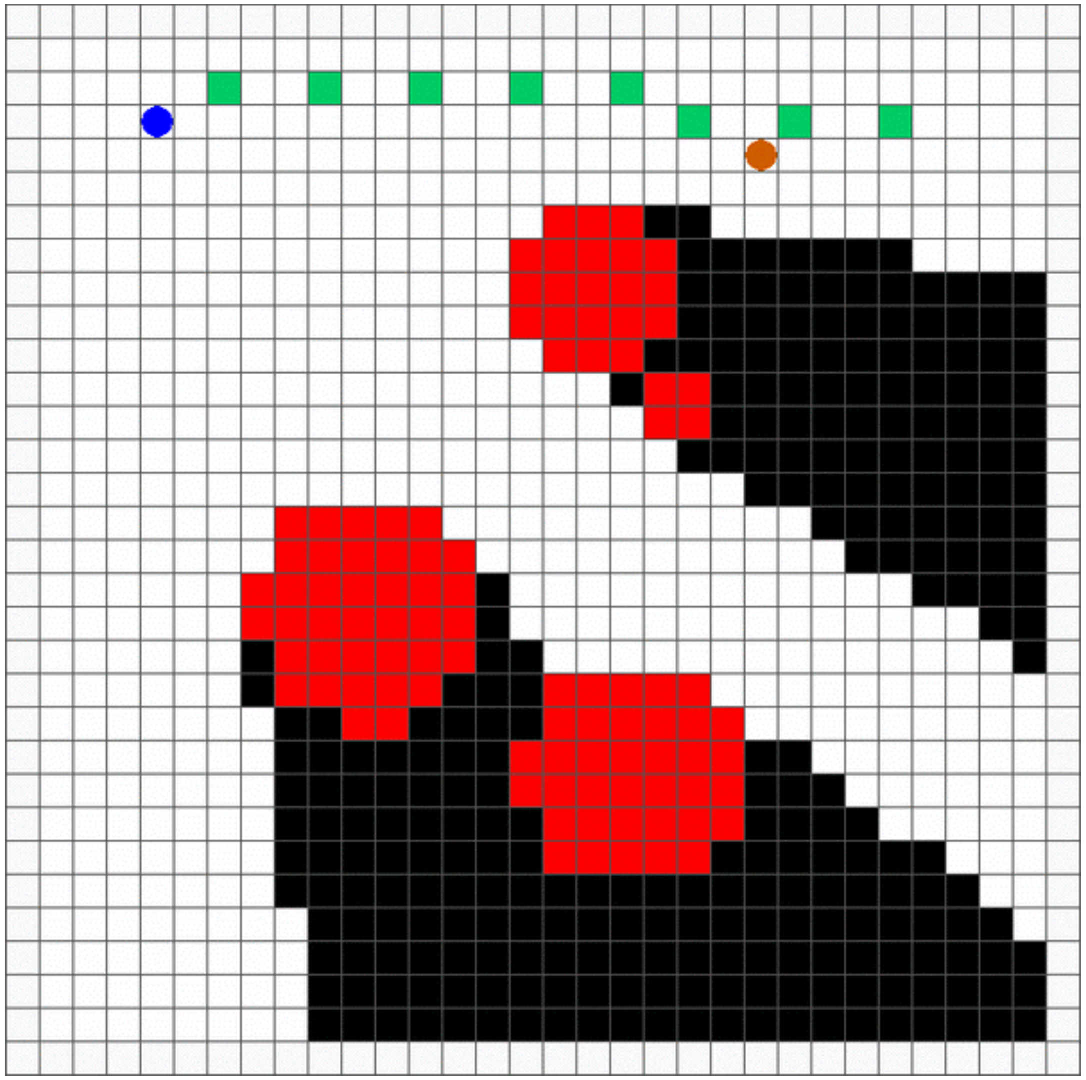
CNN-aided MCTS algorithm (1000 search steps)



CNN-aided MCTS algorithm (2000 search steps)

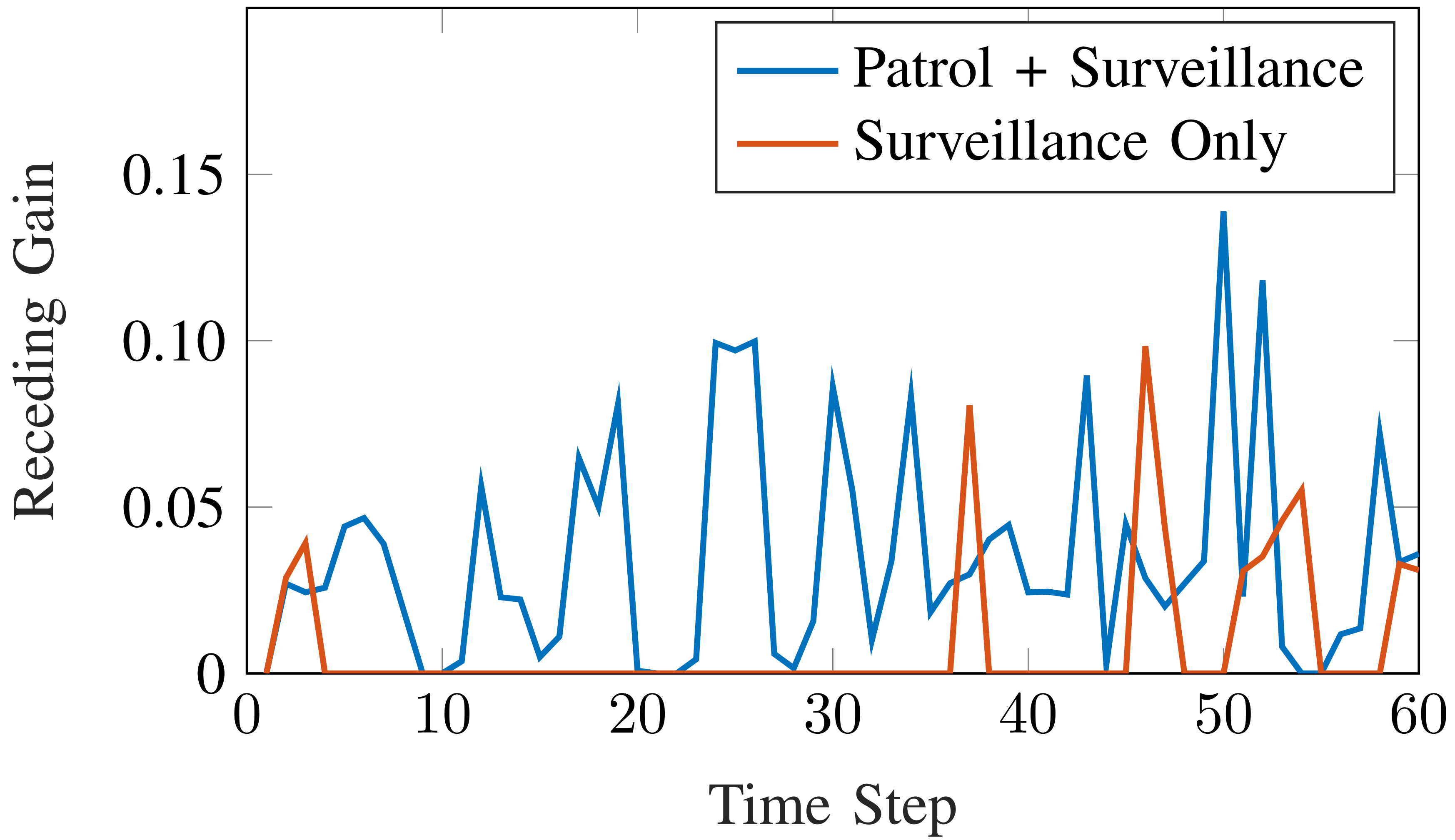
Games with multi-objectives and guarantees

- Dynamic game where agent has to always maintain visibility of a moving (adversarial) target.
- Secondary objective: maximize short-term surveillance of environment.
- Need to guarantee that current action does not jeopardize future.
- Use gain function to determine optimal vantage points.



[Ly, Bharadwaj, Wu, T, Topcu, 2019]

Performance of patrol



Receding gain: proportion of the newly observed states (int he past 5 time steps)
Relative to the total reachable states

Thanks for your attention!