Path Planning using Neural A* Search

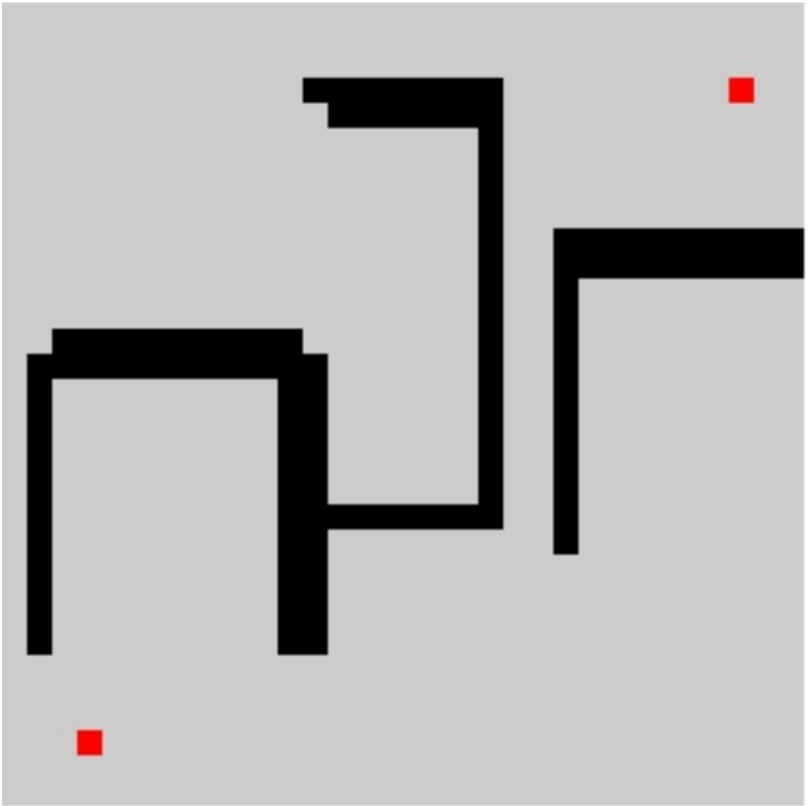
Ryo Yonetani[1*] Tatsunori Taniai[1*] Mohammadamin Barekatain[1] Mai Nishimura[1] Asako Kanezaki[2]

*Equal contribution

[1]OMRON SINIC X [2]Tokyo Institute of Technology

Problem

- Path planning
  - Finding a low-cost path from start to goal in an environment map
- Search-based planners
  - Guaranteed to find a solution path (if one exists) by incrementally and extensively exploring the map
  - E.g., A* search

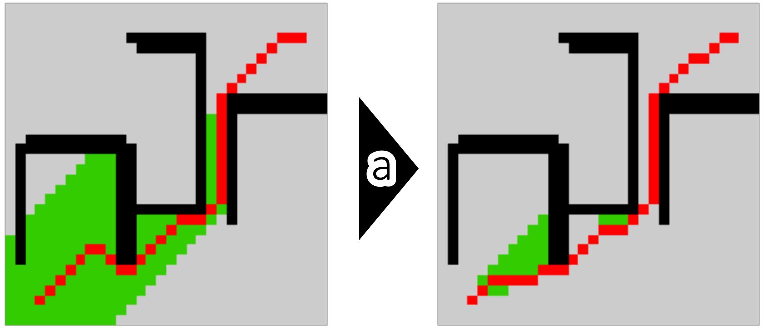


Data-driven planning

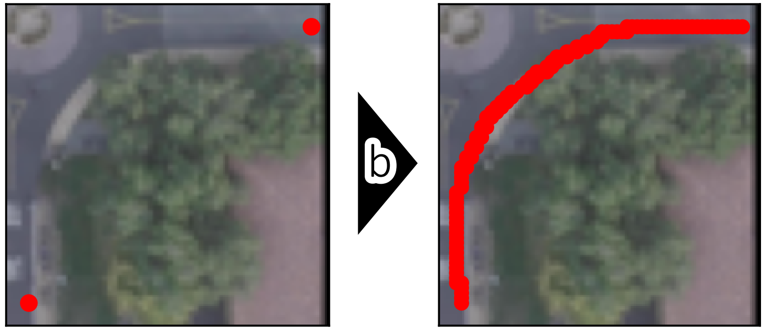
- Learning from demonstrations for:
  a. Improving planning efficiency [Choudhury+, 2018; Qurenshi+, 2019; Chen+, 2020]
  b. Enabling planning on raw image inputs [Tamar+, 2016; Lee+, 2018; Vlastelica+, 2020]

- Our goal:
  - Achieving these separately-studied objectives in a principled fashion
  - By data-driven search-based planners

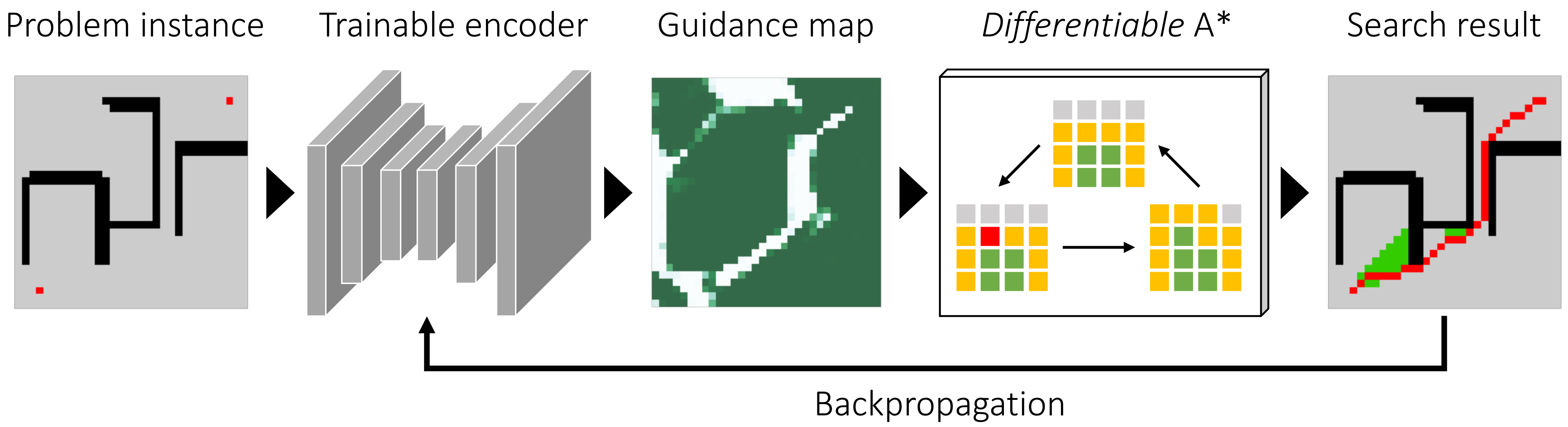
Faster & near-optimal



Planning without knowing actual costs



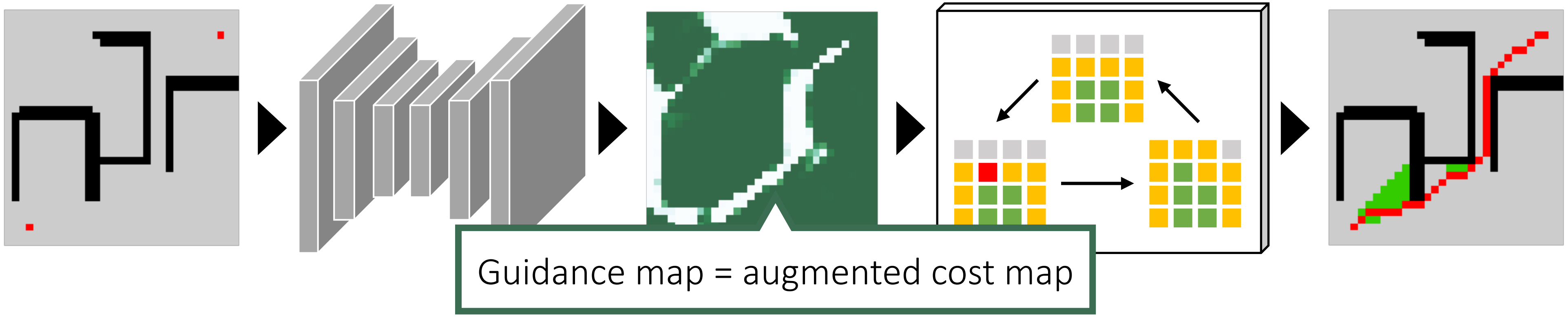
Neural A*



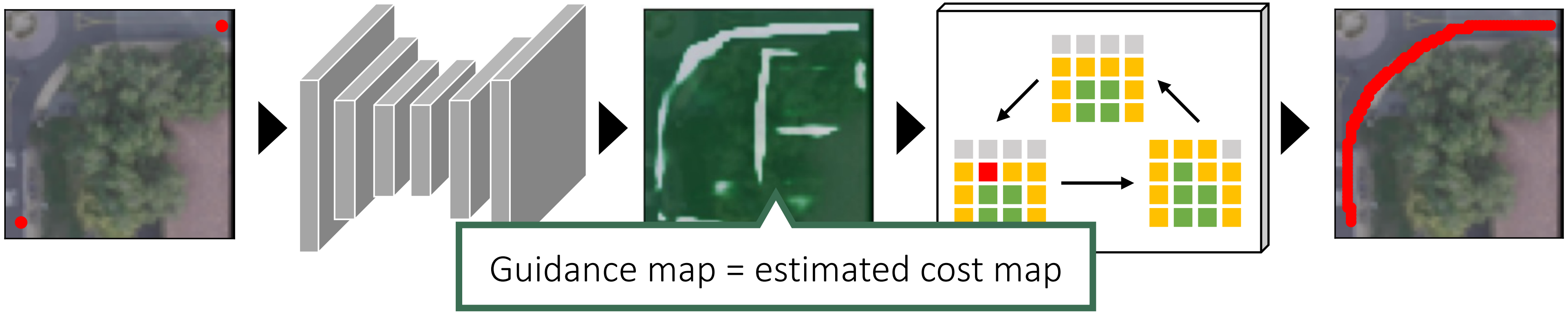
Problem instance
Trainable encoder
Guidance map
Differentiable A*
Search result
Backpropagation

Neural A*

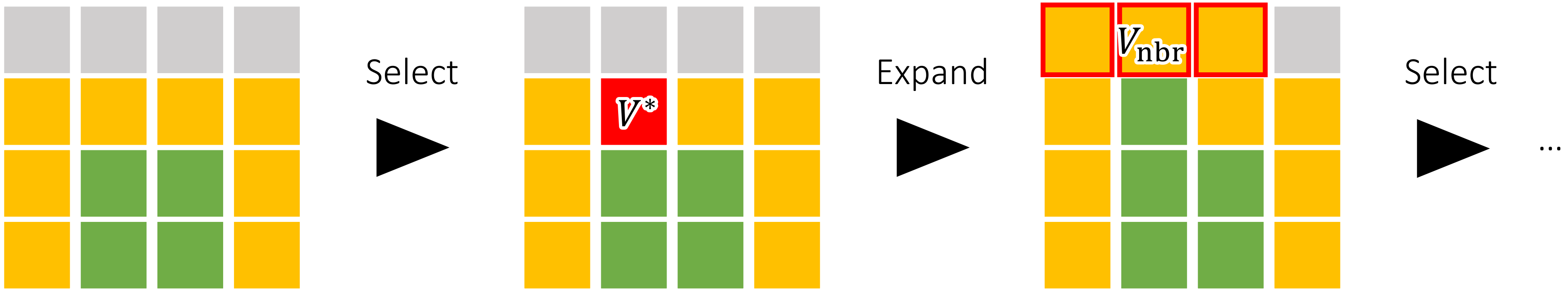
Shortest path search: augmenting cost maps for finding near-optimal paths efficiently



Planning on raw images: estimating cost maps for imitating ground-truth paths



Differentiable A* module



Node selection

- Finding nodes for constructing a shortest path
- Soft-max + discretized activation

$$V^* = \mathcal{I}_{\max}\left(\frac{\exp(-(G+H)/\tau)\odot O}{\langle \exp(-(G+H)/\tau), O\rangle}\right)$$

Total cost so far +
Estimated cost to go

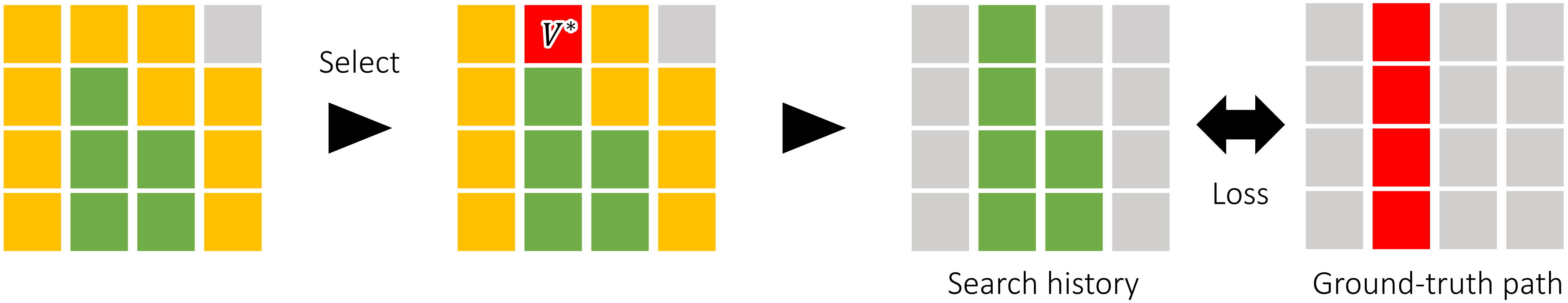
of nodes
in the open list

Node expansion

- Adding neighboring nodes to the list of next selection candidates
- Fixed convolution + binary masking

$$V_{\mathrm{nbr}} = \underbrace{(V^* * K)}_{\text{Neighbor}} \odot \underbrace{X}_{\text{Obstacle}} \odot \underbrace{(\mathbb{1} - O)}_{\text{Not opend}} \odot \underbrace{(\mathbb{1} - C)}_{\text{Nor selected}}$$

Differentiable A* module



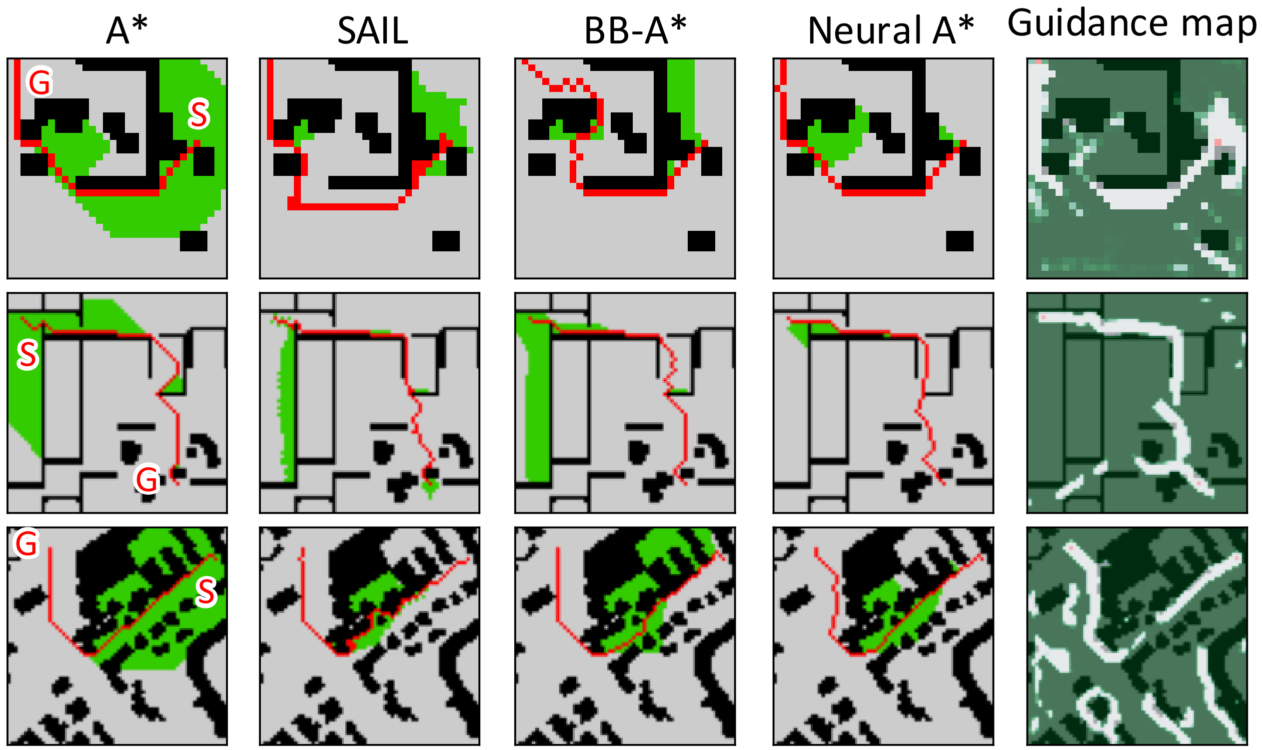
Loss: L1 between search history and ground-truth path

- Backpropagated thorugh every search step to the encoder
- Making guidance maps to indicate nodes to/not to explore

$$\mathcal{L} = \|C - \bar{P}\|_1 / |\mathcal{V}|.$$

Search results   Ground truth   # nodes

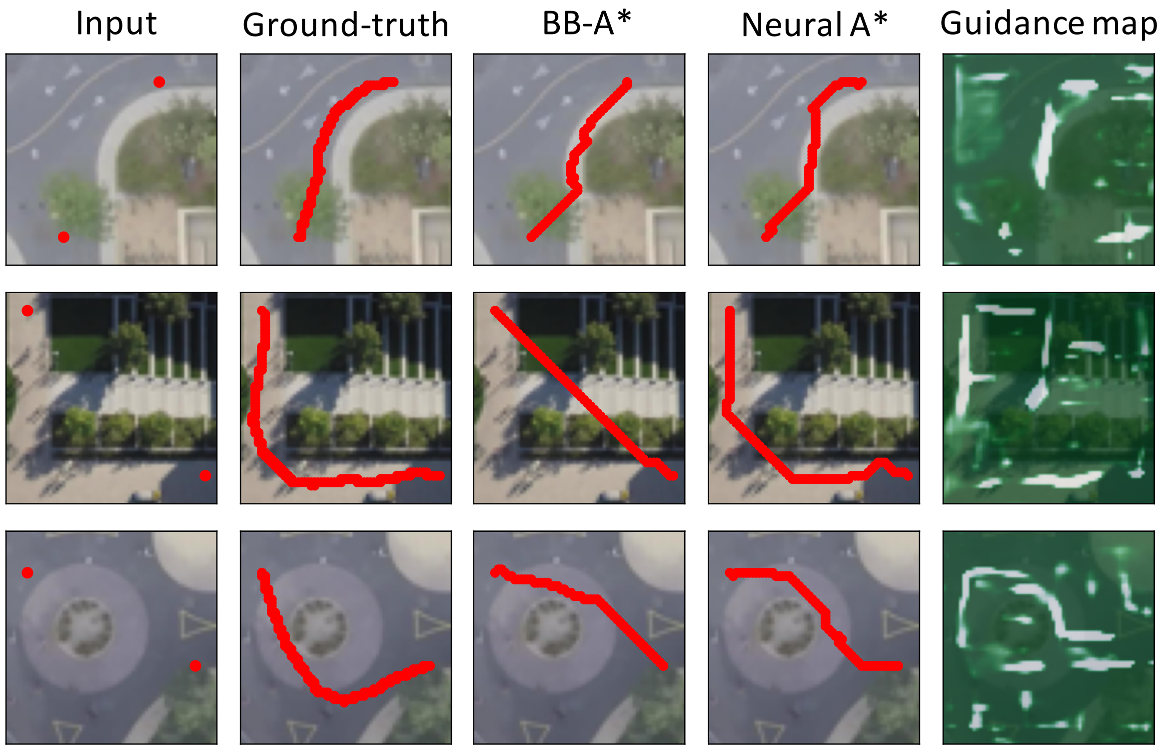
Point-to-point shortest path search



* The higher the better

	(a) Search optimality	(b) Search efficiency	Hmean of (a) and (b)
SAIL [Choudhury+, 2018]	34.6	48.6	26.3
BB-A* [Vlastelica+, 2020]	62.7	42.0	42.1
Neural A*	**87.7**	40.1	**52.0**

Path planning on raw image inputs



Experimental setup

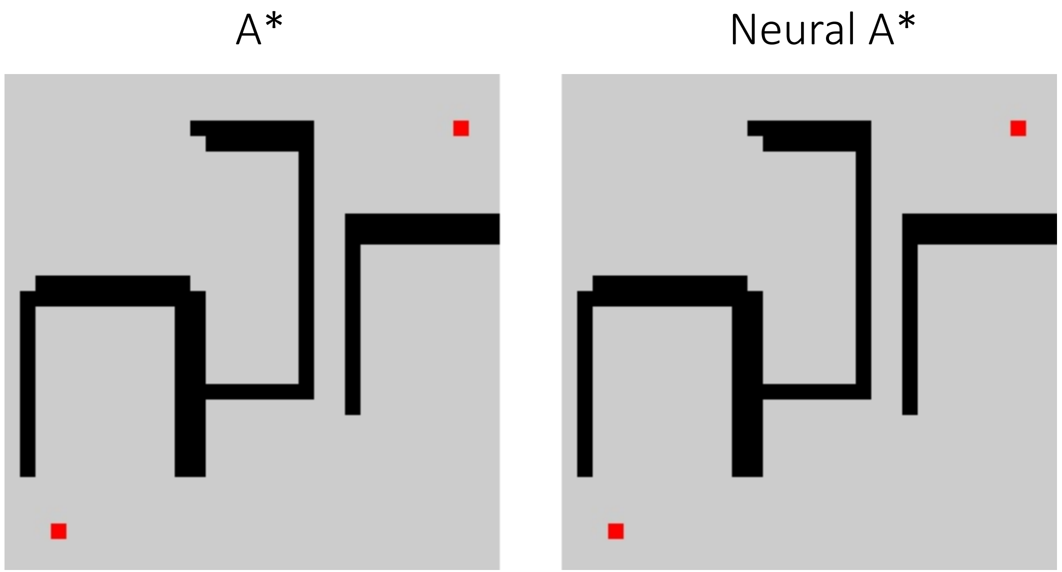
- Surveillance images + pedestrian trajectories as demonstrations
- Task: Predicting realistic trajectories consistent with those of pedestrians when start and goal locations are provided

	BB-A* [Vlastelica+, 2020]	Neural A*
Chamfer distance (the lower the better)	152.2	**16.1**

Summary

Neural A* = Trainable encoder + differentiable A* search

✓ Improving search optimality-efficiency trade-off ✓ Enabling path planning on raw image inputs



Project page:

https://omron-sinicx.github.io/neural-astar/

*Code and data available soon

Contact:

Ryo Yonetani | ryo.yonetani@sinicx.com

Interested in research internship? | internships@sinicx.com