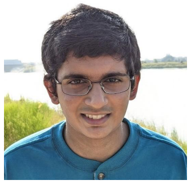
# From Agile Ground to Aerial Navigation: Learning from Learned Hallucination (LfLH)

Zizhao Wang[1], Xuesu Xiao[2], Alexander J Nettekoven[3], Kadhiravan Umasankar[4], Anika Singh[1], Sriram Bommakanti[4], Ufuk Topcu[4], and Peter Stone[2,5]





[1]ECE, [2]CS, [3]Mechanical Engr, [4]Aerospace Engr & Engr Mechanics, The University of Texas at Austin
[5]Sony AI

# Background

Navigation in **highly-constrained** environments.

Xiao, et al., *Toward Agile Maneuvers in Highly Constrained Spaces: Learning from Hallucination*. RA-L 21

# Background

Navigation in **highly-constrained** environments.

- Classical methods require **increased computation**.
  - Sampling-based methods require more samples to find feasible motion. [Kavraki, et al., TRA96, Fox, et al., RAM97, LaValle, TechReport98]
  - Optimization-based methods require more optimization iterations. [Quinlan, et al., 93, Zucker, et al., IJRR13, Zhou, et al,. RA-L21]

# Background

Navigation in **highly-constrained** environments.

- Classical methods require **increased computation**.
  - Sampling-based methods require more samples to find feasible motion. [Kavraki, et al., TRA96, Fox, et al., RAM97, LaValle, TechReport98]
  - Optimization-based methods require more optimization iterations. [Quinlan, et al., 93, Zucker, et al., IJRR13, Zhou, et al,. RA-L21]
- Learning methods are fast but require **good-quality training data**.
  - Imitation learning: demonstrations are hard to acquire. [Pfeiffer, et al., ICRA17, Tai, et al., IROS16]
  - Reinforcement learning: trial-and-error is dangerous [Tai, et al., IROS17, Chiang, et al., RA-L19]

# Background

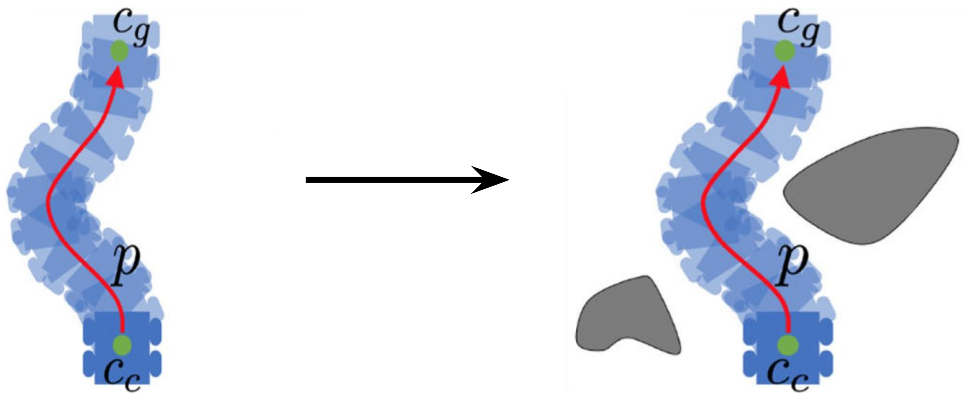
## Inspiration

It’s safe for the robot to perform agile maneuvers in open space, which can be optimal for certain highly-constrained environments.

Can we hallucinate obstacles that make those maneuvers **optimal**?

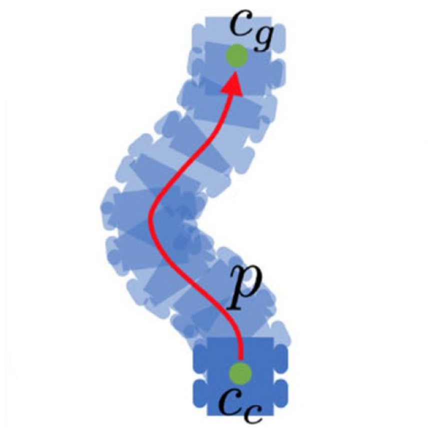
If so, open space motion plans become **cheap training data** for learning methods.

$c_g$: goal configuration
$c_c$: current configuration
$p$ : motion plan

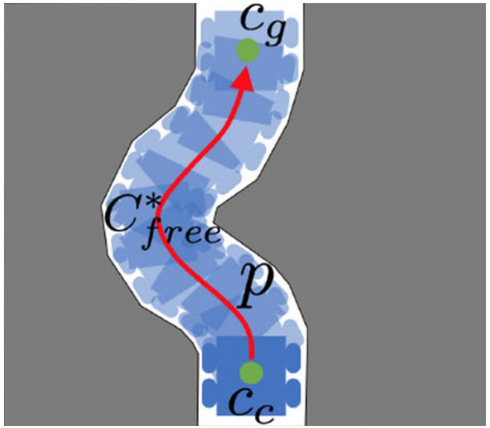


# Background

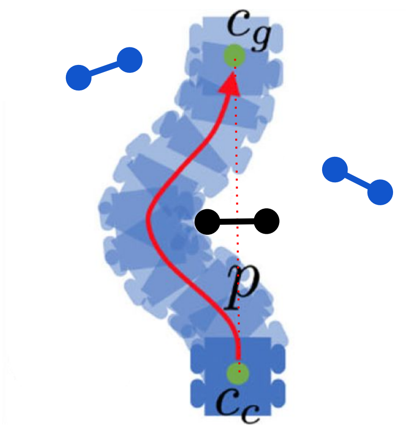
## Learning from Hallucination




$c_g$: goal configuration
$c_c$: current configuration
$p$ : motion plan




Most Constrained Obstacle
(Xiao, et al., RA-L 21)

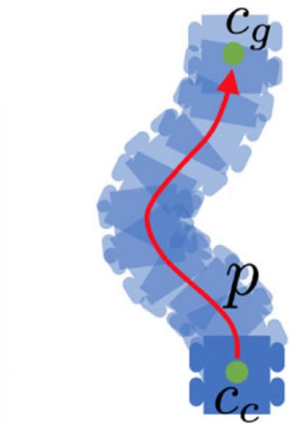


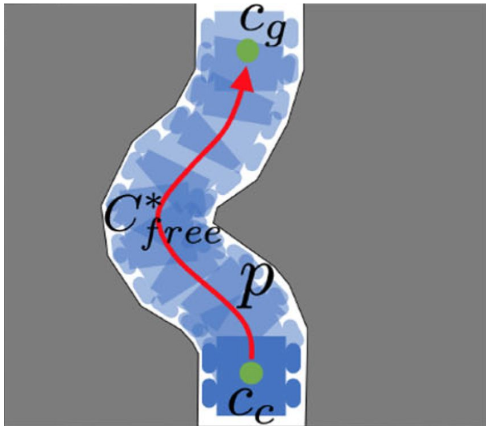

Minimal Obstacle Set +
Additional Obstacles
(Xiao, et al., ICRA 21)

# Learning from **Learned** Hallucination (LfLH)

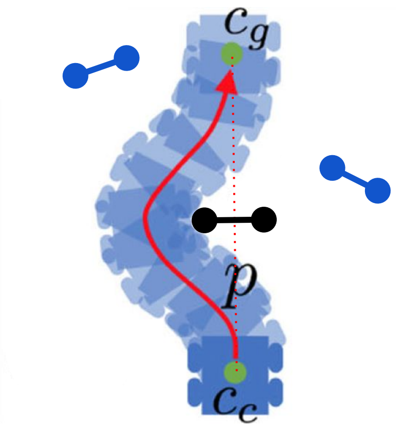
## Motivation & Contribution

- Previous methods use hand-crafted hallucination techniques.
  - Laboriously designed for specific robots (takes expert several weeks through lots of tuning iterations)
  - Only works for a short planning horizon (1m).




Most Constrained Obstacle
(Xiao, et al., RA-L 21)



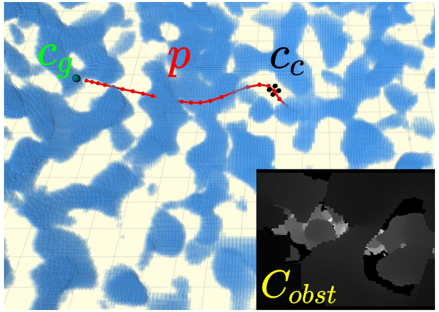

Minimal Obstacle Set + Additional Obstacles
(Xiao, et al., ICRA 21)

# Learning from **Learned** Hallucination (LfLH)

## Motivation & Contribution

- Previous methods use hand-crafted hallucination techniques.
  - Laboriously designed for specific robots (takes expert several weeks through lots of tuning iterations)
  - Only works for a short planning horizon (1m).
- LfLH uses self-supervised learning to hallucinate obstacles
  - Works with any robot type or planning horizon.

2D navigation
planning horizon: 45 m



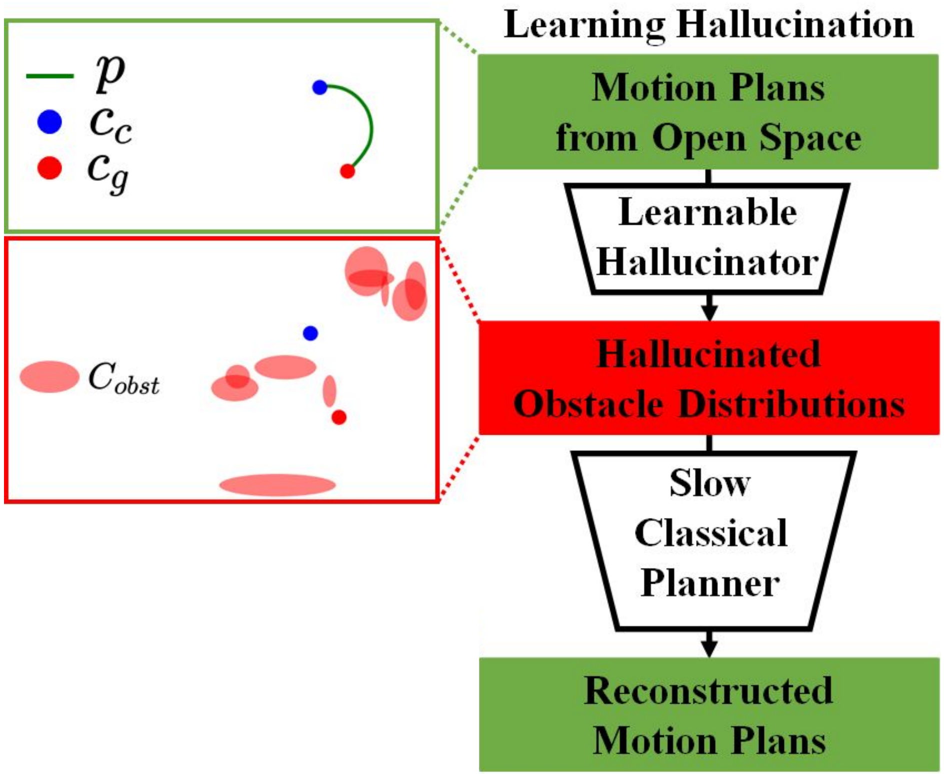

3D navigation
planning horizon: 75 m

# Learning from Learned Hallucination

**Method - Learning Hallucination**

Hallucinator

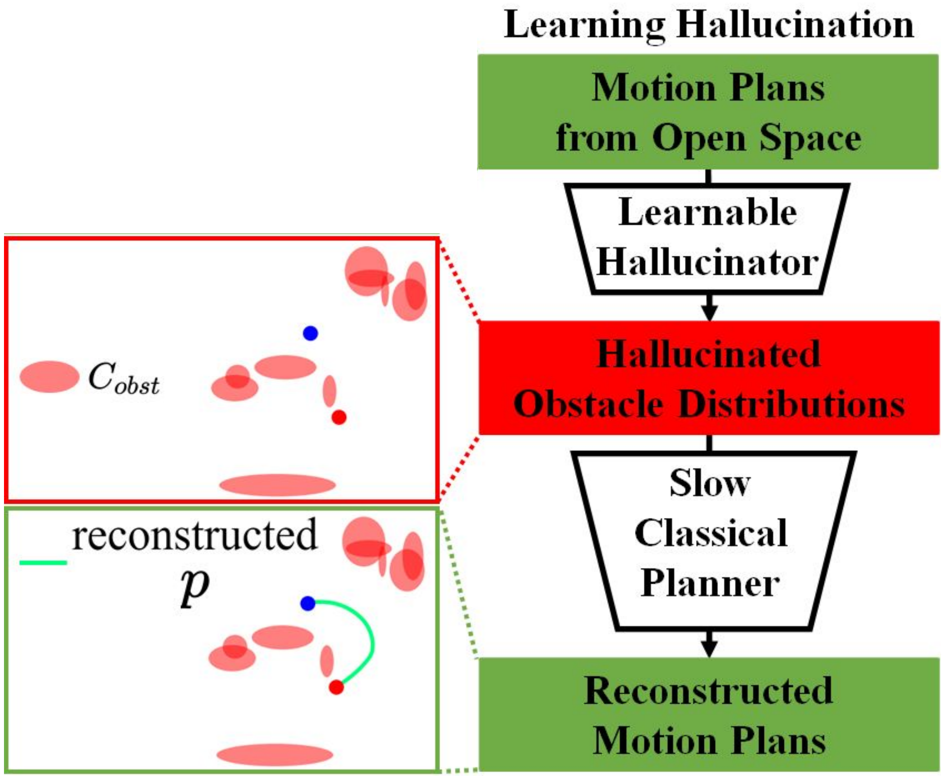
- Hallucination function to learn
- Input: motion plan (time series of positions + velocities)
- Output: obstacle distribution (normal distributions of obstacle locations + sizes)
- Parametrized as neural network



# Learning from Learned Hallucination

**Method - Learning Hallucination**
Classical Planner

- Find the optimal motion plan given obstacles
- Input: sampled obstacles (locations + sizes)
- Output: optimal motion plan (time series of positions + velocities)
- **No parameters to learn**
- Slow for data collection, but still can be used for training

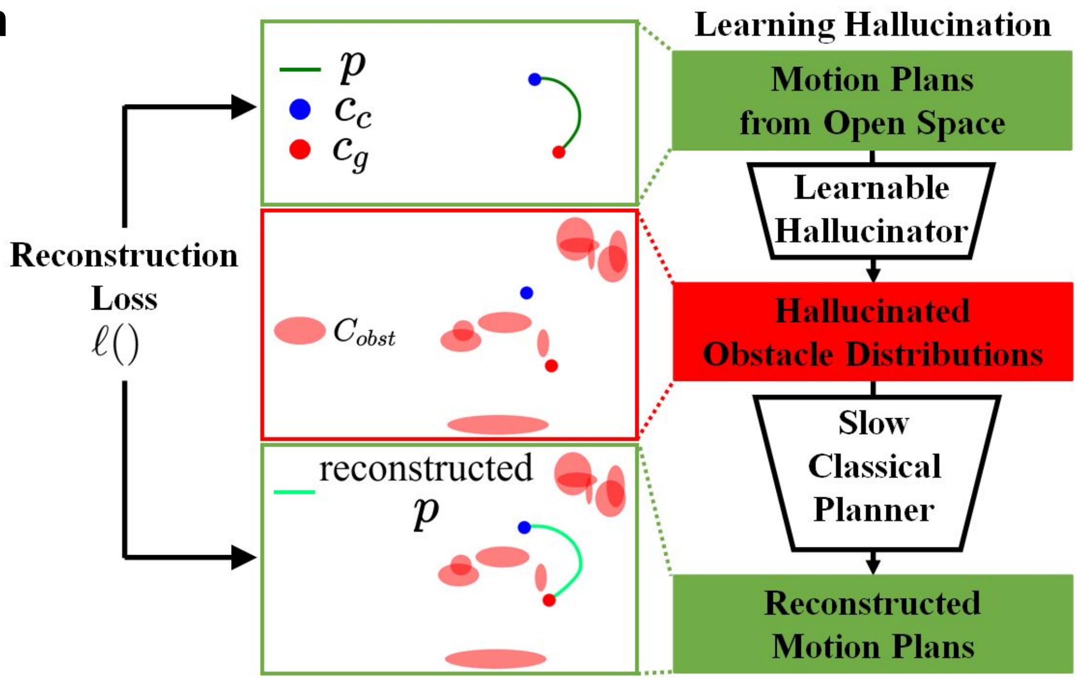


# Learning from Learned Hallucination

## Method - Learning Hallucination

Training

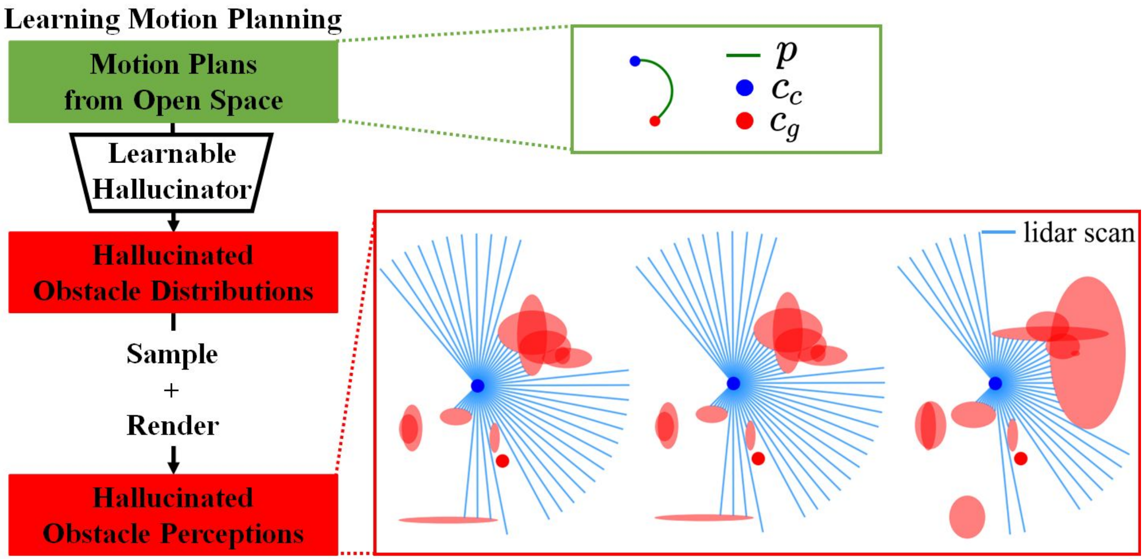
- Train the hallucinator with a reconstruction loss.
- If reconstruction loss = 0, hallucinator finds obstacles where the open space motion plan is the optimal solution.
- We use a differentiable optimization-based planner. For non-differentiable planners, one can use approximate gradients.



# Learning from Learned Hallucination

## Method - Learning Motion Planning

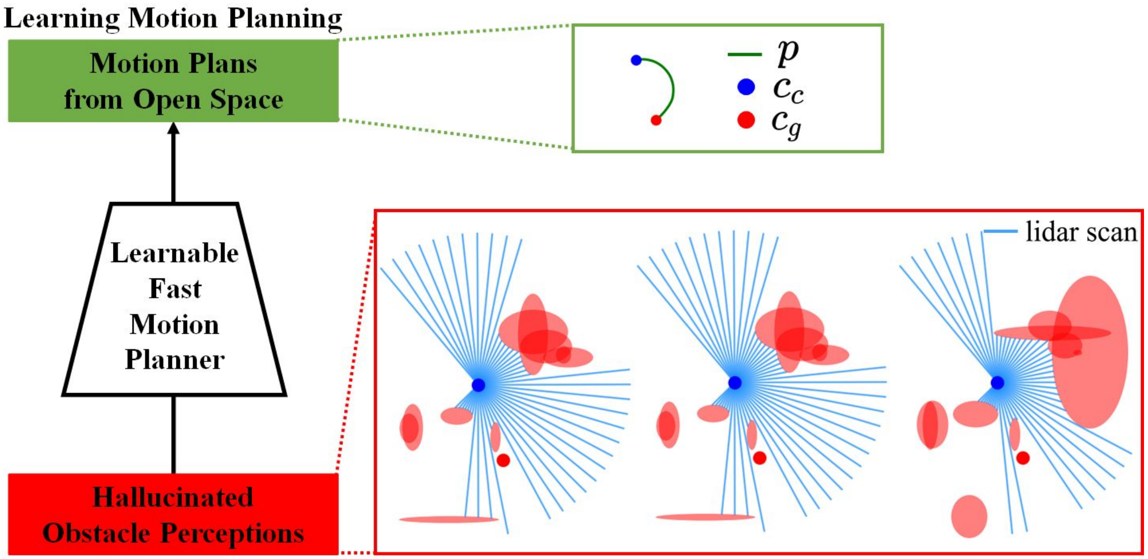
- Use learned hallucinator to sample obstacles.
- Render observations according to robot’s sensor modalities.



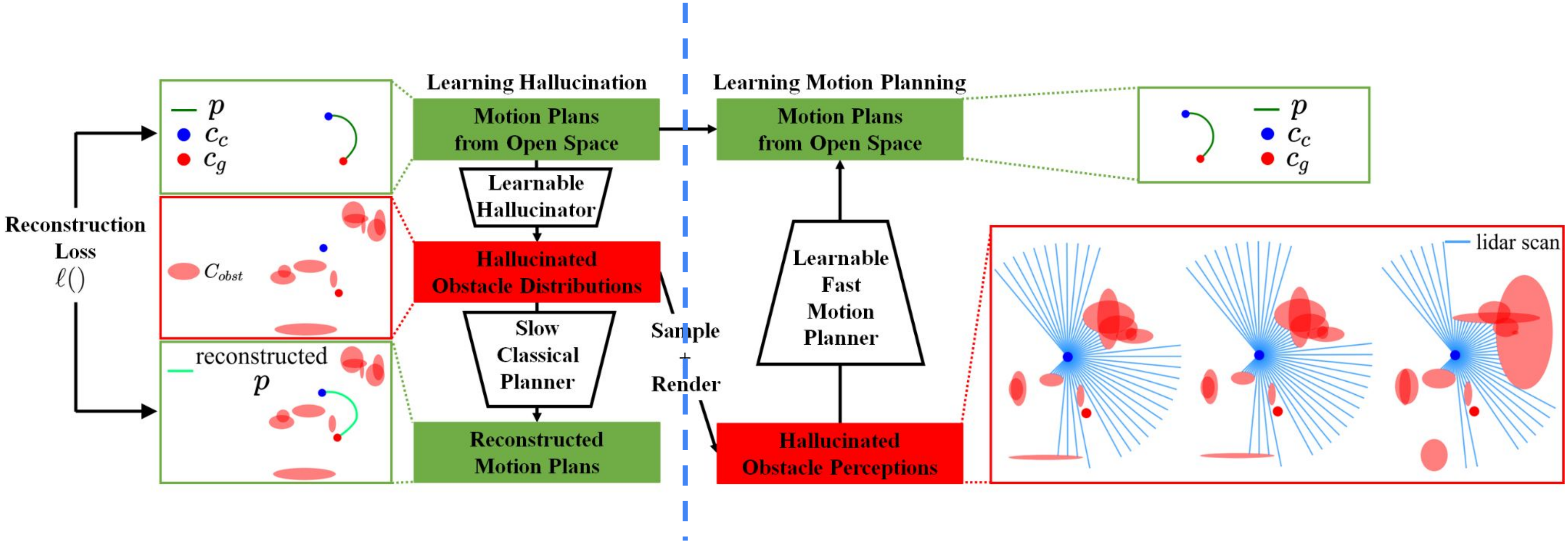
# Learning from Learned Hallucination

## Method - Learning Motion Planning

- Use learned hallucinator to sample obstacles.
- Render observations according to robot's sensor modalities.
- Train imitation learning motion planner with open space motion plans as training data.



# Learning from Learned Hallucination

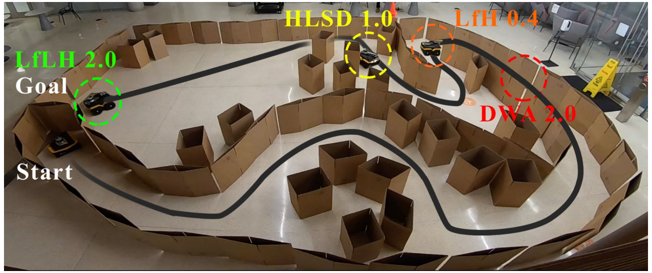


# Learning from Learned Hallucination

## 2D Navigation Experiment - Setup

Baselines
- DWA planner with max speed 2.0 m/s
- Most constrained hallucination (LfH)
- Minimal hallucination + additional obstacles (HLSD)

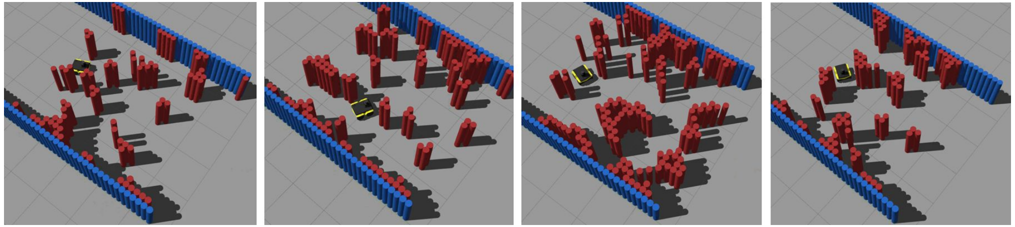
**Robot**: Clearpath Jackal




**Physical Environment**

Perille, et al., *Benchmarking Metric Ground Navigation*, SSRR'20

Dataset of varying max speed
- LfH learns up to 0.4 m/s
- HLSD learns up to 1.0 m/s
- LfLH learns 2.0 m/s and beyond



**Simulated Environment**

# Learning from Learned Hallucination

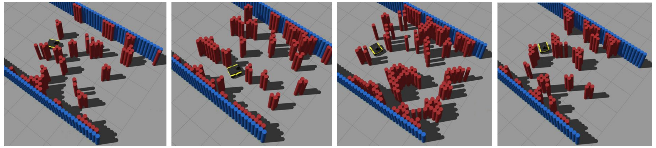
## 2D Navigation Experiment - Results

| DWA 2.0 | Dataset | Most Constrained | Minimal Obstacles | LfLH |
|---|---|---|---|---|
| 22.1 ± 11.4 s | 0.4 m/s | 13.8 ± 5.3 s | 13.2 ± 7.9 s | 13.4 ± 6.4 s |
| | 1.0 m/s | N/A | 8.5 ± 5.2 s | 8.3 ± 3.8 s |
| | 2.0 m/s | N/A | N/A | **8.1** ± 5.4 s |

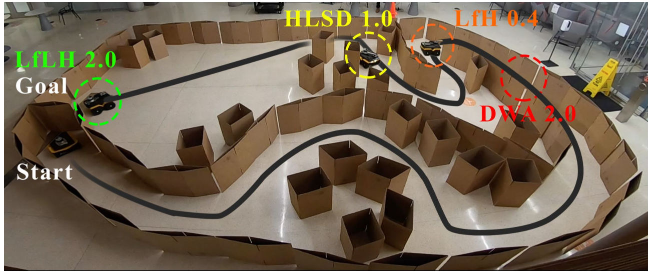
**Simulated Average Traversal Time**

| DWA 2.0 m/s | Most Constrained 0.4 m/s | Minimal Obstacles 1.0 m/s | LfLH 2.0 m/s |
|---|---|---|---|
| 73.6 ± 3.8 s | 78.4 ± 1.8 s | 50.6 ± 0.8 s | **41.1** ± 0.9 s |

**Physical Average Traversal Time**



**Simulated Environment**




**Physical Environment**

# Learning from Learned Hallucination

## 3D Navigation Experiment - Setup

Baselines

- Previous hallucination methods cannot handle 3D navigation, so they are not tested.
- EGO-planner (Zhou, et al., RA-L 20)

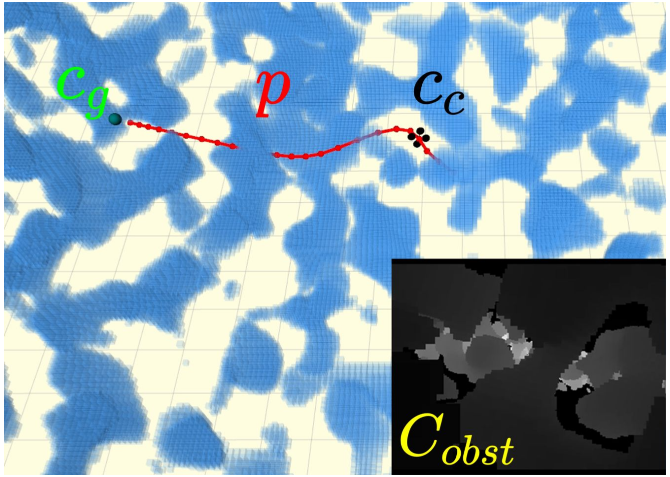
Dataset: collected by EGO-planner rather than random policy.

Task: keep navigating to randomly-generated goals until collision.

Metrics: survival distance, survival time, success weighted by path length.

(S: success boolean, l: shortest path length, p: path length)

$$\mathrm{SPL} = \frac{1}{K}\sum_{i=1}^{K} S_i \frac{l_i}{p_i}$$



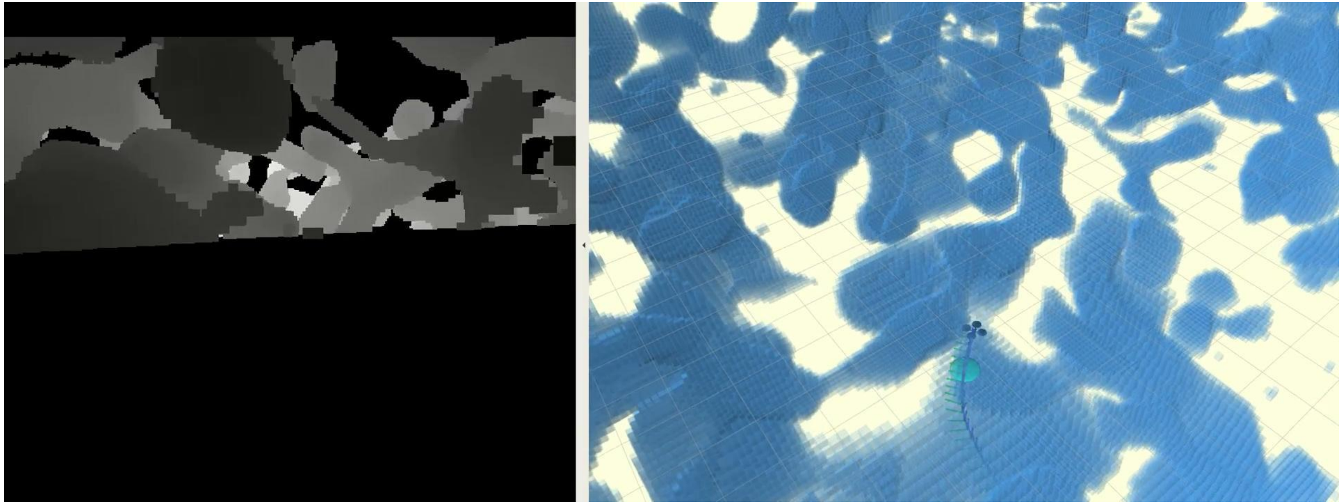

**Simulated Environment & Observation (Depth Image)**

# Learning from Learned Hallucination

## 3D Navigation Experiment - Results

- LfLH survives longer in both distance and time but has lower SPL.
- LfLH trades off aggressive motions for safety.

| Metrics | Ego-Planner | LfLH |
|---|---|---|
| Survival Time (s) | 101.99±62.83 | **192.87**±161.37 |
| Survival Distance (m) | 174.15±106.74 | **213.07**±172.98 |
| SPL | **0.74** | 0.56 |



# Future Work

- Use random policy to collect data for aerial vehicle.
- Design good exploration policy to cover necessary navigation skills for all obstacle configurations.
- Can we apply Learning from Hallucination to dynamic obstacles?

# From Agile Ground to Aerial Navigation: Learning from Learned Hallucination (LfLH)

Zizhao Wang[1], Xuesu Xiao[2], Alexander J Nettekoven[3], Kadhiravan Umasankar[4], Anika Singh[1], Sriram Bommakanti[4], Ufuk Topcu[4], and Peter Stone[2,5]

[1]ECE, [2]CS, [3]Mechanical Engr,
[4]Aerospace Engr & Engr Mechanics,
The University of Texas at Austin
[5]Sony AI

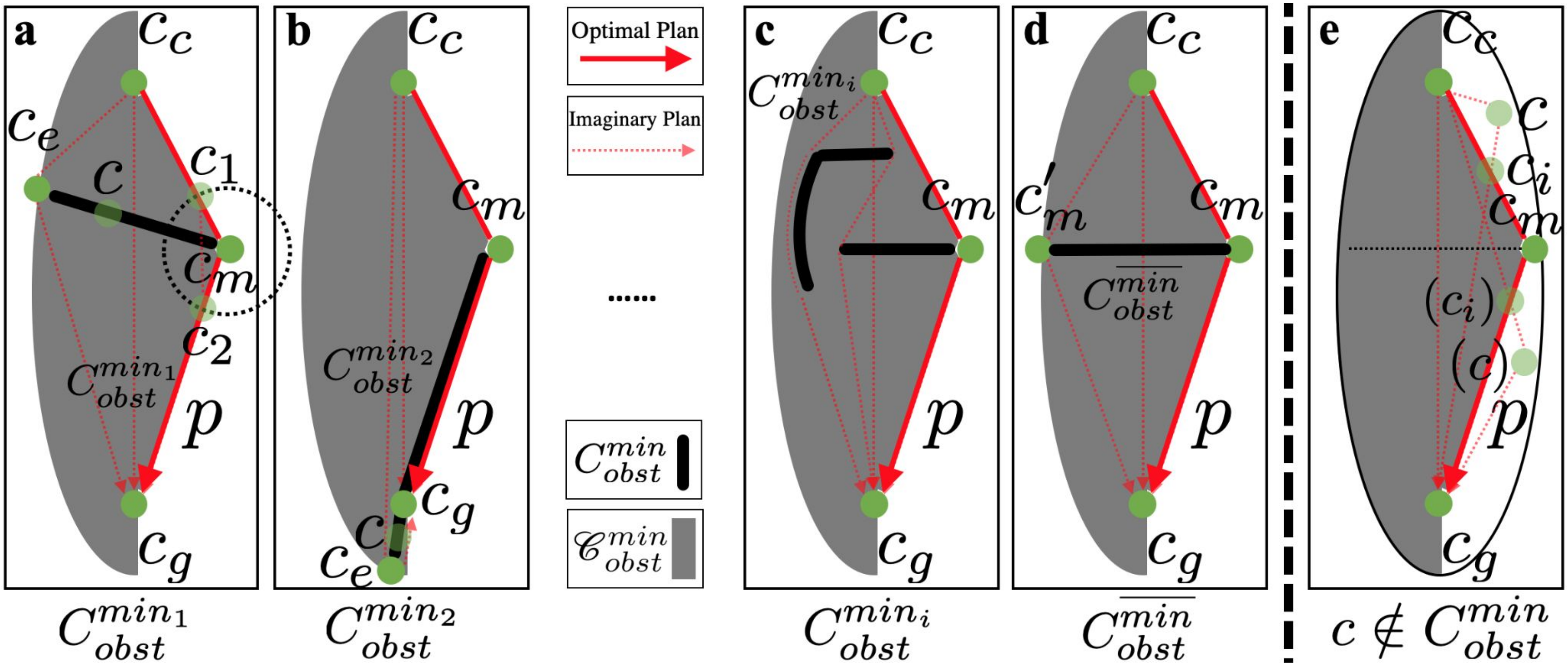
Contact Information:
Zizhao Wang: zizhao.wang@utexas.edu

Link to the Paper: https://arxiv.org/pdf/2108.09801.pdf

Learning Agents Research Group
The University of Texas at Austin

autonomous SYSTEMS GROUP

Scan to access the paper



a
b
c
d
e
$c_c$
$c_e$
$c$
$c_1$
$c_m$
$c_2$
$c_g$
$p$
$C_{obst}^{min_1}$
$C_{obst}^{min_2}$
$C_{obst}^{min_i}$
$c'_m$
$\overline{C_{obst}^{min}}$
$c_i$
$(c_i)$
$(c)$
Optimal Plan
Imaginary Plan
......
$C_{obst}^{min}$
$\mathscr{C}_{obst}^{min}$
$C_{obst}^{min_1}$
$C_{obst}^{min_2}$
$C_{obst}^{min_i}$
$\overline{C_{obst}^{min}}$
$c \notin C_{obst}^{min}$