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

Zizhao Wang¹, Xuesu Xiao², Alexander J Nettekoven³, Kadiravan Umasankar⁴, Anika Singh¹, Sriram Bommakanti⁴, Ufuk Topcu⁴, and Peter Stone²,⁵

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

- **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.

\[ c_g: \text{goal configuration} \]
\[ c_c: \text{current configuration} \]
\[ p: \text{motion plan} \]
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).
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.
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 Hallucination
Motion Plans from Open Space

Learnable Hallucinator
Hallucinated Obstacle Distributions

Slow Classical Planner
Reconstructed Motion Plans
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 Hallucination
Motion Plans from Open Space
Learnable Hallucinator
Hallucinated Obstacle Distributions
Slow Classical Planner
Reconstructed Motion Plans
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)

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

### 2D Navigation Experiment - Results

<table>
<thead>
<tr>
<th></th>
<th>Dataset</th>
<th>Most Constrained</th>
<th>Minimal Obstacles</th>
<th>LfLH</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DWA 2.0</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.4 m/s</td>
<td>13.8 ± 5.3 s</td>
<td>13.2 ± 7.9 s</td>
<td>13.4 ± 6.4 s</td>
<td></td>
</tr>
<tr>
<td>1.0 m/s</td>
<td>N/A</td>
<td>8.5 ± 5.2 s</td>
<td>8.3 ± 3.8 s</td>
<td></td>
</tr>
<tr>
<td>2.0 m/s</td>
<td>N/A</td>
<td>N/A</td>
<td>8.1 ± 5.4 s</td>
<td></td>
</tr>
</tbody>
</table>

**Simulated Average Traversal Time**

<table>
<thead>
<tr>
<th></th>
<th>Dataset</th>
<th>Min Constrained</th>
<th>Min Obstacles</th>
<th>LfLH</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DWA 2.0</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.4 m/s</td>
<td>78.4 ± 1.8 s</td>
<td>50.6 ± 0.8 s</td>
<td>41.1 ± 0.9 s</td>
<td></td>
</tr>
</tbody>
</table>

**Physical Average Traversal Time**

<table>
<thead>
<tr>
<th></th>
<th>Dataset</th>
<th>Min Constrained</th>
<th>Min Obstacles</th>
<th>LfLH</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DWA 2.0</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.4 m/s</td>
<td>78.4 ± 1.8 s</td>
<td>50.6 ± 0.8 s</td>
<td>41.1 ± 0.9 s</td>
<td></td>
</tr>
</tbody>
</table>

**Simulated Environment**

**Physical Environment**

- **DWA 2.0**
- **LfLH**

**Learning from Learned Hallucination**
Learning from Learned Hallucination

3D Navigation Experiment - Setup

Baselines
- Previous hallucination methods cannot handle 3D navigation, so they are not tested.
- EGO-planner \cite{Zhou: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.

\[
\text{SPL} = \frac{1}{K} \sum_{i=1}^{K} S_i \frac{l_i}{p_i}
\]
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.

<table>
<thead>
<tr>
<th>Metrics</th>
<th>Ego-Planner</th>
<th>LfLH</th>
</tr>
</thead>
<tbody>
<tr>
<td>Survival Time (s)</td>
<td>101.99±62.83</td>
<td>192.87±161.37</td>
</tr>
<tr>
<td>Survival Distance (m)</td>
<td>174.15±106.74</td>
<td>213.07±172.98</td>
</tr>
<tr>
<td>SPL</td>
<td>0.74</td>
<td>0.56</td>
</tr>
</tbody>
</table>
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 Wang¹, Xuesu Xiao², Alexander J Nettekoven³, Kadhiravan Umasankar⁴, Anika Singh¹, Sriram Bommakanti⁴, Ufuk Topcu⁴, and Peter Stone²,⁵

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

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


Scan to access the paper