Navigation to Multiple Semantic Targets in Novel Indoor Environments

Siddharth Goel

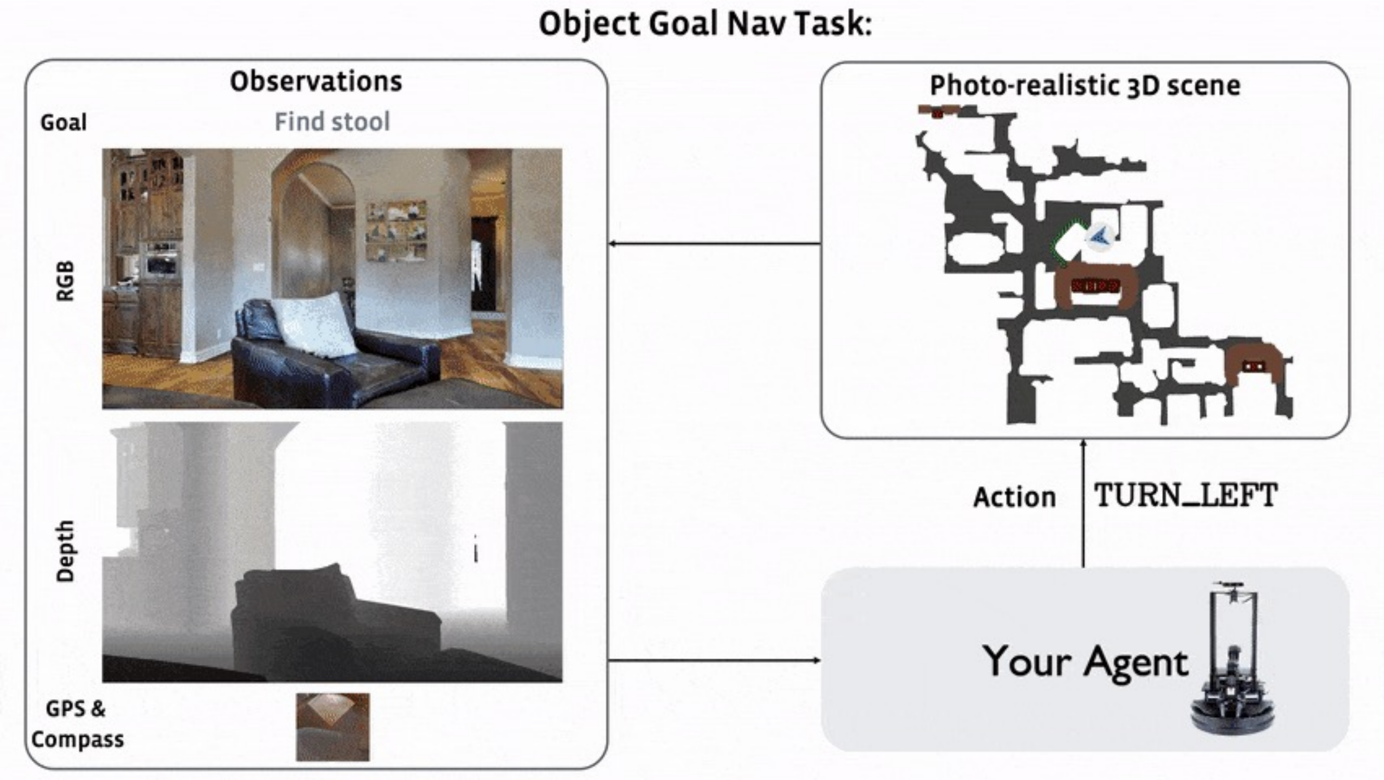
Advised By – Kostas Daniilidis, Georgios Georgakis, Bernadette Bucher

Penn Engineering | GRASP Laboratory

General Robotics, Automation, Sensing & Perception Lab

Visual Navigation in Indoor Environments

Navigation from a random starting position to a point, object, or area using egocentric perception (RGB-D images) in an unseen novel environment



Key Challenges

- ➢No access to environment map
- ➢Layout complexity of indoor environments
- ➢Dynamic layouts from scene to scene (generalization)
- ➢Large number of semantic object categories

Multi-object Navigation (MultiObjNav)

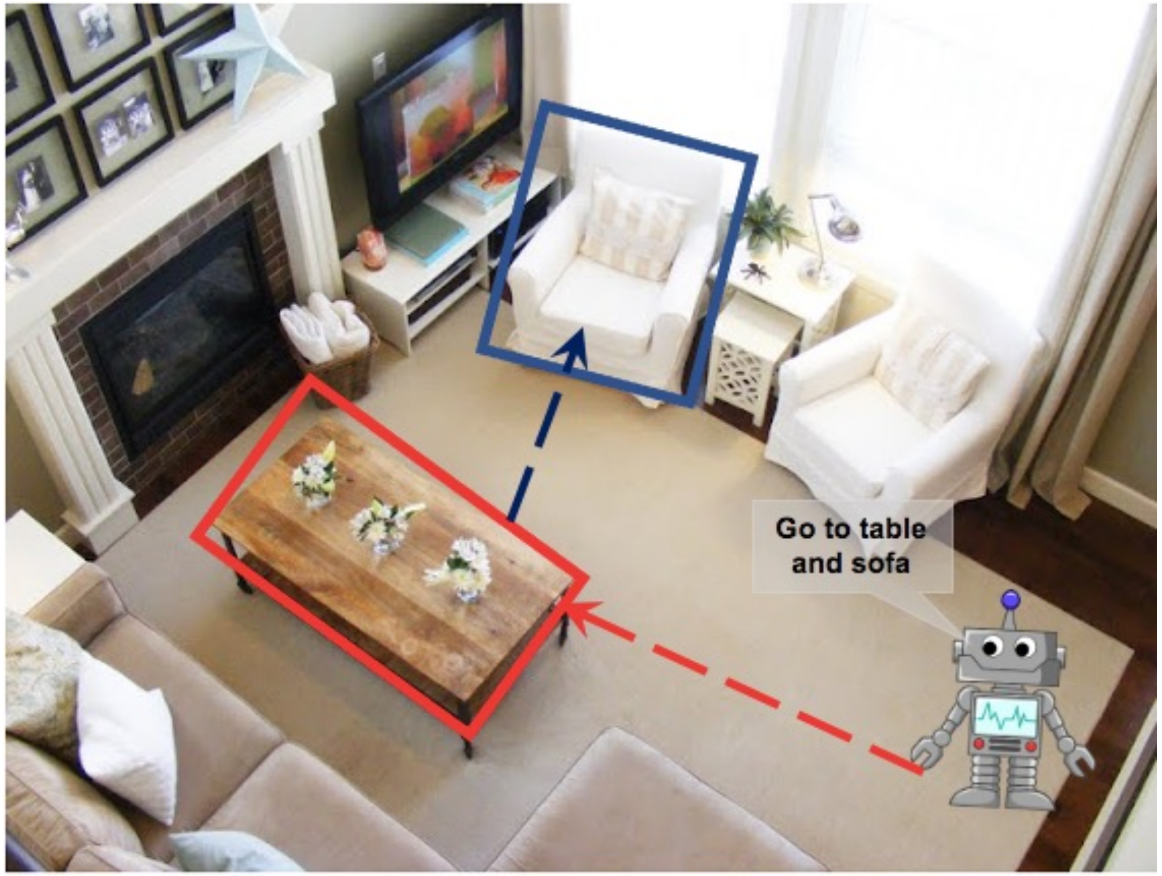
Penn Engineering GRASP LABORATORY
General Robotics, Automation, Sensing & Perception Lab

Motivation
Real life scenarios: Get a glass of water from the refrigerator or asking the agent to pick an item from the table and hand it over to the person on the sofa

Task
- Navigation to *N* (more than one) semantic object
- Unique and non-repetitive target objects
- Generalization of object goal navigation task

Assumptions
Same as visual navigation task i.e.
- No map of the environment
- Only access to egocentric perception images

Complexity
Increases with the number of target objects. 3-object navigation is considered more difficult than 2-object navigation

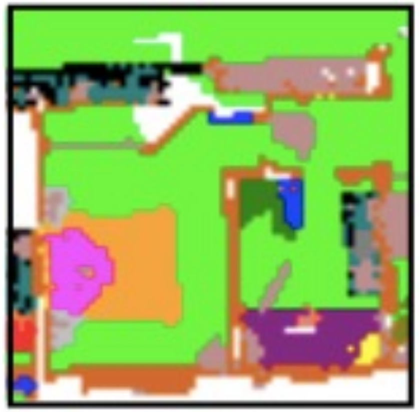


Multi-Object Navigation Approach

Depth

RGB

Semantic Map Prediction



Target list {chair, table, bed, sofa}

Goal Selection policy

Goal Position

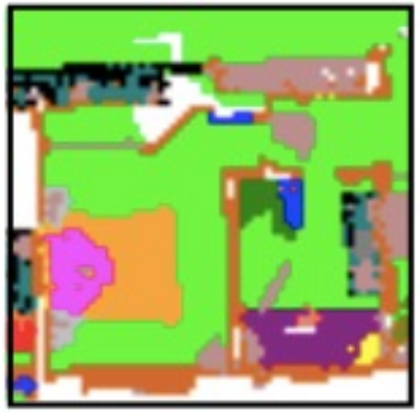
Multi-Object Navigation- Semantic Map

Depth

RGB

1

Semantic Map Prediction



Target list
{chair, table, bed, sofa}

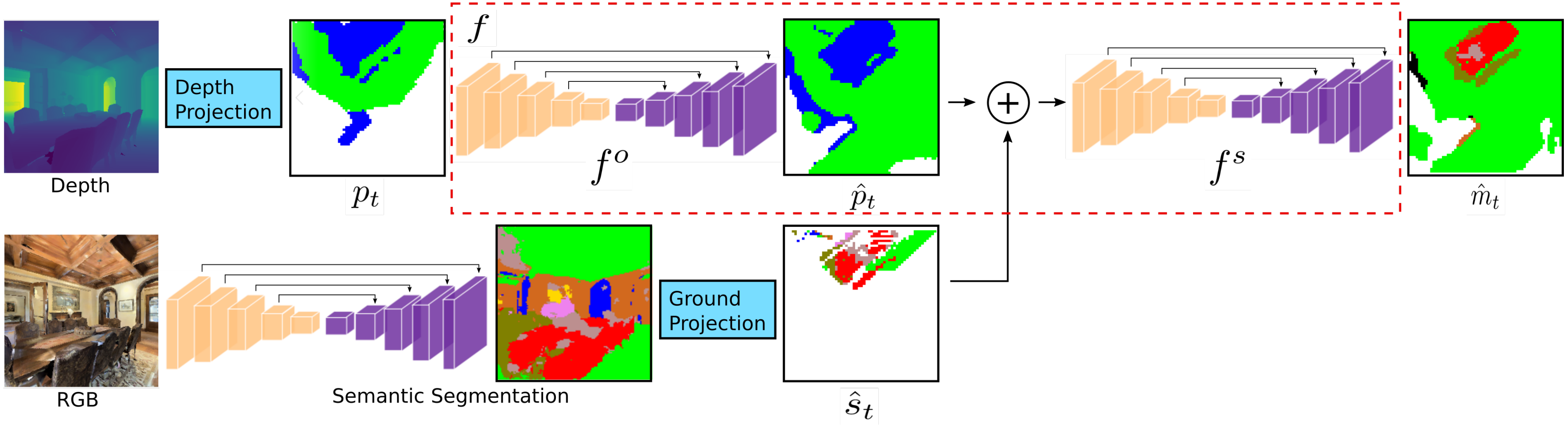
Goal Selection policy

Goal Position

Learning to Map (L2M)[1]

We investigate and improve upon the semantic map prediction module presented in L2M[1]

- Learns to predict the semantic information outside the field of view of the agent
- Ensemble of hierarchical segmentation models
- Two stage prediction – occupancy (unknown, free, occupied) f^o and semantic (chair, table, bed) f^s
- Trained end-to-end using pixel-wise cross-entropy losses for both occupancy and semantic prediction

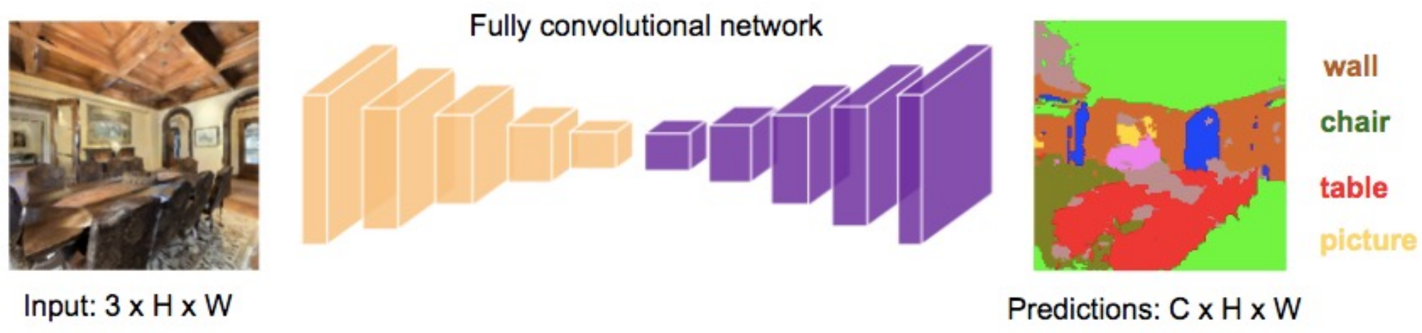


[1] Learning to Map for Active Semantic Goal Navigation
Georgios Georgakis*, Bernadette Bucher*, Karl Schmeckpeper, Siddharth Singh, Kostas Daniilidis
arXiv:2106.15648

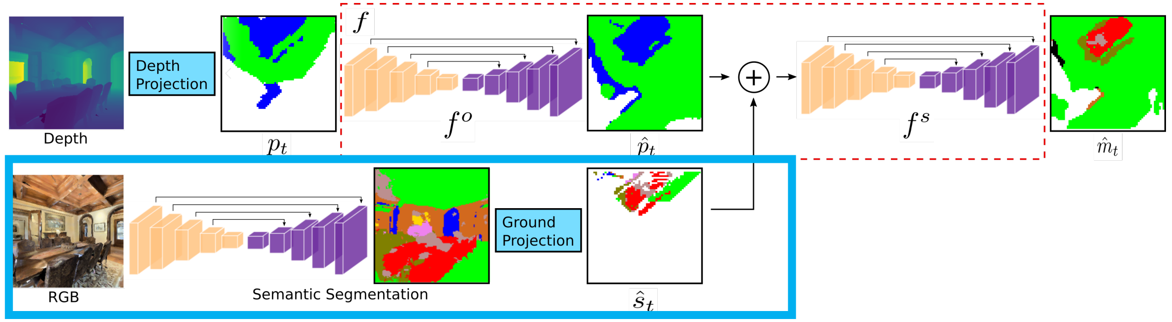
*equal contribution

L2M - Semantic Map Prediction

Pre-trained UNet model for predicting semantic segmentation $(\hat{s}_t)$ of RGB observations



Pixel-wise loss: $$L_{\text{cross-entropy}}(\hat{\mathbf{y}}, \mathbf{y}) = -\sum_i y_i \log(\hat{y}_i)$$

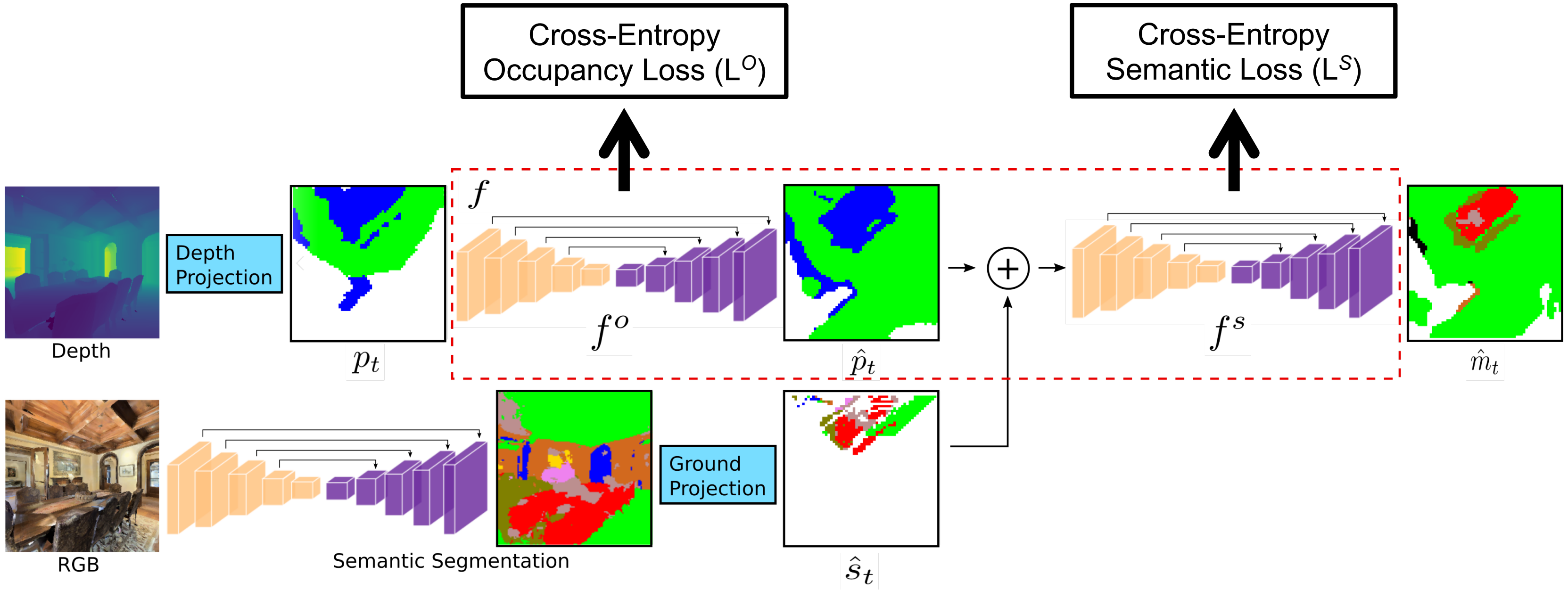


L2M - Semantic Map Prediction Loss

- Both the occupancy and semantic models train end-to-end.
- Total loss L_{sem} is the sum of occupancy loss (L^O) and semantic loss (L^S)

$$L_{sem} = \lambda^O L^O + \lambda^S L^S$$

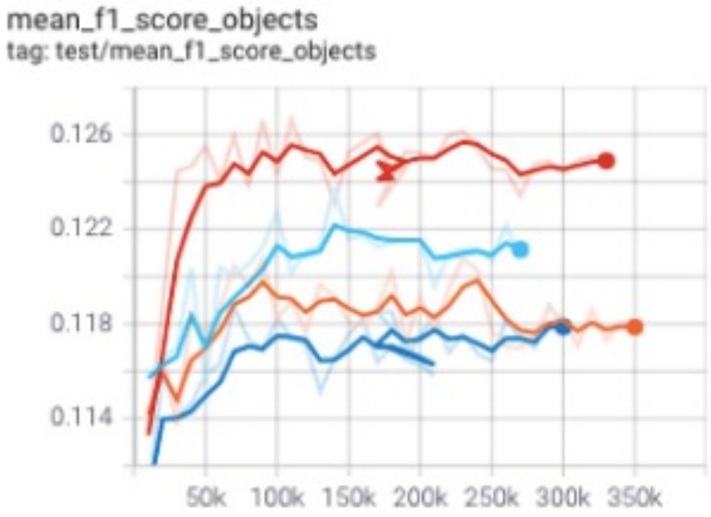
- Both L^O and L^S are pixel-wise cross-entropy losses

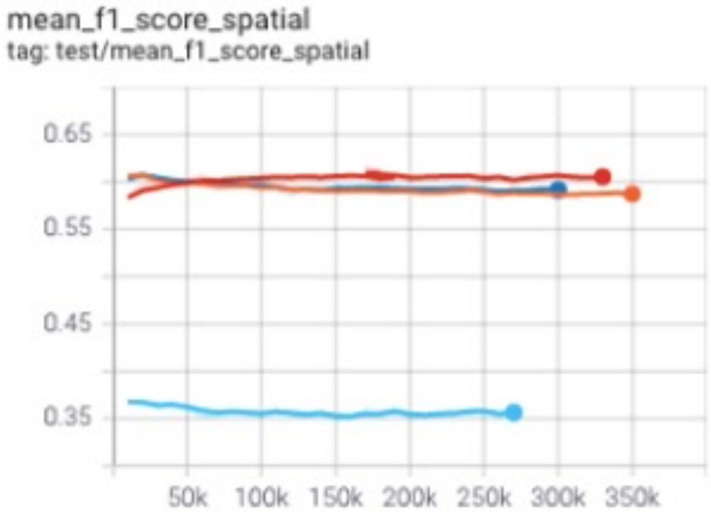


Improving L2M semantic map prediction

Higher weight for Semantic prediction model loss

- Observations comprising semantic objects (chair, table, bed) are much less in number than observations comprising free space, walls, and floor resulting in an extreme class imbalance.
- L^O tends to dominate the total loss in $L_{sem} = \lambda^O L^O + \lambda^S L^S$ when $\lambda^O = \lambda^S$
- The loss function must put more emphasis on identifying objects to counter the overwhelming effect of L^O
- Fine tune values of λ^o and λ^s in $L_{sem} = \lambda^O L^O + \lambda^S L^S$





Occupancy Loss weight	Semantic loss weight
1	1
1	10
0.1	10
0.1	100

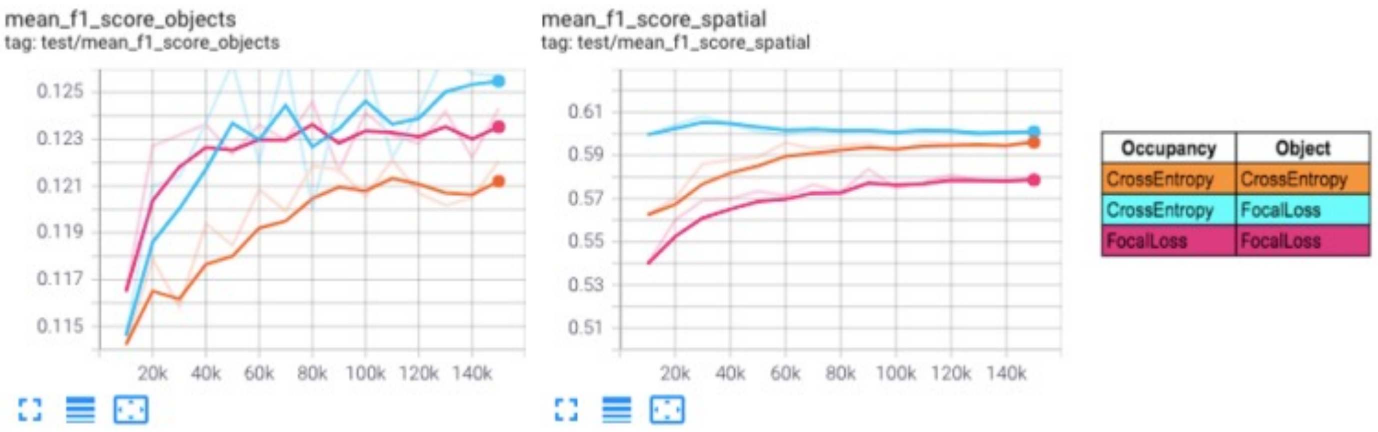
Improving L2M semantic map prediction

Use focal loss in place of cross-entropy (CE) loss for semantic object prediction

- Focal loss is a specialized loss function for the scenario with exponentially large number of easy negatives (*unknown, occupied, free*) and very less number of hard positives (semantic objects).
- It employs a multiplicative factor of $(1 - p_i)^\gamma$ which weighs down the loss value for easy negatives, where γ is a tunable hyperparameter.

$$\text{CE Loss} = -\sum_{i=1}^{N} y_i \log(p_i)$$

$$\text{Focal Loss} = -\sum_{i=1}^{N} y_i (1 - p_i)^\gamma \log(p_i)$$

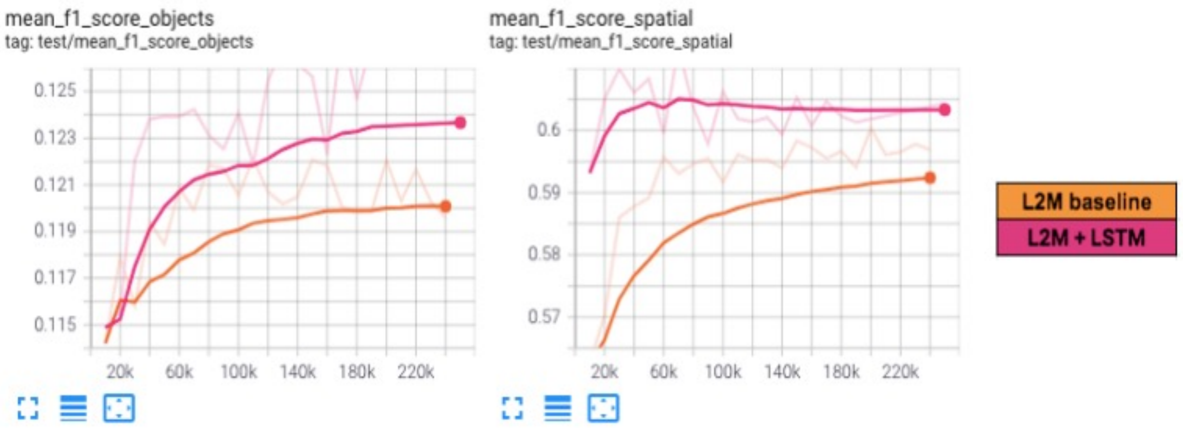


Mean F1 score for spatial and object prediction for different loss functions

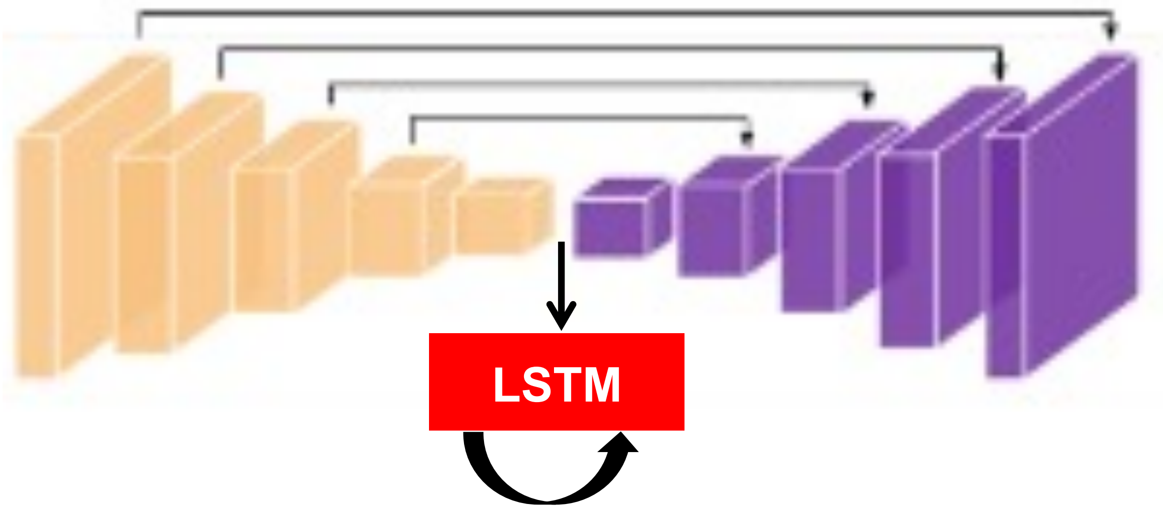
Improving L2M semantic map prediction

Incorporate LSTM layer

- Each episode is a sequence of observations
- The temporal information such as chairs are in vicinity of table or cushion co-occur with bed or sofa should be incorporated in the model
- Incorporate LSTM layer in the neural net architecture to maintain temporal consistency among the sequence of RGB-D egocentric observations



Mean F1 score for spatial and object prediction with LSTM layer



Experimental Results

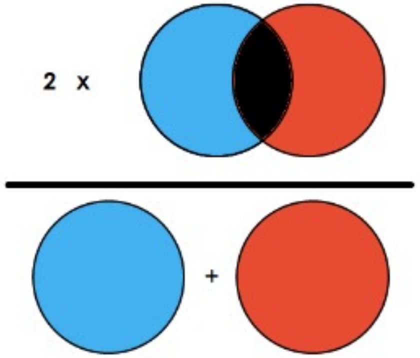
Penn Engineering GRASP LABORATORY

Semantic Segmentation Metrics

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

IoU = Area of Overlap / Area of Union

F1 = 2 x Area of Overlap / Total Pixels Combined =



Semantic map prediction results

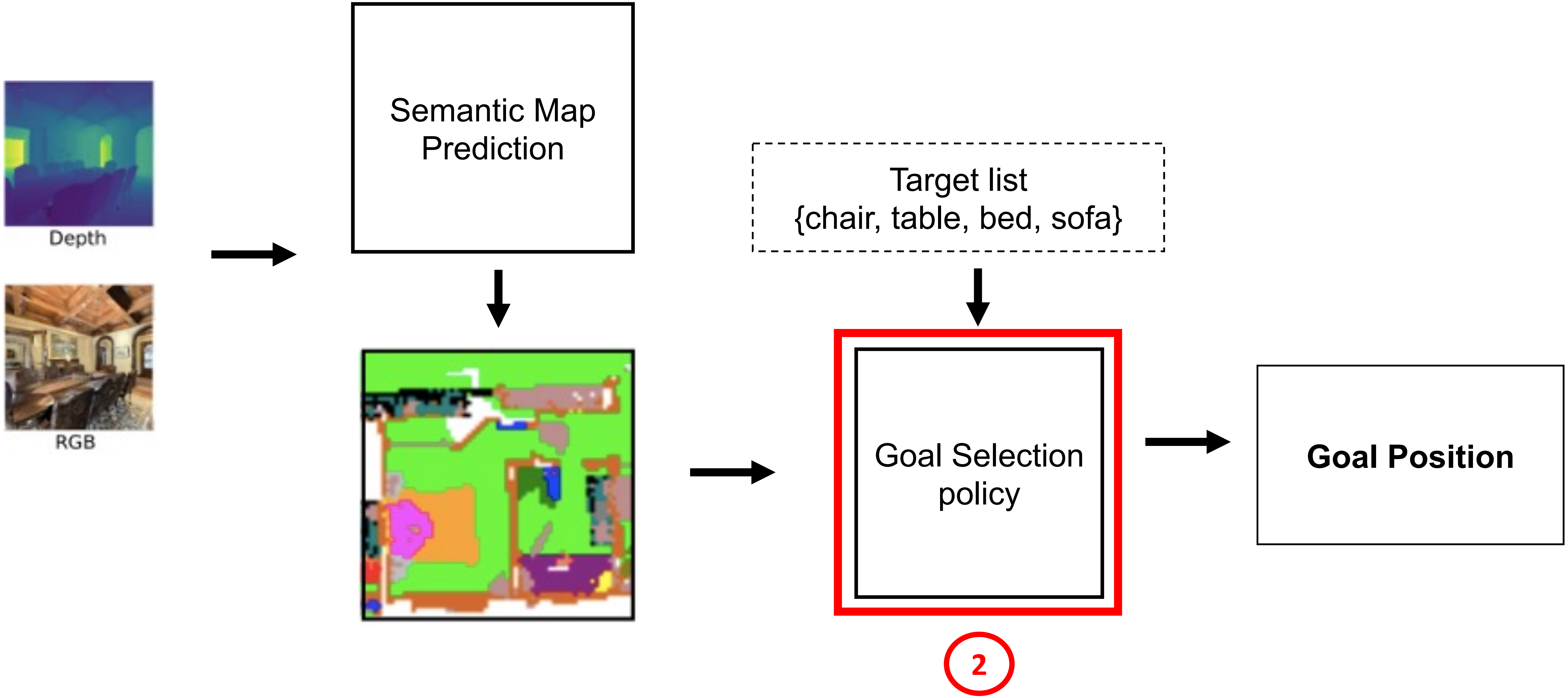
Method	Acc(%)	IoU(%)	F1(%)
Occupancy Prediction			
L2M	65.2	45.5	61.9
L2M+FocalLoss+LSTM	**66.0**	**46.5**	**63.0**
Semantic Prediction			
L2M	**31.2**	20.1	30.5
L2M+FocalLoss+LSTM	29.0	**21.1**	**31.7**

Image credit: https://towardsdatascience.com/metrics-to-evaluate-your-semantic-segmentation-model-6bcb99639aa2

Goal Selection Policy

Multi-object Navigation

Multi-Object Navigation- Goal Selection

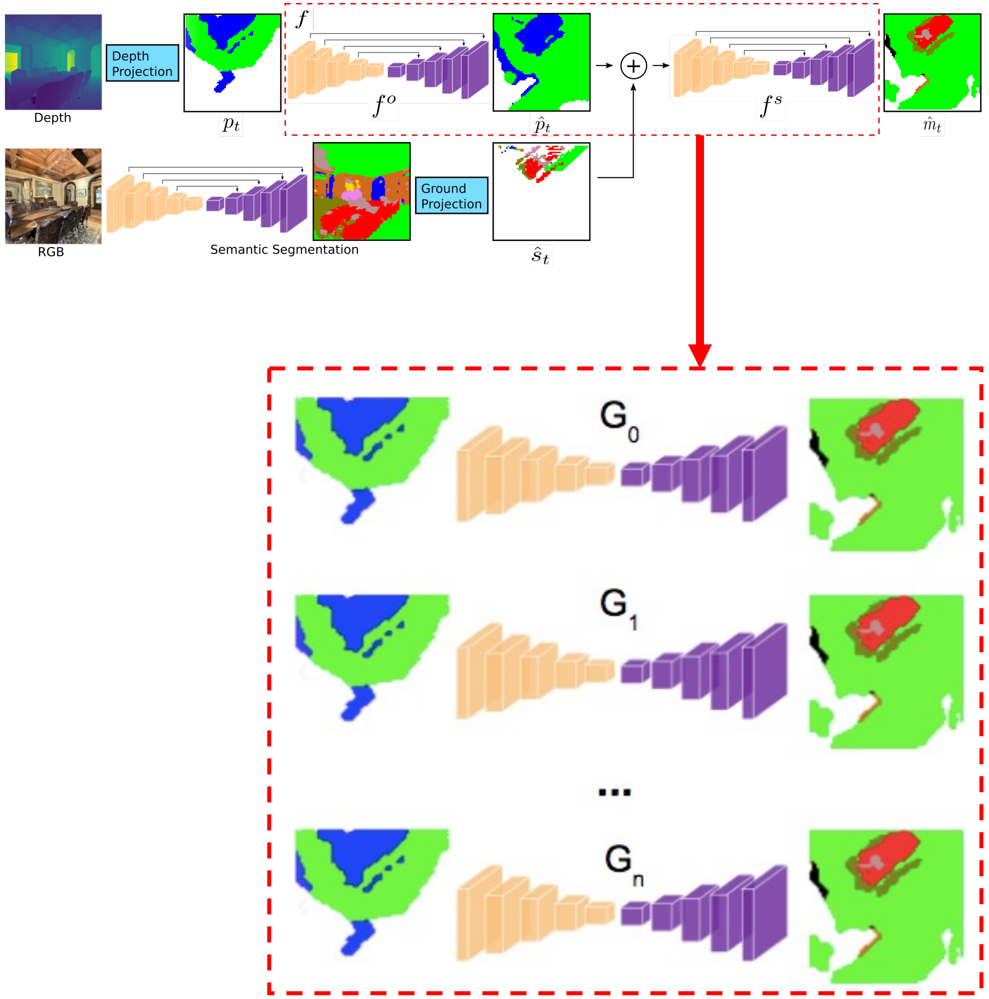


Goal Selection policy - MultiObjNav

- Pursue success over short length paths

- Balance exploitation of semantic information with exploration of the map

$$\underset{\rho_i \in \rho}{\arg\max} \sum_{j=0}^{N-1} \alpha_0^{-j} \left(\mu_j(p_t, \hat{s}_t) + \alpha_1 \sigma_j(p_t, \hat{s}_t) - \alpha_2 d_{j,j+1} \right)$$



where

p_t : observation

$\hat{s}_t$: semantic segmentation

$\mu_j(p_t, \hat{s}_t)$: mean estimate of the ensemble models

$\sigma_j(p_t, \hat{s}_t)$: the standard deviation of the target class probability

$d_{j,j+1}$: euclidean distance between j^{th} and $(j+1)^{th}$ node

$\alpha_o, \alpha_1, \alpha_2$: hyperparameters

N : number of target objects

ρ : Set of candidate paths

Experiments & Results

Multi-object Navigation

Experimental Setup

- **AI-Habitat** [(https://aihabitat.org/)](https://aihabitat.org/)

High-performance 3D simulator with configurable agents, multiple sensors, and generic 3D dataset handling

- **Matterport 3D**

Dataset containing reconstructions of real indoor scenes from 90 buildings

- Total number of scenes: 10
- Total number of episodes: 680
- Experiments conducted for 2-object navigation
- Target object combinations - [chair, table], [bed,cabinet], [bed, sofa], [table, cabinet], [table, sofa], [cabinet, sofa], [table, bed], [chair, cabinet], [chair, sofa], [chair , bed]

Habitat Simulator configuration

Parameter	Value
Max. episode steps	1000
Sensors	['RGB', 'Depth']
Height of agent(m)	1.5
Task type	ObjectNav-v1
Possible Actions	['Stop', 'Move Forward', 'Turn Left', 'Turn Right']
Move forward distance (cm)	25
Turn left/right angle	10°

RGB frames from two different scenes in Matterport3D dataset along with their corresponding depth images

Multi Object Navigation Metrics

- **Success** – Binary indicator for episode success – if the agent is able to navigate to all the target semantic objects within the allowed number of steps
- **Progress** – Ratio of number of semantic objects reached successfully by the agent to the total number of target semantic objects. If the agent navigates to 2 out of 3 target objects then progress is equal to 2/3 = 0.66
- **Success weighted by path length (SPL)** - quantifies the distance covered by an agent in a successful episode.

$$SPL = Success \,.\, d/max(p, d)$$

- **Progress weighted by Path Length (PPL)** - measures the distance covered by an agent in an unsuccessful episode.

$$PPL = Progress \,.\, d/max(p, d)$$

where

d : length of the shortest route spanning agent's starting position and all the objects

p : total distance travelled by the agent

Experimental Results - MultiObjNav

Two sets of experiments were performed for 2-object navigation

- Agent *does not have access* to stop oracle
  The agent agent must take stop decision by itself after recognizing a goal state
- Agent *has access* to stop oracle
  The agent refers to the oracle to check if it has reached the goal state

Multi-object Navigation Experimental Results

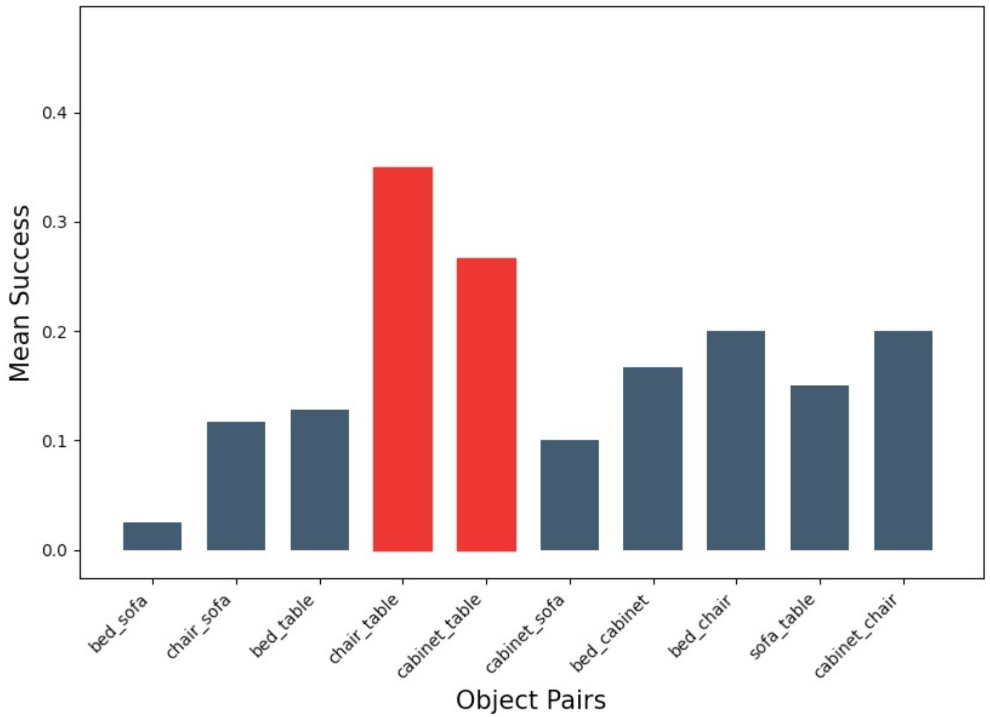
Method	Success(%)	Progress(%)	SPL(%)	PPL(%)
Multi-obj-L2M	2.35	11.98	2.46	9.27
Multi-obj-L2M-OS	17.60	41.53	15.98	35.30

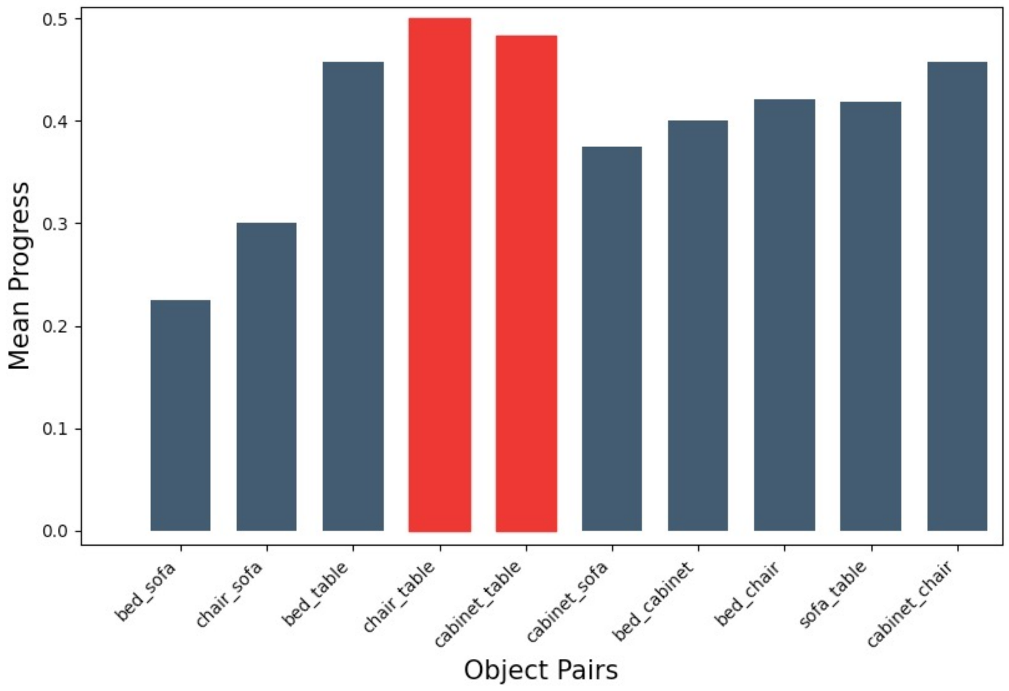
OS: oracle stop

Experimental Results - MultiObjNav

Penn Engineering
GRASP LABORATORY
General Robotics, Automation, Sensing & Perception Lab

- Agent performance based on target object categories
- Agent performs better on objects which
  - co-occur
  - have high frequency





Thank You !!!