OPEN SCENE GRAPHS
FOR OPEN-WORLD OBJECT GOAL NAVIGATION

Joel Loo* [joell@u.nus.edu], **Zhanxin Wu*** [zhanxinwu@u.nus.edu], **David Hsu** [dyhsu@comp.nus.edu.sg]

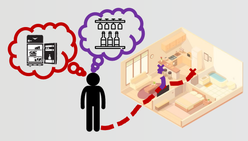
NUS National University of Singapore

Check out our website!

Open-World ObjectNav

Can we search novel scenes for an open-set object class, in *any* environment, with *any* embodiment?

- Requires semantic reasoning, common-sense priors

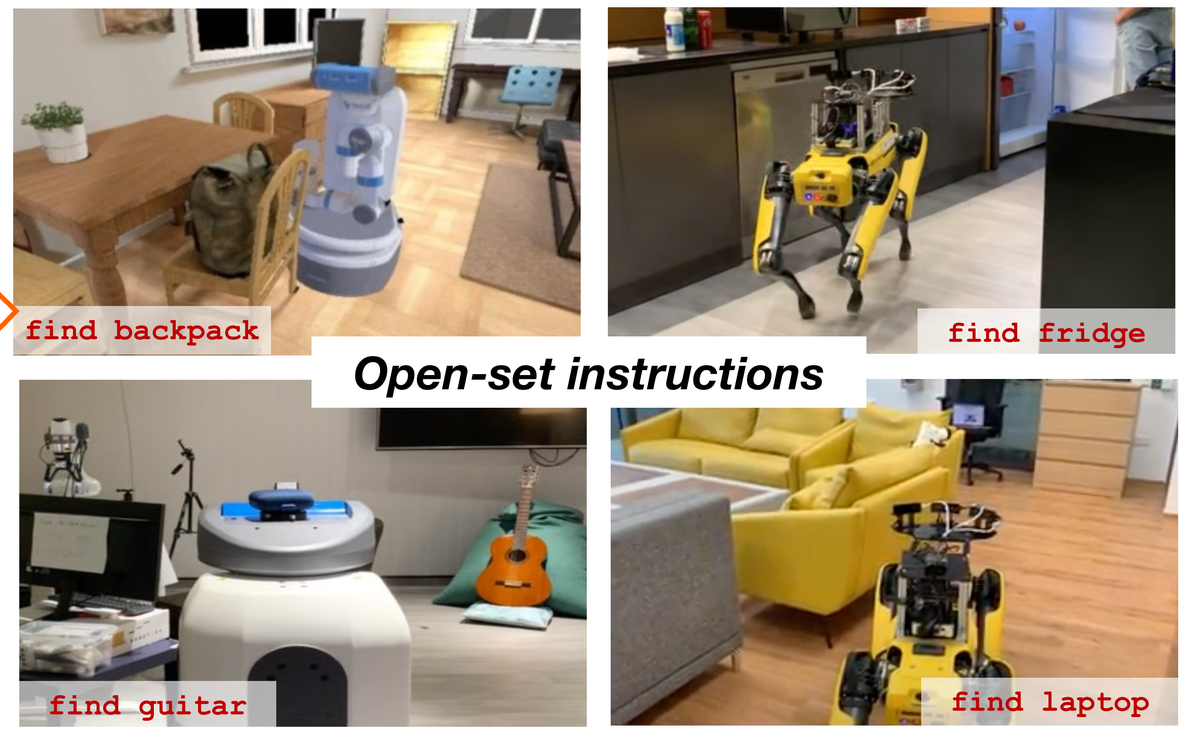


- Must handle diverse instructions, environments & embodiments

Approach: Compose an ObjectNav robot system purely built from Foundation Models (FMs)

Problem: Need a structured scene memory to retain information for FMs, that is also built with FMs

Explorer, a general system for Open-World ObjectNav



Open Scene Graphs

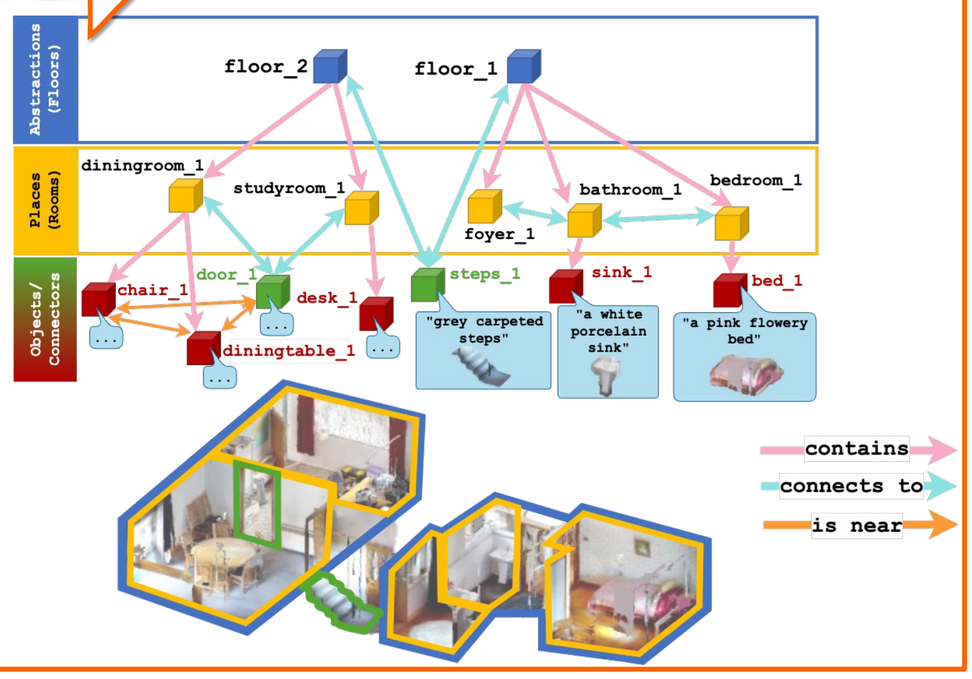
Meta-structure *are constraints on* **OSG Specification**: *E.g.*, of household environments *specifies structure of* **OSG instance**: *E.g.*, of a specific household scene

Defines ***minimal structure and scene information*** needed in OSGs to enable ***localisation, planning and control***.

I.e.,

- Minimally needed layer, node and edge types
- Minimally needed attributes in each node
- Allowed edge connections

```
⋮
"RegionAbstraction1_Layer": {
# [Optional] Semantically meaningful spatial abstractions
    "layer_type": "floor",
"Places_Layer": {
# [Required] Smallest semantically meaningful regions
    "layer_type": "room",
"Connectors_Layer": {
#[Optional] Connective structures between regions
    "layer_type": ["stairs", "doors"]
"Objects_Layer": {
# [Required] Task-relevant semantic features
⋮
```



OSG Mapper: Builds OSGs online, following OSG spec

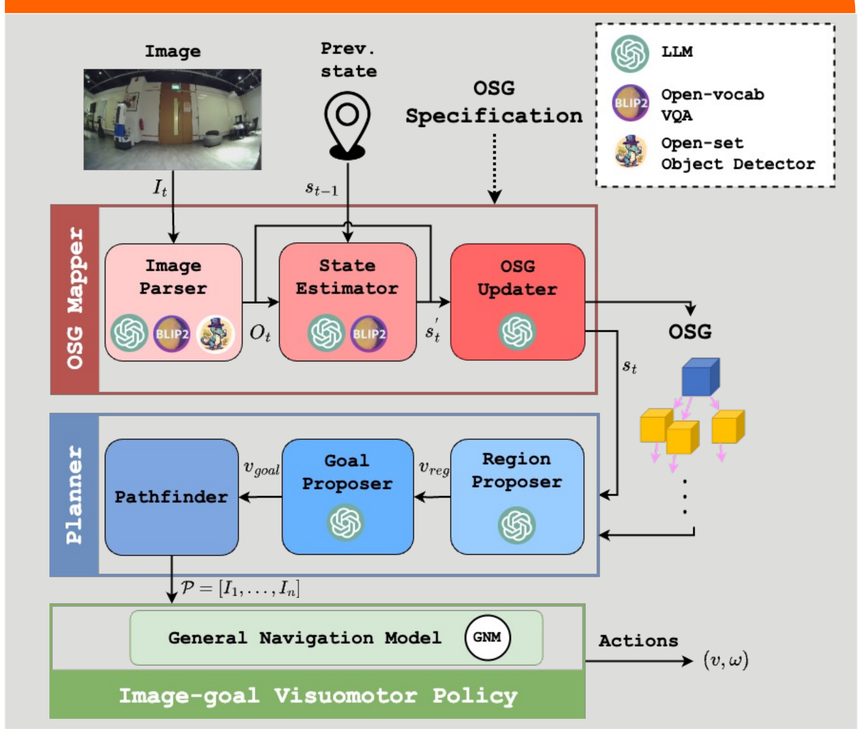
- Uses only ***FMs***: *i.e.*, LLM, open-vocab VQA, open-set Object Detector
- Uses ***objects as features*** for data association

fn ImageParser(I_t):
  // Extract Place, Objects, Connectors from observations
  P_t = LabelPlace_VLM_VQA(I_t)
  D_t = DetectObjectsConnectors_VLM_OD(I_t)
  O_t, C_t = ClassifyObjectsAndConnectors_LLM(D_t)
  for *o in* $O_t \cup C_t$ **do**
    $o.attr$ = LabelWithTextualAttribs_VLM_VQA(o)
  return P_t, O_t, C_t

fn StateEstimator(P_t, O_t, C_t):
  // Estimate robot's current location (Place)
  P^{OSG} = SortByDistance(*all Place nodes in OSG*)
  for p *in* P^{OSG} **do**
    if IsPlaceMatchedWithObs_LLM($p, O_t \cup C_t$)
      return p
  return None

fn OSGUpdater($\hat{p}, O_t, C_t$):
  // Update OSG with observed Place, Objects, Connectors
  if $\hat{p}$ *is* None
    p_t = AddPlaceNode($\hat{p}$)
    AddObjectConnectorLeafNodes(O_t, C_t)
    AddEdges(s_t, O_t, C_t)
    InferRegionAbstractions_LLM(p_t)
  else
    $p_t = \hat{p}$
    for o *in* $O_t \cup C_t$ **do**
      $\mathcal{V}_{leaf}$ = GetObjectConnectorLeafNodes(p_t)
      v_{match} = FindMatchedLeafNode_LLM($o, \mathcal{V}_{leaf}$)
      if v_{match} is None
        AddNewLeafNode(o)
      else
        UpdateLeafNode(o)
      UpdateEdges(o)
  return p_t

Explorer System Architecture



Experiments: Simulation

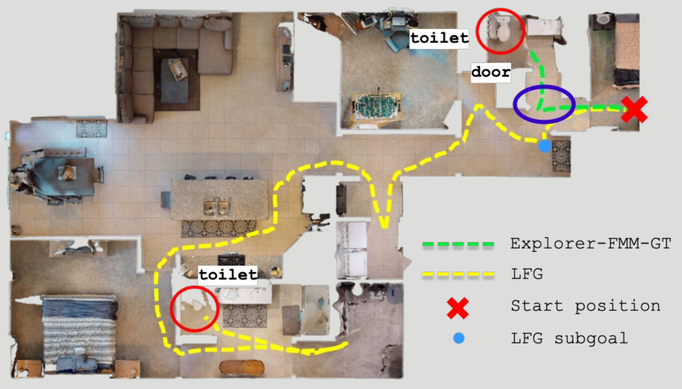
Comparison With LLM-based ObjectNav Methods On HM3D

Method	Success (↑)	SPL (↑)	DTG (m) (↓)
Greedy LLM (based on [11])	0.275	0.080	5.078
LFG [39]	0.675	**0.389**	2.411
Explorer-FMM-GT	**0.775**	0.380	**1.702**
Explorer-FMM	0.693	0.283	2.338

Comparison With ObjectNav Methods On Gibson

Method	Success (↑)	SPL (↑)	DTG (↓)	TF	NM
SemExp [5]	0.657	0.339	1.474	✗	✗
PONI [31]	**0.736**	**0.410**	**1.250**	✗	✗
FBE [50]	0.641	0.283	1.780	✓	✗
SemUtil [8]	0.693	**0.405**	**1.488**	✓	✗
Explorer-FMM	**0.734**	0.386	1.722	✓	✓

(**TF: training free. NM: Non-metric**)



- LLM reasoning with ***Open Scene Graphs*** lets ***Explorer*** strongly outperform ***LLM-based methods***
- LLMs' rich semantic priors lets the ***zero-shot Explorer*** perform on par with ***task-specific learned methods***