[졸업 작품전 – 작품]

Task-Aware Semantic Map: Autonomous Robot Task Assignment Beyond Commands

- 이호성 : 융합전자공학부 2020043481
- 황소은 : 융합전자공학부 2019041058

Contents

1 TASMap

2 Framework

3 Application

4 Dataset

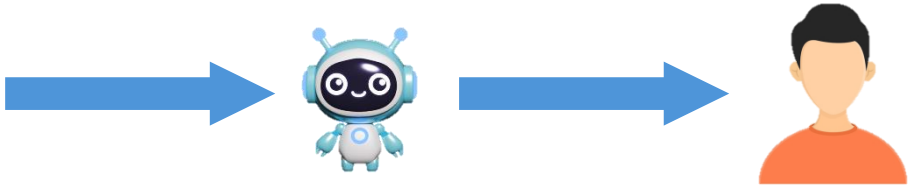
5 Experiments & Results

6 Conclusion

1 Task-Aware Semantic Map (TASMap)

- Why do we need a TASMap?

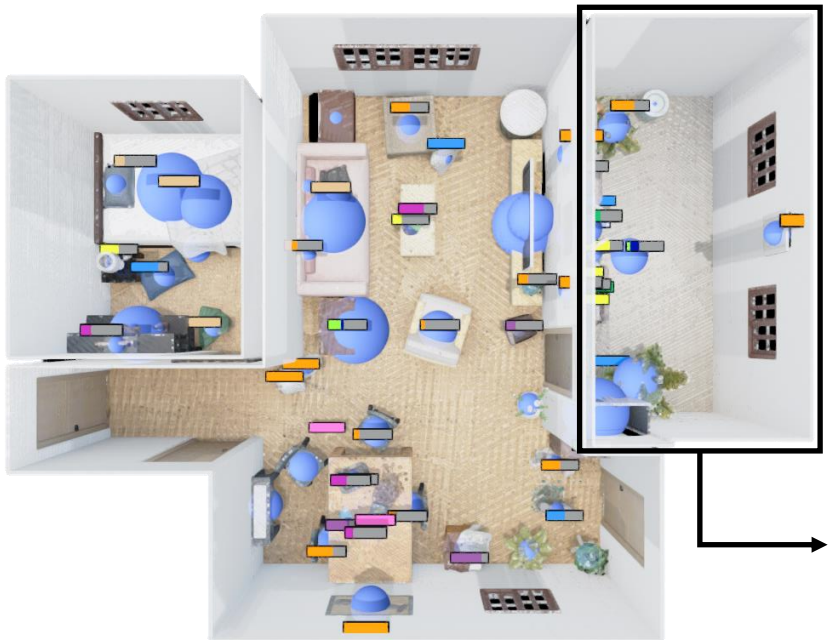
Explicit human commands



Autonomous generation of task goals

1 Task-Aware Semantic Map (TASMap)

- Why do we need a TASMap?



TASMap

I'm going to clean up around the stove

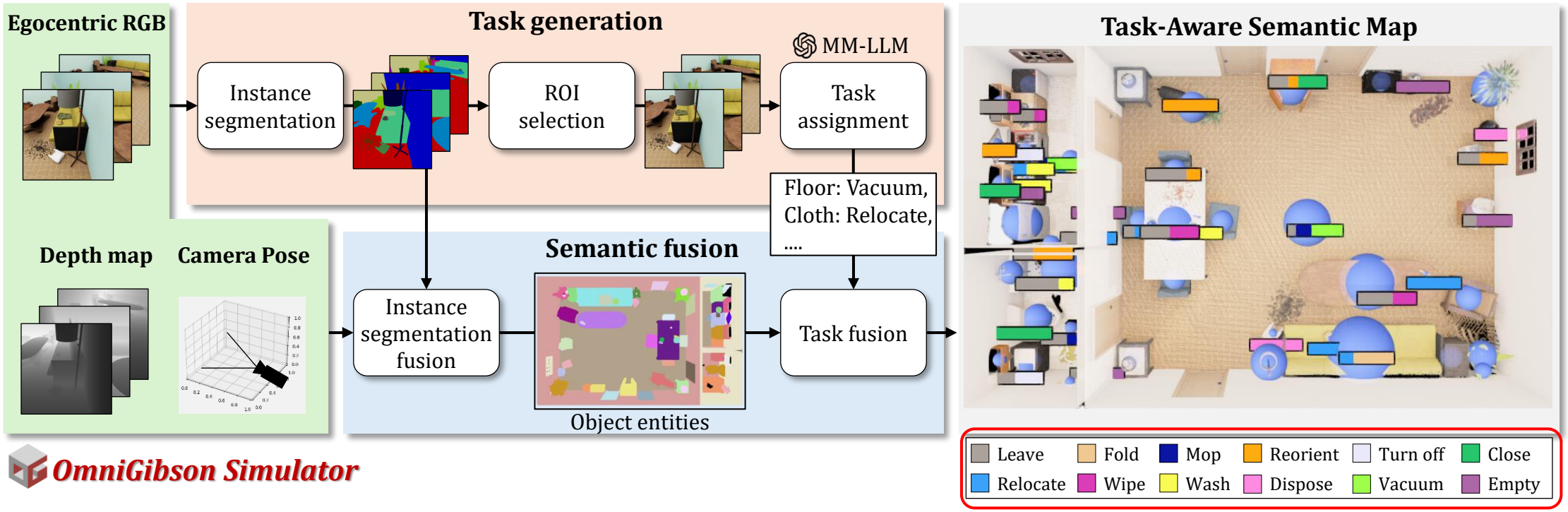
I'm going to collect utensil to wash the dishes

. . .

I'm going to find the stinky object

2 Framework

- *Overall Framework*

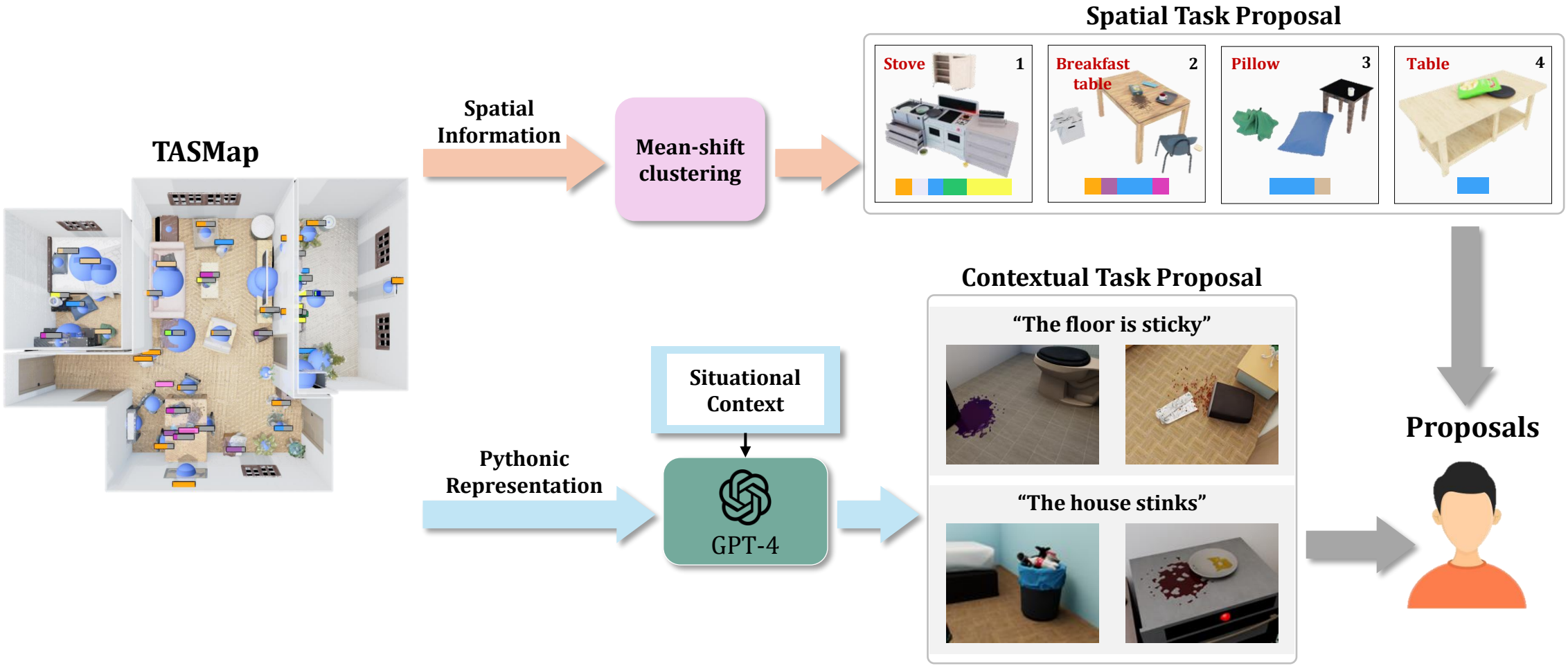


12 Tasks

[1] BEHAVIOR-1K: A Human-Centered, Embodied AI Benchmark with 1,000 Everyday Activities and Realistic Simulation

3 Application

- *Task Proposal Beyond Commands*



4 Dataset

a. Dataset Generation

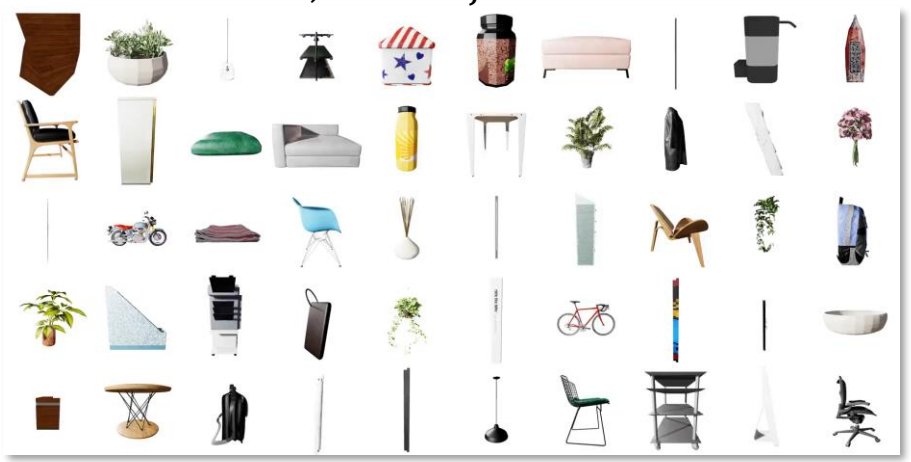
Need of an 'Untidy Indoor Scene' dataset → DATASET generation

3D-FRONT

- House structure
- Furniture placement

+

Behavior-1k: 9,000+ objects



- Mobile objects, particles system
- Omnigibson simulator

→

TASMap dataset

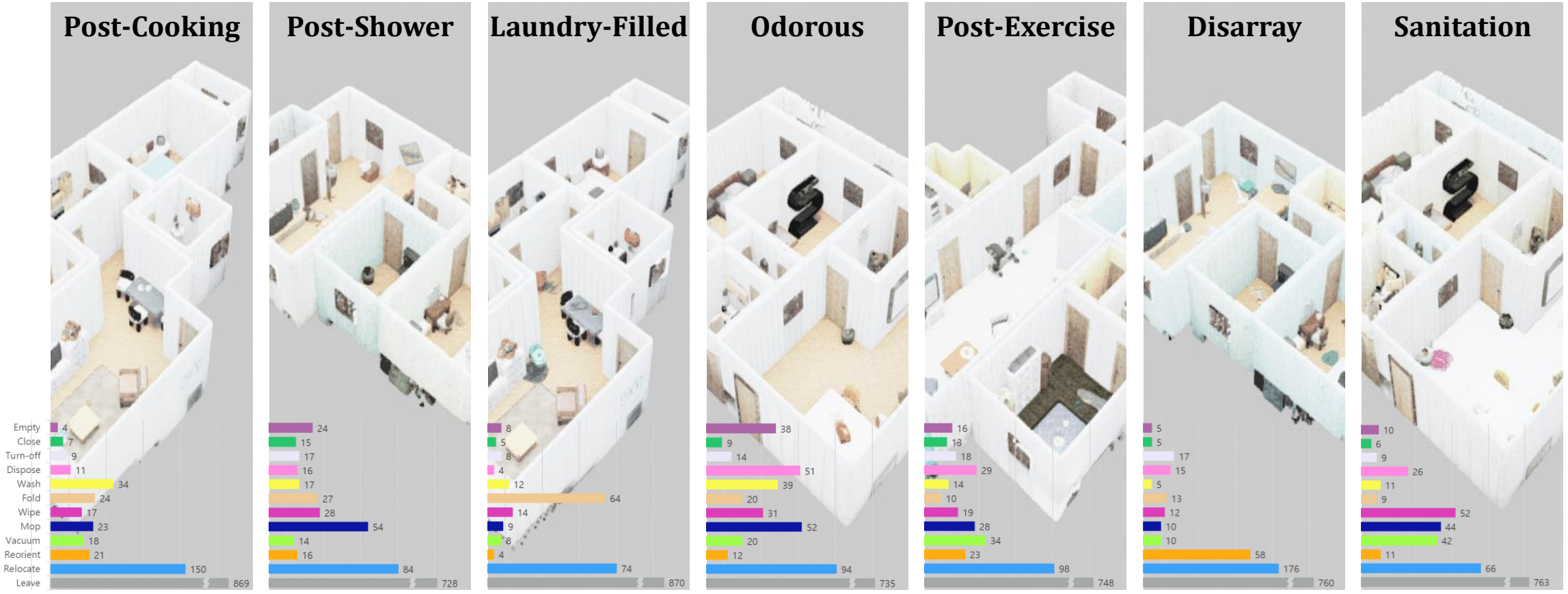
All object annotated with a list of ground truth tasks

[1] 3D-FRONT: 3D Furnished Rooms with layOuts and semantics
[2] BEHAVIOR-1K: A Human-Centered, Embodied AI Benchmark with 1,000 Everyday Activities and Realistic Simulation

4 Dataset

b. Concepts

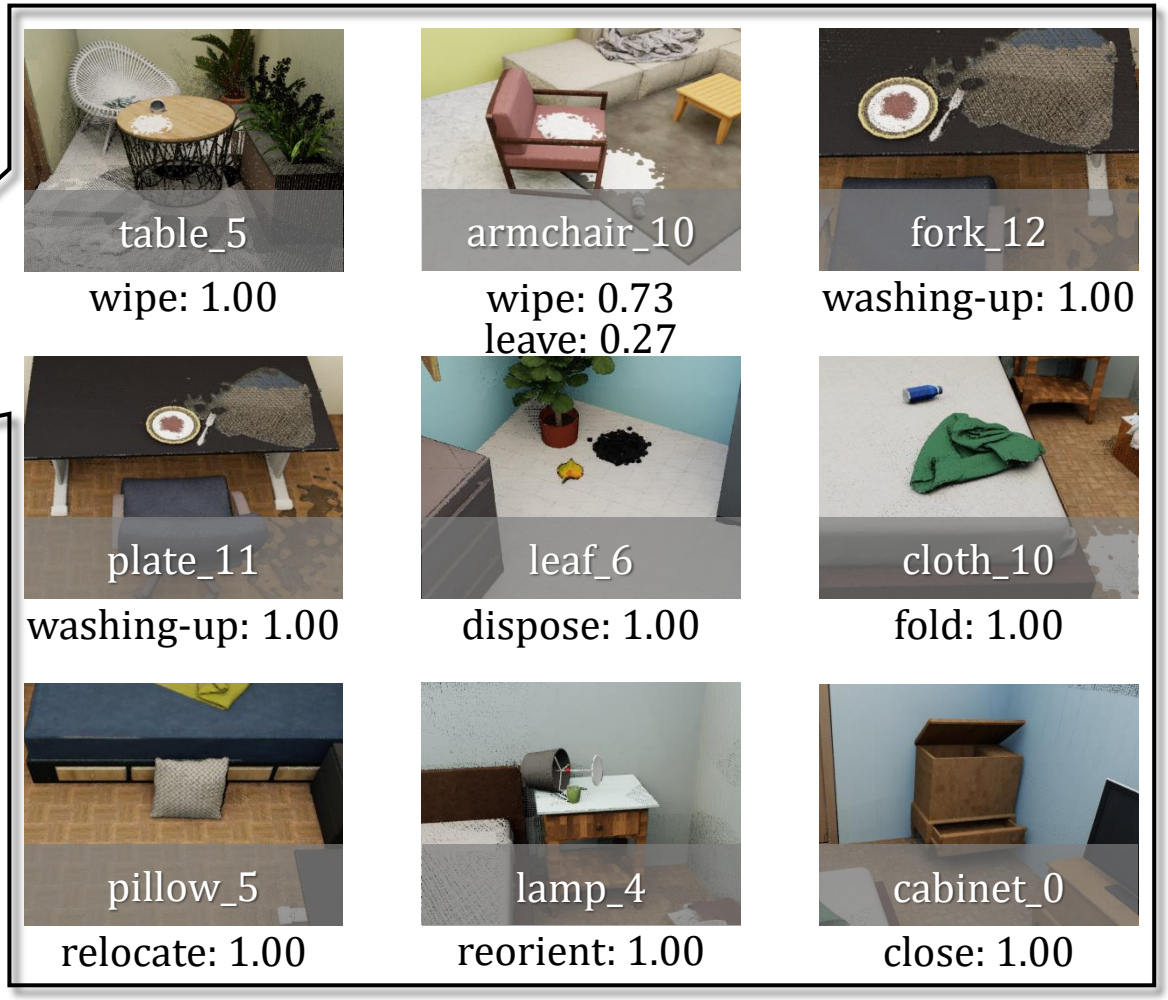
35 discrete ‘Houses’ (280 separate ‘Rooms’)



5 Experiments & Results

a. TASMap

Map #1: Post-Cooking



5 Experiments & Results

b. TASMap Evaluation

Multi-label Classification evaluation metrics

: *Class Precision (CP), Class Recall (CR), and Class F1 Score (CF1), Overall Precision (OP), Overall Recall (OR), and Overall F1 score (OF1)*

Evaluation on all 35 Scenes

Method	CP	CR	CF1	OP	OR	OF1
GPT-4 turbo	0.46	0.74	0.51	0.50	0.74	0.61
GPT-4o	**0.57**	**0.82**	**0.65**	**0.62**	**0.83**	**0.71**

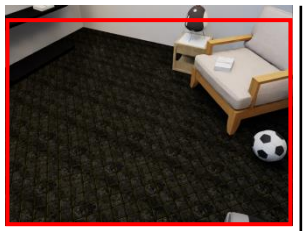
Inability to uniquely determine necessary tasks due to the diversity in human preference

Each object entity can have multiple tasks

Bowl_3
GT task:
Wash
Relocate

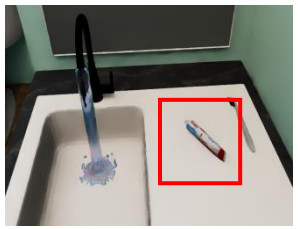
Table

Leave ↔ Wipe



Floor

Leave ↔ Vacuum



Toothpaste

Relocate ↔ Close

Toilet

Leave ↔ Wipe

5 Experiments & Results

c. Contextual Task Proposal Evaluation

TASMap

Contextual info — *"The floor is sticky"*

With *task-significant vector* ϕ

floor_0
Mop

floor_4
Mop

floor_6
Mop

floor_10
Vacuum

Without *task-significant vector* ϕ (Only label)

cloth_0
Leave

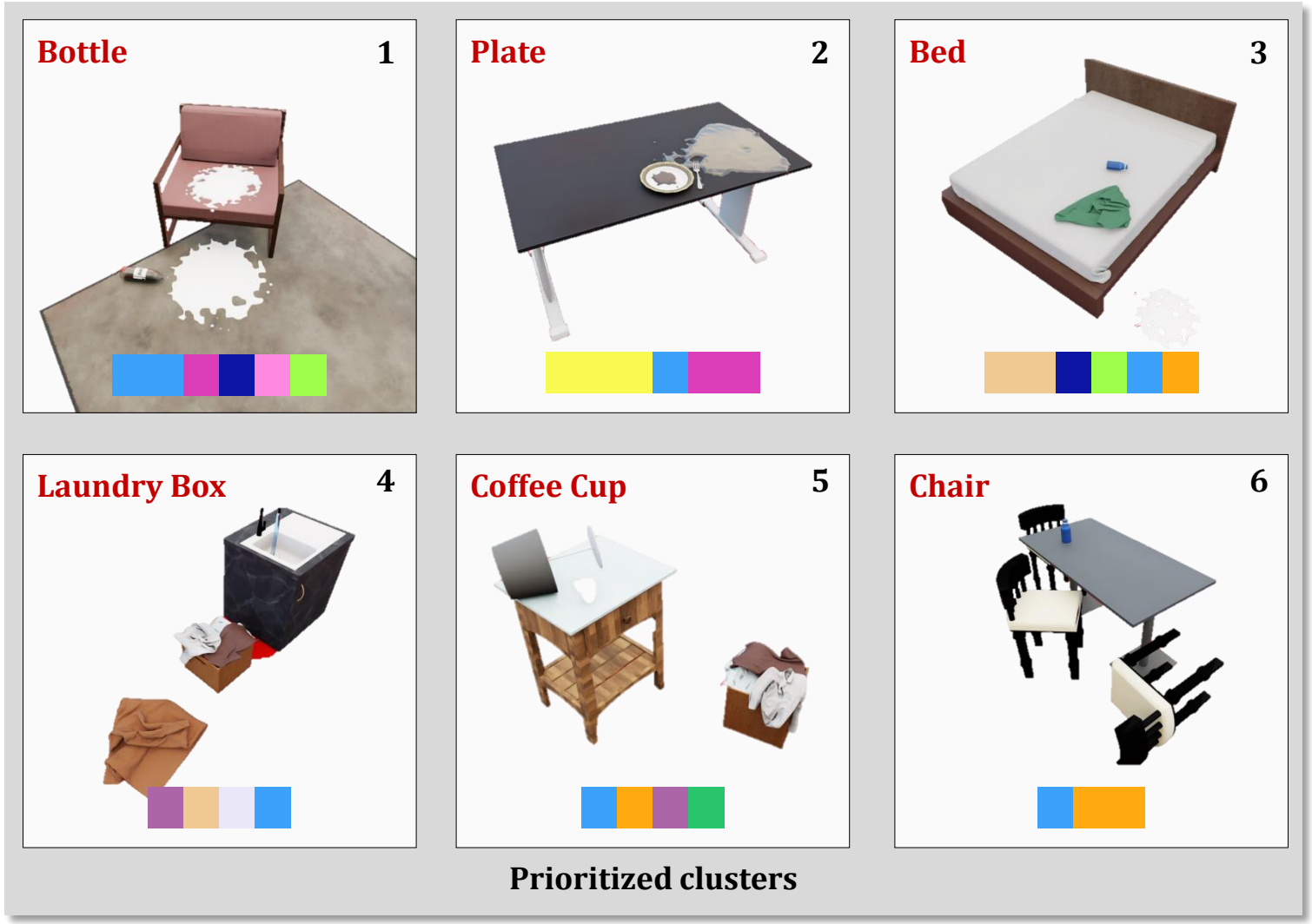
sink_8
Leave

soap_dispenser_5
Leave

5 Experiments & Results

d. Spatial Task Proposal Evaluation

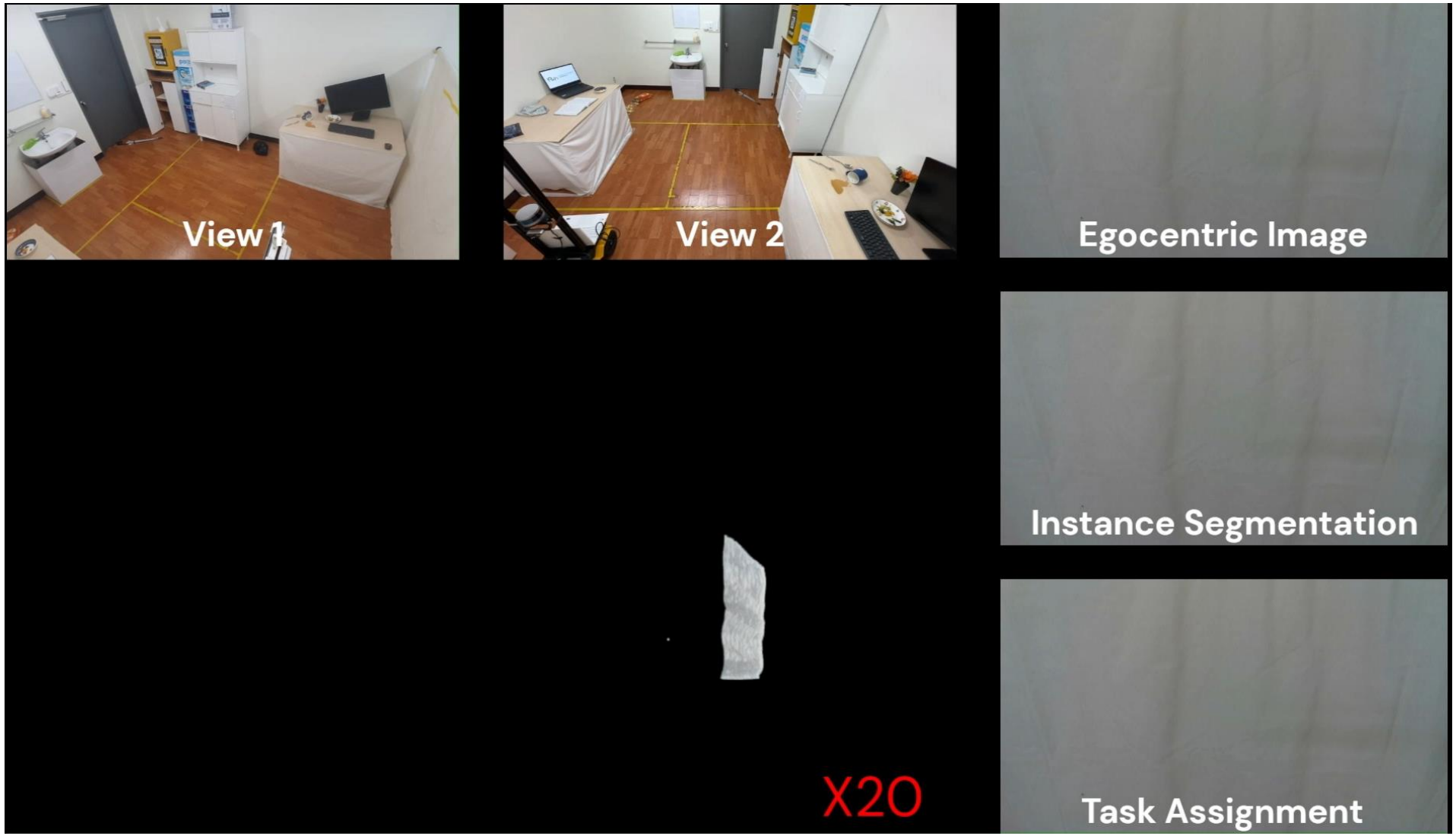
TASMap



Prioritized clusters

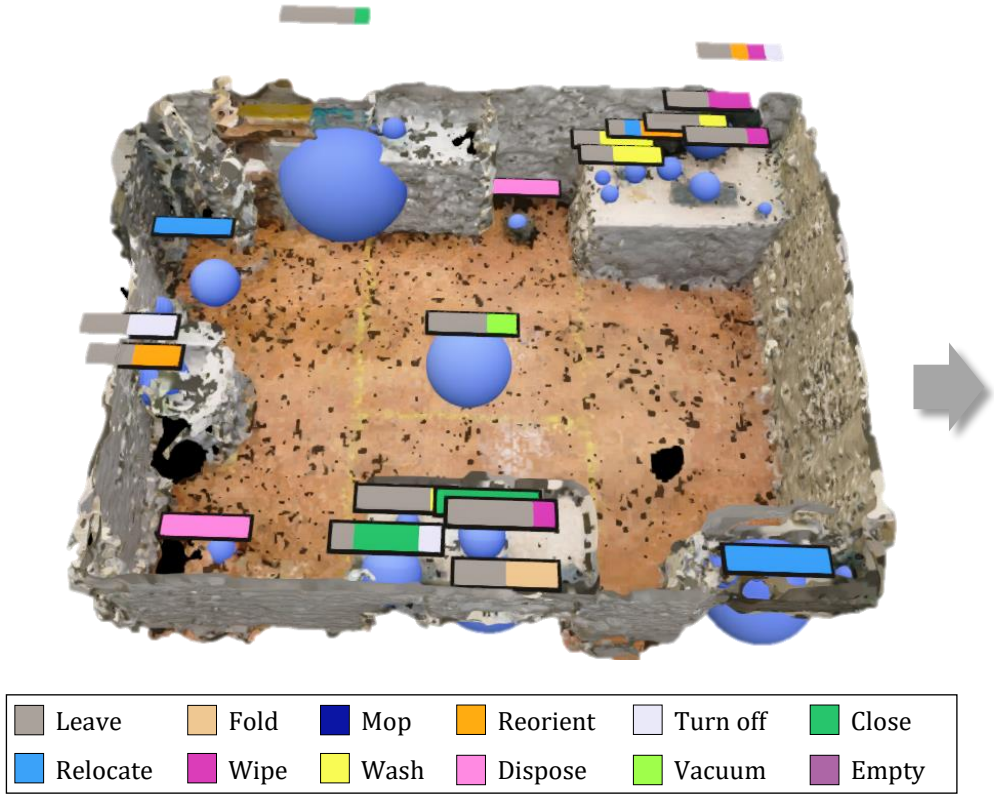
5 Experiments & Results

e. Real-world Experiments

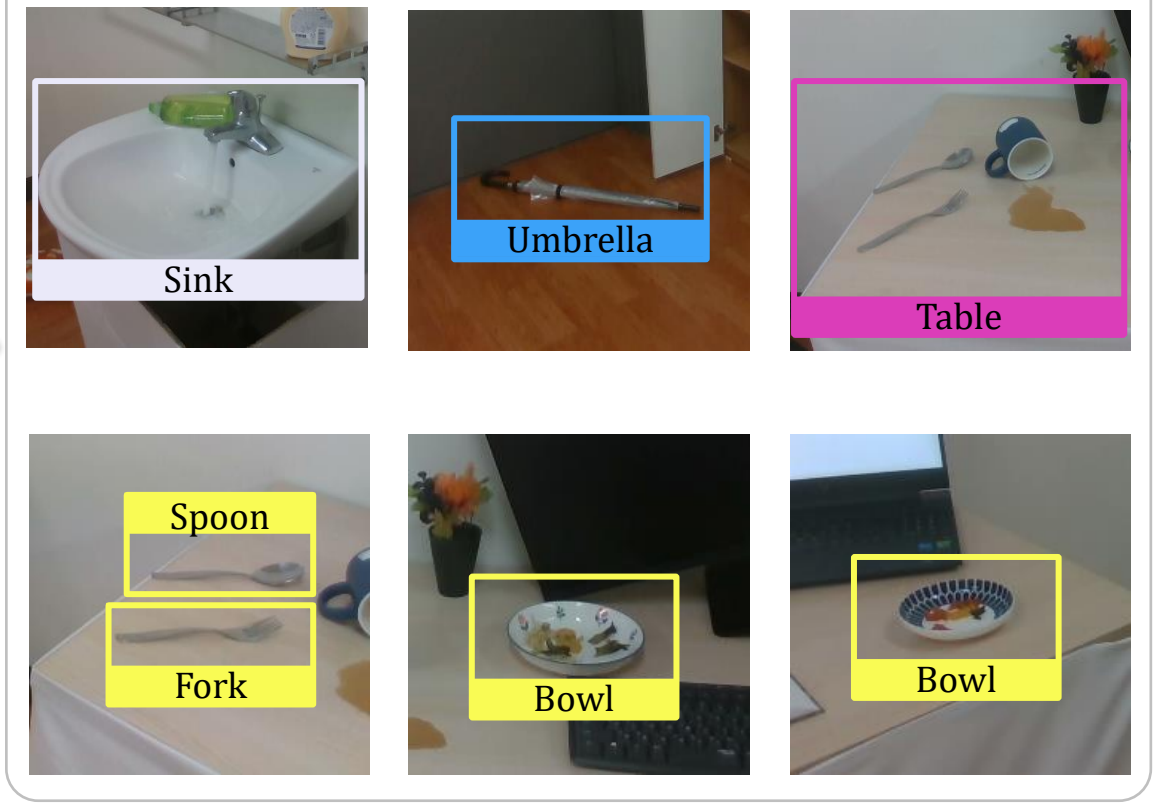


5 Experiments & Results

f. Real-world Experiments – Contextual Task Proposal



Contextual info "There is high humidity level in the house"



6 Conclusion

IEEE International Conference on Robotics and Automation (ICRA)

Task-Aware Semantic Map: Autonomous Robot Task Assignment Beyond Commands

Task-Aware Semantic Map

Task Proposal Beyond Commands :

- *Contextual Task Proposal*
- *Spatial Task Proposal*

Generation of 'Untidy Dataset'

Groundbreaking advancement in enabling robots to *autonomously* generate tasks

6 Conclusion

QnA

Task-Aware Semantic Map

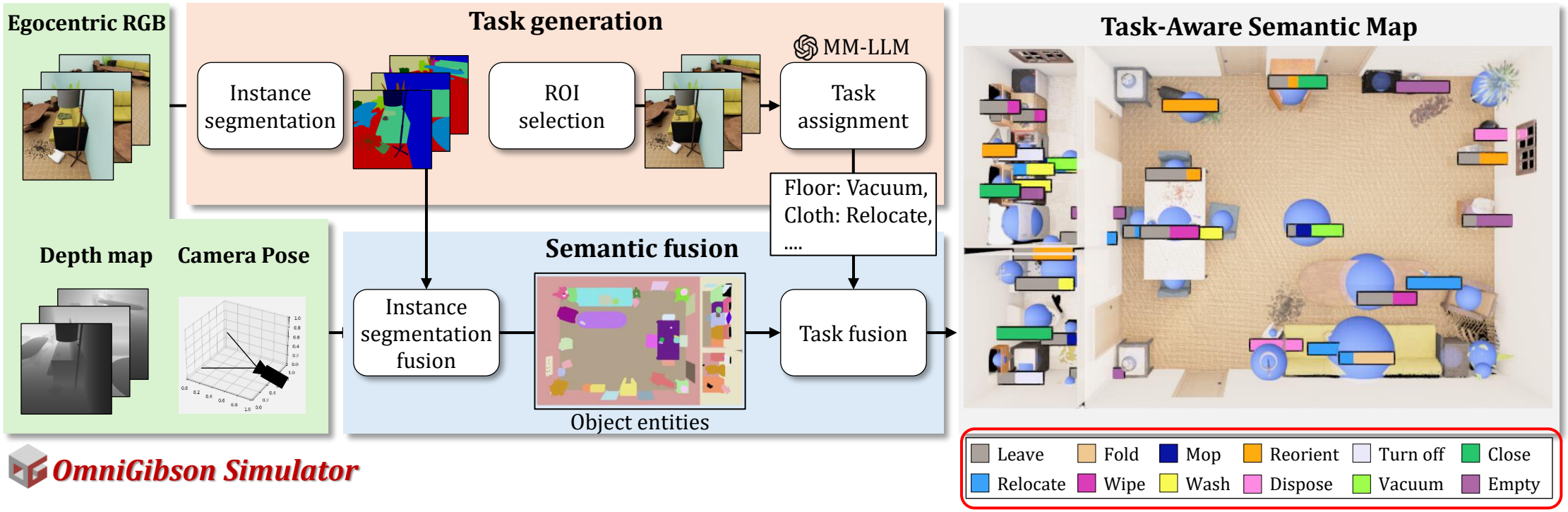
Autonomous Robot Task Assignment Beyond Commands

Thank You!

한양대학교
HANYANG UNIVERSITY

APPENDIX

- *Overall Framework*

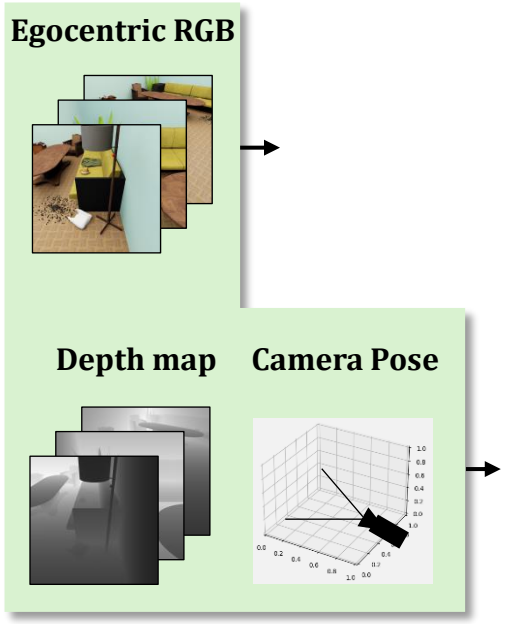


12 Tasks

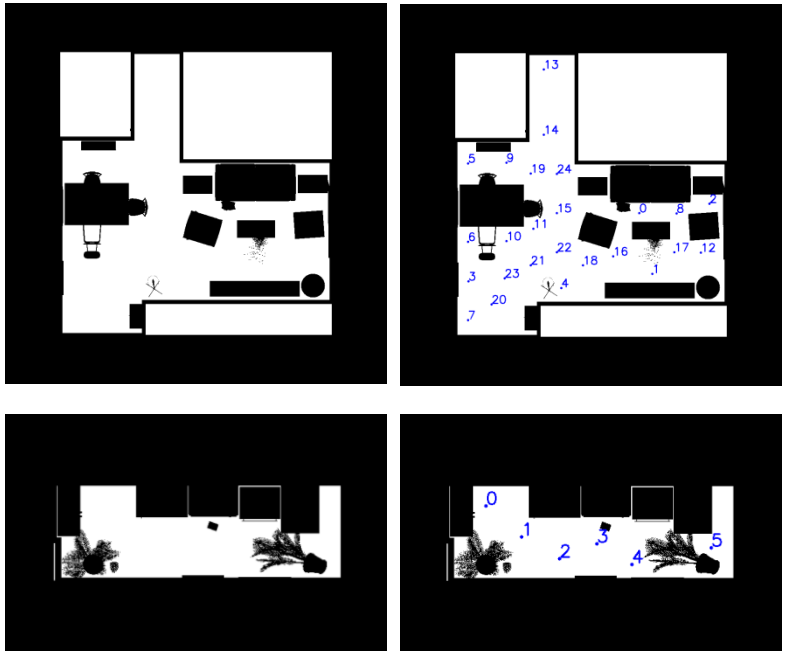
[1] BEHAVIOR-1K: A Human-Centered, Embodied AI Benchmark with 1,000 Everyday Activities and Realistic Simulation

I APPENDIX

- Framework - Input Data Generation



1. Generation of waypoints



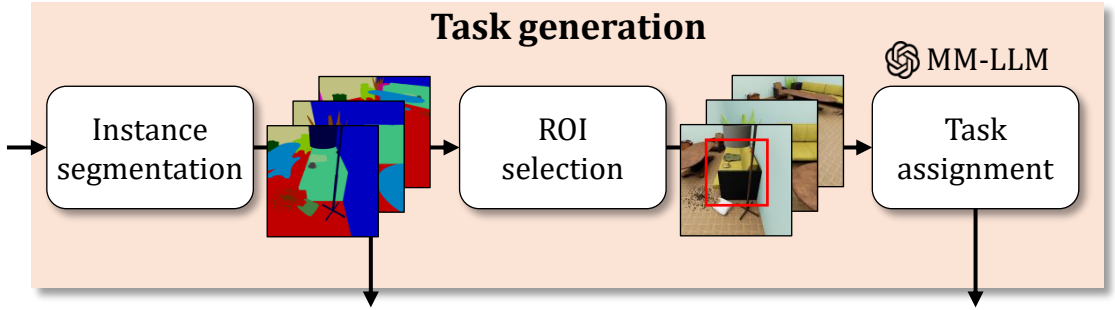
Limitation of waypoints : 4 ~ 25

2. Observation of the environment in 8 directions

Ensure sufficient data is observed

I APPENDIX

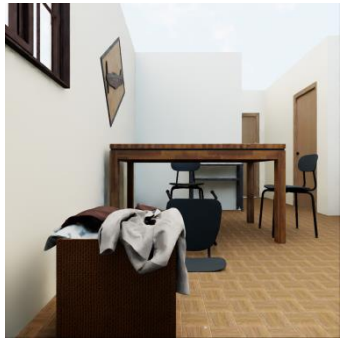
- *Framework - Task Generation Module*



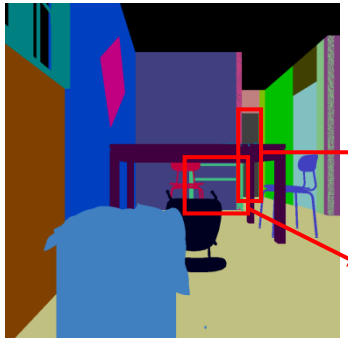
Instance Segmentation Submodule

Identifies objects in the image, $E_i^{seg} = (I, p^c, M_i, L_i)$

E_i^{seg} : segmented entity
I : image
p^c : camera pose
M_i : masks
L_i : labels



RGB Image



Instance Segmentation

NOT ALL ENTITIES CONTAIN SUFFICIENT INFORMATION FOR TASK ASSIGNMENT

Table

Door

Task Assignment Submodule

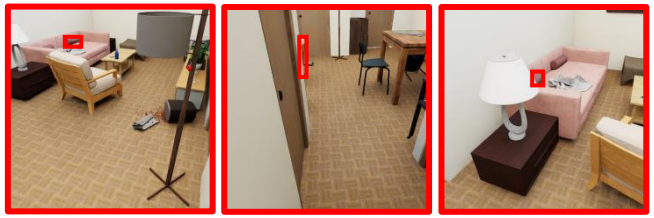
Allocates specific tasks to object within these selected ROIs, $E_i^{task} = (I, p^c, M_i, L_i, B_i, t_i)_{i=1}^{nR}$

E_i^{task} : tasked entity
I : image
p^c : camera pose
M_i : masks
L_i : labels
B_i : bounding boxes
t_i : suitable task

ROI Selection Submodule

Selects significant entities and ROIs, $(I, p^c, M_i, L_i, B_i)_{i=1}^{nR}$

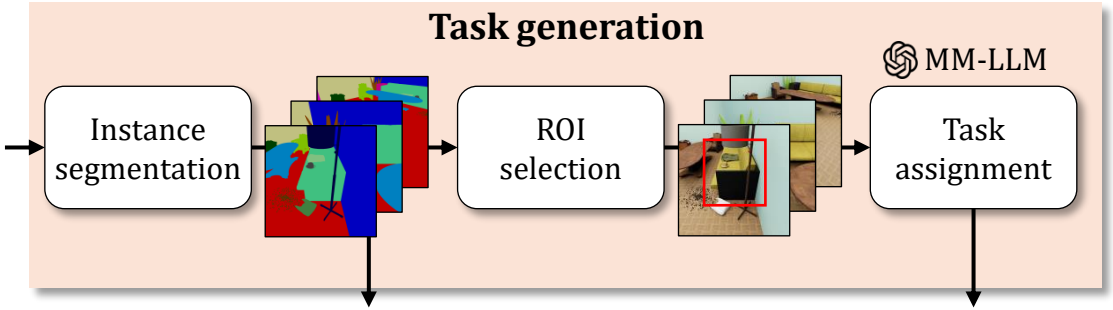
I : image
p^c : camera pose
M_i : masks
L_i : labels
B_i : bounding boxes



EXCLUSION

I APPENDIX

- *Framework - Task Generation Module*



Task Assignment Submodule

Allocates specific tasks to object within these selected ROIs, $E_i^{task} = (I, p^c, M_i, L_i, B_i, t_i)_{i=1}^{nR}$

E_i^{task} : tasked entity
I : image
p^c : camera pose
M_i : masks
L_i : labels
B_i : bounding boxes
t_i : suitable task

12 Tasks

TASK	*Criteria*
Leave as it is	The target is placed in an appropriate position
Relocate	The target is misplaced
Reorient	The target is in an appropriate location but is misaligned
Wipe	The target has smudges, spills, or debris on it
Vacuum	The target is a floor surface with particles or debris
Mop	The target is a wet floor surface

TASK	*Criteria*
Dispose	The target is no longer usable and should be thrown away
Fold	The target is crumpled or disheveled
Turn off	The target is unnecessarily turned on
Wash	The target has stains or food residues on it
Close	The target is unnecessarily open
Empty	The target's interior is full and needs to be emptied

APPENDIX

- Framework - Task Generation Module

Prompt with In-Context Learning

```
You are a robot that keeps the house clean and tidy.
Your role involves choosing the appropriate task for the target when it is not in its normal state, with a focus on the target rather than other objects.
Don't consider any potential issues, such as any unseen dirt, dust, stains, marks, particles or debris.
Follow Instructions step by step. Don't perform next instruction until the previous instruction is over.
```

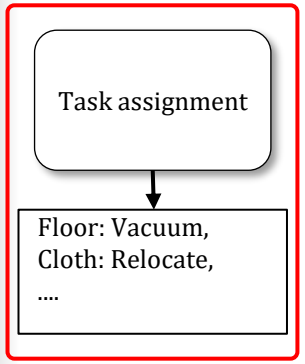
True few-shot examples

```
{few-shot example}

Target Object: {obj_name}
Task Candidates: [leave the (object) as it is, relocate the (object), reorient the (object), washing-up the (object), mop the (object), vacuum the (object), wipe the (object), fold the (object), close the (object), turn-off the (object), dispose of the (object), empty the (object)]
'(object)' should be replaced.
```

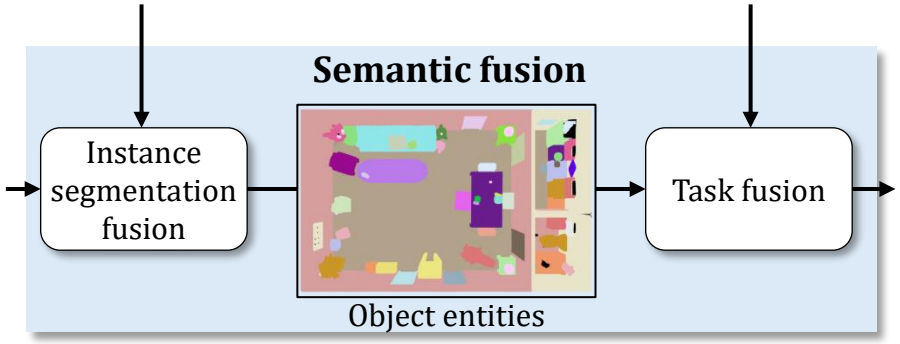
Reasoning steps & Chain-of-Thought

```
Instructions:
1. Target Existence: Find the {obj_name} in ROI {ROI}. If you found the {obj_name} in ROI {ROI}, answer "yes." If you couldn't find it, answer "no."
2. Target Description: Describe the state of the {obj_name}.
3. Tidying Phase:
- If the {obj_name} needs to be tidied, list the steps according to the 'Target Description'.
- 'task': what is executed to 'object'.
- Only the 'object' and one of 'Task Candidates' are included in the 'task', without any additional details or specific mentions.
- Don't consider any unseen potential issues.
4. Step Number:
- Check if Tidying Phase has step that is 'object: {obj_name}'.
- If 'object: {obj_name}' exists, reply with 'Step number' and if not, reply with 'None'.
5. Target's Task: 'task' in the 'Step number'. If 'Step Number' is none, reply with 'leave the {obj_name} as it is'.
6. Output JSON: Output in JSON format that contains the following keys: target_object, exists, description, task, reason according to previous instructions.
```



I APPENDIX

- *Framework - Semantic Fusion Module*

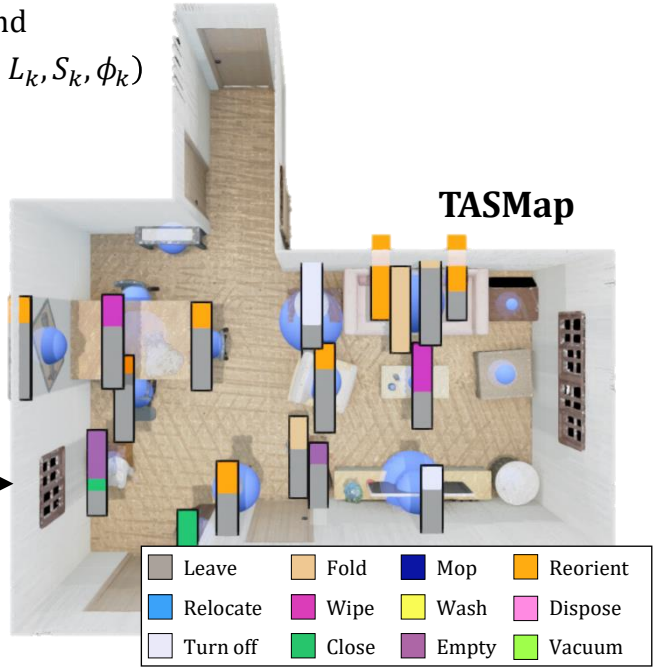


Task Fusion

Aggregates *tasked entities* corresponding to E_k^{obj} and determines the *task-significant vector* $\emptyset_k$, $E_k^{obj} = (L_k, S_k, \phi_k)$

E_k^{obj} : object entity
L_k : labels
S_k : segmented entity
Φ_k : task-significant vector

Floor: Vacuum,
Cloth: Relocate,
Chair: Reorient,
Table: Wipe,
Chair: Leave
....
Sofa: Leave



Instance Segmentation Fusion

Identifies the *segmented entities* generated by same object, fuse these entities , and generates object entities, $E_k^{obj} = (L_k, S_k, \Phi_k)$

E_k^{obj} : object entity
L_k : labels
S_k : segmented entity
(Φ_k : task-significant vector) → zero vector

II APPENDIX

- *Application – Contextual Task Proposal*

TASMap

Pythonic Representation

```
cloth_1 = Object("Cloth")

cloth_1.task({"Leave":0.2,
"Relocate": 0.8})


pot_pant_1 = Object("Pot plant")

pot_pant_1.task({"Leave":0.1,
"Relocate":0.2, "Reorient": 0.7})


bread_1 = Object("Bread")

bread_1.task({"Leave":0.1,
"Relocate":0.8, "Dispose":0.1})

. . .

                    Object label
floor_1 = Object("Floor")

floor_1.task({"Leave":0.4,
"Vacuum":0.2, "Mop":0.4})
```

Task-significant vector

Indirect information that can be obtained through sensors or given situationally

Context

The house is messy

GPT-4

Proposed objects

Cloth - Relocate

Pot plant - Reorient

Bread - Dispose

Floor - Mop

II APPENDIX

- Application – Spatial Task Proposal

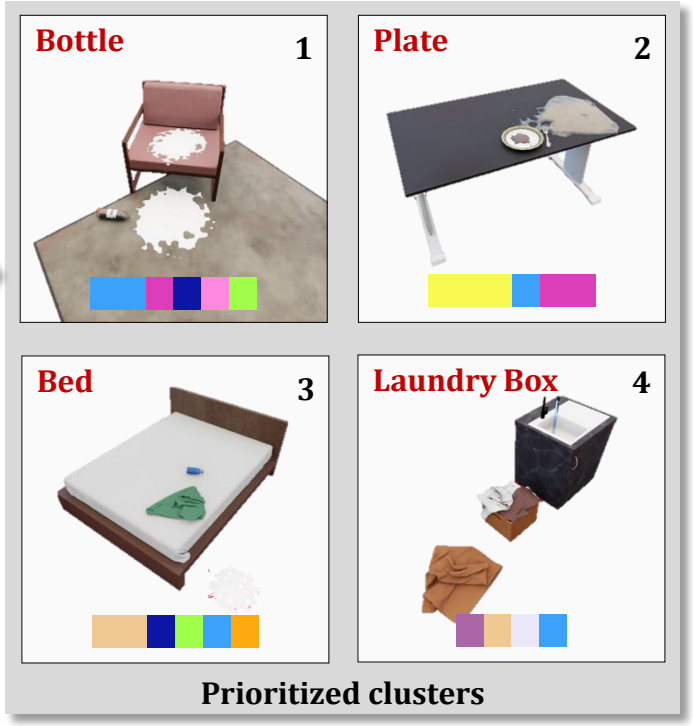
TASMap

Mean-shift clustering

clustering executed on room levels

Prioritization based on the assumption that areas near an object with the highest number of tasks is the most impactful

Prioritization on clusters

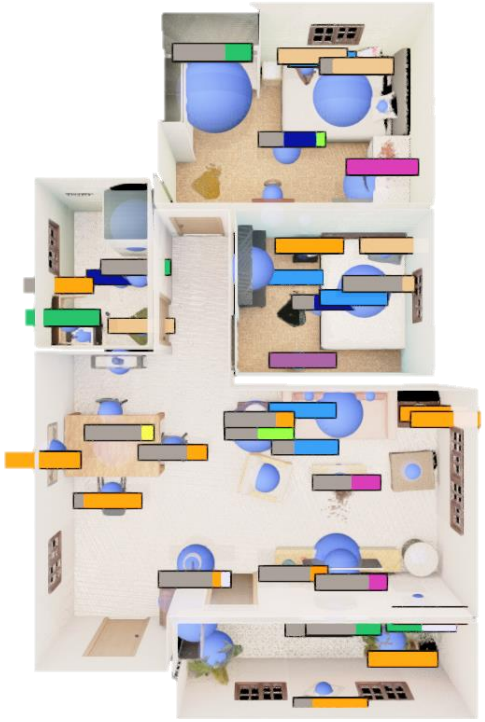


Prioritized clusters

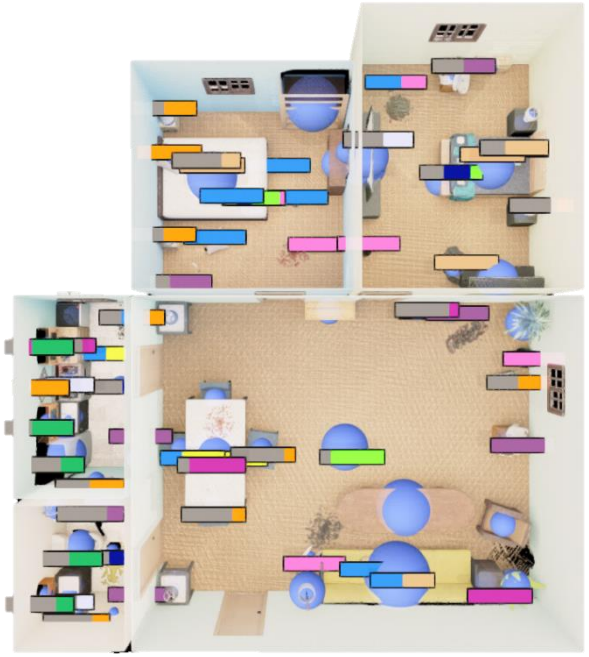
III APPENDIX

- Experiment – TASMap

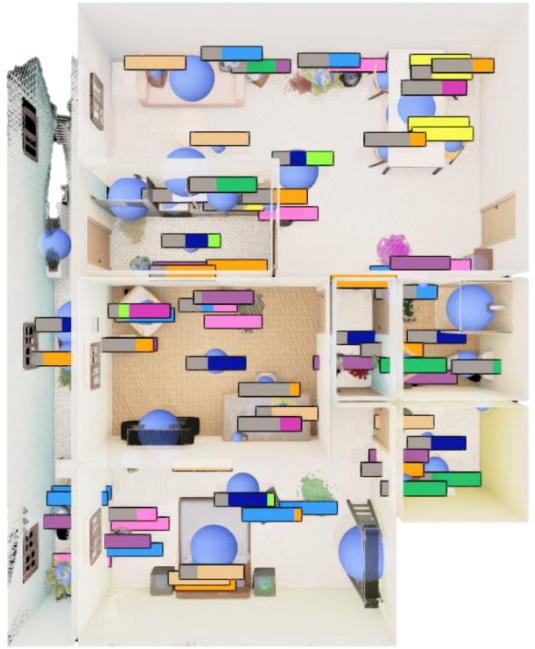
Map #1: Post-Cooking



Map #2: Post-Exercise



Map #3: Odorous



Map #4: Odorous