

427 Appendix

428 A Additional Results

429 A.1 Ablation Results of the Map Construction Modules for Region Classes

Region Precision-Recall for	Precision at Threshold [m]				Recall at Threshold [m]			
	0.5	1.0	2.0	3.0	0.5	1.0	2.0	3.0
RAM	0.49	0.51	0.54	0.57	0.06	0.13	0.40	0.55
RAM w/o ensemble	0.46	0.48	0.51	0.53	0.06	0.16	0.42	0.57
RAM w/o depth	0.48	0.51	0.54	0.56	0.06	0.19	0.42	0.56
RAM++	0.44	0.46	0.49	0.53	0.06	0.15	0.42	0.57

Table 4: Tag map ablations evaluated on precision and recall for region classes. For conciseness, crop ensemble filtering is referred to as “ensemble” and depth filtering as “depth”.

430 A.2 Tag Map Precision and Recall for Each Class in Matterport3D

Object Precision-Recall	Precision at Threshold [m]				Recall at Threshold [m]			
	0.1	0.5	1.0	2.0	0.1	0.5	1.0	2.0
bathtub	0.23	0.28	0.30	0.34	0.07	0.24	0.60	0.88
bed	0.53	0.61	0.65	0.69	0.05	0.41	0.68	0.94
cabinet	0.25	0.34	0.41	0.49	0.11	0.45	0.71	0.89
chair	0.39	0.53	0.62	0.72	0.18	0.44	0.63	0.79
chest_of_drawers	0.10	0.17	0.22	0.28	0.13	0.44	0.69	0.86
clothes	0.09	0.13	0.15	0.18	0.07	0.52	0.77	0.90
counter	0.34	0.41	0.46	0.52	0.11	0.36	0.62	0.81
cushion	0.21	0.40	0.51	0.58	0.09	0.36	0.59	0.82
fireplace	0.36	0.50	0.53	0.59	0.09	0.35	0.66	0.88
gym_equipment	0.49	0.60	0.71	0.78	0.06	0.16	0.33	0.65
picture	0.42	0.58	0.68	0.78	0.20	0.49	0.69	0.83
plant	0.26	0.36	0.43	0.51	0.13	0.43	0.64	0.82
seating	0.24	0.32	0.37	0.45	0.03	0.23	0.35	0.51
shower	0.21	0.29	0.35	0.43	0.10	0.40	0.72	0.90
sink	0.18	0.33	0.43	0.51	0.17	0.55	0.81	0.92
sofa	0.30	0.42	0.50	0.56	0.07	0.31	0.50	0.72
stool	0.06	0.09	0.12	0.17	0.15	0.42	0.58	0.76
table	0.30	0.44	0.51	0.61	0.19	0.53	0.73	0.88
toilet	0.45	0.55	0.59	0.67	0.08	0.44	0.74	0.91
towel	0.29	0.41	0.52	0.59	0.03	0.25	0.51	0.67
tv_monitor	0.16	0.31	0.39	0.49	0.17	0.46	0.73	0.86

Table 5: Precision-recall for common object classes evaluated over all 90 scenes of Matterport3D

Region Precision-Recall	Precision at Threshold [m]				Recall at Threshold [m]			
	0.5	1.0	2.0	3.0	0.5	1.0	2.0	3.0
balcony	0.26	0.31	0.38	0.43	0.00	0.00	0.14	0.45
bar	0.00	0.00	0.00	0.00	0.33	0.33	0.33	0.33
bathroom	0.30	0.33	0.35	0.38	0.21	0.38	0.68	0.86
bedroom	0.47	0.49	0.52	0.54	0.06	0.16	0.56	0.85
classroom	0.60	0.60	0.60	0.60	0.00	0.00	0.50	0.50
closet	0.08	0.09	0.13	0.16	0.16	0.23	0.48	0.63
dining room	0.47	0.50	0.53	0.56	0.03	0.16	0.60	0.81
garage	0.69	0.72	0.72	0.77	0.00	0.00	0.29	0.57
hallway	0.54	0.59	0.68	0.73	0.05	0.14	0.43	0.67
kitchen	0.38	0.40	0.42	0.45	0.09	0.29	0.69	0.88
laundryroom/mudroom	0.68	0.70	0.72	0.72	0.00	0.17	0.63	0.89
library	1.00	1.00	1.00	1.00	0.00	0.00	0.50	0.50
living room	0.40	0.42	0.47	0.51	0.07	0.19	0.43	0.61
meetingroom/conferenceroom	0.32	0.36	0.36	0.44	0.04	0.08	0.12	0.28
office	0.36	0.38	0.44	0.48	0.06	0.14	0.37	0.43
porch/terrace/deck/driveway	0.54	0.55	0.58	0.61	0.05	0.11	0.36	0.52
rec/game	1.00	1.00	1.00	1.00	0.00	0.00	0.06	0.06
spa/sauna	0.34	0.37	0.37	0.40	0.02	0.07	0.16	0.23
stairs	0.26	0.29	0.35	0.38	0.08	0.20	0.53	0.72
tv	0.71	0.71	0.82	0.88	0.00	0.31	0.62	0.62
utilityroom/toolroom	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
workout/gym/exercise	0.89	0.89	0.89	0.92	0.00	0.00	0.38	0.75

Table 6: Precision-recall for region classes evaluated over all 90 scenes of Matterport3D

431 **B Tag Map Parameters**

432 The map construction and localization parameters used for all experiments are presented in Table 7
433 and 8 respectively.

Parameter	Value
Tagging model	ram_swin_large [50]
Crop ensemble border crop percentages	[5%, 10%]
Depth filter mean threshold	0.6 m
Depth filter median threshold	0.6 m

Table 7: Tag map construction parameters

Parameter	Value
Viewpoint near plane distance	0.2 m
Viewpoint far plane distance	80 th percentile of depth values
Voxel size	0.2 m
DBSCAN eps	0.4 m
DBSCAN minimum points	5
Normalized votes clustering thresholds	[0.00, 0.25, 0.50, 0.75]

Table 8: Tag map localization parameters

434 Note that the clustering thresholds on the voxel votes are set relative to the normalized votes, i.e. the
435 voxel votes normalized by the maximum votes of any voxel.

You are a helpful robot assistant.

To give you some knowledge of your environment, you are provided with a set of tags. Each tag describes something that's likely to be found in the environment. You can use these tags to help you with assisting the user.

Note that the tags are not perfect. Some tags may be incorrect and the tags do not cover everything in the environment. Therefore, try to pick the tag which you are most confident in.

You can query for the regions in the environment where a tag is localized and also get the confidence level for each region.

When making a statement about a tag, do not say that the tag is definitely in the environment, instead reference the confidence level.

If there are no tags directly related to the user's request, suggest tags which may be related to the user's request and ask the user if they would like to know more about the suggested tags.

List of tags in the format '[id] - [tag]'

0 - <tag 0>

1 - <tag 1>

⋮

N - <tag N>

D Details of the Evaluation Implementation

D.1 Grid Graph Construction

Computing the P2E and E2P requires solving for the shortest paths within a scene, which we compute approximately using a 3D grid graph that spans the scene without collision with the scene geometry. Note that the grid graph connects nodes without consideration of traversability. For example, the free space above a table would be connected but would not be traversable by a wheeled robot. An overview of the steps to construct a grid graph is given in Algorithm 1.

Algorithm 1 Scene Grid Graph Generation

```
1: Input: Scene Mesh  $\mathcal{M}$ 
2: Output: Set of nodes  $\mathcal{N}$  and edges  $\mathcal{E}$ 
3: Determine the min and max bounds of  $\mathcal{M}$ 
4: Generate a set of nodes  $\mathcal{N}$  following a grid spanning these bounds
5: for each node  $n \in \mathcal{N}$  do
6:   if InsideScene( $n, \mathcal{M}$ ) is false then
7:     Remove  $n$  from  $\mathcal{N}$ 
8:   end if
9: end for
10: Initialize empty set of edges  $\mathcal{E}$ 
11: for each node  $n \in \mathcal{N}$  do
12:   for each immediate neighbor  $n'$  of  $n$  in the grid do
13:     if  $(n, n') \notin \mathcal{E}$  and CollisionFree( $n, n', \mathcal{M}$ ) is true then
14:       Add edge  $(n, n')$  to  $\mathcal{E}$ 
15:     end if
16:   end for
17: end for
18: Return  $\mathcal{N}, \mathcal{E}$ 
```

The grid graphs are generated at a grid resolution of 0.5 m. We check if a point is inside the scene by checking its nearest neighboring vertices. First, if the mean distance to these vertices is greater than a threshold of 2 m we consider the point outside the scene. Second, we use the normals of each neighboring vertex to check if the point is locally inside the mesh at that vertex by computing the dot product of the vertex-to-point vector and the normal. We average the dot products across all neighboring vertices and consider the point outside the mesh if the average exceeds a threshold of 0. To evaluate if two points are collision-free, we cast a ray from one to the other and check for collisions against the scene mesh.

D.2 Assigning Grid Graph Nodes

The assignment of grid graph nodes to instance proposals is given in Algorithm 2. For assigning nodes to labeled entity instances, the algorithm differs depending on whether the entity is an object or a region, as outlined in Algorithms 3 and 4 respectively.

The node assignment algorithms differ in the use of inflation to assign additional nodes. In the case of instance proposals, inflation is only used when the proposal does not contain any nodes. Here we set the inflation amount δ such that the extent of the inflated proposal is at least larger than the space between nodes in the grid graph. For object labels, inflation is always used to assign additional nodes. In this case, we set δ to capture nodes 1 m away from the label bounding box, following the success criteria of the Habitat ObjectNav Challenge [2]. Lastly, no inflation is used for region labels as they are large enough to contain at least some nodes.

If no nodes can be assigned to an instance proposal or labeled entity, then the P2E or E2P cannot be computed. In these cases, we ignore that instance proposal or labeled entity when later computing the precision and recall metrics.

Algorithm 2 Assign Grid Graph Nodes to an Instance Proposal

```
1: Input: Set of nodes  $\mathcal{N}$ , Instance proposal bounding box  $P$ , Scene mesh  $\mathcal{M}$ , Inflation amount  $\delta$ 
2: Output: Set of nodes  $\mathcal{N}_{\text{result}}$ 
3: Initialize  $\mathcal{N}_{\text{result}} \leftarrow \emptyset$ 
4: Find nodes  $\mathcal{N}_{\text{in}} \subseteq \mathcal{N}$  that are contained within  $P$ 
5: if  $\mathcal{N}_{\text{in}} \neq \emptyset$  then
6:   return  $\mathcal{N}_{\text{in}}$ 
7: end if
8: Inflate  $P$  by a fixed amount  $\delta$  to obtain a new bounding box  $P'$ 
9: Find nodes  $\mathcal{N}_{\text{in}} \subseteq \mathcal{N}$  that are contained within  $P'$ 
10: for each node  $n \in \mathcal{N}_{\text{in}}$  do
11:   Find the nearest point  $p \in P$  to  $n$ 
12:   if  $\text{CollisionFree}(n, p, \mathcal{M})$  is true then
13:     Add  $n$  to  $\mathcal{N}_{\text{result}}$ 
14:   end if
15: end for
16: return  $\mathcal{N}_{\text{result}}$ 
```

Algorithm 3 Assign Grid Graph Nodes to a Labeled Object Instance

```
1: Input: Set of nodes  $\mathcal{N}$ , Object label bounding box  $O$ , Scene mesh  $\mathcal{M}$ , Inflation amount  $\delta$ 
2: Output: Set of nodes  $\mathcal{N}_{\text{result}}$ 
3: Find nodes  $\mathcal{N}_{\text{in}} \subseteq \mathcal{N}$  that are contained within  $O$ 
4:  $\mathcal{N}_{\text{result}} \leftarrow \mathcal{N}_{\text{in}}$ 
5: Inflate  $O$  by a fixed amount  $\delta$  to obtain a new bounding box  $O'$ 
6: Find additional nodes  $\mathcal{N}'_{\text{in}} \subseteq \mathcal{N}$  that are contained within  $O' \setminus O$ 
7: for each node  $n \in \mathcal{N}'_{\text{in}}$  do
8:   Find the nearest point  $p \in O$  to  $n$ 
9:   if  $\text{CollisionFree}(n, p, \mathcal{M})$  is true then
10:    Add  $n$  to  $\mathcal{N}_{\text{result}}$ 
11:   end if
12: end for
13: return  $\mathcal{N}_{\text{result}}$ 
```

Algorithm 4 Assign Grid Graph Nodes to a Labeled Region Instance

```
1: Input: Set of nodes  $\mathcal{N}$ , Region label bounding box  $R$ , Scene mesh  $\mathcal{M}$ 
2: Output: Set of nodes  $\mathcal{N}_{\text{result}}$ 
3: Find nodes  $\mathcal{N}_{\text{in}} \subseteq \mathcal{N}$  that are contained within  $R$ 
4:  $\mathcal{N}_{\text{result}} \leftarrow \mathcal{N}_{\text{in}}$ 
5: return  $\mathcal{N}_{\text{result}}$ 
```

466 D.3 Mapping Object Class Labels to Tags

467 Matterport3D includes a raw class label for all labeled objects. For example the raw labels “arm
468 chair” and “bean bag chair” are grouped into the common object class of “chair”. Both “arm chair”
469 and “bean bag chair” are within the set of classes able to be recognized by the tagging model.
470 Because of this, for each common object class, we identify the corresponding raw classes and map
471 them to matching tags within the vocabulary of the tagging model. We also manually identified tags
472 that corresponded to a common object class and included them in the mapping. The mapping for the
473 common object classes to tags are defined as follows:

474 **bathhtub:** bath, jacuzzi

475 **bed:** bed, bed frame, bunk bed, canopy bed, cat bed, dog bed, futon, hammock, headboard, hospital
476 bed, infant bed, mattress

477 **cabinet:** armoire, bathroom cabinet, cabinet, cabinetry, closet, file cabinet, kitchen cabinet,
478 medicine cabinet, side cabinet, tv cabinet, wine cabinet

479 **chair:** armchair, beach chair, bean bag chair, beanbag, chair, computer chair, feeding chair, folding
480 chair, office chair, rocking chair, swivel chair, throne

481 **chest_of_drawers:** bureau, drawer, dresser, nightstand

482 **clothes:** baby clothe, baseball hat, bathrobe, bathroom accessory, bikini, bikini top, blouse, christ-
483 mas hat, cloak, clothing, coat, cocktail dress, corset, costume, cowboy hat, crop top, denim jacket,
484 dress, dress hat, dress shirt, dress shoe, dress suit, evening dress, fur coat, gown, halter top, hat,
485 headdress, headscarf, hoodie, jacket, jeans, jockey cap, kilt, kimono, lab coat, lace dress, laundry,
486 leather jacket, maxi dress, miniskirt, overcoat, pants, pantyhose, polo neck, polo shirt, raincoat, robe,
487 safety vest, scarf, shirt, ski jacket, sports coat, straw hat, sun hat, suspenders, sweat pant, sweater,
488 sweatshirt, t shirt, t-shirt, trench coat, underclothes, vest, visor, waterproof jacket, wedding dress,
489 wrap dress

490 **counter:** bar, buffet, counter, counter top, island, kitchen counter, kitchen island, wet bar

491 **cushion:** pillow, throw pillow

492 **fireplace:** fireplace, mantle

493 **gym.equipment:** barbell, dumbbell, stationary bicycle, training bench, treadmill, weight

494 **picture:** art, art print, couple photo, decorative picture, drawing, family photo, group photo, movie
495 poster, oil painting, photo, photo frame, picture, picture frame, portrait, poster, publicity portrait,
496 reflection, wedding photo

497 **plant:** bush, flower, grass, houseplant, plant, tree

498 **seating:** bench, church bench, park bench, seat, window seat

499 **shower:** shower, shower door, shower head

500 **sink:** basin, bathroom sink, sink

501 **sofa:** couch, loveseat

502 **stool:** bar stool, footrest, music stool, step stool, stool

503 **table:** altar, billiard table, changing table, cocktail table, computer desk, dinning table, foosball,
504 glass table, kitchen table, office desk, picnic table, poker table, round table, side table, stand, table,
505 vanity, workbench, writing desk

506 **toilet:** bidet, toilet bowl, toilet seat

507 **towel:** bath towel, beach towel, face towel, hand towel, paper towel, towel

508 **tv_monitor:** bulletin board, computer monitor, computer screen, display, monitor, television, white-
509 board

510 **D.4 Mapping Region Class Labels to Tags**

511 We map region labels directly to corresponding tags in the vocabulary of the tagging
512 model whenever possible. Some region labels are made up of multiple concepts such as
513 “porch/terrace/deck/driveway”. In these cases, we try to map each concept to a tag if possible.

514 **balcony:** balcony

515 **bar:** bar

516 **bathroom:** bathroom

517 **bedroom:** bedroom

518 **classroom:** classroom

519 **closet:** closet

520 **dining room:** dining room

521 **garage:** garage

522 **hallway:** hallway

523 **kitchen:** kitchen

524 **laundryroom/mudroom:** laundry room

525 **library:** library

526 **living room:** living room

527 **meetingroom/conferenceroom:** meeting room

528 **office:** home office, office

529 **porch/terrace/deck/driveway:** deck, driveway, porch, terrace

530 **rec/game:** recreation room

531 **spa/sauna:** sauna

532 **stairs:** stairs, stairwell

533 **tv:** cinema, home theater, theater

534 **utilityroom/toolroom:** utility room

535 **workout/gym/exercise:** gym

536 We relabeled regions labeled “familyroom” and “lounge” as “living room” since these labels de-
537 scribed analogous regions. Similarly, we also relabeled regions labeled “toilet” as “bathroom”.

E Label to Text Mappings for Embedding-Based Methods

To evaluate embedding-based methods, the class labels are mapped to strings which are then encoded into text embeddings for querying the map embeddings.

E.1 OpenScene and OpenMask3D

We followed the authors of OpenScene and OpenMask3D and applied the prompting method of “a {} in a scene” for object class labels. We modified the prompt to be grammatically correct for the following object classes:

chest_of_drawers: a chest of drawers in a scene

clothes: clothes in a scene

gym_equipment: gym equipment in a scene

tv_monitor: a television monitor in a scene

For region classes, we directly used the label as the string, except for the following classes where we modified the string to either be more grammatically correct or more accurately reflect the class semantics:

laundryroom/mudroom: laundry room or mudroom

meetingroom/conferenceroom: meeting room or conference room

porch/terrace/deck/driveway: porch or terrace or deck or driveway

rec/game: recreation or game room

spa/sauna: spa or sauna

tv: cinema or home theater or theater

utilityroom/toolroom: utility room or tool room

workout/gym/exercise: gym

E.2 CLIP Viewpoint Retrieval

For viewpoint retrieval using CLIP, we mapped each label to a string and used the ImageNet80 ensemble prompting method from Radford et al. [6] to create the text embedding.

For object classes, we used the label directly as the string, except for the following classes where we modified the string to be more grammatically correct or more accurately reflect the class semantics:

chest_of_drawers: chest of drawers

counter: countertop

gym_equipment: gym equipment

tv_monitor: television monitor

Similarly for region classes, we used the label directly as the string except for the following classes:

laundryroom/mudroom: laundry room or mudroom

meetingroom/conferenceroom: meeting room or conference room

porch/terrace/deck/driveway: porch or terrace or deck or driveway

rec/game: recreation or game room

spa/sauna: spa or sauna

575 **tv:** cinema or home theater or theater

576 **utilityroom/toolroom:** utility room or tool room

577 **workout/gym/exercise:** gym