

Topo-Field: Topometric mapping with Brain-inspired Hierarchical Layout-Object-Position Fields

A APPENDIX

A.1 SCENE PARTITION EXAMPLE

The scene can be partitioned into different regions using walls as dividers and lines can be aligned to these walls. This is similar in most scenarios, making the annotation of scene regions a straightforward task as shown in Fig. A1.

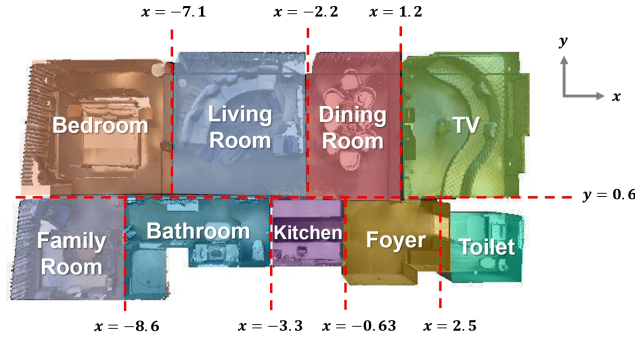


Figure A1: Using walls as dividers to associate lines with them, the scene can be divided into various regions and 3D points can be labeled with related regions easily.

A.2 VISION-LANGUAGE EMBEDDINGS SIMILARITY OF REGION AND OBJECTS

To demonstrate that the relationship of the vision-language and semantic embeddings for different regions is related to our intuition, we compare the similarity in region-region and object-region form and show the results in Fig. A2. It can be seen that based on general knowledge, cognitively related regions (e.g., the dining room and kitchen) and object-region pairs (e.g., sink and kitchen) are also more correlated in the vision-language and semantic feature spaces.

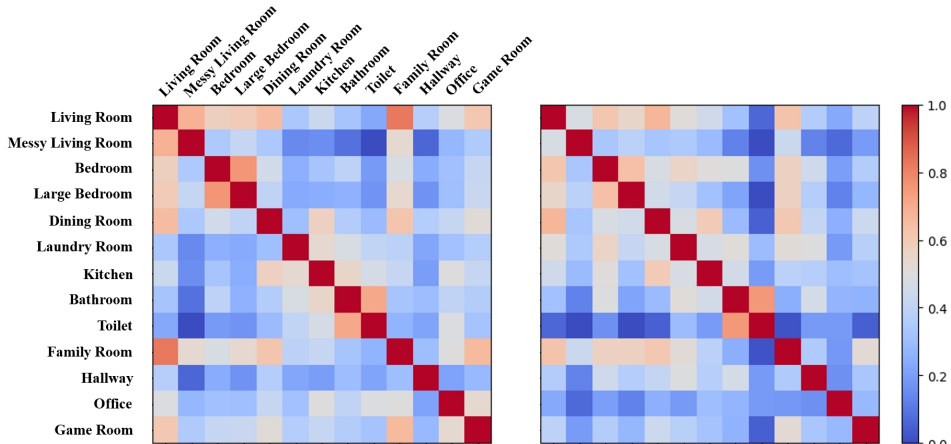
A.3 ABLATION STUDY

To explicitly encode the region information, we apply the LVM to process the background pixels out of the object bounding box and LLM to encode the region label text. What's more, for object pixels, object label text is combined with the region text in the form of 'object in the region' before being encoded by LLM. To ablate the contribution of vision-language embeddings from CLIP and semantic embeddings from Sentence-BERT in encoding region features, we compare different weight settings between the v-s embeddings when inferring the regions with 3D position inputs. Results are shown in Fig. A3. It can be seen that both vision-language embeddings and semantic embeddings are indispensable, and weight settings with the greatest results are used for Topo-Field.

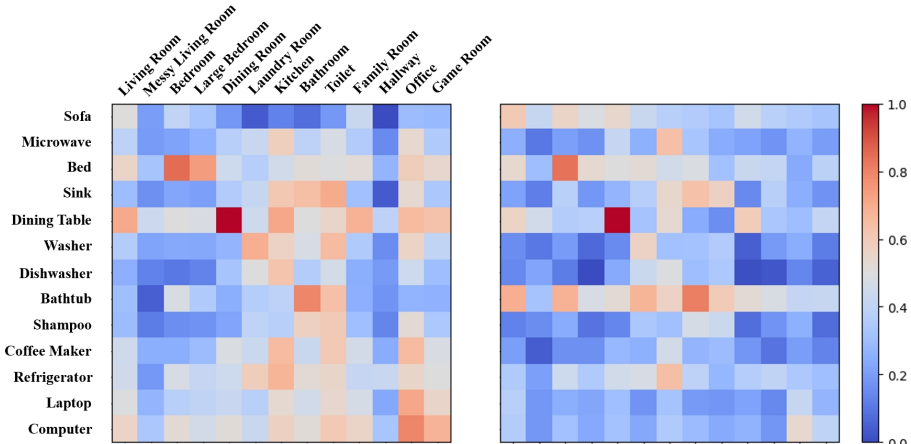
A.4 HIERARCHICAL APPROACH COMPARISON

Hierarchical scene representation is widely studied with numerous tasks, mainly employing scalable receptive fields and representations to fine-tune results of scalable objects and local relations. As Fig.A.8 shows, VoxFusion introduced octree map with various voxel sizes, LERF employed feature pyramids. As far as we know, few of them explicitly consider the layout level information and the association with objects and positions. This idea comes from recent neuroscience findings, and similar theory has not yet been introduced in scene representations.

054
 055
 056
 057
 058
 059
 060
 061
 062
 063
 064
 065
 066
 067
 068
 069
 070
 071
 072
 073
 074
 075
 076
 077
 078
 079
 080
 081
 082
 083
 084
 085
 086
 087
 088
 089
 090
 091
 092
 093
 094
 095
 096
 097
 098
 099
 100
 101
 102
 103
 104
 105
 106
 107
 108



(a)



(b)

Figure A2: The similarity of a set of region embeddings (as shown in a) and object-region embeddings (as shown in b). The left graph shows the vision-language embedding similarity and the right one shows the semantic embedding similarity.

A.5 TOPOMETRIC SEARCH FOR PLANNING

We employ a simple A* approach for planning. Given a topometric graph G , the start point p , and the target destination object text t . First, the belonged region r of p is inferred according to the main paper. The existing objects nodes embeddings are compared with the encoded visual-language and semantic embeddings of t to find the target object node o . At the same time, if the region of destination object r_d is declared, the search process would be more simple by directly search among region nodes. Here lists the pseudocode of the employed A*.

A.6 TOPOMETRIC MAP NODES EXAMPLES

We list the attributes of nodes and edges in the topometric map as example here in Listing 1 – 4, including the object nodes, region nodes, and edges.

```

1 {
2   "id": 0,
3   "node_type": region,
4   "bbox_extent": [
5     4.1633099999999999,
6     4.207343,

```

108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161

Algorithm 1 AStar(G, r, o)

```
1:  $openSet \leftarrow \{r\}$  ▷ Set of nodes to be evaluated
2:  $cameFrom \leftarrow \{\}$  ▷ Mapping of nodes to their parent nodes
3:  $gScore[r] \leftarrow 0$  ▷ Cost from start along best known path
4:  $fScore[r] \leftarrow h(r, o)$  ▷ Estimated total cost from start to goal
5: while  $openSet$  is not empty do
6:    $current \leftarrow$  node in  $openSet$  with lowest  $fScore$  value
7:   if  $current = o$  then
8:     return RECONSTRUCTPATH( $cameFrom, o$ )
9:   end if
10:  remove  $current$  from  $openSet$ 
11:  for each neighbor  $n$  of  $current$  do
12:     $tentativeGScore \leftarrow gScore[current] + d(current, n)$ 
13:    if  $tentativeGScore < gScore[n]$  then
14:       $cameFrom[n] \leftarrow current$ 
15:       $gScore[n] \leftarrow tentativeGScore$ 
16:       $fScore[n] \leftarrow gScore[n] + h(n, o)$ 
17:      if  $n$  not in  $openSet$  then
18:        add  $n$  to  $openSet$ 
19:      end if
20:    end if
21:  end for
22: end while
23: return "No path found"
24: function RECONSTRUCTPATH( $cameFrom, current$ )
25:    $path \leftarrow [current]$ 
26:   while  $current$  is in  $cameFrom$  do
27:      $current \leftarrow cameFrom[current]$ 
28:     insert  $current$  at the beginning of  $path$ 
29:   end while
30:   return  $path$ 
31: end function
```

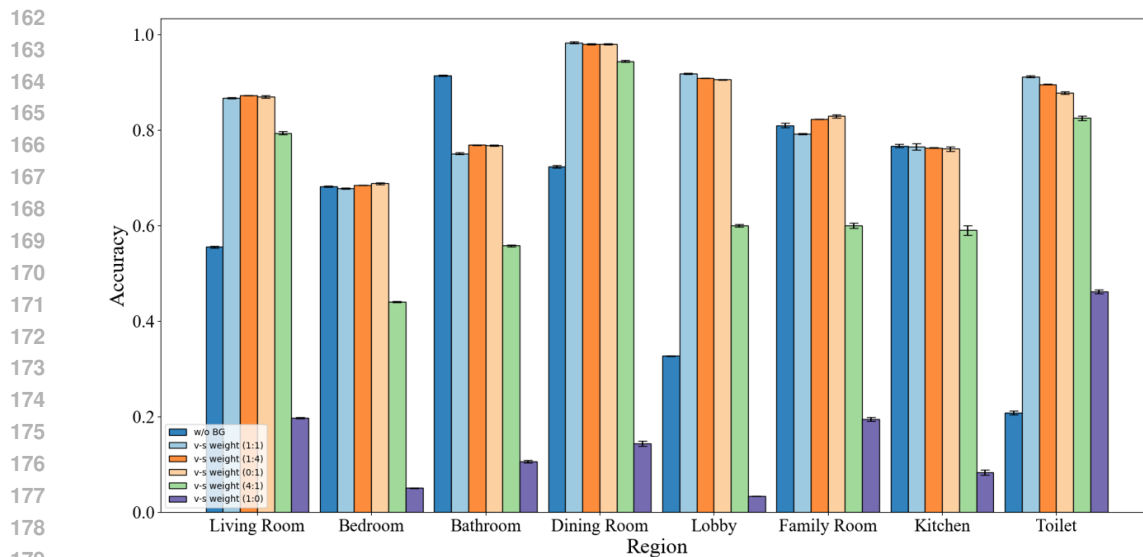


Figure A3: Ablation results on the accuracy of region prediction on Matterport3D? with 3D positions input. The w/o BG stands for not encoding background pixels to get region embeddings, and v-s weight ablates the weight of vision-language and semantic embeddings in the embeddings similarity contribution. Error bars show the results among samples from different scenes in Matterport3D?.

```

188 7         2.53566175
189 8     ],
190 9     "bbox_center": [
191 10         -8.821845,
192 11         2.6915385,
193 12         1.259409125
194 13     ],
195 14     "class": "bedroom",
196 15     "caption": "A bedroom at the northwest of the house with warm
197 16 },

```

Listing 1: Region node

```

200 1 {
201 2     "id": 1,
202 3     "node_type": object,
203 4     "bbox_extent": [
204 5         0.3569,
205 6         0.2297,
206 7         0.101.8
207 8     ],
208 9     "bbox_center": [
209 10         0.3222,
210 11         -1.1108,
211 12         -0.5062
212 13     ],
213 14     "class": "picture",
214 15     "caption": "A white framed picture hanging on the wall."
215 16 },

```

Listing 2: Object node

```

215 1 {

```

```

216 2     "id": 0,
217 3     "node_type": Entrance,
218 4     "bbox_extent": [
219 5         0.5,
220 6         1.6,
221 7         2.8,
222 8     ],
223 9     "bbox_center": [
224 10        -3.244,
225 11        -0.276,
226 12         0.487
227 13     ],
228 14     "class": "Entrance",
229 15     "caption": "Entrance connecting bedroom and living room."
230 16 },

```

Listing 3: Entrance node

```

231 1 {
232 2     "id": 2,
233 3     "edge_type": region_entrance,
234 4     "start_node": {
235 5         "id": 0,
236 6         "node_type": region,
237 7         "bbox_extent": [
238 8             4.163309999999999,
239 9             4.207343,
240 10            2.53566175
241 11         ],
242 12         "bbox_center": [
243 13             -8.821845,
244 14             2.6915385,
245 15             1.259409125
246 16         ],
247 17         "region_tag": "bedroom"
248 18     },
249 19     "end_node": {
250 20         "id": 0,
251 21         "node_type": Entrance,
252 22         "bbox_extent": [
253 23             0.5,
254 24             1.6,
255 25             2.8,
256 26         ],
257 27         "bbox_center": [
258 28             -3.244,
259 29             -0.276,
260 30             0.487
261 31         ],
262 32         "class": "Entrance",
263 33         "caption": "Entrance connecting bedroom and living room."
264 34     },
265 35     "relationship": connected,
266 36     "position_relation": "b to the southeast of a",
267 37     "position_reason": "The x-coordinate of the center of bbox of
268 38     end_node (-3.244) is larger than that of start_node (-8.821845), and
269 39     the y-coordinates of the center of bbox of end_node (-0.276) is less
    than that of start_node (4.207343). Therefore, b is to the southeast
    of a."
    "caption": "The pathway from bedroom to living room."
    },

```

Listing 4: Region entrance edge

```

269 1 {

```

```

270 2     "id": 2,
271 3     "node_type": object_region,
272 4     "start_node": {
273 5         "id": 7,
274 6         "node_type": object,
275 7         "bbox_extent": [
276 8             2.155,
277 9             2.052,
278 10            0.883
279 11        ],
280 12        "bbox_center": [
281 13            5.598,
282 14            2.566,
283 15            0.136
284 16        ],
285 17        "class": "bed",
286 18        "caption": "a bed with a white comforter and a pillow"
287 19    },
288 20    "end_node": {
289 21        "id": 0,
290 22        "node_type": region,
291 23        "bbox_extent": [
292 24            4.163309999999999,
293 25            4.207343,
294 26            2.53566175
295 27        ],
296 28        "bbox_center": [
297 29            -8.821845,
298 30            2.6915385,
299 31            1.259409125
300 32        ],
301 33        "class": "bedroom"
302 34        "caption": "A bedroom at the northwest of the house with warm
303 35        lighting. Main objects include a bed in the center, a large closet,
304 36        and a dresser at the corner."
305 37    },
306 38    "relationship": belong,
307 39    "position_relation": "a in the center of b",
308 40    "caption": "According to the bbox center position and extent, the bed
309 41    is in the center of bedroom."
310 42 },

```

Listing 5: Object region edge

A.7 PROMPT EXAMPLE FOR REGION NODE CONNECTIVITY DESCRIPTION

With topometric mapped nodes, we leverage LLM to describe the connectivity of nodes according to the general knowledge and bounding box 3D position. In listing 5, here we provide a prompt example to describe the connectivity relationship between content objects and regions and set up the edge.

```

314 1 {
315 2 DEFAULT_PROMPT_POST = ""
316 3 You are an excellent graph managing agent. Given a graph nodes set of an
317 4 environment,
318 5 you can explore the relationships of nodes with their attributes and
319 6 build edges among
320 7 them.
321 8 The input is a list of JSONS describing two types of nodes, including the
322 9 object and
323 10 region. You need to produce a JSON string (and nothing else) and set up
324 11 edges between them with keys: "relationship", "position_relation" and
325 12 "caption".

```

```

324 9
325 10 Each of the JSON fields will have the following fields:
326 11 1. id: a unique number
327 12 2. node_type: type of this node
328 13 3. bbox_extent: the 3D bounding box extents
329 14 4. bbox_center: the 3D bounding box center
330 15 5. class: an extremely brief description
331 16 6. caption: a sentence describing node attributes in detail
332 17
333 18 Produce a "relationship" field that best describes the relationship of
334 19 the object node and region node. Set "false" if the object is not
335 20 related to the area or is not reasonable, the relationship is refused
336 21 . Produce a
337 22 "position_relation" field describing the position relationship between
338 23 object and region according to their
339 24 bounding box information in the 3D space. Before producing the "
340 25 position_relation" field, produce a "caption" field that explains why
341 26 the "position_relation" field is reasonable.
342 27
343 28 The built edges should include following fields:
344 29 1. id: a unique number of each edge in order
345 30 2. node_type: according to the connected node type in the form "
346 31 start_node\_end_node"
347 32 3. start_node: keep JSON values of the object node unchanged
348 33 4. end_node: keep JSON values of the region node unchanged
349 34 5. relationship
350 35 6. position_relation
351 36 7. caption
352 37 ""
353 38

```

Listing 6: Prompt example to set up edge with nodes.

A.8 ADDITIONAL EXPERIMENT RESULTS

Additional experiments results of object localization using text query inputs and view localization using image query inputs. Also, a table is provided showing the metric on exactly each region class from 4 scenes in Matterport3D dataset.

Regions	Scene1			Scene2			Scene3			Scene4		
	Acc.	Pre.	F1	Acc.	Pre.	F1	Acc.	Pre.	F1	Acc.	Pre.	F1
Living Room	0.948	0.970	0.959	0.870	0.881	0.875	0.778	0.810	0.793	0.902	0.949	0.925
Bedroom	0.943	0.825	0.880	0.925	0.923	0.924	0.687	0.767	0.725	0.920	0.870	0.894
Bathroom	0.466	0.680	0.554	0.903	0.898	0.901	0.875	0.463	0.605	0.797	0.831	0.814
Dining Room	-	-	-	0.961	0.794	0.870	0.774	0.732	0.752	0.933	0.887	0.910
Lobby	0.681	0.941	0.790	0.853	0.951	0.899	0.978	0.510	0.671	0.855	0.698	0.769
Family Room	-	-	-	-	-	-	0.903	0.571	0.700	0.926	0.936	0.931
Kitchen	0.994	0.654	0.789	0.788	0.836	0.811	0.833	0.833	0.833	0.758	0.854	0.803
Office	-	-	-	0.969	0.848	0.905	-	-	-	0.953	0.883	0.917
Toilet	-	-	-	-	-	-	0.900	0.711	0.795	-	-	-
Avg. Acc./Samples	0.886 / 169k			0.900 / 185k			0.884 / 111k			0.894 / 112k		

Table 1: Region prediction results on the test set of different scenes from the Matterport3D dataset. Accuracy, precision, and F1 score are used as metrics.

378
 379
 380
 381
 382
 383
 384
 385
 386
 387
 388
 389
 390
 391
 392
 393
 394
 395
 396
 397
 398
 399
 400
 401
 402
 403
 404
 405
 406
 407
 408
 409
 410
 411
 412
 413
 414
 415
 416
 417
 418
 419
 420
 421
 422
 423
 424
 425
 426
 427
 428
 429
 430
 431

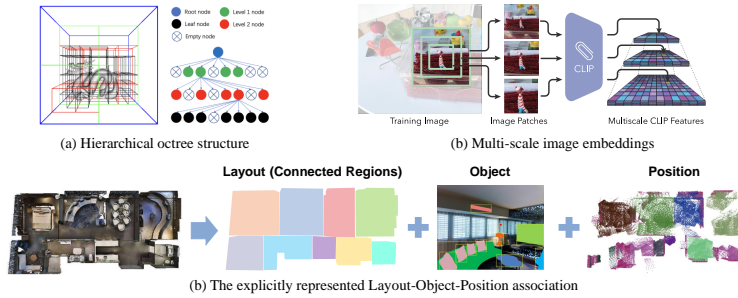


Figure A4: The comparison of the hierarchical scene representation strategy against previous works.



Figure A5: Text query localization on scene 2t7WUuJeko7?.

432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485



Figure A6: Text query localization on scene 17DRP5sb8fy?.



Figure A7: Text query localization on scene Apartment?.

486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539

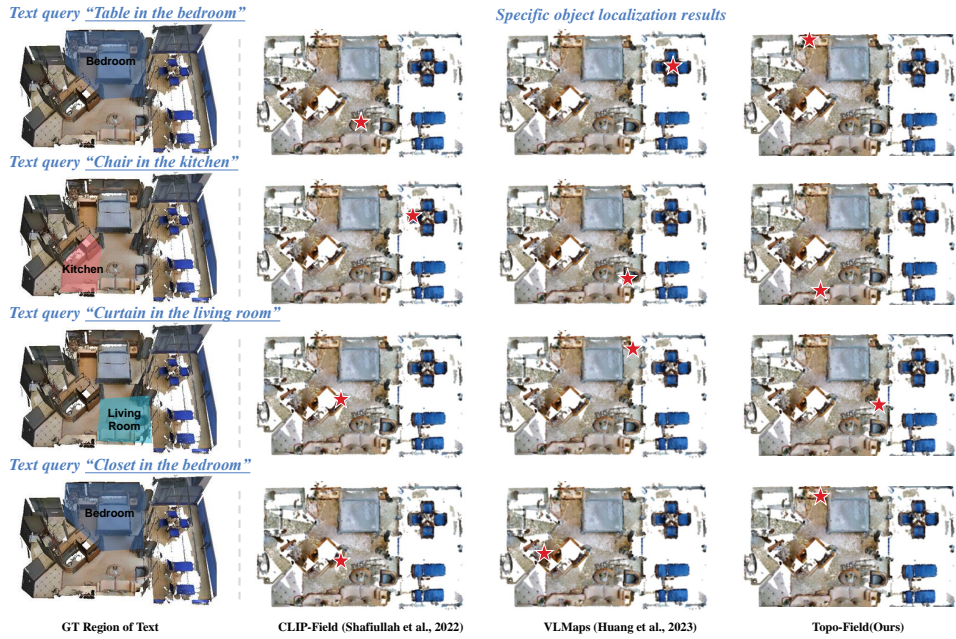


Figure A8: Text query localization on scene HxpKQynjin?.

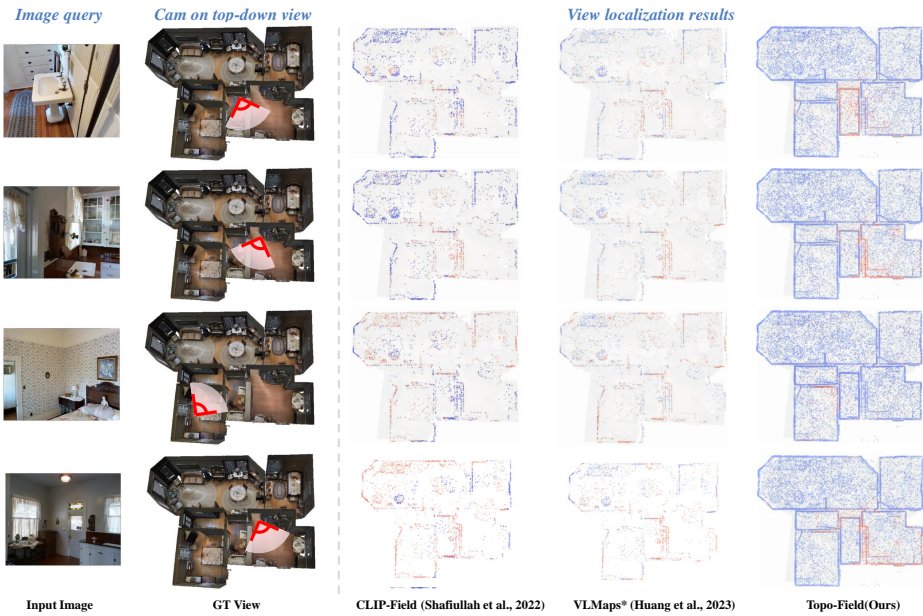


Figure A9: Image query localization on scene 2t7WUuJeko7?.

540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593

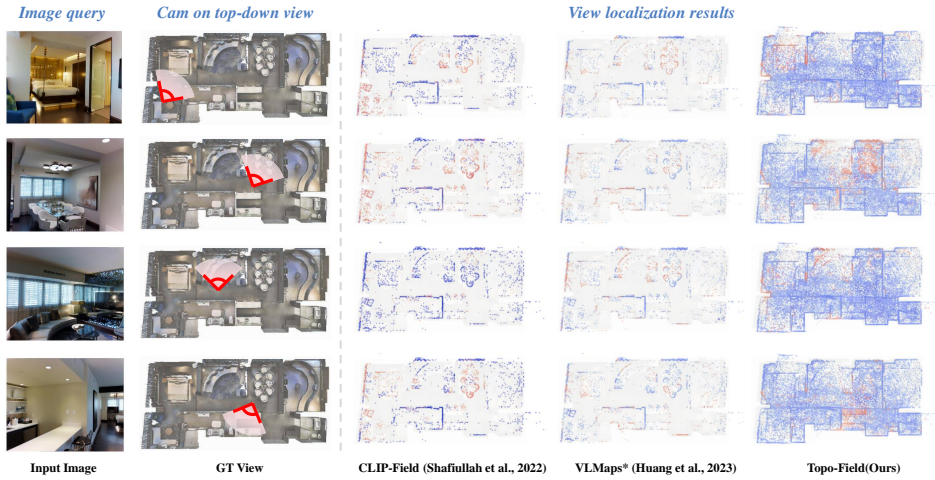


Figure A10: Image query localization on scene 17DRP5sb8fy?.

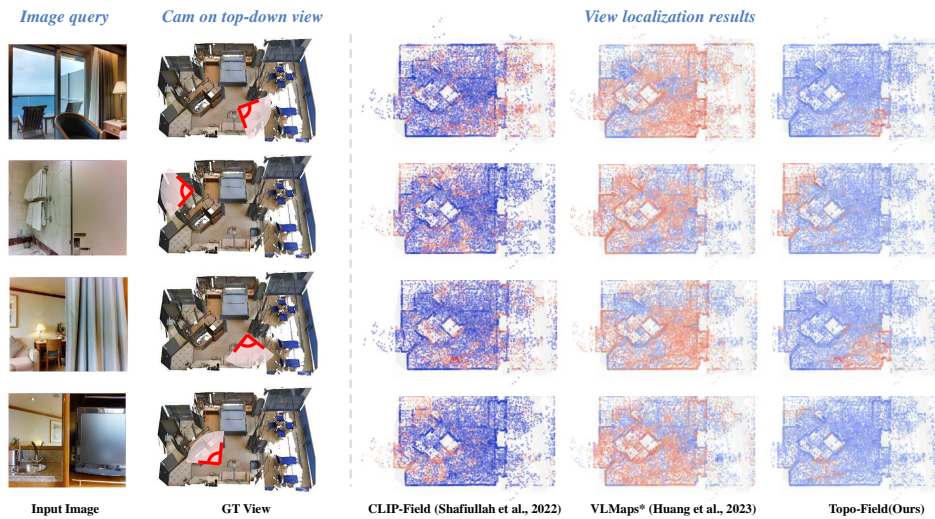


Figure A11: Image query localization on scene HxpKQynjfin?.