Supplementary Materials for ReALFRED: An Embodied Instruction Following Benchmark in Photo-Realistic Environments

Taewoong Kim[1,*], Cheolhong Min[1,*], Byeonghwi Kim[1], Jinyeon Kim[1,2], Wonje Jeung[1], and Jonghyun Choi[1,†]

[1] Seoul National University
[2] Yonsei University
{twoongg.kim, cheolhong.min, byeonghwikim}@snu.ac.kr, jinyeonkim@yonsei.ac.kr, {specific0924, jonghyunchoi}@snu.ac.kr

Note: Green denotes reference to the main paper.

A Benchmark Details

We provide all 114 types of objects in Fig. 1. The **bold** text denotes the uniquely introduced object in ours.

A.1 Annotation Interface

In this section, we describe the overall process of acquiring language annotations. Fig. 2a illustrates the interface of the Mechanical Turk used to collect human annotations from Mechanical Turk workers. We provided workers with an expert demonstration video and divided the timeline segments that have the intended subgoal (*e.g.*, *'pick up the pencil case,' 'go to the sofa'*). Workers were asked to fill each segment with their own words (*e.g.*, *'Pick up the pencil case from the coffee table,' 'Walk around the table to get closer to the sofa'*). Workers were paid \$0.7 per annotation as following the previous work [61]. Moreover, we adopted a voting survey to filter out inappropriate annotations. Fig. 2b illustrates the interface of getting votes from workers. We conducted the voting with a minimum of 2 and up to 5 reviewers per annotation. Only annotations that received more than a majority of accepts in all cases were included in our set of annotations. For annotations that did not achieve a majority of accepts, we re-collected annotations and implemented a voting system to prevent the inclusion of low-quality annotations. We paid workers \$0.35 to compare 5 sets of annotations following [61].

A.2 Vocabulary Distribution

We provide vocabulary statistics for the language instructions in the ReALFRED benchmark in Fig. 4.

*Equal contribution. †Corresponding author.

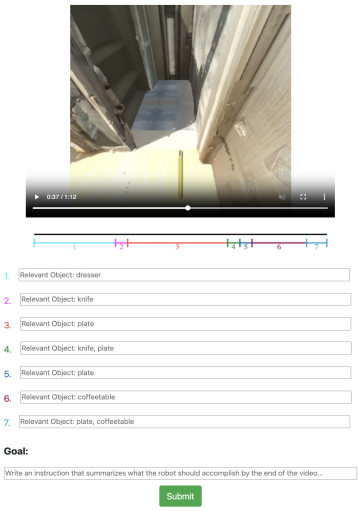
All Object Types in **ReALFRED**

Bottle, **Calculator**, **CannedFood**, **Comb**, **ComputerMouse**, **Controller**, **Cookie**, **DressingTable**, **Dumbbell**, **Eggplant**, **EggplantSliced**, **Flute**, **Grape**, **Hammer**, **Headband**, **Keyboard**, **Lemon**, **LemonSliced**, **Mango**, **Onion**, **OnionSliced**, **Peach**, **PencilCase**, **Plum**, **RemoteControl**, **Scissors**, **Sink**, **Strawberry**, **StrawberrySliced**, **TVStand**, **Table**, **Tambourine**, **Towel**, **TowelHolder**, **Watermelon**

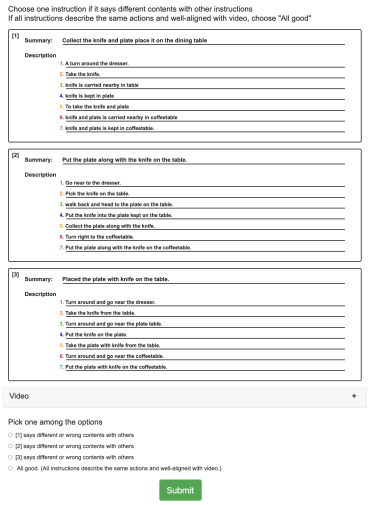
All Object Types in ALFRED

AlarmClock, Apple, AppleSliced, ArmChair, BaseballBat, BasketBall, BathtubBasin, Bed, Book, Bowl, Box, Bread, BreadSliced, ButterKnife, CD, Cabinet, Candle, Cart, CellPhone, Cloth, CoffeeMachine, CoffeeTable, CounterTop, CreditCard, Cup, Desk, DeskLamp, DiningTable, DishSponge, Drawer, Dresser, Egg, FloorLamp, Fork, Fridge, GarbageCan, Glassbottle, HandTowel, Kettle, KeyChain, Knife, Ladle, Laptop, Lettuce, LettuceSliced, Microwave, Mug, Newspaper, Ottoman, Pan, Pen, Pencil, PepperShaker, Pillow, Plate, Plunger, Pot, Potato, PotatoSliced, RemoteControl, Safe, SaltShaker, Shelf, SideTable, SinkBasin, SoapBar, SoapBottle, Sofa, Spatula, Spoon, SprayBottle, Statue, StoveBurner, TennisRacket, TissueBox, Toilet, ToiletPaper, ToiletPaperHanger, Tomato, TomatoSliced, Vase, Watch, WateringCan, WineBottle

Fig. 1: Object list in ALFRED and ReALFRED. Listed in alphabetical order. The object classes in the ReALFRED benchmark are a superset of those in ALFRED, with newly introduced objects highlighted in bold.



(a) Annotation interface.



(b) Voting survey interface.

Fig. 2: Mechanical Turk interface.

A.3 Scanned Indoor Houses

Among collected 150 scenes, we split scenes into 135 seen and 15 unseen environments. Then we further split into validation (both seen and unseen) and test (both seen and unseen) folds. The details are presented in Table 1. Note that validation and test unseen scenes are exclusive.

Degree of photorealism. We compare a degree of photorealism by measuring FID [27] and KID [4] scores following Ramakrishnan *et al.* [56]. We use rendederd

Table 1: Indoor house splits.

	Train	Validation		Test	
		Seen	Unseen	Seen	Unseen
# of scenes	135	131	6	135	9

Table 2: Photorealism comparison.

Environments	HM3D		Gibson		Gibson HQ	
	FID↓	KID×10^3 ↓	FID↓	KID×10^3 ↓	FID↓	KID×10^3 ↓
TDW [20]	129.52	89.13±1.52	122.68	80.38±1.36	124.89	79.05±1.49
BEHAVIOR-1K [41]	113.59	81.58±2.96	99.52	66.03±1.85	107.53	63.03±1.99
REALFRED	**81.06**	**69.96±1.47**	**83.25**	**71.05±1.51**	**101.60**	**89.98±1.72**
Gibson [73]	43.79	31.66±1.05	—	—	46.67	33.74±1.21
HM3D [56]	—	—	43.79	31.66±1.05	26.33	21.70±1.03

RGB images from each synthetic environments [20, 41]. We compare the image quality with a set of RGB images rendered from previous dataset, HM3D [56] and Gibson [73]. To compare with previous scanned environments, we acquire a collection of real RGB images derived from high-resolution raw panoramas in Gibson. We designate this collection as Gibson HQ. Results are presented in Table 2. We observe that ours achieves the lowest (*i.e.*, best) FID and KID scores when compared to the previous environments [20, 41] that provide interaction with the environments including the objects. However, despite these promising metrics, our results reflect lower photorealism compared to the previous scanned environments [73, 56] which do not provide interactable environments. This may be due to manually added agent-interactable objects.

Manual removal of 3D objects from background. Designers use automated tools (*i.e.*, Blender) to extract objects from the background meshes easily and set their properties (*e.g.*, labels, colliders, interactability, *etc.*).

In Fig. 3, we provide the qualitative examples of how the source data were fixed by data correction. Workers fill in the missing parts or smooth out the uneven surface by checking the overall alignment composition after mapping and alignment using automated tools such as Blender.

Qualitative examples. We show several houses used in the REALFRED benchmarks in Fig. 8 and 9.

A.4 Examples of Expert Demonstration

Fig. 10-13 illustrate the examples of expert demonstrations for 7 task types. The agent has to solve the task in interactive environments by understanding the language instructions and planning the sequential and executable actions.

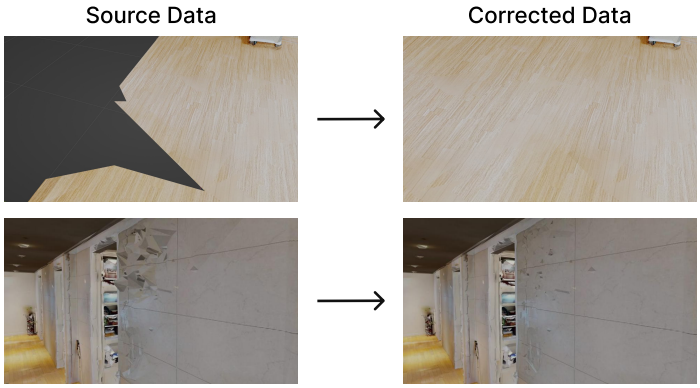


Fig. 3: Qualitative examples of the 3D mesh data correction. In each row, we provide source data and the corrected data.

A.5 Diverse Episodes with More Objects

Each episode is generated based on the combination of task-relevant objects (*e.g.*, *put a 'knife' on the 'table.'*) and this indicates that more object classes can result in more object-diverse episodes (*e.g.*, *put a 'potato' in a 'fridge.'*). We observe that our REALFRED provides more diverse episodes compared to the ALFRED benchmark [61] by enriching the number of object classes. Here, we denote an episode whose combination of task-relevant objects does not overlap with the others by a unique episode. We observe that our REALFRED benchmark provides the $4,649$ unique episodes while ALFRED provides $2,522$ ones in the combined *train* and *valid* splits. In addition, the ratio of the unique episodes among the total ones is 53.3% in REALFRED while it is 35.6% in ALFRED. This indicates that our REALFRED benchmark provides not only a larger number of episodes but also more diverse combinations of episodes.

A.6 Qualitative Comparison of Indoor Houses

We provide a qualitative comparison of the indoor houses used in our REALFRED and ALFRED [61] in the attached video files (*video1.mp4*, *video2.mp4*, and *video3.mp4*). We provide the agent's egocentric view on the left-hand side of a video and a top-down view with a red circle denoting the agent's corresponding current location. While ALFRED's indoor house environments consist of single room types on a room scale, ours are provided on a house scale, featuring multiple rooms within a single house. This implies that agents developed with our environments are enabled to perform instruction-following tasks that require navigating through multiple rooms.

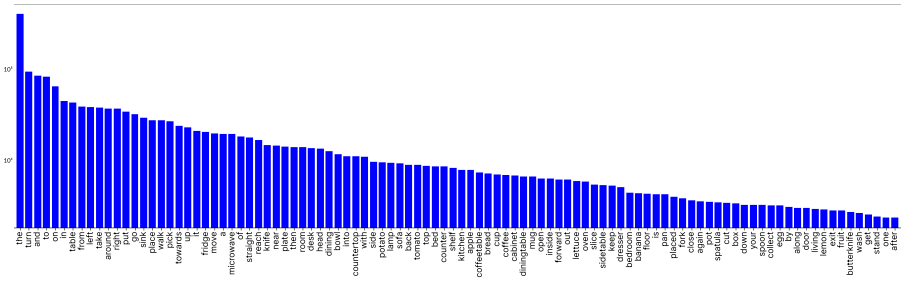


Fig. 4: Vocabulary statistics in collected human language instructions.

B Details of State of the Art Models

We provide details of state-of-the-art models in imitation learning and spatial map reconstruction, respectively.

B.1 Imitation Learning

The Seq2Seq [61] model encodes the visual input with the frozen backbone visual encoder. The natural language goal and instructions are encoded with a bidirectional LSTM encoder to produce an embedding for each word. Alongside the previous action, embeddings are passed as input to an LSTM cell to produce the current hidden state. The action and corresponding mask are finally predicted using a hidden state. MOCA [62] exploits separate branches for action prediction and object localization to better address different semantic understanding. ABP [33] extends MOCA [62] by perceiving surrounding perception for a better understanding of environments with the enlarged field of view.

B.2 Spatial Map Reconstruction

HLSM [5] uses a hierarchical controller to bridge the gap between natural language instructions and agent executable actions. The high-level controller predicts the next subgoal given the instruction and the map, and then the low-level controller outputs a sequence of actions to achieve the subgoal. FILM [46] utilizes a pre-designed template as a high-level action sequence. It uses two submodules of BERT classifiers to predict the type of instruction and the arguments to fill in the template. Finally, it uses a deterministic algorithm [60] for obstacle-free path planning. LLM-Planner [63] leverages large language model to generate subgoal sequence with a few examples. To enhance LLMs planning accuracy, it updates plans that are physically grounded in the environment. CAPEAM [34] uses context-aware planning to plan a subgoal sequence and conduct the respective subgoal with the corresponding detailed planners. It also uses additional memory to prevent the interaction of inappropriate objects.

Table 3: Task and Goal-Condition Success Rate (valid split). Path-length-weighted (PLW) metrics are given in parentheses for each value. We report mean and stndadrd deviation over multiple runs. †Authors' implementation as the code is not publicly available.

Learning	Model	Validation			
		Seen		*Unseen*	
		Success Rate	Goal Condition	Success Rate	Goal Condition
Imitation Learning	Seq2Seq [61]	0.77 ± 0.06 (0.47 ± 0.06)	6.93 ± 0.06 (4.73 ± 0.06)	0.00 ± 0.00 (0.00 ± 0.00)	4.03 ± 0.06 (2.50 ± 0.00)
	MOCA [62]	12.64 ± 0.12 (8.35 ± 0.16)	20.95 ± 0.18 (13.43 ± 0.16)	1.44 ± 0.05 (0.56 ± 0.06)	6.76 ± 0.04 (3.64 ± 0.06)
	ABP† [33]	24.71 ± 0.05 (15.49 ± 0.34)	33.80 ± 0.14 (23.27 ± 0.32)	4.22 ± 0.05 (1.70 ± 0.08)	11.71 ± 0.27 (5.42 ± 0.13)
Spatial Map Reconst.	HLSM [5]	4.23 ± 0.08 (0.72 ± 0.08)	9.14 ± 0.09 (2.67 ± 0.06)	1.08 ± 0.14 (0.19 ± 0.03)	6.12 ± 0.23 (1.52 ± 0.02)
	FILM [46]	7.08 ± 0.28 (1.87 ± 0.11)	11.93 ± 0.23 (4.82 ± 0.15)	4.44 ± 0.17 (1.25 ± 0.10)	9.26 ± 0.13 (3.84 ± 0.11)
	LLM-Planner† [63]	5.80 ± 0.19 (1.51 ± 0.03)	11.69 ± 0.35 (4.76 ± 0.07)	3.33 ± 0.22 (0.96 ± 0.05)	8.29 ± 0.19 (3.49 ± 0.09)
	CAPEAM† [34]	13.45 ± 0.05 (3.43 ± 0.06)	18.16 ± 0.27 (4.50 ± 0.05)	4.92 ± 0.22 (1.22 ± 0.03)	9.47 ± 0.23 (1.79 ± 0.04)

Table 4: Task and Goal-Condition Success Rate (test split). Path-length-weighted (PLW) metrics are given in parentheses for each value. We report mean and stndadrd deviation over multiple runs. †Authors' implementation as the code is not publicly available.

Learning	Model	Test			
		Seen		*Unseen*	
		Success Rate	Goal Condition	Success Rate	Goal Condition
Imitation Learning	Seq2Seq [61]	1.10 ± 0.00 (0.05 ± 0.01)	6.60 ± 0.00 (5.00 ± 0.00)	0.00 ± 0.00 (0.00 ± 0.00)	3.50 ± 0.00 (2.80 ± 0.00)
	MOCA [62]	14.11 ± 0.03 (9.20 ± 0.05)	22.84 ± 0.04 (16.42 ± 0.04)	0.62 ± 0.08 (0.35 ± 0.05)	5.14 ± 0.08 (3.39 ± 0.06)
	ABP† [33]	27.44 ± 0.40 (16.96 ± 0.16)	35.81 ± 0.23 (24.57 ± 0.19)	3.54 ± 0.23 (1.51 ± 0.08)	10.57 ± 0.22 (5.59 ± 0.10)
Spatial Map Reconst.	HLSM [5]	6.27 ± 0.04 (0.88 ± 0.10)	10.44 ± 0.13 (2.78 ± 0.10)	0.49 ± 0.16 (0.08 ± 0.03)	4.28 ± 0.13 (1.37 ± 0.16)
	FILM [46]	8.79 ± 0.07 (2.36 ± 0.01)	13.03 ± 0.08 (5.58 ± 0.08)	2.15 ± 0.18 (0.56 ± 0.04)	6.56 ± 0.15 (3.16 ± 0.05)
	LLM-Planner† [63]	8.16 ± 0.20 (2.20 ± 0.06)	13.20 ± 0.13 (5.72 ± 0.06)	1.90 ± 0.13 (0.57 ± 0.04)	6.33 ± 0.02 (3.09 ± 0.04)
	CAPEAM† [34]	15.61 ± 0.15 (3.68 ± 0.09)	20.22 ± 0.11 (5.39 ± 0.09)	2.87 ± 0.13 (0.84 ± 0.02)	7.36 ± 0.07 (2.01 ± 0.03)

C Extended Quantitative Results

We present experiment results with path-length-weighted success rate and goal condition (*i.e.*, PLWSR and PLWGC) over multiple runs in Table 3 and 4.

D Map Reconstruction Strategy

We provide a more detailed analysis of the challenges in recognizing narrow passages (*e.g.*, doors, aisles, *etc.*). Our observation is as follows: failure to recognize narrow navigable pathways leads an agent to *stuck* within the initial room.

To quantify this challenge, we establish a criterion for *leaving* a room by taking 1 step (*i.e.* 0.25 meter) further from the entrance of the room where the agent was initiated. Fig. 5 represents the top-down view of one of *valid unseen* scenes labeled for each space from room 1 to room 6. We observe that only 5.3% of the agents, from the total episodes conducted on the scene, left one room to another for further exploration. We also noticed that above mentioned *leaving* occurred within a short span, not exceeding 77 steps. This implies that if an agent initially overlooks narrow spaces, they will be mistaken for walls when viewed at an oblique angle.

We provide qualitative examples in Fig. 6, illustrating failure cases in reconstructing narrow passages. We observe that an agent that adopted a spatial map

Fig. 5: Top-down view of a scene with labeling. We select one of the largest scene in *valid unseen* fold and annotate the space from room 1 to room 6 based on the door gap for further analysis. The area inside the outer black line is a navigable area.

fails to reconstruct its surroundings, fails to recognize narrow doorways, gets *stuck* in a single room, and eventually, fails to complete a task.

We further explore the likelihood of an agent, under a random navigation policy, to be positioned where it can directly observe narrow spaces or door gaps to construct a semantic map accurately. We assess the likelihood within 100 steps (*i.e.* a slight buffer extended to 77 steps). In detail, we consider the agent's position, viewing direction, and the horizontal angle of the head to measure the likelihood. As a result, the agent has a 6.9% likelihood of aligning with and facing the exit passage to recognize it, similar to the empirical result of the agent leaving the room where it was initiated (*i.e.*, 5.3%). This may imply that not being positioned correctly to see doors shortly after its deployment increases the chance of getting stuck in a room. This consideration may not have been necessary in the ALFRED benchmark [61], which consists of single rooms. However, since ReALFRED is composed of multiple rooms, 'Spatial Map Reconst.' baselines may achieve much lower performance compared to [61].

E Qualitative examples for domain adaptation

We provide qualitative examples of real-to-sim domain adaptation in Fig. 7. 'Source' column denotes a visual frame from the ReALFRED benchmark, 'Cy-

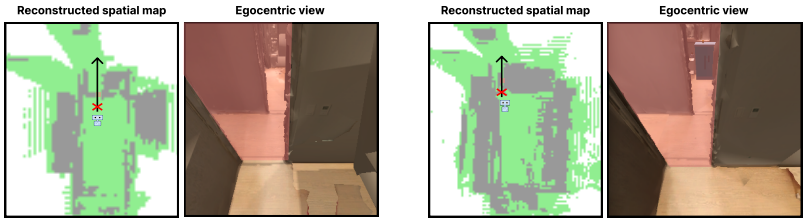


Fig. 6: Reconstructed spatial map and egocentric view. A reconstructed spatial map is presented on the left-hand side and an egocentric view of the agent is presented on the right-hand side on each figures. The green area denotes a predicted navigable area and the dark gray area denotes a predicted obstacle. Agent fails to recognize narrow navigable space (highlighted in red on the right-hand side of the each figure).

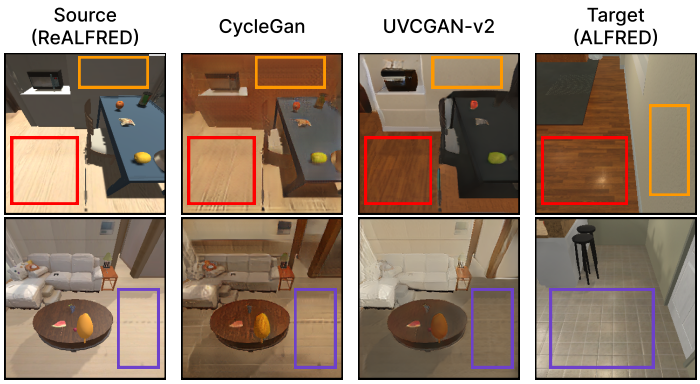


Fig. 7: Qualitative examples of the real-to-sim domain adaptation. In each column, we provide source image, domain adapted image for the source image, and an image from target domain.

cleGan' column denotes an adpated visual frame with CycleGan [79], 'UVCGAN-v2' column denotes an adapted visual frame with UVCGAN-v2 [70], and 'Target' column denotes an image from ALFRED target domain image. The advantage of real-to-sim domain adaptation is to make an agent feel at home, reducing the visual domain gap. We expect a domain adapted image to resemble some characteristics that are well represented in the target domain, where sim2real agent is trained.

We begin our examination with an example in the first row. We observe that the flooring, dominating the image frame, is adapted to resemble the flooring that frequently appears in the target domain (highlighted with ☐). We also notice that the image generated with CycleGan adapts the color of the wall to brown, while the image generated with UVCGAN-v2 adapts the wall to white tone (highlighted with ☐).

We now examine an example in the second row. We observe that the flooring, dominating the image frame as in the first example, is generated to resemble checkerboard tiles which are represented in target domain (highlighted with ☐).

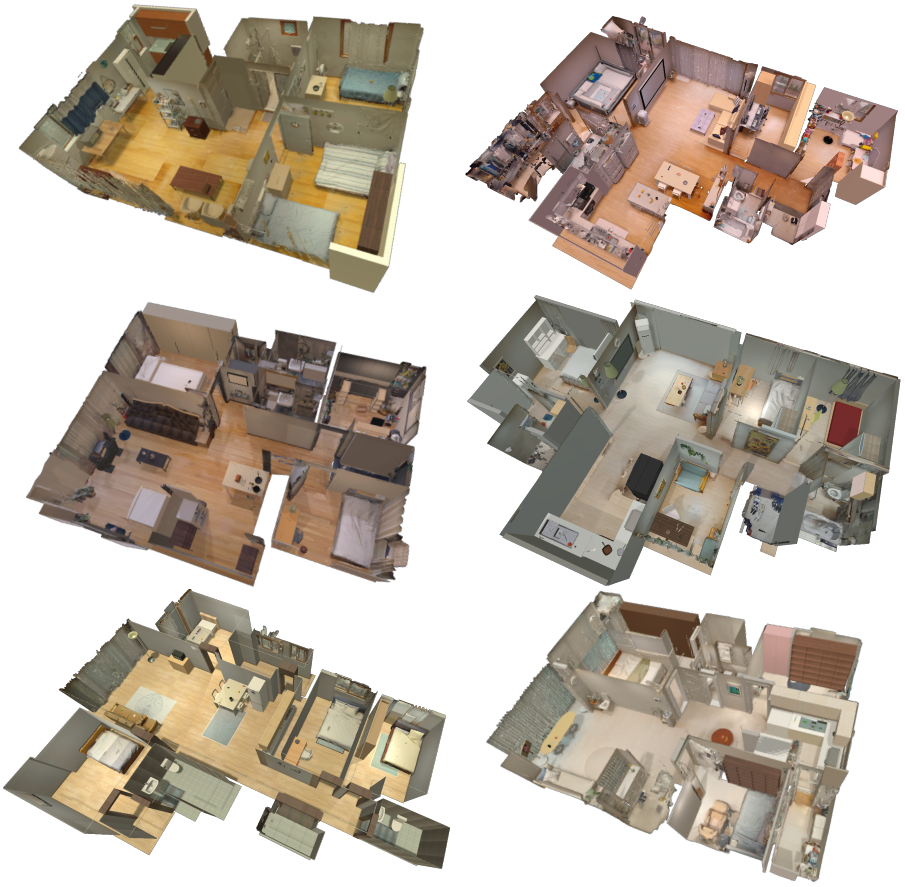


Fig. 8: Example of houses used in the ReALFRED benchmark.

Fig. 9: Additional example of houses used in the REALFRED benchmark.

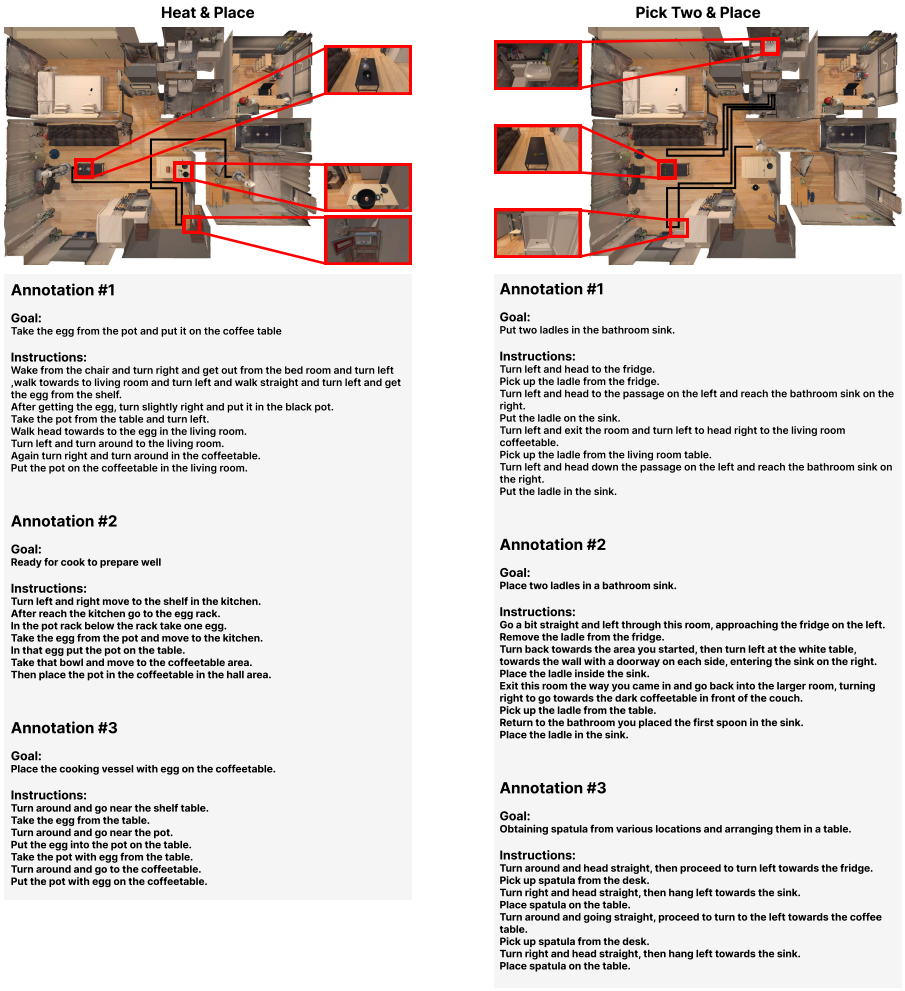


Fig. 10: Examples of expert demonstration and human annotation ('Heat & Place' on the left and 'Pick Two & Place' on the right). We provide examples of expert demonstration for tasks 'Heat & Place' and 'Pick Two & Place.' The black lines denote the expert's trajectories, and several egocentric views are presented alongside a top-down view of the scene.

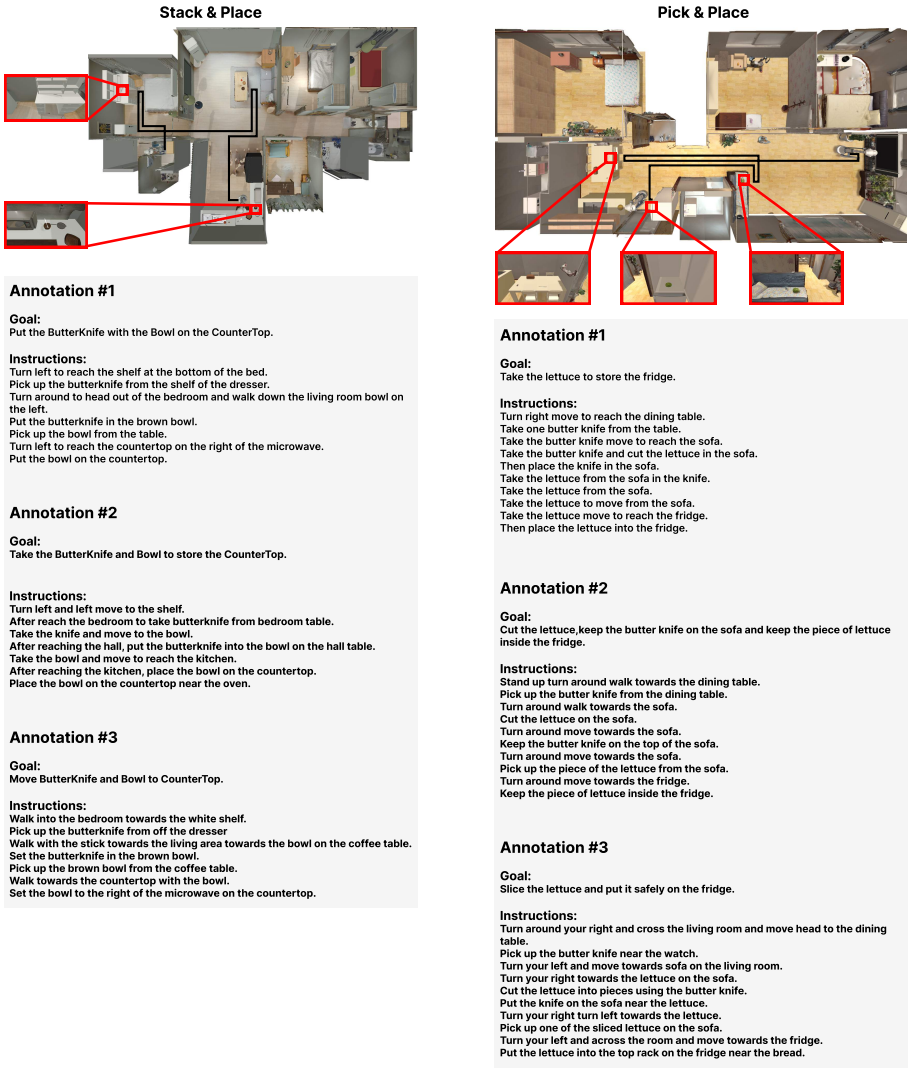
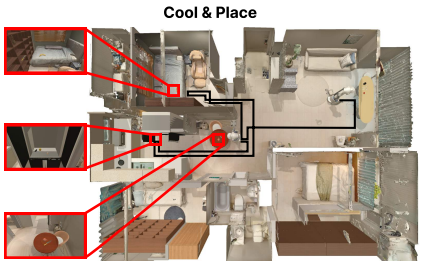


Fig. 11: Examples of expert demonstration and human annotation ('Stack & Place' on the left and 'Pick & Place' on the right). We provide examples of expert demonstration for tasks 'Stack & Place' and 'Pick & Place.' The black lines denote the expert's trajectories, and several egocentric views are presented alongside a top-down view of the scene.



Cool & Place

Annotation #1

Goal:
Place a slice of Banana on the CoffeeTable.

Instructions:
Turn right toward the chair and coffeetable in the corner, go across to the coffeetable and chair.
Pick up the knife on the table.
Turn right go to the first door on the left, go across to the banana.
Slice the banana on the bed.
Place the knife on the third shelf on the middle.
Pick up a piece of the banana from the bottom end.
Turn right at the doorway, go to the circle shaped table and go straight across to the fridge.
Place the banana in the fridge and pick it back up again.
Turn right and go across to the coffeetable.
Place the banana on the coffeetable.

Annotation #2

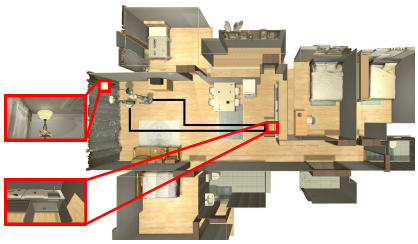
Goal:
Put the cooled slice of Banana on the CoffeeTable.

Instructions:
Turn around to head out of the room and reach the coffeetable.
Pick up the knife from the table.
Turn around and take a left turn to head to the banana.
Cut the banana on the bed with the knife.
Put the knife on the shelf.
Pick up the slice of banana from the bed.
Turn around to head to the kitchen on the right and reach the fridge.
Cool the banana in the fridge and take it out.
Head to the round coffeetable on the right.
Put the slice of banana on the coffeetable.

Annotation #3

Goal:
Put the half Banana on the CoffeeTable.

Instructions:
Go near to the coffeetable.
Take the knife and walk to the left.
Go to the banana room.
Knife to cut the banana from the bed.
Cut the banana in half and take it to the shelf to exit the room.
Go to reach the banana cupboard.
Put the banana into the fridge.
Take the banana from the cupboard.
Walk back and head to the coffeetable.
Put the half banana on the coffeetable.



Examine in Light

Annotation #1

Goal:
Examine a carry bag by the light of a tall lamp.

Instructions:
Turn right go straight to the kitchen.
Open to the cupboard under the sink.
Take the carry bag from the cupboard turn around go straight walk near to the lamp.
Turn on the lamp

Annotation #2

Goal:
Switch on the floor lamp to view the cookie.

Instructions:
Turn around in the living room and move towards the kitchen sink.
Pick the cookie from the cupboard under the sink.
Turn left and walk towards the floor lamp in the living room.
Switch on the floor lamp.

Annotation #3

Goal:
Take the cookie in the sink turn on the floor lamp.

Instructions:
Walk around the room.
Take the cookie in the sink.
With the cookie in hand to reach the floor lamp.
Turn on the floor lamp.

Fig. 12: Examples of expert demonstration and human annotation ('Cool & Place' on the left and 'Examine in Light' on the right). We provide examples of expert demonstration for tasks 'Cool & Place' and 'Examine in Light.' The black lines denote the expert's trajectories, and several egocentric views are presented alongside a top-down view of the scene.

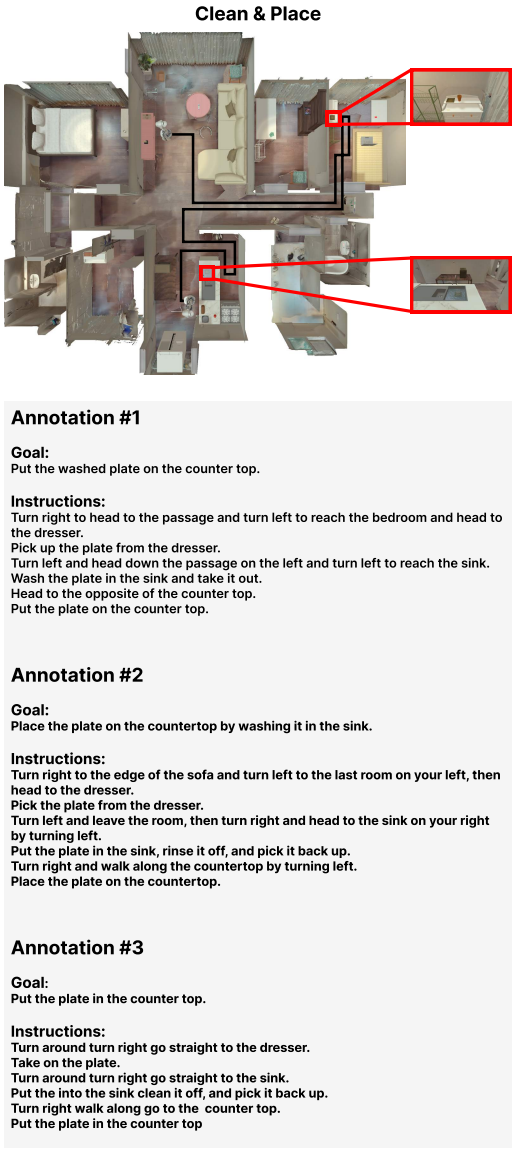


Fig. 13: Example of expert demonstration and human annotation ('Clean & Place'). We provide an example of an expert demonstration for task 'Clean & Place.' The black line denotes the expert's trajectory, and several egocentric views are presented alongside a top-down view of the scene.