Supplementary Materials for

Factorizing Perception and Policy for Interactive Instruction Following

Note: All orange characters indicate the index of the main paper.

1. ALFRED Benchmark Details

We provide the detailed description of the ALFRED benchmark here. Each expert ground-truth trajectory consists of a set of egocentric visual observation and ground-truth action pairs with corresponding natural language descriptions. We denote each trajectory by a tuple of $[\{(I_t, a_t)\}_{t=1}^{T}, S]$ where each I_t and a_t denotes the egocentric observation and the ground-truth action at the time step, t. T is the length of a trajectory and S is the natural language description. The natural language description, S, is composed of a goal statement, S_{goal}, and step-by-step instructions, S_{instr}. The goal statement describes the overall task the agent must complete. The step-by-step instructions provide detailed descriptions on how the agent can accomplish the task. For more information, please refer to [33].

Based on the egocentric observations and the language descriptions, the agent predicts an action and a mask for each time step. The action space is comprised of 5 navigation actions: MOVEAHEAD, ROTATERIGHT, ROTATELEFT, ROTATERIGHT, LOOKUP, and LOOKDOWN, and 7 interaction actions: PICKUP, PUT, OPEN, CLOSE, TOGGLEON, TOGGLEOFF, and SLICE along with the STOP action to terminate an episode. In case of interaction actions, the agent must localise objects of interest.

2. Implementation Details

The egocentric visual observations are resized to 224×224. For the visual encoder, we use a pre-trained ResNet-18 [18]. For the experimental results and analysis in subsequent sections, we use the goal statement as input for the IPM and step-by-step instructions for the APM, otherwise stated (Sec. 4.2).

The model is trained end-to-end using Adam for 30 epochs with an initial learning rate of 10^{-3} with a batch size of 16. We also use a dropout of 0.2 for visual features and LSTM decoder hidden states. We adopt data augmentation for the egocentric observations, $\{I_t\}_{t=1}^{T}$, to address the sample insufficiency of imitation learning in each trajectory.

Task-Type	Shridhar *et al.* [33]		MOCA	
	Seen	Unseen	Seen	Unseen
Pick & Place	7.0	0.0	**29.6**	**6.0**
Cool & Place	4.0	0.0	**32.5**	**2.8**
Stack & Place	0.9	0.0	**6.1**	**6.4**
Heat & Place	1.9	0.0	**31.8**	**5.1**
Clean & Place	1.8	0.0	**30.4**	**10.6**
Examine	9.6	0.0	**31.9**	**4.6**
Pick Two & Place	0.8	0.0	**19.4**	**1.2**
Average	3.7	0.0	**26.0**	**5.2**

Table 1: **Success rates across 7 task types in ALFRED.** All values are in percentage. The agent is evaluated on the Validation set. Highest values per fold are indicated in **blue**.

Subgoal	Shridhar *et al.* [33]		MOCA	
	Seen	Unseen	Seen	Unseen
Goto	51	22	**54**	**32**
Pickup	32	21	**53**	**44**
Put	**81**	**46**	62	39
Cool	**88**	**92**	87	38
Heat	**85**	**89**	84	86
Clean	**81**	57	79	**71**
Slice	25	12	**51**	**55**
Toggle	**100**	**32**	93	11
Average	68	46	**70**	**47**

Table 2: **Subgoal success rate.** The highest values per fold and task are shown in **blue**. Note all values correspond to Path-Length-Weighted success rate metric.

Specifically, we exploit two augmentation methods; color swapping and AutoAugment [10].

Color swapping randomizes the order of the RGB channels of each frame, which yields 6 combinations in total. We randomly pick 3 of them, including the original. AutoAugment randomizes each frame with predefined image operations such as rotation, shearing, and auto-contrast. We specifically adopt the augmentation policy found for ImageNet. For the details of the policy, please refer to [10].

Each augmentation method generates two perturbed tra-

	Components				Val-Seen		Val-Unseen	
#	FPP	OCL	DF	DA	Task	Goal-Cond.	Task	Goal-Cond.
(*a*)	✓	✓	✓	✓	25.85 (18.95)	34.92 (26.44)	5.36 (3.19)	16.18 (10.44)
(*b*)		✓	✓	✓	22.32 (16.17)	30.82 (23.84)	4.51 (2.59)	16.65 (10.75)
(*c*)	✓	✓	✓		15.85 (10.02)	23.19 (15.78)	2.92 (1.35)	12.78 (6.84)
(*d*)		✓	✓		12.56 (7.05)	21.29 (13.33)	2.68 (1.32)	13.49 (7.63)
(*e*)	✓	✓			14.63 (9.80)	25.56 (18.32)	2.19 (1.23)	10.76 (7.36)
(*f*)		✓			11.71 (5.42)	20.06 (11.21)	1.83 (0.82)	11.04 (6.23)
(*g*)	✓		✓		3.90 (2.40)	11.00 (7.20)	0.50 (0.30)	7.80 (4.40)
(*h*)			✓		3.30 (1.70)	10.20 (6.10)	0.40 (0.20)	8.00 (4.00)

Table 3: **Ablation Study for Each Component of MOCA.** FPP denotes factorized perception and policy. OCL denotes object-centric localisation. DF denotes language-guided dynamic filters. DA denotes data augmentation. For each metric, we report task success rates with corresponding path weighted scores in parentheses. The absence of checkmark denotes that the corresponding component is removed.

jectories for each trajectory in training our agent. This results in one original trajectory with four augmented ones (*i.e.*, five training trajectories in total).

3. Task Type and Subgoal Ablations

Tasks in ALFRED [33] are divided into 7 high-level categories. Table 1 shows the performance of our factorized agent on each task type. On short-horizon tasks such as **Pick & Place** and **Examine**, Shridhar *et al*. [33] which is a single-branch model succeeds in some trajectories in seen environments, but has near zero unseen success rates. However, our agent outperforms them in both seen and unseen scenes by large margins. **Stack & Place** and **Pick Two & Place** are the two most complex and the long tasks in ALFRED. Our agent achieves 6.1% and 19.4% seen success rates as compared to 0.9% and 0.8% of Shridhar *et al*. It also achieves improved success rates in unseen scenes whereas Shridhar *et al*. show zero unseen success rates.

Following [33], we also examine the performance of our agent on individual subgoals. For the subgoal analysis, we use the expert trajectory to move the agent to the starting time step of the respective subgoal. Then, the agent starts inference based on the current observations. Table 2 shows the agent's performance on individual subgoals.

The **Goto** subgoal is indicative of the navigation ability of an agent. Even though navigation in visually complex and unseen environments is more challenging, our model achieves 32% as opposed to 22% of Shridhar *et al*. Although the gap between average subgoal performance of Shridhar *et al*. and our agent is relatively small, our agent drastically outperforms it on full task completion as shown in Table 1 of the main paper. This indicates our agent's ability to succeed on overall task completion and not limiting itself to memorizing short term subgoals only.

4. Model Component Ablation

We provide more results about Table 3 of the main paper including goal-condition metrics for *Model Ablations* of our agent in Table 3. We investigate the significance of each component in detail. The analysis can be found in Model Ablations in Section 4.2 in the main paper.

5. Additional Qualitative Examples

We present qualitative examples (both successes and failures) of our factorized agent and contrast it with the single-branch model by Shridhar *et al*. in the accompanied videos. Each frame in the videos shows the goal statement and step-by-step instructions. The step-by-step instruction that the agent tries to accomplish at the current time step is highlighted in yellow. When our agent performs interaction, the predicted target class of the object at that time step is shown on the top-left corner of the egocentric frame. Note that we do not show object class for Shridhar *et al*. since they produce class-agnostic masks.

In *success_1.mp4*, while the method by Shridhar *et al*. fails to navigate to right object (yellow spray bottles), our agent successfully navigates and places both of them on top of the toilet, thereby satisfying the goal statement. It implies that our Action Policy Module (APM) is able to predict accurate action sequences based on vision and language inputs.

For *success_2.mp4*, both our agent and the prior work navigate correctly to the right locations at various stages of the task. However, when the instruction asks to pick up the lettuce, our agent correctly localises and picks up the correct object. The Interactive Perception Module (IPM) of our agent which enables it to reason about object classes helps it to predict the mask of the correct object (lettuce). On the contrary, the prior work picks up a cup which was not mentioned in the instruction at all, thereby failing on the tasks even though it performs all the other actions accurately. This can be attributed to its class-agnostic nature of interaction mask prediction.

Similarly in *success_3.mp4*, while the method by Shridhar *et al*. fails to pick up the knife, due to an inaccurately localised mask under limited visibility and picks up the spat-

ula instead, our agent correctly picks up the knife and successfully accomplishes the task.

success_4.mp4 demonstrates the ability of our agent to perform the tasks in a more efficient manner. Even though the prior work successfully navigates to the cup, it takes a lot of unnecessary navigation actions which harm the path-length-weighted score considerably. In addition, after picking up the cup, it fails to navigate further and ends up being stuck at a desk and therefore fails. If our agent would have faced a similar scenario, our 'Obstruction Evasion' module would have helped the agent to evade it. On the other hand, our agent navigates to the correct objects of interest (the cup, the refrigerator, and a counter) in a more efficient path. It also performs accurate interactions and therefore accomplishes the given task.

For the *fail.mp4* video, the prior work tries to interact with an irrelevant object (cloth), instead of the tissue box and fails at completing the task. Similarly, our agent also tries to interact with the wrong target object (soap bottle) as it fails to navigate to the right position to observe that object, making it invisible. This navigational failure misleads the IPM to perceive the soap bottle as a tissue box and therefore tries to place an unintended object on top of the toilet and fails at the task.