Supplementary Material for DISCO: Embodied Navigation and Interaction via Differentiable Scene Semantics and Dual-level Control

Xinyu Xu[1] Shengcheng Luo[1] Yanchao Yang[2] Yong-Lu Li[1†] Cewu Lu[1†]

[1] Shanghai Jiao Tong University
{xuxinyu2000,yonglu_li,lucewu}@sjtu.edu.cn woodroof1998@gmail.com
[2] The University of Hong Kong
yanchaoy@hku.hk

1 Limitations

We collect data and conduct experiments in simulation, but more sim2real efforts are necessary in future works. We adopt discretized and symbolized action space in AI2THOR. Grounding actions into the physical world with robotic kinematics and dynamics leads to more comprehensive applications. We will continue to work on this path.

2 Potential Negative Societal Impacts

The proposed embodied AI techniques are not expected to cause significant social harm when utilized appropriately. Nonetheless, the regulations on the safety and privacy aspects of embodied systems remain important.

3 Introduction of ALFRED

ALFRED [6] is a large-scale dataset of long-horizon vision-language navigation and interaction tasks.
Task Types. ALFRED includes 7 task types:

- Look & Examine. Examine an object under the light (*e.g.* read a book under the light).
- Pick & Place. Pick an object and place it in a receptacle (*e.g.* pick a pencil and place it in a drawer).
- Place Two. Place two object instances in the same receptacle (*e.g.* throw two apples into the garbage bin).
- Stack. Place an object in an intermediate container then place the intermediate container in a receptacle (*e.g.* place a tomato in a pot then put the pot on a stove burner).

† Corresponding authors.

Fig. 1: Examples of vision-language navigation and interaction tasks from ALFRED [6].

– Heat & Place. Place a heated object in a receptacle (*e.g.* place a heated egg on the dining table).
– Cool & Place. Place a cooled object in a receptacle (*e.g.* place a cooled apple on the countertop).
– Clean & Place. Place a cleaned object in a receptacle (*e.g.* place a cleaned cloth in the bathtub).

We illustrate more task examples in Fig. 1.

Subgoals. All tasks mentioned above can be divided into eight subgoals. (1) GotoLocation. (2) PickUp. (3) Put. (4) Slice. (5) Toggle. (6) Heat. (7) Cool. (8) Clean. Each interactive subgoal is an instructed verb-noun pair. The proposed DISCO mainly boosts the execution efficiency of each subgoal as the primitive.

Action Space. The action space of ALFRED includes 5 navigation actions (MoveAhead, RotateRight, RotateLeft, LookUp, LookDown) and 7 interactive actions (PickUp, Put, Open, Close, ToggleOn, ToggleOff, Slice). We compose three more actions (MoveLeft, MoveBack, MoveRight) in the decision space of DISCO:

– MoveLeft = RotateLeft + MoveAhead + RotateRight,
– MoveRight = RotateRight + MoveAhead + RotateLeft,

– MoveBack = RotateLeft + RotateLeft + MoveAhead + RotateRight + RotateRight.

Each interactive action requires an object mask to identify the target object to be manipulated. In our framework, the object mask is predicted by Mask R-CNN [2].

4 Applying DISCO in ALFRED

In this section, we describe the holistic process of applying DISCO in ALFRED [6].

4.1 Language Processing

To facilitate a fair comparison, we follow the language processing module in [4] that employs fine-tuned BERTs [1] to map natural language instructions into the internal parameters of ALFRED.

First, it determines one of seven task types, as detailed in Sec. 3, based on the language instruction. Then it estimates four task arguments:

– Object. The object going to be manipulated (*e.g.* apple).
– Receptacle. Where the object will be finally placed (*e.g.* dining table).
– Movable Receptacle. A special parameter to identify the intermediate container in Stack tasks (*e.g.* a pot that holds a tomato and is placed on a stove burner).
– Slice. A boolean parameter to identify whether the object needs to be sliced (*e.g.* slicing lettuce).

The task type and four arguments are the internal parameters for ALFRED tasks, initially defined in the benchmark to form the structured task nature. We adopt the off-the-shell fine-tuned BERTs from [4] to estimate the internal parameters. Though recent large language models may present potential benefits, our experimental setup ensures a fair comparison with baselines.

4.2 Task Planning with Templates

Utilizing the estimated internal parameters from the language module, we capitalize on ALFRED's highly structured task nature to generate subgoals using predefined templates. These default templates for each task type are listed in Tab. 1. Each task plan contains multiple primitives in order to achieve the final state. Each primitive is a verb-noun pair, fed into the proposed DISCO framework in execution.

We note two special points in generating task plans. First, in cases where the object is required to be sliced, our approach introduces an initial sequence of three more specific steps: `(Pickup, Knife)`, `(Slice, Object)`, and `(Put, Receptacle)`, positioned at the beginning of the subgoal series. Second, for *Place*

Table 1: Templates on generating subgoals per task type.

(1) Look & Examine	
PickUp	Object
Toggle	Lamp
(2) Pick & Place	
PickUp	Object
Put	Receptacle
(3) Place Two	
PickUp	Object
GotoLocation	Object
Put	Receptacle
PickUp	Object
Put	Receptacle
(4) Stack	
PickUp	Object
Put	Movable Receptacle
PickUp	Movable Receptacle
Put	Receptacle
(5) Heat & Place	
PickUp	Object
Heat	Microwave
Put	Receptacle
(6) Cool & Place	
PickUp	Object
Cool	Fridge
Put	Receptacle
(7) Clean & Place	
PickUp	Object
Clean	Sink Basin
Put	Receptacle

Two tasks, we insert a `(GotoLocation, Object)` subgoal to locate the second object instance. This strategy is adopted to address the challenge of distinguishing between different instances within the same semantic class. By this approach, the agent is forced to first pick up one object, and then locate the second one before placing the first object down.

4.3 Agent Setup

We record the 4-DoF agent pose, comprising a 2-DoF position, a 1-DoF rotation, and a 1-DoF camera horizon degree. At the beginning of each task episode, we take the initial location of the agent as an anchor point. It is the center point of the modeled scene. We position the camera at a 45-degree downward angle relative to the horizon at the beginning. After each navigation step, we update the agent pose by the accumulation of discrete actions.

At the start, the agent is driven to perceive its panoramic surroundings. This is achieved by executing four sequential 90-degree rotations, enabling the agent

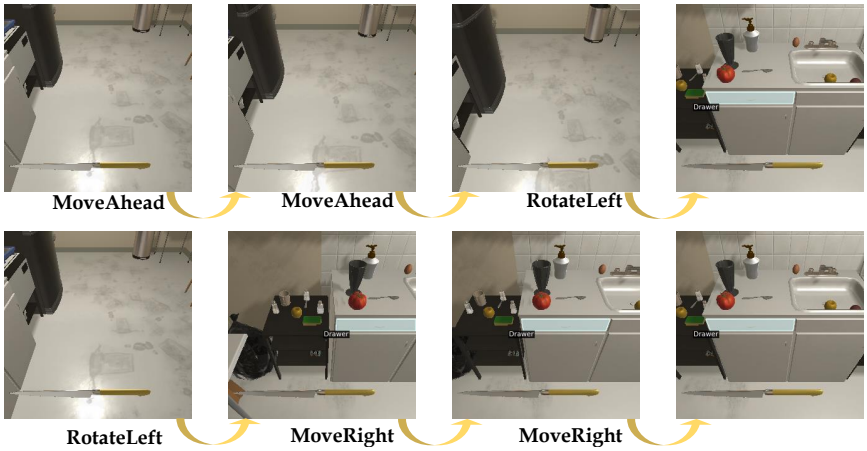

Fig. 2: Different trajectories in navigation. The top trajectory is from the original ALFRED data. The target object(drawer in white) is not visible because of a wrong view direction in moving steps, leading to learning ambiguity. The bottom trajectory is generated by us. We adjust agent rotation at the start of the fine control to force object visibility.

to learn and represent the scene comprehensively. It facilitates scene exploration and effectively finds the target object. After the initial action program, the agent sequentially executes planned task primitives.

4.4 Perception

The perception system starts with a 300×300 egocentric RGB frame. The field of view of the deployed camera is 60 degrees. We leverage three neural nets to estimate pixel-wise instance segmentation, depth, and affordances in the same resolution. Pixels are projected into the 3D space as point clouds, then localized in the allocentric map. We determine grids with more than 500 points are visible in the view. The representations of visible grids are learned by the aggregated semantic label of point clouds. Details of the perception and scene modeling are in Sec. 3.1 and Sec. 3.2, the main text.

4.5 Action Control

We employ dual-level coarse-to-fine action controls. The coarse control is based on the global scene map but the fine control is based on the local egocentric view.

Breadth-First Search (BFS) is used to determine the coarse actions. It plans the shortest navigable trajectory from the current position to the destination. Notably, the destination is a random point if the object is never found. Once it is found, the BFS destination is the localized semantics and its neighbor grids within 1 meter.

Neural policy is used to determine the fine actions. They are trained by imitating planned actions from an expert. The neural action space includes all navigation actions and an `Interact` action. At the start of the neural control, we adjust agent rotation referring to the object direction in the localized scene map. This ensures object visibility for the policy network. The neural policy control steps navigation actions and terminates until predicting the `Interact` action. Then the specific interactive action determined by the verb primitive is executed. The fine neural control holds two insights, *i.e.* object-oriented and short-horizon. First, we adjust the agent pose conditioned on the target object state, which is always visible in the fine adjustment. This reduces the ambiguity in object manipulation. Second, the dual-level controls naturally decrease the length of learning steps. Our neural policy is applied in short-horizon processes and costs fewer training steps.

After the moving steps, the agent performs specific interactions defined in Sec. 3. A notable case is to interact with receptacles. Our affordance module detects whether the receptacle is openable to plan the necessity of an open step.

More details of the control are in Sec. 3.3 of the main text.

5 Training Trajectories

We illustrate different training trajectories in Fig. 2, including the trajectory from ALFRED and those generated by us. We can find an intriguing property of different trajectories. The agent following the ALFRED trajectory may have a wrong view in the marching stage and the object is not visible in the camera view. This phenomenon causes ambiguity in actions and makes it hard to learn. In contrast, our method is object-oriented. We adjust agent rotation using the localized yaw angle at the start and force the target object to always be visible in steps for learning. The object-oriented trajectory is more learnable and predictable than default ALFRED data.

6 Additional Experiments

6.1 Analysis on Different Task Types

For a more comprehensive evaluation, we report DISCO's performances on different tasks in Tab. 2. The stack task (*e.g.* put a spoon in a mug then put the mug in a cabinet) turns out to be the most challenging for DISCO, as seen the 27.5% success rate in unseen scenes. It involves more objects to be manipulated and requires many spatial relationships to be achieved in the final state. DISCO also has subpar performances on heat and cool tasks, namely 54.4% and 52.9% success rates in unseen scenes. These two types of tasks require agents to manipulate microwaves or fridges to change object states. They turn out to be much more challenging than the common pick-place. It reveals there are also some potential improvements for DISCO to be more skilled in complex manipulations.

Table 2: Results per task type.

	Valid Seen		**Valid Unseen**	
	SR	GC	SR	GC
Look light	71.2	76.6	79.7	84.9
Pick & Place	66.9	66.9	53.0	53.0
Place two	62.1	73.1	38.3	56.7
Stack	40.8	51.4	27.5	32.4
Heat	46.7	60.3	54.4	70.6
Cool	42.0	52.7	52.9	71.2
Clean	72.3	79.9	61.1	73.0
All	57.3	63.9	55.0	65.5

Table 3: Failure analysis.

	Valid Seen	**Valid Unseen**
Success Rate	57.3	55.0
Language error	18.3	19.3
Object not found	5.0	3.0
Navigation collision	8.2	14.0
Interaction failure	10.3	7.4
Others	0.9	1.3

6.2 Failure Analysis

We conduct a comprehensive analysis of DISCO's failures in this section. Quantitative results are reported in Tab. 3. DISCO can achieve 55%+ success rate in both seen and unseen scenes, but there is still potential room for improvement. Language misunderstanding errors contribute to the largest proportion of failures, nearly one-fifth. It suggests the necessity of stronger language models in embodied applications. In some episodes, the agent fails to find the target object. A possible reason is that the object may be located in a closed receptacle and is not visible in the wild. Active exploration with commonsense knowledge is required to address this issue. Other common failures include navigation collision and interaction failure, approximately 10% for each.

6.3 Ablation Study of Differentiable Representations

The quality of differentiable representations is affected by its optimization process. We provide an additional ablation study of the optimization steps of differentiable features. We use 10 optimization steps by default but also report results using 5 and 15 optimization steps in Tab. 4. In *valid unseen* set, DISCO achieves 53.6%, 55.0%, and 54.7% success rates using 5, 10, and 15 optimization

Table 4: Ablation study of differentiable representations.

Representation	Num. Optim. Steps	Success Rate
Differentiable	5	53.6
Differentiable	10	55.0
Differentiable	15	54.7
Cell	-	42.7

Table 5: Results in interaction exploration.

	Coverage	**Precision**
IntExp [5]	22.2	8.5
DISCO(Ours)	**26.2**	**13.8**

steps respectively. And they all greatly outperform the cell representation baseline of 42.7% success rate. It validates both the effectiveness and robustness of differentiable representations.

6.4 Application in Interaction Exploration

To evaluate the generalized ability of DISCO, we conduct additional experiments in an interaction exploration task following [5]. The interaction exploration task is to quickly discover what objects can be interacted and how to interact in an embodied environment. It puts requirements on scene understanding, object interaction localization, and action planning.

We apply DISCO in interaction exploration tasks. Same to the implementation in ALFRED tasks, our perception module and differentiable scene representations can be seamlessly integrated to understand environments in interaction exploration. In the action planning stage, we localize interactable objects using the affordance module and select the nearest object as the target. Our dual-level control drives the agent to perform all possible interactions with the target object.

Following [5], we simulate in *kitchen* scenes in AI2-THOR [3]. There are 20 scenes in the training set and 5 scenes in the testing set respectively. The action space aligns ALFRED setting. The maximum time step is 1,024. We use two metrics to evaluate our method: **(1) Coverage**: the ratio of executed interactions to the maximum possible interactions. **(2) Precision**: the ratio of executed interactions to the number of attempts.

Results in the interaction exploration task are reported in Tab. 5. DISCO outperforms the IntExp [5] in both metrics, namely +4.0% in Coverage and +5.3% in Precision. It validates the generalization of DISCO.

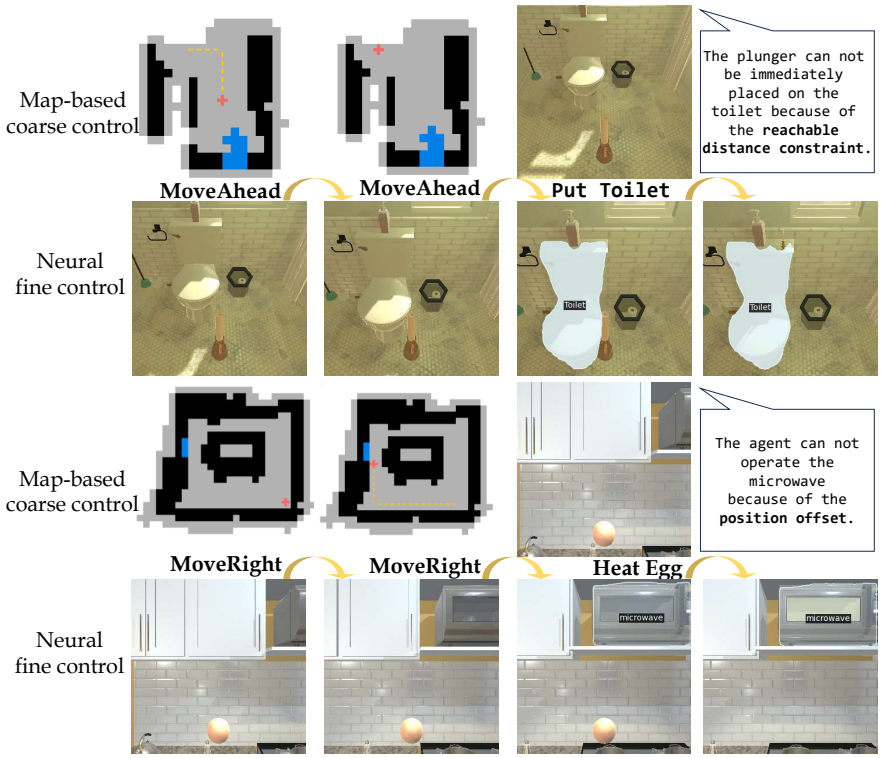

Fig. 3: Qualitative running cases. After coarse navigation, the agent may suffer from reachable distance constraint (upper) or position offset (bottom) which causes interaction failure. Fine actions perform self-adjustment to address the issues.

6.5 Qualitative Results

We provide more qualitative results in Fig. 3 to demonstrate the effectiveness of our method, as a supplement to Sec. 4.6 of the main text. In these cases, the agent can not interact with objects after the coarse control process, because of reachable distance constraint or position offset. Our dual-level controls solve the issue.

References

1. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018) 3
2. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: Proceedings of the IEEE international conference on computer vision. pp. 2961–2969 (2017) 3
3. Kolve, E., Mottaghi, R., Han, W., VanderBilt, E., Weihs, L., Herrasti, A., Gordon, D., Zhu, Y., Gupta, A., Farhadi, A.: Ai2-thor: An interactive 3d environment for visual ai. arXiv preprint arXiv:1712.05474 (2017) 8
4. Min, S.Y., Chaplot, D.S., Ravikumar, P., Bisk, Y., Salakhutdinov, R.: Film: Following instructions in language with modular methods. arXiv preprint arXiv:2110.07342 (2021) 3
5. Nagarajan, T., Grauman, K.: Learning affordance landscapes for interaction exploration in 3d environments (2020) 8
6. Shridhar, M., Thomason, J., Gordon, D., Bisk, Y., Han, W., Mottaghi, R., Zettlemoyer, L., Fox, D.: Alfred: A benchmark for interpreting grounded instructions for everyday tasks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10740–10749 (2020) 1, 2, 3