Supplementary Material For "Embodied Contrastive Learning with Geometric Consistency and Behavioral Awareness for Object Navigation"

Anonymous Author(s)

Algorithm 1 Embodied Contrastive Learning (ECL)

Input: Environment E, Online rollout buffer $\mathcal{B}$, Data buffer $\mathcal{D}$, Curiosity reward $\mathcal{R}_{Exp}(o)$, Exploration strategy E^2-CL π_θ with the visual encoder f_θ, PCL encoder f_φ
Output: Representation learning models f_θ and f_φ, Downstream training samples $\mathcal{D}$
while not converged **do**
  //Collect observations from environments using π_θ
  for each timestep t **do**
    o_t = get_obs(E), $a_t = \pi_\theta(o_t)$, o_{t+1} = step(E, o_t, a_t)
    //Relabel transitions with curiiosity reward $\mathcal{R}_{Exp}(o_t)$
    $\mathcal{B} \leftarrow \mathcal{B} \cup \{\tau = (o_t, a_t, o_{t+1}, \mathcal{R}_{Exp}(o_t))\}$
  end for
  //Update π_θ, f_θ, *and* f_φ
  Update f_θ with $\mathcal{L}_V$ and update f_φ with $\mathcal{L}_{ECL}$
  Update π_θ with $\mathcal{L}_{PPO}(\pi_\theta, \mathcal{B})$
  //Collect training samples for downstream tasks
  if record labeled data **then**
    Randomly sample a small subset from current rollout buffer $\{(x_{image}, y_{label})\} \sim \mathcal{B}$
    $\mathcal{D} \leftarrow \mathcal{D} \cup \{(x_{image}, y_{label}\}$
  end if
  Empty online rollout buffer $\mathcal{B} \leftarrow \emptyset$
end while

Table 6: Hyperparameters of ECL.

Hyperparameter	Value
Observation	(640,480), RGB
Downsample layer	AveragePooling (2, 2)
Hidden size (LSTM)	512
Optimizer of π_θ	Adam
Learning rate of π_θ	2.5×10^{-4}
Learning rate annealing	Linear
Rollout buffer length	32
PPO epochs	4
PPO mini-batches	2
Discount γ	0.99
GAE λ	0.95
Normalize advantage	False
Entropy coefficient	0.01
Value loss term coefficient	0.5
Maximum norm of gradient	0.5
Clipping ϵ	0.1 with linear annealing
Learning rate of BA and GC	2.5×10^{-4}
Optimizer of BA and GC	Adam
Number of timesteps	8
BA and GC epochs	4
Projection network	[512]
Prediction network	[512,512]
α in $\mathcal{R}_{Exp}$	1.0
β in $\mathcal{L}_V$	0.2

A SUCCESS RATE ANALYSIS OF OBJECTNAV

Fig. 5 illustrates the trend of the success rate with the number of training steps during the training process of 3DAware [62]. As shown in Fig. 4, RGB image-based semantic segmentation errors may lead to low-quality and ambiguous semantic maps, which further leads to erroneous object recognition and localization. *Agent_Succ* in Fig. 5 indicates that the agent recognizes the wrong object and performs a stop action, i.e., the agent thinks it has found the object when it actually has not. *Real_Succ* in Fig. 5 indicates the real success rate, i.e., the agent thinks it has found the object and actually does find it. As shown in Fig. 5, the difference between *Agent_Succ* and *Real_Succ* ranges from 20.1% to 27.7%, which reflects that the accuracy of object recognition seriously affects the ObjectNav performance. Therefore, one of the core ideas of this paper is to employ 2D-3D cross-modal pre-trained visual and PCL encoders to enhance object recognition and improve ObjectNav performance.

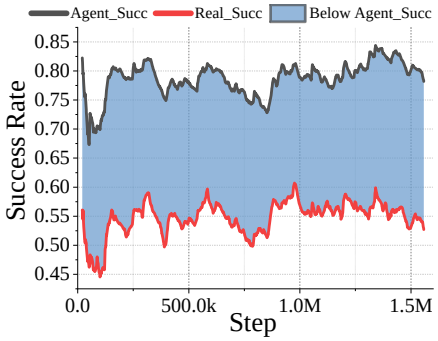

Figure 5: Success rate of ObjectNav as a function of the number of training steps.

B PSEUDO-CODE, HYPERPARAMETERS, AND TRAINING CURVES OF ECL

The details of our ECL are shown in Algorithm 1. The inputs to the algorithm include the visually realistic interactive environment E, the rollout buffer $\mathcal{B}$ for PPO, the buffer $\mathcal{D}$ for collecting retraining data, the curiosity reward $\mathcal{R}_{Exp}(o)$, the exploration strategy π_θ, the visual encoder f_θ, and the PCL encoder f_φ. To be specific, the ECL consists of three phases: (1) collecting visual observations using the exploration strategy π_θ, (2) updating the exploration strategy and the representation learning model f_θ and f_φ, and (3) collecting training data for downstream object recognition tasks. Until the training process converges, the algorithm outputs the trained representation learning models and the collected data (250K RGB images and semantic labels). The upper and lower sections of Table 6 list the hyperparameters for PPO and contrastive representation learning, respectively. We follow previous works [20, 53] as close as possible.

The training curves of our ECL are shown in Figure 6. (a)-(c) illustrate the trends of action prediction accuracy, action-aware curiosity reward $\mathcal{R}_{Exp}$, and contrastive loss $\mathcal{L}_{ECRL}$ with the number of training steps, respectively. The curiosity reward $\mathcal{R}_{Exp}$ grows with the number of training steps, which means that the agent is progressively able to seek and collect more novel visual observations while maintaining the diversity of exploration actions. The fluctuations of action prediction accuracy in Fig. 6 (a) also portend an increasing richness of visual features and the complexity of actions. It is worth noting that the action prediction accuracy is consistently at a low level. On the one hand, the increasing complexity of visual features and action sequences increases the difficulty

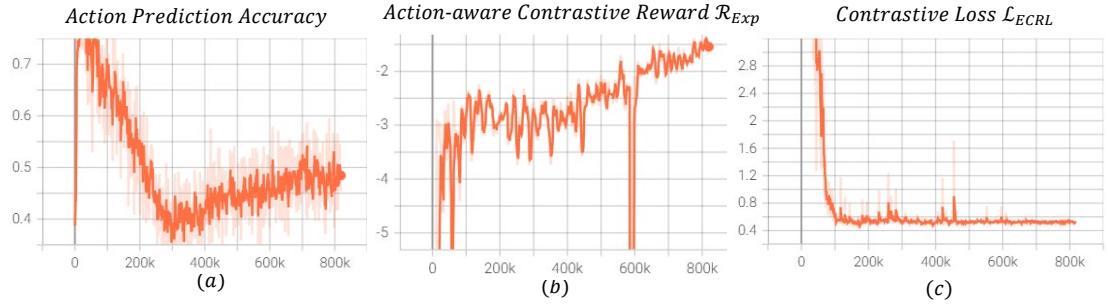

Figure 6: The training curves of ECL.

Figure 7: Examples of RGB images collected by E^2-CL for model retraining. (1) High-quality images collected by agents taking diverse actions. (2) The agent is blocked by an obstacle (bed) and is forced to slide. (3) The agent is blocked by the wall and can't move. (4) The agent's FoV is heavily obscured by obstacles.

Table 7: Effects of exploration strategy on the performance of retrained models.

Method	Train Split		Test Split	
	ObjDet	InstSeg	ObjDet	InstSeg
CRL [20]	68.78	57.16	21.93	19.44
RND [4]	71.44	60.85	25.44	22.57
E^2-CL	**72.32**	**62.11**	**25.89**	**23.81**

of action prediction. As shown in Fig. 6 (a), the action prediction accuracy is somewhat regained after sufficiently interactive learning. On the other hand, since the agent in the Habitat simulator is not equipped with collision avoidance, it is easily blocked by obstacles, resulting in collecting a portion of low-quality image-action data. The low-quality RGB images shown in Fig. 7 (2)-(4) are an important factor causing the low action prediction accuracy. The contrastive loss $\mathcal{L}_{ECRL}$ shown in Fig. 6 (c) shows a trend of decreasing, then fluctuating, and finally converging as the number of training steps increases. This trend implies adequate information exchange between the 2D visual features and the 3D geometric structures.

Table 8: Effects of different window lengths on contrastive representation learning.

Window Length	Train Split		Test Split	
	ObjDet	InstSeg	ObjDet	InstSeg
l= 4	68.98	60.47	22.78	21.05
l= 6	71.09	61.71	24.38	22.66
l= 8	**72.32**	62.11	**25.89**	23.81
l= 10	72.21	**62.83**	25.19	**24.01**
l= 12	72.02	62.28	24.80	23.19

C EXPLORATION EVALUATIONS

To demonstrate the superiority of our exploration strategy, E^2-CL, CRL, and RND are first employed to collect data for object detection and instance segmentation tasks in Gibson scenarios, respectively. By adopting the three categories of data to retrain our pre-trained visual encoder f_θ, the specific experimental results are shown in Table 7. Our method achieves the best AP metrics on both train and val splits, suggesting that E^2-CL can adequately explore diverse scenes and capture high-quality visual observations conducive to object recognition.

To intuitively evaluate CRL, RND, and E^2-CL, we employ three exploration strategies to explore five different scenarios in the Gibson dataset, respectively, as shown in Fig. 8. The dark and light gray colors in the maps indicate unexplored and explored areas, respectively. The trajectories that fade from dark blue to light blue indicate the exploration process. Intuitively, E^2-CL can explore 5 different environments to the fullest extent. Notably, RND likewise fully explores the third environment, but its trajectory is longer and more convoluted than that of E^2-CL. In the more complex fourth scene, the CRL and RND wander through narrow spaces, leading to failed explorations. However, E^2-CL can escape the narrow space and can fully perceive the fourth scene. We believe that the shorter exploration trajectories and the ability to escape from narrow spaces benefit from the design of our action-aware contrastive reward, as it encourages attempting diverse actions to discover novel visual stimuli.

D EFFECTS OF WINDOW LENGTH AND DATA MAGNITUDE

Given that different window lengths will yield different behavioral awareness learning scopes and different 3D semantic map scales in the CRL, we evaluate the effects of different window lengths l.

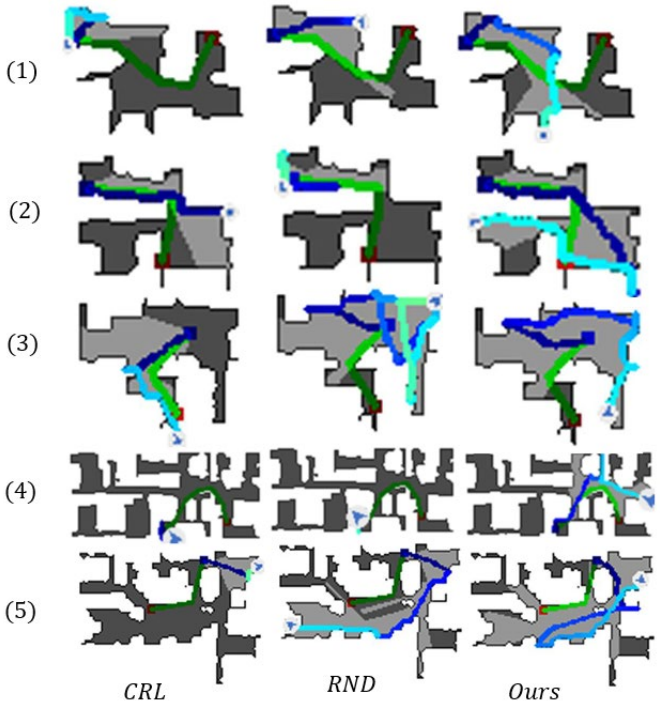

Figure 8: Qualitative comparisons of CRL, RND, and E^2-CL.

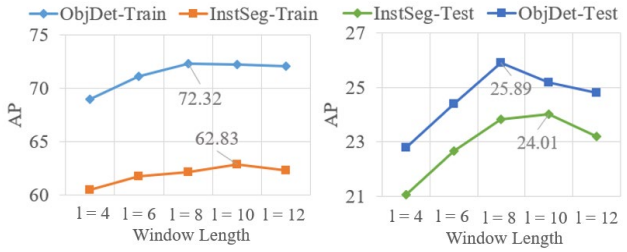

Figure 9: Effects of window length l on model performance.

Specifically, we adopt different l for CRL in the pre-training phase, respectively. The pre-training visual encoders are further retrained for the object detection and instance segmentation tasks. Table 8 and Fig. 9 show that $l = 8$ and $l = 10$ yield similarly optimal AP metrics. Interestingly, the AP metrics begin to decrease slightly when using $l = 12$. The results reflect that larger behavioral awareness learning ranges and 3D semantic map scales may negatively affect the performance of CRL. After weighing the computational cost and performance, $l = 8$ is finally used in this study.

In addition, we evaluate the effects of different data magnitudes on the retrained visual encoder's performance, and the specific experimental results are shown in Table 9. We find that the model's performance on both tasks decays severely as the data magnitude decreases uniformly. The experimental results demonstrate the necessity of adequately exploring diverse scenarios and collecting novel data.

Table 9: Effects of data magnitude on the performance of retrained models.

Data Magnitude	Train Split		Test Split	
	ObjDet	InstSeg	ObjDet	InstSeg
20% (50K)	43.49	36.81	16.71	13.92
40% (100K)	58.07	49.08	14.27	11.94
60% (150K)	66.12	57.26	19.71	16.09
80% (200K)	69.63	58.71	24.93	22.16
100% (250K)	**72.32**	**62.11**	**25.89**	**23.81**

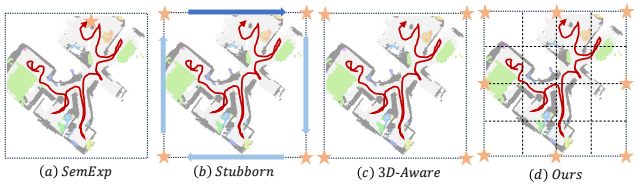

Figure 10: Visualizations of the action space for different exploration strategies.

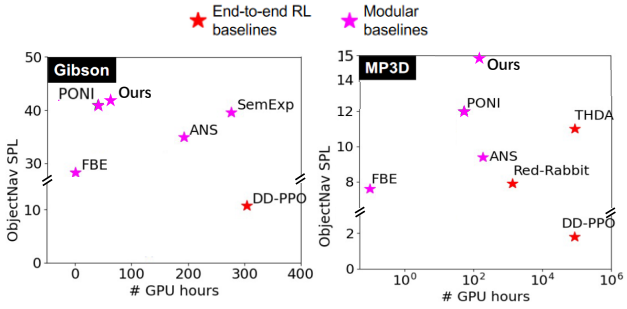

Figure 11: GPU hours is used to quantify the training cost on Gibson and MP3D datasets. Note: the MP3D plot uses a log-scale for X axis.

E CROSS-DATASET GENERALIZATION OF ECL

The following two testing schemes are utilized to verify whether our ECL pre-training learns features that are generalized across datasets:

(1) The visual encoder pre-trained on the MP3D train split is migrated to the Gibson dataset for retraining and completing the object recognition tasks. The specific comparative results are shown in Table 10.

(2) The visual encoder pre-trained on Gibson train split is integrated into our ObjectNav policy for further retraining and evaluation on the MP3D dataset. The specific comparative results are shown in Table 11. Notably, the parameters of the PCL encoder of the ObjectNav policy are randomly initialized. Since the Gibson and MP3D datasets contain different semantic categories, the corresponding 3D semantic maps have different categories and channels. Therefore, we can not directly migrate the PCL encoder across datasets.

The experimental results show that ECL pre-training across datasets likewise enhances the performance of the object recognition task and ObjectNav. As expected, the behavior-aware visual

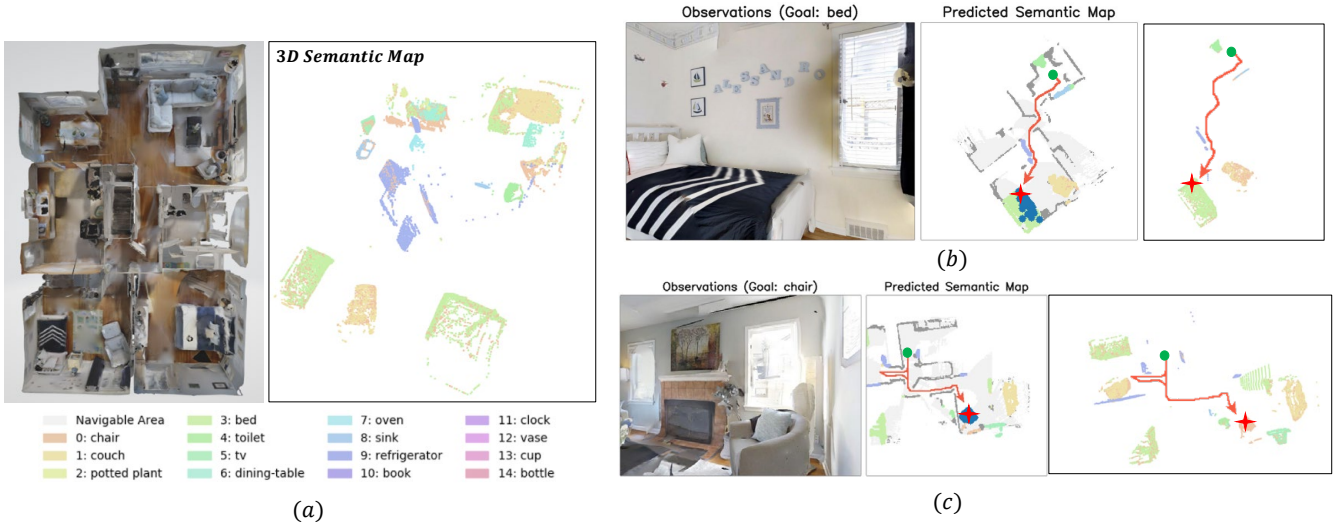

Figure 12: Visualizations of the 3D semantic maps and navigation processes.

Table 10: Cross-Dataset Generalization of ECL on Object Recognition Tasks.

Method (Venue)	Train Split		Test Split	
	ObjDet	InstSeg	ObjDet	InstSeg
OVRL [5, 55] (ICLR 2023)	67.56	54.82	22.23	20.18
Ego^2-MAP† [28] (ICCV 2023)	67.29	54.97	20.72	19.89
Pri3D [29] (ICCV 2021)	70.28	59.86	25.34	23.04
MIT [57] (ICCV 2023)	68.30	56.88	24.19	22.61
From ECL (Gibson)	72.32	62.11	25.89	23.81
From ECL (MP3D)	71.61	60.94	25.26	23.48

Table 11: Cross-Dataset Generalization of ECL on ObjectNav. † denotes the results we obtained using the official open-source code.

Method (Venue)	MP3D (val)			
	SR (%) ↑	SPL (%) ↑	DTS (m) ↓	Ext. Data
OVRL [55] (ICLR 2023)	28.6	7.4	-	no
Ego^2-MAP [28] (ICCV 2023)	29.0	10.6	5.17	yes
3D-Aware† [62] (CVPR 2023)	33.4	13.6	5.03	no
Ours (ECL-MP3D)	34.8	14.7	4.95	no
Ours (ECL-Gibson)	34.1	14.3	5.06	no

features that fuse 2D-3D cross-modal scene priors can be generalized to novel and unseen scenes. This phenomenon suggests that our ECL method can adequately represent the visual patterns and structural cues of the scenarios in an embodied manner, which are friendly to downstream tasks.

F DETAILS OF THE OBJECTNAV STRATEGY

The SOTA modular approaches [9, 39, 45, 62] usually decouple ObjectNav into a scene exploration sub-task and an object recognition sub-task to solve the problems of "Where to look?" and "How to localize and navigate to a specific object?". Therefore, utilizing scene structures and semantic contexts to achieve efficient exploration decisions is a prerequisite for recognizing and navigating to a given object category. SemExp [9] proposes to use the entire local map as an action space for the exploration sub-strategy. Since exploration actions are regressively predicted, the huge action space will result in inefficient sampling, further reducing the efficiency of exploration decisions. To alleviate this problem, Stubborn [39] and 3D-Aware [62] propose heuristic and learning-based corner-guided exploration strategies, respectively. Specifically, these two works argue that exploration actions only need to provide direction guidance to the agent, and therefore define the exploratory action space as the four corners of a local map, as shown in Fig. 10 (b) and (c). The design of our exploration sub-strategy continues this thought, but in practice the exploration action space is expanded from four corners to eight directions as shown in Fig. 10 (d). For each exploration decision-making, the agent picks one of the eight locations as the exploration goal g^t_{Exp}.

Notably, the point cloud encoder is employed to extract features of the local 3D semantic map corresponding to the l+1 image frames. Similar to humans localizing objects in the current field of view, we believe that agents are most likely to recognize an object goal within the local 3D region where the agent is currently located. Such an assumption alleviates the computational burden imposed by larger-scale point clouds. Notably, Gated Recurrent Units (GRU) are utilized to pass local features in the temporal dimension, facilitating long-term exploration and object recognition, as shown in Fig. 3.

G ANALYSIS OF COMPUTATIONAL COST

As shown in Fig. 11, we quantify training cost using GPU hours used to train the model. Our method costs relatively little to train but achieves the best ObjectNav performance. Specifically, our method is trained on the Gibson and MP3D datasets for about 70 GPU hours and 195 GPU hours, respectively. In addition, we find that end-to-end methods typically cost more training hours than modular methods.

H MORE OBJECTNAV DEMOS

Fig. 12 (a) illustrates the 3D semantic map for a specific scenario. The background has been omitted to emphasize the category and shape of the objects. Fig. 12 (b) illustrates the efficient navigation processes with a bed and a chair as target categories, respectively.