Supplemental Material for " Symmetry-aware Neural Architecture for Embodied Visual Exploration"

Shuang Liu
RIKEN Center for AIP
shuang.liu.ej@riken.jp

Takayuki Okatani
Tohoku University, RIKEN Center for AIP
okatani@vision.is.tohoku.ac.jp

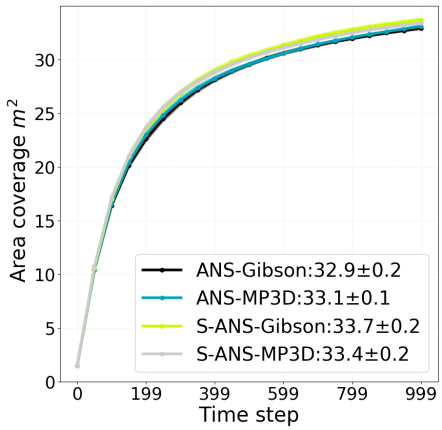
A. More Experimental Results

A.1. Experimental Results of Training on MP3D

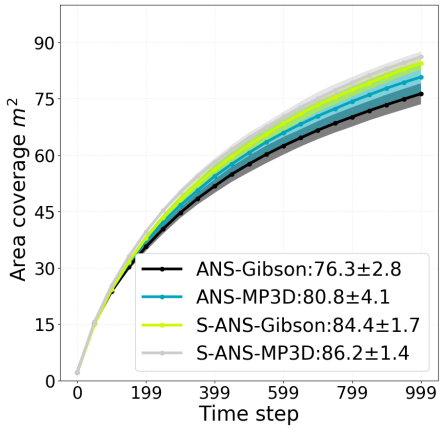
In the main paper, we show the results of experiments in which we train the models on the Gibson dataset; the models are then tested on Gibson and the MP3D dataset. We show here the results when we train the models on MP3D and test them on Gibson and MP3D. Specifically, we evaluate two models, the original ANS and the proposed S-ANS. Fig. 1 shows the results including those trained on Gibson (already shown in the main paper). Method $\mathbb{X}$ trained on Gibson and MP3D is denoted by $\mathbb{X}$-Gibson and $\mathbb{X}$-MP3D, respectively.

We can make the following observations. First, it is seen from Fig. 1(b) that when trained and tested on MP3D, S-ANS outperforms ANS by a large margin of $5.4m^2$ (i.e., S-ANS-MP3D = $86.2m^2$ vs. ANS-MP3D = $80.8m^2$)). This validates the effectiveness of the proposed method (i.e., S-ANS). Second, it is also seen from Fig. 1(b) that the performance gap between models trained on Gibson and MP3D is smaller for S-ANS (i.e., S-ANS-Gibson = $84.4m^2$ vs. S-ANS-MP3D = $86.2m^2$) than for ANS (i.e., ANS-Gibson = $76.3m^2$ vs. ANS-MP3D = $80.8m^2$). Generally, we may consider the performance of models trained and tested on the same dataset as the upper bound of their performance. S-ANS is closer to it, supporting our conclusion that the proposed approach better handles the domain gap of the two datasets by equipping the network with the symmetries necessary for the task.

Third, when tested on Gibson, the gaps between the models and between training datasets are small, as shown in Fig. 1(a). Thus, the above two tendencies are not observed. We believe this is because Gibson contains smaller scenes and is simpler in complexity than MP3D. Thus, models trained on MP3D tend to achieve good performance on Gibson, e.g., ANS-Gibson = $32.9m^2$ vs. ANS-MP3D = $33.1m^2$.



(a) Tested on Gibson



(b) Tested on MP3D

Figure 1. Exploration performance (in area coverage, m^2) of ANS trained on Gibson and MP3D and S-ANS trained on Gibson and MP3D when tested on (a) Gibson and (b) MP3D. The method $\mathbb{X}$ trained on Gibson and MP3D is denoted by $\mathbb{X}$-Gibson and $\mathbb{X}$-MP3D, respectively.

A.2. Qualitative & Quantitative Analysis for Invariant Representation

We experimentally evaluate rotation invariance of the critic of S-ANS. Specifically, we compute the standard de-

	Gibson	MP3D
ANS	0.115	0.160
S-ANS	0.078	0.084

Table 1. Rotation invariance (i.e., std of (1)) of the critics of ANS and S-ANS trained on Gibson seen over the evaluation episodes of Gibson and MP3D.

viation of its output and the similarity of its feature representations over inputs with different orientations.

To compute the standard deviation of the critic's output over input rotation, we firstly sampled Q state inputs $s_i, i = 1, 2, \dots Q$ of the global policy from the evaluation episodes of Gibson ($Q = 1988$) and MP3D ($Q = 3960$), respectively. Then, we compute a rotated state inputs set $S^* = \{s_i^k | s_i^k = r^k \cdot s_i, i \in \{1, 2, \dots, Q\}, k \in \{0, 1, \dots, K-1\}\}$ for all the samples, where r^k represents rotating s_i by $2\pi k/K$[rad] about its center. Then, the standard deviation is given by

$$std = \frac{1}{Q} \sum_{i=1}^{Q} \sqrt{\sum_{k=0}^{K} (y_i^k - \bar{y_i})^2}, \quad (1)$$

where $\bar{y_i} = \frac{1}{K-1} \sum_{k=0}^{K-1} y_i$ and $y_i^k = q(s_i^k)$; $q(\cdot)$ represents the function approximated by the critic. A smaller std indicates better rotation invariance.

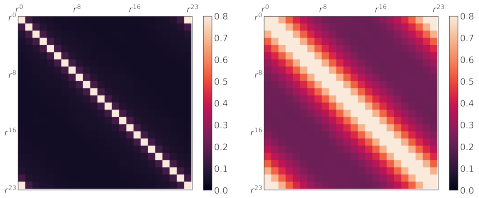
Table 1 shows std's of the critic of ANS and that of S-ANS (both trained on Gibson) when we set $K = 24$. It is seen that S-ANS achieves better rotation invariance than ANS for the both test datasets. It is worth noting that S-ANS employs $p4$ G-convolution, which theoretically attains only invariance to 90 degree rotations, and has fully connected layers that are not invariant to rotation; it nevertheless achieves better invariance over $K = 24$ sampling of the rotation angles.

Next, we evaluate the similarity of the internal features of ANS and S-ANS over rotated inputs. We use the feature vector before the fully-connected layers for each model. For this purpose, we compute the similarity between two rotated inputs as

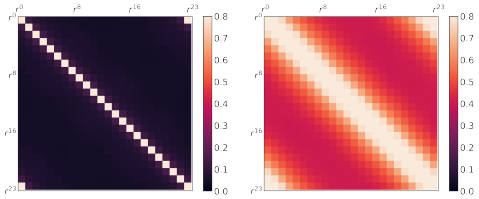
$$sim(\xi(s^\alpha), \xi(s^\beta)) = \frac{1}{Q-1} \sum_{i=0}^{Q-1} \frac{\xi(s_i^\alpha) \cdot \xi(s_i^\beta)}{\|\xi(s_i^\alpha)\| \cdot \|\xi(s_i^\beta)\|}, \quad (2)$$

where $\alpha, \beta \in \{0, 1, \cdots, K-1\}$; $s^\alpha = r^\alpha \cdot s$, $s \in S^*$; and $\xi(\cdot)$ represents the function approximated by the layers before fully connected layers in the critic networks.

Fig. 2 shows the matrices storing the above similarity as elements for ANS and S-ANS over the evaluation episodes of Gibson and MP3D. The average similarity increases from 0.06 of ANS to 0.40 of S-ANS on Gibson and from 0.07 to 0.55 on MP3D, respectively. These verify that S-ANS achieves better rotation invariance in its feature representation.



(a) Avg.=0.06 (b) Avg.=0.40



(c) Avg.=0.07 (d) Avg.=0.55

Figure 2. Similarity of internal features over rotated inputs for (a) ANS on Gibson, (b) S-ANS on Gibson, (c) ANS on MP3D, and (d) S-ANS on MP3D. The two models are trained on Gibson. Avg. indicates the mean value except the diagonal elements.

B. Implementation Details of FBE-RL

This section gives implementation details of FBE-RL. FBE-RL is a RL based Frontier based exploration (FBE). It is created by combining FBE and the global policy network of ANS. Concretely, FBE-RL first computes the frontiers of the local map h_t^l, gaining its frontier map $m_f \in \mathbb{R}^{G \times G}$. The elements on h_t^l are 0 except for those at frontiers. Then it is combined with the map of long-term goal m^*, computed by the global policy network of ANS, to obtain a frontier likelihood map $m'_f \in \mathbb{R}^{G \times G}$ by element-wise multiplication $m'_f = m^* \odot m_f$. At last the the normalized frontier likelihood map $m''_f(x, y)$ is computed by the softmax function

$$m''_i = \frac{e^{m_i}}{\sum_j e^{m_j}}, \quad (3)$$

where m_i and m''_i is the ith element of m'_f and m''_f respectively. A long-term goal is sampled from m''_f for navigation.