Symmetry-aware Neural Architecture for Embodied Visual Navigation

Shuang Liu¹ · Masanori Suganuma² · Takayuki Okatani^{1,2}

Received: 8 November 2022 / Accepted: 13 September 2023 / Published online: 25 October 2023 © The Author(s) 2023

Abstract

The existing methods for addressing visual navigation employ deep reinforcement learning as the standard tool for the task. However, they tend to be vulnerable to statistical shifts between the training and test data, resulting in poor generalization over novel environments that are out-of-distribution from the training data. In this study, we attempt to improve the generalization ability by utilizing the inductive biases available for the task. Employing the active neural SLAM that learns policies with the advantage actor-critic method as the base framework, we first point out that the mappings represented by the actor and the critic should satisfy specific symmetries. We then propose a network design for the actor and the critic to inherently attain these symmetries. Specifically, we use *G*-convolution instead of the standard convolution and insert the semi-global polar pooling layer, which we newly design in this study, in the last section of the critic network. Our method can be integrated into existing methods that utilize intermediate goals and 2D occupancy maps. Experimental results show that our method improves generalization ability by a good margin over visual exploration and object goal navigation, which are two main embodied visual navigation tasks.

Keywords Embodied visual navigation · Symmetry · Generalization · Equivariance · Invariance

1 Introduction

Embodied visual navigation is one of the key problems of autonomous navigation and has attracted increasing attention recently. Researchers have studied various target-oriented tasks of visual navigation so far. These include point goal navigation aiming to reach a given coordinate in an environment (Ye et al., 2021b), object goal navigation seeking to find a specific object in an environment (Chaplot et al., 2020b; Gupta et al., 2017; Zhu et al., 2017), image goal navigation focusing on navigating to a location in an environment designated by an image (Savinov et al., 2018; Singh Chaplot et al., 2020), reconstruction navigation deciding where to look next to reconstruct a scene (Jayaraman & Grauman,

Communicated by Gang Hua.

 Takayuki Okatani okatani@vision.is.tohoku.ac.jp
 Shuang Liu shuang.liu.ej@riken.jp
 Masanori Suganuma suganuma@vision.is.tohoku.ac.jp

¹ RIKEN Center for AIP, Tokyo 103-0027, Japan

² GSIS, Tohoku University, Sendai 980-8579, Miyagi, Japan

2018; Seifi & Tuytelaars, 2019), and room goal navigation aiming to navigate to a specified room (Wu et al., 2019). Previous methods are categorized into intermediate-goal-free navigation and navigation with intermediate goals. The former methods (Chen et al., 2019; Gupta et al., 2017; Mishkin et al., 2019; Mirowski et al., 2016) predict step-by-step actions that moving agents towards the final goal. The latter (Chaplot et al., 2020c, a; Wani et al., 2020; Chen et al., 2021) iteratively computes subgoals to guide agents reaching the final goal, inspired by hierarchical learning (Nachum et al., 2018).

This study considers such visual navigation tasks with particular focuses on visual exploration and object goal navigation with intermediate goals. Visual exploration is universal and task-agnostic; an agent is asked to visit all the navigable areas of an unknown environment as quickly as possible. It can serve as an upstream task for the above targetoriented tasks. For instance, the agent may first explore the unseen environment and establish knowledge about the environment, after which the agent utilizes it to perform specific tasks efficiently (Savinov et al., 2018). On the other hand, object goal navigation is task-oriented, which aims at searching for given target objects in novel environments as quickly as possible.



Numerous methods have been proposed for the problem of visual exploration so far. The existing methods are divided into two categories, heuristic methods (Yamauchi, 1997; Dai et al., 2020; Visser et al., 2008) and learningbased methods(Chaplot et al., 2020a; Ramakrishnan et al., 2020; Yu et al., 2021). Recent studies have mainly considered learning-based methods. They employ simulators of virtual environments, such as Habitat (Szot et al., 2021), to consider the problem under realistic conditions. In such simulators, photorealistic visual inputs from virtual environments are available. To predict the optimal action from the agents' input, including the visual inputs, learning-based methods typically use reinforcement learning (RL) to acquire the mapping between the two from an enormous amount of agents' interaction with the virtual environments.

Similarly, previous studies mainly consider learningbased approach for object goal navigation. Most existing methods (Mayo et al., 2021; Shen et al., 2019; Ye et al., 2021a; Chaplot et al., 2020c; Zhang et al., 2021; Pal et al., 2021) address it by learning policies in reinforcement manner. Models compute actions to guide agents to approach target goals given partial observations of the environments, according to the learned object-object relationship.

Although these learning-based methods have achieved reasonably good exploration and object goal navigation performance, the learning-based methods are inherently vulnerable to the statistical deviations between training and test data, as Gibson (Xia et al., 2018) and Matterport3D (MP3D) (Chang et al., 2017) datasets shown in Fig. 1. Specifically, a model trained on a set of training data works well on novel environments that are statistically similar to the training data but yields suboptimal performance on novel environments that are dissimilar, i.e., out-of-distribution (OOD) inputs.

To apply the learned models to real-world environments, it is necessary to increase their generalization ability in the above sense. How is it possible? A promising approach is



Fig. 1 Environment layout from the Gibson and Matterport3D environments. Learning-based navigation methods will encounter challenges when there is a gap between training and test data, e.g., using Gibson for training and Matterport3D for test. The notation #*Area* denotes the obstacle space area in square meters; #*CC* represents the number of connected components in the map

data augmentation, and it has been widely adopted in both deep learning (Calimeri et al., 2017; Madani et al., 2018; Liu et al., 2019) and reinforcement learning (Raileanu et al., 2021; Laskin et al., 2020; Yarats et al., 2021). However, this leads to an increase in training cost, and more importantly, data augmentation does not provide a guarantee of achieving the desired properties.

In this study, we pay attention to the inductive biases specific to the task, namely equivariance and invariance. While acknowledging the presence of diverse types of environment variation that contribute to OOD challenges, our research delves into the specific investigation of the roles played by equivariance and invariance. Using them, we want to narrow the solution space for RL and thus avoid overfitting to training data, aiming to improve the above generalization ability. Specifically, we employ the framework of active neural SLAM (ANS) (Chaplot et al., 2020a), which is the most successful for the task, and consider how to do the above within the framework.

ANS is a universal visual navigation framework using 2D occupancy maps. We will refer to it and its variants by ANS, which have achieved great success in visual exploration (Chaplot et al., 2020a; Ramakrishnan et al., 2020), object goal navigation (Chaplot et al., 2020c), visual-audio navigation (Chen et al., 2021), and so on. ANS is a method with a modular and hierarchical structure.

The module that plays a key role in ANS is the neural network that represents the global policy. (The problems solved by the other modules are not so hard. For a comprehensive understanding of these modules, we have presented detailed information in Sect. 3.2.) ANS employs the advantage actorcritic (A2C) method to learn the global policy. The actor yields a 2D likelihood map for the long-term goal, whereas the critic predicts the future accumulated reward as a scalar value. We provide a thorough explanation in Sect. 3.2.

In this paper, we first point out that specific symmetries should exist for each mapping represented by the actor and the critic. Concretely, the output predicted by the actor should be equivariant to translation and rotation of its input having the form of a 2D map. The reward predicted by the critic should be invariant to input rotation and should not be invariant to input translation. Fig. 2 illustrates the impact of rotation equivariance and invariance on the inputs and outputs of the actor and critic networks. The input map m_{90} is obtained by rotating m_0 by 90 degrees clockwise. The actor with rotation equivariance generates a long-term goal A_{90} (indicated by the red solid line), corresponding to a 90-degree rotation of A_0 , whereas the actor without equivariance fails to achieve this (red dashed line). Similarly, the critic with rotation invariance produces a scalar C_{90} (indicated by the blue solid circle) equal to C_0 , while the critic without invariance cannot achieve this (red dashed circle). However, standard



Fig. 2 Illustration of the effects of rotation equivariance over the actor and rotation invariance over the critic, respectively. See the text for explanations. Additional desired symmetries are presented in Table 1

convolutional neural networks (CNNs), employed in ANS, do not by nature have these symmetries.

To cope with this, we design the network to inherently possess the desired symmetries for the actor and the critic. Specifically, we redesign the global policy network of ANS as follows. First, we employ p4 G-convolution (Cohen & Welling, 2016), which approximately achieves rotation equivariance, for all the convolutional layers and replace the max pooling with blur pooling (Zhang, 2019), which achieves translation equivariance more accurately. Then, the resulting network representing the actor becomes equivariant to input translation and approximately to input rotation. For the critic, employing the same stack of convolutional layers (rigorously, it is shared by the actor), we design and place a new building block named a semi-global polar pooling (SGPP) layer on top of the convolutional block. The SGPP layer applies polar mapping to its input and then pooling in the circumferential direction in the input map. Then, the resulting critic becomes not invariant to input translation and approximately invariant to input rotation.

Our method can be combined easily with the navigation methods with intermediate goals using 2D occupancy maps, to improve their generalization ability over many tasks. This study demonstrates it by conducting experiments over visual exploration and object goal navigation tasks, using the Gibson (Xia et al., 2018) and the MP3D (Chang et al., 2017) datasets. The results show that the generalization is enhanced by a good margin on both tasks, by integrating our method with ANS, proving the approach's effectiveness.

This paper extends our previous study that was published as a conference paper (Liu & Okatani, 2022). The previous study focuses on the visual exploration task, aiming to enhance the generalization ability of models for the task. The present study extends it to be able to consider another task, object goal navigation, improving its generalization ability. Specifically, we first review related work on object goal navigation. Choosing one of the state-of-the-art methods for it, we then show how to integrate our proposed modules with it. Finally, we experimentally compare the resulting method with baseline methods to validate the effectiveness of the proposed approach. In addition, we provide more quantitative and qualitative analyses of how well our method learns symmetry.

2 Related Work

2.1 Visual Navigation

Classical navigation methods leverage simultaneous localization and mapping (SLAM) to construct environmental maps, including occupancy map (Cadena et al., 2016) and topological map (Savinov et al., 2018). Agents navigate to points in the environments. This is known as map-based navigation (Bonin-Font et al., 2008). In contrast with the map-based navigation that assumes a known map, targetdriven navigation tasks have recently emerged, which are termed mapless navigation. In these tasks, the movement of the agent is determined by visual cues provided by the environment. Example tasks are object-goal, room-goal, and image-goal navigation.

Object-goal navigation requires agents to navigate to designated objects in an unknown environment. Either implicitly learned (Zhu et al., 2017) or explicitly encoded (Du et al., 2020; Lv et al., 2020) relationship of objects is utilized to facilitate finding the objects. Room-goal navigation requires agents to arrive at a specific room as quickly as possible. A method is proposed in (Wu et al., 2019) to learn a probabilistic relation graph to acquire prior knowledge about the layout of environments.

Image-goal navigation asks agents to navigate to a location in an unknown environment that is specified by its image. A popular solution is to take both current and goal observation as inputs and employ a Siamese network to perform the navigation effectively. The method proposed in (Mezghani et al., 2021) combines an attention mechanism with the Siamese network for building the memory of environments, which is then used by the policy for image-driven navigation. The work (Choi & Oh, 2021) follows the same formulation based on a Siamese network but exploits information obtained through keypoint matching, generating a self-supervised reward.

2.2 Visual Exploration

Visual exploration has received considerable attention due to its task-agnostic nature (Jayaraman & Grauman, 2018; Mezghani et al., 2020; Chaplot et al., 2020a; Ramakrishnan et al., 2020; Chen et al., 2019). Classical approaches continually select vantage points, such as frontier points (Yamauchi, 1997), to visit. The downside of these approaches is that they are vulnerable to external noises such as sensor noises.

The recent boom of deep reinforcement learning (RL) has created a new wave of development of visual exploration. Numerous works have casted the visual exploration as a partially observable Markov decision process and solve it under the framework of RL (Jayaraman & Grauman, 2018; Mezghani et al., 2020; Beeching et al., 2020; Nagarajan & Grauman, 2020; Pathak et al., 2017; Chen et al., 2019; Chaplot et al., 2020a; Ramakrishnan et al., 2020, 2021; Qi et al., 2020). Most of them use an actor-critic architecture (Beeching et al., 2020; Nagarajan & Grauman, 2020; Pathak et al., 2017; Chaplot et al., 2020a; Ramakrishnan et al., 2020, 2021).

The actor learns a navigation policy, and the critic estimates a value function given the agent's current state. Existing approaches to improve performance have been geared towards novel designs of reward, such as coverage, curiosity, and novelty (Ramakrishnan et al., 2021), or novel architecture design, e.g., the hierarchical architecture (Chaplot et al., 2020a). There is a study (Chen et al., 2019) that proposes to utilize area coverage reward along with supervision from human experience to learn policies.

An excellent standard framework, which is actor-critic based as well, was constructed by (Ramakrishnan et al., 2021) to evaluate multiple rewards for exploration. The approaches above all solve the exploration problem in an end-to-end manner, directly mapping visual data to actions.

In contrast to this, the works (Chaplot et al., 2020a; Yu et al., 2021) tackled it hierarchically by computing intermediate goals. It divided the mapping into two steps. First, an intermediate target point, which was similar to vantage points in classical exploration methods, was computed by a global policy network. Then a local policy network produces actions to reach the waypoints generated by the path planner according to the global target point. Occupancy anticipation (Ramakrishnan et al., 2020) also predicts and rewards invisible spatial maps to improve map accuracy.

2.3 Object Goal Navigation

Unlike visual exploration, object goal navigation is taskoriented and target-driven. It asks agents to find target objects in an unknown environment. Most existing methods learn object-object relationships using reinforcement learning and utilize the learned relationships to search the target object given partial observations of the environment. The objectobject relationships are usually represented by spatial or graph memory. Chaplot et al. (2020c) and Wani et al. (2020) utilize 2D semantic occupancy maps as spatial memory to represent object-object relationships. The maps are computed by projecting semantic point clouds onto the ground in the direction of gravity. Each object occupies a single channel of a semantic map. Graph memory is leveraged to represent explicitly object-object (Du et al., 2020), parent-target (Pal et al., 2021), object-zone (Zhang et al., 2021) relationship. Cues can be provided to facilitate finding the target object once the object in the graph memory appears. Besides designing a new memory mechanism representing object-object relationships, other methods improve object goal navigation by proposing attention probability model(Mayo et al., 2021), using auxiliary tasks(Ye et al., 2021a), fusing diverse visual representations (Shen et al., 2019) and so on.

So far, existing methods have produced impressive results where the testing environments are similar to training environments over visual exploration and object goal navigation, in terms of layout, area, etc. Nevertheless, they still struggle with the cases where training and testing environments are different, hindering their deployment in practice. It is primarily because RL notoriously struggles with (poor data efficiency and) generalization capabilities. We aim to enhance generalization ability for visual navigation by injecting inductive bias about symmetry into networks.

2.4 Equivariance and Invariance

Recent years have witnessed the great success of convolutional neural networks (CNNs) in computer vision tasks. CNNs have built-in translation-equivariance, and those with downsampling/pooling operations have built-in local translation invariance. These contribute to the successes in applications to various problems. It could be possible to learn these symmetries from data if the model has sufficient parameters. However, it will increase the risk of overfitting.

A function is equivariant if the output changes in the same way that the input changes (Goodfellow et al., 2016). Novel convolution layers with different equivariance have been proposed so far, group equivariant convolution networks (Cohen & Welling, 2016; Dieleman et al., 2016), steerable convolution networks and harmonic networks for rotation equivariance, scale equivariance (Lindeberg, 2021; Worrall & Welling, 2019), and permutation equivariance (Thiede et al., 2020). Such convolution layers equipped with various types of equivariance have been proven to be beneficial to better performance in tracking (Sosnovik et al., 2021), classification, trajectory prediction (Walters et al., 2021), segmentation (Müller et al., 2021), and image generation (Dey et al., 2020). However, these methods have never been applied to visual navigation.

A primary purpose of pursuing various types of equivariance is to achieve a certain type of invariance. A mapping is called invariant if the output remains the same no matter how the input changes. Although data augmentation and invariance-oriented loss functions (Cheng et al., 2016) may enhance global invariance, it is not guaranteed to generalize to OOD data. By contrast, global invariance can be imposed by a global pooling layer following equivariant layers. Global rotation invariance is enforced in (Cheng et al., 2016) for texture classification by combining equivariant convolution layers with a global average pooling layer in this order. Several studies (Lindeberg, 2021; Sosnovik et al., 2020) attain global scale-invariance by combining layers having scale-equivariance with global max pooling.

3 Symmetry-aware Neural Architecture

3.1 Problem of Visual Exploration and Object Goal Navigation

This study considers visual exploration and object goal navigation tasks. Visual exploration asks an agent to explore an unknown 3D environment, e.g., a floor of a building, that allows only 2D motion. The goal for the agent is to go everywhere in the environment it can go while creating an environment's 2D map. Object goal navigation requires an agent to quickly find a designated object specified by instructions in a novel environment. We employ the Habitat simulator (Savva et al., 2019) as the framework for studying both tasks.

For both tasks, the agent receives several inputs from the environment, i.e., an image or a depth map of the scene in front of it, the agent's pose, and an odometry signal measuring its motion. Then, the agent moves to visit the environment by feeding the actuation signal to its motor. Thus, the problem is to compute the actuation signal and update the environment map at every time step, given the history of the inputs the agent has received until then. See the literature (Chaplot et al., 2020a, b) for more details.

Recent studies have formulated the problem as learning policies that yield actions (or intermediate representation leading to actions) from the received inputs; most of them employ reinforcement learning (RL) (Chen et al., 2019; Chaplot et al., 2020a; Ramakrishnan et al., 2020; Jayaraman & Grauman, 2018; Chaplot et al., 2020b). Recently, Chaplot et al. (2020a) proposed a method named active neural SLAM (ANS) for visual exploration, having established the new state-of-the-art. Subsequently, Chaplot et al. (2020b) extended ANS to goal-oriented semantic exploration (SemExp) for object goal navigation by integrating semantic knowledge and object goal reward. Our study is built upon ANS and its variant SemExp. We will first revisit and summarize them below.

3.2 Reivisiting Active Neural SLAM (ANS) and Its Variants

ANS has a modular and hierarchical structure to better solve the visual exploration problem. Instead of learning a direct mapping from the inputs to an action, ANS learns two different policies in a hierarchy, i.e., a global policy that yields an intermediate goal, named long-term goal, in the 2D environment map and a local policy that yields actions to approach a short-term goal, which is subsampled from the path to the long-term goal. More specifically, ANS consists of four modules, i.e., a neural SLAM module, a global policy module, a path planner, and a local policy module.

The neural SLAM module computes a local top-view 2D egocentric map $p_t \in [0, 1]^{2 \times v \times v}$ and estimates an accurate agent pose from the current inputs. (The first and second channel of p_t represents the obstacle and the explored region at time t, respectively.) Then, the computed local egocentric map is registered to the global map $h_t \in [0, 1]^{4 \times M \times M}$ using the estimated pose. (The first and second channel of h_t represents the obstacles and explored area at time t respectively.) The third channel indicates the computed agent position. The last channel records the path that has been visited by the agent.)

The global policy module makes ANS the most distinct from other methods. It predicts a long-term goal g_t given two different views h_t^g and h_t^l of the latest map h_t ; h_t^g is a rescaled version of h_t , and h_t^l is a local view cropped from h_t with the agent position as its center; and h_t^g and h_t^l have the same size $G \times G$ and are concatenated in the channel dimension.

The module is implemented as an advantage actor-critic (A2C) network (Mnih et al., 2016). As shown in Fig. 3, it consists of an actor network and a critic network. Each network has a similar design of a convolutional block, which consists of a stack of convolutional and max-pooling layers, and subsequent fully connected layers. In a standard design, they share the convolutional block.

The actor network outputs a 2D likelihood map of the long-term goal g_t represented in the coordinates of the global map h_t^g . It learns a global policy, denoted by $g_t = \pi(s_t | \theta_G)$, where $s_t = (h_t^l, h_t^g)$ is the state at time *t* and θ_G are parame-



Fig. 3 Overall architecture of Symmetric Global Policy Network (using *G*-convolution, blur pooling and SGPP) and Global Policy Network of ANS (using convolution, max pooling and without using SGPP). The red dashed circle illustrates the feature maps produced by p4G-convolution assuming its input is **E** and the filter is a horizontal edge extractor (Color figure online)

ters. The critic network represents a value function $V(s_t|\theta_V)$; $V(s_t|\theta_V)$ estimates the expected future accumulated reward, area coverage in the exploration task, with the agent, which is currently at state s_t and will take actions by following the current policy $\pi(s_t|\theta_G)$. The parameters θ_G and θ_V are optimized to maximize the area coverage.

The other two modules, i.e., the path planner and the local policy module, play the following roles. The path planner computes the shortest path from the current location to the long-term goal, which is subsampled to generate a number of short-term goals. The local policy module predicts actions to reach the next short-term goal.

SemExp extends ANS to address object goal navigation. It upgrades the global policy of ANS mainly in two aspects. The first is the input map fed to the global policy. It combines a semantic layout map $h_t^s \in [0, 1]^{C \times G \times G}$, where *C* is the number of semantic categories, with the egocentric maps, i.e., h_t^g and h_t^l , inputting it into the global policy. To obtain h_t^s , SemExp first predicts semantic segmentation maps for the RGB inputs using pretrained Mask-RCNN (He et al., 2017) and combines them with corresponding scene depths, creating semantic point clouds of the environment. Then it converts the semantic point clouds to voxel representation and projects it to a 2D semantic layout map by summing over the height dimension, yielding h_t^s . The second upgrade in SemExp is with the reward. It implements a reward

$$r_t = r_t^e + \gamma * r_t^o, \tag{1}$$

to encourage exploration as in ANS and also moving towards the target object, where r_t^e is the increased coverage area at time t; r_t^o is the distance to the nearest target object; and γ is a hyperparameter to balance r_t^e and r_t^o . Note that ANS uses only r_t^e as its reward.

3.3 Symmetries in the Global Policy

3.3.1 Outline

Although ANS and SemExp are the current state-of-theart for visual exploration and object goal navigation, they employ neural networks consisting only of generic network components, such as convolutional and fully connected layers. This leaves room for improvement in architectural design.

While maintaining the basic framework of ANS that is proven to be effective, we propose to redesign its global policy module, which consists of the actor and critic networks. We consider what conditions the mappings realized by the two networks should satisfy. We pay attention to symmetry of the mappings, more specifically, their equivariance and invariance to translation, rotation, and scaling. Leveraging the fact that the exploration task is a geometric problem, we
 Table 1 Upper three rows: ideal symmetries that should be implemented by the actor and critic networks of the ANS framework.

	Equivariance			Invariance		
	Trans	Rot	Scale	Trans	Rot	Scale
Ideal actor	1	1	x			
Ideal critic				X	1	X
\uparrow at the last conv.	_	1	_			
ANS actor	1	×	×			
ANS critic				X	X	X
\uparrow at the last conv.	1	×	×			

Lower three rows: actual symmetries realized by the original actor and critic networks of ANS. \checkmark indicates the network is equipped with the symmetry. \varkappa indicates the network does not have the symmetry. '--' indicates the symmetry is not specified. Blank cells mean the symmetry is irrelevant. \varkappa in red color indicates that the symmetry of ANS networks differs from the ideal one

derive what symmetry the ideal actor and critic networks should implement, based on which we redesign the two networks.

In other words, we incorporate the inductive bias of the task into the network design. When training a neural network, the goal is to search for a target function, within a set of functions known as the hypothesis space (or solution space). Inductive bias, based on prior knowledge, constrains the search space to a specific hypothesis space, biasing the learning algorithm towards that particular space (Shalev-Shwartz & Ben-David, 2014). By having a precise hypothesis space, the learning algorithm is more likely to find the target function or a good approximation of it. This, in turn, can lead to avoid overfitting and better generalization on unseen data. Convolutional neural networks (CNNs) provide a notable example of models that enhance generalization ability through the use of inductive biases. The convolution operation restricts the hypothesis space to a set of functions that are equivariant to translation, which has been a key factor in the success of CNNs in image processing tasks. In our proposed method, we introduce navigation-specific symmetries (as described in Table 1) as inductive biases to enhance the generalization ability for visual navigation tasks.

3.3.2 Invariance and Equivariance

As mentioned above, we are interested in *equivariance* and *invariance*. Their mathematical definitions are as follows. Given a group \mathcal{G} on a homogeneous space \mathcal{X} , a mapping $\Psi : f(x) \to f'(x)$ that has the property

$$\Psi[\mathbb{T}_g f(x)] = \mathbb{T}_{g'} \Psi[f(x)], \forall f, x \in \mathcal{X}, g \in \mathcal{G},$$
(2)

is said to be *equivariant* to the group \mathcal{G} (or \mathcal{G} -equivariant) if g = g' and to be *invariant* to the group \mathcal{G} (or \mathcal{G} -invariant) if

g' = e, the identity. \mathbb{T}_g is the transformation corresponding to its group action g.

Intuitively, they are intepreted as follows. Suppose a mapping that receives an input and yields an output and also applying a geometric transformation to the input of the mapping. If the mapping is invariant to the transformation, the output for the transformed input will remain the same as the original output. If the mappping is equivariant to the transformation, that will be the same as the result of applying the same transformation to the original output.

Convolutional layer are equivariant to input translation. Thus, shifting an input map results in the same shift of the output map. This applies to the input/output of a stack of any number of convolutional layers. (Rigorously, the output map undergoes the input shift plus downscaling corresponding to the downsampling in those layers.) On the other hand, global average pooling (GAP) layers are invariant to input translation. Thus, CNNs having a GAP layer on top of the stack of convolution layers are invariant to input translation. CNNs and a stack of convolution layers are not equivariant to rotation nor scaling. They are not invariant to them, either.

3.3.3 Ideal Symmetries of Actor and Critic Networks

Now we consider what symmetry the actor and critic networks should have. First, the actor network receives a 2D map encoding the current state of the agent, and outputs a 2D map containing a long-term goal, as explained above and shown in Fig. 3. It is easy to see that this mapping should ideally be equivariant to translation and rotation. It should not be equivariant to scaling.

The critic network receives the same input map and outputs a scalar value, the expected future accumulated reward when following the current policy given the current state. The mapping should be invariant to rotation since the future accumulated reward should be independent of the orientation of the input map. It should not be invariant to translation or scaling, since applying these transformation to the input map should change the reward.

Table 1 (the first two rows) summarizes the requirements for the ideal actor and critic networks.

To design a new critic network, we limit our attention to networks having a stack of convolutional layers at their initial section. This is also the case with the critic network of ANS, which has a stack of fully connected layers right after the convolutional layer stack, and we will redesign this section. Now, for the mapping from the input to the final output to be invariant to rotation as mentioned above, the mapping to the last convolutional layer needs to be equivariant to rotation. This is because otherwise, it will be extremely hard, if not impossible, to achieve the rotation-invariance at the final output. We express this requirement in the '↑ at the last conv.' row in Table 1. The actor and critic networks of the original ANS global policy does not have the same symmetries as the ideal ones. The missing symmetries are indicated by the \times 's in Table 1. Specifically, the actor should be rotation-equivariant but is not; the critic should be rotation-invariant but is not. As discussed above, the mapping from the input to the last convolutional layer of the critic should be rotation-equivariant but is not.

3.4 Rotation-Equivariance of the Actor Network

Thus, we need to newly equip the actor network with rotationequivariance while maintaining its translation-equivariance. To do so, we propose to employ *G*-convolution (Cohen & Welling, 2016) and blur pooling (Zhang, 2019); specifically, we replace the standard convolution and max pooling of the original ANS actor network with the two, respectively.

Considering computational efficiency, we employ p4 *G*convolution. All combinations of translations and rotations by 90 degrees form the group p4, which can be parameterized by

$$g(m, z_1, z_2) = \begin{bmatrix} \cos(m\pi/2) & -\sin(m\pi/2) & z_1 \\ \sin(m\pi/2) & \cos(m\pi/2) & z_2 \\ 0 & 0 & 1 \end{bmatrix},$$
 (3)

where $m \in \{0, 1, 2, 3\}$ and $(z_1, z_2) \in \mathbb{Z}^2$ (Cohen & Welling, 2016).

As shown in Fig. 3, p4 G-convolution can be performed by first rotating filters with angles $m\pi/2$, $m = \{0, 1, 2, 3\}$ to form a filter bank, and then applying it to the input feature map.

As mentioned above, we redesign the convolutional layer stack of the actor network with the same number of layers performing p4 G-convolution. The original ANS actor network employs max pooling for downsampling the feature map. However, max pooling makes the translationequivariance inaccurate. Max pooling ignores the Nyquist sampling thereom, breaking translation equivariance. Blur pooling filters the signal before downsampling to better preserve translation equivariance. Hence, we employ blur pooling and replace max pooling with this.

By revising the convolutional block (i.e., the stack of convolutional and pooling layers) in the original ANS actor network as above, the mapping represented by the block becomes approximately equivariant to rotation and precisely equivariant to translation. We need to maintain the fully connected layers after the convolutoinal block since predicting the agent's goal seems to need them to integrate the global features in a non-simple manner. Fully connected layers are not inherently equipped with the desired equivariance, and we leave it to training; owing to their flexibility, we expect the actor network to gain the desired equivariance at the final output. On the other hand, we want to make the critic network have rotation-invariance. As mentioned earlier, to do so, we want the mapping represented from the input to the output of the convolutional block to be rotation-equivariant. We employ here the design of the original ANS that the critic network share the convolutional block with the actor network. Then, the above redesign of the convolutional block achieves what we want. We further revise the section on top of the convolutional block to realize the desired symmetries for the critic network.

3.5 Rotation-Invariance of the Critic Network: Semi-global Polar Pooling

The critic network should represent a mapping that is rotation-invariant and is not translation-invariant or scaleinvariant, as shown in Table 1. We assume the convolutional block to have rotation-equivariance due to its design, as explained above. To attain the above (in)variance, we then propose a new network component, named the semi-global polar pooling (SGPP) layer.

SGPP is illustrated in Fig. 4. It first converts the input feature map from Cartesian space to polar space. Let $I(x, y) \in \mathbb{R}^{c \times h \times w}$ denote the feature map represented in Cartesian space, where *x*, *y* are the Cartesian coordinates. The feature map $I'(\rho, \phi) \in \mathbb{R}^{c \times h \times w}$ represented in polar space is given by

$$\rho = \sqrt{x^2 + y^2},$$

$$\phi = \arctan \frac{y}{r},$$
(4)

where ρ and ϕ are the coordinates in polar space. We finally apply average-pooling to the features over all ϕ 's as

$$\Phi = \frac{1}{n} \sum_{\phi} I'(\rho, \phi), \tag{5}$$

to obtain a pooled feature $\Phi \in \mathbb{R}^{c \times h}$.

As the convolutional block, whose output is the input to the SGPP layer, is equivariant to rotation, Φ is invariant to



Fig. 4 Illustration of the proposed semi-global polar pooling (SGPP)

rotation since it is pooled over the coordinate ϕ (i.e., the circumferential direction). Φ is further processed by fully connected layers, computing the final output of the critic network. As Φ is already invariant to rotation, these additional layers do not change the invariance. While the convolutional block is also equivariant to translation, the Cartesian-polar conversion invalidates the equivariance; thus Φ is not invariant to translation. It is not invariant to scaling, either. Thus, the critic network having the SGPP layer in between the convolutional block and the fully connected layers attains the desired (in)variance to translation, rotation, and scaling, as illustrated in Table 1.

3.6 Symmetry-aware Exploration and Object Goal Navigation

We apply the above symmetry-aware architecture to ANS (Chaplot et al., 2020a) for exploration and SemExp (Chaplot et al., 2020b) for object goal navigation, respectively. We will refer them as *S-ANS (Symmetry-aware ANS)* and *S-SemExp (Symmetry-aware SemExp)*, respectively, in what follows.

To implement S-ANS, we integrate our method for generating global policy to ANS by replacing the original global policy module with ours. Specifically, the *G*-convolution, blur pooling and proposed SGPP layer are integrated into the global policy module.

Object goal navigation aims at finding designated objects in novel environments. SemExp (Chaplot et al., 2020b) addresses it by incorporating semantic layout to the global policy of ANS, as in Sect. 3.2, establishing state-of-theart. S-SemExp replaces the global policy in SemExp with ours. Specially, we equip the actor of SemExp with rotationequivariance by employing G-convolution and blur pooling, given in Sect. 3.4. The critic is equipped with rotation invariance by inserting our proposed SGPP layer, as described in Sect. 3.5.

4 Experimental Results

We conduct experiments to examine the effectiveness of the proposed approach.

4.1 Experimental Setup

As with previous studies (Singh Chaplot et al., 2020; Chaplot et al., 2020a; Ramakrishnan et al., 2020; Chen et al., 2019; Chaplot et al., 2020b; Gan et al., 2020; Chaplot et al., 2020c), we use the Habitat simulator (Savva et al., 2019) for our experiments. We choose the configuration that depth images are available as the visual input and the actuation and sensory signals include noises for visual exploration. This applies to the training and test times. For object goal navigation, we

follow the setting in Chaplot et al. (2020b), assuming the availability of RGBD images and noise-free actuation and sensory signals.

To evaluate the performance for visual exploration, we use the area coverage (i.e., the area seen during exploration) within a time budget for the primary metric, following Chen et al. (2019); Chaplot et al. (2020a). A greater coverage of the area indicates better exploration performance with a designated time budget. We compare our method with several baselines. We run each method five times and report their average area coverage with the standard deviation.

Similarly, we compare our method with several baseline methods for object goal navigation. To evaluate their performance, we employ the area under the success rate curve (AUC), average completion time, and SPL metrics. The success rate is the ratio between the successful episodes and all the test episodes. AUC quantifies the comprehensive performance throughout the entire time duration by calculating the area under the success rate curve. Average completion time is the time steps spent finding target objects, averaged over all successful episodes. SPL (Anderson et al., 2018) computes success rate weighted by path length, indicating the efficiency of reaching the target goal. A higher AUC, larger SPL and reduced average completion time indicate superior performance in object goal navigation. For the target goals, we use the object categories shared among Gibson, MP3D, and MS-COCO (Lin et al., 2014), following (Chaplot et al., 2020b). They are the categories of 'bed', 'chair', 'couch', 'potted plant', 'tv,' and 'toilet'.

4.2 Datasets

Following previous studies, we employ two datasets, Gibson (Xia et al., 2018) and MP3D (Chang et al., 2017). Both of them contain photorealistic virtual environments created from real-world scenes. Most environments contained in Gibson are office spaces, while those in MP3D are homes. Thus, the environments of Gibson differ from those of MP3D in terms of scenarios, layout, area, and so on. Both the Gibson and MP3D datasets comprise 3D reconstructions of real-world environments. However, Gibson dataset utilizes a distinct set of cameras and primarily focuses on office spaces, whereas MP3D dataset predominantly includes homes and exhibits a larger average scene area and more irregular layout. Figure 1 provides the layouts of several representative examples from Gibson and MP3D. It is seen that there is a gap in various aspects between the two.

The visual exploration task leverages depth images to build occupancy maps, while the object goal navigation task uses RGB and depth images to compute semantic maps. Images are with a spatial size of 640×480 . The visual exploration uses the standard Gibson and MP3D dataset for training and evaluation, including 72/16 train/test scenes in

Gibson, 61/18 train/test scenes in MP3D. Object goal navigation task utilizes Gibson tiny and MP3D datasets for training and evaluation, following Chaplot et al. (2020b). The Gibson tiny dataset contains 25/5 train/test scenes. The MP3D test set contains 8 scenes for evaluation. The validation set in Gibson is used as the test set because the true test set is held-out for the online evaluation server. The reported results are averaged over 1120/1008 episodes on Gibson/MP3D test set for visual exploration and 1000/1600 episodes on Gibson tiny/MP3D test set for object goal navigation.

To evaluate the generalization ability of the methods across different datasets, we train each model on Gibson and test it either on Gibson or on MP3D, following the experiments of Chaplot et al. (2020a) and Chaplot et al. (2020b) for visual exploration and object goal navigation respectively.

4.3 Compared Methods

To evaluate our approach for visual exploration, we experimentally compare S-ANS, which employs our symmetryaware architecture to extend ANS as explained in Sect. 3.6, with FBE (Frontier based Exploration)(Yamauchi, 1997), FBE-RL, a variant of FBE, and ANS. Details of the baseline methods are provided in appendix A. We replace the global policy module in ANS with FBE and FBE-RL and evaluate their performance, respectively. We run each model five times and report their average and standard deviation of the area coverage at time step 1000, aligning with the setting in ANS.

To evaluate our approach for object goal navigation, we compare S-SemExp, which extends SemExp in the same way as ANS/S-ANS as explained in Sect. 3.6, with several baselines. The first is the original SemExp. We also consider FBE and ANS for comparison, searching objects via exploration. The maximum number of timesteps is set to 1000.

4.4 Results of Visual Exploration

4.4.1 Results of Training on Gibson

We first show the results when we train the models on the Gibson training set and test them on either the Gibson or MP3D test sets. Table 2 shows the results. Comparing the methods in the case of training and testing on Gibson, the proposed S-ANS outperforms ANS with the margin of $0.8m^2$ (33.7 vs. $32.9m^2$). The margin becomes more significant when testing the same models on MP3D, i.e., $8.1m^2$ (84.4 vs. $76.3m^2$). This demonstrates the better generalization ability of S-ANS.

Figure 5 shows the representative examples of their exploration paths of ANS and S-ANS on the same four environments of MP3D. The first and second columns of Fig. 5 show typical examples for which S-ANS explores a much larger area than ANS; S-ANS explores twice as large area

models on Gibson and MP3D.				
	Gibson (m^2)	MP3D (<i>m</i> ²)		
FBE	26.5 ± 0.5	69.6 ± 1.9		
FBE-RL	28.2 ± 0.3	63.0 ± 3.2		
ANS	32.9 ± 0.2	76.3 ± 2.8		
S-ANS	33.7 ± 0.2	$\textbf{84.4} \pm \textbf{1.7}$		

Table 2 Exploration performance (in area coverage, m^2) of different models on Gibson and MP3D.

The bold text indicates the best performance achieved among different methods

All the models are trained on Gibson



(b) S-ANS.

Fig. 5 Exploration paths of ANS (top row) and S-ANS (bottom row) for four representative environments from MP3D. The number at the bottom left of each box indicates the area coverage (m^2) . The dissimilarity of the environment from the training data tends to widen the gap between the two methods

as ANS. As with these examples, S-ANS tends to show better exploration performance for environments that are more dissimilar from those of Gibson. When the environments are similar, their performances tend to be close, as shown in the third and fourth columns of Fig. 5. Figure 6 illustrates a comparison between the global target points computed by ANS and S-ANS, using the same input from the Gibson dataset. The input maps in Fig. 6b and d are derived by rotating the maps in Fig. 6a and c by 15 degrees counterclockwise. Upon observing Fig. 6a and c, it is evident that the global target point computed by S-ANS is more reasonable, providing an easily reachable target point and greater opportunities to explore unknown area. Figure 6a and b exhibit approximate rotation equivariance, as the global target point computed by S-ANS rotates accordingly with the input map. However, this is not observed for the target points computed by ANS, as depicted in Fig. 6c and d. These observations further verify the improved generalization ability of S-ANS.

It is seen from Table 2 that FBE and FBE-RL show inferior performance. It is noteworthy that FBE and FBE-RL behave differently for different combinations of the train and test data. While FBE-RL performs better on Gibson by



(a) Global target point computed by S-ANS for a given map.



(b) Global target point computed by S-ANS for a map rotated 15 degrees from the original map in (a).



(c) Global target point computed by ANS for the map shown in (a).



(d) Global target point computed by ANS for a map rotated 15 degrees from the original map in (a).

Fig. 6 Comparison of global target points computed by S-ANS and ANS for original inputs **a** and **c** and their corresponding rotated inputs **b** and **d**. The yellow point represents the location of the robot, while the red point indicates the target point (Color figure online)

 $1.7m^2$ (28.2 vs $26.5m^2$), it performs worse on MP3D by $-6.6m^2$ (63.0 vs $69.6m^2$). We can say that learning better policies improves exploration performance when there is only a little gap between the train and test data. How-

ever, this improvement comes at a cost as it introduces the risk of overfitting, consequently resulting in reduced exploration capability when there exists a significant gap between the train and test data. This demonstrates the difficulty with learning-based approach to the exploration task.

4.4.2 Results of Training on MP3D

We next show the results when we train the models on MP3D and test them on Gibson and MP3D. We evaluate two models, the original ANS and the proposed S-ANS. Figure 7 and 8 show the results including those trained on Gibson. Method X trained on Gibson and MP3D is denoted by X-Gibson and X-MP3D, respectively. We can make the following observations. First, it is seen from Fig. 8 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. 8 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 between the two datasets by equipping the network with



Fig.7 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 Gibson. The method X trained on Gibson and MP3D is denoted by X-Gibson and X-MP3D, respectively. The dashed box shows the data points mapped linearly to the normalized [0,1] range for better visibility



Fig.8 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 MP3D. The method X trained on Gibson and MP3D is denoted by X-Gibson and X-MP3D, respectively. The dashed box shows the data points mapped linearly to the normalized [0,1] range for better visibility

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. 7. 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 = $33.1m^2$.

4.4.3 Error Cases

Figure 9 shows example error cases on the MP3D dataset; the target points predicted by the model, which was trained on the Gibson dataset, exhibit less reasonable outcomes in some cases. For instance, in Fig. 9(a), the predicted target point is situated in the southeastern direction, whereas a more reasonable target point would be positioned along the corridor. Similarly, in Fig. 9(b), a preferable target point would be in the north direction, but the model predicts a location in the southeastern direction instead. These discrepancies can be attributed to the substantial deviation of these cases from the training samples in the Gibson dataset, particularly concerning their irregular shapes. It is noteworthy that the Gibson dataset predominantly consists of environment maps with regular shapes, which could have influenced the model's predictions in these atypical scenarios.



Fig. 9 Error cases in MP3D dataset produced by the model trained on Gibson dataset. The yellow point represents the location of the robot, while the red point indicates the target point (Color figure online)

4.5 Results of Object Goal Navigation

We then show the results for object goal navigation. Figure 10(a) shows the mean and standard deviation of success rate from time step 0 to 1000 when the methods are trained on the Gibson training set and tested on the Gibson test set. Table 3 provides their corresponding *AUC*, SPL and average completion time at the time step of 1000. As shown in the table, S-SemExp achieves the highest *AUC* = 530.0, SPL = 0.29, and least average completion time of 146.6 time steps on the Gibson dataset. Namely, S-SemExp can navigate to targets more successful and quickly than others, demonstrating the effectiveness of our method.

ANS does not access semantic layout information, resulting in inferior performance compared with SemExp. FBE falls behind other methods with the longest time steps = 173.4, smallest AUC = 443.0 and SPL = 0.25, as expected. When they are given more time, all but FBE tend to reach a similar success rate, as shown in Fig. 10.

We can draw the same conclusion as the Gibson test set, for the MP3D test set, which simulates testing on OOD data. The results are shown in Fig. 10(b) and Table 3. S-SemExp demonstrates superior performance in terms of *AUC*, SPL, and average completion time.

To summarize, S-SemExp obtains better object goal navigation performance than the benchmark methods, establishing new state-of-the-art. It enhances state-of-the-art method SemExp by 6.0(1.1%)/6.0(1.7%) in terms of AUC and saves



Fig. 10 Success rate curve of FBE, ANS, SemExp and S-SemExp tested on **a** Gibson and **b** MP3D. All the models are trained on the Gibson training set

3.5 (2.3%)/15.1 (6.2%) time steps on average to reach goal objects on Gibson/MP3D test set.

4.6 Ablation Study

As explained earlier, our method imposes translation- and rotation-equivariance on the actor network and (approximate) rotation-invariance on the critic network. The former is Table 3Performance of FBE,ANS, SemExp and S-SemExpon the Gibson and MP3D testset at time step 1000 in terms ofAUC, SPL and averagecompletion time.

Method	Gibson	Gibson			MP3D		
	AUC	SPL	Comp. Time	AUC	SPL	Comp. Time	
FBE	443.0	0.25	173.4	282.0	0.18	267.5	
ANS	516.0	0.28	155.5	351.0	0.18	238.7	
SemExp	524.0	0.29	150.1	354.0	0.19	242.3	
S-SemExp	530.0 ↑	0.29	146.6 ↑	360.0 ↑	0.19	227.2 ↑	

The bold text indicates the best performance achieved among different methods

The upward arrow \uparrow indicates an improvement in performance compared to the baseline methods

implemented by the *G*-convolution and the blur pooling. The latter is implemented by SGPP in addition to *G*-convolution and the blur pooling. To examine the effectiveness of each component, we create two variants of the proposed network, named E-ANS and G-ANS, and compare their performance on the visual exploration task.

E-ANS is a model created by removing the SGPP layer from the critic network of S-ANS. Without SGPP, it is not (even approximately) invariant to rotation, translation, or scaling. Its actor network maintains the ideal property of equivariance. G-ANS is an intermediate model between E-ANS and S-ANS; it is created by replacing the SGPP layer with a global average pooling (GAP) layer in the critic network of S-ANS. The added GAP layer makes the critic network invariant to both rotation and translation; recall that the convolutional block before the GAP layer is equivariant to rotation and translation due to the employment of the G-convolution and the blur-pooling. The added translation invariance is superfluous compared with the ideal invariance of the critic network shown in Table 1. In summary, E-ANS introduces rotation equivariance on top of ANS, while S-ANS integrates rotation invariance on top of E-ANS. Additionally, G-ANS further introduces other invariance to S-ANS by replacing SGPP with GAP. In short, considering the excesses and deficiencies of the implemented invariance/equivariance, the expected performance will be S-ANS > E-ANS, E-ANS > G-ANS.

We train E-ANS and G-ANS on Gibson and test them on Gibson and MP3D in the same way as ANS and S-ANS. Figure 11(a) and (b) show the explored areas by the four models on the test splits of Gibson and MP3D, respectively. Each solid curve and shadowed area indicates the mean and standard deviation over five runs, respectively, of the area coverage at a different time step.

It is seen from the results that the four models are ranked in the performance as S-ANS > E-ANS > G-ANS > ANS. This matches well with our expectation mentioned above. The results further tell us that the excessive translation-invariance of the critic network in G-ANS does more harm than deficient rot-invariance of that of E-ANS. Overall, these results validate the effectiveness of our approach.

4.7 Qualitative & Quantitative Analysis for Invariant Representation

We experimentally evaluate rotation invariance of the critic of S-ANS over the visual exploration task. Specifically, we compute the standard deviation 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, ..., 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 $S^* = \{s_i^k | s_i^k = r^k \cdot s_i, i \in \{1, 2, ..., Q\}, k \in \{0, 1, ..., K - 1\}\}$ 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

$$std = \frac{1}{Q} \sum_{i=1}^{Q} \sqrt{\sum_{k=0}^{K} (y_i^k - \bar{y_i})^2},$$
(6)

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.

Figure 12 shows *std*'s of the critic of ANS and that of S-ANS (both trained on Gibson) when we set K = 24. In this case, Each rotation angle corresponds to a 15-degree increment, which we consider to be fine enough to demonstrate the rotation invariance property. 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 *p*4 *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



Fig. 11 Expolaration performance (in area coverage, m^2) of ANS, S-ANS, G-ANS, and E-ANS tested on **a** Gibson and **b** MP3D. All the models are trained on the Gibson dataset

$$sim(\xi(s^{\alpha}), \xi(s^{\beta})) = \frac{1}{Q-1} \sum_{i=0}^{Q-1} \frac{\xi(s_i^{\alpha}) \cdot \xi(s_i^{\beta})}{\|\xi(s_i^{\alpha})\| \cdot \|\xi(s_i^{\beta})\|}, \quad (7)$$

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



Fig. 12 Rotation invariance (i.e., *std*) of the critics of ANS and S-ANS trained on Gibson seen over the evaluation episodes of Gibson and MP3D

Figure 13 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.

5 Limitation

While our approach offers valuable benefits, one notable limitation with it is the increased memory consumption when using *G*-convolution compared to standard convolution. Specifically, a CNN consisting of p4 *G*-convolution requires four times the memory space compared to a network utilizing standard convolution. This increased memory requirement can be a concern, particularly when working with large backbone networks or resource-constrained environments. Researchers should carefully consider the memory limitations of their computational resources when applying our method.

6 Conclusion

Existing learning-based methods for visual navigation tend to struggle with generalization to out-of-distribution environments, i.e., statistically different environments from those used for training. To cope with this, we propose to use the



Fig. 13 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

inductive biases available for the task that previous methods do not use effectively. Employing the framework of active neural SLAM (ANS), we have shown that the actor and the critic should satisfy specific symmetries with their mappings. We then propose a design of neural networks that inherently possesses the ideal symmetries. Specifically, we propose to use G-convolution instead of the standard convolution layer. We also propose the semi-global polar pooling (SGPP) layer, which makes the network invariant to rotation and not invariant to translation when using it at the final section of the critic network. Experimental results show that our method, integrated into the current state-of-the-art, ANS for visual exploration and SemExp for object goal navigation, improves their performance, particularly in terms of generalization ability, having established the new state-of-the-art with the standard datasets, Gibson and MP3D.

Acknowledgements This work was partly supported by JSPS KAK-ENHI Grant Number 23H00482, 20H05952 and 19H01110.

Data Availability The datasets that support the findings of this study are all publicly available for the research purpose.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecomm ons.org/licenses/by/4.0/.

Appendix A Implementation Details of Baseline Methods

This section gives implementation details of baseline methods FBE, FBE-RL and ANS. FBE is a heuristic method that iteratively selects a point at the frontiers (boundaries between explored free region and unexplored region) using various strategies. We follow the strategy implemented in Ramakrishnan et al. (2021), selecting a random point at the longest boundary. We replace the global policy module in ANS with FBE and evaluate its performance. FBE-RL, a variant of FBE, which replaces the global policy module in ANS with a RLbased 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

$$m_i'' = \frac{e^{m_i}}{\sum_j e^{m_j}},\tag{A1}$$

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

For ANS, we use the code 1 and the settings given in Ramakrishnan et al. (2020).

References

- Anderson, P., Chang, A., Chaplot, D. S., et al. (2018). On evaluation of embodied navigation agents. arXiv preprint arXiv:1807.06757
- Beeching, E., Dibangoye, J., Simonin, O., et al. (2020). Egomap: Projective mapping and structured egocentric memory for deep rl. In: Joint European conference on machine learning and knowledge discovery in databases, Springer, pp 525–540
- Bonin-Font, F., Ortiz, A., & Oliver, G. (2008). Visual navigation for mobile robots: A survey. *Journal of intelligent and robotic systems*, 53(3), 263–296.
- Cadena, C., Carlone, L., Carrillo, H., et al. (2016). Past, present, and future of simultaneous localization and mapping: Toward

¹ github.com/facebookresearch/OccupancyAnticipation.

the robust-perception age. *IEEE Transactions on robotics*, *32*(6), 1309–1332.

- Calimeri, F., Marzullo, A., Stamile, C., et al. (2017). Biomedical data augmentation using generative adversarial neural networks. In: *International conference on artificial neural networks*, Springer, pp 626–634
- Chang, A., Dai, A., Funkhouser, T., et al. (2017). Matterport3d: Learning from rgb-d data in indoor environments. In: *International conference on 3D vision (3DV)*.
- Chaplot, D. S., Gandhi, D., Gupta, S., et al. (2020a). Learning to explore using active neural slam. In: *International conference* on learning representations, URI https://openreview.net/forum? id=HklXn1BKDH
- Chaplot, D. S., Gandhi, D. P., Gupta, A., et al. (2020). Object goal navigation using goal-oriented semantic exploration. Advances in Neural Information Processing Systems, 33, 4247.
- Chaplot, D. S., Jiang, H., Gupta, S., et al. (2020c). Semantic curiosity for active visual learning. In: *European conference on computer* vision, Springer, pp 309–326.
- Chen, C., Majumder, S., Al-Halah, Z., et al. (2021). Learning to set waypoints for audio-visual navigation. In: *International conference* on learning representations, URL https://openreview.net/forum? id=cR91FAodFMe
- Chen, T., Gupta, S., & Gupta, A. (2019). Learning exploration policies for navigation. In: *International conference on learning representations*, URL https://openreview.net/forum?id=SyMWn05F7
- Cheng, G., Zhou, P., & Han, J. (2016). Learning rotation-invariant convolutional neural networks for object detection in vhr optical remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 54(12), 7405–7415.
- Choi, Y., & Oh, S. (2021). Image-goal navigation via keypoint-based reinforcement learning. In: 2021 18th international conference on ubiquitous robots (UR), IEEE, pp 18–21.
- Cohen, T., & Welling, M. (2016). Group equivariant convolutional networks. In: Balcan, M. F., Weinberger, K. Q. (eds) Proceedings of The 33rd international conference on machine learning, proceedings of machine learning research, vol 48. PMLR, pp 2990–2999, URL https://proceedings.mlr.press/v48/cohenc16.html.
- Dai, A., Papatheodorou, S., Funk, N., et al. (2020). Fast frontierbased information-driven autonomous exploration with an mav. In: 2020 IEEE International conference on robotics and automation (ICRA), pp 9570–9576, https://doi.org/10.1109/ICRA40945. 2020.9196707.
- Dey, N., Chen, A., & Ghafurian, S. (2020). Group equivariant generative adversarial networks. CoRR arXiv:2005.01683.
- Dieleman, S., Fauw, J. D., & Kavukcuoglu, K. (2016). Exploiting Cyclic Symmetry in Convolutional Neural Networks. In: *Proceedings of* the 33rd international conference on machine learning. JMLR, pp 1889–1898.
- Du, H., Yu, X., & Zheng, L. (2020). Learning object relation graph and tentative policy for visual navigation. In: *European conference on computer vision*, Springer, pp 19–34
- Gan, C., Zhang, Y., Wu, J., et al. (2020). Look, listen, and act: Towards audio-visual embodied navigation. In: 2020 IEEE International conference on robotics and automation (ICRA), IEEE, pp 9701– 9707.
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.
- Gupta, S., Davidson, J., Levine, S., et al. (2017). Cognitive mapping and planning for visual navigation. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp 2616– 2625.
- He, K., Gkioxari, G., Dollár, P., et al. (2017). Mask r-cnn. In: Proceedings of the IEEE international conference on computer vision, pp 2961–2969.

- Jayaraman, D., & Grauman, K. (2018). Learning to look around: Intelligently exploring unseen environments for unknown tasks. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp 1238–1247.
- Laskin, M., Lee, K., Stooke, A., et al. (2020). Reinforcement learning with augmented data. Advances in Neural Information Processing Systems, 33, 19884–19895.
- Lin, T. Y., Maire, M., Belongie, S., et al. (2014). Microsoft coco: Common objects in context. In: *European conference on computer* vision, Springer, pp 740–755.
- Lindeberg, T. (2021). Scale-covariant and scale-invariant gaussian derivative networks. In: *International conference on scale space* and variational methods in computer vision, Springer, pp 3–14.
- Liu, S., & Okatani, T. (2022). Symmetry-aware neural architecture for embodied visual exploration. In: *Proceedings of the IEEE/CVF* conference on computer vision and pattern recognition, pp 17,242– 17,251.
- Liu, S., Ozay, M., Xu, H., et al. (2019). A generative model of underwater images for active landmark detection and docking. In: 2019 IEEE/RSJ International conference on intelligent robots and systems (IROS), pp 8034–8039, https://doi.org/10.1109/IROS40897. 2019.8968146.
- Lv, Y., Xie, N., Shi, Y., et al. (2020). Improving target-driven visual navigation with attention on 3d spatial relationships. CoRR arXiv:2005.02153.
- Madani, A., Moradi, M., Karargyris, A., et al. (2018). Chest x-ray generation and data augmentation for cardiovascular abnormality classification. In: *Medical imaging 2018: Image processing, international society for optics and photonics*, p 105741M.
- Mayo, B., Hazan, T., & Tal, A. (2021). Visual navigation with spatial attention. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 16,898–16,907.
- Mezghani, L., Sukhbaatar, S., Szlam, A., et al. (2020). Learning to visually navigate in photorealistic environments without any supervision. CoRR arXiv:2004.04954.
- Mezghani, L., Sukhbaatar, S., Lavril, T., et al. (2021). Memoryaugmented reinforcement learning for image-goal navigation. CoRR arXiv:2101.05181.
- Mirowski, P., Pascanu, R., Viola, F., et al. (2016). Learning to navigate in complex environments. arXiv preprint arXiv:1611.03673
- Mishkin, D., Dosovitskiy, A., & Koltun, V. (2019). Benchmarking classic and learned navigation in complex 3d environments. arXiv preprint arXiv:1901.10915
- Mnih, V., Badia, A. P., Mirza, M., et al. (2016). Asynchronous methods for deep reinforcement learning. In: *International conference on machine learning*, *PMLR*, pp 1928–1937.
- Müller P, Golkov V, Tomassini V, et al (2021) Rotation-equivariant deep learning for diffusion MRI. CoRR arXiv:2102.06942.
- Nachum, O., Gu, S. S., Lee, H., et al. (2018). Data-efficient hierarchical reinforcement learning. Advances in Neural Information Processing Systems 31. https://dl.acm.org/doi/abs/10.5555/3327144. 3327250
- Nagarajan, T., & Grauman, K. (2020). Learning affordance landscapes for interaction exploration in 3d environments. Advances in Neural Information Processing Systems, 33, 2005.
- Pal, A., Qiu, Y., & Christensen, H. (2021). Learning hierarchical relationships for object-goal navigation. In: *Conference on robot learning*, *PMLR*, pp 517–528.
- Pathak, D., Agrawal, P., Efros, A. A., et al. (2017). Curiosity-driven exploration by self-supervised prediction. In: *International conference on machine learning*, *PMLR*, pp 2778–2787.
- Qi, W., Mullapudi, R. T., Gupta, S., et al. (2020). Learning to move with affordance maps. In: *International conference on learning representations*, URL https://openreview.net/forum?id=BJgMFxrYPB

- Raileanu, R., Goldstein, M., Yarats, D., et al. (2021). Automatic data augmentation for generalization in deep reinforcement learning. arXiv:2006.12862
- Ramakrishnan, S. K., Al-Halah, Z., & Grauman, K. (2020). Occupancy anticipation for efficient exploration and navigation. In: *European conference on computer vision*, Springer, pp 400–418.
- Ramakrishnan, S. K., Jayaraman, D., & Grauman, K. (2021). An exploration of embodied visual exploration. *International Journal of Computer Vision*, 129(5), 1616–1649.
- Savinov, N., Dosovitskiy, A., & Koltun, V. (2018). Semi-parametric topological memory for navigation. In: *International conference* on learning representations.
- Savva, M., Kadian, A., Maksymets, O., et al. (2019). Habitat: A Platform for Embodied AI Research. In: Proceedings of the IEEE/CVF international conference on computer vision (ICCV).
- Seifi, S., & Tuytelaars, T. (2019). Where to look next: Unsupervised active visual exploration on 360° input. CoRR arXiv:1909.10304.
- Shalev-Shwartz, S., & Ben-David, S. (2014). Understanding machine learning: From theory to algorithms. Cambridge: Cambridge University Press.
- Shen, W. B., Xu, D., Zhu, Y., et al. (2019). Situational fusion of visual representation for visual navigation. In: *Proceedings of the IEEE/CVF international conference on computer vision*, pp 2881– 2890.
- Singh Chaplot, D., Salakhutdinov, R., Gupta, A., et al. (2020). Neural topological slam for visual navigation. In: 2020 IEEE/CVF conference on computer vision and pattern recognition (CVPR), pp 12,872–12,881, https://doi.org/10.1109/CVPR42600.2020.01289
- Sosnovik, I., Szmaja, M., & Smeulders, A. (2020). Scale-equivariant steerable networks. In: *International conference on learning representations*, URL https://openreview.net/forum?id=HJgpugrKPS.
- Sosnovik, I., Moskalev, A., & Smeulders, A. W. (2021). Scale equivariance improves siamese tracking. In: *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pp 2765–2774.
- Szot, A., Clegg, A., Undersander, E., et al. (2021). Habitat 2.0: Training home assistants to rearrange their habitat. Advances in Neural Information Processing Systems, 34, 251–266.
- Thiede, E. H., Hy, T., & Kondor, R. (2020). The general theory of permutation equivarant neural networks and higher order graph variational encoders. CoRR arXiv:2004.03990.
- Visser, A., Xingrui-Ji, van Ittersum M, et al. (2008). Beyond frontier exploration. In U. Visser, F. Ribeiro, T. Ohashi, et al. (Eds.), *RoboCup 2007: Robot Soccer World Cup XI* (pp. 113–123). Berlin: Springer.
- Walters, R., Li, J., & Yu, R. (2021). Trajectory prediction using equivariant continuous convolution. In: *International conference* on learning representations, URL https://openreview.net/forum? id=J8_GttYLFgr
- Wani, S., Patel, S., Jain, U., et al. (2020). Multion: Benchmarking semantic map memory using multi-object navigation. Advances in Neural Information Processing Systems, 33, 9700–9712.

- Worrall, D. E., & Welling, M. (2019). Deep scale-spaces: Equivariance over scale. In: Advances in Neural Information Processing Systems, pp 7364–7376.
- Wu, Y., Wu, Y., Tamar, A., et al. (2019). Bayesian relational memory for semantic visual navigation. In: 2019 IEEE/CVF International conference on computer vision (ICCV), pp 2769–2779, https://doi. org/10.1109/ICCV.2019.00286
- Xia, F., R. Zamir, A., He, Z. Y., et al. (2018). Gibson env: real-world perception for embodied agents. In: *Computer vision and pattern* recognition (CVPR), 2018 IEEE conference on, IEEE.
- Yamauchi, B. (1997). A frontier-based approach for autonomous exploration. In: *Proceedings 1997 IEEE international sympo*sium on computational intelligence in robotics and automation CIRA'97. 'Towards New Computational Principles for Robotics and Automation', pp 146–151, https://doi.org/10.1109/CIRA. 1997.613851
- Yarats, D., Kostrikov, I., & Fergus, R. (2021). Image augmentation is all you need: Regularizing deep reinforcement learning from pixels. In: *International conference on learning representations*, URL https://openreview.net/forum?id=GY6-6sTvGaf
- Ye, J., Batra, D., Das, A., et al. (2021a). Auxiliary tasks and exploration enable objectgoal navigation. In: *Proceedings of the IEEE/CVF international conference on computer vision*, pp 16,117–16,126.
- Ye, J., Batra, D., Wijmans, E., et al. (2021b). Auxiliary tasks speed up learning point goal navigation. In: Kober, J., Ramos, F., & Tomlin, C. (eds) *Proceedings of the 2020 conference on robot learning*, proceedings of machine learning research, vol 155. PMLR, pp 498–516.
- Yu, C., Yang, X., Gao, J., et al. (2021). Learning efficient multi-agent cooperative visual exploration. In: *Deep RL Workshop NeurIPS* 2021, URL https://openreview.net/forum?id=-4Yz4vU4uN5
- Zhang, R. (2019). Making convolutional networks shift-invariant again. In: International conference on machine learning, PMLR, pp 7324–7334.
- Zhang, S., Song, X., Bai, Y., et al. (2021). Hierarchical object-to-zone graph for object navigation. In: *Proceedings of the IEEE/CVF international conference on computer vision*, pp 15,130–15,140.
- Zhu, Y., Mottaghi, R., Kolve, E., et al. (2017). Target-driven visual navigation in indoor scenes using deep reinforcement learning. In: 2017 IEEE International conference on robotics and automation (ICRA), pp 3357–3364, https://doi.org/10.1109/ICRA.2017. 7989381

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.