

# Unsupervised Domain Adaptation for Visual Navigation

**Abstract.** Advances in visual navigation methods have led to intelligent embodied navigation agents capable of learning meaningful representations from raw RGB images and perform a wide variety of tasks involving structural and semantic reasoning. However, most learning-based navigation policies are trained and tested in simulation environments. In order for these policies to be practically useful, they need to be transferred to the real-world. In this paper, we propose an unsupervised domain adaptation method for visual navigation. Our method translates the images in the target domain to the source domain such that the translation is consistent with the representations learned by the navigation policy. The proposed method outperforms several baselines across two different navigation tasks in simulation. We further show that our method can be used to transfer the navigation policies learned in simulation to the real world.

## 1 Introduction

In the past few years, a lot of progress has been made in learning to navigate from first-person RGB images. Reinforcement learning have been applied to train navigation policies to navigate to goals according to coordinates [17, 5, 42], images [51], object labels [17, 47], room labels [43, 44] and language instructions [19, 7, 3, 14, 8, 41]. However, such navigation policies are predominantly trained and tested in simulation environments. Our goal is to have such navigation capabilities in the real-world. While some progress has been made towards moving from game-like simulation environments to more realistic simulation environments based on reconstructions [46, 4, 36] or 3D modeling [25], there is still a significant visual domain gap between simulation environments and real-world.

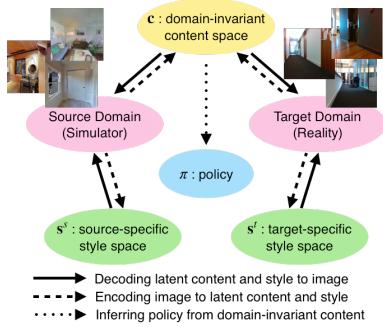
Training the above navigation policies in the real-world has not been possible as current reinforcement learning methods typically require tens of millions of samples for training. Even if we parallelize the training across multiple robots, it will still require multiple weeks on training with constant human supervision due to safety concerns and battery limitations. This makes real-world training practically infeasible and leaves us with the other option of transferring models trained in simulation to the real-world, which highlights the importance of domain adaptation methods.

Among domain adaptation techniques, unsupervised methods are most favourable because it is extremely expensive to collect parallel data for the purpose of visual navigation: It essentially requires reconstructing real-world scenes in the simulator separately for all possible scenarios one might deploy the navigation

045 model in such as different lightning conditions, time of day, indoor vs outdoor,  
 046 weather conditions and so on. Undoubtedly, reconstructing real-world scenes  
 047 from aligned simulation-reality scene pairs is a tedious job requiring specialized  
 048 cameras and significant human effort. Unsupervised learning method has the po-  
 049 tential to overcome this difficulty since it considers only a few real-world images  
 050 taken by regular cameras.

051 One possible solution involves using unsupervised image translation tech-  
 052 niques to translate visual perception from simulation to reality and adapt the  
 053 navigation policy learned in simulation to the real-world. Although there already  
 054 exists a rich amount of prior works in unsupervised image translation techniques  
 055 that transfers images from one domain to another [49, 28, 21], prior techniques  
 056 are not well suited for navigation since the image translations are agnostic of  
 057 the navigation policy and instead focus on photo-realisticity and clarity.

058 In this paper, we propose a unique un-  
 059 supervised domain adaptation method for  
 060 transferring navigation policies from simu-  
 061 lation to the real-world, by unsupervised  
 062 image translation subject to the constraint  
 063 that the image translation respects agent’s  
 064 policy. In order to learn policy-based image  
 065 translation (PBIT) in an unsupervised fash-  
 066 ion, we devise a disentanglement of content  
 067 and style in images such that the represen-  
 068 tations learnt by the navigation policy are  
 069 consistent for images with the same content  
 070 with different styles. See Figure 1 for the  
 071 illustration of PBIT. Our experiments show  
 072 that the proposed method outperforms the  
 073 baselines in transferring navigation policies for different tasks between two sim-  
 074 ulation domains and from simulation to the real-world.



069 Fig. 1: **PBIT**. The proposed policy-  
 070 based image translation for unsuper-  
 071 vised visual navigation adaptation.

## 075 2 Related Work

076 Simulation to Reality (Sim2Real) visual navigation requires the adaptation from  
 077 simulation to reality for both visual perception and agent’s policy. Among its  
 078 wide range of relevant literature, we focus on discussing related work on *visual*  
 079 *navigation* and *visual domain adaptation*.

080 **Visual Navigation.** Prior work on learning-based visual navigation can  
 081 broadly be categorized into two classes based on whether the location of the  
 082 goal is known or unknown. Navigation scenarios where the location of the goal  
 083 is known includes the most common *pointgoal* task where the coordinate to the  
 084 goal is given [17, 31]. Another example of a task in this category is vision and  
 085 language navigation [3] where the path to the goal is described in natural lan-  
 086 guage. Navigation scenarios where the location of the goal is not known include  
 087 a wide variety of tasks. These include navigating to a fixed set of objects [26],

12, 45, 29, 17], navigating to an object specified by language [19, 7] or by an image [51], and navigating to a set of objects in order to answer a question [11, 16]. Tasks in this category essentially involve efficiently and exhaustively exploring the environment to search the desired object. Some recent works explicitly tackle the problem of exploration by training end-to-end RL policies maximizing the explored area [9, 13, 5]. In this paper, we tackle one task in each category, PointGoal and Exploration.

Most of the above works train navigation policies using reinforcement or imitation learning and test in simulation and test on different scenes in the same domain in the simulator. Some prior works which tackle sim2real transfer for navigation policies directly transfer the policy trained in simulation to the real-world without any domain adaptation technique [17, 5]. We show that the proposed domain adaptation method can lead to large improvements over direct policy transfer.

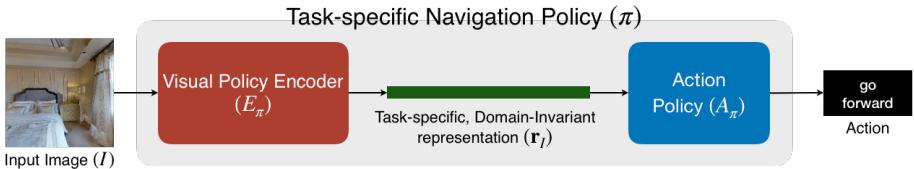
**Visual Domain Adaptation.** Simulation and reality can be viewed as two distinct visual domains, and adapting their visual perceptions can be regarded as an image-to-image translation task. Thanks to the success of Generative Adversarial Networks (GANs) [15] for matching cross-domain distribution, we are able to adapt an image across domains without changing its context. For example, pix2pix [22] changes only the style of an image (e.g., photograph  $\rightarrow$  portrait) while preserving its context (e.g., the same face of a person). We note that, for Sim2Real navigation, some amount of context should be preserved across domains, such as the barriers and the walls, to prevent collisions of our agent.

If we have access to the paired cross-domain images, then pix2pix [22] and BicycleGAN [50] serve as good candidates to model the context-preserving adaptation. However, the paired data between simulation and reality is notoriously hard to collect [38] or even do not exist (e.g., we cannot always build simulators for new environments). To tackle this challenge, numerous visual domain adaptation approaches [37, 35, 49, 23, 48, 28] have been proposed to relax the constraint of requiring paired data during training time. Nevertheless, the above methods still assume one-to-one correspondence across domains. As an example, these models can only generate the same target-domain image given a source-domain image. We argue that it is more realistic to assume many-to-many mappings between simulation and reality.

To achieve both multimodal mappings and training without paired data, MUNIT [21] and DRIT [27] propose to disentangle the context and style of an image. Precisely, they assume the context is shared across domains and the styles are specific to each domain. Note that these models focus on realistic image generation, and hence it remains unclear on how image translation benefits cross-domain visual navigation. To further bridge the gap between navigation and image translation, our key idea is to ensure the agent’s navigation policy be consistent under domain translation. As a consequence, we propose to enforce constraints such that the agent’s policy is only inferred from the shared context across simulation and reality.

### 3 An Unsupervised Domain Adaptation Method for Visual Navigation

Denote the source domain as  $(\mathcal{S}^s, \mathcal{A}, P^s)$  and the target domain as  $(\mathcal{S}^t, \mathcal{A}, P^t)$ .  $\mathcal{S}$  is the state space,  $\mathcal{A}$  is the set of actions, and  $P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow \mathbb{R}$  is the transition probability distribution. Note that we assume action spaces  $\mathcal{A}$  are shared across domains. Let  $\pi^s : \mathcal{S}^s \rightarrow \mathcal{A}$  be a navigation policy in the source domain  $s$  (the navigation policy is given). For our task setup, we have access to some target-domain images  $I^t \in \mathcal{S}^t$  during training, but we cannot perform target-domain policy ( $\pi^t$ ) training. Our objective is to learn a many-to-many mapping  $F : \mathcal{S}^t \rightarrow \mathcal{S}^s$  such that the navigation policy under the source-to-target mapping  $F$ ,  $\pi^t(I^t) = \pi^s(F(I^t))$ , is effective in the target domain  $t$ . Under Sim2Real setting, the source domain refers to simulator and the target domain refers to reality. Unless specified, we abbreviate  $\pi^s$  as  $\pi$  for the rest of the paper. The remaining part of this section shall describe our proposed ‘Policy-Based Image Translation’ (PBIT).



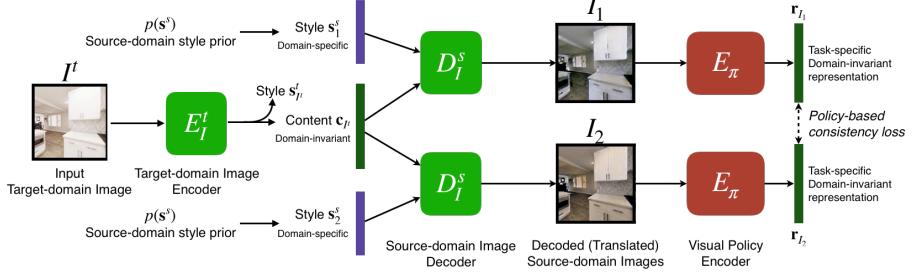
**Fig. 2: Policy Decomposition.** The Task-specific Navigation Policy ( $\pi$ ) can be sequentially decomposed into a Visual Policy Encoder ( $E_\pi$ ) and a Action Policy ( $A_\pi$ ) such that  $E_\pi$  extracts all task-specific features ( $r_I$ ) and throws away all domain-specific features from the input image ( $I$ ) and  $A_\pi$  learns an action distribution function over the task-specific features.

#### 3.1 Policy Decomposition

As our objective is to transfer the task-specific navigation policy across domains, we assume that the task itself is domain-invariant. As a consequence, given a policy  $\pi$  (for the navigation task in the source domain), we assume that some intermediate task-specific representation inferred by the policy is invariant from the source to the target domain. For example, a simple obstacle avoidance navigation policy would extract task-specific and domain-invariant features such as distance to obstacles at various angles and then learn a policy over these features. Let  $\pi$  be sequentially decomposed into a Visual Policy Encoder ( $E_\pi$ ) and a Action Policy ( $A_\pi$ ).  $E_\pi$  extracts all task-specific features ( $r_I$  with  $I$  indicating the input image) and throws away all domain-specific features in the input image.  $A_\pi$  learns an action distribution function over the task-specific features. We illustrate the policy decomposition in Figure 2.

#### 3.2 Policy-based Consistency Loss

Recall that our objective is to learn an image translation model  $F : \mathcal{S}^t \rightarrow \mathcal{S}^s$ , such that  $\pi^t(I^t) = \pi^s(F(I^t))$ . By policy decomposition, given a target-domain



**Fig. 3: Policy-based Consistency Loss.** Since the task is domain-invariant, task-specific representations obtained from different domain-specific styles but the same domain-invariant content should be similar.

image, different translated images to the source domain would have similar task-specific features. Precisely, if  $I_1^s$  and  $I_2^s$  are the translated images to the source domain from the same target-domain image  $I^t$ . In other words, if  $I_1^s, I_2^s \sim F(I^t)$ , then  $E_\pi(I_1^s) \approx E_\pi(I_2^s)$ .

To achieve the above policy consistency in an unsupervised fashion, we take inspiration from style and content-based unsupervised methods designed for image translation [21]. We assume that each image can be decomposed into a domain-invariant content representation ( $\mathbf{c}$ ) and a domain-specific style representation ( $\mathbf{s}$ ). Let  $E_I^s$  be an Image Encoder for domain  $s$  which encodes an image ( $I$ ) to domain-invariant content ( $\mathbf{c}$ ) and domain-specific style ( $\mathbf{s}^s$ ):  $E_I^s(I) = (\mathbf{c}_I, \mathbf{s}_I^s)$ . On the contrary, let  $D_I^s$  be an Image Decoder which is the inverse of the Image Encoder:  $D_I^s(\mathbf{c}_I, \mathbf{s}_I^s) = I$ .

Since we assume the navigation task is domain-invariant, all the the task-specific features are a subset of content representation  $\mathbf{r}_I \in \mathbf{c}_I$ . Therefore, images generated from different styles but same content should lead to the same task-specific features as shown in Figure 3. We operationalize this idea using the following policy-based consistency loss:

$$\mathcal{L}_{pol} = \mathbb{E}_{\mathbf{c}_{I^t}: (\mathbf{c}_{I^t}, \cdot) \in E_I^t(I^t), I^t \sim S^t, \mathbf{s}_1^s \sim p(s^s), \mathbf{s}_2^s \sim p(s^s)} [||E_\pi(D_I^s(\mathbf{c}_{I^t}, \mathbf{s}_1^s)) - E_\pi(D_I^s(\mathbf{c}_{I^t}, \mathbf{s}_2^s))||_1] \quad (1)$$

with  $\mathbf{s}_1^s$  and  $\mathbf{s}_2^s$  being two distinct styles sampled from the the prior distribution  $p(s^s) := \mathcal{N}(\mathbf{0}, \mathbf{I})$  being a multivariate Gaussian with zero mean and diagonal unit covariance.

Note that in the above equation,  $E_\pi(\cdot)$  is part of the given navigation policy. We assume the navigation policy is only trained before deciding the target domain; hence,  $E_\pi(\cdot)$  is fixed during the domain adaptation phase. This adoption ensures that PBIT can be used for transferring a policy across domains (potentially not anticipated during policy training) without re-training the the navigation policy.

### 3.3 Reconstruction and Adversarial Loss

Using just policy-based consistency loss would make decoder  $D_I$  ignore the style and decode based only on the content. Inspired by prior work [49, 21], to encour-

age the content to be domain-invariant and style representations to be domain-specific, we adopt the following image and latent representation reconstruction losses, and use  $\mathcal{N}(\mathbf{0}, \mathbf{I})$  for the prior distributions of styles  $p(\mathbf{s}^s)$  and  $p(\mathbf{s}^t)$ :

$$\begin{aligned}\mathcal{L}_{im\_rec} &= \mathbb{E}_{I^t \sim \mathcal{S}^t} [| | | D_I^t(E_I^t(I^t)) - I^t | |_1] + \mathbb{E}_{I^s \sim \mathcal{S}^s} [| | | D_I^s(E_I^s(I^s)) - I^s | |_1], \\ \mathcal{L}_{lat\_rec} &= \mathbb{E}_{\mathbf{c}_{It}: (\mathbf{c}_{It}, \cdot) \in E_I^t(I^t), I^t \sim \mathcal{S}^t, \mathbf{s}^s \sim p(\mathbf{s}^s)} [| | | E_I^s(D_I^s(\mathbf{c}_{It}, \mathbf{s}^s)) - (\mathbf{c}_{It}, \mathbf{s}^s) | |_1] \\ &\quad + \mathbb{E}_{\mathbf{c}_{Is}: (\mathbf{c}_{Is}, \cdot) \in E_I^s(I^s), I^s \sim \mathcal{S}^s, \mathbf{s}^t \sim p(\mathbf{s}^t)} [| | | E_I^t(D_I^t(\mathbf{c}_{Is}, \mathbf{s}^t)) - (\mathbf{c}_{Is}, \mathbf{s}^t) | |_1].\end{aligned}\quad (2)$$

We also use adversarial losses to match the distribution of images to their respective domains. Let  $\text{Dis}^t$  be the discriminator for the target domain  $t$  and  $\text{Dis}^s$  be the discriminator for the source domain  $s$ :

$$\begin{aligned}\mathcal{L}_{adv} &= \mathbb{E}_{\mathbf{c}_{It}: (\mathbf{c}_{It}, \cdot) \in E_I^t(I^t), I^t \sim \mathcal{S}^t, \mathbf{s}^s \sim p(\mathbf{s}^s)} [\log \text{Dis}^s(D_I^s(\mathbf{c}_{It}, \mathbf{s}^s))] \\ &\quad + \mathbb{E}_{\mathbf{c}_{Is}: (\mathbf{c}_{Is}, \cdot) \in E_I^s(I^s), I^s \sim \mathcal{S}^s, \mathbf{s}^t \sim p(\mathbf{s}^t)} [\log \text{Dis}^t(D_I^t(\mathbf{c}_{Is}, \mathbf{s}^t))] \\ &\quad + \mathbb{E}_{I^s \sim \mathcal{S}^s} [\log(1 - \text{Dis}^s(I^s))] + \mathbb{E}_{I^t \sim \mathcal{S}^t} [\log(1 - \text{Dis}^t(I^t))].\end{aligned}\quad (3)$$

Putting everything together, our overall objective is

$$\mathcal{L}_{full} := \lambda_{pol} \mathcal{L}_{pol} + \lambda_{im\_rec} \mathcal{L}_{im\_rec} + \lambda_{lat\_rec} \mathcal{L}_{lat\_rec} + \lambda_{adv} \mathcal{L}_{adv}, \quad (4)$$

where  $\lambda$ .s are hyper-parameters controlling the weight of each loss during training. The cross-domain image translation model consists of  $D_I^t, E_I^t, D_I^s, E_I^s$ , and the optimization admits a mix-max objective:

$$D_I^t, E_I^t, D_I^s, E_I^s = \arg \min_{D_I^t, E_I^t, D_I^s, E_I^s} \max_{\text{Dis}^t, \text{Dis}^s} \mathcal{L}_{full}. \quad (5)$$

## 4 Experimental Setup

We conduct experiments in both the simulator and the real world. For the simulator setting, we use the Habitat simulator [32] with the Gibson [46] and Replica [36] datasets, and consider two visual navigation tasks for domain adaptation: PointGoal and Exploration. We firstly train an RL policy on Gibson for each task, and create a dataset with unlabeled and unpaired images from Gibson and Replica scenes to train PBIT and a CycleGAN baseline. We then benchmark PBIT against direct transfer policy, CycleGAN transfer, and PBIT without policy consistency constraint on both tasks in the Replica dataset.

For the real world, we use LoCoBot [1] as our real-world agent, train PBIT model with 1125 random real-world images from 3 different indoor scenes and 7200 random Gibson images. We then benchmark PBIT against the direct transfer policy baseline.

### 4.1 Navigation Tasks Definitions

**PointGoal Task.** The PointGoal Task is natively implemented in the Habitat simulator [32]. An agent is positioned at a random starting location and orientation in each episode, and is supposed to navigate to a target location. The

agent has two sensors: RGB camera and GPS+Compass. The observation space consists of RGB images of shape  $3 \times 256 \times 256$  and GPS+Compass input of shape  $2 \times 1$ . The action space consists of four actions: STOP (indicating the agent has reached the target location), MOVE-FORWARD ( $0.25m$ ), TURN-LEFT ( $10^\circ$ ), TURN-RIGHT ( $10^\circ$ ). The episode ends immediately after the agent takes the STOP action. Otherwise, the episode automatically ends after 500 steps. As [2] suggested, we used two evaluation metrics: Success and Success weighted by Path Length (SPL). The episode is considered successful if the agent is within  $0.2m$  of the target location when the episode ends. SPL measures also measures the efficiency of the policy in addition to the success, i.e. shorter trajectories lead to higher SPL:  $SPL = \frac{l}{\max(l, p)} S$  where  $l$  is the length of the shortest path possible between the starting location and the target location,  $p$  is the length of the agent's path and  $S$  denotes success. The reward for training RL policies on this task is the decrease in geodesic distance to the point goal.

**Exploration Task.** We follow the Exploartion task setup used in [10] and [6], where an agent is positioned at a random starting location and orientation in each episode, and is supposed to maximize coverage given a fixed time budget of 500 steps. Coverage is defined to be the total area of explored traversable points from the agent's starting location. A traversable point is explored by the agent if it is in the field-of-view of the agent and less than  $3m$  away from the agent. The agent is equipped with two sensors: RGB camera and base odometry sensor. The spec of the RGB camera is the same as in PointGoal task. The base odometry sensor provides the agent with readings that denote the change in the agent's x-y coordinates and orientation. Thus the observation space consists of RGB images of shape  $3 \times 256 \times 256$  and base odometry sensor input of shape  $3 \times 1$ . The action space consists of three actions: MOVE-FORWARD ( $0.25m$ ), TURN-LEFT ( $10^\circ$ ), TURN-RIGHT ( $10^\circ$ ). Each episode ends after 500 steps. As [6] suggested, we use two evaluation metrics, the absolute coverage area in  $m^2$  (Explored Area) and proportion of area explored in the scene (Explored Ratio). Explored Ratio is defined as ratio of coverage to maximum possible coverage in the corresponding scene. During training, the reward received by the agent at each step is equal to the amount of new area explored by that step.

## 4.2 RL Training Details

**Agent Architecture.** Our agent architecture consists of two neural networks: a visual encoder  $E_\pi$  and a policy encoder  $A_\pi$ . The visual encoder  $E_\pi$  is based on the 18-layer ResNet [18].  $E_\pi$  outputs 128-dimensional policy-related representations  $\mathbf{r}_I$  given RGB images of shape  $3 \times 256 \times 256$ . The policy encoder  $A_\pi$  is based on a 2-layer GRU, which takes  $\mathbf{r}_I$  together with readings from either the GPS+Compass sensor in PointGoal task or the base odometry sensor in Exploration task. Details for the architecture are provided in the supplementary material.

**Training.** We train three RL agents in Gibson using PPO [34] with Generalized Advantage Estimation [33]. The first agent is for PointGoal task with  $1.25m$  camera height, which is tested to transfer to Replica. The second is for PointGoal task with  $60cm$  camera height, which is tested to transfer to Real World. The last agent is for Exploration task, which is tested to transfer to Replica. All three agents are trained on the train split of the *pointnav\_gibson\_v1* dataset provided by [32]. The dataset contains episode definitions for all 72 scenes in Gibson. The two PointGoal agents are trained with 8 concurrent workers for around 30 million frames, and achieve SPL=0.80 on the val split of the dataset. The Exploration agent is trained with 24 concurrent workers for around 3 million frames. At each update, each work collects 128 steps of experience and perform 2 PPO epochs with minibatch size  $128 \times 2$  and clipping parameter 0.2. We use discount factor 0.99, GAE parameter 0.95, and the Adam optimizer [24] with learning rate  $2.5 \times 10^{-4}$ .

### 4.3 Baselines

We transfer the trained RL policies to the target domain using the proposed model, Policy-Based Image Translation (PBIT) against 3 baselines:

1. **Direct Transfer:** This is the most common method of transferring navigation policies across domains, which involves directly testing the policy in the target domain without any fine-tuning.
2. **CycleGAN:** CycleGAN is a competitive and popular unsupervised image translation method. This method is designed for static image translation and is agnostic to the navigation policy.
3. **Policy-Based Image Translation w.o. Policy Loss:** This is an ablation of the proposed method without the policy-based consistency loss.

### 4.4 PBIT & Baselines Training/Testing Details

**Domain Translation Dataset.** For Gibson to Replica experiment, we build a dataset of unlabeled and unpaired images from Gibson and Replica scenes. Gibson environment contains 72 scenes and Replica environment contains 18 scenes. For each scene in Gibson, using the function provided by the Habitat simulator [32], we sample 100 random *navigable location*  $\times$  *orientation* pairs and save the first-person view RGB images obtained from those locations and orientations. Similarly, we sample 400 for each scene in Replica. Thus, our Gibson-Replica dataset contains 7200 images from Gibson and 7200 images from Replica. For Gibson to Real World experiment, we use the robot to create a similar unpaired image dataset of 7200 images from Gibson and 1125 images from Real World. Details about how the real-world images are collected is described in Section 4.5.

**Model Architecture.** For PBIT, we follow the setup suggested by [21]. We use several convolutional layers and residual blocks to construct the image encoders  $E_I^s, E_I^t$  and image decoders  $D_I^s, D_I^t$ . We use Instance Normalization [39]

in  $E_I^s, E_I^t$  and Adaptive Instance Normalization [20] in  $D_I^s, D_I^t$ . For the discriminators  $\text{Dis}^s, \text{Dis}^{st}$ , we adopt the multi-scale discriminator architecture proposed by [40]. Detailed descriptions of the architecture are given in the supplementary material. For the CycleGAN baseline, we use the architecture proposed in the CycleGAN paper [49].

**Hyperparameters for PBIT.** PBIT is trained using the full objective defined in Equation (4) and Equation (5). For each task (PointGoal/Exploration) and scenario (Replica/Real-World), we use the same set of hyperparameters  $\lambda_{im\_rec} = 10, \lambda_{lat\_rec} = 1, \lambda_{adv} = 1$ .  $\lambda_{pol}$  is policy-dependent, because the scale of the loss  $\mathcal{L}_{pol}$  defined in Equation (1) depends on the scale of task-specific features  $\mathbf{r}_I$  outputted by  $E_\pi$ . In our case  $\mathbf{r}_I$  is a 128-dimensional vector after ReLU activation layer. For different tasks and scenarios, we use the same formula to calculate the proper  $\lambda_{pol} = \frac{1.0}{\mathbb{E}_{I^s \sim \mathcal{S}^s}[\|E_\pi(I^s)\|_1]}$ , where  $\mathbb{E}_{I^s \sim \mathcal{S}^s}[\|E_\pi(I^s)\|_1]$  is estimated using the Gibson images from the domain translation dataset.

**Training.** For PBIT, PBIT without Policy Loss, and CycleGAN, we use the Adam optimizer [24] with 0.0001 initial learning rate,  $\beta_1 = 0.5$  and  $\beta_2 = 0.999$ . The learning rate is halved every 100k iterations. We train all the models for 500k iterations with batch size 1, on the domain translation dataset for each task (PointGoal/Exploration) and scenario (Replica/Real-World).

**Testing.** When the RL agent trained in Gibson is tested in Replica or Real-World, the Direct Transfer baseline directly performs on the raw input images from Replica or Real-World. For our proposed model, at each step, PBIT translates the input Replica/Real-World image to Gibson image, and then the Gibson RL agent takes in the translated image as input and output the action for the task. The testing procedure for the CycleGAN baseline is the same as PBIT.

## 4.5 Real World Experiment

**Robot configuration.** We present results of unsupervised zero-shot simulation to real-world indoor navigation on a LoCoBot [1]. The robot has an RGB camera 60cm from the ground, and is programmed to take action space: [stop, forward 0.25m, left 10°, right 10°].

**Data collection and training.** By taking pictures at random locations with the robot, we collect a total of 1125 images from three indoor locations: 1) 397 images from a meeting room with chairs and tables. 2) 728 images from the corridor of a building. 3) 151 images from a large study place chairs and tables.

We train an agent of camera height 60cm in Gibson with the same architecture as in section 4.2, and a policy based transfer model from 7500 random Gibson images and the 1125 real-world images.

405  
406  
407  
408  
409  
410  
411  
412  
413  
414  
415  
416  
417  
418  
419  
420  
421  
422  
423  
424  
425  
426  
427  
428  
429  
430  
431  
432  
433  
434  
435  
436  
437  
438  
439  
440  
441  
442  
443  
444  
445  
446  
447  
448  
449  
Table 1: **PointGoal Results.** The performance of the proposed method Policy-Based Image Translation (PBIT) as compared to the baselines on the PointGoal task when transferred from Gibson domain to the Replica domain.

	SPL	Success Rate	Collisions
Direct Transfer	0.505	0.688	<b>38.7</b>
CycleGAN	0.605	0.803	50.6
PBIT w.o. Policy Loss	0.669	0.852	40.6
<b>PBIT</b>	<b>0.712</b>	<b>0.881</b>	39.5

410  
411  
412  
413  
414  
415  
416  
417  
418  
419  
420  
421  
422  
423  
424  
425  
426  
427  
428  
429  
430  
431  
432  
433  
434  
435  
436  
437  
438  
439  
440  
441  
442  
443  
444  
445  
446  
447  
448  
449  
Table 2: **Exploration Results.** The performance of the proposed method Policy-Based Image Translation (PBIT) as compared to the baselines on the Exploration task when transferred from Gibson domain to the Replica domain.

	Explored Ratio	Explored Area ( $m^2$ )	Collisions
Direct Transfer	0.832	22.9	<b>59.4</b>
CycleGAN	0.885	24.7	84.2
PBIT w.o. Policy Loss	0.879	24.6	70.1
<b>PBIT</b>	<b>0.897</b>	<b>25.3</b>	73.6

## 5 Results

We transfer the trained RL policies to the target domain using the proposed model, Policy-Based Image Translation (PBIT) against 3 baselines defined in Section 4.3. We first present domain adaptation results from Gibson to Replica domains within the Habitat simulator and then present results of sim-to-real transfer from Gibson to real-world office scenes.

### 5.1 Gibson to Replica

We use the script provided by Habitat [32] to generate 50 test episodes for each of the 18 scenes in Replica (900 episodes in total). The script makes sure that the each test episode is reasonably difficult (an episode is trivial if there is a obstacle-free straight line between the starting and target locations). We use the same 900 episodes across 18 scenes to evaluate the performance of each model in Replica.

For the **PointGoal** navigation task, we compare all the methods across Success Weighted by Path Length (SPL), Success Rate and number of collisions. For the **Exploration** task, we compare all the methods using Explored Area in  $m^2$ , ratio of the environment explored and number of collisions. The performances of our method and all the baselines for the PointGoal task are presented in Table 1 and for the Exploration task are presented in Table 2.

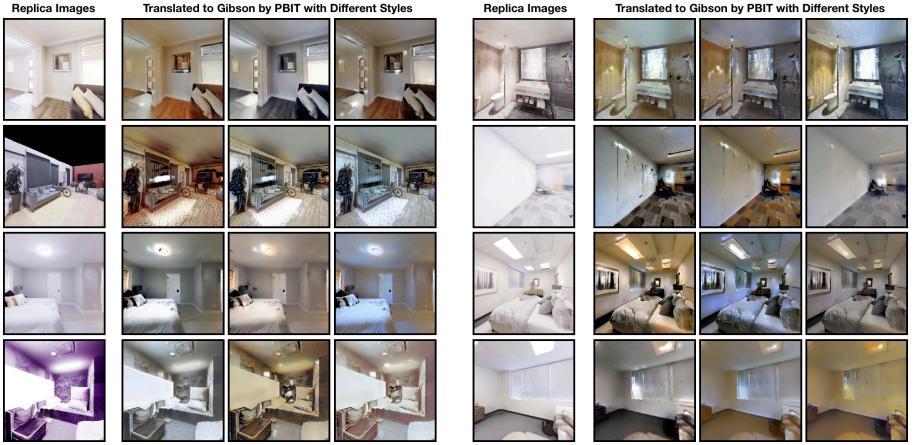


Fig. 4: **Replica Images Translated to Gibson.** The first and fifth columns are the input images from Replica during test time. The other columns are images translated to Gibson domain by PBIT. The Gibson RL agent relies on the translated images (which preserve the policy-relevant features) to perform well in Replica.

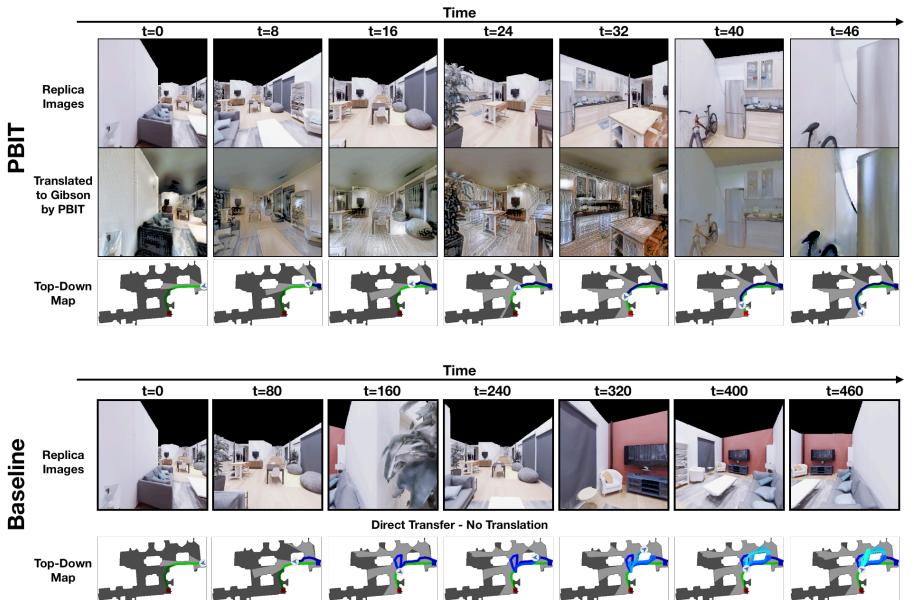


Fig. 5: **Trajectory Comparison between PBIT and Baseline (Direct Transfer) on PointGoal Task in Replica.** The upper half of the figure is the trajectory of PBIT: the agent successfully navigates from a corner of the apartment to the fridge in 46 steps, by seeing the translated images by PBIT. The agent takes almost the shortest path possible, as shown in the Top-Down Map (not visible to the agent). The lower half of the figure is the trajectory of the Direct Transfer baseline on the same test episode. The Direct Transfer agent fails to navigate to the target location and gets lost, even after 460 steps.



Fig. 6: **Real World Images Translated to Gibson.** The first and fifth columns are the input images from Real World during test time. The other columns are images translated to Gibson domain by PBIT. Although the agent has never been trained to navigate with Real World images, it can recognize the translated images (which preserve the policy-relevant features) and perform well on PointGoal in Real World.

PBIT outperforms all the baselines on both the tasks. It improves the SPL from 0.605 to 0.712 for PointGoal and Explored Ratio from 0.885 to 0.897 for Exploration as compared to the CycleGAN baseline. Note that the number of collisions for the Direct Transfer baseline are low because in many episodes this baseline does not move and just turns around on the spot. This highlights the visual domain gap between the two domains. Lower performance of ‘PBIT w.o. Policy Loss’ ablation highlights the importance of policy-based consistency loss. The exploration ratios of all the methods in Table 2 are high on a absolute level because the Replica scenes are small usually having one or two rooms. Just turning on the spot leads to an exploration ratio of 0.75.

In Figure 4, we visualize some examples of images in the target Replica domain translated to the source Gibson domain using PBIT using different styles. The examples indicate that policy-relevant characteristics of the image such as corners of obstacles, walls and free space are preserved during the translation. In Figure 5, we visualize an example trajectory for the PointGoal task using the proposed method PBIT (Fig 5 above) and the Direct Transfer baseline (Fig 5 below). The figure shows the images observed by the agent in the target domain, the translated images and a top-down map (not visible to the agent) showing the point goal coordinates and the agent’s path.

## 5.2 Gibson to Real-world

We now transfer the PointGoal navigation policy to the real-world using the proposed method PBIT and compare it to the Direct Transfer baseline. We transfer the navigation policy to a LoCoBot [1] using the PyRobot API [30] for both the

540  
 541 **Table 3: Real-world results.** Table comparing the performance of baseline and our  
 542 method under 20 goals within 3 scenes. We the average the collision rate, the success  
 543 rate, the final stopping distance from the PointGoal in meters, and the average number  
 544 of steps the robot takes to finish a trial excluding trials with collisions. Although both  
 545 agent fail in first unseen scene where there is high ground reflection, our agent demon-  
 546 strates better performance in terms of completion rate and average steps taken in the  
 547 second unseen location. Our PBIT agent achieves overwhelmingly better performance  
 548 in all metrics in the third scene since 151 sample images have been included in the  
 549 **transfer training set.**

Episode Specification				Baseline: Direct Transfer				PBIT			
Ep No	Dist	Angle	Obstacle in way	Steps	Collision	Stopping Distance	Success	Steps	Collision	Stopping Distance	Success
<b>Scene 1: Wooden corridor with intense ground reflection (Not in training set of PBIT)</b>											
1	4.00	0.00	FALSE	99	FALSE	5.62	FALSE	99	FALSE	3.48	FALSE
2	2.83	45.00	FALSE	68	TRUE	4.12	FALSE	99	FALSE	1.80	FALSE
SCENE AVG	-	-	-	99	50%	4.87	0%	99	0%	2.64	0%
<b>Scene 2: Public kitchen area with high traffic (Not in training set of PBIT)</b>											
3	2.00	0.00	FALSE	32	TRUE	0.50	FALSE	36	FALSE	0.18	TRUE
4	2.00	0.00	FALSE	35	TRUE	0.56	FALSE	10	FALSE	0.04	TRUE
5	2.24	333.43	FALSE	31	FALSE	0.02	TRUE	16	FALSE	0.07	TRUE
6	2.24	153.43	FALSE	38	FALSE	0.08	TRUE	43	FALSE	0.09	TRUE
7	4.12	345.96	TRUE	80	TRUE	4.10	FALSE	44	TRUE	1.96	FALSE
8	4.47	26.57	TRUE	99	FALSE	4.01	FALSE	99	FALSE	2.85	FALSE
9	4.47	26.57	TRUE	99	FALSE	4.49	FALSE	70	FALSE	0.14	TRUE
10	5.39	21.80	TRUE	99	FALSE	5.50	FALSE	99	FALSE	2.73	FALSE
11	2.83	45.00	TRUE	99	FALSE	3.57	FALSE	99	FALSE	2.84	FALSE
SCENE AVG	-	-	-	77.5	33.3%	2.54	22.2%	59	11.1%	1.21	55.6%
<b>Scene 3: Large common area with few traffic (151 randomly-sampled images in training set of PBIT)</b>											
12	2.83	45.00	FALSE	99	FALSE	3.41	FALSE	21	FALSE	0.15	TRUE
13	4.47	333.43	TRUE	99	FALSE	4.68	FALSE	41	FALSE	0.02	TRUE
14	4.00	0.00	TRUE	99	FALSE	4.43	FALSE	41	FALSE	0.13	TRUE
15	5.39	21.80	TRUE	10	TRUE	4.60	FALSE	32	FALSE	0.13	TRUE
16	4.24	315.00	FALSE	99	FALSE	6.20	FALSE	73	FALSE	0.14	TRUE
17	4.00	0.00	FALSE	36	TRUE	3.88	FALSE	63	FALSE	0.09	TRUE
18	1.41	45.00	FALSE	99	FALSE	1.95	FALSE	9	FALSE	0.19	TRUE
19	1.41	135.00	FALSE	50	FALSE	0.15	TRUE	29	FALSE	0.07	TRUE
20	4.47	26.57	TRUE	99	FALSE	4.73	FALSE	27	FALSE	0.19	TRUE
SCENE AVG	-	-	-	92	22.2%	3.78	11.1%	37.3	0%	0.13	100%
AVG	-	-	-	86.29	30%	3.33	15%	52.9	5%	0.87	70%

571 methods. We conduct 20 trials across 3 different scenes in the real-world. For  
 572 each trial, we set the target point for the robot through PointGoal, and update  
 573 the PointGoal each step according to the relative position calculated through the  
 574 internal odometry sensors, and stop the trial either when the robot have reached  
 575 the goal (distance less than 20cm) or after 100 steps. Each trial specification  
 576 and the corresponding results are presented in Table 3. PBIT achieves an overall  
 577 55% improvement in success rate over the Direct Transfer baseline across all the  
 578 trials. PBIT also has a much lower collision rate as compared to the baseline.

579 In Figure 6, we visualize some examples of images seen by the agent in the  
 580 real-world and their translation to the source Gibson domain using our PBIT-  
 581 model. The examples indicate that the model generates good translations similar  
 582 to images in the Gibson domain. For example, the dark grey carpet floors in the  
 583 office space scenes in the real-world are successfully translated to brown floors,  
 584 representative of wooden floors of apartment scenes in the Gibson domain. At the



Fig. 7: **Sample Real World Trajectory on PointGoal Task.** Figure contains raw inputs from Real World (row one), translated Gibson images by PBIT (row 2), and a third-person perspective from the back. The PBIT agent successfully reached its destination (a trash can) by avoiding an obstacle (a chair) in its way.

same time, navigation relevant details such as the boundary between the floor and walls and free space area, are preserved during translation. In Figure 7, we show an example of a successful trajectory in the real-world using PBIT. It shows some of the images seen by the agent during the trajectory, the corresponding translations and a third-person view of the robot. The trajectory shows the PBIT is able to successfully navigate around the blue chair obstacle to reach the pointgoal.

## 6 Conclusion

In this paper, we proposed domain adaptation method for transferring navigation policies from simulation to the real-world. Given a navigation policy in the source domain, our method translates images from the target domain to the source domain such that the translations are consistent with the task-specific and domain-invariant representations learnt by the given policy. Our experiments across two different tasks for domain transfer in simulation show that the proposed method can improve the performance on the transferred navigation policies over baselines. We also show strong performance of navigation policies transferred from simulation to the real-world using our method. In this paper, we considered navigation tasks involving mostly spatial reasoning. In the future, the proposed method can be extended to navigation tasks involving more semantic reasoning by incorporating semantic consistency losses along with the policy consistency losses.

## Acknowledgements

This work was supported in part by the DARPA grants FA875018C0150 and HR00111990016, NSF IIS1763562, Apple, and NVIDIA’s GPUs.

## References

- Locobot · an open source low cost robot, <http://www.locobot.org/>
- Anderson, P., Chang, A.X., Chaplot, D.S., Dosovitskiy, A., Gupta, S., Koltun, V., Kosecka, J., Malik, J., Mottaghi, R., Savva, M., Zamir, A.R.: On evaluation of embodied navigation agents. CoRR **abs/1807.06757** (2018), <http://arxiv.org/abs/1807.06757>
- Anderson, P., Wu, Q., Teney, D., Bruce, J., Johnson, M., Sünderhauf, N., Reid, I., Gould, S., van den Hengel, A.: Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3674–3683 (2018)
- Chang, A., Dai, A., Funkhouser, T., Halber, M., Niessner, M., Savva, M., Song, S., Zeng, A., Zhang, Y.: Matterport3d: Learning from rgb-d data in indoor environments. International Conference on 3D Vision (3DV) (2017)
- Chaplot, D.S., Gandhi, D., Gupta, S., Gupta, A., Salakhutdinov, R.: Learning to explore using active neural slam. In: ICLR (2020)
- Chaplot, D.S., Gandhi, D., Gupta, S., Gupta, A., Salakhutdinov, R.: Learning to explore using active neural slam. In: International Conference on Learning Representations (2020), <https://openreview.net/forum?id=HklXn1BKDH>
- Chaplot, D.S., Sathyendra, K.M., Pasumarthi, R.K., Rajagopal, D., Salakhutdinov, R.: Gated-attention architectures for task-oriented language grounding. In: Thirty-Second AAAI Conference on Artificial Intelligence (2018)
- Chen, H., Suhr, A., Misra, D., Snavely, N., Artzi, Y.: Touchdown: Natural language navigation and spatial reasoning in visual street environments. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 12538–12547 (2019)
- Chen, T., Gupta, S., Gupta, A.: Learning exploration policies for navigation. arXiv preprint arXiv:1903.01959 (2019)
- Chen, T., Gupta, S., Gupta, A.: Learning exploration policies for navigation (1 2019), 7th International Conference on Learning Representations, ICLR 2019 ; Conference date: 06-05-2019 Through 09-05-2019
- Das, A., Datta, S., Gkioxari, G., Lee, S., Parikh, D., Batra, D.: Embodied question answering. In: CVPR (2018)
- Dosovitskiy, A., Koltun, V.: Learning to act by predicting the future. In: ICLR (2017)
- Fang, K., Toshev, A., Fei-Fei, L., Savarese, S.: Scene memory transformer for embodied agents in long-horizon tasks. In: CVPR (2019)
- Fried, D., Hu, R., Cirik, V., Rohrbach, A., Andreas, J., Morency, L.P., Berg-Kirkpatrick, T., Saenko, K., Klein, D., Darrell, T.: Speaker-follower models for vision-and-language navigation. In: Advances in Neural Information Processing Systems. pp. 3314–3325 (2018)
- Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Advances in neural information processing systems. pp. 2672–2680 (2014)
- Gordon, D., Kembhavi, A., Rastegari, M., Redmon, J., Fox, D., Farhadi, A.: Iqa: Visual question answering in interactive environments. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4089–4098 (2018)
- Gupta, S., Davidson, J., Levine, S., Sukthankar, R., Malik, J.: Cognitive mapping and planning for visual navigation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2616–2625 (2017)

- 675 18. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition  
676 (2015) 675
- 677 19. Hermann, K.M., Hill, F., Green, S., Wang, F., Faulkner, R., Soyer, H., Szepesvari,  
678 D., Czarnecki, W.M., Jaderberg, M., Teplyashin, D., et al.: Grounded language  
679 learning in a simulated 3d world. arXiv preprint arXiv:1706.06551 (2017) 677
- 680 20. Huang, X., Belongie, S.: Arbitrary style transfer in real-time with adaptive instance  
681 normalization. In: The IEEE International Conference on Computer Vision (ICCV)  
682 (Oct 2017) 680
- 683 21. Huang, X., Liu, M.Y., Belongie, S., Kautz, J.: Multimodal unsupervised image-  
684 to-image translation. In: Proceedings of the European Conference on Computer  
685 Vision (ECCV). pp. 172–189 (2018) 683
- 686 22. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with  
687 conditional adversarial networks. In: Proceedings of the IEEE conference on computer  
688 vision and pattern recognition. pp. 1125–1134 (2017) 686
- 689 23. Kim, T., Cha, M., Kim, H., Lee, J.K., Kim, J.: Learning to discover cross-domain  
690 relations with generative adversarial networks. In: Proceedings of the 34th Interna-  
691 tional Conference on Machine Learning-Volume 70. pp. 1857–1865. JMLR. org  
692 (2017) 689
- 693 24. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization (2014) 691
- 694 25. Kolve, E., Mottaghi, R., Han, W., VanderBilt, E., Weihs, L., Herrasti, A., Gordon,  
695 D., Zhu, Y., Gupta, A., Farhadi, A.: AI2-THOR: An Interactive 3D Environment  
696 for Visual AI. arXiv (2017) 692
- 697 26. Lample, G., Chaptal, D.S.: Playing FPS games with deep reinforcement learning.  
698 In: Thirty-First AAAI Conference on Artificial Intelligence (2017) 693
- 699 27. Lee, H.Y., Tseng, H.Y., Huang, J.B., Singh, M., Yang, M.H.: Diverse image-to-  
700 image translation via disentangled representations. In: Proceedings of the Euro-  
701 pean conference on computer vision (ECCV). pp. 35–51 (2018) 698
- 702 28. Liu, M.Y., Breuel, T., Kautz, J.: Unsupervised image-to-image translation net-  
703 works. In: Advances in neural information processing systems. pp. 700–708 (2017) 700
- 704 29. Mirowski, P., Pascanu, R., Viola, F., Soyer, H., Ballard, A.J., Banino, A., Denil, M.,  
705 Goroshin, R., Sifre, L., Kavukcuoglu, K., et al.: Learning to navigate in complex  
706 environments. ICLR (2017) 701
- 707 30. Murali, A., Chen, T., Alwala, K.V., Gandhi, D., Pinto, L., Gupta, S., Gupta, A.:  
708 Pyrobot: An open-source robotics framework for research and benchmarking. arXiv  
709 preprint arXiv:1906.08236 (2019) 705
- 710 31. Savva, M., Chang, A.X., Dosovitskiy, A., Funkhouser, T., Koltun, V.: MI-  
711 NOS: Multimodal indoor simulator for navigation in complex environments.  
712 arXiv:1712.03931 (2017) 708
- 713 32. Savva, M., Kadian, A., Maksmets, O., Zhao, Y., Wijmans, E., Jain, B., Straub, J.,  
714 Liu, J., Koltun, V., Malik, J., et al.: Habitat: A platform for embodied ai research.  
715 In: ICCV (2019) 711
- 716 33. Schulman, J., Moritz, P., Levine, S., Jordan, M., Abbeel, P.: High-dimensional  
717 continuous control using generalized advantage estimation (2015) 713
- 718 34. Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O.: Proximal policy  
719 optimization algorithms. arXiv preprint arXiv:1707.06347 (2017) 715
- 720 35. Shrivastava, A., Pfister, T., Tuzel, O., Susskind, J., Wang, W., Webb, R.: Learn-  
721 ing from simulated and unsupervised images through adversarial training. In: Pro-  
722 ceedings of the IEEE conference on computer vision and pattern recognition. pp.  
723 2107–2116 (2017) 718

- 720 36. Straub, J., Whelan, T., Ma, L., Chen, Y., Wijmans, E., Green, S., Engel, J.J., Mur-  
721 Artal, R., Ren, C., Verma, S., Clarkson, A., Yan, M., Budge, B., Yan, Y., Pan,  
722 X., Yon, J., Zou, Y., Leon, K., Carter, N., Briales, J., Gillingham, T., Mueggler,  
723 E., Pesqueira, L., Savva, M., Batra, D., Strasdat, H.M., Nardi, R.D., Goesele, M.,  
724 Lovegrove, S., Newcombe, R.: The Replica dataset: A digital replica of indoor  
725 spaces. arXiv preprint arXiv:1906.05797 (2019)
- 726 37. Taigman, Y., Polyak, A., Wolf, L.: Unsupervised cross-domain image generation.  
727 arXiv preprint arXiv:1611.02200 (2016)
- 728 38. Tzeng, E., Devin, C., Hoffman, J., Finn, C., Abbeel, P., Levine, S., Saenko, K., Dar-  
729 rell, T.: Adapting deep visuomotor representations with weak pairwise constraints.  
730 arXiv preprint arXiv:1511.07111 (2015)
- 731 39. Ulyanov, D., Vedaldi, A., Lempitsky, V.: Improved texture networks: Maximizing  
732 quality and diversity in feed-forward stylization and texture synthesis. In: The  
733 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (July  
734 2017)
- 735 40. Wang, T.C., Liu, M.Y., Zhu, J.Y., Tao, A., Kautz, J., Catanzaro, B.: High-  
736 resolution image synthesis and semantic manipulation with conditional gans. In:  
737 The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June  
738 2018)
- 739 41. Wang, X., Huang, Q., Celikyilmaz, A., Gao, J., Shen, D., Wang, Y.F., Wang,  
740 W.Y., Zhang, L.: Reinforced cross-modal matching and self-supervised imitation  
741 learning for vision-language navigation. In: Proceedings of the IEEE Conference  
742 on Computer Vision and Pattern Recognition. pp. 6629–6638 (2019)
- 743 42. Wijmans, E., Kadian, A., Morcos, A., Lee, S., Essa, I., Parikh, D., Savva, M.,  
744 Batra, D.: Decentralized distributed ppo: Solving pointgoal navigation. In: ICLR  
745 (2020)
- 746 43. Wu, Y., Wu, Y., Gkioxari, G., Tian, Y.: Building generalizable agents with a  
747 realistic and rich 3d environment. arXiv preprint arXiv:1801.02209 (2018)
- 748 44. Wu, Y., Wu, Y., Tamar, A., Russell, S., Gkioxari, G., Tian, Y.: Bayesian relational  
749 memory for semantic visual navigation. In: Proceedings of the IEEE International  
750 Conference on Computer Vision. pp. 2769–2779 (2019)
- 751 45. Wu, Y., Tian, Y.: Training agent for first-person shooter game with actor-critic  
752 curriculum learning. In: ICLR (2017)
- 753 46. Xia, F., R. Zamir, A., He, Z.Y., Sax, A., Malik, J., Savarese, S.: Gibson Env: real-  
754 world perception for embodied agents. In: Computer Vision and Pattern Recog-  
755 nition (CVPR), 2018 IEEE Conference on. IEEE (2018)
- 756 47. Yang, W., Wang, X., Farhadi, A., Gupta, A., Mottaghi, R.: Visual semantic navi-  
757 gation using scene priors. arXiv preprint arXiv:1810.06543 (2018)
- 758 48. Yi, Z., Zhang, H., Tan, P., Gong, M.: Dualgan: Unsupervised dual learning for  
759 image-to-image translation. In: Proceedings of the IEEE international conference  
760 on computer vision. pp. 2849–2857 (2017)
- 761 49. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation  
762 using cycle-consistent adversarial networks. In: Proceedings of the IEEE interna-  
763 tional conference on computer vision. pp. 2223–2232 (2017)
- 764 50. Zhu, J.Y., Zhang, R., Pathak, D., Darrell, T., Efros, A.A., Wang, O., Shechtman,  
E.: Toward multimodal image-to-image translation. In: Advances in neural information  
processing systems. pp. 465–476 (2017)
51. Zhu, Y., Mottaghi, R., Kolve, E., Lim, J.J., Gupta, A., Fei-Fei, L., Farhadi, A.: Target-driven visual navigation in indoor scenes using deep reinforcement learning.  
In: 2017 IEEE international conference on robotics and automation (ICRA). pp.  
3357–3364. IEEE (2017)