

# Localising In Complex Scenes Using Balanced Adversarial Adaptation

Gil Avraham\* Yan Zuo\*  
Tom Drummond

ARC Centre of Excellence for Robotic Vision, Monash University, Australia

{gil.avraham, yan.zuo, tom.drummond}@monash.edu

## Abstract

*Domain adaptation and generative modelling have collectively mitigated the expensive nature of data collection and labelling by leveraging the rich abundance of accurate, labelled data in simulation environments. In this work, we study the performance gap that exists between representations optimised for localisation on simulation environments and the application of such representations in a real-world setting. Our method exploits the shared geometric similarities between simulation and real-world environments whilst maintaining invariance towards visual discrepancies. This is achieved by optimising a representation extractor to project both simulated and real representations into a shared representation space. Our method uses a symmetrical adversarial approach which encourages the representation extractor to conceal the domain that features are extracted from and simultaneously preserves robust attributes between source and target domains that are beneficial for localisation. We evaluate our method by adapting representations optimised for indoor Habitat simulated environments (Matterport3D and Replica) to a real-world indoor environment (Active Vision Dataset), showing that it compares favourably against fully-supervised approaches.*

## 1. Introduction

Despite deep learning largely entrenching itself as a cornerstone of modern computer vision, the technique commonly relies on training models which require heavy supervision with labelled data that is costly to obtain [33]. Whilst simulation offers potential refuge from this problem, this data often does not closely resemble real-world data. Subsequently, when models trained on simulated data are deployed in the real-world, large performance gaps are often observed [59]. Domain adaptation aims to bridge this gap, although modern approaches are largely restricted to image understanding tasks such as recognition, segmenta-

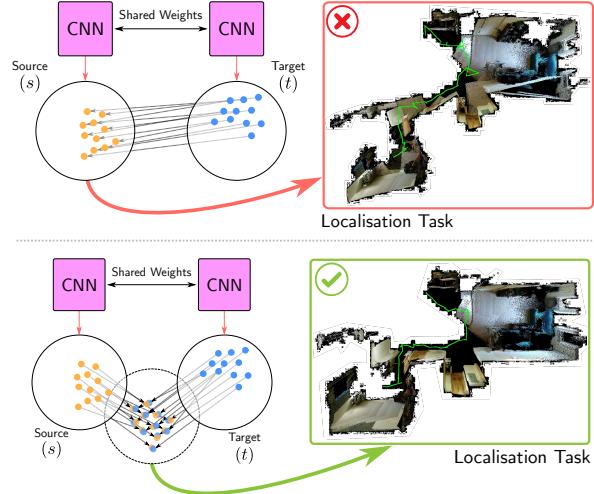


Figure 1: The Balanced Adversarial Adaptation (BAA) Network learns a mapping from a source and target domain to a shared domain. BAA bridges the generalisation gap between domains by leveraging cheap and accurate representations acquired from a simulation environment and adapting these representations to operate on real-world environments.

tion and detection [20, 34, 47], where it has found success across these tasks through training on a source domain and leveraging data from the target domain to minimise the performance gap between domains. This domain gap in image understanding extends into the area of inference in 3D space and geometry of a scene, where tasks such as monocular visual odometry (VO) has seen a renewal of interest with the advent of deep representational learning [60, 9].

Performing this task well is crucial towards creating fully autonomous agents, where spatial awareness about the scene plays a key role in assisting with applications such as navigation and human-robot interaction [55, 17]. This is an active area of research where supervised learning of the localisation task has resulted in frameworks that learn robust representations which carry the geometric proper-

\*Authors contributed equally

ties of the scene [3, 64, 8]. Although these approaches have demonstrated promising results, they require copious amounts of training data which are both difficult to obtain and biases the model towards becoming highly attuned towards a specific domain. Despite this, it is apparent that the rich representations offered by deep learning are desirable for building effective systems for tackling difficult visual odometry tasks (such as large camera baseline shifts between frames) [21, 2].

In this paper, we explore a novel method to adapt between a simulated and real-world domain for the task of localisation. In particular, our framework adapts directly on the feature representations used for localising using a symmetrical adversarial technique, where both the source and target domain representations are mapped to a shared “in-between” domain (see Fig. 1). The representation extractor is encouraged to optimise the representations that both conceal the domain from which they came from and also preserve mutual geometrical information between the domains which are used to estimate an agent’s pose. We show the effectiveness of this approach, where we use our Balanced Adversarial Adaptation (BAA) Network to adapt from simulation environments to real-world environments and show localisation performance that is on-par or surpasses *supervised models trained on real-world data*. Our approach offers the following contributions:

- A novel, bi-directional adaptation approach which utilises a symmetrical adversarial technique. This symmetrical approach yields demonstrably improved adapted localisation over methods which use unidirectional adaptation.
- Empirically, we show that our weakly-supervised model (trained on simulation data) can perform on-par or even outperform *supervised* methods for localising on the challenging real-world Active Vision dataset.
- To the best of our knowledge, this is the first work to explore domain adaptation in the context of localisation. Our work establishes the baseline for any future research on this important task.

## 2. Related Work

### 2.1. Localisation

The task of localisation involves estimating the camera pose from observed image data. Initial frame-by-frame methods such as [10, 29] treated the localisation task as a regression problem and directly optimised on the camera pose, where the former included optical flow between pairs of sequential frames as additional information to the input. Later works exploited the geometric properties of the scene, using these as constraints to restrict the search space for optimisation of their models [36, 12]. [3] used latent codes to

capture scene geometry, using an encoder-decoder setup to jointly optimise for both depth and camera pose.

Related sequence-to-sequence methods maintain an internal memory of previous observations [60, 9] where a Convolution Neural Network (CNN) and an LSTM [23] is used for feature extraction and inference of dynamics across sequences of frames respectively. [64] proposed to optimise for depth estimation and camera pose together, through regressing on the camera pose and estimating a cost volume for reconstructing the environment. [8] uses bi-directional LSTMs to smoothly estimate camera pose and localise over video sequences.

Map-based approaches attempt to explicitly incorporate previously visited states internally where a fixed sized latent map from a 2D top-down viewpoint perspective is maintained [62, 43]. [18] extended up on this approach towards exploring real-world environments, modifying the global map representation to an egocentric one. MapNet [21] further extended upon the approach of [18], which maintained a ground-projected allocentric map. The approach used a brute-force localisation approach where the current state was localised against each patch of the maintained allocentric map and in doing so, temporal information was lost in this process, where non-viable areas of the map were considered for localising. [2] addressed this issue through the maintaining a short term memory point-embedding map to localise against incoming states. Most recently, [31] proposed an online meta-learning algorithm which enabled a VO framework to continuously adapt to new environments in a self-supervised manner.

### 2.2. Domain Adaptation

A popular type of transfer learning is the task of domain adaption. Domain adaptation methods aim to address the detrimental effects of domain shift through learning a model which reduces this shift between the source and target distributions [42]. Shallow adaptation methods used fixed feature representations, building a mapping between the source and target domains [11, 5, 7, 15, 41, 14, 40]. These shallow approaches were also used in conjunction with deep neural networks to extract representations which were used to align feature sub-spaces between domains [24, 46, 38, 63].

Recently, Deep Domain Adaptation has emerged as a method of choice when dealing with large amounts of unlabelled data. Deep domain adaptation was first proposed by [13], where both domain adaptation and feature learning were combined within a single training process. This enabled the learning of representations that were invariant to changes in domain. Most modern approaches to domain adaptation learn transformations that map features from the source domain to the target domain’s feature space [57, 35, 4].

Commonly, modern deep domain adaptation methods

utilise adversarial training as part of their process [58, 44, 32, 56]. In particular, Generative Adversarial Networks (GANs) [16] are often directly incorporated into adversarial mapping approaches and several training setups encourage the learning of domain-invariant features in this manner [65, 54, 48, 61, 25].

### 3. Preliminaries

As this work addresses the problem of adapting representations that are suited towards localisation, the choice of representations is critical for this task. For an RGB image  $\mathcal{I} \in \mathbb{R}^{H \times W \times 3}$ , it is desirable to have a mapping function  $\mathcal{F} : \mathcal{I} \rightarrow \mathcal{H}$  that can transform the given image  $\mathcal{I}$  to spatial representations  $\mathcal{H} \in \mathbb{R}^{H_r \times W_r \times C}$ , where  $H_r$  and  $W_r$  denote the respective height and width of a representation tensor and  $C$  denotes the number of channels. Furthermore, if we observe every spatial point in  $\mathcal{H}$ ,  $\ell_i \in \mathbb{R}^C$  as a list of vectors  $\{\ell_i\}_{i=0}^{H_r W_r}$ , then a single vector representation is sampled from:  $\ell_i \sim p_{\mathcal{F}}(\ell | P_i)$ , where  $P_i \in \mathbb{R}^{H_u \times W_u \times 3}$  corresponds to a patch in the image  $\mathcal{I}$ . For the task of localisation, ideally the mapping function  $\mathcal{F}$  should be optimised such that:

1. For a spatial representation  $\mathcal{H}$ , two representation vectors are mutually exclusive (*i.e.*  $\{\forall \ell_i, \ell_j \in \mathcal{H} \wedge i \neq j, 0 = \langle \ell_i, \ell_j \rangle\}$ ). This ensures the uniqueness of a local descriptor that allows representations across a sequence of images to be matched distinctively [28].
2. Given a representation vector  $\ell_{1,i}$  which corresponds to a patch  $P_{1,i}$  in image  $I_1$ , applying an homography on an image  $\mathcal{I}_2 = H\mathcal{I}_1$ , yields a corresponding patch  $P_{2,j}$  with representation vector  $\ell_{2,j}$ . If the patches  $P_{1,i}$  and  $P_{2,j}$  correspond to the same 3D location, then the mapping function  $\mathcal{F}$  should yield  $\ell_{1,i} = \ell_{2,j}$  [53].

#### 3.1. A Primer on EMP-Net

EMP-Net [2] provides a recent baseline which meets the requirements covered above. The mapping function  $\mathcal{F}$  is optimised such that every representation vector  $\ell_i$ , corresponds to a 3D location in a scene, and the representations are mutually exclusive. This is achieved by first constructing a ground-truth correspondence matrix  $\mathcal{C}_{gt,t}$  at time  $t$ , between a sequence of frames. The operation is performed by taking every frame  $F_t$  in a sequence, which contains RGB-D information and the pose,  $T_t \in \mathbb{SE}(3)$ , and using camera intrinsics  $K$ , depth information  $D_t$  and pose  $T_t$  for generating 3D point correspondences relative to previous frames  $\{F_{t'}\}_{t'=t-b}^{t-1}$  (a buffer of size  $b = 4$  is used in [2]). This formulation produces a sparse correspondence matrix  $\mathcal{C}_{gt,t}$ , where every 3D point in frame  $F_t$  corresponds only to matching 3D points in frames  $\{F_{t'}\}_{t'=t-b}^{t-1}$ .

By using this correspondence matrix  $\mathcal{C}_{gt,t}$ , as one-hot ground-truth labels, a mapping  $\mathcal{F}$  can be optimised to pro-

duce representations that yield an inferred correspondence matrix  $\mathcal{C}_{f,t}$ . This is performed by computing a spatial representation map  $\mathcal{H}_{t-b} = \mathcal{F}(F_{t-b})$ , where  $\mathcal{F}$  is implemented using a convolutional neural network. A list of vector representations  $\{\ell_{i,t-b}\}_{i=0}^{H_r W_r}$  is obtained from the spatial map  $\mathcal{H}_{t-b}$ . By iteratively performing this operation up to  $F_t$ , the inferred correspondence matrix  $\mathcal{C}_{f,t}$  is produced by taking the negative pairwise distance between  $\{\ell_{i,t}\}_{i=0}^{H_r W_r}$  and  $\{\ell_{i,t-b}\}_{i=0}^{H_r W_r}$  and by applying a Softmax operation, we obtain a probability distribution for every representation vector  $\ell_{i,t} \in \mathcal{H}_t$ . For optimising the mapping function  $\mathcal{F}$ , a cross-entropy loss is taken between the ground-truth correspondence matrix and the inferred correspondence matrix as follows:

$$CE_{loss} = -\frac{1}{H_r W_r} \sum_{j=1}^{H_r W_r} \sum_{i=1}^{b H_r W_r} \mathcal{C}_{gt,t}[i, j] \log \mathcal{C}_{f,t}[i, j] \quad (1)$$

### 4. Balanced Adversarial Adaptation

#### 4.1. Problem Statement

In this adaptation problem, there exists source and target domains. The information available in the source domain contains  $N_s$  sequences of length  $L$ . Each sequence captures RGB-D frames  $\{I_{s,i}, D_{s,i}\}_{i=1}^L$ , and pose information  $\{T_{s,i}\}_{i=1}^L$ . The source domain is typically a simulated environment where the measurements are noiseless sensory information. The information available in the target domain is a list of  $N_t$  RGB images  $\{I_{t,i}\}_{i=1}^{N_t}$ , without any specific order. The target domain is typically a real-world environment, and the available data are RGB camera images taken from different real-world scenes (not sequences). Using the method outlined in Section 3.1, we can optimise a mapping function  $\mathcal{F}$  to obtain pose invariant representations for the source domain which have desirable characteristics for estimating the camera pose. For convenience, we denote the concatenation of these representations as  $\bar{\ell} \sim p_{\mathcal{F}}(\bar{\ell} | \mathcal{I}_s)$ , where  $\mathcal{I}_s$  denotes a random variable of an RGB image from the source domain.

Alternatively, we can view our source and target data through a careful assumption that they share a hidden variable  $\mathcal{L}$  which describes a property that underlies both domains (*i.e.*  $I_s \sim p_s(\mathcal{I}_s | \mathcal{L})$  and  $I_t \sim p_t(\mathcal{I}_t | \mathcal{L})$ ). For the case of indoor scenes,  $\mathcal{L}$  could represent the geometry (shape information) of a scene upon which the texture is conditioned, and either a simulated or real image can be sampled.

Without performing any adaptation, the representations  $\bar{\ell} \sim p_{\mathcal{F}}(\bar{\ell} | \mathcal{I}_t)$  will not be effective for localising on the target domain, as they were optimised to operate on the source domain. Our main objective is to obtain representations for the target domain so that they can be used for localisation in a real-world setting. The goal is to find a procedure that

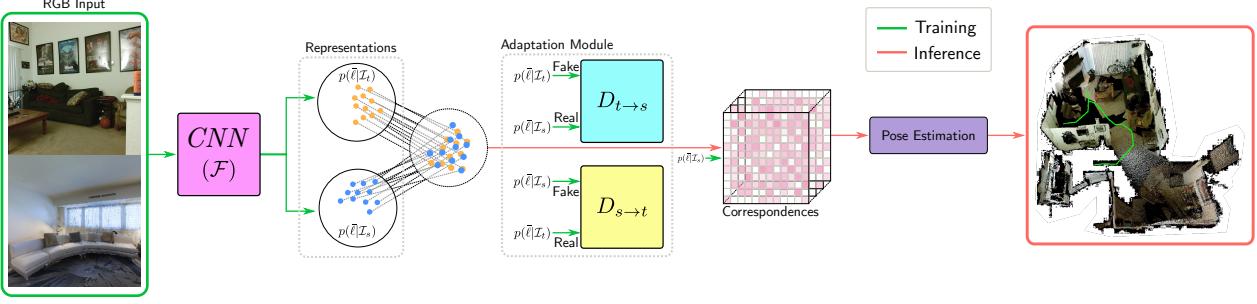


Figure 2: The BAA system is optimised by learning representations which are effective for localising in a source domain and simultaneously applies a symmetrical adversarial loss on both source and target domains. The convolutional neural network ( $\mathcal{F}$ ) employs EMP-Net [2] for extracting representations used for pose estimation, while concurrently acting as a generator in an adversarial setup. BAA’s adaptation module uses a dual discriminator setup,  $D_{s \rightarrow t}$  and  $D_{t \rightarrow s}$ , which have complementing objectives. The representations are fed in a swapped order to each of the discriminators. This setup forces the generator ( $\mathcal{F}$ ) to conceal each of the domain representations into a shared domain. In the training phase, source domain information is available as RGB-D trajectories with ground-truth poses for learning representations that are effective for pose estimation. The adaptation module receives RGB images from both source domain and target domain and pushes both the learned source representations  $p(\bar{\ell}|\mathcal{I}_s)$  and target representations  $p(\bar{\ell}|\mathcal{I}_t)$  to a shared domain, which is invariant to both domains. At inference stage, given a trajectory of a RGB sequence from the target domain, the representations from the shared domain are used to perform pose estimation.

re-optimises the mapping function  $\mathcal{F}$  so that when provided with an image  $\mathcal{I}_t$  from the target domain, the extracted representations will exhibit the desired attributes outlined in Section 3.

#### 4.2. Conditional Dual Adversarial Adaptation

The mapping function  $\mathcal{F}$  is initially optimised to produce representations for source domain data (*i.e.*  $\mathcal{I}_s$ ). Our objective is to re-optimize our mapping function  $\mathcal{F}$  which provides representations invariant to the domain of the input. Given both source and target distributions:

$$\mathcal{I}_s \sim p_s(\mathcal{I}_s | \mathcal{L}), \mathcal{I}_t \sim p_t(\mathcal{I}_t | \mathcal{L}) \quad (2)$$

and a mapping function which emits the two distributions:

$$\bar{\ell}_s \sim p_{\mathcal{F}}(\bar{\ell} | \mathcal{I}_s), \bar{\ell}_t \sim p_{\mathcal{F}}(\bar{\ell} | \mathcal{I}_t) \quad (3)$$

we use the dual adversarial objective proposed by [39]:

$$\begin{aligned} \min_{\mathcal{F}} \max_{D_1, D_2} V(\mathcal{F}, D_1, D_2) = & \alpha E_{\bar{\ell} \sim p_{\bar{\ell} | \mathcal{I}_s}} [\log D_1(\bar{\ell})] \\ & + E_{\bar{\ell} \sim p_{\bar{\ell} | \mathcal{I}_t}} [-D_1(\bar{\ell})] \\ & + E_{\bar{\ell} \sim p_{\bar{\ell} | \mathcal{I}_s}} [-D_2(\bar{\ell})] \\ & + \beta E_{\bar{\ell} \sim p_{\bar{\ell} | \mathcal{I}_t}} [\log D_2(\bar{\ell})] \end{aligned} \quad (4)$$

which is equivalent to optimising our mapping function  $\mathcal{F}$  when both discriminators are trained to convergence,  $D_1^*, D_2^*$ :

$$\begin{aligned} \min_{\mathcal{F}} V(\mathcal{F}, D_1^*, D_2^*) = & \alpha(\log \alpha - 1) + \beta(\log \beta - 1) \\ & \alpha D_{KL}(p(\bar{\ell} | \mathcal{I}_s) || p(\bar{\ell} | \mathcal{I}_t)) + \beta D_{KL}(p(\bar{\ell} | \mathcal{I}_t) || p(\bar{\ell} | \mathcal{I}_s)) \end{aligned} \quad (5)$$

where we have two  $KL$ -divergence terms which allow us to “push” the distributions in both directions and the applied force can be controlled using the  $\alpha, \beta$  terms. The mapping function will minimise the objective  $\min_{\mathcal{F}} V \iff p(\bar{\ell} | \mathcal{I}_s) = p(\bar{\ell} | \mathcal{I}_t)$ , so that  $\bar{\ell}$  is invariant to the domain in which it was originated from.

Given that the model uses an additional regularisation term for the mapping function  $\mathcal{F}$  which optimises the representations for localisation in the source domain data (see Section 4.3) where we have labelled information, we can use  $\alpha$  and  $\beta$  to dictate which  $KL$ -divergence is activated in the optimisation process. This encourages the mapping function  $\mathcal{F}$  to conceal the non-mutual information between domains whilst retaining their mutual information (*i.e.* the hidden variable  $\mathcal{L}$  useful for the task). For localisation, retaining the shape information and discarding the texture can potentially impact performance in the simulation; however, this also allows shape features to be extracted from the real domain so that  $\mathcal{F}$  can operate invariantly on both domains.

#### 4.3. Symmetrical Adversarial Architecture

The mapping function  $\mathcal{F}$  is a pretrained EMP-Net [2] using the loss function in Eq. 1, trained on RGB-D sequences from the source domain. In this work, we slightly diverge from the original method outlined in [2]. The input to the EMP-Net CNN is an RGB image and not RGB-D. The depth is still required for applying the loss defined in Eq. 1 on the CNN but it is not used for inferring representations from a given RGB image. This choice of input aligns well with the available information contained in the target

domain.

Provided images from the source domain,  $\{I_{s,i}\}_{i=0}^{LN_s}$ , and target domain,  $\{I_{t,i}\}_{i=0}^{N_t}$ , we define a dual adversarial optimisation scheme which minimises each domain representation  $p(\ell|\mathcal{I}_s)$  and  $p(\ell|\mathcal{I}_t)$  towards the shared domain representation. As seen in Fig. 2, there are three main components in our system: the EMP-Net CNN, which defines the initial mapping function  $\mathcal{F}$  and operates as a generator in a GAN setup and two discriminators  $D_{s \rightarrow t}$  and  $D_{t \rightarrow s}$ , where each discriminator is responsible for pulling the generator towards the shared representation distribution  $\ell \sim p(\ell|\mathcal{I}_s) = p(\ell|\mathcal{I}_t)$ . The loss applied to  $D_{t \rightarrow s}$ , between the target and source domains is defined as:

$$V_{D_{t \rightarrow s}} = \frac{1}{B} \sum_{i=1}^B [\alpha \log D_{t \rightarrow s}(\mathcal{F}(I_{s,i})) - D_{t \rightarrow s}(\mathcal{F}(I_{t,i}))] \quad (6)$$

Similarly, the loss applied to  $D_{s \rightarrow t}$ , between the source and target domain is defined as:

$$V_{D_{s \rightarrow t}} = \frac{1}{B} \sum_{i=1}^B [\beta \log D_{s \rightarrow t}(\mathcal{F}(I_{t,i})) - D_{s \rightarrow t}(\mathcal{F}(I_{s,i}))] \quad (7)$$

The adversarial loss on the mapping function  $\mathcal{F}$  is the loss we would apply to a generator in a dual discriminator GAN setup:

$$\begin{aligned} GEN_{loss} = & \frac{1}{B} \sum_{i=1}^B [\alpha \log D_{t \rightarrow s}(\mathcal{F}(I_{s,i})) \\ & + \beta \log D_{s \rightarrow t}(\mathcal{F}(I_{t,i})) \\ & - D_{t \rightarrow s}(\mathcal{F}(I_{t,i})) \\ & - D_{s \rightarrow t}(\mathcal{F}(I_{s,i}))] \end{aligned} \quad (8)$$

where  $B$  is the batch size. The final loss applied to the mapping function  $\mathcal{F}$  is the adversarial loss from Eq. 8, regularised by the original cross-entropy loss from Eq. 1:

$$V_{\mathcal{F}} = GEN_{loss} + \lambda_{CE} CE_{loss} \quad (9)$$

where  $\lambda_{CE}$  is a regularisation term defined in Section 5.

## 5. Experiments

We perform extensive experiments for evaluating our method. This includes assessing the attributes of the adapted representations, and performing localisation on representations that have been optimised for a simulation environment and adapted to a real environment. Finally, we ablate various adaptation methods which show the effectiveness of our symmetrical adversarial approach.

### 5.1. Datasets

The simulation/source environments which were chosen for this work both use the Habitat [50] simulation environment. Habitat provides access to RGB and Depth sensors which are perfectly aligned, along with odometry. The size of the RGB and Depth images can be specified in the simulation environment, which are both set to be  $256 \times 256$ .

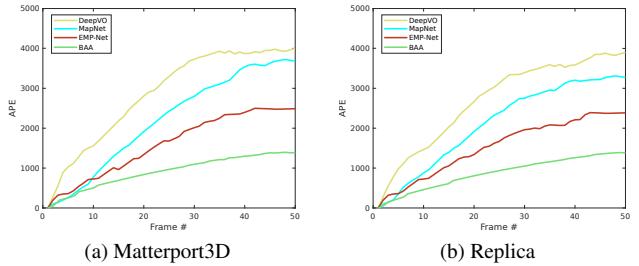


Figure 3: Performance of BAA and baseline models as indicated by Average Positional Error (APE) over different sequence lengths (up to 50 frames) on the Active Vision dataset when trained on (a) Matterport3D and (b) Replica datasets. Our BAA approach offers a significant improvement over the compared baselines when no label information from the target dataset is given.

**Matterport3D** The Matterport3D (MP3D) dataset [6] contains 90 different scenes with varying navigable space. There are 58 scenes in the training split which contain 4.8M trajectories of variable length (normally above 5 frames) which is publicly available<sup>1</sup> and is part of the point-goal navigation task [50]. From those 4.8M trajectories, we randomly sampled 50k uniformly distributed trajectories across the scenes and randomly chose slices of 5 consecutive frames from within those 50k trajectories. This results in 200k trajectories of length 5 used for training. For the test set, we use the test split which consists of 18 scenes and 1008 combined trajectories for all the scenes. From those, we sample 3 trajectories per test scene which has a trajectory length of at least 50 frames.

**Replica** The Replica dataset [52] contains 12 different scenes with an additional 6 scenes (“FRL apartment”) where the same scene is used, but the layout is changed. This totals to 18 scenes with a smaller amount of navigable space per scene on average when compared to MP3D [6]. For this dataset there were no publicly available point-goal task navigation trajectories; as part of this work, trajectories were created. We took 11 out of the 12 scenes (excluding

<sup>1</sup><https://github.com/facebookresearch/habitat-api>

| Model       | Matterport3D [6] |               |              | Replica [52] |               |              |
|-------------|------------------|---------------|--------------|--------------|---------------|--------------|
|             | APE-5            | APE-50        | ATE-50       | APE-5        | APE-50        | ATE-50       |
| DeepVO [60] | 1106.3           | 4011.5        | 1348.8       | 904.1        | 3861.2        | 1289.9       |
| MapNet [21] | 208.1            | 3669.1        | 1050.2       | 310.4        | 3288.0        | 933.3        |
| EMP-Net [2] | 335.8            | 2509.7        | 841.3        | 304.1        | 2336.9        | 801.8        |
| BAA (Ours)  | <b>245.1</b>     | <b>1387.2</b> | <b>475.4</b> | <b>221.6</b> | <b>1345.8</b> | <b>470.1</b> |

Table 1: Domain Generalisation when training on the Matterport3D and Replica source datasets and evaluated on the Active Vision target dataset using Average Position Error (APE) and Absolute Trajectory Error (ATE).

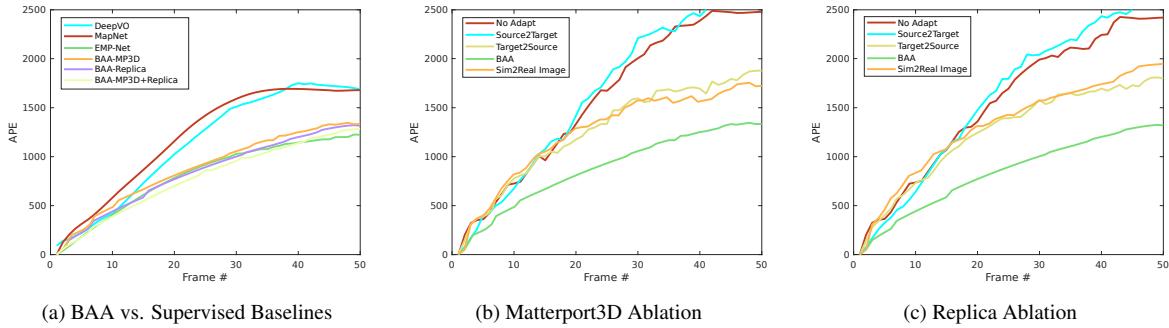


Figure 4: (a) Performance of BAA and baseline models as indicated by Average Positional Error (APE) over different sequence lengths (up to 50 frames) on the Active Vision dataset. Each competing baseline model was trained under supervision using labels from the target domain. Note the performance of our BAA approach which is able to outperform all supervised baseline models except the model in [2], without *any label information* from the target domain. (b) Ablation performance across different adaptation schemes as indicated by Average Positional Error (APE) over different sequence lengths (up to 50 frames) on the Active Vision dataset when trained on Matterport3D (**left**) and Replica (**right**) datasets. Our BAA approach significantly outperforms other adaptation approaches, including the image adaptation approach of [26].

“office 1” due to size) and randomly generated 50k trajectories of variable length, uniformly distributed among the scenes. A single trajectory was generated by randomly selecting two navigable points, computing the shortest path and recording the agent’s traversal. Following that, similar to the procedure that was performed for MP3D, we randomly chose slices of 5 consecutive frames and construct 200k trajectories of length 5. For the test set, we use the 6 scenes of the “FRL apartment” and follow a similar procedure of selecting two random navigable points and recording the agent’s traversal, but impose a minimal length of 50 frames on a sequence. As result, 50 test trajectories are recorded with a length of 50 frames per trajectory.

**Active Vision** The Active Vision Dataset (AVD) [1] is chosen as the real/target domain and consists of 20 indoor scenes. We use the same test (“Home 001 1” and “Home 001 2”) and train (the rest of scenes) splits as in [21, 2] except that for training we do not generate trajectories. Instead, we use the entire set of images from each scene as images which represent the target domain. This amounts to

a little less than 23k images. For the test set, we use the two test scenes and generate 50 random trajectories of 50 frames each. The RGB images are downsampled from  $1920 \times 1080$  by a factor of 4 and then center cropped to obtain images of  $256 \times 256$ . The depth information is not used, as it is only needed for the source domain.

## 5.2. Implementation

**Localisation Network Architecture** To extract feature embeddings, we employ the same CNN architecture as [2] which is based off the U-Net architecture [49], initialised using the approach in [19]. The U-Net architecture consists of an encoder and decoder network where the encoder network contains U-Net blocks and U-Net downsampling blocks. Each U-Net block consists of a repeated sequence of a  $3 \times 3$  convolution layer, followed by BatchNorm [27] and ReLU activation. For each U-Net downsampling block, the U-Net block is used where the first  $3 \times 3$  convolution layer in the block is modified to use 2-strided convolutions to downsample the input. The number of output channels is upsampled with each U-Net downsampling block by a

| Model                       | Matterport3D [6] |               |              | Replica [52] |               |              |
|-----------------------------|------------------|---------------|--------------|--------------|---------------|--------------|
|                             | APE-5            | APE-50        | ATE-50       | APE-5        | APE-50        | ATE-50       |
| EMP-Net [2] (No Adaptation) | 335.8            | 2509.7        | 841.3        | 304.1        | 2336.9        | 801.8        |
| EMP-Net [2] (Sim2Real [26]) | 375.2            | 1707.3        | 699.7        | 437.2        | 1967.6        | 789.2        |
| EMP-Net (S2T)               | 375.1            | 2656.2        | 877.7        | 314.2        | 2669.1        | 886.9        |
| EMP-Net (T2S)               | 361.3            | 1925.3        | 752.3        | 402.1        | 1819.6        | 724.2        |
| BAA (Ours)                  | <b>245.1</b>     | <b>1387.2</b> | <b>475.4</b> | <b>221.6</b> | <b>1345.8</b> | <b>470.1</b> |

Table 2: Ablation of different adaptation schemes on the EMP-Net baseline trained on (a) Matterport3D and (b) Replica datasets and evaluated using Average Position Error (APE) and Absolute Trajectory Error (ATE). Our symmetrical adaptation approach significantly outperforms other adaptation approaches, including the image adaptation approach of [26].

factor of 2 starting with a base output channel of 8 and upsampled to 128 output channels. For the decoder, we used a U-Net upsampling block, which is a sequence consisting of a  $2 \times 2$  transposed convolution with stride 2 followed by a U-Net block. The output of each U-Net upsampling block is then concatenated to its matching output from the encoder network. The localisation network also functions as the generator in our GAN setup where the choice of loss is consistent with the objective function defined in Eq. 9.

**GAN architecture** Our GAN setup consists of a set of dual discriminator networks paired with a generator network (which also serves as our representation extractor). Each discriminator network uses the architecture of [45], which consists of 2-strided  $5 \times 5$  convolution layers with Leaky ReLU activation to downsample our input RGB images from  $256 \times 256$  to  $4 \times 4$  in height and width. Following the setup of [39], we use Batch Normalisation [27] after each convolutional layer in both discriminators (with the exception of the first layer). Our choice of adversarial loss is consistent with the objective functions defined in Eqs. 6 & 7.

**Training Settings** For training the localisation network, we follow the settings in [2], training with batch size of 16, where each sample in the batch in a sequence of 5 consecutive RGB frames. RGB and depth data were normalised to lie between  $[0, 1]$ . We maintain the buffer size  $b$  of [2] to 4 frames, and extract 1024 feature representations for each observation frame. We use the ADAM optimiser [30], with first and second momentum terms of 0.5 and 0.999 values respectively. Across all experiments, we choose  $\lambda_{CE} = 0.1$ , which was found to be the optimal value via performing a random walk parameter sweep of  $\lambda_{CE}$  values in a range over  $[0.001, 10]$ . We set  $\alpha = 0.02$  and  $\beta = 0.04$ ; similarly these values were found through a random walk parameter sweep of  $\alpha$  and  $\beta$  values in a range over  $[0.001, 0.5]$ . These values of  $\alpha$  and  $\beta$  are consistent with what we would expect; since there is a regularising

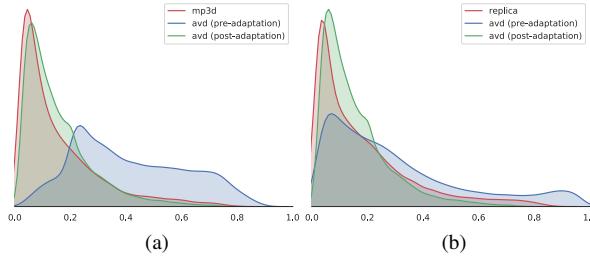


Figure 5: Histogram plots of the mutual exclusiveness property between representations. This property is important for generating unique descriptors which are effective for finding correspondences. The histogram of a pretrained EMP-Net [2] on two simulation environments (MP3D and Replica) can be seen in (a) and (b) respectively, along with pre-adaptation and post-adaptation histograms to AVD.

term  $L_{CE}$  (Eq. 1) which affects  $p(\bar{\ell}|\mathcal{I}_s)$  from the source domain, in order to *balance* the adaptation, a larger  $\beta$  (attracting force towards the target domain) was found to be more effective.

We first pretrain our localisation network with a learning rate of  $10^{-3}$  for 10 epochs on source domain data before adaptation. For adaptation, we train with a batch size of 32. We employ the Two-Time Update Rule (TTUR) [22], using a 3:1 ratio in learning rates between the discriminator and generator networks of  $3e-4$  and  $1e-4$  respectively. In this GAN setup, our pretrained localisation network acts as the generator network which is concurrently trained with both discriminator networks for 10 epochs on the target domain data.

### 5.3. Adapting Properties of Representations

An important property that the adapted representations need to maintain is uniqueness; or as mentioned in Section 3, in a given image every two representation vectors should be mutually exclusive. In this experiment, we

test whether this property is preserved after performing the adaptation on a target domain.

Generally, for a single image, this is done by taking the negative pair-wise distance between the representations, performing a Softmax operation and measuring the average value for each representation *w.r.t* every other representation. Ideally, this value should be 0. This process can be repeated for multiple images, which allows us to accumulate and produce a histogram of the aforementioned property. Four experiments are shown in Figs. 5a & 5b, where the source domains are MP3D and Replica respectively. The histogram for both source domains is shown, and we observe that the mutual exclusiveness property is preserved, similar to [2]. Furthermore, the histograms for the target domain, AVD, are shown for MP3D (Fig. 5a) and Replica (Fig. 5b) before performing our method (AVD pre-adaptation) and after (AVD post-adaptation). The effectiveness of our approach can be seen as the mutual exclusiveness property between representations is more concentrated near 0 post-adaptation in relation to pre-adaptation where it is significantly weaker.

#### 5.4. Evaluation on Real-World Data

**Training on Simulation and Evaluating on Real** In Fig. 3 and Tab. 1, we compare against the methods from [60], [21] and [2] when trained on source domain simulation data from Matterport3D and Replica and evaluated on target domain real-world data from AVD. We can observe that without label information from the target domain, our BAA method substantially outperforms competing baselines.

**Evaluating against Fully-Supervised methods** In Fig. 4a and Table 3, we compare our BAA method against the methods of [60], [21] and [2] when each competing method is trained in a fully-supervised manner on training data from the target domain. Additionally, to evaluate the performance of our method against a pure geometrical approach, we compare to [37], where we show that we are able to significantly outperform ORB-SLAM2 on the AVD dataset. Note that in this case, we show the *unsupervised* performance of our method on the target domain (our BAA model is not given target domain labels). We can see that our BAA method compares very favourably against the fully-supervised method, outperforming the supervised methods of [60] and [21] and obtaining competitive results with the supervised method from [2]. This result highlights the ability to leverage noiseless depth measurements provided in the simulation environment for learning representations on real-world RGB measurements; in contrast to the fully supervised methods which rely on depth information that is noisy and comes from depth sensors. We reiterate that BAA *does not use depth information* as an

| AVD data [1]            | APE-5        | APE-50        | ATE-50       |
|-------------------------|--------------|---------------|--------------|
| ORB-SLAM2 (RGB-D) [37]  | 432.0        | 3090.0        | 794.0        |
| DeepVO [60]             | 220.0        | 1690.0        | 741.0        |
| MapNet [21]             | 312.3        | 1680.0        | 601.0        |
| EMP-Net [2]             | <b>181.6</b> | <b>1201.0</b> | <b>381.0</b> |
| BAA-MP3D (Ours)         | 245.1        | 1387.2        | 475.4        |
| BAA-Replica (Ours)      | 221.6        | 1345.8        | 470.1        |
| BAA-MP3D+Replica (Ours) | 185.2        | 1286.4        | 433.8        |

Table 3: Average Position Error (APE) and Absolute Trajectory Error (ATE) on the Active Vision Dataset. Note that all competing baselines have been trained using label information from the Active Vision Dataset, whilst our BAA approach has only been trained using source domain data (Matterport3D and Replica datasets) and adapted for localisation on the Active Vision Dataset.

input to our framework in the inference stage.

#### 5.5. Ablation of Different Adaptation Schemes

In Figs. 4b & 4c and Table 2, we show ablation results for different schemes adapting from Matterport3D and Replica to AVD. We observe that our BAA approach offers significant improvements in terms of APE and ATE performance than uni-directional adaptation approaches [51] and an image adaptation approach [26]. This experiment highlights the key intuition, that if two environments share geometrical similarities and differ in textural and lighting appearances, using a symmetrical adversarial loss encourages the generator to perform a decoupling of these two properties.

## 6. Conclusion

In this paper, we presented a domain adaptation approach which uses a symmetrical distance between source and target domains for adapting representations between domains. We show that by adapting directly on localisation features optimised on simulation data and using our symmetrical adaptation approach, we can gain large improvements in localisation performance on real-world data without any label information from the target domain. We show that our approach is not only effective in an unsupervised learning setting, but can achieve very favourable results when compared to supervised methods on real-world data.

**Acknowledgments** The authors would like to thank the anonymous reviewers for their useful and constructive comments. This work was supported by the Australian Research Council Centre of Excellence for Robotic Vision (project number CE1401000016).

## References

- [1] P. Ammirato, P. Poirson, E. Park, J. Košecká, and A. C. Berg. A dataset for developing and benchmarking active vision. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1378–1385. IEEE, 2017. [6](#), [8](#)
- [2] G. Avraham, Y. Zuo, T. Dharmasiri, and T. Drummond. Empnet: Neural localisation and mapping using embedded memory points. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 8120–8129, 2019. [2](#), [3](#), [4](#), [6](#), [7](#), [8](#)
- [3] M. Bloesch, J. Czarnowski, R. Clark, S. Leutenegger, and A. J. Davison. Codeslam—learning a compact, optimisable representation for dense visual slam. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2560–2568, 2018. [2](#)
- [4] K. Bousmalis, G. Trigeorgis, N. Silberman, D. Krishnan, and D. Erhan. Domain separation networks. In *Advances in neural information processing systems*, pages 343–351, 2016. [2](#)
- [5] L. Bruzzone and M. Marconcini. Domain adaptation problems: A dasvm classification technique and a circular validation strategy. *IEEE transactions on pattern analysis and machine intelligence*, 32(5):770–787, 2009. [2](#)
- [6] A. Chang, A. Dai, T. Funkhouser, M. Halber, M. Niessner, M. Savva, S. Song, A. Zeng, and Y. Zhang. Matterport3d: Learning from rgbd data in indoor environments. *arXiv preprint arXiv:1709.06158*, 2017. [5](#), [6](#), [7](#)
- [7] W.-S. Chu, F. De la Torre, and J. F. Cohn. Selective transfer machine for personalized facial action unit detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3515–3522, 2013. [2](#)
- [8] R. Clark, S. Wang, A. Markham, N. Trigoni, and H. Wen. Vidloc: A deep spatio-temporal model for 6-dof video-clip relocalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6856–6864, 2017. [2](#)
- [9] R. Clark, S. Wang, H. Wen, A. Markham, and N. Trigoni. Vinet: Visual-inertial odometry as a sequence-to-sequence learning problem. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017. [1](#), [2](#)
- [10] G. Costante, M. Mancini, P. Valigi, and T. A. Ciarruglia. Exploring representation learning with cnns for frame-to-frame ego-motion estimation. *IEEE robotics and automation letters*, 1(1):18–25, 2015. [2](#)
- [11] H. Daumé III. Frustratingly easy domain adaptation. *arXiv preprint arXiv:0907.1815*, 2009. [2](#)
- [12] T. Dharmasiri, A. Spek, and T. Drummond. Eng: End-to-end neural geometry for robust depth and pose estimation using cnns. In *Asian Conference on Computer Vision*, pages 625–642. Springer, 2018. [2](#)
- [13] Y. Ganin and V. Lempitsky. Unsupervised domain adaptation by backpropagation. *arXiv preprint arXiv:1409.7495*, 2014. [2](#)
- [14] M. Gheisari and M. S. Baghshah. Unsupervised domain adaptation via representation learning and adaptive classifier learning. *Neurocomputing*, 165:300–311, 2015. [2](#)
- [15] B. Gong, K. Grauman, and F. Sha. Connecting the dots with landmarks: Discriminatively learning domain-invariant features for unsupervised domain adaptation. In *International Conference on Machine Learning*, pages 222–230, 2013. [2](#)
- [16] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014. [3](#)
- [17] M. A. Goodrich, A. C. Schultz, et al. Human–robot interaction: a survey. *Foundations and Trends® in Human–Computer Interaction*, 1(3):203–275, 2008. [1](#)
- [18] S. Gupta, J. Davidson, S. Levine, R. Sukthankar, and J. Malik. Cognitive mapping and planning for visual navigation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2616–2625, 2017. [2](#)
- [19] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015. [6](#)
- [20] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. [1](#)
- [21] J. F. Henriques and A. Vedaldi. Mapnet: An allocentric spatial memory for mapping environments. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8476–8484, 2018. [2](#), [6](#), [8](#)
- [22] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*, pages 6626–6637, 2017. [7](#)

- [23] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 2
- [24] J. Hoffman, E. Tzeng, J. Donahue, Y. Jia, K. Saenko, and T. Darrell. One-shot adaptation of supervised deep convolutional models. *arXiv preprint arXiv:1312.6204*, 2013. 2
- [25] J. Hoffman, E. Tzeng, T. Park, J.-Y. Zhu, P. Isola, K. Saenko, A. A. Efros, and T. Darrell. Cycada: Cycle-consistent adversarial domain adaptation. *arXiv preprint arXiv:1711.03213*, 2017. 3
- [26] X. Huang, M.-Y. Liu, S. Belongie, and J. Kautz. Multimodal unsupervised image-to-image translation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 172–189, 2018. 6, 7, 8
- [27] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015. 6, 7
- [28] Y. Ke, R. Sukthankar, et al. Pca-sift: A more distinctive representation for local image descriptors. *CVPR* (2), 4:506–513, 2004. 3
- [29] A. Kendall, M. Grimes, and R. Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *Proceedings of the IEEE international conference on computer vision*, pages 2938–2946, 2015. 2
- [30] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 7
- [31] S. Li, X. Wang, Y. Cao, F. Xue, Z. Yan, and H. Zha. Self-supervised deep visual odometry with online adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6339–6348, 2020. 2
- [32] M.-Y. Liu and O. Tuzel. Coupled generative adversarial networks. In *Advances in neural information processing systems*, pages 469–477, 2016. 3
- [33] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, and F. E. Alsaadi. A survey of deep neural network architectures and their applications. *Neurocomputing*, 234:11–26, 2017. 1
- [34] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015. 1
- [35] M. Long, Y. Cao, J. Wang, and M. I. Jordan. Learning transferable features with deep adaptation networks. *arXiv preprint arXiv:1502.02791*, 2015. 2
- [36] R. Mahjourian, M. Wicke, and A. Angelova. Unsupervised learning of depth and ego-motion from monocular video using 3d geometric constraints. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5667–5675, 2018. 2
- [37] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos. Orb-slam: a versatile and accurate monocular slam system. *IEEE transactions on robotics*, 31(5):1147–1163, 2015. 8
- [38] H. V. Nguyen, H. T. Ho, V. M. Patel, and R. Chellappa. Dash-n: Joint hierarchical domain adaptation and feature learning. *IEEE Transactions on Image Processing*, 24(12):5479–5491, 2015. 2
- [39] T. Nguyen, T. Le, H. Vu, and D. Phung. Dual discriminator generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2670–2680, 2017. 4, 7
- [40] S. Pachori, A. Deshpande, and S. Raman. Hashing in the zero shot framework with domain adaptation. *Neurocomputing*, 275:2137–2149, 2018. 2
- [41] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang. Domain adaptation via transfer component analysis. *IEEE Transactions on Neural Networks*, 22(2):199–210, 2010. 2
- [42] S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2009. 2
- [43] E. Parisotto and R. Salakhutdinov. Neural map: Structured memory for deep reinforcement learning. *arXiv preprint arXiv:1702.08360*, 2017. 2
- [44] K.-C. Peng, Z. Wu, and J. Ernst. Zero-shot deep domain adaptation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 764–781, 2018. 3
- [45] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015. 7
- [46] A. Raj, V. P. Namboodiri, and T. Tuytelaars. Subspace alignment based domain adaptation for rcnn detector. *arXiv preprint arXiv:1507.05578*, 2015. 2
- [47] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016. 1
- [48] S. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee. Generative adversarial text to image synthesis. *arXiv preprint arXiv:1605.05396*, 2016. 3

- [49] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. [6](#)
- [50] M. Savva, A. Kadian, O. Maksymets, Y. Zhao, E. Wijmans, B. Jain, J. Straub, J. Liu, V. Koltun, J. Malik, et al. Habitat: A platform for embodied ai research. *arXiv preprint arXiv:1904.01201*, 2019. [5](#)
- [51] J. Shen, Y. Qu, W. Zhang, and Y. Yu. Adversarial representation learning for domain adaptation. *stat*, 1050:5, 2017. [8](#)
- [52] J. Straub, T. Whelan, L. Ma, Y. Chen, E. Wijmans, S. Green, J. J. Engel, R. Mur-Artal, C. Ren, S. Verma, et al. The replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019. [5](#), [6](#), [7](#)
- [53] R. Szeliski. *Computer vision: algorithms and applications*. Springer Science & Business Media, 2010. [3](#)
- [54] Y. Taigman, A. Polyak, and L. Wolf. Unsupervised cross-domain image generation. *arXiv preprint arXiv:1611.02200*, 2016. [3](#)
- [55] S. Thrun. Toward a framework for human-robot interaction. *Human-Computer Interaction*, 19(1):9–24, 2004. [1](#)
- [56] E. Tzeng, C. Devin, J. Hoffman, C. Finn, P. Abbeel, S. Levine, K. Saenko, and T. Darrell. Adapting deep visuomotor representations with weak pairwise constraints. *arXiv preprint arXiv:1511.07111*, 2015. [3](#)
- [57] E. Tzeng, J. Hoffman, T. Darrell, and K. Saenko. Simultaneous deep transfer across domains and tasks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4068–4076, 2015. [2](#)
- [58] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7167–7176, 2017. [3](#)
- [59] M. Wang and W. Deng. Deep visual domain adaptation: A survey. *Neurocomputing*, 312:135–153, 2018. [1](#)
- [60] S. Wang, R. Clark, H. Wen, and N. Trigoni. Deepvo: Towards end-to-end visual odometry with deep recurrent convolutional neural networks. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2043–2050. IEEE, 2017. [1](#), [2](#), [6](#), [8](#)
- [61] H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, and D. N. Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5907–5915, 2017. [3](#)
- [62] J. Zhang, L. Tai, J. Boedecker, W. Burgard, and M. Liu. Neural slam: Learning to explore with external memory. *arXiv preprint arXiv:1706.09520*, 2017. [2](#)
- [63] L. Zhang, Z. He, and Y. Liu. Deep object recognition across domains based on adaptive extreme learning machine. *Neurocomputing*, 239:194–203, 2017. [2](#)
- [64] H. Zhou, B. Ummenhofer, and T. Brox. Deeptam: Deep tracking and mapping. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 822–838, 2018. [2](#)
- [65] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017. [3](#)