Self-Supervised Domain Adaptation for Visual Navigation with Global Map Consistency

Eun Sun Lee Junho Kim Young Min Kim
Dept. of Electrical and Computer Engineering, Seoul National University, Korea
{ eunsunlee, 82magnolia, youngmin.kim } @snu.ac.kr

Abstract

We propose a light-weight, self-supervised adaptation for a visual navigation agent to generalize to unseen environment. Given an embodied agent trained in a noiseless environment, our objective is to transfer the agent to a noisy environment where actuation and odometry sensor noise is present. Our method encourages the agent to maximize the consistency between the global maps generated at different time steps in a round-trip trajectory. The proposed task is completely self-supervised, not requiring any supervision from ground-truth pose data or explicit noise model. In addition, optimization of the task objective is extremely light-weight, as training terminates within a few minutes on a commodity GPU. Our experiments show that the proposed task helps the agent to successfully transfer to new, noisy environments. The transferred agent exhibits improved localization and mapping accuracy, further leading to enhanced performance in downstream visual navigation tasks. Moreover, we demonstrate test-time adaptation with our self-supervised task to show its potential applicability in real-world deployment.

1. Introduction

Vision-based navigation is a fundamental task for embodied AI agents that traverse the environment while conducting a series of actions. The agent needs to plan and execute a sequence of movements based on its perception of the surrounding area to interact with the environment in a desirable way. Training an agent in the real world is costly and thus, many research has made progress by leveraging simulators [40, 42] utilizing 3D scene datasets [6, 47]. While recent works demonstrate a nearly perfect performance with an agent trained in a noiseless simulator [46], the agent is bound to experience a critical decrease in its performance when deployed in a noisy, realistic environment [26]. One may try to close the domain gap by fine-tuning the policy in a new, noisy environment [7, 26], but it is difficult to obtain ground truth labels or poses in a real-world deployment. Furthermore, it is impossible to exactly model the various actuation and odometry sensor noise given the unlimited combinations of motors and sensors for custom robots deployed in various spatial layout and environmental configurations.

We investigate the scenario where the pre-trained model in a noiseless environment needs to be transferred to a new environment where the agent's actuator is noisy and the odometry measurement is also erroneous. As a mean to generalize in different environments, recent works propose modular approaches that separate the observation space and the action space [8, 22, 28], as shown in Figure 1. The observation module transfers the sensory perception into the domain-agnostic intermediate representation, and a subsequent policy module deduces the agent's action from the intermediate representation. Then it suffices to fine-tune the observation module to produce the stable intermediate representation given corrupted measurements. Still, adapting the observation module requires either ground-truth poses or explicit noise models, which are not available in our realistic set up.

To this end, we formulate a self-supervised learning task where the agent is encouraged to generate consistent global maps over a round-trip trajectory. The agent generates the first global map during the forward trajectory. After the turn point, the agent resets and generates a new map during the backward path from which we train the prediction of newly generated map. The self-supervision signal compares the consistency between the overlapping maps generated from different trajectories and fine-tunes the pre-trained navigation agent to the noisy real world. Our method does not require ground-truth poses as long as there exist overlapping observations of the global map. After the observation module is fine-tuned to suppress the errors caused by the domain shift, we can combine it with the pre-trained action policy to perform various downstream tasks.

We extensively evaluate our self-supervised learning approach in fundamental navigation tasks such as localization, mapping, and exploration [15] and demonstrate that our approach can quickly adapt the pre-trained agents to the new environment. Our proposed method finishes the adaptation process within 5 minutes, much faster than the domain randomization approach which converges after 65 minutes of fine-tuning. We further enhance the performance by data augmentation method utilizing random crops of trajectories. Lastly, we train our agent in a test-time adaptation setting to demonstrate the task's applicability in the real world.

In summary, our main contributions are as follows: i) we introduce a simple self-supervised task for robust visual navigation amidst actuation and sensor noise, ii) we demonstrate that the suggested formulation enhances the localization, mapping, and the final task performance, iii) we show that our method is applicable for test-time adaptation indicating the potential applicability for real-world deployment.

2. Related Work

Our work proposes a self-supervised approach to transfer an embodied agent trained in a noiseless environment to a noisy environment where actuation and odometry sensor noise is present. We thus discuss relevant literature from the field of visual navigation for embodied AI, Sim2Real, and self-supervised learning.

Visual Navigation In visual navigation for an embodied agent, the principal tasks are mapping and planning. Classical robotics approaches focus first on building a map of the environment [2, 18, 33] followed by path planning [30, 44, 49] on the map. Recent deep-learning based approaches, on the other hand, often train an intelligent agent jointly for mapping and planning [23, 38]. Such approaches have enabled to train the embodied agents for end-to-end tasks and adapt to a large number of challenging tasks including goal-oriented navigation [8, 45, 46], audio-visual navigation [13], or embodied Q&A [16].

To perform the various navigational tasks, the intelligent agents utilize diverse forms of memory systems to represent the observed environment. Neural networks can create implicit memory structure with LSTM or GRU [15, 32] or a topological graph that aligns with location information such as landmarks or image frames on the graph nodes [9, 14, 17, 39, 41]. However, the traditional map-based memories such as occupancy grid maps [7, 23, 28, 45] still perform competitively, especially for long-range navigation tasks. The explicit maps can also be easily extended into other map-based planning tasks and therefore widely deployed for visual navigation of embodied agents. We utilize the occupancy map as the memory representation and the mean to define the self-supervision task.

Sim2Real When training an agent in a real-world setting, it is difficult to collect a large amount of data with ground-truth labels. Simulators are great tools for training embodied agents before deploying in the real world. However, the policy learned from simulated environment often fails to maintain its performance in the real world setting [27]. Imitation learning observes the videos of desired trajectories collected by humans demonstration in the real environment [15, 16], but there still exists the gap between human and robot execution. Thus, bridging the domain gap has become a new direction of research [11, 24]. Inspired by the domain randomization approach in visual domain [24], the approach from [10] randomizes the dynamic corruptions such as actuation and odometry noise. However, we demonstrate that our self-supervision approach can better compensate for the unknown noise in terms of both accuracy and efficiency, by transforming the observations into an intermediate representation [12, 31].

Self-Supervised Learning The major success of deep learning owes to a large amount of training data. However, it is not scalable to collect the desired amount of labelled data for every task. Self-supervised learning proposes to obtain labels from known information, namely auxiliary tasks, and apply techniques of supervised learning to extract useful representation for desired tasks. Examples of such auxiliary tasks include predicting the random rotation of a given image or estimating depth and surface normals from observation frames [19, 21, 24]. With the careful design of auxiliary tasks, it has been demonstrated that the self-supervision can improve the performance of vision tasks such as image translation [35], or video object segmentation [25]. We suggest an orthogonal self-supervision task for various sensor noises, which could complement the aforementioned self-supervision for visual representation learning. While previous studies on indoor navigation suggest selecting more reliable information from multi-modal sensor data [3, 37], our proposed self-supervision does not utilize any additional sensors. Instead, we demonstrate that enforcing the consistency of the global maps is a useful auxiliary task to quickly adapt to the new sensor noise without ground truth labels.

3. Method

We introduce a simple and generic, self-supervised learning method for noise-robust visual navigation. Our objective is to transfer a pretrained agent to the novel, unseen environments where different types of actuation and odometry sensor noises are present. The proposed method enforces consistency on global maps generated over time in a round-trip trajectory. In this section, we explain the overall setup, the various types of actuation and sensor noise, and how the global map consistency is implemented. We also introduce our data augmentation method.

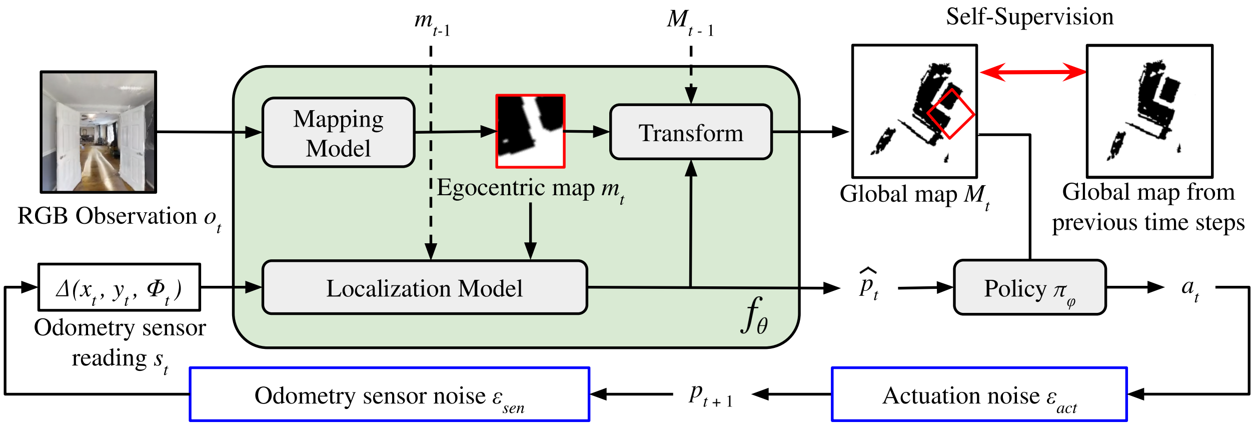


Figure 1. **Overview of our approach.** Our domain adaption builds on a navigation agent that is composed of the mapping and localization module f_θ (shaded in green) and the planning module π_ψ. Given an RGB observation o_t and the odometry sensor reading s_t, f_θ estimates the pose $\hat{p}_t$ and the global map M_t, from which π_ψ generates the task-specific policy to perform the action a_t. After the mapping model predicts the egocentric map m_t from the observation o_t, and the localization model estimates the agent's pose $\hat{p}_t$ by comparing the egocentric maps from two consecutive timesteps, m_{t-1} and m_t, in addition to reading the odometry sensor s_t. Then the generated map m_t is transformed based on the estimated pose $\hat{p}_t$ and fused with the previous global map M_{t-1} to update the global map M_t. In a noiseless simulator, the agent can learn the parameters of the module θ from ground-truth p_t and m_t. In a real-world deployment with unknown actuation and odometry noise (outlined in blue), the ground-truth supervision is not obtainable. Our self-supervised learning approach transfers the agent to the noisy environment by computing the proposed global map consistency loss for the global map M_t.

3.1. Visual Navigation with Spatial Memory

Our self-supervision works on the consistency of the global map, therefore it can be applied to most visual navigation agents that are based on spatial map memory [7, 8, 28, 38, 45, 50]. Such approaches train two neural networks for mapping and planning, which we will refer as f_θ and π_ψ respectively.

Figure 1 shows the overall flow of our approach. Given the current observation o_t and the odometry sensor measurement s_t, the mapping model predicts the egocentric map m_t and the localization model estimates the current position $\hat{p}_t$. Two models are jointly referred as the mapping and localization module f_θ, where the global map M_t is generated by combining m_t using $\hat{p}_t$. Note that the predictions from the previous time step, the estimated egocentric map m_{t-1} and global map M_{t-1}, are also utilized. Simply, we represent our mapping and localization module as

$$f_\theta(o_{1:t}, s_{1:t}) = M_t, \hat{p}_t. \tag{1}$$

For the purpose of proposition, we demonstrate the applicability using a sequence of RGB image observations for o_t as the input, and the 2D occupancy grid map for M_t and the 2D pose for $p_t = (x, y, \phi)$. Here (x, y) denote the 2D coordinate and ϕ indicates the 1D orientation.

Then the planning module π_ψ is implemented with a separate neural network, which is a policy network that observes the output of the mapping module and generates the action $a_t \in \mathcal{A}$

$$\pi_\psi(M_t, \hat{p}_t) = a_t. \tag{2}$$

We allow a discrete set of agent actions, specifically $\mathcal{A}$={Forward, Turn Right, Turn Left}.

Our supervision works on the intermediate representation between the two modules (f_θ and π_ψ), namely the consistency of M_t. Note that the consistency of M_t is tightly coupled with the accuracy of $\hat{p}_t$ because the pose $\hat{p}_t$ can be defined with respect to the generated map M_t and the network f_θ fuses the current observation o_t to M_t using the current pose $\hat{p}_t$. The formulation allows disentangling the variations in measurements from the policies designed for different downstream tasks. We can train a desired policy network π_ψ in an ideal simulation setup with the ground truth oracle and fine-tune only the observation module f_θ in a different environment set-up. More specifically, in our setup, we only fine-tune the localization part of the module as the egocentric map prediction is not disturbed by the odometry or actuation noise. Nevertheless, the proposed method is generic as it can be applied to various visual navigation models with distinct modules for perception and action sharing an intermediate representation.

Actuation and Odometry Sensor Noise We first train both networks in a noiseless setup and demonstrate the feasibility of the proposed self-supervision in realistic noise models. There are two main streams of expected variation; actuation and sensor noise. A large amount of both classical and latest work [20, 29, 40] has presented the approach to model realistic actuation and odometry noise. In this section, we briefly describe the noise models used as an un-

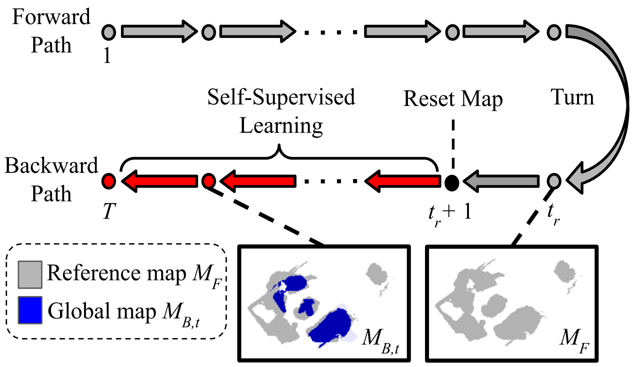


Figure 2. **Global Map Consistency from a Round Trip.** The forward path from the time 1 to t_r builds a reference map M_F. At $t_r + 1$, the agent resets the map and generates another map $M_{B,t}$ from $t_r + 1$ to T. At each step during the backward path, we impose the consistency between M_F and $M_{B,t}$.

seen environment for the pre-trained agent. An exhaustive discussion is available in [10].

The actuation noise disturbs an agent's movement. The agent fails to reach the intended position, and it becomes difficult to estimate the agent's pose solely by the control commands. The properties of actuation noise fall into two classes; motion bias, and motion drift. First, the motion bias adds a constant bias, $\delta_c \in \mathbb{R}^3$, and a stochastic bias, $\delta_s \in \mathbb{R}^3$ drawn from $\mathcal{N}(\mu_s, \Sigma_s)$. Then, the motion drift, $\alpha \in \mathbb{R}^3$, is added towards left or right only for the forward movement. The actuation noise is the addition of both motion bias and motion drift

$$\epsilon_{act} = \delta_c + \delta_s + \alpha. \tag{3}$$

The ground-truth pose of the agent after an action is defined as

$$(x_{t+1}, y_{t+1}, \phi_{t+1}) = (x_t, y_t, \phi_t) + (u_x, u_y, u_\phi) + \epsilon_{act}, \tag{4}$$

where the (u_x, u_y, u_ϕ) indicates the intended action control.

In addition, the odometry sensor noise affects the pose estimation accuracy, which further degrades the mapping accuracy and the navigation performance. The pose reading noise from odometry sensors is drawn from a Gaussian Mixture Model. The final pose reading is measured as

$$(x'_{t+1}, y'_{t+1}, \phi'_{t+1}) = (x_{t+1}, y_{t+1}, \phi_{t+1}) + k\epsilon_{sen}, \tag{5}$$

where the k denotes the sensor noise severity. The noise models are derived from the real-world observation, and we employ them based on the noise parameters modeled from actual physical deployment [1, 34] to test the performance of our algorithm in a noisy environment.

3.2. Self Supervision with Global Map Consistency

We assume deploying a pre-trained agent in an unseen environment, where the ground truth poses or the changes in the observation noise model are not known. Instead, we can create a supervision signal by enforcing consistency of the generated global map M_t. While we cannot assure that the map is error-free, we can assume that the error accumulates over time. This leads to incorporating more accurate pose data in generating the global map in the earlier steps compared to the ones generated later in a continuing trajectory.

To implement the self-supervised learning task efficiently, we deliberately design overlapping trajectories and drive an embodied agent in round trips as shown in Figure 2. The round trip is easily generated by executing any navigational task for a designated number of steps, turning around 180°, and executing the previous action sequence in reverse. The global map is first generated during the forward path, then the agent resets and generates another map from scratch during the backward path. In a noiseless setting, the agent observes the same area at each one-way trip. Therefore, the global map from its forward path should be equal to the global map generated during its backward path. When the actuation noise is present, the agent may not step on the same waypoints it has traversed during its forward path. Nonetheless, our proposed formulation is still valid as the agent generates maps from the overlapping area.

To summarize, during the forward path, the agent generates a reference global map M_F

$$f_\theta(o_{1:t_r}, s_{1:t_r}) = M_F, \hat{p}_F, \tag{6}$$

where t_r is the step when the robot turns around. On its way back, the agent generates a new map. However, the same global coordinate is shared across all global maps and the agent continues incremental pose estimation using the full odometry $s_{1:t}$. At each reverse step, a new global map, $M_{B,t}$, is predicted,

$$f_\theta(o_{t_r+1:t}, s_{1:t}) = M_{B,t}, \hat{p}_{B,t}, \quad t > t_r. \tag{7}$$

Lastly, our self-supervised task imposes the consistency between the predicted M_F and $M_{B,t}$ to fine-tune the mapping module f_θ under the new environment. In the equation below, T denotes the ending time of the trajectory and `stopgrad` indicates that no backpropagation should happen through M_F

$$\mathcal{L} = \sum_{t=t_r+1}^{T} \|\texttt{stopgrad}(M_F) - M_{B,t}\|_2. \tag{8}$$

Therefore we compare the estimated map at every step during the backward path, providing a rich set for loss computations per acquired trajectory. While there are several ways of finding the difference between the two occupancy grid maps, we incorporated the mean squared error (MSE) loss. The choice of losses is further discussed in the experiment section.

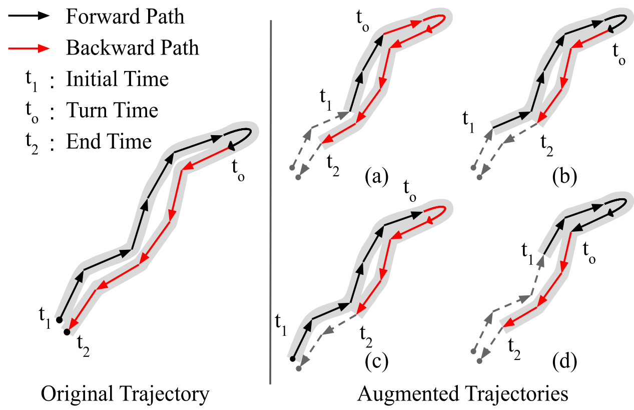


Figure 3. **Data Augmentation.** A round trip trajectory can be augmented by randomly sampling the three time steps, initial time, t_1, the turn time, t_o, and the end time, t_2, in a chronological order.

Data Augmentation While our vanilla formulation utilize the bisections of round-trip trajectories, the consistency can be enforced for any subsets of trajectories with overlapping observations. We introduce a data augmentation method based on random cropping which has been used in various types of data such as videos or images [24, 43]. Given a trajectory, we randomly sample three timesteps in a chronological order: the initial time t_1, the turn time t_o, and the end time t_2 ($t_1 < t_o < t_2$). A few example trajectories are shown in Figure 3. Then the reference map M'_F is produced from $f_\theta(o_{t_1:t_o}, s_{t_1:t_o})$ and compared to the map $M'_{B,t}$ generated from the backward step $f_\theta(o_{t_o+1:t}, s_{1:t})$, $t_o < t < t_2$. Our loss is similarly defined as

$$\mathcal{L} = \sum_{t=t_o+1}^{t_2} \|\texttt{stopgrad}(M'_F) - M'_{B,t}\|_2. \tag{9}$$

We analyze the effectiveness of the proposed data augmentation method in the experiment section.

4. Experiments

We extensively evaluate the performance of a pre-trained embodied agent when adapted to a new environment with our self-supervised learning task. In all our experiments we report the results using the agent from Active Neural SLAM [7], as it is a widely used baseline agent for visual navigation [8, 9, 38]. Nonetheless, our proposed self-supervised learning task is generic as it is easily applicable to most navigation models or agents based on spatial representation. For navigation task analysis, we follow the task setup proposed by [15]. We evaluate the navigation agent by the area coverage within a fixed number of steps, which is 1000 steps in all of our experiments. We show our result for exploration since it is the fundamental task for most navigation agents [7].

Method		NA	GT	DR	Ours
Localization	x, y (m)	0.15	0.02	0.12	0.03
	ϕ (°)	2.67	0.27	1.7	0.36
Mapping	MSE	0.25	0.17	0.23	0.17
Exploration	Ratio (%)	82.02	93.75	89.06	91.79
	Area (m^2)	26.36	31.83	29.19	30.79
Training	Time (min)	-	-	65	5

Table 1. Generalization to noisy environment evaluated on localization, mapping and exploration. Note that 'MSE' denotes 'mean squared error'.

We use Habitat [40] simulator in the scenes from Gibson dataset [48] and analyze on the standard train/val split [40] comparing against the ground truth poses and maps. For our self-supervised learning approach, we adapt the original model [7] implemented using PyTorch [36] to allow learning from global map consistency. The fine-tuning time of the self-supervised learning, accelerated with an RTX 2080 GPU, completes within 5 minutes, and the test-time adaptation finishes in 2.5 minutes. Additional details about the experimental setup are available in the supplementary material.

4.1. Task Adaptation to Noisy Environments

We first investigate if our proposed self-supervised learning task helps agents to transfer to a new, noisy environment. We pre-train an agent in a noiseless environment where the mapping and localization module f_θ is trained with the ground-truth pose and egocentric map. We then observe if our self-supervision can help the agent to generalize across various unseen noisy environments without ground-truth supervision. For unseen noisy environment, we apply the odometry noise and actuation noise models based on real data of LoCoBot [1] collected from the previous work [7, 27]. For each episode, we first report the median localization error in terms of translation (x, y) and rotation (ϕ), following [4, 5]. We additionally report the mean squared error (MSE) of the generated occupancy grid maps with respect to the ground-truth maps. To quantify exploration performance, we report two metrics, namely the absolute coverage area in m^2 and the ratio of area coverage, similar to [7].

As seen in Table 1, performance enhancement occurs in all evaluations. Our agent learns to better perceive the surroundings and explores 91.79% of its environment when tested for exploration. "No Adaptation (**NA**)" is the pre-trained model in a noiseless environment with ground-truth information, deployed in a new, noisy environment without any adaptation. Thus, the errors in NA reflect the performance degradation due to the domain gap and sets the lower bound of adaption. Our agent distinctively outperforms the baseline agent and compensates for the unknown

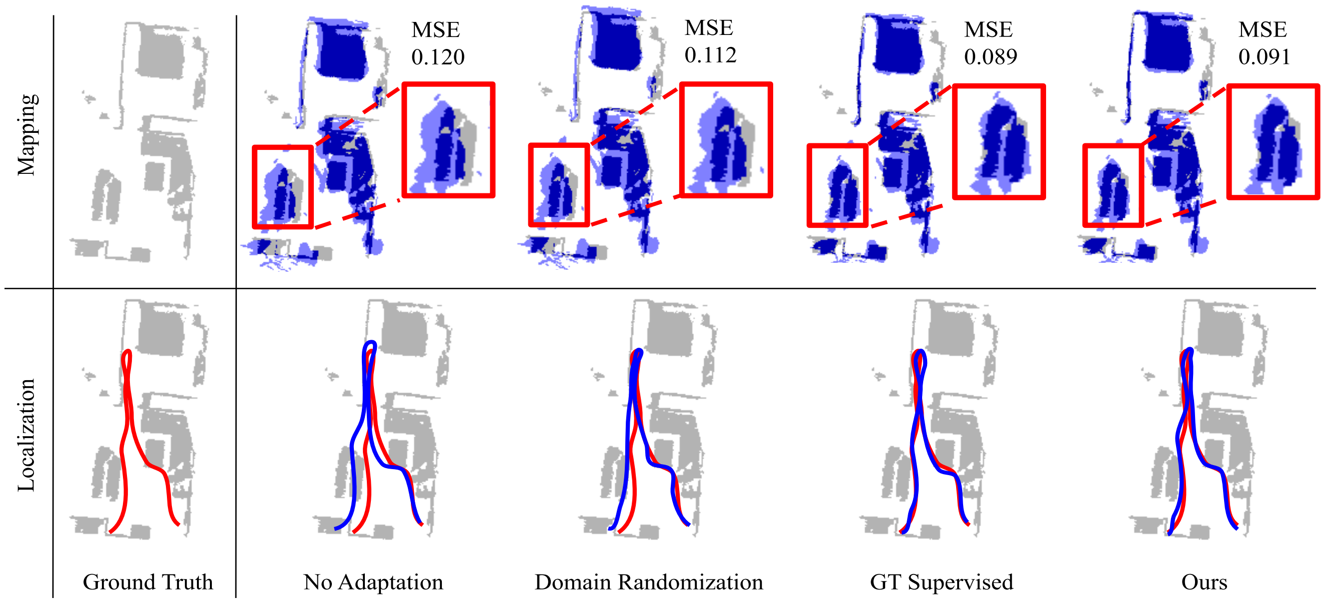


Figure 4. **Qualitative Result of Mapping and Localization.** The first row shows the qualitative result of the global maps generated from the same sequence of actions. The generated map is indicated in blue and the ground truth map is indicated in gray. The second row shows the localization result where red is the ground-truth trajectory and blue is the estimated trajectory of each agent.

noise. In contrast, "GT Train (**GT**)" is the model fine-tuned with the ground-truth supervision in the new environment. While the ground-truth pose information is not available in the real world, GT serves as the upper bound of the performance for our adaptation model. In addition to the enhancement in pose estimation, the self-supervised agent successfully generates occupancy representation as accurately as the GT does. We also compare our method to the "Domain Randomization (**DR**)" [10], which exposes the model randomly to the various combinations of actuation and odometry noise and train with ground-truth supervision. To expose the agent with sufficient variations of unknown noise, DR needs a significant amount of training data (∼1.5k trajectories) for fine-tuning. Nonetheless, our model trained with the self-supervised learning method outperforms in all performance metrics by successfully compensating for the unknown noise within a faster training time.

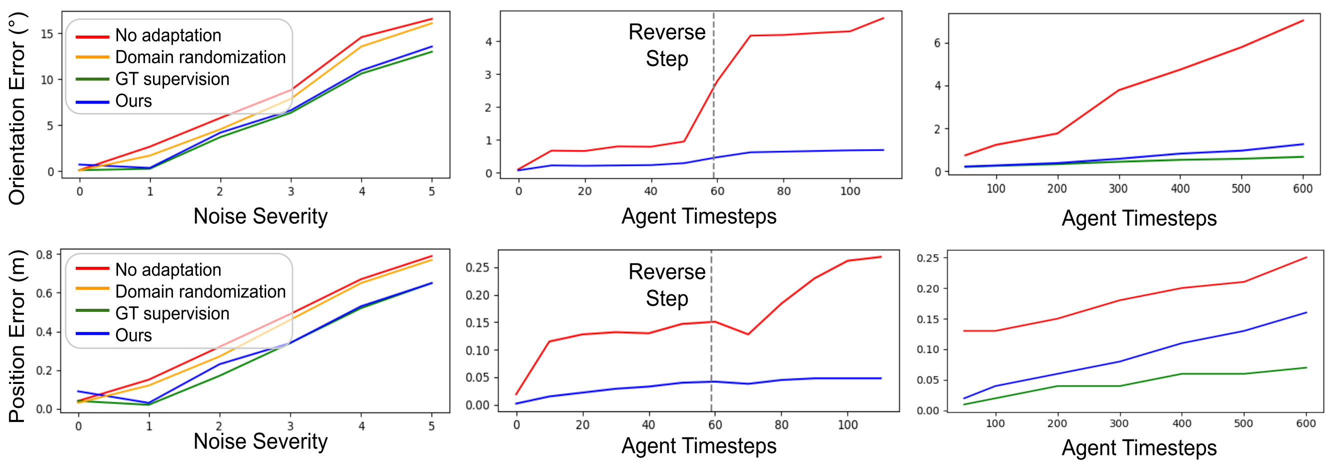
The result exhibits that the global map consistency is a powerful self-supervision task for adapting a pre-trained agent to an unseen environment. Figure 4 shows the visualization of the global maps and pose trajectories from all agents executing the same sequence of actions. The reconstructed map and the trajectory better align with the ground truth and therefore visually confirm that our model outperforms the pre-trained model and the domain randomization model. More qualitative results are available in the supplementary material. Below we also provide the analysis on various conditions of our proposed method.

Method		NA	GT	DR	Ours
Constant Motion Bias	x, y (m)	0.16	0.03	0.14	0.04
	ϕ (°)	2.69	0.28	1.93	0.3
Stochastic Motion Bias	x, y (m)	0.16	0.04	0.14	0.05
	ϕ (°)	2.77	0.32	2.06	0.34
Motion Drift	x, y (m)	0.18	0.05	0.16	0.06
	ϕ (°)	3.91	0.49	2.39	0.82

Table 2. Localization error with respect to various types of actuation noise.

Analysis on Various Noise Conditions We additionally demonstrate the performance of environment transfer under various noise conditions. We first set up six environments where the sensor noise severity k in Equation 5 ranges from 0 to 5. Our agent is separately trained in each environment with the self-supervised learning task and compared against the original pre-trained model and the agent trained with ground-truth supervision. Figure 5(a) shows that our self-supervised agent consistently reports superior performance against the agent which is not fine-tuned. The result guarantees that the proposed self-supervised learning task can be utilized for various agents regardless of its odometry sensor noise level.

Our self-supervision task does not assume any noise model and can be applied in various other types of noise. To test the stability of the performance of our fine-tuned agent, we further set up three environments where the three types of actuation noise introduced in Section 3.1 are present, namely constant / stochastic motion bias and motion drift.



(a) Effect of odometry noise severity (b) Comparison of forward / backward paths (c) Effect of path length

Figure 5. **Localization Error.** Performance analysis in task adaptation to noisy environments

The outlined noise corruption reflects the actuation noise model from LoCoBot [26, 40] and it creates a more challenging environment for agents. The result is reported in Table 2. In all cases, our transferred agent shows superior performance against the pre-trained model and exhibits high performance on a par with the GT Train model with the simple self-supervision regardless of unknown variations of noise models.

Forward vs. Backward Error Our formulation in Equation 8 treats the map generated during the forward step M_F as the supervision signal to backward map $M_{B,t}$. However, M_F is also error-prone, and one may question whether it is a valid assumption to treat the forward and backward path separately. To answer the question, we analyze the accumulated pose errors in agent time steps during the round-trip trajectories. In Figure 5(b), we observe that both position and orientation errors increase for all agents as they progress in a trajectory. The motion of turning steps induces significant movement compared to ordinary progression, and introduces a large amount of variation in both position and orientation, especially for the agent trained in a noiseless environment. Although a similar trend is shown in our agent, the self-supervised learning task optimizes the agent to experience less variation. As shown in the graph, our agent consistently accumulates the pose estimation error within a similar extent during its forward and backward path. Therefore the backward path indeed suffers from comparably severe pose errors after the turning point, which could greatly be alleviated from the self-supervision.

Task Robustness to Path Length For our proposed self-supervision task, the only constraint we impose on the set of trajectories collected for training is round-trip. We report that agents trained on a set of fixed-length round-trip trajectories can further generalize to longer trajectories. In the standard setup presented in Table 1, our proposed task fine-tunes the agent using short trajectories of 100 agent steps. We show the adaptation performance when an agent is evaluated on trajectories of different lengths, varying from 50 steps up to 600. As the step length increases, we observe diminishing enhancement as shown in Figure 5(c). However, our proposed method still noticeably exceeds the lower bound performance across various trajectory lengths.

Number of Trajectories		NA	40	80	120	160
Localization	x, y (m)	0.15	0.11	0.05	0.04	0.03
	ϕ (°)	2.67	1.47	0.4	0.38	0.36
Mapping	MSE	0.25	0.23	0.18	0.18	0.17

Table 3. Robustness to training data size evaluated on localization and mapping. Note that 'MSE' denotes 'mean squared error'.

Task Robustness to Training Data Size We observe if the self-supervised learning task is trainable from a smaller set of collected trajectories. Our standard experiment uses 160 trajectories for training. In Table 3, we show the result of our self-supervised agent trained on a different amount of trajectory data; 40, 80, 120, and 160. It demonstrates the correlation that more number of trajectories leads to higher pose estimation and mapping performance. Nonetheless, an agent fine-tuned with 40 trajectories still achieves performance enhancement over the pre-trained agent. Furthermore, we suggest an offline data augmentation method in Section 3.2 which increases the number of trainable trajectories size under a limited set of conditions.

4.2. Test-Time Adaptation

In the previous experiments, we followed the conventional train/val split in the Gibson dataset [48] and observed that the self-supervised learning task can adapt an agent to unknown actuation and odometry noise. The training tra-

Method		NA	GT	Ours
Localization	x, y (m)	0.15	0.02	0.05
	ϕ (°)	2.73	0.25	0.5
Mapping	MSE	0.16	0.11	0.12
Exploration	Ratio (%)	82.02	91.06	92.25
	Area (m^2)	26.36	30.65	31.27

Table 4. Performance analysis on test-time adaptation.

Task	Localization		Mapping
Method	x, y (m)	ϕ (°)	MSE
Ours	0.03	0.36	0.17
Ours w/o data augmentation	0.06	0.41	0.19
Ours w/ last step update	0.15	2.57	0.24
Ours w/ BCE Loss	0.16	2.78	0.26

Table 5. Ablation Study

jectories are collected from 72 training scenes and the performance is evaluated in 14 val scenes. The training scenes for fine-tuning were the same set of scenes used for the pre-training. However, it is not realistic for an agent to observe the real-world noises in the pre-trained environment. In a practical scenario, a pre-trained agent might be expected to adapt to a new environment that is not available in a simulator.

For test-time adaptation, we design an experiment to demonstrate the task's applicability for practical real-world deployment. We investigate the scenario where the pre-trained agent is fine-tuned in a single scene that has not been observed before. The new scene contains noise identical to the standard experiment in Table 1, which the agent was never exposed to during pre-training. We also use a small set of trajectories and see if an agent can be fine-tuned without over-fitting. We collect 20 trajectories for training and test the agent's performance on 80 trajectories both within the same scene, compared to 160 trajectories used for training the agent presented in Section 4.1. The GT train agent is fine-tuned in a new scene with ground-truth data provided and tested in the same scene during 80 test trajectories.

As shown in Table 4, our agent performs highly accurate pose estimation and mapping. Moreover, in the exploration task result, our agent outperforms the GT train agent. Using only a small set of collected trajectories, our light-weight self-supervised learning finishes training an agent within 2.5 minutes. Our result guarantees that the proposed task can quickly adapt agents in a real-world deployment setting.

4.3. Ablation study

In this section, we study ablated versions of our self-supervised learning method. All experiments are conducted in the same condition as described in Section 4.1.

Data Augmentation Given 160 trajectories collected with the pre-trained agent, we augment the data with the proposed random cropping method from Section 3.2. In Table 5, we compare our agent against the agent trained without data augmentation. Our agent estimates the pose more accurately, showing that data augmentation helps agents to generalize across environments.

Step-wise Supervision Our agent back-propagates the global map consistency loss at every step during the backward path. We examine if the step-wise supervision is necessary by comparing against the agent learns from the self-supervised learning only at its last step. The result presented in Table 5 shows that training the self-supervised learning task at every step indeed is helpful and exhibits more accurate pose estimation and mapping compared to the ablated agent which trains only at the last step.

MSE vs. BCE Loss Another possible loss to compare the binary occupancy map is binary cross entropy (BCE) loss, which is used in many existing approaches for map prediction [7, 28]. When used in our self-supervised task, BCE loss shows inferior performance in pose estimation than MSE. We empirically observe that BCE loss incurs noisy gradients compared to MSE loss, with higher variance in magnitude. Such instability in gradients leads to poor convergence and deteriorates fine-tuning. Further analysis on gradients of the two losses is in the supplementary material.

5. Conclusion

In this paper, we presented a light-weight and simple, self-supervised learning task for agents transferring to new, noisy environments. We demonstrated the quick and robust adaptation performance of the self-supervised agent to various actuation and odometry sensor noise on a par with the agent trained with ground-truth data. Our result presents an insight that the global map consistency provides a better understanding of the agent's pose and its surroundings. We also observe that improvement in observation space leads to the enhancement in action space for embodied visual navigation tasks. We expect our proposed method to be easily applicable to most visual navigation agents due to its simplicity. Furthermore, our optimistic test-time adaptation result points at the real-world deployment as the future research direction.

Acknowledgements

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. NRF-2020R1C1C1008195), Samsung Electronics Co., Ltd, and the BK21 FOUR program of the Education and Research Program for Future ICT Pioneers, Seoul National University in 2021.

References

[1] *"LoCoBot: An open source low cost robot"*. http://www.locobot.org/, 2021.

[2] Cesar Cadena, Luca Carlone, Henry Carrillo, Yasir Latif, Davide Scaramuzza, José Neira, Ian Reid, and John J Leonard. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on robotics*, 32(6):1309–1332, 2016.

[3] Peide Cai, Sukai Wang, Yuxiang Sun, and Ming Liu. Probabilistic end-to-end vehicle navigation in complex dynamic environments with multimodal sensor fusion. *IEEE Robotics and Automation Letters*, 5(3):4218–4224, 2020.

[4] Dylan Campbell, Lars Petersson, Laurent Kneip, and Hongdong Li. Globally-optimal inlier set maximisation for camera pose and correspondence estimation. *IEEE transactions on pattern analysis and machine intelligence*, 42(2):328–342, 2018.

[5] Dylan Campbell, Lars Petersson, Laurent Kneip, Hongdong Li, and Stephen Gould. The alignment of the spheres: Globally-optimal spherical mixture alignment for camera pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11796–11806, 2019.

[6] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments. *arXiv preprint arXiv:1709.06158*, 2017.

[7] Devendra Singh Chaplot, Dhiraj Gandhi, Saurabh Gupta, Abhinav Gupta, and Ruslan Salakhutdinov. Learning to explore using active neural slam. *arXiv preprint arXiv:2004.05155*, 2020.

[8] Devendra Singh Chaplot, Dhiraj Prakashchand Gandhi, Abhinav Gupta, and Russ R Salakhutdinov. Object goal navigation using goal-oriented semantic exploration. *Advances in Neural Information Processing Systems*, 33, 2020.

[9] Devendra Singh Chaplot, Ruslan Salakhutdinov, Abhinav Gupta, and Saurabh Gupta. Neural topological slam for visual navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12875–12884, 2020.

[10] Prithvijit Chattopadhyay, Judy Hoffman, R. Mottaghi, and Aniruddha Kembhavi. Robustnav: Towards benchmarking robustness in embodied navigation. *ArXiv*, abs/2106.04531, 2021.

[11] Yevgen Chebotar, Ankur Handa, Viktor Makoviychuk, Miles Macklin, Jan Issac, Nathan Ratliff, and Dieter Fox. Closing the sim-to-real loop: Adapting simulation randomization with real world experience. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8973–8979. IEEE, 2019.

[12] Bryan Chen, Alexander Sax, Gene Lewis, Iro Armeni, Silvio Savarese, Amir Zamir, Jitendra Malik, and Lerrel Pinto. Robust policies via mid-level visual representations: An experimental study in manipulation and navigation. *arXiv preprint arXiv:2011.06698*, 2020.

[13] Changan Chen, Unnat Jain, Carl Schissler, Sebastia Vicenc Amengual Gari, Ziad Al-Halah, Vamsi Krishna Ithapu, Philip Robinson, and Kristen Grauman. Audio-visual embodied navigation. *environment*, 97:103, 2019.

[14] Kevin Chen, Juan Pablo de Vicente, Gabriel Sepulveda, Fei Xia, Alvaro Soto, Marynel Vázquez, and Silvio Savarese. A behavioral approach to visual navigation with graph localization networks. *arXiv preprint arXiv:1903.00445*, 2019.

[15] Tao Chen, Saurabh Gupta, and Abhinav Gupta. Learning exploration policies for navigation. *arXiv preprint arXiv:1903.01959*, 2019.

[16] Abhishek Das, Samyak Datta, Georgia Gkioxari, Stefan Lee, Devi Parikh, and Dhruv Batra. Embodied question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–10, 2018.

[17] Zhiwei Deng, Karthik Narasimhan, and Olga Russakovsky. Evolving graphical planner: Contextual global planning for vision-and-language navigation. *arXiv preprint arXiv:2007.05655*, 2020.

[18] Hugh Durrant-Whyte and Tim Bailey. Simultaneous localization and mapping: part i. *IEEE robotics & automation magazine*, 13(2):99–110, 2006.

[19] Spyros Gidaris, Praveer Singh, and N. Komodakis. Unsupervised representation learning by predicting image rotations. *ArXiv*, abs/1803.07728, 2018.

[20] José Gonçalves, José Lima, Hélder Oliveira, and Paulo Costa. Sensor and actuator modeling of a realistic wheeled mobile robot simulator. In *2008 IEEE International Conference on Emerging Technologies and Factory Automation*, pages 980–985. IEEE, 2008.

[21] Daniel Gordon, Abhishek Kadian, Devi Parikh, Judy Hoffman, and Dhruv Batra. Splitnet: Sim2sim and task2task transfer for embodied visual navigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1022–1031, 2019.

[22] Daniel Gordon, Aniruddha Kembhavi, Mohammad Rastegari, Joseph Redmon, Dieter Fox, and Ali Farhadi. Iqa: Visual question answering in interactive environments. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4089–4098, 2018.

[23] Saurabh Gupta, James Davidson, Sergey Levine, Rahul Sukthankar, and Jitendra Malik. Cognitive mapping and planning for visual navigation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2616–2625, 2017.

[24] Nicklas Hansen, Rishabh Jangir, Yu Sun, Guillem Alenyà, Pieter Abbeel, Alexei A Efros, Lerrel Pinto, and Xiaolong Wang. Self-supervised policy adaptation during deployment. *arXiv preprint arXiv:2007.04309*, 2020.

[25] Allan Jabri, Andrew Owens, and Alexei A Efros. Space-time correspondence as a contrastive random walk. *Advances in Neural Information Processing Systems*, 2020.

[26] Abhishek Kadian, Joanne Truong, Aaron Gokaslan, Alexander Clegg, Erik Wijmans, Stefan Lee, Manolis Savva, Sonia Chernova, and Dhruv Batra. Are we making real progress in simulated environments? measuring the sim2real gap in embodied visual navigation. 2019.

[27] Abhishek Kadian, Joanne Truong, Aaron Gokaslan, Alexander Clegg, Erik Wijmans, Stefan Lee, Manolis Savva, Sonia Chernova, and Dhruv Batra. Sim2real predictivity: Does evaluation in simulation predict real-world performance? *IEEE Robotics and Automation Letters*, 5(4):6670–6677, 2020.

[28] Peter Karkus, Shaojun Cai, and David Hsu. Differentiable slam-net: Learning particle slam for visual navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2815–2825, 2021.

[29] Pradeep K Khosla. Categorization of parameters in the dynamic robot model. *IEEE Transactions on Robotics and Automation*, 5(3):261–268, 1989.

[30] Sven Koenig and Maxim Likhachev. Dˆ* lite. *Aaai/iaai*, 15, 2002.

[31] Shangda Li, Devendra Singh Chaplot, Yao-Hung Hubert Tsai, Yue Wu, Louis-Philippe Morency, and Ruslan Salakhutdinov. Unsupervised domain adaptation for visual navigation. *arXiv preprint arXiv:2010.14543*, 2020.

[32] Lina Mezghani, Sainbayar Sukhbaatar, Thibaut Lavril, Oleksandr Maksymets, Dhruv Batra, Piotr Bojanowski, and Karteek Alahari. Memory-augmented reinforcement learning for image-goal navigation. *arXiv preprint arXiv:2101.05181*, 2021.

[33] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. Orb-slam: a versatile and accurate monocular slam system. *IEEE transactions on robotics*, 31(5):1147–1163, 2015.

[34] Adithyavairavan Murali, Tao Chen, Kalyan Vasudev Alwala, Dhiraj Gandhi, Lerrel Pinto, Saurabh Gupta, and Abhinav Gupta. Pyrobot: An open-source robotics framework for research and benchmarking. *arXiv preprint arXiv:1906.08236*, 2019.

[35] Taesung Park, Alexei A Efros, Richard Zhang, and Jun-Yan Zhu. Contrastive learning for unpaired image-to-image translation. In *European Conference on Computer Vision*, pages 319–345. Springer, 2020.

[36] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32:8026–8037, 2019.

[37] Naman Patel, Anna Choromanska, Prashanth Krishnamurthy, and Farshad Khorrami. A deep learning gated architecture for ugv navigation robust to sensor failures. *Robotics and Autonomous Systems*, 116:80–97, 2019.

[38] Santhosh K Ramakrishnan, Ziad Al-Halah, and Kristen Grauman. Occupancy anticipation for efficient exploration and navigation. In *European Conference on Computer Vision*, pages 400–418. Springer, 2020.

[39] Nikolay Savinov, Alexey Dosovitskiy, and Vladlen Koltun. Semi-parametric topological memory for navigation. *arXiv preprint arXiv:1803.00653*, 2018.

[40] Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, Devi Parikh, and Dhruv Batra. Habitat: A Platform for Embodied AI Research. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[41] Dhruv Shah, Benjamin Eysenbach, Gregory Kahn, Nicholas Rhinehart, and Sergey Levine. Ving: Learning open-world navigation with visual goals. *arXiv preprint arXiv:2012.09812*, 2020.

[42] Andrew Szot, Alex Clegg, Eric Undersander, Erik Wijmans, Yili Zhao, John Turner, Noah Maestre, Mustafa Mukadam, Devendra Chaplot, Oleksandr Maksymets, Aaron Gokaslan, Vladimir Vondrus, Sameer Dharur, Franziska Meier, Wojciech Galuba, Angel Chang, Zsolt Kira, Vladlen Koltun, Jitendra Malik, Manolis Savva, and Dhruv Batra. Habitat 2.0: Training home assistants to rearrange their habitat. *arXiv preprint arXiv:2106.14405*, 2021.

[43] Ryo Takahashi, Takashi Matsubara, and Kuniaki Uehara. Data augmentation using random image cropping and patching for deep cnns. *IEEE Transactions on Circuits and Systems for Video Technology*, 30(9):2917–2931, 2019.

[44] Huijuan Wang, Yuan Yu, and Quanbo Yuan. Application of dijkstra algorithm in robot path-planning. In *2011 second international conference on mechanic automation and control engineering*, pages 1067–1069. IEEE, 2011.

[45] Saim Wani, Shivansh Patel, Unnat Jain, Angel X Chang, and Manolis Savva. Multion: Benchmarking semantic map memory using multi-object navigation. *arXiv preprint arXiv:2012.03912*, 2020.

[46] Erik Wijmans, Abhishek Kadian, Ari Morcos, Stefan Lee, Irfan Essa, Devi Parikh, Manolis Savva, and Dhruv Batra. Dd-ppo: Learning near-perfect pointgoal navigators from 2.5 billion frames. *arXiv preprint arXiv:1911.00357*, 2019.

[47] Fei Xia, Amir R. Zamir, Zhi-Yang He, Alexander Sax, Jitendra Malik, and Silvio Savarese. Gibson Env: real-world perception for embodied agents. In *Computer Vision and Pattern Recognition (CVPR), 2018 IEEE Conference on*. IEEE, 2018.

[48] Fei Xia, Amir R Zamir, Zhiyang He, Alexander Sax, Jitendra Malik, and Silvio Savarese. Gibson env: Real-world perception for embodied agents. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9068–9079, 2018.

[49] B. Yamauchi. A frontier-based approach for autonomous exploration. In *Proceedings 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation CIRA'97. 'Towards New Computational Principles for Robotics and Automation'*, pages 146–151, 1997.

[50] Jingwei Zhang, Lei Tai, Ming Liu, Joschka Boedecker, and Wolfram Burgard. Neural slam: Learning to explore with external memory. *arXiv preprint arXiv:1706.09520*, 2017.