Check for updates

NAVIGATION

Navigating to objects in the real world

Theophile Gervet[1]*, Soumith Chintala[2], Dhruv Batra[2,3], Jitendra Malik[2,4], Devendra Singh Chaplot[2]

Copyright © 2023 The Authors, some rights reserved; exclusive licensee American Association for the Advancement of Science. No claim to original U.S. Government Works

Semantic navigation is necessary to deploy mobile robots in uncontrolled environments such as homes or hospitals. Many learning-based approaches have been proposed in response to the lack of semantic understanding of the classical pipeline for spatial navigation, which builds a geometric map using depth sensors and plans to reach point goals. Broadly, end-to-end learning approaches reactively map sensor inputs to actions with deep neural networks, whereas modular learning approaches enrich the classical pipeline with learning-based semantic sensing and exploration. However, learned visual navigation policies have predominantly been evaluated in sim, with little known about what works on a robot. We present a large-scale empirical study of semantic visual navigation methods comparing representative methods with classical, modular, and end-to-end learning approaches across six homes with no prior experience, maps, or instrumentation. We found that modular learning works well in the real world, attaining a 90% success rate. In contrast, end-to-end learning does not, dropping from 77% sim to a 23% real-world success rate because of a large image domain gap between sim and reality. For practitioners, we show that modular learning is a reliable approach to navigate to objects: Modularity and abstraction in policy design enable sim-to-real transfer. For researchers, we identify two key issues that prevent today's simulators from being reliable evaluation benchmarks—a large sim-to-real gap in images and a disconnect between sim and real-world error modes—and propose concrete steps forward.

INTRODUCTION

Humans can navigate in unseen environments effortlessly. We can use our experience in prior environments to explore any new environment and find any target object efficiently. For example, when looking for a glass of water at a friend's house that we are visiting for the first time, we can easily find the kitchen without going to bedrooms or storage closets. Learning such semantic priors is essential to deploy autonomous mobile robots in uncontrolled environments such as homes, schools, and hospitals. In this work, we tackled the object goal navigation (*1*) task, where a robot is asked to find an object belonging to a particular category, such as a bed or a couch, in a completely unseen environment.

Navigation has been studied in robotics literature for over three decades (*2*–*13*). Many classical approaches to navigation require access to precomputed maps (*7*, *10*, *11*). Among approaches that can operate in unseen environments, classical approaches typically build a geometric map of the environment using depth sensors (*12*–*14*) and, later, a monocular RGB (red-green-blue) camera (*15*–*17*) while simultaneously localizing the robot relative to its growing map. Building on this simultaneous localization and mapping (SLAM) module, classical methods explore with heuristics, such as frontier-based exploration (*18*), and leverage an analytical planner for low-level control to avoid obstacles and reach exploration or point goals. Adapting these methods to navigate to objects requires detecting objects, keeping objects in memory, and exploring semantically toward objects. Semantic SLAM methods (*19*–*25*) naturally extend SLAM to detect objects and keep them in memory but offer no solution for efficient semantic exploration.

After recent advances in machine learning and computer vision, there has been a lot of interest in designing learning-based policies for visual navigation capable of learning these semantic priors. The most common learning-based methods for semantic navigation use a deep neural network, usually consisting of a visual encoder, followed by a recurrent layer for memory, to predict actions directly from raw observations. These end-to-end learning approaches are trained using backpropagation with imitation learning (IL) or reinforcement learning (RL) losses. Inspired by seminal proofs of concept in autonomous driving (*26*, *27*) and early successes of pixel-to-action deep RL (*28*, *29*), early applications of end-to-end learning to semantic navigation include studies in (*30*–*40*). Most relevant to us, Ye *et al.* (*41*), Maksymets *et al.* (*42*), and Ramrakhya *et al.* (*43*) proposed end-to-end policies to navigate to object goals. End-to-end RL methods have been scaled to train with billions of frames corresponding to 80 years of navigation time (in sim) using distributed training (*44*) or tens of thousands of procedurally generated scenes (*45*). Although end-to-end policies often directly map sensor data to actions, akin to a modern version of R. Brook's subsumption architecture (*46*), researchers have also introduced some structure into the neural network, such as intermediate spatial (*47*–*51*) and topological representations (*52*–*54*) or a finite set machine of visual state abstractions (*55*).

Classical and end-to-end learning–based approaches offer distinct advantages. End-to-end learning policies can learn semantic priors for goal-directed exploration, whereas the modularity of the classical pipeline offers easier engineering and interpretability (*56*). After successful applications across robotics in autonomous driving (*56*, *57*), flight (*58*), and grasping (*59*–*61*), much work has focused on designing modular learning approaches that aim to combine the benefits of both learning and classical methods. Modular learning approaches preserve the structure of the classical pipeline and replace analytical modules for specific subtasks with learned ones. In semantic navigation, the subtask decomposition typically includes separate modules for perception (object detection, mapping, pose estimation, and SLAM), encoding goals,

[1]Carnegie Mellon University, Pittsburgh, PA, USA. [2]Meta AI Research, Menlo Park, CA, USA. [3]Georgia Institute of Technology, Atlanta, GA, USA. [4]University of California, Berkeley, CA, USA.
*Corresponding author. Email: tgervet@andrew.cmu.edu

Downloaded from https://www.science.org at The Hong Kong University of Science and Technology (Guangzhou) on May 25, 2026

global waypoint selection policies, planning, and local obstacle avoidance policies. Learned modules are trained using direct supervision, which offers better sample efficiency than end-to-end learning (*56*). Modular learning can enable sim-to-real transfer (*57*, *62*, *63*): One can shield learned modules from sim-to-real domain gaps by designing abstractions of the input raw sensor data that contain sufficient information to solve the task while being invariant to environmental factors that are hard to simulate accurately (such as photorealistic RGB images) and thus unlikely to transfer from sim to reality. Among representative examples of modular learning for exploration and for reaching object goals and image goals (*62*–*67*), Chaplot *et al.* (*63*) is most relevant to us, cleanly isolating the problem of learning a policy to explore semantically toward objects from the rest of the navigation problem. Modular learning methods have also been applied to longer-horizon tasks, such as following language navigation instructions (*68*, *69*), executing language instructions interactively in ALFRED (Action Learning From Realistic Environments and Directives) (*70*–*72*), executing object rearrangement in AI2-Thor (*73*, *74*), and improving perception using active exploration (*75*, *76*).

Over the past few years, the semantic navigation community has proposed hundreds of methods and organized tens of benchmarks in sim (*77*). In our view, the missing piece of the puzzle to enable robots to navigate semantically is a large-scale real-world evaluation. Learned navigation policies have predominantly been evaluated in sim (*78*). We can attribute this to the emergence of sophisticated embodied simulators (*35*, *79*, *80*), which reduced the field's barrier to entry by bypassing the operational difficulties of bringing a robot out of the laboratory into diverse, realistic deployment environments. Although simulators can be very useful for training, they are insufficient for evaluation. In the end, how well a method performs on a robot is the only thing that matters, and we only have partial answers, if any, as to whether sim is a good evaluation benchmark for semantic navigation. We do not know whether sim performance reflects real-world performance, whether design choices that improve sim performance improve real-world performance, or whether sim error modes reflect real-world error modes. Most prior sim-to-real studies in navigation focused on spatial (point goal) navigation and legged locomotion (*81*–*86*), as opposed to semantic navigation. The few other real-world semantic navigation works directly trained on real-world images for outdoor visual goals (*87*, *88*) and language instruction following the study of Anderson *et al.* (*89*). A lack of real-world evaluation opens semantic navigation to the risk of sim-only research that does not generalize to the real world (*90*).

Our proposed work addresses this issue through a large-scale empirical evaluation of semantic navigation policies. We compared representative methods from classical, modular learning, and end-to-end learning approaches across six visually diverse real home environments, as illustrated in Fig. 1. This represents 45 hours of robot experiments (3 methods × 6 homes × 10 episodes per home × 15 min per episode). In addition, we replicated one home in sim to ensure that our experimental setting matched that of sim benchmarks and to decompose the sim-to-real performance gap. We found that modular learning transferred to the real world very well, with performance rising from an 81% success rate in sim to 90% in the real world (within a limited time budget). In contrast, end-to-end learning performance dropped sharply from 77% sim to 23% real-world success rate because of a large image domain gap between sim and reality. Classical approaches fell in between, with an 80% real-world success rate.

The takeaway for practitioners looking to build robots that navigate to objects is that the modular learning pipeline is very reliable, with a 90% success rate in limited time and efficient object search. In addition, we show that the remaining errors were primarily due to depth sensor failures (mapping failures and reflections in mirrors and TVs), which offers a path toward even greater reliability through better sensing or methods better able to deal with depth noise.

The takeaway for researchers is that much work remains to close the sim-to-real gap for semantic navigation. We identified two key issues that prevent today's simulators, three-dimensional (3D) assets, and task definitions from being reliable evaluation benchmarks. Then, we propose concrete steps along two orthogonal paths forward: improve sim to better reflect real-world conditions and improve practices to work with imperfect sim.

First, there is a large visual gap between sim and reality. Because of this, design choices easily overfit to sim. For example, policy architectures that directly operate on RGB images do not transfer because they overfit to sim images, and the common practice of training segmentation models on sim data improves sim performance at the cost of real-world performance. Improving sim to close this gap would involve increasing RGB photorealism or providing extensive RGB randomization, both hard open problems (*90*). This leaves us with no choice but to prioritize real-world transfer when designing policies: replace policy architectures that directly operate on RGB images with ones leveraging abstractions, as is common practice in other domains (*57*, *60*, *91*), and avoid training a segmentation model on sim data if the architecture does not allow easily swapping it for one trained on real-world data.

Second, sim error modes do not accurately reflect real-world error modes, which limits the usefulness of sim to diagnose bottlenecks and further improve methods. Modular learning errors in the real world largely stem from depth sensor errors, whereas sim benchmarks usually assume perfect depth or do not provide realistic depth noise models. In contrast, errors in sim largely stem from reconstruction errors that do not happen in reality—both visual (imperfect RGB reconstruction that makes semantic segmentation harder in sim than reality) and physical (noisy navigation meshes that make planning harder in sim than reality). This explains the increase in performance from sim to reality for modular learning and stresses the need to always evaluate semantic navigation policies on real robots. We propose concrete steps forward to close this gap: introduce realistic depth noise models for target deployment robots in sim benchmarks, improve the visual quality of sim 3D scans, and improve the quality of sim navigation meshes. We hope that our analysis sparks work to close the sim-to-real gap for semantic navigation.

RESULTS

Movie 1 summarizes our results. We have deployed a policy representative of each of the classical, end-to-end, and modular learning approaches to object goal navigation on a Hello Robot Stretch robot (*92*). Stretch is a lightweight, compact, low-cost mobile manipulator with an RGB-D camera and LIDAR (light detection and ranging), which we used only for localization and collision avoidance. We evaluated the policies over 60 episodes split across six goal object

Downloaded from https://www.science.org at The Hong Kong University of Science and Technology (Guangzhou) on May 25, 2026

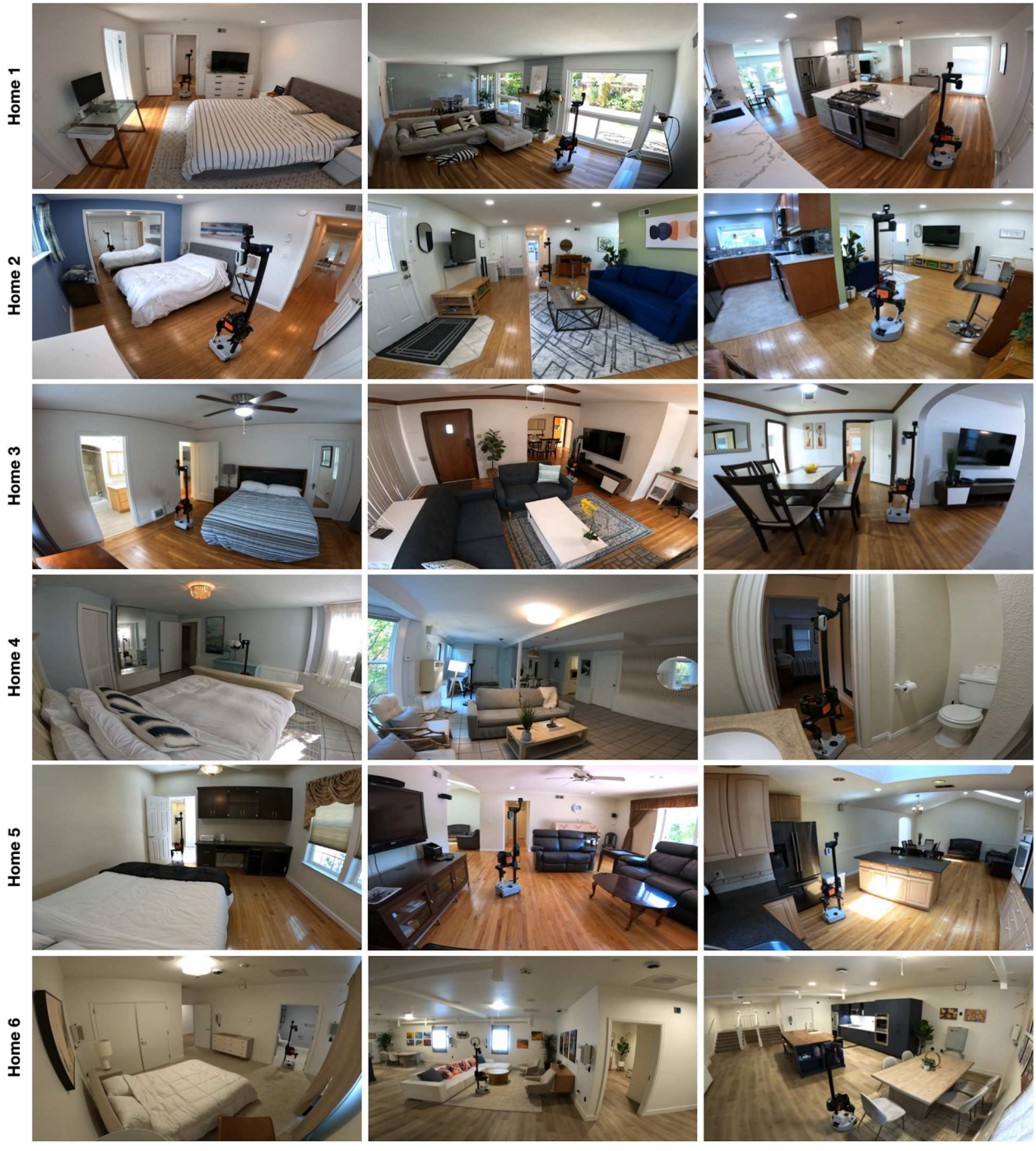


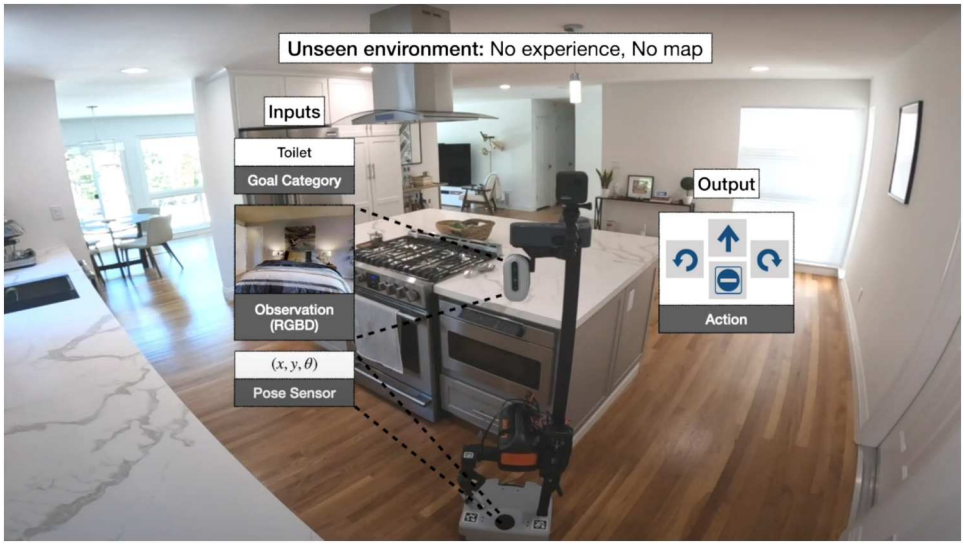
Fig. 1. Deployment of the semantic navigation policies in six visually diverse homes.

categories in six different homes and a controlled study with one home replicated in sim. Before presenting our results, we give minimal formal background on the object goal task and the methods that we evaluated.

Navigating to objects: Task and approaches

We consider the problem of semantic navigation instantiated by the object goal task (*1*, *63*, *93*). In the object goal task, the robot's objective was to navigate to an instance of a particular object category (in our case, "chair," "couch," "potted plant," "toilet," "tv," or "bed") as efficiently as possible. The robot started at a random location in an unknown home and received the goal object category. At each step, the robot observed first-person RGB and depth images and took a discrete navigation action: move forward (25 cm), turn left or right (30°), or stop. The robot needed to take the stop action when it believed that it reached the goal object. An episode was

Downloaded from https://www.science.org at The Hong Kong University of Science and Technology (Guangzhou) on May 25, 2026



Movie 1. Illustration of how the modular learning approach improved over the classical approach for robot visual navigation to objects in the home.

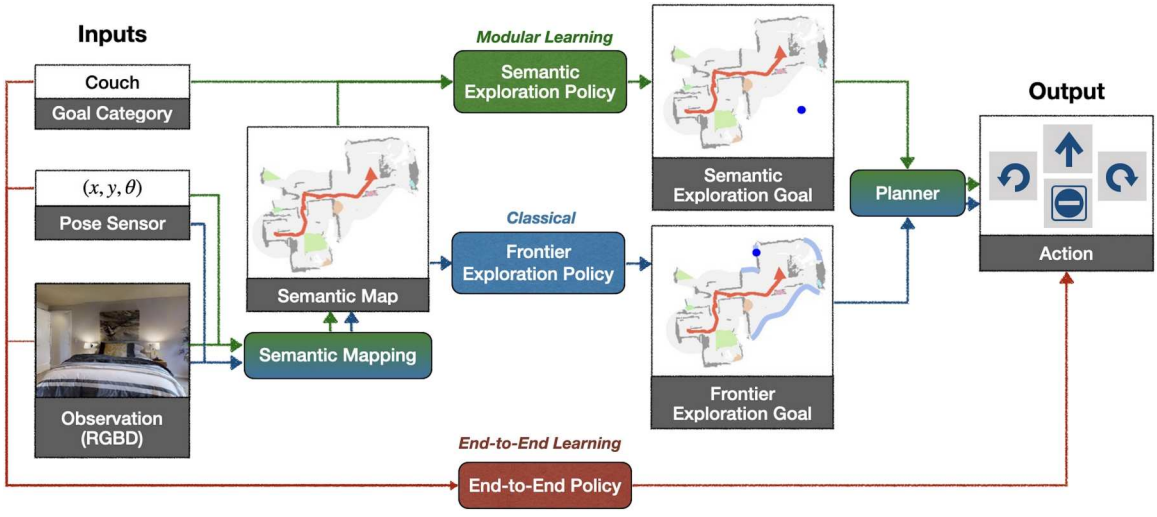


Fig. 2. Three approaches to navigate to objects. (Green) The modular learning approach builds a top-down semantic map, selects a semantic exploration goal in this space, and plans low-level actions to reach this goal. (Blue) The classical approach also builds a semantic map but selects the closest unexplored region as the exploration goal, independently of the goal object. (Red) The end-to-end learning approach directly maps sensor inputs and the goal object to low-level actions with a deep neural network.

considered successful if the robot's distance to an instance of the goal object category was less than some threshold (1 m) when the robot took the stop action. In contrast, an episode was a failure if the agent called the stop action too far from the goal object, never called the stop action before a fixed maximum number of time steps (500), or collided too many times (more than 20) with its environment. We used two metrics for comparing methods: success rate, the ratio of successful episodes, and success weighted by path length (SPL) (*1*), the ratio of path length over optimal path length for successful episodes, which measures exploration efficiency. The object goal task requires spatial scene understanding (obstacle and navigable space detection), semantic scene understanding (object detection), learning semantic priors (for efficient exploration), and episodic memory (keeping track of explored and unexplored areas).

We evaluated three methods, each representative of one class of approaches. To represent modular learning for object goal navigation, we picked Chaplot *et al.* (*63*). To represent classical approaches, we replaced the semantic exploration policy of Chaplot *et al.* (*63*) with frontier exploration (*18*), which navigates toward the closest unexplored region. Last, to represent end-to-end learning, we picked Ramrakhya *et al.* (*43*). We describe each method and what makes it representative of its class of approaches in detail in Materials and Methods. For now, Fig. 2 illustrates all the necessary background to understand the Results and Discussion.

Natural home environments

We deployed navigation policies representative of each class of approaches in six natural home environments never seen in sim or

Downloaded from https://www.science.org at The Hong Kong University of Science and Technology (Guangzhou) on May 25, 2026

reality. We evaluated the policies over 60 episodes split across the six goal object categories and six homes. This represents 45 hours of robot experiments (3 methods × 6 homes × 10 episodes per home × 15 min per episode).

Figure 3 illustrates our main results quantitatively. Classical and modular learning approaches performed better in the real world than in sim, up from 78 to 80% and from 81 to a 90% success rate, respectively. In contrast, end-to-end learning performed much worse in the real world, down from 77 to a 23% success rate. This trend holds across all homes and all goal object categories, as shown in tables S1 and S2. Figure 4 and fig. S2 compare all approaches on the same episode to illustrate these results qualitatively. Figure S3 illustrates two typical failure modes of the end-to-end learning policy, besides collisions. First, the policy often detected the goal object but failed to stop nearby, which is consistent with prior work observing this "last mile" failure in sim (*41*). Second, the policy revisited the same locations, often semantically unrelated to the goal object. These failure modes seemed to display a lack of semantic understanding, a lack of long-term memory, and poor exploration. Figure S1 and Movie 1 illustrate how the modular learning approach improved over the classical approach. The learned exploration policy leveraged the semantics of the top-down map to search for a specific goal object effectively. In contrast, the frontier exploration policy, which selected the closest unexplored region as the exploration goal, exhibited depth-first search behavior and failed to backtrack using semantics.

Controlled experiments in a home replicated in sim

We ran a controlled study in one home replicated in sim. This served two purposes. First, this let us decompose the discrepancy between results on a sim benchmark and results on the real world into the discrepancy between the sim benchmark and the sim replica as well as the discrepancy between the sim replica and the real home. This let us verify that our experimental setting matched the sim benchmark: Results in our sim replica should be close to those of the sim benchmark. Second, this allowed us to run an ablation study to select the best end-to-end policy to evaluate at scale across all six homes. We replicated a single home in sim because 3D scanning and digitizing a house—with the same Matterport camera used to digitize homes for the sim benchmark (*95*)—takes a few hours.

We first present the results of the ablation study that we conducted to select the best end-to-end policy to evaluate at scale across all six homes. Modular learning and classical approaches only use first-person RGB-D and predicted segmentation images through a top-down semantic map. This makes them independent of changes in camera parameters and lets us easily swap semantic segmentation

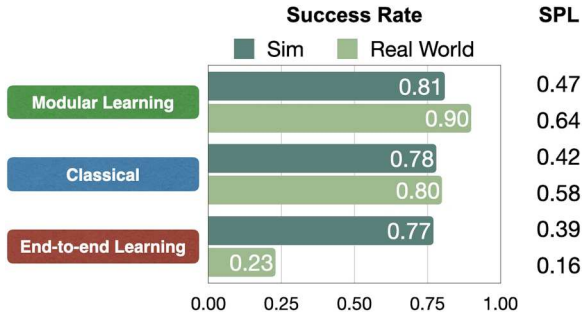


Fig. 3. Navigation performance in sim versus reality at scale. We compared the success rate and SPL of methods representative of classical, end-to-end-learning, and modular learning approaches on large real-world (60 episodes in six homes) and sim datasets [single-floor navigation episodes of validation split of the 2022 Habitat Challenge with 1093 episodes in 20 simulated homes of the HM3D Semantics dataset (*94*)]. Performance for all methods was comparable in sim, at around an 80% success rate. Classical and modular learning approaches transferred well, up from 78 to 80% and 81 to 90%, respectively. End-to-end learning failed to transfer, down from 77 to a 23% success rate.

Table 1. End-to-end learning ablations. In an ablation study of end-to-end policies, we varied training camera parameters, the segmentation model training domain, and the policy training algorithm.

	Policy training settings		
End-to-end policy	**Camera parameters**	**Segmentation training domain**	**Policy training algorithm**
End-to-end 1	Sim benchmark	Sim	IL + RL
End-to-end 2	Robot	Sim	IL + RL
End-to-end 3	Robot	Real-world	IL + RL
End-to-end 4	Robot	Real-world	IL

Table 2. Evaluation results. We compared policies by success rate (SR) on a sim benchmark [single-floor navigation episodes of val split of the 2022 Habitat Challenge with 1093 episodes in 20 simulated homes of the HM3D Semantics dataset (*94*)] and the same home in sim and reality (10 episodes). We computed the SRCC introduced in (*81*), which is the Pearson correlation between episode outcomes in sim versus reality (in [0, 1], higher is better). (Rows one to four) Real-world performance of end-to-end policy ablations was inversely correlated to sim performance. Policy 1 was the best in sim but the worst in the real world, whereas policy 4, which we selected for large-scale evaluation, was the worst in sim but the best in the real world. All variants other than policy 4 have a low SRCC, indicating that their design overfit to sim. (Rows five to seven) Performance on the sim replica was close to that on the sim benchmark, which shows that our experimental setting matched that of the sim benchmark. Although performance in the real-world home was close to that on its sim replica, the SRCC was still relatively low because policies failed different episodes in sim and the real world. As we will see in Discussion, this is due to a disconnect between error modes in sim and reality.

Navigation policy	**Sim benchmark SR**	**Real-home sim replica SR**	**Real-home SR**	**Real-home SRCC**
End-to-end 1	0.77	0.80	0.00	0.20
End-to-end 2	0.71	0.70	0.00	0.30
End-to-end 3	0.61	0.60	0.10	0.40
End-to-end 4	0.48	0.50	0.30	0.60
End-to-end	0.48	0.50	0.30	0.60
Modular	0.81	0.80	0.90	0.70
Classical	0.78	0.80	0.90	0.70

Downloaded from https://www.science.org at The Hong Kong University of Science and Technology (Guangzhou) on May 25, 2026

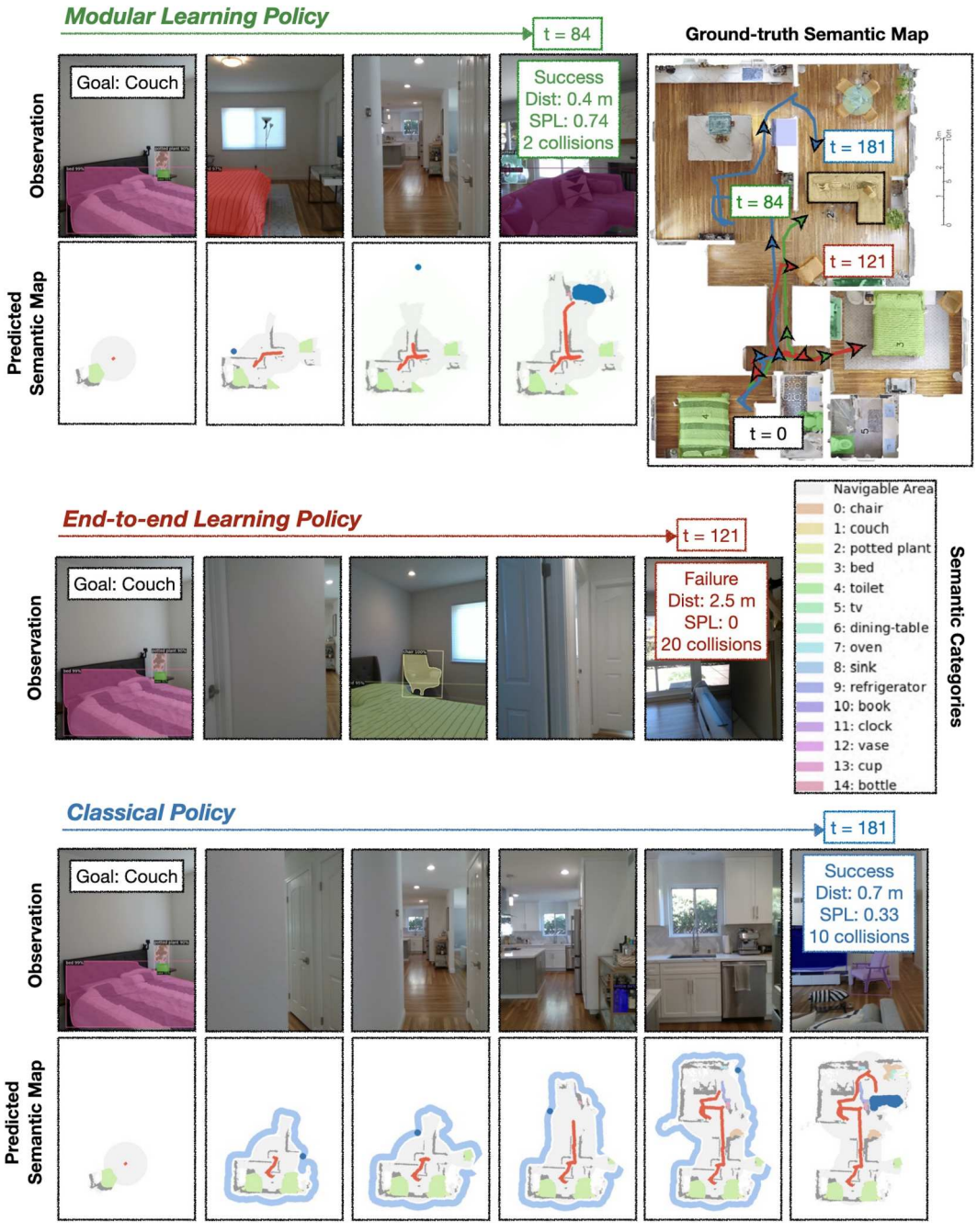


Fig. 4. Three approaches in the same episode. (Green) Modular learning reached the couch goal in 84 steps (SPL = 0.74). (Red) End-to-end learning collided too many times (20 max) after 121 steps. (Blue) The classical policy reached the goal after 181 steps and a detour through the kitchen (SPL = 0.33).

models between training and evaluation. In contrast, end-to-end learning approaches directly operate on RGB-D and predicted segmentation images. This means that varying camera parameters between training and evaluation settings introduced a domain gap, and an end-to-end policy is closely tied to the segmentation model used in its network architecture during training. These characteristics of end-to-end policies mean that to give them the best chance to work on a robot, an ablation study was needed to select the best-performing policy.

Table 1 shows results of this ablation study. We trained four different end-to-end policies, varying the camera parameters, segmentation model training domain, and training algorithm. We measured each policy's performance in sim, both on a large-scale benchmark and on our sim replica, and in the real home. Overall, policy 1—trained with the camera parameters of the sim benchmark, which we detail in Materials and Methods, and using a segmentation model trained in sim—performed the best in sim but the worst in the real world. Policy 2, trained with robot camera

Downloaded from https://www.science.org at The Hong Kong University of Science and Technology (Guangzhou) on May 25, 2026

parameters, performed slightly worse in sim, which shows that our robot camera parameters made the task slightly harder than the sim benchmark camera parameters. Policies 3 and 4, which used a segmentation model trained in the real world, performed worse in sim but better in the real world. This shows that using a segmentation model trained in sim is overfitting to sim. Policy 4, which was trained with IL only as opposed to IL followed by RL fine-tuning, performed the worst in sim but the best in reality. This shows that RL fine-tuning can overfit to sim. We selected policy 4 to evaluate at scale across all six homes.

Overall, these results show that performance in sim is often inversely proportional to performance in the real world because design choices to improve performance in sim can easily overfit to sim. We could quantify this observation through the Sim-vs-Real Correlation Coefficient (SRCC) introduced in Kadian *et al.* (*81*), which measures how well sim results can predict real-world results. The SRCC was low for all end-to-end policy variants, starting as low as 0.20 for policy 1, which performed best in sim, and increasing to 0.60 for policy 4 as we corrected for design choices that overfit to sim.

Now that we have selected the best end-to-end policy to evaluate at scale, Table 2 decomposes the discrepancy between results for all selected methods on the sim benchmark and the real world into the discrepancy between the sim benchmark and the sim replica as well as the discrepancy between the sim replica and the real home. First, performance on the sim replica was close to that on the sim benchmark, which verifies that our experimental setting matched that of the sim benchmark. Second, although absolute performance was fairly close in sim and the real world for the policies we evaluated at scale (0.80 sim versus 0.90 real success rate for modular learning), the SRCC was still low (only 0.70 for modular learning) because each method failed different episodes in sim and the real world. As we will see in Discussion, this is due to a disconnect between error modes in sim and reality.

DISCUSSION

In this section, we analyze error modes of the modular learning approach in sim versus reality to make sense of its rise in performance from sim to real, to emphasize the importance of the 90% real-world success rate of modular learning, to investigate why modular learning transfers better than end-to-end learning, and to reflect on what this means for the field.

Modular learning error modes in real world versus sim

To make sense of the rise from 81% sim to a 90% real-world success rate for the modular learning approach, we compare its sim and real error modes in table S3. Unexpectedly, table S3 shows that there was nearly no overlap between sim and real error modes. Errors in the real world largely stemmed from depth sensor errors (five of six total errors), as illustrated in fig. S4 and Movie 1, whereas the sim benchmark assumed perfect depth sensing. When approaching a door at an angle, noise in depth could block it in the map, making a room inaccessible without a map-denoising mechanism. Reflection in mirrors and TVs could also cause depth sensor errors and downstream navigation failures. In contrast, a lot of the failures that occurred in sim were due to reconstruction errors—both visual and physical—that did not happen in reality. Moreover, 10.1% of the total 18.6% episode failures in the sim benchmark were due to segmentation errors, whereas segmentation errors did not cause any episode failures for modular learning in the real world. Segmentation errors are more common in sim because visual reconstruction can make objects unrecognizable, as illustrated in fig. S5A and Movie 1. Physical reconstruction errors represent another 5.5% of the total 18.6% episode failures in sim. They led to noisy navigation meshes with narrow pathways that were hard to navigate for discrete planners that work well in the real world, as illustrated in fig. S5B and Movie 1. This gap in error modes explains the performance gap between sim and reality for modular learning.

The lack of overlap between sim and real-world error modes is concerning because it limits the usefulness of sim to diagnose bottlenecks and further improve policies. This is a practical working definition of the sim-to-real gap: We only care about improving sim realism to the extent that it lets us develop better real-world policies. A gap in error modes prevents us from doing so. This stresses the need to always evaluate semantic navigation policies on real robots for results to be meaningful. We propose concrete steps forward to close this gap: introduce realistic depth noise models for target deployment robots in sim benchmarks, improve the visual quality of sim 3D scans, and improve the quality of sim navigation meshes.

Toward solving navigation to objects with modular learning

The main takeaway of this study for practitioners looking to build robots that navigate to objects is that the modular learning pipeline was very reliable, with a 90% success rate in limited time and efficient object search with an SPL of 0.64. In addition, we show that the remaining errors were primarily due to depth sensor failures, which offers a path toward even greater reliability through better sensing or methods better able to deal with depth noise.

One limitation of our study is the restriction to six goal object categories to align with current sim benchmarks. All methods that we evaluated are straightforward to extend to a larger finite set of categories, and opportunities for future work include extensions to an unbounded set of object categories with open-vocabulary detectors (*96*) and instance-level object goals (*97*).

Why modular learning transfers better than end-to-end learning

There are multiple annual competitions evaluating navigation to objects in sim (one run by the Habitat team and one by AI2). Looking at the top three teams on each challenge from 2022, four approaches can be characterized as pixel-to-action end to end (*43*, *45*, *98*, *99*), only one can be characterized as modular (*100*), and one is unpublished. There is no evidence here of the dominance of modular learning; these results that were naively interpreted would point to the opposite conclusion. End-to-end learning pixel-to-action approaches with fewer inductive biases benefit the most from the ease of generating large-scale training data in sim and are able to compete with or even outperform modular learning approaches on sim visual navigation benchmarks. However, our study demonstrates that the performance of end-to-end learning pixel-to-action policies in sim does not yet translate into real-world performance because of a large sim-to-real visual domain gap in today's simulators, whereas modularity and abstraction in policy design enable sim-to-real transfer.

Downloaded from https://www.science.org at The Hong Kong University of Science and Technology (Guangzhou) on May 25, 2026

Figure 5 illustrates why modular learning transfers better than end-to-end learning. The semantic exploration policy of the modular learning approach took a semantic map as input, whereas the end-to-end policy directly took the RGB-D frames as input. The figure shows that the semantic map space was invariant between the simulation and the real world. This is due to the semantic map abstracting away pixels in favor of semantic categories and reducing the spatial granularity of the map voxel size (5 cm in our experiments). In contrast, the RGB image space exhibited a large gap between the real world and sim because current reconstruction engines cannot yet generate photorealistic images. This caused a large drift between the training and evaluation domains for the deep neural network of the end-to-end policy.

The importance of the gap from sim images to real-world images is well illustrated by the segmentation errors it caused in Fig. 5. A segmentation model trained in the real world suffered a severe performance drop when transferred to sim, down from 0.50 to 0.35 mean average precision (mAP) at a 0.50 intersection over union threshold (mAP@0.5). Similarly, a segmentation model trained in sim suffered an even larger performance drop when transferred to the real world (down from 0.45 to 0.10 mAP@0.5). If semantic segmentation transfers poorly from sim to reality, then it is reasonable to expect an end-to-end semantic navigation policy trained on sim images to transfer poorly to real-world images. The image domain gap affects a segmentation model over a single prediction, whereas the gap accumulates over many predictions made over a long horizon for an end-to-end navigation policy.

Because of this image domain gap, design choices easily overfit to sim. Two examples in our results are that end-to-end policy architectures that directly operated on RGB images did not transfer because they overfit to sim images and that the common practice of training segmentation models on sim data improved sim performance at the cost of real-world performance.

At this point, it is worth taking a step back to briefly survey how other domains of robotics tackle sim-to-real gaps. Prevalent options today (*90*) are trained on real-world data, trained in sim with domain randomization, and trained in sim with modularity and abstraction. Training on real-world data bypasses any sim-to-real gap but is expensive, slow, and potentially unsafe. It is the option of choice for perception stacks across robotics and has been applied to end-to-end control in domains where a suboptimal policy operation is safe, such as grasping and static manipulation (*101*, *102*). However, it is not straightforward to scale up for semantic navigation because mobile robots require constant supervision. This leaves us with sim training. Domain randomization (*103*)—randomizing environmental factors that are hard to simulate accurately during training to make policies robust to these factors—has been applied successfully in domains where randomization is straightforward, such as dexterous manipulation (*104*), where one needs to randomize over physics and single-object appearance. However, it is still an open problem whether RGB image randomization can be scaled up to entire houses to bring the same robustness to semantic navigation (*90*). Our final option, training in simulation with modularity and abstraction (*57*)—designing abstractions of the input raw sensor data that contain sufficient information to solve the task while being invariant to environmental factors that are hard to simulate accurately—is the practice of choice in autonomous flight (*91*, *105*), legged locomotion (*84*, *85*), and grasping (*59*, *60*) and is a promising path in autonomous driving (*57*).

In our view, training in sim and sim-to-real transfer via modularity and abstraction is the most promising path forward for semantic navigation. Example semantic abstractions could be first-

Downloaded from https://www.science.org at The Hong Kong University of Science and Technology (Guangzhou) on May 25, 2026

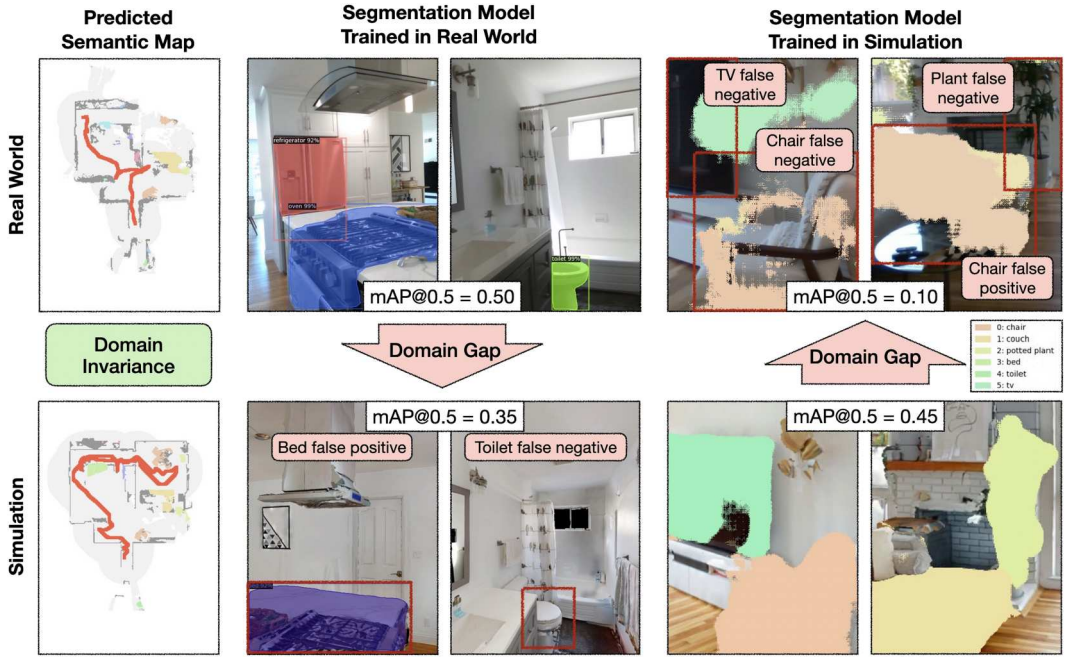


Fig. 5. Sim versus real domain invariances, gaps, and their effects on segmentation. From left to right, all images came from episodes in our controlled study. The semantic map space was invariant between the real world and the sim. The image space exhibited a large gap between the real world and sim. This gap caused a large drop in performance when transferring a segmentation model trained in the real world to sim and vice versa.

person semantic segmentation masks (*57*), topological scene representations (*53*), or top-down spatial semantic maps (*63*). One limitation of our study and opportunity for future work is that we did not take further steps in this direction, such as evaluating an end-to-end policy abstracting RGB frames through semantic frames.

MATERIALS AND METHODS

We deployed three navigation policies on a robot across six homes to study how well different methods to navigate to object categories transfer from sim to the real world. This section outlines our sim-to-real transfer methodology, details the specific methods we evaluate, and explains what makes them representative of broader classes of approaches.

Sim-to-real transfer methodology

Hardware and software stack

We deployed navigation policies on a Hello Robot Stretch robot (*92*). We selected Stretch because it is low cost, lightweight, and compact, allowing us to transport it out of the laboratory into real homes easily. We trained all learned components of navigation policies in sim with the Habitat platform (*106*). We selected Habitat for its sim speed, which is crucial for training navigation policy components with RL. At inference time, policies were deployed with the fairo library (*107*) to run the same navigation code in sim and the real world.

Matching policy inputs and outputs from sim to real

All navigation policies that we evaluated were trained in sim and had, so far, only been evaluated in sim. To transfer these policies to a robot and to be able to compare real-world results with those in sim, we had to match the object goal task input and output spaces from sim on the robot. In the object goal task, the robot received an RGB-D image and a sensor pose at each step. Stretch's Intel RealSense D435i camera gave us (640 pixels by 480 pixels) RGB-D images with a 42° horizontal field of view. We preprocessed the depth with standard spatial and temporal filters and thresholded all values beyond the camera's confidence range (4 m). We used off-the-shelf LIDAR-based Hector SLAM (*108*) to estimate the robot's pose. We also used LIDAR for collision detection to deploy policies safely. To guarantee a fair comparison with sim settings, we restricted the use of LIDAR to collision detection and localization and did not use it for mapping. Given that the output action space was discrete—move forward 25 cm, turn left/right 30°, or stop—it was straightforward to match on the robot. Sim-to-real transfer is illustrated in fig. S6A.

Generating episodes in real homes

We evaluated the robot on 60 episodes across six object goal categories (chair, couch, potted plant, toilet, TV, and bed) in six homes. We selected these goal object categories to be able to directly compare results in the real world with those of a leading sim benchmark: the Habitat 2022 Object Navigation Challenge. We selected homes for their visual diversity and size. To generate an episode within a home, we selected a goal category and a starting location for the robot while trying to balance the distribution of goal object categories and to match the distribution of geodesic distances to the goal to that of the Habitat Challenge, as shown in fig. S7. We considered an episode successful if the robot called the stop action close enough (less than 1 m) to an instance of the goal category within the time (200 steps) and collision (20 collisions) budgets. To compute the SPL (*1*), we measured the geodesic distance to the goal object instance closest to the starting location. To replicate one home and its episodes in sim in our controlled study, we scanned and digitized the home with a Matterport Pro2 3D camera and matched the real-world goals and starting positions in sim.

Figure S6B illustrates the architectures of the three policies we evaluated. We first describe the classical and modular learning semantic mapping approaches before turning our attention to end-to-end learning approaches.

Classical and modular learning approaches

Extending classical approaches to navigate to objects

Classical SLAM approaches to spatial navigation typically build a spatial map of the environment while simultaneously localizing the robot relative to its growing map (*12*, *13*) and navigate to geometric points within this map via path planning. Adapting these methods to navigate to objects requires detecting objects, keeping objects in memory, and exploring semantically toward objects. Semantic SLAM methods (*19*–*24*) naturally extend SLAM to detect objects and keep them in memory in a spatial semantic map but offer no solution for efficient semantic exploration. Among the many heuristics for goal-agnostic exploration proposed in the classical navigation literature, we selected frontier-based exploration (*18*)—navigate toward the closest unexplored region—to represent classical exploration methods because it was particularly effective in prior work (*63*). As shown in fig. S6B, this completes a pipeline representative of classical approaches: semantic mapping to keep seen objects in memory, frontier exploration to select high-level goals, and path planning to select low-level actions.

Note that if we only care about navigating to a single object starting with zero information about the environment—the setting evaluated in this paper—we do not even need to keep objects in memory in a semantic map in the classical approach. We can simply use a geometric map for exploration and planning and head toward the goal object as soon as we detect it in the frame with a pretrained object detector. We added semantic mapping to the classical approach for ease of implementation—we could head toward an object by projecting it into a single semantic channel in the map and planning toward it—and to make it applicable to the more practical but harder-to-evaluate setting where the robot might need to execute several such object search commands in a row.

Modular learning to navigate to objects

Although frontier exploration is effective, as shown in our results, it is necessarily suboptimal because it ignores the goal object. We intuitively understand where a couch is more likely to be found. This is where modular learning comes into play: Can we train an exploration policy to leverage the statistical regularities in the layout of objects in homes to explore more efficiently? This is what the semantic exploration method proposed in (*63*) does. We selected it to represent modular learning because it cleanly isolates the problem of learning an exploration module from the rest of the navigation problem. As shown in fig. S6A, it preserves the structure of the classical pipeline to navigate to the objects presented above but replaces frontier exploration with a learned semantic exploration module. With this high-level picture in mind, we explain each component in detail.

Semantic map representation

The semantic map is a spatial representation of the environment that keeps track of objects, obstacles, and explored areas. Concretely,

Downloaded from https://www.science.org at The Hong Kong University of Science and Technology (Guangzhou) on May 25, 2026

it is a binary $K \times M \times M$ matrix where $M \times M$ is the map size and K is the number of map channels. Each cell of this spatial map corresponds to 25 cm^2 (5 cm × 5 cm) in the physical world. Map channels K equals $C + 4$, where C is the number of semantic object categories and the remaining four channels represent the obstacles, the explored area, and the agent's current and past locations. An entry in the map is one if the cell contains an object of a particular semantic category, an obstacle, or is explored, depending on the channel, and is zero otherwise. The map is initialized with all zeros at the beginning of an episode, and the agent starts at the center of the map facing east.

Semantic mapping module

To build the semantic map, we needed to predict semantic categories and segmentation masks of objects in first-person observations. We used a Mask-RCNN (*109*) with a ResNet50 (*110*) backbone pretrained on MS-COCO (Microsoft Common Objects in Context) for object detection and instance segmentation. We projected first-person semantic segmentation into a point cloud using the depth, binned the point cloud into a 3D semantic voxel map, transformed it from the robot's coordinate system to that of the semantic map using the robot pose, and finally summed over the height to compute a 2D semantic map.

Frontier exploration policy

With our semantic map at hand, it was straightforward to implement frontier exploration: At each step, we selected the boundary between the explored and the unexplored regions of the map—the frontier—and, within this boundary, selected the point closest to the robot in geodesic distance. This exploration strategy exhibited depth-first search behavior: Once the robot headed in a direction, the closest unexplored region was in front of it until an obstacle blocked the way. As soon as the goal object was seen, as soon as the channel of the semantic map for the object goal had a nonzero element, we stopped exploring and selected all nonzero elements as the goal.

Semantic exploration policy

The semantic exploration strategy decided a goal as a function of the current semantic map and the goal object. This required learning semantic priors on the relative arrangement of objects and areas in homes. As shown in fig. S6B, map features were computed with a convolutional neural network (CNN) and passed through a feedforward neural network along with a learnable embedding for the goal object to compute a goal in $[0,1]^2$, which was then converted to top-down map space. The policy was trained using RL, with the distance reduced to the nearest goal object as the reward. We used a policy trained in (*63*) on home layouts from the Gibson dataset (*111*). As in (*63*), we sampled the long-term goal at a coarse time scale, once every 25 steps. This reduced the time horizon for exploration in RL exponentially and, consequently, reduced the sample complexity. As for the frontier exploration policy, as soon as the goal object was seen, we stopped exploring and selected it as the goal.

Planner

Given a long-term goal output by the frontier or semantic exploration policy, we used the fast marching method (*112*) as in (*63*) to plan a path and the first low-level action along this path deterministically. Although the semantic exploration policy acted at a coarse time scale, the planner acted at a fine time scale: At every step, we updated the map and replanned the path to the long-term goal.

Sim-to-real transfer

Semantic mapping methods were straightforward to transfer to the robot because they were independent of camera parameters, and the semantic map space was invariant between sim and the real-world, as presented in Discussion.

End-to-end learning approaches

End-to-end learning to navigate to objects

In contrast to modular approaches that explicitly build a semantic map of the environment and plan actions, end-to-end learning approaches directly predict actions from raw sensor data with a deep neural network. The network needs to learn to understand the scene spatially and semantically, to keep track of long-term memory, and to plan actions. Given that each of these tasks is hard in isolation, end-to-end learning approaches typically require tens of thousands of expert demonstrations or hundreds of millions of steps of RL to train. Given that neither of these can realistically be collected in the real world for our object goal navigation task, training must be done in sim. As illustrated in fig. S6B, at each step, the typical end-to-end architecture for long horizon tasks with high-dimensional image input computes image features with one or multiple CNNs, which can be trained from scratch or pretrained, and keeps track of memory and plans implicitly with a recurrent neural network, in our case, a gated recurrent unit (GRU) (*113*).

Habitat-Web policy

We selected Habitat-Web (*43*) as a representative example of end-to-end learning for the object goal navigation task because it closely follows the above paradigm and has the highest performance on a leading sim benchmark, the 2022 Habitat Challenge, at the time of publication. Habitat-Web (*43*) trains an end-to-end policy from human demonstrations on the object goal task in the Habitat simulator. We preserved the general architecture of the policy and swapped different components in our ablation study to find the policy that worked best in the real world. Semantic segmentation was predicted from RGB (and possibly depth) with a pretrained and frozen segmentation model. First-person RGB, depth, and semantic frame features were computed with multiple CNN backbones, leveraging pretrained models when available to reduce sample complexity. RGB, depth, and semantic frames were then fed into a GRU (*113*) along with other low-dimensional features—the robot pose, a learnable goal object embedding, the fraction of the visual input occupied by the goal category, and the previous action—to predict a distribution over the next actions.

Architecture and training details

The segmentation model used in the original architecture was a RedNet (*114*) pretrained on the SUN RGB-D dataset (*115*) and fine-tuned on images from the Habitat simulator. As shown in our results, using a segmentation model fine-tuned in sim hurts performance in the real world. In our ablation study of end-to-end architectures, we thus replaced this model with the same Mask-RCNN pretrained on MS-COCO used by modular semantic mapping methods. Depth features were computed with a ResNet50 pretrained on point goal navigation (*44*). RGB and semantic frame features were computed with a ResNet18 trained from scratch. The network was first trained with IL on 80,000 human demonstrations (or approximately 20 million actions) collected for the object goal task in the Habitat simulator over 3 days with 128 Nvidia V100 32-GB GPUs. It was then fine-tuned with RL with a sparse binary reward when the goal was found, over 150 million steps in an

Downloaded from https://www.science.org at The Hong Kong University of Science and Technology (Guangzhou) on May 25, 2026

additional 3 days on 32 Nvidia V100 32GB GPUs. Our ablation study of end-to-end policies shows that RL fine-tuning helped in sim but hurt in the real world.

Sim-to-real transfer

Semantic mapping methods are independent of camera parameters, whereas end-to-end methods directly operate on first-person frames, and any change in camera parameters causes a large domain gap. Given that our robot camera parameters were different from those used to train the original Habitat-Web policy—(640 pixels by 480 pixels) frames with a 42° horizontal field of view, as opposed to (480 pixels by 640 pixels) frames with a 79° horizontal field of view—we retrained the policy with robot camera parameters replicated in sim.

As shown in fig. S6A, real-world depth estimated via a stereo camera is noisy, which introduced an additional domain gap. To compensate for this, we tried training with the indoor depth noise model provided by Habitat (*116*), because we did not have any noise model tuned specifically for our robot's Intel RealSense D435i camera. We found this to hurt real-world performance, suggesting that this noise model does not match our robot's camera noise. Although we could easily swap a different segmentation model in modular semantic mapping methods, changing the segmentation model caused a large domain gap for end-to-end methods and required training a new policy.

In contrast to sim, robot discrete actions are not deterministic in the real world because of actuation noise: A 25-cm forward move or 30° turn command will not have the exact intended effect. The usual way to compensate for this is to simulate actuation noise during training. We trained an additional end-to-end navigation policy with simulated actuation noise and ran additional evaluation experiments in the real-world home that we used for the controlled study. We leveraged an actuation noise model derived from motion capture–based benchmarking by the PyRobot authors (*117*). We found this to hurt real-world performance, dropping from 0.30 to a 0.20 success rate. This result is consistent with a prior sim-to-real study of point goal navigation (*81*) and suggests that our chosen noise model may not reflect conditions in reality.

Last, although we can easily inspect the output of various modules, such as the semantic map, the exploration goals, or the plans, to diagnose issues when transferring modular methods, our only recourse when transferring end-to-end policies is to try matching training and evaluation inputs as closely as possible. All these considerations make transferring end-to-end policies much more challenging than modular methods.

Supplementary Materials

This PDF file includes:

Figs. S1 to S7
Tables S1 to S6

REFERENCES AND NOTES

1. P. Anderson, A. Chang, D. S. Chaplot, A. Dosovitskiy, S. Gupta, V. Koltun, J. Kosecka, J. Malik, R. Mottaghi, M. Savva, A. R. Zamir, On evaluation of embodied navigation agents. arXiv:1807.06757 [cs.AI] (18 July 2018).
2. H. P. Moravec, Obstacle avoidance and navigation in the real world by a seeing robot rover, thesis, Stanford University, Palo Alto, CA (1980).
3. R. Chatila, J.-P. Laumond, "Position referencing and consistent world modeling for mobile robots," in *Proceedings of the 1985 IEEE International Conference on Robotics and Automation*, St. Louis, MO, 25 to 28 March 1985 (IEEE, 1985); vol. 2, pp. 138–145.
4. A. Elfes, Sonar-based real-world mapping and navigation. *IEEE J. Robot. Autom.* **3**, 249–265 (1987).
5. A. Elfes, Using occupancy grids for mobile robot perception and navigation. *Computer* **22**, 46–57 (1989).
6. B. Kuipers, Y.-T. Byun, A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations. *Rob. Auton. Syst.* **8**, 47–63 (1991).
7. I. J. Cox, Blanche—An experiment in guidance and navigation of an autonomous robot vehicle. *IEEE Trans. Robot. Autom.* **7**, 193–204 (1991).
8. J. J. Leonard, H. F. Durrant-Whyte, I. J. Cox, Dynamic map building for an autonomous mobile robot. *Int. J. Robot. Res.* **11**, 286–298 (1992).
9. D. Fox, W. Burgard, S. Thrun, The dynamic window approach to collision avoidance. *IEEE Robot. Autom. Mag.* **4**, 23–33 (1997).
10. W. Burgard, A. B. Cremers, D. Fox, D. Hähnel, G. Lakemeyer, D. Schulz, W. Steiner, S. Thrun, Experiences with an interactive museum tour-guide robot. *Artif. Intell.* **114**, 3–55 (1999).
11. S. Thrun, M. Bennewitz, W. Burgard, A. B. Cremers, F. Dellaert, D. Fox, D. Hahnel, C. Rosenberg, N. Roy, J. Schulte, D. Schulz, "MINERVA: A second-generation museum tour-guide robot," in *Proceedings 1999 IEEE International Conference on Robotics and Automation*, Detroit, MI, 10 to 15 May 1999 (IEEE, 1999); vol. 3.
12. S. Thrun, D. Fox, W. Burgard, F. Dellaert, Robust Monte Carlo localization for mobile robots. *Artif. Intell.* **128**, 99–141 (2001).
13. S. Thrun, Robotic mapping: A survey, in *Exploring Artificial Intelligence in the New Millennium* (Morgan Kaufmann Publishers Inc., 2002), pp. 1–35.
14. R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, A. Fitzgibbon, "Kinectfusion: Real-time dense surface mapping and tracking," in *Proceedings of the 2011 10th IEEE International Symposium on Mixed and Augmented Reality*, Basel, Switzerland, 26 to 29 October 2011 (IEEE, 2011), pp. 127–136.
15. A. J. Davison, I. D. Reid, N. D. Molton, O. Stasse, MonoSLAM: Real-time single camera SLAM. *IEEE Trans. Pattern Anal. Mach. Intell.* **29**, 1052–1067 (2007).
16. E. S. Jones, S. Soatto, Visual-inertial navigation, mapping and localization: A scalable real-time causal approach. *Int. J. Robot. Res.* **30**, 407–430 (2011).
17. T. Sattler, W. Maddern, C. Toft, A. Torii, L. Hammarstrand, E. Stenborg, D. Safari, M. Okutomi, M. Pollefeys, J. Sivic, F. Kahl, T. Pajdla, "Benchmarking 6dof outdoor visual localization in changing conditions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, 18 to 23 June 2018 (IEEE, 2018), pp. 8601–8610.
18. 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'*, Monterey, CA, 10 to 11 July 1997 (IEEE, 1997), pp. 146–151.
19. A. Flint, D. Murray, I. Reid, "Manhattan scene understanding using monocular, stereo, and 3d features," in *Proceedings of the 2011 International Conference on Computer Vision*, Barcelona, Spain, 6 to 13 November 2011 (IEEE, 2011), pp. 2228–2235.
20. A. Kundu, Y. Li, F. Dellaert, F. Li, J. M. Rehg, Joint semantic segmentation and 3d reconstruction from monocular video, in *European Conference on Computer Vision* (Springer, 2014), pp. 703–718.
21. S. L. Bowman, N. Atanasov, K. Daniilidis, G. J. Pappas, "Probabilistic data association for semantic SLAM," in *Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA)*, Singapore, 29 May to 3 June 2017 (IEEE, 2017), pp. 1722–1729.
22. L. Ma, J. Stückler, C. Kerl, D. Cremers, "Multi-view deep learning for consistent semantic mapping with RGB-D cameras," in *Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE, 2017), pp. 598–605.
23. L. Zhang, L. Wei, P. Shen, W. Wei, G. Zhu, J. Song, Semantic SLAM based on object detection and improved octomap. *IEEE Access* **6**, 75545–75559 (2018).
24. A. Rosinol, M. Abate, Y. Chang, L. Carlone, "Kimera: An open-source library for real-time metric-semantic localization and mapping," in *Proceedings of the 2020 IEEE International Conference on Robotics and Automation (ICRA)*, Paris, France, 31 May to 31 August 2020 (IEEE, 2020), pp. 1689–1696.
25. R. F. Salas-Moreno, R. A. Newcombe, H. Strasdat, P. H. Kelly, A. J. Davison, "Slam++: Simultaneous localisation and mapping at the level of objects," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, Portland, OR, 23 to 28 June 2013 (IEEE, 2013), pp. 1352–1359.
26. D. A. Pomerleau, Alvinn: An autonomous land vehicle in a neural network, in *Advances in Neural Information Processing Systems* (Morgan Kaufmann Publishers Inc., 1988), vol. 1.
27. U. Muller, J. Ben, E. Cosatto, B. Flepp, Y. Cun, Off-road obstacle avoidance through end-to-end learning, in *Advances in Neural Information Processing Systems* (MIT Press, 2005), vol. 18.

Downloaded from https://www.science.org at The Hong Kong University of Science and Technology (Guangzhou) on May 25, 2026

28. V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, D. Hassabis, Human-level control through deep reinforcement learning. *Nature* **518**, 529–533 (2015).
29. T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, D. Wierstra, Continuous control with deep reinforcement learning. arXiv:1509.02971 [cs.LG] (9 September 2015).
30. G. Lample, D. S. Chaplot, Playing FPS games with deep reinforcement learning, in *The Thirty-First AAAI Conference on Artificial Intelligence (AAAI)* (AAAI, 2017); 10.1609/aaai.v31i1.10827.
31. Y. Zhu, R. Mottaghi, E. Kolve, J. J. Lim, A. Gupta, L. Fei-Fei, A. Farhadi, "Target-driven visual navigation in indoor scenes using deep reinforcement learning," in *Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA)*, Singapore, 29 May to 03 June 2017 (IEEE, 2017), pp. 3357–3364.
32. P. Mirowski, R. Pascanu, F. Viola, H. Soyer, A. Ballard, A. Banino, M. Denil, R. Goroshin, L. Sifre, K. Kavukcuoglu, D. Kumaran, R. Hadsell, Learning to navigate in complex environments," paper presented at the 5th International Conference on Learning Representations, Toulon, France, 24 to 26 2017 (ICLR, 2017).
33. A. Dosovitskiy, V. Koltun, "Learning to act by predicting the future," paper presented at the 5th International Conference on Learning Representations, Toulon, France, 24 to 26 2017 (ICLR, 2017).
34. D. S. Chaplot, G. Lample, "Arnold: An autonomous agent to play fps games," in *The Thirty-First AAAI Conference on Artificial Intelligence (AAAI)* (AAAI, 2017).
35. M. Savva, A. X. Chang, A. Dosovitskiy, T. Funkhouser, V. Koltun, MINOS: Multimodal indoor simulator for navigation in complex environments. arXiv:1712.03931 [cs.LG] (11 December 2017).
36. K. M. Hermann, F. Hill, S. Green, F. Wang, R. Faulkner, H. Soyer, D. Szepesvari, W. M. Czarnecki, M. Jaderberg, D. Teplyashin, M. Wainwright, C. Apps, D. Hassabis, P. Blunsom, Grounded language learning in a simulated 3D world. arXiv:1706.06551 [cs.CL] (20 June 2017).
37. D. S. Chaplot, K. M. Sathyendra, R. K. Pasumarthi, D. Rajagopal, R. Salakhutdinov, Gated-attention architectures for task-oriented language grounding. arXiv:1706.07230 [cs.LG] (22 June 2017).
38. P. Mirowski, M. K. Grimes, M. Malinowski, K. M. Hermann, K. Anderson, D. Teplyashin, K. Simonyan, K. Kavukcuoglu, A. Zisserman, R. Hadsell, Learning to navigate in cities without a map, in *Advances in Neural Information Processing Systems* (Curran Associates Inc., 2018), vol. 31, pp. 2419–2430.
39. F. Codevilla, M. Müller, A. López, V. Koltun, A. Dosovitskiy, "End-to-end driving via conditional imitation learning," in *Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA)*, Brisbane, Queensland, Australia, 21 to 25 May 2018 (IEEE, 2018), pp. 4693–4700.
40. A. Banino, C. Barry, B. Uria, C. Blundell, T. Lillicrap, P. Mirowski, A. Pritzel, M. J. Chadwick, T. Degris, J. Modayil, G. Wayne, H. Soyer, F. Viola, B. Zhang, R. Goroshin, N. Rabinowitz, R. Pascanu, C. Beattie, S. Petersen, A. Sadik, S. Gaffney, H. King, K. Kavukcuoglu, D. Hassabis, R. Hadsell, D. Kumaran, Vector-based navigation using grid-like representations in artificial agents. *Nature* **557**, 429–433 (2018).
41. J. Ye, D. Batra, A. Das, E. Wijmans, "Auxiliary tasks and exploration enable objectgoal navigation," in *Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision*, Montreal, Quebec, Canada, 10 to 17 October 2021 (IEEE, 2021), pp. 16117–16126.
42. O. Maksymets, V. Cartillier, A. Gokaslan, E. Wijmans, W. Galuba, S. Lee, D. Batra, "THDA: Treasure hunt data augmentation for semantic navigation," in *Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision*, Montreal, Quebec, Canada, 10 to 17 October 2021 (IEEE, 2021), pp. 15374–15383.
43. R. Ramrakhya, E. Undersander, D. Batra, A. Das, "Habitat-Web: Learning embodied object-search strategies from human demonstrations at scale," in *Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, New Orleans, LA, 18-24 June 2022 (IEEE, 2022), pp. 5173–5183.
44. E. Wijmans, A. Kadian, A. Morcos, S. Lee, I. Essa, D. Parikh, M. Savva, D. Batra, DD-PPO: Learning near-perfect PointGoal navigators from 2.5 billion frames. arXiv:1911.00357 [cs.CV] (1 November 2019).
45. M. Deitke, E. V. Bilt, A. Herrasti, L. Weihs, J. Salvador, K. Ehsani, W. Han, E. Kolve, A. Farhadi, A. Kembhavi, R. Mottaghi, ProcTHOR: Large-scale embodied AI using procedural generation. arXiv:2206.06994 [cs.AI] (14 June 2022).
46. R. Brooks, A robust layered control system for a mobile robot. *IEEE J. Robot. Autom.* **2**, 14–23 (1986).
47. S. Gupta, J. Davidson, S. Levine, R. Sukthankar, J. Malik, "Cognitive mapping and planning for visual navigation," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, 21 to 26 July 2017 (IEEE, 2017), pp. 2616–2625.
48. E. Parisotto, R. Salakhutdinov, Neural map: Structured memory for deep reinforcement learning, in *International Conference on Learning Representations (ICLR)* (2018).
49. D. S. Chaplot, E. Parisotto, R. Salakhutdinov, Active neural localization, in *International Conference on Learning Representations (ICLR)* (2018).
50. J. F. Henriques, A. Vedaldi, "Mapnet: An allocentric spatial memory for mapping environments," in *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, 18 to 23 June 2018 (IEEE, 2018), pp. 8476–8484.
51. D. Gordon, A. Kembhavi, M. Rastegari, J. Redmon, D. Fox, A. Farhadi, "Iqa: Visual question answering in interactive environments," in *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, 18 to 23 June 2018 (IEEE, 2018), pp. 4089–4098.
52. W. Yang, X. Wang, A. Farhadi, A. Gupta, R. Mottaghi, Visual semantic navigation using scene priors. arXiv:1810.06543 [cs.CV] (15 October 2018).
53. N. Savinov, A. Dosovitskiy, V. Koltun, "Semi-parametric topological memory for navigation," paper presented at the 6th International Conference on Learning Representations (ICLR 2018), Vancouver, British Columbia, Canada, 30 April to 3 May, 2018.
54. N. Savinov, A. Raichuk, D. Vincent, R. Marinier, M. Pollefeys, T. P. Lillicrap, S. Gelly, "Episodic curiosity through reachability," paper presented at the 7th International Conference on Learning Representations (ICLR 2019), New Orleans, LA, 6 to 9 May, 2019.
55. T. Campari, L. Lamanna, P. Traverso, L. Serafini, L. Ballan, "Online learning of reusable abstract models for object goal navigation," in *Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, New Orleans, LA, 18 to 24 June 2022 (IEEE, 2022), pp. 14870–14879.
56. R. McAllister, Y. Gal, A. Kendall, M. van der Wilk, A. Shah, R. Cipolla, A. Weller, "Concrete problems for autonomous vehicle safety: Advantages of Bayesian deep learning," paper presented at the Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, Melbourne, Australia, 19 to 25 August 2017.
57. M. Müller, A. Dosovitskiy, B. Ghanem, V. Koltun, Driving policy transfer via modularity and abstraction. arXiv:1804.09364 [cs.RO] (25 April 2018).
58. D. Scaramuzza, M. C. Achtelik, L. Doitsidis, F. Friedrich, E. Kosmatopoulos, A. Martinelli, M. W. Achtelik, M. Chli, S. Chatzichristofis, L. Kneip, D. Gurdan, L. Heng, G. H. Lee, S. Lynen, M. Pollefeys, A. Renzaglia, R. Siegwart, J. C. Stumpf, P. Tanskanen, C. Troiani, S. Weiss, L. Meier, Vision-controlled micro flying robots: From system design to autonomous navigation and mapping in GPS-denied environments. *IEEE Robot. Autom. Mag.* **21**, 26–40 (2014).
59. A. Mousavian, C. Eppner, D. Fox, "6-dof graspnet: Variational grasp generation for object manipulation," in *Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, Seoul, South Korea, 27 October–2 November 2019 (IEEE, 2019), pp. 2901–2910.
60. J. Mahler, J. Liang, S. Niyaz, M. Laskey, R. Doan, X. Liu, J. A. Ojea, K. Goldberg, Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics. arXiv:1703.09312 [cs.RO] (27 March 2017).
61. D. Morrison, A. W. Tow, M. McTaggart, R. Smith, N. Kelly-Boxall, S. Wade-Mc Cue, J. Erskine, R. Grinover, A. Gurman, T. Hunn, D. Lee, A. Milan, T. Pham, G. Rallos, A. Razjigaev, T. Rowntree, K. Vijay, Z. Zhuang, C. Lehnert, I. Reid, P. Corke, J. Leitner, "Cartman: The low-cost cartesian manipulator that won the amazon robotics challenge," in *Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA)*, Brisbane, Queensland, Australia, 21 to 25 May 2018 (IEEE, 2018), pp. 7757–7764.
62. D. S. Chaplot, D. Gandhi, S. Gupta, A. Gupta, R. Salakhutdinov, "Learning To Explore Using Active Neural SLAM," paper presented at the 8th International Conference on Learning Representations (ICLR 2020), Addis Ababa, Ethiopia, 26 to 30 April 2020.
63. D. S. Chaplot, D. P. Gandhi, A. Gupta, R. R. Salakhutdinov, Object goal navigation using goal-oriented semantic exploration, in *Advances in Neural Information Processing Systems* (Curran Associates Inc., 2020), vol. 33, p. 4247.
64. S. K. Ramakrishnan, Z. Al-Halah, K. Grauman, Occupancy anticipation for efficient exploration and navigation, in *European Conference on Computer Vision* (Springer, 2020), pp. 400–418.
65. D. S. Chaplot, R. Salakhutdinov, A. Gupta, S. Gupta, "Neural topological SLAM for visual navigation," paper presented at the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, 13 to 19 June 2020.
66. S. K. Ramakrishnan, D. S. Chaplot, Z. Al-Halah, J. Malik, K. Grauman, PONI: Potential functions for ObjectGoal navigation with interaction-free learning, in *2022 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE, 2022).
67. M. Hahn, D. S. Chaplot, S. Tulsiani, M. Mukadam, J. M. Rehg, A. Gupta, No RL, no simulation: Learning to navigate without navigating, in *Advances in Neural Information Processing Systems* (Curran Associates Inc., 2021), vol. 34.
68. J. Krantz, A. Gokaslan, D. Batra, S. Lee, O. Maksymets, "Waypoint models for instruction-guided navigation in continuous environments," paper presented at the 2021 IEEE/CVF

Downloaded from https://www.science.org at The Hong Kong University of Science and Technology (Guangzhou) on May 25, 2026

International Conference on Computer Vision (ICCV), Montreal, Quebec, Canada, 10 to 17 October 2021.
69. D. An, Z. Wang, Y. Li, Y. Wang, Y. Hong, Y. Huang, L. Wang, J. Shao, 1st place solutions for RxR-Habitat Vision-and-Language Navigation Competition (CVPR 2022). arXiv:2206.11610 [cs.CV] (23 June 2022).
70. S. Y. Min, D. S. Chaplot, P. Ravikumar, Y. Bisk, R. Salakhutdinov, "FILM: Following Instructions in Language with Modular methods," paper presented at the International Conference on Learning Representations (ICLR 2022), 25 to 29 April 2022, Virtual.
71. X. Liu, H. Palacios, C. Muise, A planning based neural-symbolic approach for embodied instruction following. *Interactions* **9**, 17 (2022).
72. M. Murray, M. Cakmak, Following Natural Language Instructions for Household Tasks with Landmark Guided Search and Reinforced Pose Adjustment. *IEEE Robot. Autom. Lett.* **7**, 6870–6877 (2022).
73. G. Sarch, Z. Fang, A. W. Harley, P. Schydlo, M. J. Tarr, S. Gupta, K. Fragkiadaki, "TIDEE: Tidying up novel rooms using visuo-semantic commonsense priors," paper presented at the 17th European Conference on Computer Vision (ECCV 2022), Tel Aviv, Israel, 23 to 27 October 2022.
74. B. Trabucco, G. Sigurdsson, R. Piramuthu, G. S. Sukhatme, R. Salakhutdinov, A simple approach for visual rearrangement: 3D mapping and semantic search. arXiv:2206.13396 [cs.CV] (21 June 2022).
75. D. S. Chaplot, H. Jiang, S. Gupta, A. Gupta, Semantic Curiosity for Active Visual Learning, paper presented at the Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, 23 to 28 August 2020.
76. D. S. Chaplot, M. Dalal, S. Gupta, J. Malik, R. Salakhutdinov, SEAL: Self-supervised Embodied Active Learning using exploration and 3D consistency, in *Advances in Neural Information Processing Systems* (Curran Associates Inc., 2021).
77. J. Duan, S. Yu, H. L. Tan, H. Zhu, C. Tan, A survey of embodied ai: From simulators to research tasks. *IEEE Trans. Emerg. Topics Comput. Intell.* **6**, 230–244 (2022).
78. D. Mishkin, A. Dosovitskiy, V. Koltun, Benchmarking classic and learned navigation in complex 3D environments. arXiv:1901.10915 [cs.CV] (30 January 2019).
79. M. Savva, Z. Kira, G. A. Regib, J. Yoo, R. Chen, J. Zheng, "Habitat: A platform for embodied AI research," paper presented at the 2019 International Conference on Computer Vision (ICCV 2019), Seoul, South Korea, 27 October to 2 November 2019.
80. E. Kolve, R. Mottaghi, W. Han, E. V. Bilt, L. Weihs, A. Herrasti, M. Deitke, K. Ehsani, D. Gordon, Y. Zhu, A. Kembhavi, A. Gupta, A. Farhadi, AI2-THOR: An interactive 3D environment for visual AI. arXiv:1712.05474 [cs.CV] (26 August 2022).
81. A. Kadian, J. Truong, A. Gokaslan, A. Clegg, E. Wijmans, S. Lee, M. Savva, S. Chernova, D. Batra, Sim2real predictivity: Does evaluation in simulation predict real-world performance? *IEEE Robot. Autom. Lett.* **5**, 6670–6677 (2020).
82. J. Truong, S. Chernova, D. Batra, Bi-directional domain adaptation for sim2real transfer of embodied navigation agents. *IEEE Robot. Autom. Lett.* **6**, 2634–2641 (2021).
83. J. Truong, M. Rudolph, N. Yokoyama, S. Chernova, D. Batra, A. Rai, Rethinking Sim2Real: Lower fidelity simulation leads to higher Sim2Real transfer in navigation. arXiv:2207.10821 [cs.RO] (21 July 2022).
84. Z. Fu, A. Kumar, A. Agarwal, H. Qi, J. Malik, D. Pathak, "Coupling vision and proprioception for navigation of legged robots," in *Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, New Orleans, LA, 19 to 20 June 2022 (IEEE, 2022), pp. 17273–17283.
85. T. Miki, J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, M. Hutter, Learning robust perceptive locomotion for quadrupedal robots in the wild. *Sci. Robot.* **7**, eabk2822 (2022).
86. R. Partsey, E. Wijmans, N. Yokoyama, O. Dobosevych, D. Batra, O. Maksymets, "Is mapping necessary for realistic PointGoal navigation?," in *Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, New Orleans, LA, 19 to 20 June 2022 (IEEE, 2022), pp. 17232–17241.
87. D. Shah, B. Eysenbach, G. Kahn, N. Rhinehart, S. Levine, "Ving: Learning open-world navigation with visual goals," in *Proceedings of the 2021 IEEE International Conference on Robotics and Automation (ICRA)*, Xi'an, China, 30 May to 5 June 2021 (IEEE, 2021), pp. 13215–13222.
88. D. Shah, S. Levine, ViKiNG: Vision-based kilometer-scale navigation with geographic hints. arXiv:2202.11271 [cs.RO] (2022).
89. P. Anderson, Q. Wu, D. Teney, J. Bruce, M. Johnson, N. Sünderhauf, I. Reid, S. Gould, A. van den Hengel, "Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments," in *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, 18 to 23 June 2018 (IEEE, 2018), pp. 3674–3683.
90. S. Höfer, K. Bekris, A. Handa, J. C. Gamboa, M. Mozifian, F. Golemo, C. Atkeson, D. Fox, K. Goldberg, J. Leonard, C. K. Liu, J. Peters, S. Song, P. Welinder, M. White, Sim2Real in robotics and automation: Applications and challenges. *IEEE Trans. Autom. Sci. Eng.* **18**, 398–400 (2021).
91. A. Loquercio, E. Kaufmann, R. Ranftl, M. Müller, V. Koltun, D. Scaramuzza, Learning high-speed flight in the wild. *Sci. Robot.* **6**, eabg5810 (2021).
92. C. C. Kemp, A. Edsinger, H. M. Clever, B. Matulevich, "The design of Stretch: A compact, lightweight mobile manipulator for indoor human environments," in *Proceedings of the 2022 International Conference on Robotics and Automation (ICRA)*, Philadelphia, PA, 23 to 27 May 2022 (IEEE, 2022), pp. 3150–3157.
93. D. Batra, A. Gokaslan, A. Kembhavi, O. Maksymets, R. Mottaghi, M. Savva, A. Toshev, E. Wijmans, ObjectNav revisited: On evaluation of embodied agents navigating to objects. arXiv:2006.13171 [cs.CV] (23 June 2020).
94. K. Yadav, R. Ramrakhya, S. K. Ramakrishnan, T. Gervet, J. Turner, A. Gokaslan, N. Maestre, A. X. Chang, D. Batra, M. Savva, A. W. Clegg, D. S. Chaplot, Habitat-Matterport 3D semantics dataset. arXiv:2210.05633 [cs.CV] (11 October 2022).
95. S. K. Ramakrishnan, A. Gokaslan, E. Wijmans, O. Maksymets, A. Clegg, J. Turner, E. Undersander, W. Galuba, A. Westbury, A. X. Chang, M. Savva, Y. Zhao, D. Batra, Habitat-matterport 3D dataset (HM3D): 1000 large-scale 3D environments for embodied AI. arXiv:2109.08238 [cs.CV] (16 September 2021).
96. X. Zhou, R. Girdhar, A. Joulin, P. Krähenbühl, I. Misra, Detecting twenty-thousand classes using image-level supervision. arXiv:2201.02605 [cs.CV] (7 January 2022).
97. W. Li, X. Song, Y. Bai, S. Zhang, S. Jiang, "ION: Instance-level Object Navigation," in *Proceedings of the 29th ACM International Conference on Multimedia*, Virtual Event, China, 20 to 24 October 2021 (Association for Computing Machinery, 2021), pp. 4343–4352.
98. A. Khandelwal, L. Weihs, R. Mottaghi, A. Kembhavi, "Simple but effective: Clip embeddings for embodied AI," in *Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, New Orleans, LA, 18 to 24 June 2022 (IEEE, 2022), pp. 14809–14818.
99. K. Fang, A. Toshev, L. Fei-Fei, S. Savarese, "Scene memory transformer for embodied agents in long-horizon tasks," paper presented at the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, 15 to 20 June 2019.
100. M. Zhu, B. Zhao, T. Kong, "Navigating to objects in unseen environments by distance prediction," paper presented at the 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Kyoto, Japan, 23 to 27 October 2022.
101. L. Pinto, A. Gupta, "Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours," in *Proceedings of the 2016 IEEE international conference on robotics and automation (ICRA)*, Stockholm, Sweden, 16 to 21 May 2016 (IEEE, 2016), pp. 3406–3413.
102. D. Kalashnikov, A. Irpan, P. Pastor, J. Ibarz, A. Herzog, E. Jang, D. Quillen, E. Holly, M. Kalakrishnan, V. Vanhoucke, S. Levine, "Scalable deep reinforcement learning for vision-based robotic manipulation," paper presented at the 2nd Annual Conference on Robot Learning (CoRL 2018), Zürich, Switzerland, 29 to 31 October 2018, pp. 651–673.
103. J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," in *Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Vancouver, British Columbia, Canada, 24 to 28 September 2017 (IEEE, 2017), pp. 23–30.
104. O. M. Andrychowicz, B. Baker, M. Chociej, R. Józefowicz, B. M. Grew, J. Pachocki, A. Petron, M. Plappert, G. Powell, A. Ray, J. Schneider, S. Sidor, J. Tobin, P. Welinder, L. Weng, W. Zaremba, Learning dexterous in-hand manipulation. *Int. J. Robot. Res.* **39**, 3–20 (2020).
105. E. Kaufmann, A. Loquercio, R. Ranftl, M. Müller, V. Koltun, D. Scaramuzza, Deep drone acrobatics. arXiv:2006.05768 [cs.RO] (10 June 2020).
106. A. Szot, A. Clegg, E. Undersander, E. Wijmans, Y. Zhao, J. Turner, N. Maestre, M. Mukadam, D. Chaplot, O. Maksymets, A. Gokaslan, V. Vondrus, S. Dharur, F. Meier, W. Galuba, A. Chang, Z. Kira, V. Koltun, J. Malik, M. Savva, D. Batra, Habitat 2.0: Training home assistants to rearrange their habitat, in *Advances in Neural Information Processing Systems* (Curran Associates Inc., 2021), vol. 34, p. 251.
107. MetaAI, Fairo: A modular embodied agent architecture and platform for building embodied agents (2021); https://github.com/facebookresearch/fairo.
108. S. Kohlbrecher, J. Meyer, O. von Stryk, U. Klingauf, "A flexible and scalable SLAM system with full 3D motion estimation," paper presented at 2011 IEEE International Symposium on Safety, Security, and Rescue Robotics, Kyoto, Japan, 1 to 5 November 2011.
109. K. He, G. Gkioxari, P. Dollár, R. Girshick, "Mask R-CNN," in *Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV 2017)*, Venice, Italy, 22 to 29 October 2017 (IEEE, 2017), pp. 2980–2988.
110. K. He, X. Zhang, S. Ren, J. Sun, "Deep residual learning for image recognition," in *Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, NV, 27 to 30 June 2016 (IEEE, 2016), pp. 770–778.
111. F. Xia, A. R. Zamir, Z. He, A. Sax, J. Malik, S. Savarese, "Gibson Env: Real-world perception for embodied agents," in *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, 18 to 23 June 2018 (IEEE, 2018), pp. 9068–9079.
112. J. A. Sethian, A fast marching level set method for monotonically advancing fronts. *Proc. Natl. Acad. Sci. U.S.A.* **93**, 1591–1595 (1996).

Downloaded from https://www.science.org at The Hong Kong University of Science and Technology (Guangzhou) on May 25, 2026

113. J. Chung, C. Gulcehre, K. Cho, Y. Bengio, Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv:1412.3555 [cs.NE] (11 December 2014).
114. J. Jiang, L. Zheng, F. Luo, Z. Zhang, Rednet: Residual encoder-decoder network for indoor rgb-d semantic segmentation. arXiv:1806.01054 [cs.CV] (4 June 2018).
115. S. Song, S. P. Lichtenberg, J. Xiao, "Sun RGB-D: A RGB-D scene understanding benchmark suite," in *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, 7 to 12 June 2015 (IEEE, 2015), pp. 567–576.
116. S. Choi, Q.-Y. Zhou, V. Koltun, "Robust reconstruction of indoor scenes," in *Proceedings of 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, 7 to 12 June 2015 (IEEE, 2015), pp. 5556–5565.
117. A. Murali, T. Chen, K. V. Alwala, D. Gandhi, L. Pinto, S. Gupta, A. Gupta, PyRobot: An open-source robotics framework for research and benchmarking. arXiv:1906.08236 [cs.RO] (19 June 2019).

Acknowledgments: We thank S. Gupta for reviewing the manuscript and B. Trabucco for voice recording the video. **Funding:** This work was supported by Meta AI Research. **Author contributions:** T.G. and D.S.C. identified the problem and designed experiments with input from J.M. T.G. implemented the methods with input from S.C. T.G. performed experiments. T.G. and D.S.C. analyzed the data and drew conclusions with input from J.M. and D.B. T.G. wrote the paper with input from D.S.C., J.M., and D.B. T.G. and D.S.C. designed the figures and video. **Competing interests:** The authors declare that they have no competing interests. **Data and materials availability:** All data needed to evaluate the conclusions in the paper are present in the paper or the Supplementary Materials. The project website is https://theophilegervet.github.io/projects/real-world-object-navigation.

Submitted 9 November 2022
Accepted 29 May 2023
Published 28 June 2023
10.1126/scirobotics.adf6991

Downloaded from https://www.science.org at The Hong Kong University of Science and Technology (Guangzhou) on May 25, 2026

Science Robotics

Navigating to objects in the real world

Theophile Gervet, Soumith Chintala, Dhruv Batra, Jitendra Malik, and Devendra Singh Chaplot

Sci. Robot. **8** (79), eadf6991. DOI: 10.1126/scirobotics.adf6991

View the article online
https://www.science.org/doi/10.1126/scirobotics.adf6991
Permissions
https://www.science.org/help/reprints-and-permissions

Downloaded from https://www.science.org at The Hong Kong University of Science and Technology (Guangzhou) on May 25, 2026

Use of this article is subject to the Terms of service

Science Robotics (ISSN 2470-9476) is published by the American Association for the Advancement of Science. 1200 New York Avenue NW, Washington, DC 20005. The title *Science Robotics* is a registered trademark of AAAS.

Copyright © 2023 The Authors, some rights reserved; exclusive licensee American Association for the Advancement of Science. No claim to original U.S. Government Works