Iterative Vision-and-Language Navigation

Jacob Krantz[1]* Shurjo Banerjee[2]* Wang Zhu[3]
Jason Corso[2] Peter Anderson[4] Stefan Lee[1] Jesse Thomason[3]
[1]Oregon State University [2]University of Michigan [3]University of Southern California [4]Google Research

Abstract

We present Iterative Vision-and-Language Navigation (IVLN), a paradigm for evaluating language-guided agents navigating in a persistent environment over time. Existing Vision-and-Language Navigation (VLN) benchmarks erase the agent's memory at the beginning of every episode, testing the ability to perform cold-start navigation with no prior information. However, deployed robots occupy the same environment for long periods of time. The IVLN paradigm addresses this disparity by training and evaluating VLN agents that maintain memory across tours of scenes that consist of up to 100 ordered instruction-following Room-to-Room (R2R) episodes, each defined by an individual language instruction and a target path. We present discrete and continuous Iterative Room-to-Room (IR2R) benchmarks comprising about 400 tours each in 80 indoor scenes. We find that extending the implicit memory of high-performing transformer VLN agents is not sufficient for IVLN, but agents that build maps can benefit from environment persistence, motivating a renewed focus on map-building agents in VLN.

1. Introduction

Robots and virtual agents that persistently operate in human spaces like homes should improve over time. For example, a smart vacuum told to *clean the living room, which is down the hall past the guest bedroom* should learn about both the living room and guest bedroom. Likewise, agents should be able to associate references in past instructions, such as *guest bedroom*, with spatial and visual information from the environment to understand future instructions.

Most work on language-guided, embodied agents performing navigation [3, 25] or household tasks [38] is *episodic* in nature—agent memory is erased before issuing each new instruction. In contrast, physical robots build maps [12,43,49] *iteratively* from visual observations [32,39] as an explicit form of long-term memory. Agents trained to perform language-guided navigation in simulation that are deployed on physical robots [2] fail to take advantage of the mapping-based strategies that facilitate robot navigation.

We propose Iterative Vision-and-Language Navigation (IVLN), in which an agent follows an *ordered sequence* of language instructions that conduct a *tour* of an indoor space. Each tour is composed of individual *episodes* of language instructions with target paths. Agents can utilize *memory* to better understand future tour instructions. After just 10 episodes an agent has seen on average over 50% of the target path associated with the next language instruction in a tour. While performing an IVLN tour, agents iteratively explore the environment, meaning regions irrelevant to task instructions need not ever be visited. By conditioning exploration on language, IVLN enables rich semantic representations, *e.g*., unusual, novel, and scene-specific referents grounded during one episode can be reasoned about later.

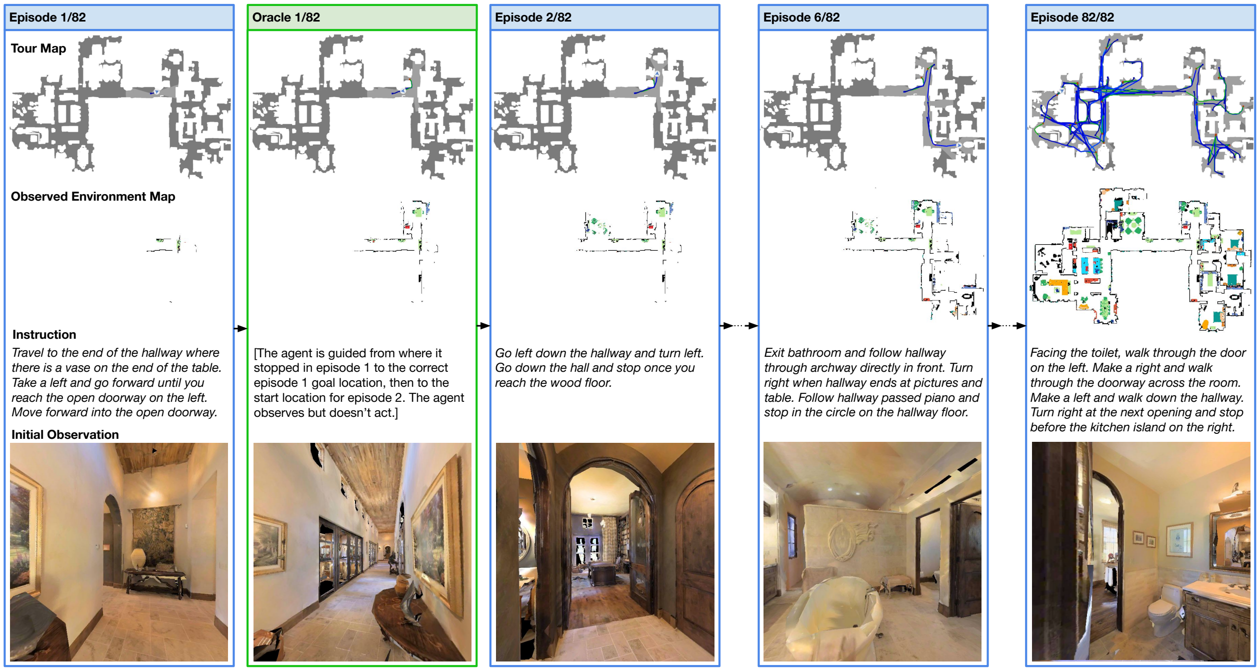
We explore both a discrete VLN setting based on Room-to-Room [3] episodes and navigation graphs (IR2R) and a continuous simulation VLN-CE [25] setting (IR2R-CE). The markedly different action and visual observation spaces of these settings may require different memory mechanisms. In the discrete setting, agents move on graph edges and observe clear, well-framed images. For IR2R, we extend a state-of-the-art transformer agent [11] that learns an implicit memory based on path history when interpreting instructions. In the continuous setting, agents take motion actions while observing noisy images of a 3D environment reconstructed from discrete panorama images. For IR2R-CE, we propose an agent that builds and interprets an explicit semantic map.

In short, we define Iterative Vision-and-Language Navigation (IVLN), a paradigm for persistent VLN, and release IR2R and IR2R-CE to study discrete and continuous navigation agents in the IVLN setting. We create initial agents for both benchmarks, including explicit mapping and implicit memory models for continuous navigation. Please see jacobkrantz.github.io/ivln for code and more details.

2. Related Work

Instruction-guided navigation is a growing area in grounded language understanding with many task settings

*Equal contributions. Correspondence: krantzja@oregonstate.edu



Travel to the end of the hallway where there is a vase on the end of the table. Take a left and go forward until you reach the open doorway on the left. Move forward into the open doorway.

[The agent is guided from where it stopped in episode 1 to the correct episode 1 goal location, then to the start location for episode 2. The agent observes but doesn't act.]

Go left down the hallway and turn left. Go down the hall and stop once you reach the wood floor.

Exit bathroom and follow hallway through archway directly in front. Turn right when hallway ends at pictures and table. Follow hallway passed piano and stop in the circle on the hallway floor.

Facing the toilet, walk through the door on the left. Make a right and walk through the doorway across the room. Make a left and walk down the hallway. Turn right at the next opening and stop before the kitchen island on the right.

Figure 1. In IVLN, agents are given language instructions corresponding to a sequence of paths that form a *tour* around a 3D scene. After attempting to follow each instruction, the agent is teleoperated by an oracle to the correct goal location, then to the start of the next path where the next instruction is issued. Unlike conventional *episodic* paradigms, the agent retains memory between episodes.

developed [3, 9, 26, 33, 38, 42]. Among these, the Vision-and-Language Navigation (VLN) task setting based on the Room-to-Room (R2R) dataset [3] has become a popular benchmark. An agent in VLN must follow a natural language instruction by navigating along the described path in a *never-before-seen environment*. By design, this paradigm does not consider how persistent agents operating over time might leverage prior experiences to better follow future instructions within the same environment. In contrast, accumulating prior experience within an environment is a staple of robotic deployment – e.g. building semantic maps for localization and reasoning [35, 41]. Our IVLN paradigm is designed to better align VLN with a realistic robotic deployment scenario.

Benchmarks for VLN in Discrete Settings VLN tasks frequently involve inferring agent actions in a rendered 2D or 3D scene in response to language commands [8, 28]. Agent control is typically limited to changing position and orientation by discrete amounts or to predefined possible options. Advances in camera technology have enabled language-guided navigation in photorealistic indoor scenes [3, 7] and outdoor city spaces [9]. In "Room-to-Room" (R2R) [3] VLN, an agent interprets a single English instruction to navigate along a short, indoor path. In a survey of VLN modeling methods, environment exploration and memorization were identified as frequent strategies for aligning a language instruction to a desired goal location in a scene [16]. However, R2R evaluates policies on single instructions, limiting the incentive to perform efficient, effective memorization or mapping. To study longer horizon planning, researchers have extended R2R by concatenating language-aligned paths and their associated instructions [21, 51], tasking agents not just with arriving to the goal but with following closely the described path. Others have collected longer paths with instructions in three languages [26] or given as a cooperative conversation [42]. With IR2R *tours*, we present the longest such paths with substantial overlap in areas- covered-before through time, challenging researchers to utilize information from prior instructions and experience in the scene.

Benchmarks for VLN in Continuous Settings Moving a physical robot, such as a quad-copter [5] or a toy car [4], in response to language instructions requires contending with the real, continuous world. Existing work has transferred policies for discrete VLN to the physical world by manually curating a discrete representation of the world map as a navigation graph [2] with limited success. VLN-CE [25] re-introduces Room-to-Room [3] with a *continuous*, 3D reconstruction of indoor MatterPort3D scenes. However, VLN-CE evaluates agents on single instructions and associated paths in an i.i.d. fashion. In contrast, our IR2R-CE benchmark incentivizes policies that respect environment persistence found in the real world. Beyond removing the abstractions of discrete VLN (VLN-CE), IR2R-CE situates agents in a scene for long time horizons with many language instructions; a logical next step towards learning useful world representations through visual and linguistic information.

Pre-Exploration in VLN Some approaches in VLN have embraced a setting where agents can fully explore the environment before following an instruction, either explicitly through pretraining (e.g. [40,44,50]) or through beam-search at inference time (e.g. [14, 30]). Pre-exploration methods outperform standard VLN approaches and serve as a natural upper bound to IVLN where an agent has fully explored the environment. In contrast, IVLN studies how environment information can be collected while performing the task (rather than a priori) and how this partial, opportunistic information can be leveraged to perform better over time.

Persistent Environments in Embodied AI Zooming out, visual navigation tasks in embodied AI have seen significant progress, fueled by increased scale and quality of 3D scene datasets (e.g. [7, 34]) and high-performance simulation platforms (e.g. [23,31,37,48]). A focus on real-world complexity has emerged. One recognition is that agents act in, and interact with, persistent environments. Tasks such as multi-object navigation [45] and visual room rearrangement [46] involve solving sequences of subtasks that, when approached independently, cannot be solved optimally. Instead, reasoning over persistent semantic and spatial information is required. The proposed IVLN paradigm enriches this scene perception problem with natural language and enables the association of persistent visual semantics with linguistic information.

3. Iterative Vision-and-Language Navigation

We facilitate the study of agents given sequential navigation instructions in natural language. We extend the Room-to-Room (R2R) [3] dataset of independent *episodes*—natural language instructions and associated target paths in a particular scene—to *tours*—sequences of many episodes that cover large swaths of the scene and include backtracking. The resulting Iterative Room-to-Room tours contain substantially longer paths and navigation instruction context than prior discrete (IR2R) or continuous (IR2R-CE) VLN benchmarks.

The Iterative Paradigm We define a *tour* to be an ordered sequence of episodes within a scene. Tours alternate between two phases. In the *agent navigation* phase, the agent is given a language instruction and infers navigation actions, equivalent to a VLN *episode*. The phase ends when the agent emits the STOP signal or takes a maximum number of actions. The *oracle navigation* phase immediately follows in two parts. First, if the agent has not successfully navigated to within 0.5m of the episode goal, it is guided without language to that goal by an oracle that forces its actions, analogous to a human teaching the robot where the path should have ended. Second, the agent is oracle-guided to the starting point of the next episode in the tour, analogous to following a human and waiting to receive the next instruction. The agent passively observes the environment during this phase.

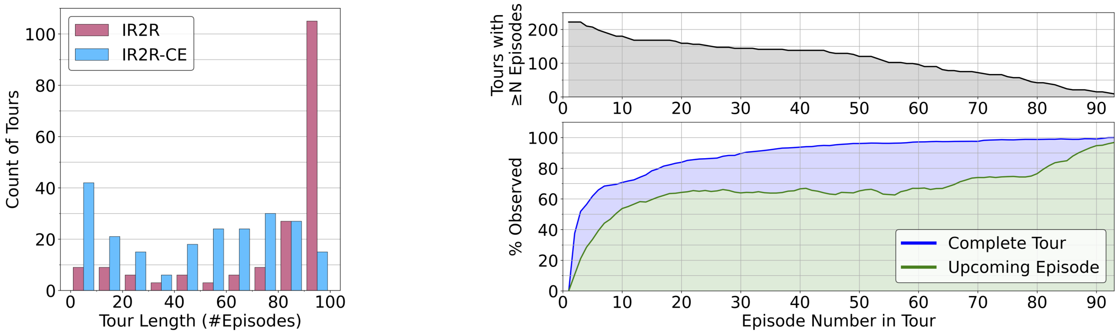
Generating Tours from VLN Data We generate tours that minimize the distance between end and start points of sequential episodes. We also maximize the number of included episodes as path finding between poses can fail in IR2R-CE.

Dataset	Split	Scenes	Episodes	Tours	Tours/ Scene	Tour Length (Episodes)			
						Mean	Min	Max	SD
IR2R	Train	61	14025	183	3.0	76.6	2	99	28.4
	Val-Seen	53	1011	159	3.0	6.4	2	11	2.1
	Val-Unseen	11	2349	33	3.0	71.2	6	100	34.0
IR2R-CE	Train	60	10668	222	3.7	48.1	3	93	30.5
	Val-Seen	50	747	156	3.1	4.8	2	10	2.1
	Val-Unseen	11	1824	36	3.3	50.7	3	100	31.3

Table 1. We construct sequences of episodes—*tours*—from the Room-to-Room dataset [3] to create the discrete IR2R and continuous IR2R-CE benchmarks. Here we detail characteristics of these benchmarks, including the average number of episodes per tour.

Each R2R split contains a set of scenes, which each contain a set of episodes E. For each E, we seek to derive a set of disjoint tours $\mathcal{T}$ where each tour $T \in \mathcal{T}$ is a sequence of episodes that can be inter-navigated. That is, for episode i and $i+1$ in T, navigation from the end of i to the start of $i+1$ is possible. Letting X be the set of unique paths in an episode set E, we first partition $P(X)$ such that the paths in each subset p are inter-navigable; closed doors or obstacles can create disjoint regions in the scene. To determine $P(X)$, we compute the navigable geodesic distance between each path pair where a finite distance implies connectivity. In IR2R, this distance is computed on a navigation graph; in IR2R-CE, it is computed on a 3D navigation mesh and assumes agent dimensions and actions common to VLN-CE [25]. We then order the paths in each subset p to define a tour T. Minimizing the oracle navigation distance in a tour is equivalent to an asymmetric traveling salesperson problem (ATSP) which we approximately solve using the Lin-Kernighan heuristic (LKH) [17]. Finally, if E contains n instructions per path and $n > 1$, we duplicate each tour n times, sampling an instruction for each path without replacement.

Dataset Characteristics We generate tours in the Train, Validation-Seen, and Validation-Unseen splits of discrete R2R to form IR2R and continuous R2R to form IR2R-CE (Tab. 1). Validation-Seen (Val-Seen) contains episodes from scenes seen during training, while Validation-Unseen (Val-Unseen) contains episodes from scenes not seen during training. In total, IR2R contains 375 tours and IR2R-CE contains 414. There are fewer discrete tours, which are longer on average than continuous tours (Fig. 2a), due to discontinuities in the navigable area of continuous environments. In discrete VLN, a path exists from each node to every other node in a scene, but in continuous environments navigation between episode endpoints can fail, resulting in disjoint spaces within a scene that have shorter tours. The distribution of episodes per tour has a high variance for both benchmarks, a reflection



(a) Episodes per tour.

(b) IR2R-CE observation coverage relative to tour progress.

Figure 2. (a) We compare the distributions of tour lengths between the IR2R and IR2R-CE Train splits. (b) We consider an oracle agent following the target paths of each tour in IR2R-CE Train. Before starting each episode, we measure the percentage of that episode's target path observed earlier in the tour (Upcoming Episode). To do this, we compute what percentage of an episodic coverage map is accounted for in a map iteratively constructed by the oracle agent. We also measure what percentage of the entire tour has been observed (Complete Tour).

of path sampling in R2R and a diversity of scene sizes.

Since episodes are chained together in IVLN, locations along the target path for the current episode may have been observed earlier in the tour. In Fig. 2b, we visualize how much of a tour's scene region has been observed relative to the number of episodes performed. We find that following target paths quickly observes a majority of the region; after just 10 episodes, on average, over 50% of the next target path and over 70% of the entire tour region have been observed.

Metrics for Iterative Evaluation Agents should be successful and efficient from the start of a tour and improve as the tour progresses. For iterative (tour-based) evaluation, we adapt the normalized dynamic time warping (nDTW [29]) metric. We select nDTW over success-based metrics for unification across IVLN benchmarks: IR2R and IR2R-CE contain shortest-path priors (all instructions describe the shortest path to the goal), but other datasets that can be converted to IVLN do not, such as RxR [26]. In episodic VLN, the nDTW metric is computed per-episode and averaged across all episodes. However, in IVLN, episodes within a tour need collective evaluation. Otherwise, an agent could exploit per-episode averaging by using the first episode to explore the entire environment, improving subsequent performance at the cost of poor performance in just a single episode—when the number of episodes in a scene is large, this would have minimal impact on averaged metrics.

Formally, given a candidate path Q and a target path R, each consisting of a sequence of 3D points, nDTW computes the dynamic time warping (DTW) cost between Q and R normalized by the number of points in the reference path ($|R|$) and a distance threshold of success (d_{th}):

$$\texttt{nDTW}(R, Q) = \exp\left(-\frac{\texttt{DTW}(R, Q)}{|R| \cdot d_{th}}\right) \tag{1}$$

As in tour generation, we use navigation graph distance in IR2R and geodesic distance in IR2R-CE.

To extend this definition to tours, we make two changes to the episodic nDTW calculation. First, we compute nDTW over tour paths Q^T and R^T instead of episode paths Q and R. The candidate path Q^T includes the points visited during agent navigation phases of tour T, and the target path R^T is the concatenation of the target paths for each episode in T. Oracle navigation is excluded from both Q^T and R^T. Second, to ensure that episode boundaries are respected when aligning candidate and target points in the DTW calculation, candidate and target points from Q^T and R^T are assigned infinite distance unless they belong to the same episode in the tour. This penalty ensures that an agent can't receive credit for completing a path while following a different instruction.

To compute performance for a dataset split, we aggregate nDTW weighted by episode count in each tour T_i, avoiding inflated scores from performing well only on short tours:

$$\texttt{t-nDTW} = \sum_i \frac{|T_i| \cdot \texttt{nDTW}(R^{T_i}, Q^{T_i})}{\sum_j |T_j|} \tag{2}$$

The tour nDTW score (`t-nDTW`) is bounded between 0 and 1, with 1 indicating perfect alignment of the agent's path and the target path for every episode of every tour in the split. In the following experiments, we report `t-nDTW` scaled between 0 and 100 as is common practice for episodic `nDTW`. `t-nDTW` functions in discrete and continuous environments and serves as the primary metric in IR2R and IR2R-CE. Fig. 3 contains example `t-nDTW` evaluations of an IR2R-CE agent to illustrate the relationship between `t-nDTW` and path alignment. We include more examples in the appendix.

4. Methods

We demonstrate how VLN and VLN-CE baseline models generalize to our iterative task and explore whether adding persistent *tour memory* (either unstructured latent memory or a spatial semantic map) improves performance.

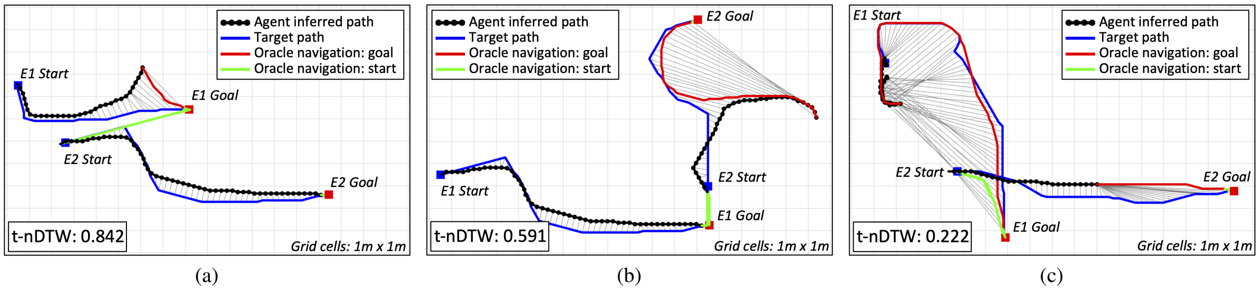


Figure 3. Evaluations in IR2R-CE using t-nDTW for example 2-episode tours. We visualize the DTW alignment of the **agent's inferred path** by its match to the target path. Between episodes, an oracle conveys the agent from inferred episode stop point to true stop, then on to the next episode start. In Fig. 3c, poor performance in episode 1 (E1) drops the overall t-nDTW score significantly.

4.1. VLN Baseline Agents

HAMT We adopt the History-Aware Multimodal Transformer (HAMT) [11] agent. Like many recent methods, HAMT is a transformer-based agent pretrained on proxy tasks and finetuned on VLN. HAMT has three transformer-based encoders: an instruction encoder, a visual encoder for the current observation, and a history encoder for previous state-action pairs. A cross-modal transformer fuses these to predict the next action. The history embedding at time step t is represented as $\{h_{\texttt{CLS}}, h_1, ..., h_{t-1}\}$, where h_t is the features of the state-action pair at step t and $h_{\texttt{CLS}}$ is the features of the `CLS` token used to gather sequence-level information.

TourHAMT We enable tour-level reasoning by including state-action pairs from previous episodes in the history embedding. For episode i, we denote the total number of steps as l_i, including the oracle navigation after termination. We denote the state-action embedding at step t for episode i as h^i_t. At step t in i, we set the history embedding as $\{h_{\texttt{PREV}}, h^1_1, ..., h^1_{l_1}, ..., h^{i-1}_1, ..., h^{i-1}_{l_{i-1}}, h_{\texttt{CLS}}, h^i_1, ..., h^i_{t-1}\}$, where `PREV` is a token delineating episode boundaries. We limit to the latest 50 steps. Unlike the original HAMT, we unfreeze the history encoder to learn this modified history encoding. We train via teacher-forcing with inflection weighting [47] and update gradients per episode in a tour.

4.2. VLN-CE Baseline Agents

The Cross-Modal Attention (CMA) agent defined in VLN-CE [25] is a common baseline in recent works [10, 15, 18–20, 24]. CMA is an end-to-end recurrent model that observes RGBD, the instruction, and the previous action to predict an action from `TURN-LEFT`, `TURN-RIGHT`, `MOVE-FORWARD`, and `STOP`. CMA has a two-GRU structure to track episodic history; one tracks vision and the other tracks general state from which the action is predicted.

4.2.1 Agents with Unstructured Latent Memory

CMA We consider adaptations of the CMA agent's unstructured latent memory to tours. Against these, the original CMA agent is included as a baseline with no cross-episode reasoning ability; that is, hidden states are reset each episode.

TourCMA We reset the hidden state of the vision GRU only at the start of each tour, thereby extending the temporal receptive field to all tour steps. We reset the state GRU episodically. This model provides structure for reasoning over both scene-level vision and episode-specific visuo-linguistics.

PoolCMA We enable both tour and episodic memory within the vision GRU. We reset the hidden state each episode but temporally max-pool the hidden state into a tour-persistent vector reset each tour. This vector is input to the vision GRU.

PoolEndCMA Observations from previous episodes provide utility if they are relevant for planning, *e.g.*, when considering a previously-traversed hallway. Agents may learn to ignore this signal in favor of episode-specific alignments. We encourage this signal by coupling tour memory to action prediction; we concatenate the `PoolCMA` tour memory with the final state vector and use the result to predict an action.

Training Method We train the above models in IR2R-CE using teacher forcing and update parameters following each episode rollout. For tour-persistent memory structures, we disable gradients from steps prior to the current episode. This equates to an adaptive truncated backpropagation through time (TBPTT) where the time step is episode length.

4.2.2 MAP-CMA: Agents with Semantic Maps

We experiment with providing agent-centric, metrically accurate local crops of maps of an agent's surroundings to our models to evaluate the impact of structured memory.

MAP-CMA We build occupancy maps [13] where cells are either 0 (empty) or 1 (occupied), and semantic maps contain one-hot vectors of thirteen common labels in R2R environments [6]. Following [6], we use the inverse pinhole camera projection model to unproject depth measurements to 3D pointclouds. We also unproject egocentric semantics—both ground-truth [7] and from a fine-tuned Rednet feature encoder [6, 22]—to form a semantic pointcloud. We collapse these pointclouds along the height dimension to generate 2D

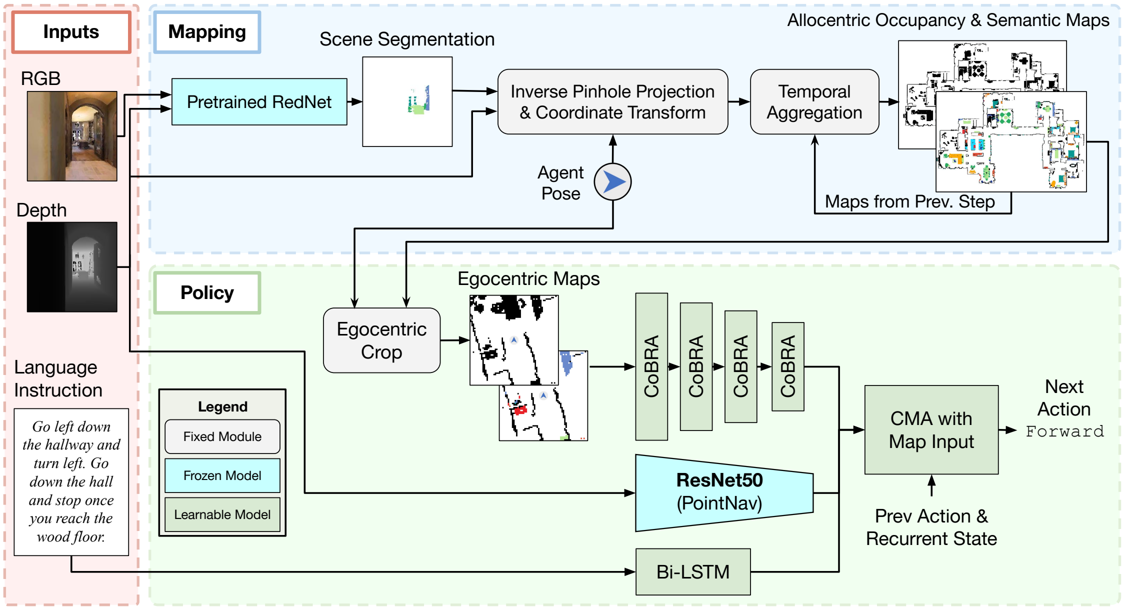


Figure 4. In addition to the encoders for language instructions and depth frames, MAP-CMA model learns an encoding of an egocentric crop of a top-down semantic map of the environment constructed by the agent during navigation in order to predict the next navigation action.

maps. When more than one semantic label exists in a height column, we choose the highest semantic label available to project. Agents may traverse between floors. As such, we constrain pointclouds to project only features lying between the floor and ceiling planes relative to the agent's 3D pose.

We augment the existing CMA architecture with semantic and occupancy maps by replacing the RGB input with maps. Specifically, we channel-wise concatenate the semantic and occupancy maps, encode them through a learned convolutional encoder, and produce a spatial embedding that propagates through CMA in place of RGB features. The map input is $14{\times}64{\times}64$, representing a $64{\times}64$ spatial grid with 13 one-hot semantic channels and one occupancy channel. Four convolutional blocks each consisting of a **Co**nvolution, **B**atch normalization, **Re**LU activation, and **A**verage pool (**CoBRA** in Fig. 4) encode the map to a $128{\times}4{\times}4$ output. These semantic spatial features are used as a drop-in replacement of visual features in CMA as depicted in Fig. 4.

Training Method We train this model in IR2R-CE following the two-step method presented in [25] used to train the CMA model. We initially train using teacher forcing on the augmented EnvDrop [40] ported to IR2R-CE, then finetune the best-performing EnvDrop Val-Unseen checkpoint using DAgger [36] on IR2R-CE Train. The best performing checkpoint on Val-Unseen is accepted as the final model. We use the Progress Monitor [27] auxiliary loss in both training steps. We train with episodic maps (reset each episode), iterative maps (reset each tour), and known maps (pre-computed for each scene). For each temporal construction method, we train one model on ground-truth semantic labels and one model with predicted semantics, resulting in 6 total trained models capable of evaluating the impacts of temporal map construction and semantic segmentation in IR2R-CE.

5. Experiments and Results

We present results on IR2R and IR2R-CE below and center our discussion on key observations.

Evaluation Metrics We use `t-nDTW` (Sec. 3) as our primary metric and report as a percentage. We also include metrics standard to VLN and VLN-CE [1, 29] to describe average single-episode (*episodic*) performance: trajectory length (`TL`), navigation error (`NE`), oracle success (`OS`), normalized dynamic time warping (`nDTW`), success rate (`SR`), and success weighted by inverse path length (`SPL`).

5.1. Unstructured Memory in IR2R and IR2R-CE

Naive extensions of unstructured memory fail to exploit tour information. Across both IR2R and IR2R-CE, we find that the simple extensions of unstructured memory explored in this work fail to improve (and often hurt) tour performance compared to purely episodic agents.

IR2R. Tab. 2 shows results of HAMT and TourHAMT variations in the discrete IR2R setting. We find the episodic HAMT model to be a strong baseline even without any tour memory; our attempts to add tour memory significantly degraded performance. Simply extending the memory reduces `t-NDTW` by a factor of two from HAMT (row 1 vs. 5). Finetuning the history encoding for this extended memory setting (row 4), adding a `PREV` token to separate memory from prior

						Val-Seen							Val-Unseen						
#	Model	PH	TH	PHI	IW	TL	NE ↓	OS ↑	nDTW ↑	SR ↑	SPL ↑	t-nDTW ↑	TL	NE ↓	OS ↑	nDTW ↑	SR ↑	SPL ↑	t-nDTW ↑
1	HAMT					10.1 ±0.1	**4.2** ±0.1	**70** ±1	**71** ±1	**63** ±1	**61** ±1	**58** ±1	9.4 ±0.1	**4.7** ±0.0	**64** ±1	**66** ±0	**56** ±0	**54** ±0	**50** ±0
2	TourHAMT	✓	✓	✓	✓	9.4 ±0.4	5.8 ±0.1	56 ±1	59 ±0	45 ±1	43 ±1	45 ±0	10.0 ±0.2	6.2 ±0.1	52 ±2	52 ±0	39 ±1	36 ±0	32 ±1
3		✓	✓	✓		10.5 ±0.3	6.0 ±0.2	60 ±1	58 ±1	45 ±2	43 ±2	42 ±1	10.9 ±0.2	6.8 ±0.2	54 ±1	51 ±1	38 ±1	34 ±1	31 ±1
4		✓	✓			10.6 ±0.3	6.0 ±0.1	61 ±1	58 ±1	45 ±1	42 ±1	42 ±1	10.3 ±0.3	6.7 ±0.2	52 ±1	50 ±1	38 ±1	34 ±1	29 ±1
5		✓				10.9 ±0.3	6.1 ±0.1	60 ±2	58 ±1	45 ±1	42 ±1	41 ±0	11.0 ±0.6	6.7 ±0.1	52 ±2	51 ±0	38 ±0	34 ±0	28 ±1

Table 2. TourHAMT falls short of HAMT on IR2R. Simple changes to an episodic model are not enough to leverage tours. PH: previous episodes' history; TH: trainable history encoder; PHI: previous history identifier; IW: inflection weighting. Metrics are $\bar{x} \pm \sigma_{\bar{x}}$ over 3 runs.

		Val-Seen							Val-Unseen						
#	Model	TL	NE ↓	OS ↑	nDTW ↑	SR ↑	SPL ↑	t-nDTW ↑	TL	NE ↓	OS ↑	nDTW ↑	SR ↑	SPL ↑	t-nDTW ↑
1	CMA	7.8 ±0.4	8.8 ±0.6	27 ±3	42 ±3	18 ±3	17 ±3	39 ±1	7.5 ±0.3	8.8 ±0.2	**26** ±1	**44** ±1	**19** ±1	**18** ±1	**38** ±2
2	TourCMA	8.0 ±0.4	**8.2** ±0.9	**30** ±2	**44** ±2	**20** ±3	**19** ±2	**40** ±1	7.8 ±0.1	9.0 ±0.2	**26** ±1	42 ±1	18 ±0	17 ±1	36 ±1
3	PoolCMA	7.2 ±0.5	9.1 ±0.4	24 ±4	41 ±2	17 ±4	16 ±2	37 ±2	7.3 ±0.2	9.0 ±0.3	23 ±1	42 ±1	16 ±1	15 ±0	36 ±2
4	PoolEndCMA	7.6 ±0.8	8.9 ±0.9	27 ±3	42 ±3	18 ±4	17 ±2	38 ±2	6.9 ±0.2	**8.7** ±0.2	25 ±2	**44** ±1	18 ±1	16 ±1	**38** ±2

Table 3. Cross-modal attention (CMA) model performance with *unstructured memory* on IR2R-CE. We compare tour-persistent memory (rows 2-4) against an episodic-memory baseline (row 1). Persisting in the environment over a tour does not improve performance on scenes not seen in training (Val-Unseen), unlike the behavior of *semantic map models* (Tab. 4). Metrics are $\bar{x} \pm \sigma_{\bar{x}}$ over 3 runs.

episodes (row 3), and applying inflection weighting [47] (row 2) collectively regain 4% t-nDTW but still falls short of the HAMT baseline (row 1 vs. 2). As our experiments start from a pretrained HAMT model, we speculate these results are due to the history encoder coping poorly with a distribution shift in its inputs compared to the pretraining tasks and that direct finetuning is insufficient to correct for this.

IR2R-CE. Tab. 3 shows results of the CMA-based models in the continuous IR2R-CE setting. We find that naive extension of unstructured memory reduces performance (row 1 vs. 2), but pooling versions recover tour performance (rows 3-4). Notably, all models augmented with tour memory yield reductions in episodic metrics even when tour metrics are comparable. We draw attention to the TourCMA model performance on Val-Seen (row 2, left) which exhibits stronger performance than the baseline CMA. This result suggests tour-memory allows more overfitting than episodic memory.

We find that more sophisticated memory structures or training methods will be needed to capitalize on IVLN. We explore one such memory structure.

5.2. Semantic Map Memory in IR2R-CE

Agents in IR2R-CE complete long tours (average of 50 R2R episodes per tour) using low-level actions, resulting in some tours requiring over two thousand actions. Storing tour memory as an unstructured vector that is updated per-step (as in the TourCMA/PoolCMA/PoolEndCMA agents) does not effectively remember the environment, and Transformer-based agents may face adverse scaling. We instead consider a *structured* memory in the form of a metric map of semantics and occupancy. We present results varying: the map source {Ground Truth, Inferred via RedNet}; mapping procedure during training {Episodic, Iterative, Known}; and mapping procedure during evaluation {Episodic, Iterative, Known} on IR2R-CE in Tab. 4 for a total of 18 settings.[1]

Map-based memory can leverage prior experience. Across all map sources and map procedures during training, we find agents perform better with iteratively-updated maps persisting across tours (Eval: It.) than with episodic maps reset between episodes (Eval: Ep.). This difference confirms that information gathered from previous episodes in a tour can benefit following novel instructions along novel paths. This effect is strongest when the agent was trained with iteratively-updated maps (Train: It.) with t-nDTW improving by 3-4% for ground truth (row 4 vs. 5) and RedNet-inferred maps (row 13 vs. 14). Models trained on episodic maps find the least benefit from iteratively-updated maps (row 1 vs. 2 and 10 vs. 11), suggesting utilizing semantic map memory beyond the current episode is a learned skill.

Map-structured memories may be better suited to IR2R-CE. Across all settings, we find map-based agents outperform the CMA models from Tab. 3, with best performing iterative agents achieving +9 t-nDTW (24%). However, differences in training procedure may account for some of this gap. Map-based agents underwent DAgger-based fine-tuning but CMA agents did not. In prior work in VLN-CE, DAgger

[1] 'Ground-Truth' agents have access to oracle semantic and occupancy information and evaluations performed with the 'Known' mapping procedure assume a pre-explored environment. These methods do not constitute valid submissions to the IR2R-CE leaderboard, but are provided for analysis.

		Map Construction		Val-Seen							Val-Unseen						
#	Semantics	Train	Eval	TL	NE ↓	OS ↑	nDTW ↑	SR ↑	SPL ↑	t-nDTW ↑	TL	NE ↓	OS ↑	nDTW ↑	SR ↑	SPL ↑	t-nDTW ↑
1	Ground-Truth	Ep.	Ep.	10.6	6.3	51	54	34	32	49	9.8	6.9	42	50	29	26	42
2			It.	10.4	6.3	50	54	34	31	50	9.4	7.0	41	50	29	27	43
3			Kn.	10.1	6.2	50	55	34	32	50	9.5	6.9	40	50	29	26	43
4		It.	Ep.	9.4	6.8	45	54	36	34	51	8.7	7.1	40	52	32	30	44
5			It.	9.5	6.3	52	**58**	41	**39**	**54**	8.5	**6.7**	43	**54**	**36**	**33**	48
6			Kn.	9.4	6.2	51	**58**	**42**	**39**	**54**	8.6	**6.7**	43	**54**	34	32	**49**
7		Kn.	Ep.	10.0	7.3	43	50	32	29	46	9.4	7.7	37	48	29	27	40
8			It.	9.9	6.4	49	55	37	34	51	9.1	6.8	**45**	53	34	31	46
9			Kn.	9.9	**6.1**	**54**	**58**	41	**39**	51	9.2	**6.7**	44	53	34	32	46
10	Inferred	Ep.	Ep.	10.2	6.6	51	54	35	32	50	9.8	7.2	43	50	33	30	43
11			It.	10.1	6.6	49	54	36	33	51	9.3	7.5	39	49	29	27	42
12			Kn.	10.0	6.9	49	53	34	32	50	9.3	7.3	41	50	29	27	43
13		It.	Ep.	9.5	6.9	43	53	34	31	48	8.8	7.3	40	51	31	29	44
14			It.	9.4	6.4	48	56	39	36	52	8.5	**6.8**	**44**	**54**	**35**	**32**	**47**
15			Kn.	9.3	6.4	51	56	38	36	52	8.6	6.9	43	**54**	34	31	46
16		Kn.	Ep.	10.2	6.7	50	53	35	32	50	9.6	7.5	39	49	28	25	41
17			It.	10.0	**6.1**	**57**	**57**	**40**	36	**54**	9.4	7.1	**44**	51	30	27	43
18			Kn.	10.1	6.2	55	**57**	**40**	**37**	53	9.4	7.2	43	51	31	28	44

Table 4. Performance of MAP-CMA agents in IR2R-CE. We consider resetting maps each episode (*Ep.*), constructing maps throughout tours (*It.*), and knowing maps from the start (*Kn.*). We construct maps from ground-truth semantics in rows 1-9 and infer semantics from RedNet [22] in rows 10-18. We use bolding to highlight best scores in ground-truth and inferred semantics separately. Iteratively constructing tour maps leads to better performance than using single-episode maps.

training CMA improved episodic nDTW by 1-5 points [25].

Ground truth and inferred semantics perform similarly. Inferring semantics with RedNet (rows 10-18) leads to a small, consistent drop in performance when compared to using ground-truth semantics (rows 1-9). However, the best performing agent (row 14) sees only a 1 point drop (row 5) in t-nDTW. This limited sensitivity may be due to agents not making full use of the semantic labels or that the scope of labels is not broad enough for highly-performant agents.

Known maps fail to outperform iterative maps. Agents trained and evaluated with known maps fail to outperform those trained and evaluated with iterative maps for both ground-truth (row 9 *vs*. 5) and inferred (row 18 *vs*. 14) semantics, yet known maps outperform training on episodic maps (rows 1, 10). Models trained with iterative maps may benefit from exposure to the divide between explored and unexplored regions. The relatively low performance of known maps points to the open question of how to effectively encode and decode scene perception for downstream reasoning. IR2R-CE can be a fruitful arena for such a study.

6. Conclusions

We define Iterative Vision-and-Language Navigation (IVLN), a paradigm for studying how language-guided agents persisting in a scene, like robots in a home, can utilize past experience to follow instructions. We create the IR2R and IR2R-CE benchmarks to study discrete and continuous navigation across *tours* comprising many single-instruction *episodes*. Initial models for both benchmarks show that extending unstructured latent memory beyond episode scope is insufficient to generalize to tours, but agents that build explicit maps benefit from environment persistence.

Limitations Our benchmarks are limited to English instructions and the indoor spaces are largely lavish, staged homes and offices. Deployed assistive robots performing navigation should respond to more than English, and should be able to navigate cluttered, realistic home environments. Such biases in a benchmark serve the needs of English-speaking, able-bodied folks as a "default," and will hinder such long term goals in spaces of human-robot interaction and accessibility.

Future Work Even with the benefit of tours, MAP-CMA lags far behind human performance on independent episodes, with an episodic SPL of 32 in IR2R-CE Val-Unseen vs. 76 for humans on R2R Test. We anticipate progress may require improved methods for grounding natural language into maps and actions; iteratively constructed maps that are more accurate, flexible and expressive; and improved methods for transfer learning or generating synthetic data.

Acknowledgments and Disclosure of Funding The Oregon State effort is supported in part by the DARPA Machine Common Sense program. The U of Michigan effort is supported in part by the NSF COVE initiative under Grant 1628987 and Google. The views and conclusions contained herein are those of the authors and should not be interpreted as representing the official policies or endorsements, either expressed or implied, of the US Government or any sponsor.

References

[1] Peter Anderson, Angel Chang, Devendra Singh Chaplot, Alexey Dosovitskiy, Saurabh Gupta, Vladlen Koltun, Jana Kosecka, Jitendra Malik, Roozbeh Mottaghi, Manolis Savva, et al. On evaluation of embodied navigation agents. *arXiv*, 2018. 6

[2] Peter Anderson, Ayush Shrivastava, Joanne Truong, Arjun Majumdar, Devi Parikh, Dhruv Batra, and Stefan Lee. Sim-to-real transfer for vision-and-language navigation. In *Conference on Robot Learning (CoRL)*, 2020. 1, 2

[3] Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton van den Hengel. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *Computer Vision and Pattern Recognition (CVPR)*, 2018. 1, 2, 3

[4] Shurjo Banerjee, Jesse Thomason, and Jason J. Corso. The RobotSlang Benchmark: Dialog-guided robot localization and navigation. In *Conference on Robot Learning (CoRL)*, 2020. 2

[5] Valts Blukis, Dipendra Misra, Ross A. Knepper, and Yoav Artzi. Mapping navigation instructions to continuous control actions with position visitation prediction. In *Conference on Robot Learning (CoRL)*, 2018. 2

[6] Vincent Cartillier, Zhile Ren, Neha Jain, Stefan Lee, Irfan Essa, and Dhruv Batra. Semantic mapnet: Building allocentric semantic maps and representations from egocentric views. In *AAAI Conference on Artificial Intelligence*, 2021. 5

[7] 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. In *3D Vision*, 2017. MatterPort3D dataset license available at: `http://kaldir.vc.in.tum.de/matterport/MP_TOS.pdf`. 2, 3, 5

[8] David L. Chen and Raymond J. Mooney. Learning to interpret natural language navigation instructions from observations. In *AAAI Conference on Artificial Intelligence*, 2011. 2

[9] Howard Chen, Alane Suhr, Dipendra Misra, Noah Snavely, and Yoav Artzi. Touchdown: Natural language navigation and spatial reasoning in visual street environments. In *Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[10] Kevin Chen, Junshen K Chen, Jo Chuang, Marynel Vázquez, and Silvio Savarese. Topological planning with transformers for vision-and-language navigation. In *Computer Vision and Pattern Recognition (CVPR)*, 2021. 5

[11] Shizhe Chen, Pierre-Louis Guhur, Cordelia Schmid, and Ivan Laptev. History aware multimodal transformer for vision-and-language navigation. *Neural Information Processing Systems (NeurIPS)*, 2021. 1, 5, 13

[12] Hugh Durrant-Whyte and Tim Bailey. Simultaneous localization and mapping: Part I. *IEEE Robotics & Automation magazine*, 13(2):99–110, 2006. 1

[13] Alberto Elfes. Using occupancy grids for mobile robot perception and navigation. *Computer*, 22(6):46–57, 1989. 5

[14] Daniel Fried, Ronghang Hu, Volkan Cirik, Anna Rohrbach, Jacob Andreas, Louis-Philippe Morency, Taylor Berg-Kirkpatrick, Kate Saenko, Dan Klein, and Trevor Darrell. Speaker-follower models for vision-and-language navigation. *Neural Information and Processing Systems (NeurIPS)*, 2018. 3

[15] Georgios Georgakis, Karl Schmeckpeper, Karan Wanchoo, Soham Dan, Eleni Miltsakaki, Dan Roth, and Kostas Daniilidis. Cross-modal map learning for vision and language navigation. In *Computer Vision and Pattern Recognition (CVPR)*, 2022. 5

[16] Jing Gu, Eliana Stefani, Qi Wu, Jesse Thomason, and Xin Eric Wang. Vision-and-language navigation: A survey of tasks, methods, and future directions. In *Association for Computational Linguistics (ACL)*, 2022. 2

[17] Keld Helsgaun. An effective implementation of the lin–kernighan traveling salesman heuristic. *European journal of operational research*, 126(1):106–130, 2000. 3

[18] Yicong Hong, Zun Wang, Qi Wu, and Stephen Gould. Bridging the gap between learning in discrete and continuous environments for vision-and-language navigation. In *Computer Vision and Pattern Recognition (CVPR)*, 2022. 5

[19] Muhammad Zubair Irshad, Chih-Yao Ma, and Zsolt Kira. Hierarchical cross-modal agent for robotics vision-and-language navigation. In *International Conference on Robotics and Automation (ICRA)*, 2021. 5

[20] Muhammad Zubair Irshad, Niluthpol Chowdhury Mithun, Zachary Seymour, Han-Pang Chiu, Supun Samarasekera, and Rakesh Kumar. SASRA: Semantically-aware spatio-temporal reasoning agent for vision-and-language navigation in continuous environments. In *International Conference on Pattern Recognition (ICPR)*, 2022. 5

[21] Vihan Jain, Gabriel Magalhaes, Alexander Ku, Ashish Vaswani, Eugene Ie, and Jason Baldridge. Stay on the path: Instruction fidelity in vision-and-language navigation. In *Association for Computational Linguistics (ACL)*, 2019. 2, 13

[22] Jindong Jiang, Lunan Zheng, Fei Luo, and Zhijun Zhang. RedNet: Residual encoder-decoder network for indoor rgb-d semantic segmentation. *arXiv*, 2018. 5, 8

[23] Eric Kolve, Roozbeh Mottaghi, Winson Han, Eli VanderBilt, Luca Weihs, Alvaro Herrasti, Daniel Gordon, Yuke Zhu, Abhinav Gupta, and Ali Farhadi. AI2-THOR: An Interactive 3D Environment for Visual AI. *arXiv*, 2017. 3

[24] Jacob Krantz, Aaron Gokaslan, Dhruv Batra, Stefan Lee, and Oleksandr Maksymets. Waypoint models for instruction-guided navigation in continuous environments. In *International Conference on Computer Vision (ICCV)*, 2021. 5

[25] Jacob Krantz, Erik Wijmans, Arjun Majundar, Dhruv Batra, and Stefan Lee. Beyond the nav-graph: Vision and language navigation in continuous environments. In *European Conference on Computer Vision (ECCV)*, 2020. 1, 2, 3, 5, 6, 8, 14

[26] Alexander Ku, Peter Anderson, Roma Patel, Eugene Ie, and Jason Baldridge. Room-Across-Room: Multilingual vision-and-language navigation with dense spatiotemporal grounding. In *Empirical Methods for Natural Language Processing (EMNLP)*, 2020. 2, 4

[27] Chih-Yao Ma, Jiasen Lu, Zuxuan Wu, Ghassan AlRegib, Zsolt Kira, Richard Socher, and Caiming Xiong. Self-monitoring navigation agent via auxiliary progress estimation. In *Interna-*

tional Conference on Learning Representations (ICLR), 2019. 6

[28] Matt MacMahon, Brian Stankiewicz, and Benjamin Kuipers. Walk the talk: Connecting language, knowledge, and action in route instructions. In *AAAI Conference on Artificial Intelligence*, 2006. 2

[29] Gabriel Magalhaes, Vihan Jain, Alexander Ku, Eugene Ie, and Jason Baldridge. General evaluation for instruction conditioned navigation using dynamic time warping. In *NeurIPS Visually Grounded Interaction and Language (ViGIL) Workshop*, 2019. 4, 6

[30] Arjun Majumdar, Ayush Shrivastava, Stefan Lee, Peter Anderson, Devi Parikh, and Dhruv Batra. Improving vision-and-language navigation with image-text pairs from the web. In *European Conference on Computer Vision (ECCV)*, 2020. 3

[31] Viktor Makoviychuk, Lukasz Wawrzyniak, Yunrong Guo, Michelle Lu, Kier Storey, Miles Macklin, David Hoeller, Nikita Rudin, Arthur Allshire, Ankur Handa, and Gavriel State. Isaac gym: High performance gpu based physics simulation for robot learning. In *Neural Information Processing Systems (NeurIPS)*, 2021. 3

[32] Alexey Merzlyakov and Steve Macenski. A comparison of modern general-purpose visual SLAM approaches. In *International Conference on Intelligent Robots and Systems (IROS)*, 2021. 1

[33] Yuankai Qi, Qi Wu, Peter Anderson, Xin Wang, William Yang Wang, Chunhua Shen, and Anton van den Hengel. REVERIE: Remote embodied visual referring expression in real indoor environments. In *Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[34] Santhosh Kumar Ramakrishnan, Aaron Gokaslan, Erik Wijmans, Oleksandr Maksymets, Alexander Clegg, John M Turner, Eric Undersander, Wojciech Galuba, Andrew Westbury, Angel X Chang, et al. Habitat-matterport 3d dataset (hm3d): 1000 large-scale 3d environments for embodied ai. In *Neural Information Processing Systems (NeurIPS)*, 2021. 3

[35] Antoni Rosinol, Marcus Abate, Yun Chang, and Luca Carlone. Kimera: an open-source library for real-time metric-semantic localization and mapping. In *International Conference on Robotics and Automation (ICRA)*, 2020. 2

[36] Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Artificial Intelligence and Statistics (AISTATS)*, 2011. 6

[37] Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, et al. Habitat: A platform for embodied ai research. In *International Conference on Computer Vision (ICCV)*, 2019. 3

[38] Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox. ALFRED: A benchmark for interpreting grounded instructions for everyday tasks. In *Computer Vision and Pattern Recognition (CVPR)*, 2020. 1, 2

[39] Takafumi Taketomi, Hideaki Uchiyama, and Sei Ikeda. Visual SLAM algorithms: A survey from 2010 to 2016. *IPSJ Transactions on Computer Vision and Applications*, 9(1):1–11, 2017. 1

[40] Hao Tan, Licheng Yu, and Mohit Bansal. Learning to navigate unseen environments: Back translation with environmental dropout. *North American Chapter of the Association for Computational Linguistics (NAACL)*, 2019. 3, 6

[41] Stefanie Tellex, Nakul Gopalan, Hadas Kress-Gazit, and Cynthia Matuszek. Robots that use language. *The Annual Review of Control, Robotics, and Autonomous Systems*, 15, 2020. 2

[42] Jesse Thomason, Michael Murray, Maya Cakmak, and Luke Zettlemoyer. Vision-and-dialog navigation. In *Conference on Robot Learning (CoRL)*, 2019. 2

[43] Sebastian Thrun. Simultaneous localization and mapping. In *Robotics and cognitive approaches to spatial mapping*, pages 13–41. Springer, 2007. 1

[44] Xin Wang, Qiuyuan Huang, Asli Celikyilmaz, Jianfeng Gao, Dinghan Shen, Yuan-Fang Wang, William Yang Wang, and Lei Zhang. Reinforced cross-modal matching and self-supervised imitation learning for vision-language navigation. In *Computer Vision and Pattern Recognition (CVPR)*, 2019. 3

[45] Saim Wani, Shivansh Patel, Unnat Jain, Angel Chang, and Manolis Savva. Multion: Benchmarking semantic map memory using multi-object navigation. *Neural Information Processing Systems (NeurIPS)*, 2020. 3

[46] Luca Weihs, Matt Deitke, Aniruddha Kembhavi, and Roozbeh Mottaghi. Visual room rearrangement. In *Computer Vision and Pattern Recognition (CVPR)*, 2021. 3

[47] Erik Wijmans, Samyak Datta, Oleksandr Maksymets, Abhishek Das, Georgia Gkioxari, Stefan Lee, Irfan Essa, Devi Parikh, and Dhruv Batra. Embodied question answering in photorealistic environments with point cloud perception. In *Computer Vision and Pattern Recognition (CVPR)*, 2019. 5, 7

[48] 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. 3

[49] Jingwei Zhang, Lei Tai, Ming Liu, Joschka Boedecker, and Wolfram Burgard. Neural SLAM: Learning to explore with external memory. *arXiv*, 2017. 1

[50] Fengda Zhu, Yi Zhu, Xiaojun Chang, and Xiaodan Liang. Vision-language navigation with self-supervised auxiliary reasoning tasks. In *Computer Vision and Pattern Recognition (CVPR)*, 2020. 3

[51] Wang Zhu, Hexiang Hu, Jiacheng Chen, Zhiwei Deng, Vihan Jain, Eugene Ie, and Fei Sha. BabyWalk: Going farther in vision-and-language navigation by taking baby steps. In *Association for Computational Linguistics (ACL)*, 2020. 2, 13