Behavioral Analysis of Vision-and-Language Navigation Agents

Zijiao Yang
Oregon State University
yangziji@oregonstate.edu

Arjun Majumdar
Georgia Institute of Technology
arjun.majumdar@gatech.edu

Stefan Lee
Oregon State University
leestef@oregonstate.edu

Abstract

To be successful, Vision-and-Language Navigation (VLN) agents must be able to ground instructions to actions based on their surroundings. In this work, we develop a methodology to study agent behavior on a skill-specific basis – examining how well existing agents ground instructions about stopping, turning, and moving towards specified objects or rooms. Our approach is based on generating skill-specific interventions and measuring changes in agent predictions. We present a detailed case study analyzing the behavior of a recent agent and then compare multiple agents in terms of skill-specific competency scores. This analysis suggests that biases from training have lasting effects on agent behavior and that existing models are able to ground simple referring expressions. Our comparisons between models show that skill-specific scores correlate with improvements in overall VLN task performance.

1. Introduction

Following navigation instructions requires coordinating observations and actions in accordance with the natural language. Stopping when told to stop. Turning when told to turn. And appropriately grounding referring expressions when an action is conditioned on some aspect of the environment. All three of these example are required when following the instruction *"Turn left then go down the hallway until you see a desk. Walk towards the desk and then stop."* In this work, we examine how well current instruction-following agents can execute different types of these sub-behaviors which we will refer to as *skills*.

We situate our study in the popular Vision-and-Language Navigation (VLN) paradigm [2]. In a VLN episode, an agent is spawned in a never-before-seen environment and must navigate to a goal location specified by a natural language navigation instruction. An agent's instruction-following capabilities are typically measured at the episode level – examining whether an agent reaches near the goal (success rate [2]), how efficiently it does so (SPL [1]), or how well its trajectory matches the ground truth path which the human-generated instruction was based on (nDTW [15]). These metrics are useful for comparing agents in aggregate, but take a perspective that has little to say about an agent's fine grained competencies or what sub-instructions it is able to ground appropriately.

In this work, we design an experimental paradigm based on controlled interventions to analyze fine-grained agent behaviors. We focus our study on an agent's ability to execute unconditional instructions like stopping or turning, as well as, conditional instructions that require more visual grounding like moving towards specified objects and rooms. Our approach leverages annotations from RxR [13] to produce truncated trajectory-instruction pairs that can then be augmented with an additional skill-specific sub-instruction. We carefully filter these trajectories and generate template-based sub-instructions to build non-trivial intervention episodes that evaluate an agent's ability to ground skill-specific language to the appropriate actions.

To demonstrate the value of this approach, we present a case study analyzing the behavior of a contemporary VLN model [6]. While we find evidence that the model can reliably ground some skill-specific language, our analysis also reveals that its errors are not random. But rather, they reflect a systematic bias towards forward actions learned during training. For object- or room-seeking skills, we find only modest relationships between instructions and agent actions. Finally, we derive aggregate skill-specific scores and compare across VLN models with different overall task performance. We find that higher skill-specific scores correlate with higher task performance; however, not all skills share the same scale of improvement between weaker and stronger VLN models – suggesting that improvements in VLN may be fueled by some skills more than others.

Contributions. To summarize this work, we:

- Develop an intervention-based behavioral analysis paradigm for evaluating the behavior of VLN agents,[1]
- Provide a case study on a contemporary VLN agent [6], evaluating fine-grained competencies and biases, and
- Examine the relationships between skill-specific metrics

[1] https://github.com/Yoark/vln-behave

and overall VLN task performance.

2. Related Work

Vision-and-Language Navigation (VLN). Since its introduction in [2], many variants of the Vision-and-Language Navigation (VLN) task have been proposed including those in continuous simulators [12]. We refer the reader to [7] for a comprehensive survey. In this work, we examine agents in the Room-Across-Room (RxR) dataset [13] which extends the original VLN task to a multilingual setting with longer, more complex trajectories and pose traces which provide temporal alignment between instruction words and visual observations. There has also been significant modelling work to develop instruction-following agents [6, 8, 10, 11, 14, 16, 20, 21] and we examine three recent models in our analysis [6, 20, 21].

The RxR task is situated in the MATTERPORT3D [4] environments which provide an interface for agents to move through the environment along a graph of panoramic viewpoints taken in real environments. Matterport3D also provides region annotations for room type which we utilize in our experiments. We also leverage annotations from the REVERIE [17] dataset which extends VLN settings with an additional goal of identifying an object described by a referring expression. Specifically, using the annotations from REVERIEv1 to identify visible objects at each viewpoint.

Evaluating VLN Agents. In standard settings, VLN agents are evaluated by metrics that focus on either the agent reaching the goal efficiently (Success weighted by inverse Path Length [1]) or by their trajectory's alignment with the ground truth path (Normalized Dynamic Time Warping [2]). These metrics focus on the agent's performance in aggregate and do not examine agent performance on the level of sub-instruction or skills.

Some works have examined VLN agent behavior more closely by masking or replacing portions of the instructions [9, 23] and observing the resulting change to overall task metrics like those described above. Zhu *et al.* [23] find that VLN agents still achieve relatively high success rates even when all references to visual objects are masked from instructions. These findings cast doubt on the vision-language alignment ability of these agents. [9] also perform masking experiments but come to different conclusions, with some models relying more heavily on nouns than directional words. In both works, agent performance is measured on an episodic level that relies on a sequence of agent decisions; however, single errors in a trajectory may compound and result in misestimating the impact of masked terms. In contrast to these works, we present a skill-based analysis of VLN agents by constructing specific intervention episodes wherein the appropriate next action is known.

Behavioral Analysis of AI Models. Recent work in natural language processes has applied behavioral analysis to examine specific skill capabilities. Like us, Riberio *et al.* [19] develop an intervention paradigm wherein dataset examples are modified in ways such that the desired change in model behavior is knowable. These examples and their associated skills are collected into a "checklist" that can be used to evaluate models. Likewise, our work can be construed as generating a checklist of skills for VLN. Yang *et al.* [22] follow a similar paradigm and develop a method to automatically generate test cases using a large language models [3].

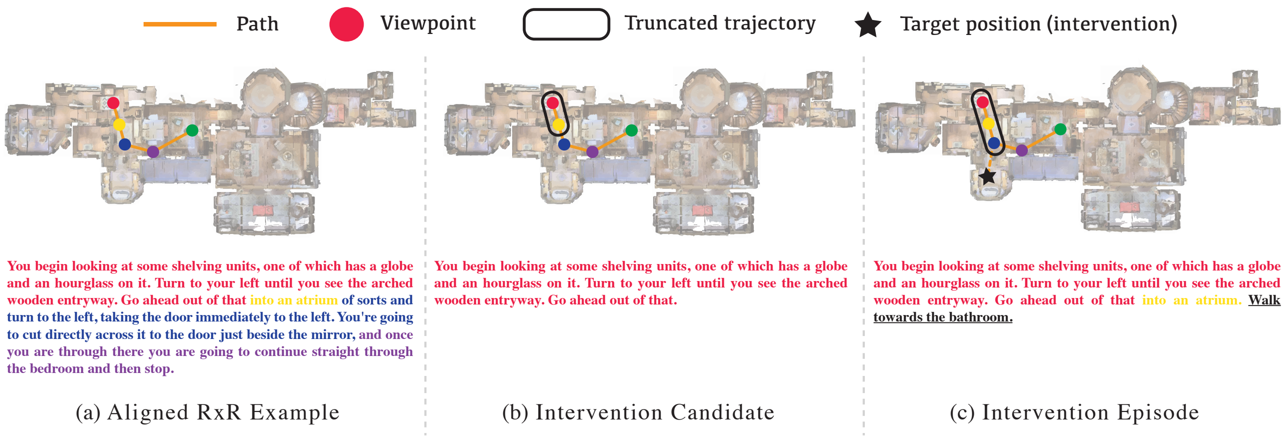
3. Analyzing the Behavioral of VLN Agents

In this work, we examine fine-grained agent competency through the lens of behavioral analysis – studying how agent decisions change in the presence and absence of skill-specific language instructions. For example, consider object-conditional instructions like "walk towards the couch". To demonstrate sensitivity to this instruction, an agent must be more likely to face a couch when presented with this instruction than it would otherwise across a wide range of settings. Note that this analysis examines what agents *do* rather than how they arrive at those decisions and is thus applicable to any VLN agent. To enable this analysis, we develop an intervention strategy that produces trajectories-instruction pairs with and without skill-specific language included. We deploy existing VLN agents on these to examine their decisions and measure their sensitivity to the intervention. The remainder of this section describes our overall methodology – how intervention episodes are generated and how agents are evaluated – and additional skill-specific experimental details are provided in the following section alongside example agent results.

3.1. Building Intervention Episodes

We consider an intervention sample to be a tuple consisting of a trajectory τ, an instruction I that guides an agent to the end of that trajectory, and an intervention instruction I_{int} that describes some desired skill-specific behavior to be taken from that point. For intervened episodes, an agent will be given the augmented instruction $I + I_{int}$, guided through the trajectory τ and then its decision will be compared to the expected behavior described in I_{int}. We choose this partial-path construction so we can vary pre-intervention path length while keeping trajectory-instruction alignment similar to standard VLN training settings. Fig. 1 (c) shows one such triplet with a 3-step trajectory, an instruction describing it, and an underlined intervention prompting the agent to move "towards the bathroom". We design different interventions to study fine-grained skills in Sec. 4.

Candidate Instruction-Trajectories Pairs. To create trajectory-instruction pairs for intervention samples, we leverage the detailed annotations in the RxR dataset [13]. RxR provides trajectory-instruction pairs along with 'pose



You begin looking at some shelving units, one of which has a globe and an hourglass on it. Turn to your left until you see the arched wooden entryway. Go ahead out of that into an atrium of sorts and turn to the left, taking the door immediately to the left. You're going to cut directly across it to the door just beside the mirror, and once you are through there you are going to continue straight through the bedroom and then stop.

(a) Aligned RxR Example

You begin looking at some shelving units, one of which has a globe and an hourglass on it. Turn to your left until you see the arched wooden entryway. Go ahead out of that.

(b) Intervention Candidate

You begin looking at some shelving units, one of which has a globe and an hourglass on it. Turn to your left until you see the arched wooden entryway. Go ahead out of that into an atrium. Walk towards the bathroom.

(c) Intervention Episode

Figure 1. To build skill-specific interventions, we truncate existing RxR episodes based on RxR trajectory-instruction alignments (a→b). We can then extend the instruction with skill-specific language and identify the next step described by this new instruction (c).

traces' that provide temporal alignment between instruction words and the annotator's pose in the trajectory. This allows us to associate instruction text segments with nodes along the trajectory – writing the trajectory as a sequence of node visitations $n_1, n_2, ..., n_T$ with associated sub-instructions $i_1, i_2, ..., i_T$ uttered at each. Fig. 1 (a) demonstrates this by color-coding. For each RxR example, we form candidate τ, I pairs by truncation – taking the trajectory up to node n_j ($\tau = \{n_{\leq j}\}$) and instruction text prior to arriving at n ($I_p = \{i_{<j}\}$). This provides a trajectory and the instructions delivering an agent up to the final node. An example truncation is shown in Fig. 1 (b). For a length T trajectory, this generates $T-2$ candidates; we exclude full-length trajectories because instructions given at penultimate nodes often include explicit directions to stop after moving. Keeping these would add confusion about the appropriate behavior to take after intervention text is appended. We consider all English trajectory-instruction pairs from the `val-unseen-guide` split of RxR [13] for this process.

Filtering and Intervention Instructions. Not all candidate instruction-trajectory pairs are useful for all interventions – for example, it may be impossible to "turn left and go forward" for a trajectory that ends on a node without a leftward neighbor. Likewise, an example where a leftward turn is the *only* option could not demonstrate differential agent behavior with and without such an intervention. As described in Sec. 4 below, we filter candidate trajectories for each experiment to ensure agents have both intervention relevant and irrelevant action options in every episode.

To develop intervention instructions, we manually examined the RxR dataset to identify common skills. In this work, we examine four common skills related to stopping, responding to directional language, and moving relative to objects or room references. For each, we develop a set of language templates to build intervention instructions I_{int}. These templates are based on common phrases from RxR instructions and may be conditioned on objects or rooms present in the trajectory. For these object or room references, we leverage the REVERIEv1 [17] and MATTERPORT3D [5] datasets respectively. A full list of these templates is provided in the supplementary materials.

3.2. Evaluating Agent Sensitivity

To evaluate agents on a given intervention episode, we consider both the truncated (τ, I) and intervened $(\tau, I{+}I_{int})$ trajectory-instruction pairs for that episode. For each, we provide the agent with the instruction and then force it to follow the trajectory τ until reaching its final node. Matching common teacher-forcing training paradigms, the agent is provided with the observations along the trajectory but its action predictions at each node are ignored in favor of the true next step. At the final node, we then record the agent's predicted action distribution $P(a|\tau, I)$ for each setting where I is either the intervened or truncated instruction. Done over all intervention episodes, this yields a set of distribution pairs $P(a|\tau_j, I_j)$ and $P(a|\tau_j, I_j + I_{int_j})$ which we can use to examine how an agent's beliefs shift under the intervention while everything else about the agent's experience is held constant. We examine the distributions rather than argmax actions to improve sensitivity to shifts in agent belief.

For modern VLN agents, these predicted action distributions correspond to distributions over neighboring viewpoints in the navigation graph as well as a stop action. As such, we can map the desired behavior of most of our interventions to a single or set of neighboring nodes which should have increased or decreased probability under the intervention. For example, telling a skillful agent to "turn left" rather than "go forward" should shift the predicted probabilities towards neighboring nodes on its left.

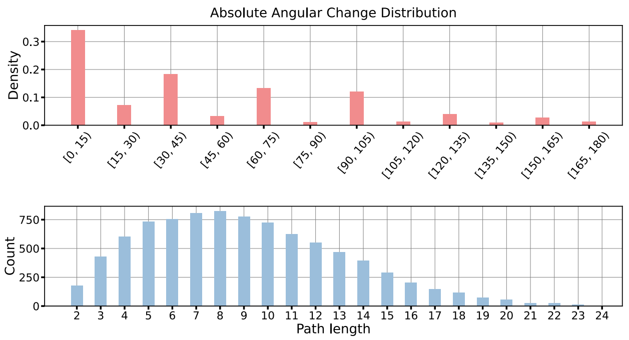


Figure 2. RxR Trajectory Statistics. (Top) Absolute heading change between sequential nodes in training trajectories. There is a strong "forward" biases with 34% of the mass in the 0 to 15° bin. (Bottom) The path length distribution of training dataset, showing agents have been exposed to varied length trajectories.

For some experiments, we introduce additional instruction modification setting to examine the effects of dataset biases in addition to the truncation and interventions constructions. Details are provided in each experimental section.

RxR *vs*. Intervention Instructions. By construction, our generated intervention instructions will tend to contain more short trajectories and instructions than RxR; however, RxR does exhibit a significant variance in path length (see Fig. 2). On the language side, we use templates that match commonly used phrases in RxR to minimize differences.

Sample Correlation. We note that multiple intervention episodes may be drawn from a single trajectory and many will come from each environment. As such, our measurements may exhibit correlations due to this sampling strategy. To account for these effects when discussing the significance of our results, we apply hierarchical bootstrapping [18] when providing confidence intervals and linear mixed effect models when estimating intervention effects. Additional details are provided in the supplementary.

4. Case Studies on a Recent VLN Agent

We instantiate our intervention paradigm for four common skills related to stopping and turning and show the resulting analysis for the recent HAMT [6] model. HAMT is near state-of-the-art agent for VLN that is based on a multimodal transformer model that jointly encodes trajectory history and instruction text in a hierarchical fashion. HAMT is trained in multiple stages including auxiliary losses and joint imitation and reinforcement learning finetuning. For the experiments below, we use pretrained models and inference code provided by the authors. Like other VLN models, HAMT predicts a distribution over neighboring nodes in the navigation graph and the stop action.

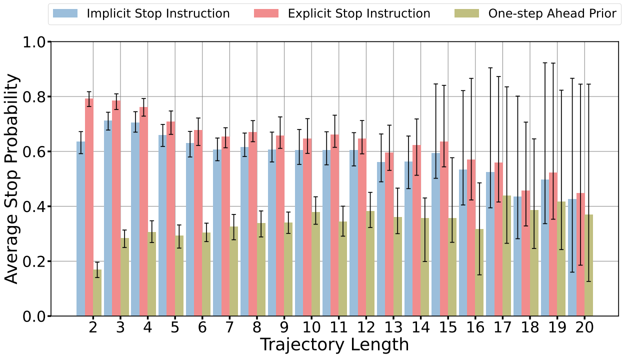


Figure 3. Average Stop Probability vs Trajectory Length for "implicit stop instruction", "explicit stop instruction" and "one-step ahead prior". Agents respond strongly to both stop interventions – stopping with high probability across all trajectory lengths. Explicit stop instruction produce a stronger effect than implicit.

4.1. Stop Instructions

To be successful at the VLN task, an agent must declare the `stop` action within a fixed radius of the goal location described by an instruction. As such, grounding explicit (*"Go to the bedroom and stop."*) and implicit (*"... then go into the bedroom.* `EOS`*"*) stopping instructions to the `stop` action is an important skill. In this experiment, we analyze stop behavior for explicit or implicit stop instructions. To assess the effect of path length distributions in RxR, we examine stop behavior across a range of path lengths.

Intervention Details. All intervention candidates are viable for this experiment as the stop action is always an option and alternative actions (neighboring nodes) always exist. For intervention instructions, we append a short stop instruction such as *"This is your destination."* We note however that the stop experiment offers a complication – both the truncated and intervened instructions imply stop actions. The difference being whether this instruction is implicit (truncation) or explicit (intervention). To provide additional comparison with non-stop instructions, we also consider a *one-step ahead* instruction that includes the instruction segment from the terminal node as well (*i.e*. the agent is instructed to make the next step in the trajectory). In total, we produce 8221 intervention episodes. For each episode, we measure the probability of the `stop` action from agent at the final node of the trajectory.

Results. Fig. 3 shows average stop probabilities across different trajectory lengths for the truncated implicit stop, intervened explicit stop, and one-step ahead instruction settings. Error bars are 95% hierarchical bootstrap confidence intervals. We find average stop probability to remain fairly constant for implicit and explicit stop instructions.[2] This suggests agents consistently ground the stop instruction re-

[2] Note that longer trajectories have fewer episodes and larger variation.

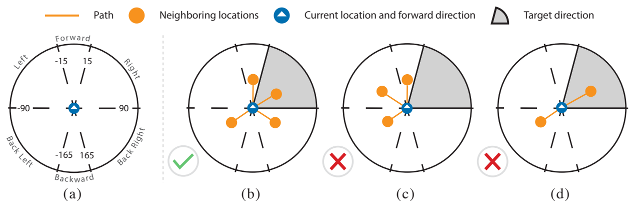


Figure 4. For our directional intervention, we define six directions on the polar axis (a) and establish filters to avoid ungroundable (c) or trivial episodes (d) – requiring that at a neighboring node is in the target direction and at least two other neighbors are not.

gardless of trajectory length despite biased RxR training trajectory length (see Fig. 2). The plot also suggests stop probability is higher for explicit than implicit stop which are both naturally higher than the one-step ahead setting.

To evaluate statistical significance of the effect, we consider a linear mixed effect model (`lmer`) where the observed stop probability is assumed to be an effect of the intervention plus random effects from the environment and source trajectory. We find agents have a higher probability of stopping when given explicit rather than implicit stop instructions (0.72 *vs*. 0.65, effect: 0.07 `anova`: $p{\approx}0$) and that agents respond to both implicit and explicit stop instructions by increasing stop probability compared to the one-step ahead baseline (effect: 0.36, $p{\approx}0$).

Summary. We find the agent responds strongly to implicit and explicit stops across all trajectory lengths and that explicit stop instructions have a stronger effect.

4.2. Unconditional Directional Instructions

Another foundational skill for following navigation instructions is responding appropriately to directional language. In this experiment, we examine *unconditional directional instructions* which specify directions like *"turn left and go forward"* without referencing entities in the environment. This language is frequently used to orient agents in the absence of clear landmarks. We consider language describing forward/backward motions and turns. Specifically, we explore six direction categories – forward, backward, left, right, back left, and back right. For each, we define an angular region relative to the agent's heading (canonically 0 degrees) as shown in the top-left of Fig. 4. During our experiment, we can examine the amount of probability placed on neighboring nodes within these regions.

Intervention Details. For each direction, we filter intervention candidates to ensure that a) the final node has at least one neighbor in the corresponding direction region and b) that there exist at least two other neighbors outside the direction region. These criteria are demonstrated in Fig. 4. This ensures that there exists a next step that matches the intervention instruction and that there are multiple alternative actions besides `stop`. Recall that the agent's action space is to move to a neighbor or to `stop`, such that turning in place is not possible. So for intervention instructions, we build templates that instruct the agent to face a direction and then go forward (*e.g*. *"Turn right and walk forward"*). Early experiments with only direction commands resulted in weaker directional effects. We generate between 3091 and 6745 intervention episodes depending on the direction.

Results. For each episode, we record the agent's predicted distribution over neighboring nodes. These can be mapped to beliefs over relative angles by associating the probability of visiting neighbor k with the relative angle θ_k between the agent's heading and neighbor k. Fig. 5 shows the distribution of these probabilities over all episodes for each intervention as histograms on polar axes. For convenience, we denote the target direction region with a green arc.

Across all directions, we find the agent either stops (roughly 65% of the time) or moves roughly forward in the no intervention setting. For left and right, we see a minor bias towards the corresponding direction despite the agent not receiving any left/right instruction. This reflects a minor structural bias caused by the filtering process. All left (right) episodes include a neighbor to the left (right) and an agent with a bias towards moving roughly forward may select them at a higher rate than other nodes.

For the intervention setting, we see a strong response to directional language. In all cases, the agent stops significantly less (roughly 19% of the time on average) and we observe a shift in distribution towards the corresponding direction. One outlier is the backwards setting which smears the probability significantly to both the back left and back right. In all settings, some mass still remains in the forward direction – reflecting the agent's bias towards moving forward learned from the training data as reflected in Fig. 2.

We accumulate the probability mass into directional bins and evaluate the effect of intervention on the accumulated probability. We again use a linear mixed effect model of the same from as the previous experiment to account for potential correlations in scenes and trajectories. We find the agent exhibits a significantly higher accumulated probability for the corresponding direction with directional instruction than without – estimating intervention effects as increased accumulated probability for left (0.38, $p \approx 0$), right (0.44, $p \approx 0$), forward (0.16, $p \approx 0$), backward (0.16, $p \approx 0$), back left (0.43, $p \approx 0$), and back right (0.48, $p \approx 0$).

Summary. We find the HAMT [6] agent strongly respond to directional language but some dataset biases from training are still evident in a bias towards forward actions.

4.3. Object-seeking Instructions

Beyond directional language, instructions also often use references to nearby objects as convenient landmarks, *e.g.* *"Walk towards the fireplace"*. Unlike the language studied

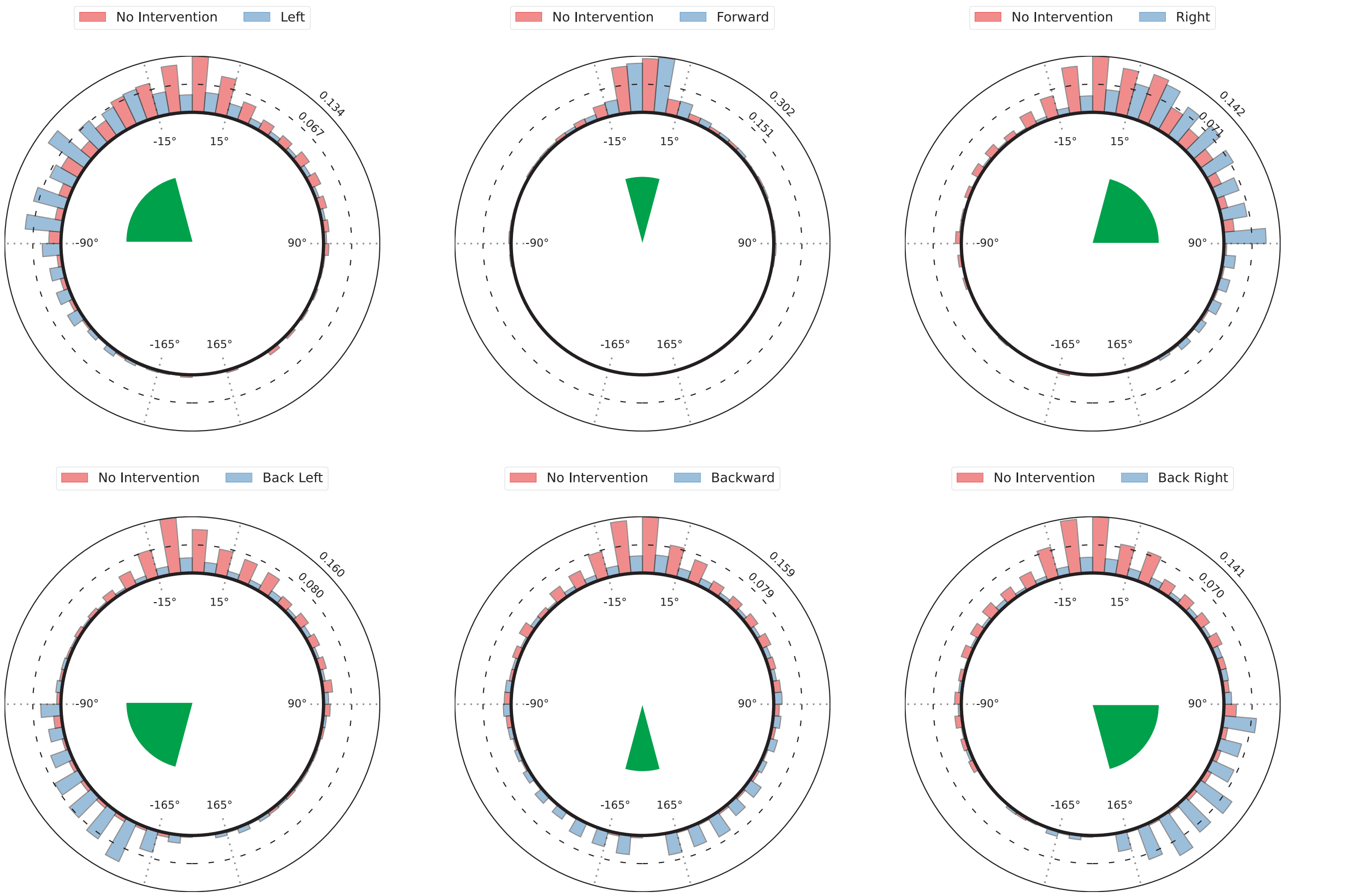


Figure 5. We plot the agent's next step direction probability distribution onto polar axis for easier visualization. We provide results for 6 directions defined in 4 and contrast between "No Intervention" (red) with "Direction" (blue). The number on outer circle and middle dotted circle are max and $\frac{max}{2}$ respectively. We found the HAMT agent is responsive to all six directional instructions: the probability mass of directional interventions shifts toward the area indicated by directional instructions across all directions comparing to "No Intervention"

in the previous sections, object-seeking instructions require grounding the instruction to the visual scene. We examine simple "walk towards X" style object-seeking instructions.

Intervention Details. We leverage the object annotations from the REVERIE [17] dataset to build intervention episodes. Specifically, we retain an intervention candidate if a REVERIE object is visible from its terminal node, the object is no more than 3m away, there exists a neighboring node with a heading that is within 15 degrees of the object's heading, and there at least two neighbors. That is to say, trajectories that end near a visible object and a non-trivial navigation action can reasonably move towards it. We exclude common structural objects like doors, windows, shelving, railings, etc. as it is often unclear which of multiple occurrences an agent should move towards. For instructions intervention instructions, we append the template *"Walk towards the [object]"* where `object` is the object name from REVERIE. In total, we generate 839 intervention episodes targeting the following objects in decreasing order of occurrence: chair, table, picture, cushion, curtain, plant, cabinet, gym equipment, stool, chest of drawers, bed, towel, bathtub, tv monitor, and seating.

Results. For each episode, we record the agent's predicted distribution over neighboring nodes at the terminal node. We map these to a distribution over absolute angular errors relative to the object. For each neighbor k, we compute the difference in heading angle between node k and the object. We can then associate the probability of visiting neighbor k with an angular distance to the target object. These probabilities are accumulated and normalized to produce Fig. 6 which presents distributions over angular distance for the intervention and no-intervention settings.

This intervention shows a weak effect – with the agent reducing angular error in the presence of the intervention instruction somewhat (blue *vs*. red bars in Fig. 6). We again leverage a linear mixed effect model to evaluate the effect of

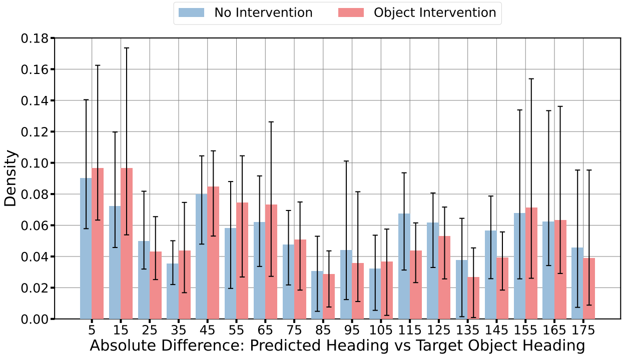


Figure 6. The distribution of absolute difference between model prediction and target object direction for intervention and no intervention settings.

intervention on the accumulated probability within 15 degree of absolute angular difference. We find weak fixed effect of 0.069 (`anova`, $p \approx 0$) for intervention *vs*. non-intervention. However, both settings exhibit a wide range of angular errors suggesting that target objects are not being grounded accurately – recall that all trajectories have neighboring nodes that would incur no more than 15 degrees of error. To explore this further, we also examine a baseline `Forward Bias` agent that places probability on neighbors inversely proportional to their relative heading. We find this baseline exhibits a similarly shaped error distribution – suggesting the agent may be taking forward actions when uncertain about the target object. As in our other experiments, the no intervention setting is more likely to stop than the intervention (65% *vs*. 37%).

Summary. We find evidence for only a weak tendency to move towards referenced objects for this agent.

4.4. Room-seeking Instructions

Agents may also be asked to navigate to specific rooms, often without specific directional language describing how to access them – *e.g*. *"go to the kitchen."* In this experiment, we examine these *room-seeking* instructions both in the setting where the room is likely visible and when an agent may need to search for it nearby. We note that this latter task is beyond the scope of standard VLN instructions and examine it as a test of generalization. Below, we denote the case where a room is likely visible as a 1-hop setting and extensions beyond this as k-hop.

Intervention Details. Using MATTERPORT3D [5] room region annotations, we associate each node in all trajectories with a room label. For 1-hop settings, we retain intervention candidates where the terminal node has at least one neighboring node with a different room type. For k-hop, we extend this to consider neighbors that are k steps away from the current node. For intervention instructions, we append a template *"Walk towards the [room]."* where `room` is replaced with the corresponding room name from MATTERPORT3D. As k-hop episodes involve agents making multiple decisions, we also append *"This is your destination."* afterwards to encourage agents to stop once reaching the room. We generate 8614 intervention episodes for 1-hop setting, and 17204 to 27454 episodes for n-hop settings.

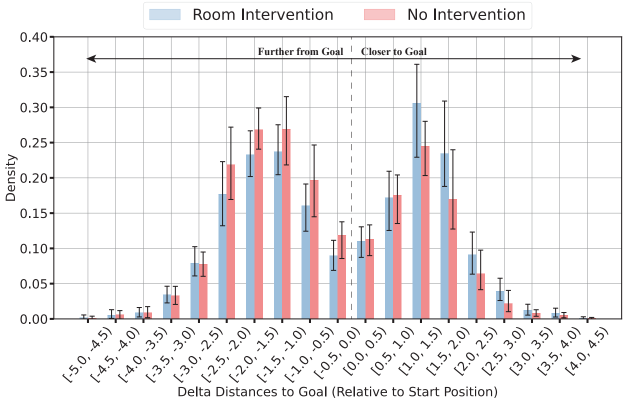


Figure 7. Distribution of delta distance to target room type. The delta distance difference of distance to target room (relative to start position) with or without intervention. Positive delta distance means the agent move closer to room of target type with intervention than otherwise. The distribution shift towards right with intervention than otherwise, indicates the agent is responsive to room-seeking instruction. (-0.15 *vs*.-0.41, $p \approx 0$)

1-Hop Results. For each episode, we record the agent's predicted distribution over neighboring nodes at end of the trajectory. To measure agent progress towards nodes with target room type, we map these to a distribution over change in geodesic distances to the *nearest* node with the target room type. This is done analogously to previous sections such that the probability of visiting node j from the final trajectory node T is associated with the delta geodesic distance $\Delta d = d_{geo}(n_T, n_{near}) - d_{geo}(n_j, n_{near})$ to the nearest node with the target room type n_{near}. The probabilities are accumulated and shown in Fig. 7 – values greater than zero represent the agent moving *closer* to nodes with the target room type. We observe a right-shift in the density suggesting the agent responds somewhat to the intervention. Again using a linear mixed effect model, we estimate the effect of intervention on the delta geodesic distance as 0.26 (`anova`, $p \approx 0$) for intervention vs no intervention. However, the agent does not reliably place strong beliefs on neighbors with the target room type – negative median delta distance and significant mass to the left of zero.

k-Hop Results. For each k-hop episode, we force the agent to follow the trajectory until it ends at node n_T and then execute the agent by taking argmax actions until `stop` is called and then record the final position n_{end}. We report the distance to the nearest node with target room type from here, $d_{geo}(n_{end}, n_{near})$. We shows a ridgeline plot in Fig. 8

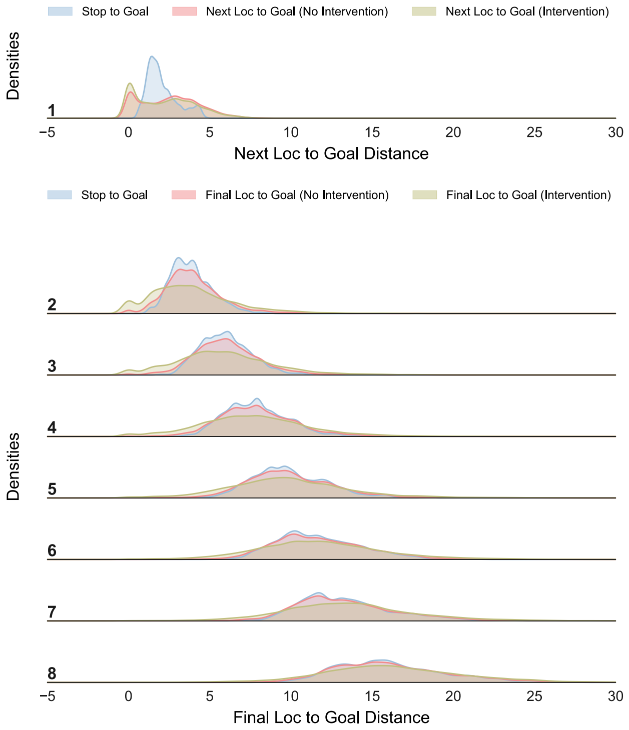


Figure 8. Distribution of geodesic distance to nearest target room location for k-hop room-seeking experiments. `Stop to Goal` is a baseline agent that always takes the stop action.

comparing these distributions for 1- to 8-hops.

We find error increases with target room distance. We again leverage a linear mixed effect model to evaluate the effect of intervention on $d_{geo}(n_{end}, n_{near})$. We find weak fixed effect of ≤ -0.1 for intervention *vs*. non-intervention for 3,4, and 5 hops with 95% confidence). Note that sample size varies with number of hops. Overall, this suggests agents have limited ability to search for rooms based on common sense exploration – perhaps unsurprising given that RxR instructions typically provide step-by-step guidance.

Summary. The HAMT [6] agent is weakly sensitive to room type reference instructions when the room is visible (within one hop) but lacks the ability (room type instruction have weak or none effects) to perform common sense exploration to find further away rooms (k-hop). Overall sensitivity is low, suggesting the agent may not rely on room-specifying portions of instructions when navigating.

5. Comparing Sensitivity Across Agents

For each skill-specific intervention, we can identify a set of neighbor actions (neighboring nodes or `stop`) that correspond to a correct grounding of the intervention instruction – matching the filtering criteria used in their constructions. For stop instructions, this is the stop action. For turn instructions, neighbors within the corresponding direction angle range are valid. For object-seeking, neighbors within 15° of the object heading. For room-seeking, neighbors with target room type.

Method	Stop	Turn	Object	Room	Avg.
EnvDrop [21]	62.65	27.14	11.06	23.64	31.12
EnvDrop (ViL CLIP) [20]	66.76	27.45	12.83	26.82	33.47
HAMT [6]	71.65	43.74	12.00	26.63	38.50

Table 1. We report scores for each skill type for VLN models with varying RxR task performance (EnvDrop < EnvDrop (ViL CLIP) < HAMT). We find individual skill performance tends to improve with overall task performance, but not equally over all skills. Object- and room-seeking instructions require further study.

To compare across VLN models, we examine the average probability mass they place on these correct actions. Denoting the set of correct actions as $\mathcal{N}_e$ for an intervention episode e, we can write a scoring function over a set of intervention episodes $\mathcal{E}_s$ for skill s for an agent f as

$$Score(f, s) = \frac{1}{|\mathcal{E}_s|} \sum_{e \in \mathcal{E}_s} \sum_{a_n \in \mathcal{N}_e} P_f(a_n | \tau_e, I_e + I_{int_e}) \quad (1)$$

where $P_f(a_n | \cdots)$ is agent f's predicted probability for action a_n at the end of the intervention trajectory. Higher scores reflect greater certainty in selecting a grounded action on average across all episodes.

Tab. 1 shows these scores for three VLN agents of varying RxR task performance (EnvDrop < EnvDrop (ViL CLIP) < HAMT). We find that improved model performance on the overall RxR task tends to also leads to improvements on skill-specific scores. However, improvements are not uniform across the skills and agents are more proficient at stopping and turning instructions than those referencing objects or rooms.

6. Discussion

In this work, we introduced an analysis paradigm for studying fine-grained skill competency in VLN agents. To show its value, we presented a case study on a recent VLN model. This provided insights into agent behavior including significant differences in performance on unconditional (stop and turn) and conditional (object- and room-seeking) skills. Finally, we presented a comparison between models in terms of skill-specific scores.

Skill-specific analysis like we present here can provide insight into the types of things we can reasonable expect from our models and those which will require further study. Building explicit test cases for desired skills can serve as "unit tests" on our path to more complex instruction following systems. This work is a step in that direction.

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 preprint arXiv:1807.06757*, 2018. 1, 2

[2] 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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3674–3683, 2018. 1, 2

[3] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020. 2

[4] 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. *International Conference on 3D Vision (3DV)*, 2017. 2

[5] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments. *arXiv preprint arXiv:1709.06158*, 2017. 3, 7

[6] Shizhe Chen, Pierre-Louis Guhur, Cordelia Schmid, and Ivan Laptev. History aware multimodal transformer for vision-and-language navigation. *Advances in Neural Information Processing Systems*, 34:5834–5847, 2021. 1, 2, 4, 5, 8

[7] Jing Gu, Eliana Stefani, Qi Wu, Jesse Thomason, and Xin Wang. Vision-and-language navigation: A survey of tasks, methods, and future directions. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7606–7623, Dublin, Ireland, May 2022. Association for Computational Linguistics. 2

[8] Pierre-Louis Guhur, Makarand Tapaswi, Shizhe Chen, Ivan Laptev, and Cordelia Schmid. Airbert: In-domain pretraining for vision-and-language navigation. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1614–1623, 2021. 2

[9] Meera Hahn and James M Rehg. Which way is 'right'?: Uncovering limitations of vision-and-language navigation models. 2

[10] Yicong Hong, Qi Wu, Yuankai Qi, Cristian Rodriguez-Opazo, and Stephen Gould. Vlnbert: A recurrent vision-and-language bert for navigation. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1643–1653, 2021. 2

[11] Aishwarya Kamath, Peter Anderson, Su Wang, Jing Yu Koh, Alexander Ku, Austin Waters, Yinfei Yang, Jason Baldridge, and Zarana Parekh. A new path: Scaling vision-and-language navigation with synthetic instructions and imitation learning. *ArXiv*, abs/2210.03112, 2022. 2

[12] Jacob Krantz, Erik Wijmans, Arjun Majumdar, Dhruv Batra, and Stefan Lee. Beyond the nav-graph: Vision-and-language navigation in continuous environments. In *European Conference on Computer Vision (ECCV)*, pages 104–120. Springer, 2020. 2

[13] Alexander Ku, Peter Anderson, Roma Patel, Eugene Ie, and Jason Baldridge. Room-across-room: Multilingual vision-and-language navigation with dense spatiotemporal grounding. *arXiv preprint arXiv:2010.07954*, 2020. 1, 2, 3

[14] Jialu Li, Hao Tan, and Mohit Bansal. Envedit: Environment editing for vision-and-language navigation. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15386–15396, 2022. 2

[15] Gabriel Ilharco 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. 1

[16] A. Moudgil, Arjun Majumdar, Harsh Agrawal, Stefan Lee, and Dhruv Batra. Soat: A scene- and object-aware transformer for vision-and-language navigation. In *NeurIPS*, 2021. 2

[17] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9982–9991, 2020. 2, 3, 6

[18] Shiquan Ren, Hong Lai, Wenjing Tong, Mostafa S. Aminzadeh, Xuezhang Hou, and Shenghan Lai. Nonparametric bootstrapping for hierarchical data. *Journal of Applied Statistics*, 37:1487 – 1498, 2010. 4

[19] Marco Tulio Ribeiro, Tongshuang Sherry Wu, Carlos Guestrin, and Sameer Singh. Beyond accuracy: Behavioral testing of nlp models with checklist. In *ACL*, 2020. 2

[20] Sheng Shen, Liunian Harold Li, Hao Tan, Mohit Bansal, Anna Rohrbach, Kai-Wei Chang, Zhewei Yao, and Kurt Keutzer. How much can clip benefit vision-and-language tasks? In *International Conference on Learning Representations*, 2021. 2, 8

[21] Hao Tan, Licheng Yu, and Mohit Bansal. Learning to navigate unseen environments: Back translation with environmental dropout. In *Proceedings of NAACL-HLT*, pages 2610–2621, 2019. 2, 8

[22] Guanqun Yang, Mirazul Haque, Qiaochu Song, Wei Yang, and Xue-Min Liu. Testaug: A framework for augmenting capability-based nlp tests. In *COLING*, 2022. 2

[23] Wanrong Zhu et al. Diagnosing vision-and-language navigation: What really matters. In *NAACL: Human Language Technologies*, pages 5981–5993, 2022. 2