Exploring Strategies for Efficient Real-World VLN Evaluation

Abrar Anwar*[1] Rohan Gupta*[1] Elle Szabo*[1] Jesse Thomason[1]

***Abstract*— Evaluations in the real world are time-consuming, and the relatively small number of experiments that can realistically be run may not explain the performance on the combinatorially large space of instructions that language can specify in complex scenes. In this work, we provide the first real-world evaluation of the Vision-and-Language Navigation in Continuous Environments (VLN-CE) task, a benchmark for evaluating language-guided navigation in simulation. To address the challenges of real-world evaluation in VLN-CE, we propose key desiderata for efficiently evaluating the linguistic and visual components of end-to-end robot policies. We introduce a contrast set-based evaluation based on our proposed criteria that strategically modify test instructions and scenes to efficiently gain component-level insights about a language-guided policy. We hope to spark discussion with the community on efficient evaluation of language-guided policies to bring these robots closer to real-world deployment.**

I. Introduction

To seamlessly integrate robots human environments, they must demonstrate the ability to understand and execute natural language instructions. Language can be used for providing guidance on tasks like high-level task planning, manipulation, and navigation. However, benchmarks for language-guided robotics are almost exclusively in simulation, such as VLMBench [1], ALFRED [2], and CALVIN [3]. Vision-and-Language Navigation (VLN) is a popular task for language-guided navigation, where an embodied agent receives a description on how to navigate to a goal location. Bringing these methods to the real world is difficult as real-world data collection is expensive and past work has found sim2real for VLN to be difficult [4]. Then evaluating these physical robot policies is time-consuming, especially as one needs to evaluate various policies. In this work, we propose desiderata to evaluate the language and visual components of language-guided robot policies efficiently, and we design a contrast set-based evaluation method to probe these components of VLN policies in the real world.

To evaluate a manipulation task, an experimenter has to simply move small tabletop items to modify a scene; however, navigation tasks are harder to evaluate as the environment is often difficult to change. Changing the environment in navigational settings often means moving furniture or adding new large objects, which is labor-intensive. Additionally, when language is involved, these objects must also be semantically-relevant. As a consequence, it is harder to probe the capabilities of a language-guided navigation policy at scale.

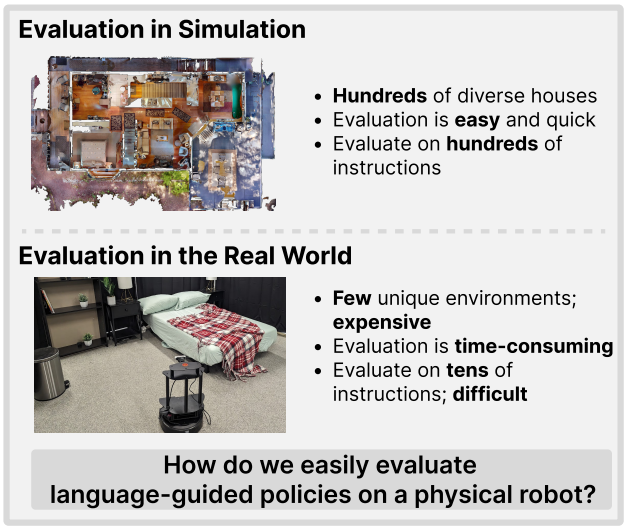

Fig. 1: Much work on Vision-and-Language Navigation (VLN) is conducted exclusively on simulation as there is a large amount of data to experiment on and it takes very little time to run evaluation. However, evaluating language-guided policies such as VLN in the real world is difficult as it is time-consuming while using less diverse scenes. In this work, we consider how to efficiently evaluate VLN policies on a physical robot.

We characterize three important components to consider when evaluating VLN policies: language, scene, and actions. Language serves as the source of task specification in the form of step-by-step instructions. Language must then be grounded through vision to understand object references and actions about how an agent should move. Subsequently, these instructions guide actions to be carried out within the environment, whether it is within a simulated environment with artificial actuator noise or on a physical robot.

We take inspiration from contrast sets [5] and propose to strategically perturb the language and scene components to understand the behavior of a VLN policy. For example, one can add new objects to a scene or present different instances of known objects. These scene-level perturbations probe a model's ability to generalize to different scenes. Similarly, these scene-level perturbations, or similar language-level perturbations, can effect the correct sequence of actions needed to succeed. These ablations are particularly important for language-guided tasks as performance hinges on a combination of the task specification and the environment.

*Authors contributed equally to this work.

[1]Abrar Anwar, Rohan Gupta, Elle Szabo, Jesse Thomason are with the Thomas Lord Department of Computer Science, University of Southern California, Los Angeles, CA, USA

Contact: abrar.anwar@usc.edu

The embodied AI community has introduced many simulated environments such as AI2-THOR [6], [7] and Habitat [8]. After training a policy on a task in these simulators, the policy is typically evaluated by comparing performance to a large number of predefined or collected testing examples within simulation. Since these simulations are often insufficient for understanding a policy's real-world performance [4], [7], it becomes evident that there is a need for a straightforward evaluation framework to assess language-guided robot policies in the physical world.

Therefore, a crucial objective in the field of VLN research is to understand the performance of an agent's linguistic and visual capabilities in the real world. In this work, we propose desiderata for VLN on physical robots that push toward targeted evaluation of real-world VLN policies. We explore the use of contrast sets as a first step in this direction of developing efficient protocols under these criteria. These criteria are intended to guide efforts toward the realization of practical language-guided robots suitable for the real world, and we are excited to further discuss methods for efficient evaluation with the community.

In particular, our contributions are:

- We showcase the first real-world sim2real transfer of VLN-CE on a physical robot, for which we provide additional ablations to provide insights on this task
- We propose several key desiderata for VLN evaluation methods that encourage targeted evaluation of the linguistic and visual capabilities of physical ground robots without the need for dramatic physical costs required to make simulations-sized test sets
- We showcase an evaluation method based on contrast sets that allow us to gain a component-level understanding of the overall performance of a language-conditioned robot policy

II. Related Work

We discuss methods and evaluations used for language-guided robot manipulation and navigation.

Language-Conditioned Robot Learning. Several works have focused on instructing robots with natural language, mostly focusing on navigation or manipulation. Some works deconstruct instructions by separating task planning and action generation [9]–[11], while others have unified, end-to-end architectures [12], [13]. Within language-guided manipulation, language is used to define the objects to be manipulated, specifying the desired end state or providing instructions on how to interact with specific objects. Therefore, probing the capabilities of these manipulation policies is relatively easy, as instructions can be easily changed. For example, changing the task instruction from "pick up the coke can" to "grasp the coffee mug" is simple. New objects are small and light, and the environment can be quickly reset.

Language-guided navigation, on the other hand, presents distinct challenges when evaluating these models' capabilities. Vision-and-language navigation focuses on using fine-grained instructions to control a navigation agent from visual observations [14]. Some work adds the ability for navigation agents to manipulate objects, further increasing complexity [2], [15]. SayCan [11] takes a modular approach and separates language understanding from taking actions to execute household tasks in the real world. Methods that are modular can often evaluate each component separately, such as a language model's output generation in SayCan. However, for end-to-end policies, it is expensive to evaluate a robot hundreds of times and understand which component contributes to performance improvement or degradation.

It is evident in these examples that simulated benchmarks are popular for training and evaluating robot policies as it is easy to scale. However, there is a need for well-defined probing criteria to effectively understand the capabilities of end-to-end language-guided navigation policies.

Vision-and-Language Navigation. Vision-and-Language Navigation (VLN) [14], [16] is a task for language-guided navigation, where an agent receives instructions on how to reach a goal location. There are multiple variations of the VLN task based on different task objectives. In this work, we focus on fine-grained navigation, where an agent is given step-by-step descriptions of its route. The Room-to-Room (R2R) dataset [16] contains these instructions in the Matterport3D simulator, where an agent must traverse the edges of a navigation graph. The RxR dataset [17] builds a larger dataset with English, Hindi, and Telegu instructions. Since these datasets use navigation graphs, it is not well-designed for physical robots. VLN-CE [18] converts trajectories in R2R and RxR on the navigation graph into a continuous environment in Habitat [8] that is more suitable for robots. In this work, we pretrain our policy on the RxR VLN-CE task and then finetune the policy on a robot.

Sim2Real Evaluation. Sim2real transfer strategies encompass various approaches, often employing domain randomization [19] or using generative adversarial networks to shift the target domain observation closer to the source domain [20]. However, this work's focus is not on the actual sim2real transfer process. Instead, our aim is to evaluate policies more effectively in the real world.

Simulated environments for understanding sim2real transfer, such as RoboTHOR [7], CODA [4], [21], and others [22], often recreate physical counterparts to run controlled experiments. These works generally show ineffective direct sim2real performance unless domain randomization or real-world finetuning strategies are used. While these environments are effective in evaluating task performance in simulation, there are no guarantees about real-world performance. VLN-CE poses unique challenges, including language grounding, compositionality of instructions, and diverse scenes. We find that a policy trained only on simulation does not perform well, and finetuning on real-world demonstrations improves performance. However, to *know* how well a policy performs in the real world and which components contribute to that performance requires a large number of evaluations which is expensive. In this work, we discuss strategies to evaluate these components by strategically perturbing these components.

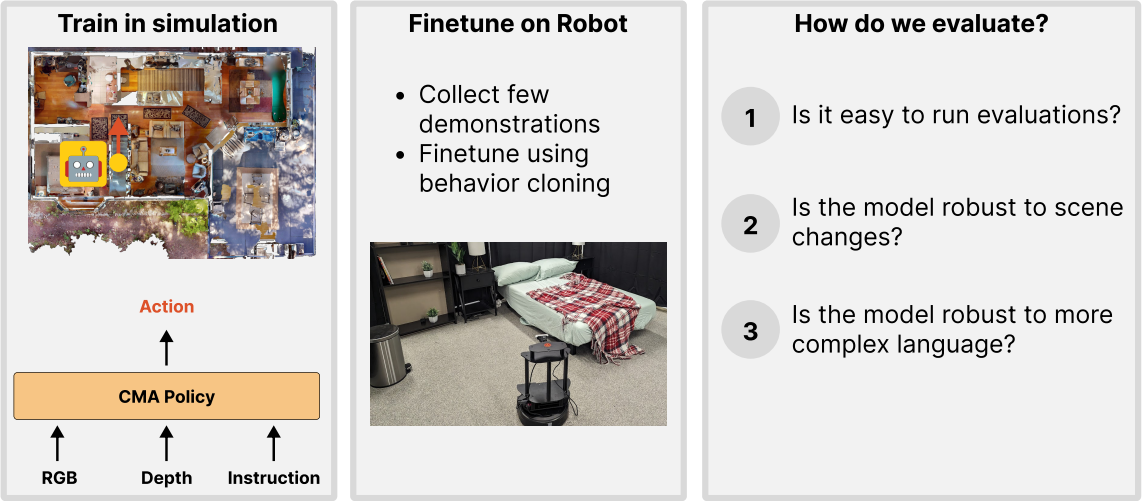

Fig. 2: To ensure we have a good policy in the real world, we first pretrain a Cross-Modal Attention (CMA) policy [18] in simulation on the VLN-CE tas. Then we use 50 collected real-world demonstrations to finetune our policy using behavior cloning. We then consider how to efficiently evaluate our policy based on three desiderata.

III. Desiderata for Real-Robot VLN-CE Evaluation

Recent work has focused on scaling training data for VLN policies [23], [24]; however, as the capabilities of these agents increase, the research community needs to develop criteria to probe what these methods learn. Unlike single-task or multi-task robot policies, language-guided policies can combinatorially scale in the number of instructions they can be given, so these policies are typically evaluated in simulation given thousands of language instructions. When brought to the physical world, several works have evaluated image-based navigation policies in Airbnbs or rented homes [22], [25]. The outcomes of these evaluations are often then aggregated, providing a measure of a policy's performance. For example, the RxR-Habitat Challenge test set of the RxR VLN-CE task has 17 unique houses with a total of 11k instructions, 1.3k of which are in English. Assuming we have access to many houses and conservatively estimate that a single trial takes an experimenter 5 minutes to set up and run, executing the 1.3k English instructions from the test set would take around 108 hours. This paradigm is clearly not scalable in the real world, as even a subset of this test set would require dozens of hours to evaluate just a single model. Therefore, the most fundamental desiderata for an evaluation framework for VLN-CE on physical robots is that it must be easy to run. Thus, we must carefully consider our expectations and capabilities for these robots as we develop other guidelines.

A VLN task consists of three components: language instructions, the scene, and the actions executed by the robot. Any VLN-CE method that is trained in a simulated environment should be able to transfer some key abilities into the physical world. Specifically, a language-guided policy should handle changes in the visual scene while still demonstrating proficiency in handling complex, long-horizon language instructions. Thus, we can define two more desiderata for an efficient evaluation framework: the framework should be capable of measuring a policy's robustness to visual perturbations and it should also measure a policy's robustness to compositional language instructions.

A. Desiderata for Evaluation

Easy to Run. Given that VLN evaluations must be conducted in the physical world, evaluation should prioritize targeted easy deployment. Rather than gathering a random set of instructions to evaluate a language-guided policy and computing aggregate metrics, evaluation methods should efficiently assess a policy's capabilities.

Robust within the scene. VLN agents in the physical world should be adaptable to new scenes. Visual scenes can vary drastically between houses, with objects in unexpected locations or lighting fluctuations causing objects to appear differently. An effective evaluation method should measure this robustness in the face of such visual differences. This entails assessing the model's capability to navigate and make decisions in real-world scenarios that may deviate from its training environment.

Robust in language. Language understanding in VLN-CE extends beyond simpler single-step instructions. Models should demonstrate robustness in handling complex, multi-step language compositions that involve sequential reasoning. An evaluation method should encompass these language compositions, measuring the model's ability to comprehend and execute instructions of varying lengths and levels of complexity.

To design an evaluation method around our proposed desiderata, we propose selectively perturbing various axes that can tell us more about the component-level capabilities of our agent. We can measure the differences between these perturbations to gain insights about the performance of each of these axes. In particular, we propose investigating

ΔL	ΔV	ΔT	Language Instruction (**L**) and Scene (**S**)
Original Instruction			**L:** "You are currently facing a couch. Turn left and you will see a bed. Go to the bed and then stop when you reach it." **S:** Seen environment
✓			**L: (Rephrase)** "You find yourself in front of a couch. Rotate left, and there's a bed. Head to the bed and come to a stop upon reaching it." **S:** Same as Original Instruction
✓		✓	**L:** "You are currently facing a couch. Turn left and you will see a **nightstand**. Go to the **nightstand** and then stop when you reach it." **S:** Same as Original Instruction
	✓		**L:** Same as Original Instruction **S: Change the color of the bed**
	✓	✓	**L:** Same as Original Instruction **S: Move the bed much further away**

TABLE I: Given an original instruction, we propose four different perturbation strategies on the language instruction (ΔL), the visual scene (ΔV), and the expected trajectory (ΔT): *language changes, trajectory same*; *language changes, trajectory changes*; *vision changes, trajectory same*; and *vision changes, trajectory changes*.

three axes: language, vision, and trajectories. We use four experiments: two each for probing the robustness in language and the robustness within the environment.

IV. VLN-CE Evaluation on a Physical Robot

In this work, we use a Locobot [26] robot platform to run VLN-CE in the real world. We pretrain a policy for this robot on the VLN-CE task in the Habitat simulator using the RxR training set. To facilitate evaluation, we designed a physical VLN-CE test environment. This environment is populated with furniture similar to those found in simulation to resemble a studio apartment.

A. Robot Platform

We use a low-cost Locobot robot with an iRobot Create 3 base [27]. The Locobot uses an Intel Realsense D435 camera for RGB and depth images. In addition, the Create 3 base has a bump sensor that reverses backward if the sensor is triggered after a collision. All compute is done onboard on an Intel NUC. Similar to the simulator, the action space for the robot is constrained to forward, left 30 degrees, and right 30 degrees actions. While the simulator has no drift when taking actions, the physical robot has noisy actions and does not have perfect odometry information. We expect that finetuning in the real world accounts for this drift.

B. Simulator Training

We train our policies in the Habitat simulator using imitation learning. We use the Cross-Modal Attention (CMA) model from VLN-CE [18]. We train the policy for 10 epochs on the RxR dataset using teacher forcing. Unlike other work using the VLN-CE simulator which uses panoramic inputs, we do not use a panoramic camera. On standard robot platforms, the use of a panoramic input would require a robot to spin in place to build a 3D observation at every move, which contradicts our easy-to-run criteria as it is unrealistic and inconvenient for real-world evaluation. Additionally, we train in the VLN-CE simulator with a height that is shorter than the original CMA agent in VLN-CE to match the height of the LoCoBot. The pretrained policy achieves a success rate of 15.2% and an SPL of 0.15 on the validation unseen set.

C. Real-World Room Design

For the evaluation of our VLN-CE agent, we constructed an artificial room. The 100-square-foot room is built to resemble a simple studio apartment. Unlike past work that requires renting large spaces [22], [25], which does not lead to easy-to-run evaluations, we intentionally designed our space to be cost-effective. Overall, the cost was approximately $500 and can be easily iterated on. We construct the walls using wheeled, standing curtains so that the depth sensor can detect the makeshift walls. This choice also lets us construct multiple rooms within the space and easily change the floorplan. We populate the studio apartment with a bed, couch, bookshelf, nightstands, and plants.

We do note that our room design is still artificial and lacks clutter and thus lacks visual diversity. However, we are trying to evaluate the robustness of our policies to various kinds of visual and linguistic shifts with our real-world finetuned policy, so we believe that this setup is sufficient for the purposes of this work.

D. Navigation Instructions

Each step-by-step instruction can be decomposed into a series of subgoals that correspond to a single object an agent has to reason over. We consider each subgoal as a "step," and an instruction with n-subgoals is an n-step instruction. For instance, a 1-step instruction would involve reasoning over a singular object, as exemplified by an instruction like "go walk towards the bed to your left and stop in front of it". A 2-step instruction, such as "drive towards the bed, and towards the right you will see a nightstand. stop in front of it," introduces the challenge of navigating and reasoning over two objects.

To finetune our policy in the real world, we construct a real-world training dataset. The training instructions are based on our physical training setup. We created 25 instructions: 17 1-step instructions, 6 2-step instructions, and 2 3-step instructions. Each instruction has a predefined starting position, orientation, and goal destination for the robot. The navigation instructions were similar to those in simulation training, with common words and phrases like "go," "towards," "move," "turn," "forward," "stop," "to your

right," and "to your left," and the names of the objects. We collected two demonstrations per instruction for a total of 50 episodes, each consisting of sequential RGB, Depth, and action data.

E. Real-World Finetuning

Using this real-world training dataset, we use behavior cloning to finetune a policy originally trained in simulation. The contribution of this work is to explore our evaluation framework, so we use finetuning as a simple method to take our policy from simulation to reality. This choice of using a finetuned policy allows us to focus on the evaluation framework rather than explore sim2real strategies such as domain randomization. We also avoid issues with visual distribution shift but may have issues with transferring key capabilities from simulation, which our evaluation framework will allow us to investigate. We hope that our framework can inspire more work on physical robots for VLN-CE.

V. Contrast Set Evaluation

Given our three desiderata, we want our evaluation to be easy to run, robust with the environment, and robust in language. In line with recent work in contrast sets [5], we propose to perturb different axes with respect to a set of original instructions so that we can efficiently gain insights on the component-level performance of our policies.

In NLP, contrast sets are perturbed variants of the test set that help characterize the decision boundary of a classification model. A contrast set is a collection of perturbed instances that are tightly clustered in input space around a single test instance. Evaluating on a contrast set allows one to measure how similar a model's decision boundary is to the correct decision boundary in the neighborhood of the contrast set. In NLP datasets, contrast sets for language are constructed by perturbing the input or output such that the meaning/label of a test instance is inverted. For image-based perturbation strategies, the NVLR2 [28] contrast set perturbs an image by finding a new image that makes one minimal change in some concrete aspect. In this work, we design similar perturbation strategies in the language, image, and expected trajectory axes.

We propose four contrast set-based experiments to get us a better understanding of our robot's policy, with examples depicted in Table I. Given a specific instruction and environment, we investigate robustness within the environment in two ways: 1) perturb the environment such that the expected trajectory is the same and 2) perturb the environment such that the expected trajectory is different. In the first case, we simply add objects that change what the agent sees, such as changing bedsheets or adding new distractor objects to the scene. This contrast set probes the visual robustness of a model directly. In the second case, we move furniture around the room so that they are located in different places such that the expected trajectory is different while the language instruction remains the same. This involves moving objects further or closer to the robot in the start scene at varying degrees so the robot has to take a different path. This contrast set probes the model's ability to move around and find objects that may be in more difficult locations.

ΔL	ΔV	ΔT	Success Rate	Percent Completion	Percent Completion Difference
Original Instruction			30	50.0	-
✓			10	38.3	-11.7
✓		✓	30	55.0	$+5.0$
	✓		0	28.3	-21.7
	✓	✓	40	65.0	$+15.0$

TABLE II: Evaluation of contrast sets on success rate, percent completion, and the difference between contrast set against the original instruction. We find performance degradation for *language changes, trajectory same* and *vision changes, trajectory same* suggesting weaknesses in language and vision robustness.

Similarly, to investigate robustness in language, we use two more experiments: 1) we perturb the language instruction such that the expected trajectory is the same and 2) perturb language instruction such that the expected trajectory is different. In the first case, the instruction is simply reworded. This contrast set tests a policy's ability to handle textual variations. In the second case, the instruction is changed to refer to different objects in the same room such that the expected trajectory is different. This contrast set tests a policy's ability to generalize to other objects.

We define a set of five original instructions given a fixed scene that are perturbed according to each of the experiments. We evaluated each instruction two times for a total of 10 trials per experiment.

To evaluate any given policy, we need a metric to determine performance. Since every task has 1, 2, or 3 subgoals, using an overall success rate is likely too coarse of a measure. We choose to focus on percent subgoal completion based on the n-step instructions. We also investigate the Percent Completion Difference between the original instructions and scene versus the perturbed contrast sets. This is simply the difference between the average Percent Completion of perturbed instances minus the original experiments.

VI. Results

Given our contrast set evaluation, we investigate what insights we are able to learn about our VLN-CE policy. Additionally, we provide ablations on our real-world policy.

A. Contrast Set Evaluation

How well does our model perform? Table II illustrates our policy's performance on the physical robot. We denote ΔL, ΔV, ΔT to represent language, visual scene, and trajectory changes with respect to the original instructions, respectively. We find that with the original instructions, our policy has a percent subgoal completion rate of 50% However, only 30% of instructions are completed in full. Each experiment has five instructions and ten trials, for a total of 50 trials.

Our experiments allow us to understand our robot's performance in a few key dimensions. First, we find through

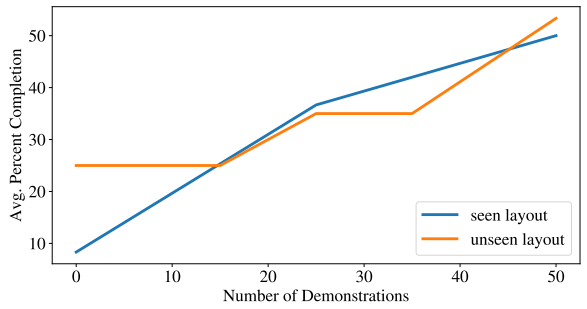

Fig. 3: Average percent completion in seen and unseen scene layouts as a function of the number of demonstrations used in finetuning. As more data is provided, performance improves linearly across scenes.

the *language changes, trajectory same* experiment that the Percent Completion Difference is negative. This directly means that our policy is not robust to rewording. However, since our rewording strategy was relatively simple, this may suggest that experiments that perform poorly might be severely impacted simply due to the wording of the instruction itself, not the semantic meaning of the instruction.

Second, the *vision changes, trajectory same* experiment reveals that seemingly minor visual alterations, like adding a plant or changing bed sheet colors, significantly reduce success rates. The Percent Completion Difference of -21.7 means these visual changes are causing a dramatic drop in performance. Often these objects are not in the instruction and are simply in the background. They do not require the policy to reason over these objects. This indicates that our policy has a weakness in visual robustness, which it was not able to fully transfer from its training in simulation.

Third, the *vision changes, trajectory changes* experiment, and *language changes, trajectory changes* experiment demonstrated positive Percent Completion Difference results. The performance gains on these contrast set experiments suggest that our model surprisingly exhibits robustness to furniture rearrangements and new instructions, respectively. These improvements might indicate that these perturbed scenarios were easier and motivate future work that considers the impact of environmental factors on performance.

Overall, we ran a total of 50 trials split among the original instructions and four contrast set experiments. We argue that our targeted contrast set evaluation offers richer insights than aggregated experiments, allowing us to get a fuller picture of our policy's capabilities. If we had simply run 50 randomly selected instructions and scenes, we would not have been able to gain insights through this comparative evaluation. Therefore, our proposed evaluation protocol offers quick and reliable insights into a policy's performance through the comparison of original instances against perturbed contrast sets. Our four experiments collectively provide us with a better holistic understanding of our policy's performance and how to improve it.

Train/Test	Percent Completion All Data	Percent Completion 1-step	Percent Completion 2-step	Percent Completion 3-step
All Data	53.0	25.0	75.0	66.6
1-step only	31.6	0.0	62.5	33.3
2-step only	30.0	0.0	50.0	50.0
3-step only	18.0	0.0	37.5	16.5

TABLE III: Ablations on Compositional Generalization of Instructions. Given a policy trained on *n*-step data, we find that the model trailed on *All Data* performs the best across *n*-step evaluation splits.

B. Ablations

In addition to the contrast set evaluation, we provide various feature ablations on the original instructions during the finetuning step to get a sense of what the model learned.

How much data do we need? To explore the data requirements for finetuning our policy on the physical robot, we analyze the relationship between the number of demonstrations used for training on the physical robot and the average percent completion of tasks, as depicted in Figure 3. In the previous experiments, the original scene was the same as that in the finetuning dataset. We define an unseen house, where the position of furniture – specifically the bookshelf, the table, and the couch – is moved around. This is meant to investigate whether the policy implicitly learned the locations of these objects. We find that both the seen and unseen sets perform similarly.

The policy improves linearly on both the seen and unseen layouts as the policy is finetuned on more data without plateauing, suggesting that more data could potentially enhance the policy's performance.

To our knowledge, Anderson et al. (2021) [4] is the only sim2real work on VLN. They found that sim2real transfer on VLN using topological navigation graphs performs poorly unless map data is provided in advance. We have a similar problem formulation; however, we evaluate in our reconstructed physical setup using continuous actions and a single camera. In contrast, Anderson et al. (2021) [4] use a large office space using a waypoint-based method given panoramic views. Similar to their results, we find that the policy performs poorly 0-shot from simulation, most likely due to the large visual distribution shift. Although we acknowledge these visual distribution shifts may lead to worse performance, the emphasis of this work is on efficient evaluation, so we are not overly concerned with collecting additional data.

Does compositional generalization of instructions transfer well? In Table III, we present additional experiments to understand the performance between different subsets of *n*-step instructions. We trained a model on only 1-step, 2-step, or 3-step data, and evaluated it on the robot. We would like to note that this evaluation encompasses a total of 10 trials for each model: four trials featuring 1-step instructions, four with 2-step instructions, and two with 3-step instructions. We hope to increase this evaluation set to

better understand which subset of data contributes most to performance. We find that training on all the data leads to the highest percent completion.

VII. Conclusion

In this work, we explore ways to evaluate VLN-CE policies trained on physical hardware. We address fundamental desiderata crucial for assessing language-guided policies in real-world settings. The first criterion for our evaluation framework is ease of use, acknowledging the practical constraints of deploying robots in physical environments. We propose and carry out pilot studies using contrast set perturbations of test instances that probe linguistic and visual variations. Our real-world evaluations easily unveil problems that exist in language and visual reasoning. After training a finetuned policy on VLN-CE, we are able to isolate potential performance issues in the linguistic reasoning and visual reasoning components of our policy. In this work, we only focused on evaluating a single policy. In future work, we hope to use this approach to quickly compare the performance of multiple language-guided policies.

In conclusion, our work serves as an initial step toward the practical deployment of language-guided robots in real-world scenarios. By emphasizing ease of evaluation and probing of language-guided policies, we hope this work can initiate a community discussion on how to efficiently evaluate these robot policies.

References

[1] K. Zheng, X. Chen, O. C. Jenkins, and X. Wang, "Vlmbench: A compositional benchmark for vision-and-language manipulation," *Conference on Neural Information Processing Systems (NeurIPS)*, 2022.

[2] M. Shridhar, J. Thomason, D. Gordon, Y. Bisk, W. Han, R. Mottaghi, L. Zettlemoyer, and D. Fox, "Alfred: A benchmark for interpreting grounded instructions for everyday tasks," *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[3] O. Mees, L. Hermann, E. Rosete-Beas, and W. Burgard, "Calvin: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks," *IEEE Robotics and Automation Letters*, 2022.

[4] P. Anderson, A. Shrivastava, J. Truong, A. Majumdar, D. Parikh, D. Batra, and S. Lee, "Sim-to-real transfer for vision-and-language navigation," *Conference on Robot Learning*, 2021.

[5] M. Gardner, Y. Artzi, V. Basmova, J. Berant, B. Bogin, S. Chen, P. Dasigi, D. Dua, Y. Elazar, A. Gottumukkala *et al.*, "Evaluating models' local decision boundaries via contrast sets," *Findings of the Association for Computational Linguistics: EMNLP*, 2020.

[6] E. Kolve, R. Mottaghi, W. Han, E. VanderBilt, L. Weihs, A. Herrasti, M. Deitke, K. Ehsani, D. Gordon, Y. Zhu *et al.*, "Ai2-thor: An interactive 3d environment for visual ai," *arXiv:1712.05474*, 2017.

[7] M. Deitke, W. Han, A. Herrasti, A. Kembhavi, E. Kolve, R. Mottaghi, J. Salvador, D. Schwenk, E. VanderBilt, M. Wallingford *et al.*, "Robothor: An open simulation-to-real embodied ai platform," *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[8] M. Savva, A. Kadian, O. Maksymets, Y. Zhao, E. Wijmans, B. Jain, J. Straub, J. Liu, V. Koltun, J. Malik *et al.*, "Habitat: A platform for embodied ai research," *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[9] I. Singh, V. Blukis, A. Mousavian, A. Goyal, D. Xu, J. Tremblay, D. Fox, J. Thomason, and A. Garg, "Progprompt: Generating situated robot task plans using large language models," *International Conference on Robotics and Automation (ICRA)*, 2023.

[10] J. Liang, W. Huang, F. Xia, P. Xu, K. Hausman, B. Ichter, P. Florence, and A. Zeng, "Code as policies: Language model programs for embodied control," *International Conference on Robotics and Automation (ICRA)*, 2023.

[11] M. Ahn, A. Brohan, N. Brown, Y. Chebotar, O. Cortes, B. David, C. Finn, C. Fu, K. Gopalakrishnan, K. Hausman *et al.*, "Do as i can, not as i say: Grounding language in robotic affordances," *arXiv:2204.01691*, 2022.

[12] M. Shridhar, L. Manuelli, and D. Fox, "Cliport: What and where pathways for robotic manipulation," *Conference on Robot Learning*, 2022.

[13] ——, "Perceiver-actor: A multi-task transformer for robotic manipulation," *Conference on Robot Learning*, 2023.

[14] J. Gu, E. Stefani, Q. Wu, J. Thomason, and X. E. Wang, "Vision-and-language navigation: A survey of tasks, methods, and future directions," *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2022.

[15] S. Yenamandra, A. Ramachandran, K. Yadav, A. Wang, M. Khanna, T. Gervet, T.-Y. Yang, V. Jain, A. W. Clegg, J. Turner *et al.*, "Homerobot: Open-vocabulary mobile manipulation," *arXiv:2306.11565*, 2023.

[16] P. Anderson, Q. Wu, D. Teney, J. Bruce, M. Johnson, N. Sünderhauf, I. Reid, S. Gould, and A. Van Den Hengel, "Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments," *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[17] A. Ku, P. Anderson, R. Patel, E. Ie, and J. Baldridge, "Room-across-room: Multilingual vision-and-language navigation with dense spatiotemporal grounding," *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020.

[18] J. Krantz, E. Wijmans, A. Majumdar, D. Batra, and S. Lee, "Beyond the nav-graph: Vision-and-language navigation in continuous environments," *European Conference on Computer Vision (ECCV)*, 2020.

[19] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," *International Conference on Intelligent Robots and Systems (IROS)*, 2017.

[20] K. Rao, C. Harris, A. Irpan, S. Levine, J. Ibarz, and M. Khansari, "Rl-cyclegan: Reinforcement learning aware simulation-to-real," *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[21] A. Kadian, J. Truong, A. Gokaslan, A. Clegg, E. Wijmans, S. Lee, M. Savva, S. Chernova, and D. Batra, "Sim2real predictivity: Does evaluation in simulation predict real-world performance?" *IEEE Robotics and Automation Letters*, 2020.

[22] T. Gervet, S. Chintala, D. Batra, J. Malik, and D. S. Chaplot, "Navigating to objects in the real world," *Science Robotics*, 2023.

[23] S. Wang, C. Montgomery, J. Orbay, V. Birodkar, A. Faust, I. Gur, N. Jaques, A. Waters, J. Baldridge, and P. Anderson, "Less is more: Generating grounded navigation instructions from landmarks," *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[24] A. Kamath, P. Anderson, S. Wang, J. Y. Koh, A. Ku, A. Waters, Y. Yang, J. Baldridge, and Z. Parekh, "A new path: Scaling vision-and-language navigation with synthetic instructions and imitation learning," *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[25] S. Y. Min, Y.-H. H. Tsai, W. Ding, A. Farhadi, R. Salakhutdinov, Y. Bisk, and J. Zhang, "Self-supervised object goal navigation with in-situ finetuning," *International Conference on Intelligent Robots and Systems (IROS)*, 2023.

[26] "LoCoBot," https://www.trossenrobotics.com/locobot-overview.aspx.

[27] "iRobot Create3 Robot," https://edu.irobot.com/what-we-offer/create3.

[28] A. Suhr, S. Zhou, A. Zhang, I. Zhang, H. Bai, and Y. Artzi, "A corpus for reasoning about natural language grounded in photographs," *Association for Computational Linguistics (ACL)*, 2019.