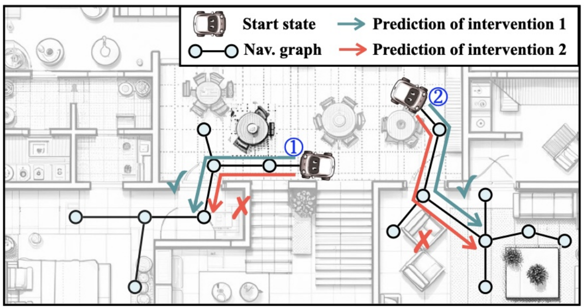
Navigating the Nuances: A Fine-grained Evaluation of Vision-Language Navigation

Anonymous ACL submission

Abstract

This study presents a novel evaluation framework for the Vision-Language Navigation (VLN) task. It aims to diagnose current models for various instruction categories at a finer-grained level. The framework is structured around the context-free grammar (CFG) of the task. The CFG serves as the basis for the problem decomposition and the core premise of the instruction categories design. We propose a semi-automatic method for CFG construction with the help of Large-Language Models (LLMs). Then, we induct and generate data spanning five principal instruction categories (*i.e.,* direction change, landmark recognition, region recognition, vertical movement, and numerical comprehension). Our analysis of different models reveals notable performance discrepancies and recurrent issues. The stagnation of numerical comprehension, heavy selective biases over directional concepts, and other interesting findings contribute to the development of future language-guided navigation systems.



Example ①:
Walk past the table, turn **left** into the bedroom, walk past the cabinet …
Intervention pair:
1. Walk past the table, turn **left** and walk straight.
2. Walk past the table, turn **right** and walk straight.

Example ②:
Go into the living room, walk towards **the table.** Walk over the rug …
Intervention pair:
1. Go into the living room, walk towards the **table**.
2. Go into the living room, walk towards the **picture**.

Figure 1: Examples of constructed interventions for VLN instructions. Example 1 demonstrates an intervention related to directional concepts, while Example 2 focuses on landmarks. Nonetheless, a subset of the model's predictions remains unchanged following the intervention, suggesting a deficiency in the model's ability to grasp underlying concepts.

1 Introduction

In the Vision-Language Navigation (VLN; Anderson et al. 2018) task, an agent is instructed to navigate through virtual environments by following detailed natural language instructions. This task requires an understanding of the interplay between natural language instructions, visual cues, and the sequence of actions undertaken by the agent. This capability is crucial for a wide range of robotic applications, extending from healthcare support to everyday household assistance.

Despite significant advancements in the latest research, we argue that the performance of VLN models may be overestimated. The current standard for evaluating vision-language navigation, as exemplified by the Room-to-Room (R2R; Anderson et al. 2018) and Room-across-Room (RxR; Ku et al. 2020) datasets, predominantly hinges on endpoint success rates and broad path alignment metrics. The recent work (Wang et al., 2023) suggests the performance of the state-of-the-art is high and even quite close to human performance on these standards. Does this mean that the major challenges of the VLN task are almost solved? This perspective might be overly optimistic. For instance, the high success rate of a randomly navigating agent (Anderson et al., 2018) is non-negligible. This indicates that current evaluation metrics may be insufficiently detailed. Furthermore, agents enhanced by Large Multimodal Models (LMMs; Zhou et al. 2023; Lin et al. 2024) perform unexpectedly low on standard VLN datasets. This contrasts with the strong multimodal understanding demonstrated by LMMs in other domains (Fu et al., 2024; Wake et al., 2023). This discrepancy motivates us to revisit the evaluation of VLN models.

In this work, we introduce a new evaluation framework that focuses on *atomic instructions*, *i.e.,* the singular actions fundamental to VLN instructions. Diagnosing VLN models at the atomic-instruction level allows us to gauge performance through various nuanced perspectives. To achieve this, we first iteratively construct a context-free grammar (CFG; Hopcroft et al. 2001) with the help of LLMs to systematically articulate the structure of VLN task instructions. CFG, treated as a comprehensive representation of VLN instructions, allows us to induct and define atomic instruction categories. We group the components in our CFG into five main categories (*i.e.,* direction change, vertical movement, landmark recognition, region recognition, and numerical comprehension) and generate data accordingly to form our novel evaluation dataset NAVNUANCES. For each entry in NAVNUANCES, a candidate path is determined by the specific path proposing strategy according to its instruction category. The instruction is then generated using CFG and further enriched by LLMs. To ensure the data correctness, we incorporate human refinement into this automated generation process in the end. The rigorous evaluation protocols in our dataset pose significant challenges, as they require models to demonstrate a thorough understanding of individual concepts.

We benchmark various types of models based on our proposed evaluation framework. Experiments with NAVNUANCES expose model discrepancies and common issues. We observe that recent advancements in the standard R2R dataset primarily stem from enhanced capabilities in vertical movement and region recognition. Despite this progress, numerical comprehension shows stagnation across various models. In terms of specific models, zero-shot agents enhanced by LLMs demonstrated even significant superiority over traditional supervised ones in handling changes in direction and recognizing landmarks. Traditional supervised approaches suffer from selective bias, often leading to deficiencies in adapting to shifts in atomic concepts, as demonstrated in Figure 1.

Our contributions are threefold: **Firstly**, we devise a comprehensive evaluation framework that addresses diverse facets of Vision-and-Language Navigation (VLN) at a granular level. **Secondly**, our work includes a thorough benchmarking of prevalent methodologies on ninety diverse scenes, coupled with an in-depth analysis. The experiments demonstrate the deficiencies and differences in the capabilities of previous models, providing valuable insights for advancing the development of VLN methods. **Thirdly**, we present a zero-shot baseline as a minor contribution, which enhances NavGPT (Zhou et al., 2023) with GPT-4-vision (Achiam et al., 2023) integrating direct vision-instruction alignment.

2 Related Work

2.1 Vision-language navigation Datasets

Vision-Language Navigation (VLN; Anderson et al. 2018) tasks integrate language guidance within embodied environments. This task is initially introduced by the Room-to-Room dataset (R2R; Anderson et al. 2018) which requires step-by-step navigation in virtual spaces. Subsequent research expanded this framework through variations like multilingual RXR datasets (Ku et al., 2020) and addressed more complex navigation challenges. The advent of conversational interfaces led to interactive VLN tasks, exemplified by CVDN (Thomason et al., 2020) and Teach (Padmakumar et al., 2022), fostering navigation via dialogue interpretation. Concurrently, efforts like VLN-CE (Krantz et al., 2020) aimed to transition VLN tasks into continuous environments. Despite these advancements, a nuanced evaluation of VLN models on atomic-level instructions remained underexplored. Our work addresses this by developing a dataset specifically designed to assess the fundamental capabilities of VLN agents, thereby contributing to the refinement of models across various VLN settings.

2.2 Models in VLN tasks

The introduction of the R2R dataset (Anderson et al., 2018) catalyzed the development of numerous models focusing on VLN tasks in discrete environments. Early efforts, such as the Seq2Seq (Anderson et al., 2018) and RCM (Wang et al., 2019) models, emphasized training strategies leveraging Imitation and Reinforcement Learning within a conventional front-view framework. Subsequent innovations like CLIP-ViL (Shen et al., 2021) augmented these models with advanced visual features from CLIP (Radford et al., 2021). Attention then turned to the effective encapsulation of historical data, with approaches like VLN-BERT (Hong et al., 2021) utilizing recurrent transformer structures, and HAMT (Chen et al., 2021) compactly encoding historical visual cues. More recent endeavors (Chen et al., 2022; An et al., 2023) have explored the integration of topological or even metric maps to enrich navigational contexts. Parallel to these model-centric advancements, initiatives such as ScaleVLN (Wang et al., 2023) aimed at scaling up training data. More recently, the research focus has switched to exploring VLN with LLMs (Zhou et al., 2023; Long et al., 2023; Chen et al., 2023; Lin et al., 2024). Despite these significant strides, a comprehensive understanding of how these methodologies enhance specific VLN abilities, particularly atomic instruction comprehension, remains unclear. Our work seeks to shed light on this fundamental aspect and offers insights into the underlying capabilities necessary for effective VLN.

List 1: Context-free grammar (partial)

```
$S \rightarrow Vp$
$Vp \rightarrow$ ActionT
      $|$ActionS
      $|$ActionO $+$ Landmark
      $|$ActionR $+$ Region
      $|Vp + Vp$
      $|Vp + Ir$
$Ir \rightarrow$ ($action\ irrelevant\ sentence$)
$Numerical \rightarrow$ first$|$second$|$third$|\ldots$
$Room \rightarrow$ room$|$kitchen$|$bathroom$|\ldots$
$Direction \rightarrow$ left$|$right
$Object \rightarrow$ bed$|$table$|$chair$|\ldots$
$Attribute \rightarrow$ red$|$yellow$|\ldots$
$Modifier \rightarrow Attribute|...|\epsilon$
$Landmark \rightarrow Modifier + Object$
$ActionO \rightarrow$ “walk past”$|$“walk towards”$|$...
...
```

3 NavNuances Dataset

The challenge of curating a nuanced dataset is to comprehensively cover the atomic categories in VLN instructions. To achieve this, our approach begins by iteratively constructing a context-free grammar (CFG) with the help of LLM to articulate and cover all components of VLN instructions in a unified representation (Section 3.1). Then, we induct and categorize the atomic components of the CFG into five principal categories (Section 3.2). Building on these categorizations, we develop a semi-automatic process for data annotation of each atomic instruction category, adhering to the CFG-defined natural instruction standards (Section 3.3).

3.1 The Context-Free Grammar for VLN

Our CFG defines a set of rules and concepts that structure the instructions in VLN. It can be formalized as a quadruple, *i.e.,* CFG $= (N, T, P, S)$. Non-terminals N (in uppercase such as Landmark in List 1) represent broader conceptual categories or composite concepts. Terminals T signify specific actionable elements or descriptors and are denoted by lowercase words (*e.g.,* left, right). Production Rules P within the CFG outline how various elements are combined to form higher-level Non-terminals. And Start Symbol S triggers the instruction generation process. An illustrative instruction such as *walk past the red chair* can be generated by the pattern “*ActionO*+ *Landmark*(*Modifier(Attribute)* + *Object*)" in List 1. The complete version of CFG is available in the supplementary materials (Appendix E).

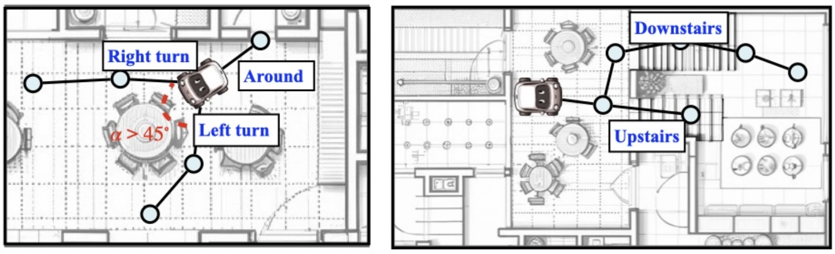
To ensure the integrity and completeness of the CFG, we instruct GPT-4 (Achiam et al., 2023) to parse the instructions in standard datasets (R2R (Anderson et al., 2018) and RxR (Ku et al., 2020)) using the CFG and identify any omissions in the current CFG. Through an iterative refinement process incorporating manual adjustment, we continuously update the CFG until GPT-4 can no longer detect missing components. An example is illustrated in Appendix I. The final CFG is defined at the conceptual level and ignores linguistic variations linked to the same concept. For example, the phrases “move towards" and “go towards" are both represented by the same terminal “walk towards" in CFG.

3.2 Atomic Instruction Categories

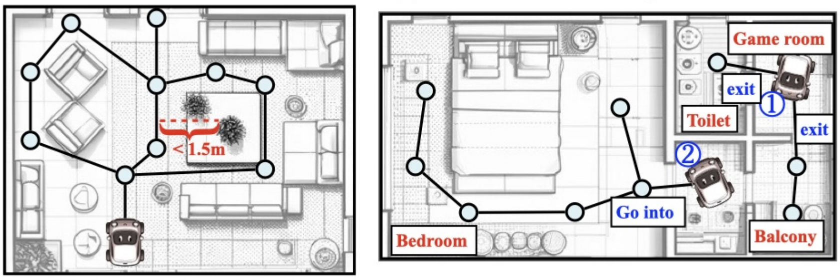
CFG provides a comprehensive yet elegant representation of VLN instructions. Based on this, we can discern what kind of concepts or patterns are fundamental to the VLN instructions. This further enables us to induct the atomic instruction categories. We define five primary categories introduced below:

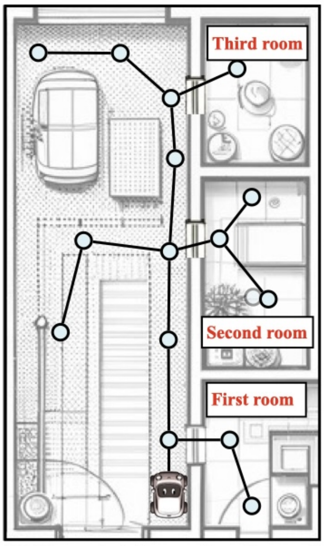
Direction Change: This category stems from the CFG’s *ActionT*, which encapsulates turning actions. These turning actions are distinct because they exclusively pertain to directional changes and are independent of the agent’s observations.

Vertical Movement: Derived from the *ActionS*, this category is associated with movements in the vertical plane, such as ascending or descending stairs. It highlights the agent’s interactions with



(a) Direction Change: turn left/right/around **(b) Vertical Movement:** Go upstairs/downstairs





(d) Landmark Recognition: Walk past/towards the glass table with plants on top.

(e) Region Recognition: 1. Exit the game room. 2. Walk into the bedroom

(c) Numerical Comprehension: walk along the corridor and enter the first/second/third room on your right.

Figure 2: Schematic diagram of annotation criteria for five main categories in the NavNuances dataset.

vertical elements in the environment.

Numerical Comprehension: Numerical comprehension, *i.e.,* being able to count or even calculate, is quite hard yet crucial for an agent. This is challenging even for the latest LLMs (Stolfo et al., 2023; Lu et al., 2023). We propose to separately evaluate this category in the VLN task. It focuses on instructions that incorporate an unambiguous numerical concept, aiding in the identification of multiple landmarks or regions.

Landmark Recognition: This category is inducted from production rules that involve a singular center object. It encompasses instructions that direct the agent towards or past specific landmarks within the environment.

Region Recognition: Similar to Landmark Recognition, this category pertains to instructions related to distinct areas or rooms.

3.3 Dataset Construction

Our dataset is collected from 90 Matterport (Chang et al., 2017) virtual environments aided by the semantic annotations in the Habitat simulator (Szot et al., 2021). Our annotation workflow consists of four stages: rule-based candidate path proposing, CFG-driven instruction crafting, human refinement, and linguistic enrichment via rephrasing by LLMs. Each annotated datum contains a natural language instruction, the agent's initial pose, and annotations for evaluation purposes, such as the ground truth paths or landmark locations. We finally annotated 1787 data in total. The statistics and examples of NavNuances are detailed in the Appendix D.

Direction Change category: Instructions in the direction change category direct the agent to make turns. We eliminate ambiguity by selecting junctions with a clear divergence in path directions (adjacent paths exceeding a large angle between them as depicted in Figure 2a), ensuring distinct navigation choices. We integrate forward movement into the instruction crafting to accommodate different VLN models and to facilitate evaluation, leading to concise instructions like *"turn left/right/around, then walk straight"*. Humans are involved in refining the dataset by excluding starting positions without central obstacles in the view. This exclusion is necessary; some instances meet the selection criteria only because the navigation graph is sparse. We additionally annotate the paired instructions for left and right turns with the same starting view.

Vertical Movement category: Vertical movement in VLN tasks is typically confined to ascending or descending stairs. Therefore, we identify the longest paths within 3D bounding boxes labeled by 'stairs' in each environment. The instruction template is straightforward containing only *"go upstairs/downstairs and stop on the next floor"*. Given the bounding boxes' imprecision, human annotators are involved to adjust the start and end positions. For views that encompass two staircases in opposite vertical directions, annotators are instructed to mark these special positions and annotate paired paths from the same starting viewpoint, as shown in Figure 2b. This subset is small but important for assessing awareness of the vertical direction. The human-refined trajectories are considered as the ground truth and included in the dataset for evaluation purposes.

Numerical comprehension category: This category emphasizes the memory of sequential elements and instance-level identification. We focus on region-level numerical comprehension, utilizing the semantic annotations of 'hallway'. The process begins by filtering out hallways with insufficient doors and using the longest paths within to determine the starting positions. Subsequently, human annotators are asked to annotate the room count and the respective sides while navigating. The instruction follows the template: *"walk along the corridor and turn into the ith room on your left/right"*. An example case is shown in Figure 2c. Paths that share identical initial poses, yet differ in numerical and directional values, are treated as negative data. These are included in the dataset to support the evaluation of numerical comprehension.

Landmark Recognition category: This category requires taking a path associated with a specific landmark. To assess landmark recognition capabilities, it is necessary to provide instance-level descriptions in the instructions. We begin by identifying potential navigable objects using semantic annotations. We leverage GPT-4-vision (Achiam et al., 2023) for precise object category identification and instance-level description generation given the view orientated towards the object. We then construct paths that meet specific criteria regarding curvature and proximity to the object's center, as shown in Figure 2d. The resulting instructions encompass actions such as *"walk past + modifier + object"*. We include manual checks and modifications to ensure the visibility of target landmarks from starting viewpoints. The object center is included in the dataset as supplementary information for evaluation.

Region Recognition category: Finally, the region recognition category is narrowed down to 'go into' and 'exit' actions due to the potential ambiguity in 'go through' instructions. Unlike specific endpoint-related data, region-related data pertains to a set of points associated with the concept. For example, in Figure 2e, given a starting point and the instruction *"go into the bedroom"*, we record all points inside adjacent bedrooms as correct responses. For *"exit the dining area"*, all areas outside the current room are marked as valid positions.

4 Experiment

We conduct a comprehensive evaluation of various existing VLN models across the five main categories in our NavNuances dataset.

4.1 Baselines

In this study, we examine baseline models categorized by input modalities, action spaces, memory representations, and supervision approaches. Input modalities range from front-view RGB images (*e.g.,* Seq2Seq model (Anderson et al., 2018)) and panorama images (*e.g.,* VLN-BERT (Hong et al., 2021)) to textual descriptions of panorama views (*e.g.,* NavGPT (Zhou et al., 2023)). Models differ in their action space, utilizing viewpoint selection (*e.g.,* ScaleVLN (Wang et al., 2023)), predefined rule-based actions (*e.g.,* Seq2Seq (Anderson et al., 2018)), or a combination thereof. Memory representation varies among models, employing hidden states (*e.g.,* CLIP-ViL (Shen et al., 2021)), past visual inputs (*e.g.,* HAMT (Chen et al., 2021)), topological (*e.g.,* DUET (Chen et al., 2022)) or metric maps (*e.g.,* BEVBERT (An et al., 2023)), or interactive chat histories (*e.g.,* NavGPT (Zhou et al., 2023)). Except for differences in the pre-training data sources, all the supervised models are fine-tuned on the R2R dataset (Anderson et al., 2018). More details are available in Appendix A.

We introduce **NavGPT4v**, an enhancement of the text-based NavGPT (Zhou et al., 2023) model with visual inputs, integrating actual image views with GPT-4-vision (Achiam et al., 2023). We modify the initial prompt in NavGPT to highlight the presence of visual resources and their relevance to a particular direction, as illustrated in Appendix F. This development targets incorporating direct visual information to capture essential details that pre-captioning might miss.

4.2 Evaluation Protocols

In this section, we introduce the evaluation protocols for our Vision-Language Navigation (VLN) evaluation set. These protocols are designed to precisely measure the performance of navigation models based on detailed success criteria for different categories of atomic instructions.

For categories **Landmark Recognition**, **Numerical Comprehension**, and **Vertical Movement**, the evaluations follow the distance-related protocols. The criteria differ slightly depending on the nature of the movement. For instance, in the vertical

	Method	**Experimental setting**			**Evaluation Results**					**R2R unseen**		
		Vision	Action	History	DC	NU	LR	RR	VM	SR	nDTW	SPL
Supervised	Random	None	viewpoint	None	36.79	7.69	30.22	57.45	11.76	15.88	24.21	14.04
	Seq2Seq	front-view	rule-based	hidden state	75.30	21.79	21.89	53.09	25.88	21.46	25.04	18.50
	CLIP-ViL	front-view	rule-based	hidden state	77.20	29.49	39.85	74.18	69.41	52.15	47.75	47.64
	VLN-BERT	panorama	viewpoint	hidden state	72.02	29.49	34.31	80.36	75.29	62.75	65.49	56.89
	HAMT	panorama	viewpoint	past views	79.62	28.21	34.74	77.81	68.82	63.22	66.37	57.70
	DUET	panorama	viewpoint	topo. map	64.76	26.92	35.62	77.45	76.47	71.52	67.78	60.85
	BEVBERT	panorama	viewpoint	topo./metric map	63.21	24.35	30.22	80.36	84.12	75.18	69.40	63.68
	ScaleVLN	panorama	viewpoint	topo. map	72.88	26.92	27.59	84.73	84.71	80.97	74.76	70.33
0-shot	NavGPT3.5	pano. text	viewpoint	text history	81.87	20.51	58.54	39.63	7.06	12.67	40.82	11.45
	NavGPT4	pano. text	viewpoint	text history	91.87	34.78	54.83	67.61	11.36	34.78	47.53	31.64
	NavGPT4v	panorama	viewpoint	text history	92.68	39.13	62.87	56.25	13.64	41.30	54.78	36.84
	Human	front-view	turn/vpt.	memory	95.83	89.13	89.44	89.89	94.42	-	-	-

Table 1: **Main Results** for baselines evaluated on five main categories of NavNuances dataset, *i.e.,* Direction Change (DC), Vertical Movement (VM), Landmark Recognition (LR), Region Recognition (RR) and Numerical Comprehension (NU). We also post the reproduced results on the standard R2R unseen dataset using three principal metrics: Success Rate (SR), normalized Dynamic Time Warping (nDTW) and Success rate weighted by normalized inverse Path Length (SPL)

movement category, success is defined by a 3-meter radius to a specified endpoint. For instructions involving more localized navigation, such as walking towards a landmark, the metric focuses more on the reduction in distance to the landmark. Further details can be found in Appendix B.

Region Recognition category is more related to inclusion-related protocol. Distance metrics are inadequate due to the lack of a precise endpoint. Success in this category is defined by the model's ability to stop within a designated region, determined by its boundaries.

For **Direction Change** category, we evaluate the model's compliance with directional instructions. The protocol involves dividing the area around the starting point into sectors to assess the accuracy of the model's initial movement direction in response to the given instruction.

4.3 Main Results

We report the performance evaluated on NAVNUANCES as well as the reproduced results on the validation unseen split of the R2R dataset (Anderson et al., 2018) in Table 1. We assess NavGPT4 and NavGPT4v using a random subset of around 130 samples, ensuring replicability of the officially reported NavGPT performance without incurring significant API costs.

Reflecting on the advancements in the standard R2R dataset, it appears that improved layout and spatial understanding underpin the progress of VLN models. This is evident from the results in **vertical movement (VM)** and **region recognition (RR)** tasks on our dataset. This correlation is probably due to the statistics of the R2R unseen split. We find that more than 35% of the instructions necessitate navigation through stairs, and the majority involve concepts related to rooms. The correlation is observed consistently across different models. For instance, CLIP-ViL's leap in performance on the R2R unseen split compared to the prior model Seq2Seq (30.69% absolute increase in success rate) correlates with significant gains in vertical movement (from 25.88% to 69.41%) and region recognition (from 53.09% to 74.18%). And the low performance of zero-shot methods on R2R also follows the lower success rates in these tasks.

Despite advancements, there is a noticeable stagnation in models' **numerical comprehension (NU)**, likely due to the sparse numerical data in existing datasets and the non-essential nature of numerical comprehension for task completion. Compared to traditional methods, LLM-enhanced models show slightly better performance but still fall significantly short of human capabilities. These findings highlight that numerical comprehension presents a substantial challenge across various model types, the inference ability w.r.t. numerical values require further improvement.

In examining **directional changes (DC)** within supervised methods, models with explicit directional commands (such as the methods with rule-based action space, Seq2Seq, and CLIP-ViL) can easily reach or even outperform those employing viewpoint selection techniques, suggesting the importance of clear action spaces for effective turning choices. This is further supported by the superior performance of zero-shot agents, as each observation in the zero-shot agent's prompt includes a clear description of its orientation.

In **landmark recognition (LR)**, zero-shot agents outperform models supervised on R2R data, which shows a contrast to their performance on the standard R2R dataset. This indicates that extensive knowledge of large pre-trained models can overcome the constraints inherent in small-scale supervised training. A notable comparison between NavGPT4 and our NavGPT4v reveals that conditioning observations on specific instructions leads to more accurate landmark recognition, attributed to the richness of visual content beyond mere captions. In addition, the high performance of the random agent suggests the choices within a fixed radius are limited. This highlights the limitations of supervised agents. Their performance, while comparable to the random agent, falls short of a true understanding of individual object instances.

Compared to traditional supervised methods, the performances of LMMs (*e.g.,* NavGPT4v) on **Region Recognition (RR)** and **Vertical Movement (LM)** are significantly lower. This trend is more clear from Figure 6 of the Appendix. The diminished performance in vertical movement may be attributed to a lack of contextual understanding, as discussed in recent multimodal benchmark CODIS (Luo et al., 2024). Regarding the surprisingly low performance on region recognition, this issue seems to stem from the LMM's imprecise boundary judgment; the model tends to prematurely halt while merely observing the target region. Further discussion is available in Appendix C.

4.4 Additional Experiments

Does the agent understand numerical values?

In this additional experiment, we aim to further study the numerical comprehension capabilities of models. Despite observing an overall low performance in this category, these models do show some improvements over a baseline random walk agent. However, the concept of "numerical" functions as a special modifier, which always links to a specific object or region with a certain spatial relation. This association leads to an overestimation of the capability of numerical comprehension. Thus, we introduce two additional random agents to isolate these factors: The first agent simulates a basic understanding of spatial layouts (*Agent 1**), enabling the agent to select a room to enter. The second agent embodies directional intelligence (*Agent 2**), allowing the agent to choose a room on the specific side, such as entering a room on the left.

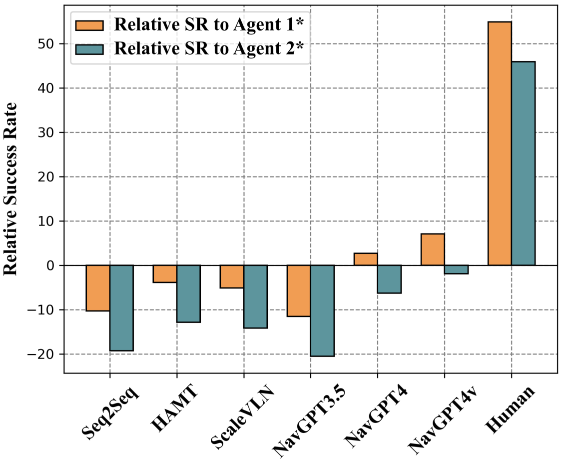


Figure 3: Success rate relative to two additional random agents in the numerical comprehension category. *Agent 1** is the random agent that knows the concept of entering the room in the corridor. *Agent 2** is the random agent which also has directional awareness. The success rate of *Agent 1** and *Agent 2** are 32.06% and 41.03%.

As shown in Figure 3, for some of the supervised models such as HAMT and ScaleVLN, the performance is comparable (relative success rate approach zero) to that of the *Agent 1** but significantly lagged behind the *Agent 2**. Zero-shot agents enhanced by GPT-4 can surpass but still have much lower performance than humans (50% success rate below). This discrepancy highlights a critical gap in current models: while they may grasp basic layout concepts to a degree, their understanding of more complex scenarios involving both numerical values and directional cues is markedly deficient. The results, as illustrated in Figure 3, highlight the need for advanced models that integrate numerical, layout, and directional understanding.

Can the model understand specific landmarks and the spatial relation with them?

In the Landmark recognition category, we further assess the models' performance in its two distinct subsets: navigating towards a specific object and navigating past an object. The former primarily tests the models' visual grounding capabilities, while the latter introduces an additional layer of complexity by requiring an understanding of spatial relationships based on sequential observations.

We evaluate these subsets against three categories of baseline models: supervised front-view models, supervised panorama-view models, and

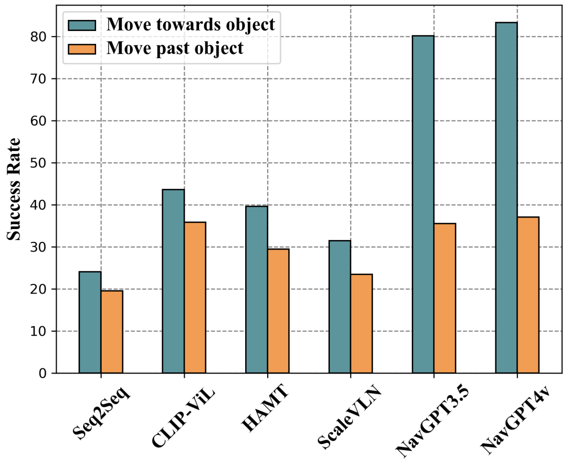


Figure 4: Results of two subsets of Landmark recognition category in NavNuances dataset. The significant gap of the 'moving towards' subset comes from large pre-trained vision models since NavGPT3.5

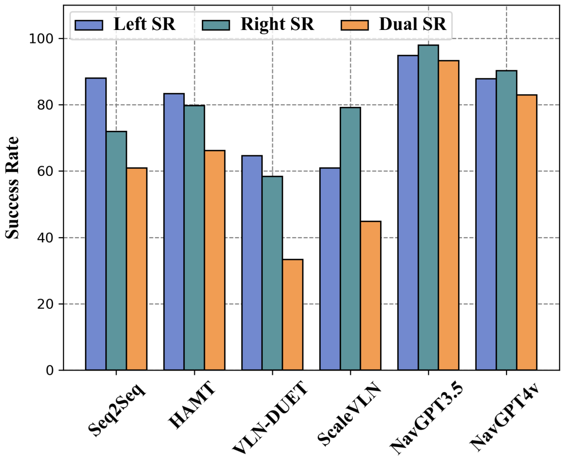


Figure 5: Results for the left/right turn subsets within the direction change category of the NavNuances dataset, with Dual SR indicating the success rates for both right and left turns from a specific starting view

zero-shot models enhanced with Large Multimodal models (LMMs). As illustrated in Figure 4, the simplest Seq2Seq model augmented with CLIP features (from the CLIP-ViL model) outperforms more recent approaches like HAMT and ScaleVLN on both tasks. This indicates that even when utilizing only frontal views, robust visual features can effectively align with object-centric instructions.

Advanced models like BLIP2 (Li et al., 2023) (in NavGPT 3.5 and 4) and GPT-4-vision (in NavGPT4v) show marked improvements in navigating towards objects. However, they still struggle with the spatial relation aspect, particularly in navigating past objects. Analysis of error cases reveals inconsistent decision-making. For instance, the models correctly interpret moving from an object's front to back as having navigated past it in some cases. However, the models sometimes assume they have passed an object merely because it is beside them, contradicting the commonsense of *walking past*. This inconsistency highlights the need for future models to better align with nuanced human commonsense in spatial reasoning.

Any bias in the choice of turning direction?

To assess if vision-language navigation models exhibit a turning direction preference, we analyze their performance on *turn left* and *turn right* commands using models like HAMT and NavGPT. Our dataset, containing paired *turn right* and *turn left* instructions for each starting viewpoint, facilitated this analysis. We introduced a "Dual Success Rate" (Dual SR) metric to measure a model's accuracy in executing both directions from the same point.

Our findings, as depicted in Figure 5, indicate a directional bias in some models. For instance, ScaleVLN shows a notable preference for *turn right* instructions, with an 18.23% higher success rate for right turns compared to left turns. There are also general selective biases across all supervised models, as evidenced by their heavily reduced performance on the Dual SR metric. One possible reason for this bias is that there might be the models' potential preference for specific visual cues over the given navigational commands. Conversely, zero-shot models like NavGPT3.5 and NavGPT4v show minimal bias and perform comparably to humans in handling both left and right turns.

5 Conclusion

In this study, we establish a systematic framework to diagnose deficiencies in the capabilities of Vision-Language Navigation (VLN) models at the atomic instruction level. Our experiment results on NavNuances across diverse models clearly uncover the limitations of specific models and reveal common issues, which highlight ongoing challenges in the VLN task. In addition, our investigation into a modified zero-shot agent enhanced by GPT-4-vision provides empirical evidence that a direct alignment between vision and instructions significantly enhances landmark recognition performance. This insight underscores the potential for leveraging advanced large multimodal models in improving VLN systems.

Limitations

Despite the data involved in our study are sufficiently representative to support the insights provided by our initial findings, the constraints imposed by the static discrete environments (Matterport3D) lead to several limitations. Since we are not able to edit the environment such as adding or removing objects, we are restricted to generating data from existing layouts. This limits the data diversity for some instruction categories. For instance, in the numerical comprehension category, due to a lack of identical object categories within single regions, we are unable to encompass numerical comprehension data in the object level, such as *“move close to the [i]-th apple on your right"*. Additionally, because we cannot rearrange object attributes and positions, it is difficult to achieve a detailed attribute-level data design in the landmark recognition category.

In addition, this study focuses exclusively on atomic-level capabilities, which do not encompass the full range of capabilities of VLN agents such as error correction for executing long instructions. Understanding sequences of multiple actions within long instructions is also a crucial aspect of the VLN task. Evaluating from this aspect is challenging but represents a promising direction for future research.

In this work, we leverage CFG as the basis of the problem decomposition and construct a diagnostic dataset based on it. Our semi-automatic approach for CFG construction is well-suited for designing specialized datasets in fields like law or finance. However, for more complex tasks, relying on manual corrections may be inefficient and challenging in ensuring comprehensive coverage of concepts. An improvement would be the development of a fully automatic induction method, leveraging the extensive world knowledge encapsulated in large language models, to potentially replace the current semi-automatic method.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Dong An, Yuankai Qi, Yangguang Li, Yan Huang, Liang Wang, Tieniu Tan, and Jing Shao. 2023. Bevbert: Multimodal map pre-training for language-guided navigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2737–2748.

Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton Van Den Hengel. 2018. 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.

Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niebner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. 2017. Matterport3d: Learning from rgb-d data in indoor environments. In *2017 International Conference on 3D Vision (3DV)*, pages 667–676. IEEE.

Peihao Chen, Xinyu Sun, Hongyan Zhi, Runhao Zeng, Thomas H Li, Gaowen Liu, Mingkui Tan, and Chuang Gan. 2023. A2nav: Action-aware zero-shot robot navigation by exploiting vision-and-language ability of foundation models. *arXiv preprint arXiv:2308.07997*.

Shizhe Chen, Pierre-Louis Guhur, Cordelia Schmid, and Ivan Laptev. 2021. History aware multimodal transformer for vision-and-language navigation. *Advances in neural information processing systems*, 34:5834–5847.

Shizhe Chen, Pierre-Louis Guhur, Makarand Tapaswi, Cordelia Schmid, and Ivan Laptev. 2022. Think global, act local: Dual-scale graph transformer for vision-and-language navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16537–16547.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee.

Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, Yunsheng Wu, and Rongrong Ji. 2024. Mme: A comprehensive evaluation benchmark for multimodal large language models. *Preprint*, arXiv:2306.13394.

Yicong Hong, Qi Wu, Yuankai Qi, Cristian Rodriguez-Opazo, and Stephen Gould. 2021. Vln bert: A recurrent vision-and-language bert for navigation. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pages 1643–1653.

John E Hopcroft, Rajeev Motwani, and Jeffrey D Ullman. 2001. Introduction to automata theory, languages, and computation. *Acm Sigact News*, 32(1):60–65.

Jacob Krantz, Erik Wijmans, Arjun Majundar, Dhruv Batra, and Stefan Lee. 2020. Beyond the nav-graph: Vision and language navigation in continuous environments. In *European Conference on Computer Vision (ECCV)*.

Alexander Ku, Peter Anderson, Roma Patel, Eugene Ie, and Jason Baldridge. 2020. Room-across-room: Multilingual vision-and-language navigation with dense spatiotemporal grounding. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4392–4412.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR.

Bingqian Lin, Yunshuang Nie, Ziming Wei, Jiaqi Chen, Shikui Ma, Jianhua Han, Hang Xu, Xiaojun Chang, and Xiaodan Liang. 2024. Navcot: Boosting llm-based vision-and-language navigation via learning disentangled reasoning. *arXiv preprint arXiv:2403.07376*.

Yuxing Long, Xiaoqi Li, Wenzhe Cai, and Hao Dong. 2023. Discuss before moving: Visual language navigation via multi-expert discussions. *arXiv preprint arXiv:2309.11382*.

Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. 2023. Mathvista: Evaluating math reasoning in visual contexts with gpt-4v, bard, and other large multimodal models. *arXiv e-prints*, pages arXiv–2310.

Fuwen Luo, Chi Chen, Zihao Wan, Zhaolu Kang, Qidong Yan, Yingjie Li, Xiaolong Wang, Siyu Wang, Ziyue Wang, Xiaoyue Mi, et al. 2024. Codis: Benchmarking context-dependent visual comprehension for multimodal large language models. *arXiv preprint arXiv:2402.13607*.

Aishwarya Padmakumar, Jesse Thomason, Ayush Shrivastava, Patrick Lange, Anjali Narayan-Chen, Spandana Gella, Robinson Piramuthu, Gokhan Tur, and Dilek Hakkani-Tur. 2022. Teach: Task-driven embodied agents that chat. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 2017–2025.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR.

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

Alessandro Stolfo, Zhijing Jin, Kumar Shridhar, Bernhard Schoelkopf, and Mrinmaya Sachan. 2023. A causal framework to quantify the robustness of mathematical reasoning with language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 545–561.

Andrew Szot, Alexander Clegg, Eric Undersander, Erik Wijmans, Yili Zhao, John Turner, Noah Maestre, Mustafa Mukadam, Devendra Singh Chaplot, Oleksandr Maksymets, et al. 2021. Habitat 2.0: Training home assistants to rearrange their habitat. *Advances in neural information processing systems*, 34:251–266.

Jesse Thomason, Michael Murray, Maya Cakmak, and Luke Zettlemoyer. 2020. Vision-and-dialog navigation. In *Conference on Robot Learning*, pages 394–406. PMLR.

Naoki Wake, Atsushi Kanehira, Kazuhiro Sasabuchi, Jun Takamatsu, and Katsushi Ikeuchi. 2023. Gpt-4v (ision) for robotics: Multimodal task planning from human demonstration. *arXiv preprint arXiv:2311.12015*.

Xin Wang, Qiuyuan Huang, Asli Celikyilmaz, Jianfeng Gao, Dinghan Shen, Yuan-Fang Wang, William Yang Wang, and Lei Zhang. 2019. Reinforced cross-modal matching and self-supervised imitation learning for vision-language navigation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6629–6638.

Zun Wang, Jialu Li, Yicong Hong, Yi Wang, Qi Wu, Mohit Bansal, Stephen Gould, Hao Tan, and Yu Qiao. 2023. Scaling data generation in vision-and-language navigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12009–12020.

Gengze Zhou, Yicong Hong, and Qi Wu. 2023. Navgpt: Explicit reasoning in vision-and-language navigation with large language models. *arXiv preprint arXiv:2305.16986*.

A Details of baseline models

We mainly study the following models:

1. Random Agent: This model, serving as a rudimentary baseline in VLN tasks, executes five arbitrary movements within the navigation graph without relying on navigational instructions or environmental observations.

2. Seq2Seq (Anderson et al., 2018) / CLIP-ViL-VLN (Shen et al., 2021): These models process only the frontal RGB visual input. The visual features for Seq2Seq and CLIP-ViL-VLN are derived from pre-trained ImageNet (Deng et al., 2009) and CLIP vision encoders (Radford et al., 2021), respectively. Navigation decisions are stored in LSTM's hidden states, with the action space confined to predefined movements such as forward, left, right, up, and down.

3. VLN-BERT (Hong et al., 2021): Distinguished by its use of panoramic visuals at each navigation point, this model alters the action space to the selection of subsequent navigation points. It utilizes the first special token in the Transformer-based model to represent the history state.

4. HAMT (Chen et al., 2021): Similar to VLN-BERT in terms of visual input and action space, this model differentiates itself by employing the features of historical observations to represent navigational memory.

5. DUET (Chen et al., 2022) / ScaleVLN (Wang et al., 2023): Both models utilize panoramic visuals and navigate by choosing subsequent points. The historical memory is encapsulated within a topological graph. ScaleVLN further enhances its capability by incorporating a vast collection of automatically gathered VLN data.

6. BEVBERT (An et al., 2023): Building upon the foundation laid by VLN-DUET, BEVBERT introduces metric maps as an additional observational and memory component, aiming for a more enriched navigational context.

7. NavGPT (Zhou et al., 2023) / NavCoT (Lin et al., 2024): These zero-shot large language models (LLMs) encapsulate navigational history within a dialogue history, offering a novel approach to VLN tasks. Observations are converted into descriptions by a pre-trained captioning model, treating the VLN task as a text-based navigation challenge.

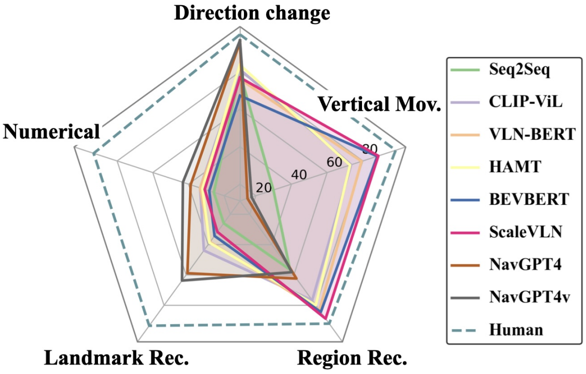


Figure 6: The success rate of models evaluated on five main categories of NavNuances dataset. Human performance is denoted by the green dashed line.

8. NavGPT4v: We enhance the text-based NavGPT model (Zhou et al., 2023) by visual input, NavGPT4v incorporates actual image views alongside a Large Multimodal Model (LMM) - GPT-4-vision (Achiam et al., 2023) with modified prompts. This addition aims to address the limitations of pre-captioning observations, which may overlook critical details in the views due to the generic nature of captions.

Through the lens of these diverse models, our study aims to shed light on the multifaceted nature of VLN tasks and the inherent capabilities and limitations of each approach.

B Detailed Evaluation metrics

This section presents the evaluation metrics for each category within the NavNuances dataset, adhering to the overarching protocols delineated in Section 4.2.

B.1 Direction Change category

In Direction Change category, we design evaluation metrics based on the direction protocol, focusing exclusively on the initial sub-path—defined as the trajectory connecting the first and second navigation points. The categorization of directional changes is as follows: if the sub-path's orientation relative to the starting point falls within a 120-degree arc to the left, it is classified as a **turn left**;

similarly, a 120-degree arc to the right is classified as a **turn right**, and a 120-degree arc to the rear is classified as a **turn around**. An agent's success is determined by the accuracy of its directional change in response to the given instruction.

B.2 Landmark Recognition category

For the Landmark Recognition category, metrics are based on a distance protocol, utilizing object center coordinates for evaluation:

walking towards a specific landmark: Success is determined if the agent's final position is nearer to the landmark's center coordinate compared to its starting position, with the landmark being visible and at a distance from the starting point.

walking past a specific landmark: the agent's decision is considered as success if the object center can projected within the line segment defined by start and end position, and the end position is within three meters of the landmark's center.

B.3 Numerical Comprehension category

This category employs a distance protocol, with a unique consideration for path similarity. Given that paths within the same hallway and identical starting points are indexed by the same set number, success criteria include:

1. The agent's final position must be within 3 meters in geometric distance of the endpoint.

2. The normalized Dynamic Time Warping (nDTW) metric, which assesses path similarity, must indicate that the agent's path more closely aligns with the ground truth path than with any other paths in the set (nDTW larger than other paths in the same set).

B.4 Vertical Movement category

Adhering to a distance protocol, an agent is deemed successful in the Vertical Movement category if it stops within a three-meter geometric radius of the annotated endpoint, emphasizing vertical navigation accuracy.

B.5 Region Recognition category

The Region Recognition category utilizes the inclusion evaluation protocol:

entering a region: Success is achieved if the agent stops within a region marked with the same room category as specified in the instruction and proximate to the starting region.

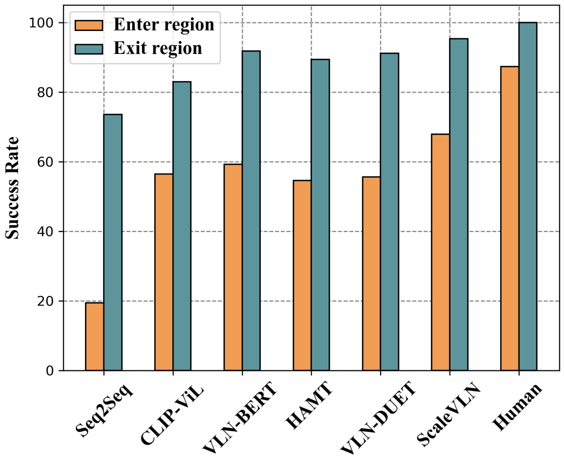


Figure 7: Results for success rate of subsets in room recognition category.

exiting a region: Success is determined if the agent's stopping point lies outside the boundaries of the starting region.

C Whether the model can understand room category very well?

In our primary results, we focused on evaluating various actions related to navigating through regions within a dataset to gauge the models' proficiency with region-associated tasks. This approach provides a general overview of a model's capability in handling layout concepts. However, the tasks of entering a region and exiting a region present unique challenges. Specifically, entering a region demands a more nuanced understanding of the region's category. For example, when given the instruction "go into the dining room" from a location adjacent to multiple rooms, the agent must discern the characteristics that define a dining room to navigate successfully. Conversely, leaving a region only involves recognizing the concept of a region, without necessitating an in-depth categorization.

To delve deeper into this distinction, we evaluate these two subsets from the data of region recognition category: one is related to entering a region, and the other is related to exiting a region. Zero-shot agents, which typically perform poorly and lack a clear understanding of region boundaries, often optimistically halt upon merely observing the room from just outside the boundary. The error cases can be found in Figure 13. In this subsection, we only discuss the results of supervised methods. As shown in Figure 7, starting from the VLN-BERT model onwards, the performance on tasks involving ’exit a region’ has remained consistently high, indicating that subsequent models have effectively grasped the concept of a region. On the other hand, the ability to understand and categorize different types of regions appears to have progressively improved with each new model iteration.

However, when comparing these results to human performance, a significant discrepancy becomes evident. The gap in understanding and categorizing regions between humans and the current state-of-the-art (SOTA) models is approximately 21.59%. This gap highlights the ongoing challenge in the field of Vision-Language Navigation (VLN) to develop models that can match human-level comprehension of spatial and categorical concepts within navigational tasks.

D Dataset statistics and examples

Our NavNuances dataset comprises 579 instances of Direction Change, 170 of Vertical Movement (with 44 having a pair of staircases in opposing directions at the initial viewpoint), 78 of Numerical Comprehension, 275 of Region Recognition, and 685 of Landmark Recognition.

The statistics for subsets in each category:

Direction Change: there are 192 instances for “turn right", 192 instances for “turn left" and 195 instances for “turn around".

Landmark Recognition: there are 353 instances for “walk towards a landmark", 332 instances for “walk past a landmark".

Numerical Comprehension: there are 31 instances for “first room", 24 instances for “second room", 13 instances for “third room", 6 instances for “fourth room", 2 instances for “fifth room", and 2 instances for “sixth room".

Region Recognition: there are 105 instances for “go into a room", 170 instances for “exit a room".

Vertical Movement: there are 87 instances for “go upstairs", 83 instances for “go downstairs".

walk past the **wooden chess board on table**.

walk past the **stainless steel double-door refrigerator**.

walk towards the **white flower in round vase**.

walk towards the **gray front-load washing machine**.

Figure 8: Landmark Recognition data samples

Walk along the corridor and turn into the **first/second/third room on your right**.

Figure 9: Numerical Comprehension data samples

Figure 10: Vertical Movement data samples

Exit the **living room**.

Go into the **dining room**.

Figure 11: Region Recognition data samples

Initial view

Head to the right and continue walking straight.

Make a **left turn** and continue walking straight.

Spin around and continue walking straight.

Figure 12: Direction Change data samples

E The context-free grammar in concept for VLN instruction

For the initial set of production rules we refer to our observations and also definitions in prior works such as A^2Nav (Chen et al., 2023; Long et al., 2023). Then we interact with GPT-4 (Achiam et al., 2023), we input the CFG definitions with long instructions, and the GPT-4 with return the parsing results. We find GPT-4 can leverage CFG very well, and automatically detect which instruction segment cannot be parsed by the CFG. Then we utilize this information to update our CFG. This iterative updating will last for about ten rounds.

List 2: Context-free grammar

```
$S \rightarrow Vp$
$Vp \rightarrow$ ActionT
    $|$ActionS
    $|$ActionO $+$ Landmark
    $|$ActionR $+$ Region
    $|Vp + Vp$
    $|Vp + Ir$
$Ir \rightarrow$ ($\textit{sentence describing the state of observation, not action}$)
$Numerical \rightarrow$ first$|$second$|$third$|$fourth$|$fifth$|\ldots$
$Room \rightarrow$ room$|$kitchen$|$bathroom$|\ldots$
$Direction \rightarrow$ left$|$right
$Object \rightarrow$ bed$|$table$|$chair$|\ldots$
$Attribute \rightarrow$ red$|$yellow$|\ldots$
$Modifier \rightarrow$
    $Object +$ "is on the" $+ Direction|Attribute|Numerical|Direction|Modifier + Modifier|\epsilon$
$Landmark \rightarrow Modifier + Object$
$Region \rightarrow Modifier + Room$
$ActionT \rightarrow$ "turn" $+ Direction|$"turn around"
$ActionO \rightarrow$ "walk towards"("wait at")$|$"walk past"$|$"walk past from" $+ Direction$
$ActionR \rightarrow$ "go into"("wait at")$|$"exit"$|$"walk through"
$ActionS \rightarrow$ "go upstairs"$|$"go downstairs"
```

F Prompts used

Listing 1: NavGPT4v prompts (extend from NavGPT). The actual implementation of api calls will split the template into several parts, vision related inputs will follow the API standard in GPT-4-vision to first transfer the image to base64 encoded string and then add special tag.

```
As an intelligent embodied agent, you will navigate an indoor environment to reach a
 target viewpoint based on a given instruction, performing the Vision and Language
Navigation (VLN) task. You'll move among static positions within a pre-defined graph
, aiming for minimal steps.

You will receive a trajectory instruction at the start and will have access to step
history (your Thought, Action, Action Input and Obeservation after the Begin! sign)
and current viewpoint observation (including the photos captured around, breif scene
 descriptions, objects, and navigable directions/distances within 3 meters). Each
photo has a blue index on its topleft corner. The correspondence between the photo
index and the viewing direction is as follows: photo 0 is Front view; photo 1 is
Front Right view; photo 2 is Right view; photo 3 is Rear Right view; photo 4 is Rear
 view; photo 5 is Rear Left view; photo 6 is Left view; photo 7 is Front Left view.
Scene descriptions and object descriptions are just for reference, might be
incomplete.

Orientations range from -180 to 180 degrees, with 0 being forward, right 90
rightward, right/left 180 backward, and left 90 leftward.

Explore the environment while avoiding revisiting viewpoints by comparing current
and previously visited IDs. Reach within 3 meters of the instructed destination, and
 if it's visible but no objects are detected, move closer.

At each step, determine if you've reached the destination.
If yes, stop and output 'Final Answer: Finished!'.
If not, continue by considering your location and the next viewpoint based on the
instruction, using the action_maker tool.
Show your reasoning in the Thought section.

Follow the given format and use provided tools.
{tool_descriptions}
Do not fabricate nonexistent viewpoint IDs.

----
Starting below, you should follow this format:

Instruction: the instruction describing the whole trajectory
Initial Observation: the initial observation of the environment
Thought: you should always think about what to do next and why
Action: the action to take, must be one of the tools [{tool_names}]
Action Input: "Viewpoint ID"
Observation: the result of the action
... (this Thought/Action/Action Input/Observation can repeat N times)
Thought: I have reached the destination, I can stop.
Final Answer: Finished!
----

Begin!

Instruction: {action_plan}
Initial Observation: {visual_observations}
Thought: I should start navigation according to the instruction, {agent_scratchpad}
```

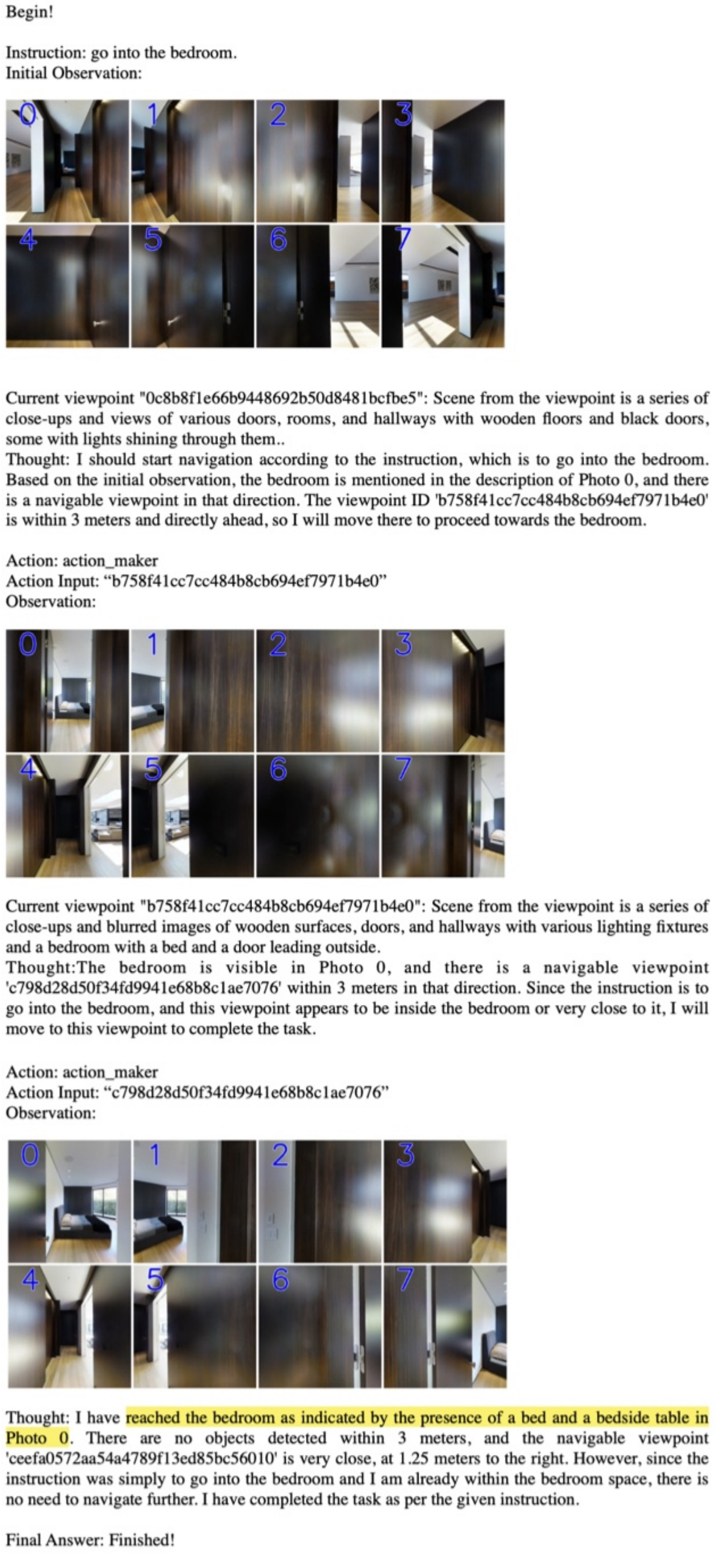
Listing 2: Prompts for landmark description

```
Here is a picture with probably some objects in the middle. Please breifly describe
the most identifiable object which is close to you in under ten words with
elementary-level vocabularies. The object should be large and unique in the given
picture, for instance a 'black round table'. The object category cannot be the
mirror, door, floor, ceiling, wall, windows, light switch, control panel or any
small objects. If you cannot find any valid object in the picture, return the '
object not found'. If in the center of the image is a wall or the doorframe, please
also return 'object not found'.

Picture: {image}
```

Listing 3: Prompts for linguistic enrichment

```
Here is a simple instruction, please rephrase it without changing its content.
Please also keep the rephrased instruction natural. For turning action about turn
around, try not modify this action.

Instruction: {instr}.
```

G Case studies



Begin!

Instruction: go into the bedroom.
Initial Observation:

Current viewpoint "0c8b8f1e66b9448692b50d8481bcfbe5": Scene from the viewpoint is a series of close-ups and views of various doors, rooms, and hallways with wooden floors and black doors, some with lights shining through them..
Thought: I should start navigation according to the instruction, which is to go into the bedroom. Based on the initial observation, the bedroom is mentioned in the description of Photo 0, and there is a navigable viewpoint in that direction. The viewpoint ID 'b758f41cc7cc484b8cb694ef7971b4e0' is within 3 meters and directly ahead, so I will move there to proceed towards the bedroom.

Action: action_maker
Action Input: “b758f41cc7cc484b8cb694ef7971b4e0”
Observation:

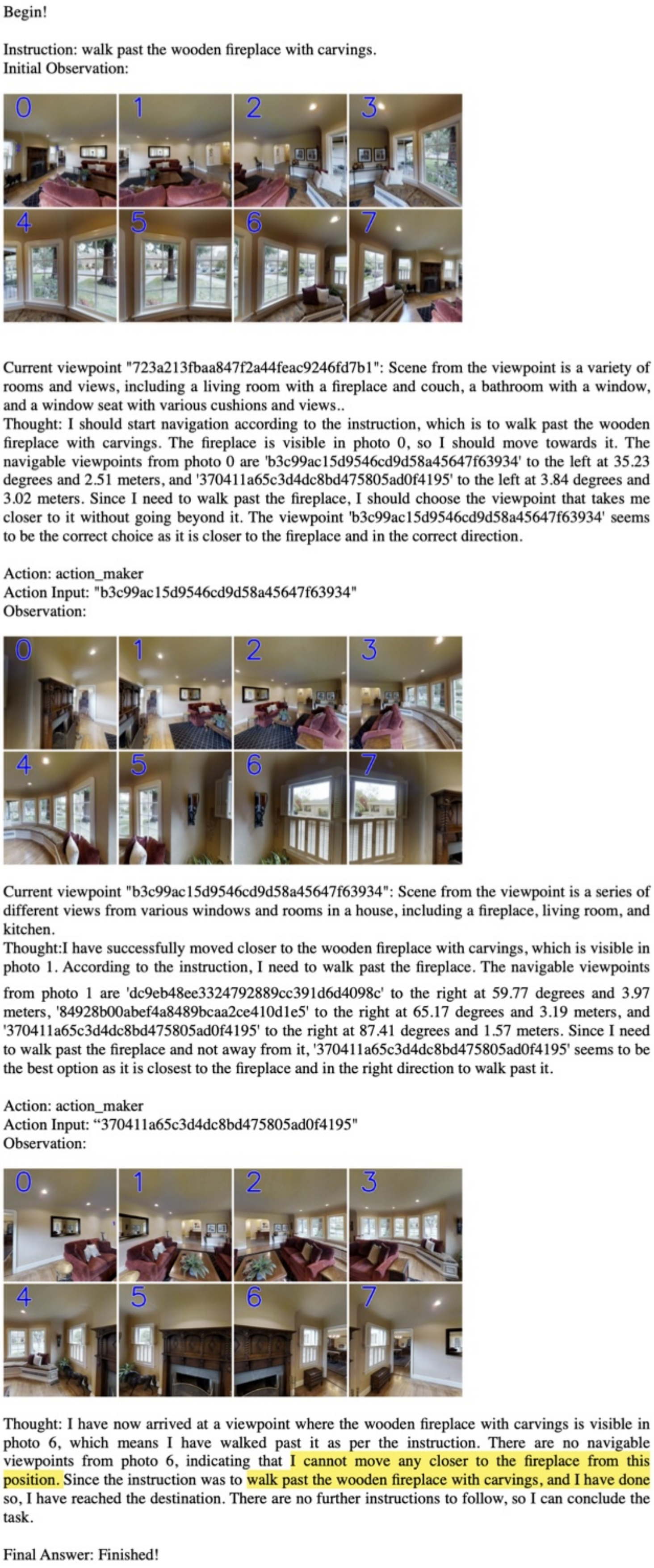
Current viewpoint "b758f41cc7cc484b8cb694ef7971b4e0": Scene from the viewpoint is a series of close-ups and blurred images of wooden surfaces, doors, and hallways with various lighting fixtures and a bedroom with a bed and a door leading outside.
Thought:The bedroom is visible in Photo 0, and there is a navigable viewpoint 'c798d28d50f34fd9941e68b8c1ae7076' within 3 meters in that direction. Since the instruction is to go into the bedroom, and this viewpoint appears to be inside the bedroom or very close to it, I will move to this viewpoint to complete the task.

Action: action_maker
Action Input: “c798d28d50f34fd9941e68b8c1ae7076”
Observation:

Thought: I have reached the bedroom as indicated by the presence of a bed and a bedside table in Photo 0. There are no objects detected within 3 meters, and the navigable viewpoint 'ceefa0572aa54a4789f13ed85bc56010' is very close, at 1.25 meters to the right. However, since the instruction was simply to go into the bedroom and I am already within the bedroom space, there is no need to navigate further. I have completed the task as per the given instruction.

Final Answer: Finished!

Figure 13: Failure case of “entering" subset of Region Recognition category: NavGPT4v stop before entering the target region



Begin!

Instruction: walk past the wooden fireplace with carvings.
Initial Observation:

Current viewpoint "723a213fbaa847f2a44feac9246fd7b1": Scene from the viewpoint is a variety of rooms and views, including a living room with a fireplace and couch, a bathroom with a window, and a window seat with various cushions and views..
Thought: I should start navigation according to the instruction, which is to walk past the wooden fireplace with carvings. The fireplace is visible in photo 0, so I should move towards it. The navigable viewpoints from photo 0 are 'b3c99ac15d9546cd9d58a45647f63934' to the left at 35.23 degrees and 2.51 meters, and '370411a65c3d4dc8bd475805ad0f4195' to the left at 3.84 degrees and 3.02 meters. Since I need to walk past the fireplace, I should choose the viewpoint that takes me closer to it without going beyond it. The viewpoint 'b3c99ac15d9546cd9d58a45647f63934' seems to be the correct choice as it is closer to the fireplace and in the correct direction.

Action: action_maker
Action Input: "b3c99ac15d9546cd9d58a45647f63934"
Observation:

Current viewpoint "b3c99ac15d9546cd9d58a45647f63934": Scene from the viewpoint is a series of different views from various windows and rooms in a house, including a fireplace, living room, and kitchen.
Thought:I have successfully moved closer to the wooden fireplace with carvings, which is visible in photo 1. According to the instruction, I need to walk past the fireplace. The navigable viewpoints from photo 1 are 'dc9eb48ee3324792889cc391d6d4098c' to the right at 59.77 degrees and 3.97 meters, '84928b00abef4a8489bcaa2ce410d1e5' to the right at 65.17 degrees and 3.19 meters, and '370411a65c3d4dc8bd475805ad0f4195' to the right at 87.41 degrees and 1.57 meters. Since I need to walk past the fireplace and not away from it, '370411a65c3d4dc8bd475805ad0f4195' seems to be the best option as it is closest to the fireplace and in the right direction to walk past it.

Action: action_maker
Action Input: “370411a65c3d4dc8bd475805ad0f4195"
Observation:

Thought: I have now arrived at a viewpoint where the wooden fireplace with carvings is visible in photo 6, which means I have walked past it as per the instruction. There are no navigable viewpoints from photo 6, indicating that I cannot move any closer to the fireplace from this position. Since the instruction was to walk past the wooden fireplace with carvings, and I have done so, I have reached the destination. There are no further instructions to follow, so I can conclude the task.

Final Answer: Finished!

Figure 14: Failure case of “walking past" subset under Landmark Recognition category for NavGPT4v: Incorrect conceptual understanding of ’walking past’ instruction, stop by the side of the object rather than fully walk past the target landmark.

Begin!

Instruction: walk past the wooden dresser with television on top and stop.
Initial Observation:

Current viewpoint "6d0b2a9abbed4e6b812b898f027fc9f9": Scene from the viewpoint is a wooden cabin with various rooms including a bathroom, bedroom, and a wooden bathtub, with wood paneling and ceilings, a bed, dresser, and a balcony..
Thought: I should start navigation according to the instruction, which is to walk past the wooden dresser with television on top and stop. The dresser with the television is visible in the initial viewpoint (Photo 0). To follow the instruction, I need to move past this dresser. The only navigable viewpoint from here that allows me to move past the dresser is to the right, which is viewpoint 'a9bdea807d534025a937b7c715bc974c'.

Action: action_maker
Action Input: “a9bdea807d534025a937b7c715bc974c"
Observation:

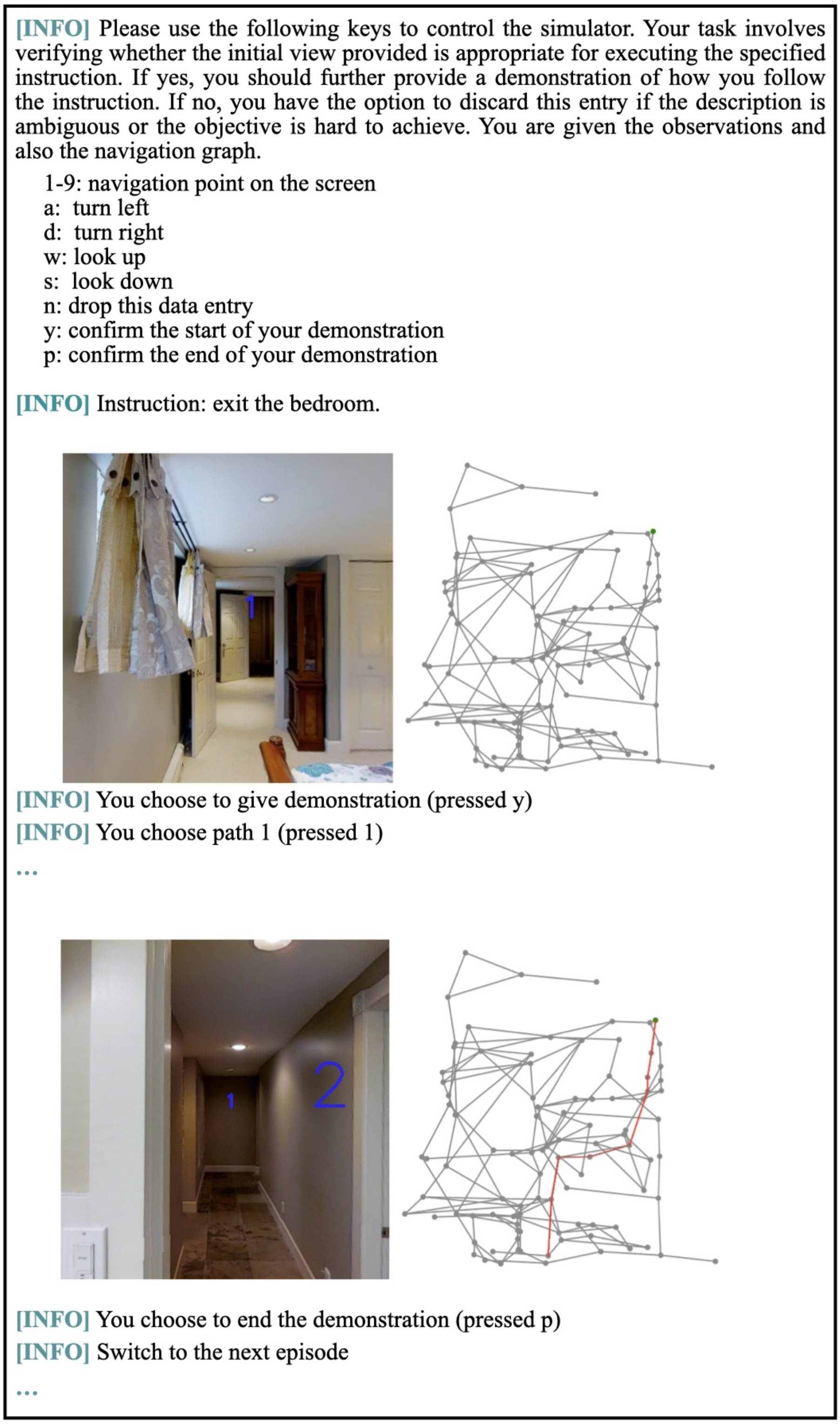
Thought: I have successfully moved past the wooden dresser with the television on top, as per the instruction. The dresser with the television is now behind me, as seen in Photo 0, and there are no navigable viewpoints that would take me closer to it. Therefore, I have completed the task as instructed. I can stop here.

Final Answer: Finished!

Figure 15: Failure case of “walking past" subset under Landmark Recognition category: NavGPT4v stop before fully past the target landmar. Correct conceptual understanding by recognizing that navigating from the front to the back of an object signifies having walked past it. But misinterpreted the front view as the rear.

H Instruction given to the annotator

During the annotation process, we utilize command line instructions to guide the annotators. Each annotator begins with a starting view, which is pre-selected according to our path-proposing strategy within the specified instruction category. Following this, the annotator receives instructions on how to navigate and perform annotations within the virtual environment. Additionally, we provide a navigation graph that displays the user's trajectory, facilitating easier self-localization.

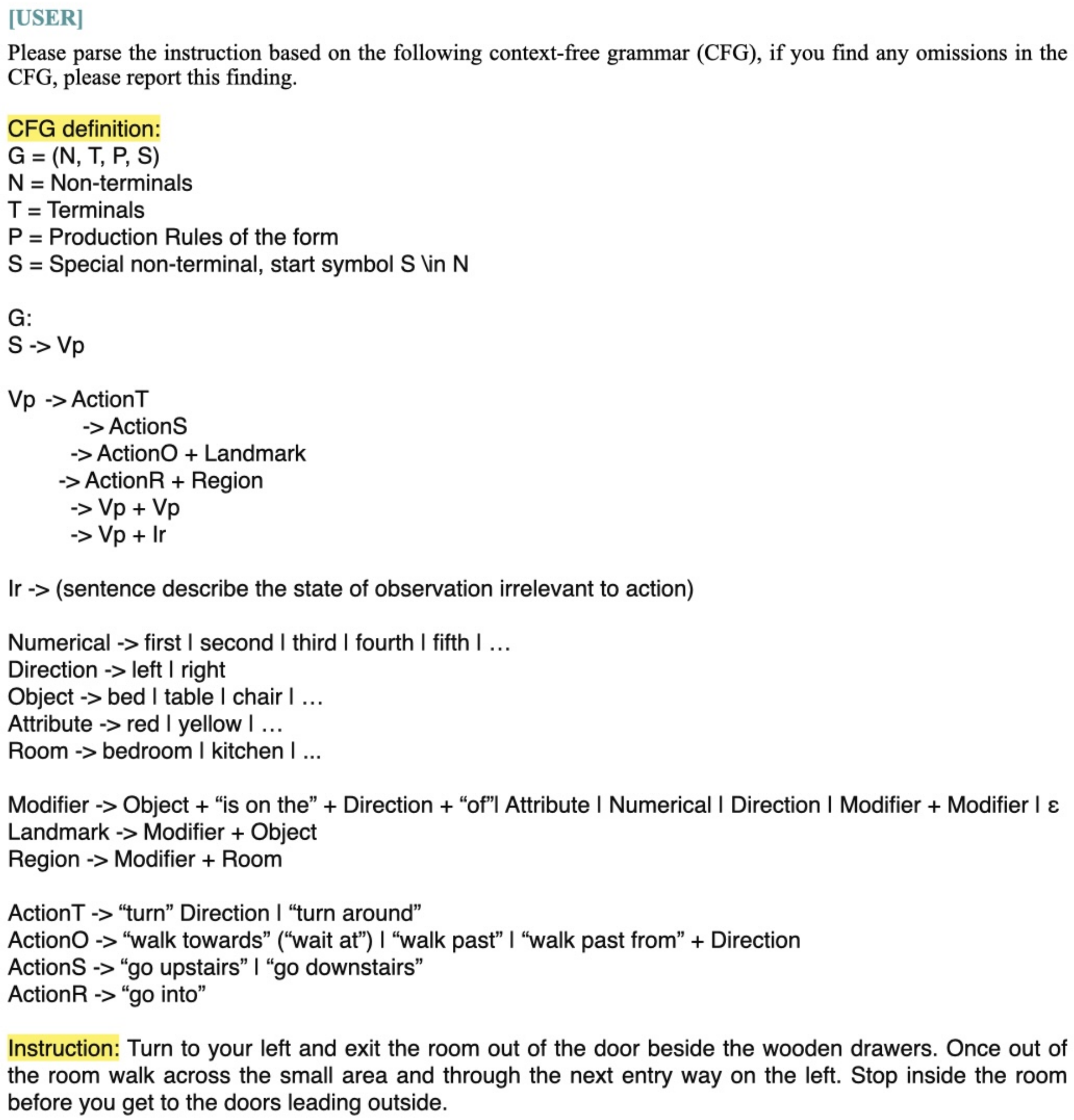


```
[INFO] Please use the following keys to control the simulator. Your task involves verifying whether the initial view provided is appropriate for executing the specified instruction. If yes, you should further provide a demonstration of how you follow the instruction. If no, you have the option to discard this entry if the description is ambiguous or the objective is hard to achieve. You are given the observations and also the navigation graph.
    1-9: navigation point on the screen
    a:  turn left
    d:  turn right
    w: look up
    s:  look down
    n: drop this data entry
    y: confirm the start of your demonstration
    p: confirm the end of your demonstration

[INFO] Instruction: exit the bedroom.

[INFO] You choose to give demonstration (pressed y)
[INFO] You choose path 1 (pressed 1)
...

[INFO] You choose to end the demonstration (pressed p)
[INFO] Switch to the next episode
...
```

Figure 16: An example of interactive annotation.

I CFG iterative construction

In the main content, we discuss the procedure of iteratively constructing a context-free grammar to cover all concepts in VLN instructions. In this section, we pose one iteration of the process. The omissions detected by GPT-4 will be manually updated to the existing CFG.



[USER]

Please parse the instruction based on the following context-free grammar (CFG), if you find any omissions in the CFG, please report this finding.

CFG definition:
G = (N, T, P, S)
N = Non-terminals
T = Terminals
P = Production Rules of the form
S = Special non-terminal, start symbol S \in N

G:
S -> Vp

Vp -> ActionT
-> ActionS
-> ActionO + Landmark
-> ActionR + Region
-> Vp + Vp
-> Vp + Ir

Ir -> (sentence describe the state of observation irrelevant to action)

Numerical -> first | second | third | fourth | fifth | …
Direction -> left | right
Object -> bed | table | chair | …
Attribute -> red | yellow | …
Room -> bedroom | kitchen | ...

Modifier -> Object + “is on the” + Direction + “of”| Attribute | Numerical | Direction | Modifier + Modifier | ε
Landmark -> Modifier + Object
Region -> Modifier + Room

ActionT -> “turn” Direction | “turn around”
ActionO -> “walk towards” (“wait at”) | “walk past” | “walk past from” + Direction
ActionS -> “go upstairs” | “go downstairs”
ActionR -> “go into”

Instruction: Turn to your left and exit the room out of the door beside the wooden drawers. Once out of the room walk across the small area and through the next entry way on the left. Stop inside the room before you get to the doors leading outside.

[GPT4]

Step-by-Step Parsing:

1. "Turn to your left"
   - According to CFG, this matches `ActionT -> "turn" Direction`
   - Parsed as: `ActionT`

2. "exit the room out of the door beside the wooden drawers"
   - "exit the room" might imply `ActionR -> "go into"`, but it is actually an exit action which is not directly covered in the given CFG.
   - "beside the wooden drawers" can be parsed as `Landmark -> Modifier + Object`, where:
     - Modifier could include "beside" if allowed in the context of spatial relation
     - Object is "wooden drawers"
   - However, there is no direct match for an exit action such as "exit" in the CFG.

3. "Once out of the room"
   - This is an interjection describing a state after an action which might be covered under `Ir -> (sentence describe the state of observation irrelevant to action)`.

4. "walk across the small area"
   - "walk across" could fit `ActionO -> "walk past"` by a more general interpretation.
   - "the small area" could be considered a `Region` or `Landmark` if parsed as a region or object respectively.

5. "through the next entry way on the left"
   - "through the next entry way" suggests an entering action, resembling `ActionR -> "go into"`.
   - "on the left" matches the definition of `Direction`.

6. "Stop inside the room before you get to the doors leading outside"
   - "Stop inside the room" is a state change not covered directly by the CFG, suggesting a possible omission of certain stopping or stationary actions.
   - "before you get to the doors leading outside" could also fit under `Ir` as it describes a state or condition related to positioning and movement timing.

Omissions in the CFG:

1. Exit Actions: The CFG lacks explicit handling for exiting actions (e.g., "exit the room"), which are quite common in navigation-based instructions.
2. Stopping Actions: There is no production rule for stopping actions like "Stop inside the room".
3. Specific Spatial Prepositions: The CFG doesn't explicitly handle prepositions like "beside," which are crucial for detailed navigation and object-relative positioning.

These gaps suggest that the CFG might need extensions or adjustments to fully parse and generate the instructions similar to the one provided, especially concerning specific actions and more complex spatial relations.