Vagueness Analysis of VLN Models

Chaitanya Chakka, Naman Gupta, Heer Patel, Astha Rastogi
Boston University
Boston, MA, USA
chvskch@bu.edu, naman11@bu.edu, heer29@bu.edu, asthar@bu.edu

Abstract

We propose dual frameworks to quantify instruction vagueness in vision-and-language navigation: continuous lexical specificity via Speciteller and discrete context-aware classification with GPT-4o-mini. Applied to ALFRED and CHORES using SPOC, MOCA, and Episodic Transformers, we find higher specificity correlates with improved success, notably in end-to-end models. Our analysis highlights architectural robustness factors and offers strategies for vagueness-aware training and data augmentation.

1. Introduction

Vision-and-Language Navigation (VLN) tasks require autonomous agents to interpret free-form natural language instructions and navigate complex, photo-realistic environments to accomplish semantically specified goals. Since the introduction of the Room-to-Room (R2R) benchmark [1] and the ALFRED benchmark [13], a wide variety of model architectures have demonstrated impressive gains in path fidelity and task success, ranging from late-fusion dual-tower designs to end-to-end episodic transformers.

However, these advances implicitly assume that instructions are sufficiently precise for an agent to unambiguously ground language to the visual scene. In practice, human-issued instructions often vary widely in their level of detail: some are "very specific" (e.g. "Walk past the blue vase on the right shelf, turn left at the red door"), while others are "very vague" (e.g. "Go down the hallway and stop"). We hypothesize that this variation in prompt vagueness contributes substantially to performance degradation and poor generalization in VLN agents.

In this work, we introduce a unified framework to quantify and analyze instruction vagueness in VLN. We first leverage Speciteller [10] to obtain a continuous specificity score based on syntactic and lexical cues. To incorporate the context of the environment, we then use a vision language model (GPT-4omini) to categorize each prompt into one of five discrete levels of vagueness. We apply these complementary metrics to two diverse datasets, ALFRED [13] and CHORES [7] and evaluate three state-of-the-art VLN agents (SPOC, MOCA, and episodic transformers).

Our contributions are threefold:

- We formalize the concept of prompt vagueness in VLN and propose a dual-metric annotation scheme combining Speciteller regression with VLM-based categorical classification.
- We conduct large-scale hypothesis tests linking vagueness scores to navigational success across three contemporary VLN architectures.
- We uncover architectural factors such as fusion strategy and memory retention that mitigate or exacerbate sensitivity to instruction vagueness, offering concrete guidance for future VLN design.

2. Previous Work

Prior work on vagueness and specificity in text has introduced a range of models for sentence-level prediction. Speciteller [10] proposed a semi-supervised logistic regression model that predicts specificity in sentences based on linguistic cues such as sentence length and lexical diversity. Ko et al. [8] improved on its generalizability by using an usupervised domain adaptive framework to produce continuous specificity scores across different domains, making it domain agnostic. These models enables fine-grained specificity estimation without requiring extensive labeled data.

Beyond specificity, recent research has focused on detecting and quantifying vagueness directly. VAGO [6] introduced a hybrid symboli-neural framework that categorizes vague terms into semantic classes, and uses them to compute sentence-level vagueness. Lebanoff et al [9] extended this work by exploring both context-agnostic models based on word embeddings and context-aware models using bidirectional LSTMs to synthesize vague sentences to improve classification performance. These efforts show the value of combiningsymbolic taxonomies like VAGO [6] with data-driven methods to model vagueness in texts and as such offer a strong foundation for analyzing vagueness in VLN instructions.

3. Datasets

To evaluate our vagueness analysis framework, we use two established vision-language navigation benchmarks: Alfred [13] and CHORES [7] each of which provides paired natural language instructions, simulator floor plans, and success/failure labels.

Alfred The Alfred dataset [13] contains 25,743 training episodes, 2,434 validation episodes, and 2,750 test episodes set in a virtual household environment. Each episode comprises a sequence of low-level navigation and manipulation instructions (e.g. "Walk over to the counter. Grab the butter knife. Wash the butter knife...") paired with RGB–D observations of the agent's egocentric view. The length of the instructions ranges from 10 to 60 tokens (mean $\approxeq$ 32), and the episodes last between 10 and 50 time steps. We use the standard train/val split and discard test labels, reserving 10% of the official train set as a held-out validation fold for hyperparameter tuning. These samples comprise of 832 episodes which are used for futher analysis.

CHORES The CHORES dataset [7] consists of 10,000 scripted "chore" tasks in kitchen-like layouts, each described by a single abstract instruction (e.g. "Go to a dish near a knife and a fork."). The corresponding floor plans feature up to eight distinct object categories (knife, fork, plate, etc.) arranged randomly. We leverage the official 80/10/10 split, yielding 8,000 training, 1,000 validation, and 1,000 test examples. The length of instructions is shorter (mean $\approxeq$ 12 tokens), which accentuates the impact of vague referents.

4. Methodology

This section outlines the different components of our framework. We begin with the different components in the framework followed by the actual flow to generate the scores. The complete design is depicted in 1.

4.1. Generating Vagueness scores

Vagueness using lexical cues To quantify the intrinsic specificity of each instruction present in natural language prompts, we utilize language-only models which can understand the semantic and syntactic properties of the given sentences to extract a score that represents how vague a given sentence is. The score ranges from 0 to 1 where scores close to 0 indicate that the sentence is vague while closer to 1 means it is more specific.

Vagueness using image and text modalities Due to the dependent nature of instruction prompts on the agent's environment, we utilize a Vision language model with strong visual reasoning capabilities to understand both the lexical cues as well as physical factors that can make a given prompt vague or specific. We demonstrate this with the following example: "Bring me a red cup" will be a specific instruction in a room with only one cup that is red but it will be a sufficient instruction in an environment where there are 5 cups each with different color. Since vision language models with language output are not good at generating continuous scores(as per the prior work done here [11]), we strategize around this by providing a set of buckets into which the model can classify into and we can use those categories for further processing.

This two-pronged approach lets us disentangle lexical from contextual vagueness: the continuous score captures word-level ambiguity, while VLN's discrete five-bin output reflects how the same words fare in situ, under real floorplan constraints.

4.2. Analysing generalizability

In order to understand how vagueness affects the performance of models, we devised a procedure which we explain below.

Vision Language Navigation models are generally evaluated based on whether the final goal is reached [4]. We utilize these binary labels to separate out episodes which are successful and failure. We now treat them as two different distributions and use different methods to compare these distributions. The hypothesis is that if the scores of successful episodes are more specific when compared to failure episodes, it means that the models are strongly adapted to the style of environment and therefore loose generalizability. To compare the distributions, we use two different techniques:

Difference in mean of the population In this technique, we find out the average of scores for each population and we find out the difference between mean of success distribution and failure distribution. We then report the percentage increase between success and failure cases. A higher positive value means failure scores are lower enforcing the fact that success episodes are having more specific instructions.

Hypothesis testing using Mann-Whitney U test The Mann-Whitney U Test evaluates whether two independent samples originate from the same underlying population distribution. Specifically, it tests if the distributions differ in central tendency, assessing whether one population tends to yield higher or lower values of a variable than the other. The hypotheses for the test are: The null hypothesis (H_o) states that the success distribution is not greater than failure distrubution. The alternative hypothesis (H_1) success distribution

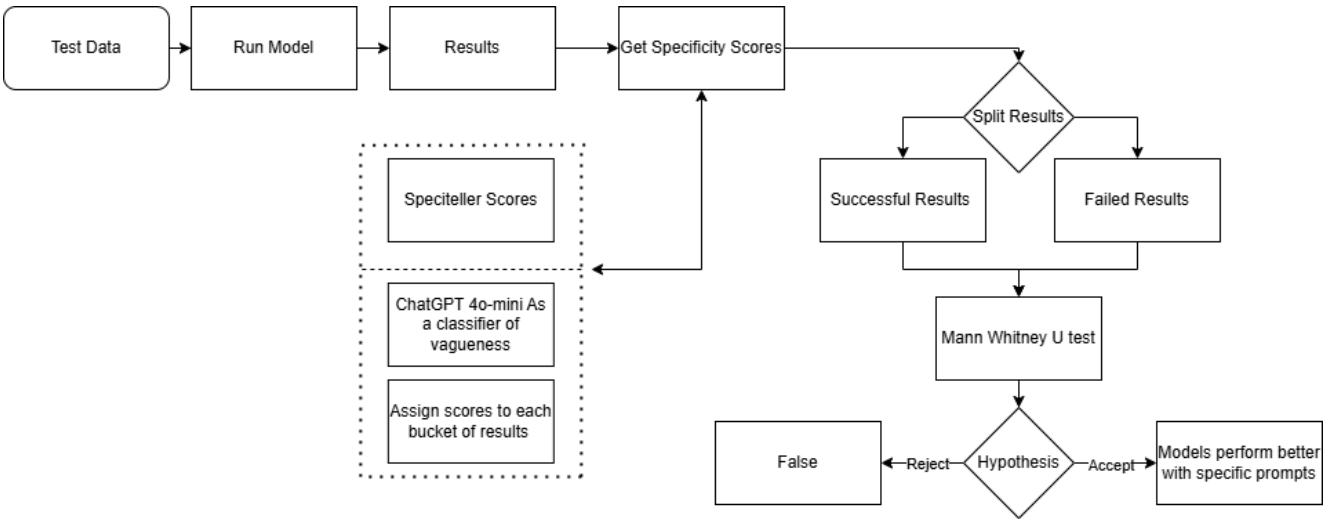


Figure 1. Flowchart illustrating the performance analysis pipeline.

is in fact greater than the failure distribution. We have chosen this method since the data is not normally distributed which is one of the main assumptions of this test.

5. Experiments

We execute our designed framework on three different models: Shortest Path Oracle Clone(SPOC) [3], Episodic Transformers(ET) [12] and Modular Object-Centric Approach(MOCA) [14]. This section discusses about generating the distributions from the models and the implementation details of other modules of the framework.

5.1. Vision Language Models

We analyze three off-the-shelf VLN agents by running zero-shot inference on the validation splits and recording their binary success labels. More details about the model architectures are given in appendix B

SPOC We load the publicly released SPOC checkpoint [3] and run it on the CHORES mini_val split. Each instruction is encoded by the language tower and fused via cross-modal attention with the visual encoder's floor-plan embedding. The simulator's built-in success flag indicates task completion.

MOCA We evaluate MOCA's official ALFRED checkpoint [14] on the ALFRED valid_seen episodes. The Interactive Perception Module generates object masks, which the Action Policy Module uses to predict actions. We extract each episode's success label directly from the environment's success/failure API.

Episodic Transformers We use the released Episodic Transformers checkpoint [12] on the ALFRED valid_seen split. This model performs intermediate fusion of text and vision at every time step and retains full-episode memory as it takes the previous actions as part of its input; we record its binary success outcome via the simulator return.

5.2. Scores using lexical cues

In order to generate vagueness scores based on text only, we utilize the Speciteller [10] tool which generates these scores using a semi-supervised approach. We utilized the training prompts of both ALFRED and CHORES dataset to fine-tune this model so that the model adapts to the navigation instruction domain.

5.3. Classification using VLM

In order to perform classification into buckets using a vision language model, we use OpenAI's GPT-4o-mini model [5]. We first extract the floor plans of each episode from the simulator implementations for SPOC model and scrape the demo website of Alfred dataset. These serve as the environment context for the VLN model to analyze the vagueness of the prompt. For each (floorplan, prompt) pair, we construct a chat request to the GPT-4o-mini model that contains a persona prompt which details about the vagueness task and some examples to incorporate few shot learning [2]. We then pass a user prompt to classify the instruction into one of the following categories: 'Very specific', 'Specific', 'Balanced', 'Vague', or 'Very vague', based solely on how clearly it can be executed in the depicted environment.

Model	p-value	Mean Spec. (Success)	Mean Spec. (Failure)	Δ (%)
SPOC	0.048	0.450	0.429	+5
Episodic Transformers	0.065	0.614	0.600	+2.5
MOCA	0.021	0.459	0.436	+5.1

Table 1. Speciteller specificity scores by episode outcome. Δ is the difference in success and failure expressed as percentage

Model	p-value	Mean Spec. (Success)	Mean Spec. (Failure)	Δ(%)
SPOC	0.262	0.295	0.285	+3.5
Episodic Transformers	0.771	0.476	0.487	-2.3
MOCA	0.265	0.486	0.478	+1.67

Table 2. VLM-based vagueness classification scores by episode outcome. Δ is the difference in success and failure expressed as percentage

6. Results

6.1. Speciteller-based Specificity Analysis

Table 1 reports the results of Mann–Whitney U tests comparing continuous Speciteller specificity scores between successful and failed navigation episodes for each model. We find:

- **SPOC** ($p = 0.048$): Success episodes have an average specificity of 0.450 compared to 0.429 for failures: 2.1 percentage-point increase that is statistically significant.
- **MOCA** ($p = 0.021$): Success episodes score 0.459 versus 0.436 for failures, yielding a 2.3 pp gap with even stronger significance.
- **Episodic Transformers** ($p = 0.065$): Success episodes score 0.614 against 0.600 for failures, a 1.4 pp advantage that shows a clear trend but falls just above the 0.05 significance threshold.

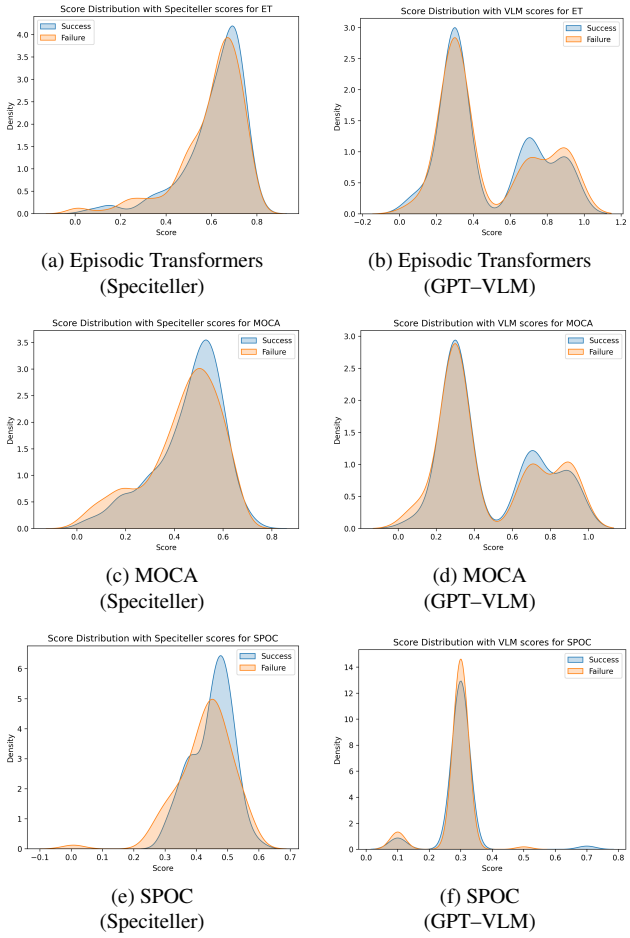
Across all three, higher lexical specificity correlates with navigation success, with success cases exhibiting 1.4 - 2.3 pp higher scores than failures. Hence this showcases that VLNs in general perform better when they are given more specific scores.

6.2. VLM-based Vagueness Classification

We repeat the same analysis using discrete vagueness categories from GPT-4o-mini (mapped to numeric scores). Table 2 shows:

- **SPOC** ($p = 0.262$): Success and failure differ by only 1.0 pp (0.295 vs. 0.285), not significant.
- **MOCA** ($p = 0.265$): A 0.8 pp gap (0.486 vs. 0.478), also not significant.
- **Episodic Transformers** ($p = 0.771$): Failures slightly outperform successes (0.487 vs. 0.476), reversing the trend.

The coarse, context-aware classification aligns loosely with Speciteller trends but yields higher variance and no statistically significant differences.



(a) Episodic Transformers (Speciteller)

(b) Episodic Transformers (GPT–VLM)

(c) MOCA (Speciteller)

(d) MOCA (GPT–VLM)

(e) SPOC (Speciteller)

(f) SPOC (GPT–VLM)

Figure 2. Vagueness score distributions for success vs. failure, arranged by model (rows) and method (columns). **Left column**: Speciteller; **Right column**: GPT–VLM. **Top-to-bottom**: Episodic Transformers, MOCA, SPOC. Speciteller shows clearer separation across all models compared to GPT–VLM classification.

6.3. Architectural Sensitivity to Vagueness

Figure 2 visualizes the specificity gap (success – failure) for each model under both metrics. Key observations:

- **Episodic Transformers** show the largest positive gap under Speciteller, reflecting strong robustness when prompts are clear, thanks to full-episode memory and intermediate fusion.
- **SPOC** and **MOCA** exhibit smaller gaps, indicating that these architectures rely more heavily on prompt detail.

Under VLM classification, all gaps shrink and even reverse for Episodic Transformers, highlighting the limitations of coarse categorization. These results collectively confirm that more specific instructions improve navigation performance, and that end-to-end architectures with memory are better equipped to handle residual vagueness.

7. Discussion

Our experiments demonstrate a clear link between prompt specificity and VLN performance, but they also surface important nuances in how vagueness should be measured and mitigated.

First, the continuous Speciteller scores yielded statistically significant distinctions between success and failure episodes for SPOC and MOCA, and a strong trend for Episodic Transformers (Table 1). This suggests that fine-grained lexical and syntactic features capture latent ambiguities that directly impact an agent's ability to ground language in visual observations. By contrast, the discrete five-way classification produced by our VLM (GPT-4o-mini) was noisier (Table 2), indicating that coarse buckets can obscure small but systematic differences in instruction clarity.

Second, comparing the model architectures B highlights that end-to-end fusion and full episode memory retention hallmarks of Episodic Transformers, confer greater robustness to vague instructions. We attribute this to two factors:

- *Intermediate vs. late fusion.* Early fusion of vision and language allows the model to align object references with visual features on every time step, compensating for missing or underspecified linguistic cues.
- *Memory over compressed history.* Retaining the full action–observation trajectory helps resolve ambiguity by aggregating context; in contrast, models that only attend to the most recent frames can lose critical referents when prompts omit explicit landmarks.

Third, our findings point to practical strategies for enhancing VLN models under real-world conditions, where users' instructions span the entire vagueness spectrum:

- *Data augmentation with paraphrasing.* Generating variants of training prompts at varied specificity levels could reduce sensitivity to missing details.
- *Vagueness-aware losses.* Incorporating a penalty for low specificity e.g. via auxiliary regression to Speciteller scores may encourage the model to seek clarifying observations.
- *Adaptive attention mechanisms.* Designing controllers that increase visual scanning when instructions are flagged as vague could mimic human behavior in ambiguous scenarios.

Finally, while continuous lexical measures proved most reliable in our tests, context-aware classification remains valuable for interpretability and user feedback. Future work should explore hybrid schemes that combine regression and categorical outputs, as well as larger sample sizes or finer granularity (e.g. seven buckets) to boost statistical power.

In summary, our analysis not only confirms that instruction vagueness degrades VLN success but also elucidates the architectural and methodological avenues by which future agents can better tolerate and even leverage ambiguous language.

Overall, our results confirm that higher prompt specificity correlates with navigation success, particularly when measured by continuous lexical cues. Discrete, context-aware categorization provides complementary insight but requires larger sample sizes or finer granularity to reach significance. These findings motivate architectural design choices that prioritize detailed grounding of language in the environment.

8. Conclusion

We have presented two complementary frameworks for quantifying instruction vagueness in vision-and-language navigation: (1) continuous specificity scores from Speciteller and (2) discrete vagueness categories from a vision–language model. Through large-scale hypothesis testing on ALFRED and CHORES, we showed that higher prompt specificity whether measured lexically or via context-aware classification correlates with increased task success, with continuous Speciteller scores yielding the clearest statistical signals. Our architectural analysis further revealed that end-to-end fusion and full-episode memory exemplified by Episodic Transformers substantially improve robustness to vagueness. Based on these insights, we proposed practical mitigation strategies, including vagueness-aware losses, data augmentation via paraphrasing, and adaptive attention controllers. Overall, our work highlights the critical role of instruction clarity in VLN performance and provides concrete guidance for designing agents capable of handling the full spectrum of human-generated commands. "'

9. Future Work

Building on our dual-metric framework for quantifying instruction vagueness, we identify several promising directions:

- **Continuous vagueness regression:** Train a vision–language model to predict a real-valued clarity score, using Speciteller outputs as weak supervision to capture subtle gradations in instruction specificity.
- **Vagueness-aware fine-tuning:** Augment VLN training with an auxiliary vagueness loss, penalizing low specificity predictions, and so agents learn to seek disambiguating cues when faced with unclear commands.

Author Contributions

- **Chaitanya Chakka:** Configured and evaluated Episodic Transformers on ALFRED, designed and applied the GPT-4o-mini few-shot prompting scheme for vagueness classification.
- **Naman Gupta:** Set up and ran the MOCA evaluation on ALFRED, executed Speciteller scoring on ALFRED prompts, and handled ALFRED dataset preprocessing.

- **Heer Patel:** Formulated the vagueness analysis framework, worked on instruction vagueness, and coordinated the hypothesis testing across all models, and assisted in prompt analysis and design.
- **Astha Rastogi:** Integrated and ran the SPOC agent on the CHORES mini_val split, and managed dataset preparation and preprocessing for SPOC experiments. She also ran speciteller for SPOC dataset.

References

[1] Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton Van Den Hengel. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3674–3683, 2018. 1

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

[3] Kiana Ehsani, Tanmay Gupta, Rose Hendrix, Jordi Salvador, Luca Weihs, Kuo-Hao Zeng, Kunal Pratap Singh, Yejin Kim, Winson Han, Alvaro Herrasti, et al. Spoc: Imitating shortest paths in simulation enables effective navigation and manipulation in the real world. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16238–16250, 2024. 3

[4] Jing Gu, Eliana Stefani, Qi Wu, Jesse Thomason, and Xin Eric Wang. Vision-and-language navigation: A survey of tasks, methods, and future directions. *arXiv preprint arXiv:2203.12667*, 2022. 2

[5] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024. 3

[6] Benjamin Icard, Vincent Claveau, Ghislain Atemezing, and Paul Égré. Measuring vagueness and subjectivity in texts: From symbolic to neural vago. In *2023 IEEE/WIC International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT)*, pages 395–401, 2023. 1

[7] Udit Khandelwal, Atiqur Anand, et al. Chores: A kitchen chore instruction dataset for vision-language research. In *In Submission (or workshop/etc.)*, 2021. 1, 2

[8] Wei-Jen Ko, Greg Durrett, and Junyi Jessy Li. Domain agnostic real-valued specificity prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 6610–6617, 2019. 1

[9] Logan Lebanoff and Fei Liu. Automatic detection of vague words and sentences in privacy policies. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3508–3517, Brussels, Belgium, 2018. Association for Computational Linguistics. 1

[10] Xin Li and Heng Ji. Speciteller: Sentence specificity prediction with implicit features. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics*, 2015. 1, 3

[11] Zhiqiu Lin, Deepak Pathak, Baiqi Li, Jiayao Li, Xide Xia, Graham Neubig, Pengchuan Zhang, and Deva Ramanan. Evaluating text-to-visual generation with image-to-text generation. In *European Conference on Computer Vision*, pages 366–384. Springer, 2024. 2

[12] Alexander Pashevich, Cordelia Schmid, and Chen Sun. Episodic transformer for vision-and-language navigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15942–15952, 2021. 3

[13] Mohit Shridhar, Jesse Thomason, Daniel Gordon, et al. ALFRED: A benchmark for instruction-following tasks in vision, language, and action. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020. 1, 2

[14] Kunal Pratap Singh, Suvaansh Bhambri, Byeonghwi Kim, Roozbeh Mottaghi, and Jonghyun Choi. Factorizing perception and policy for interactive instruction following. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1888–1897, 2021. 3

Appendix

A. GPT-4o-mini Prompt Personification

To generate context-aware vagueness labels, we use the following few-shot prompt schema when calling GPT-4o-mini via the OpenAI API:

Identity
You are a linguistic and visual expert capable of interpreting how instruction clarity varies with environmental context.
Instructions

- Your goal is to assess whether an instruction is specific or vague within a given environment.
- You are aware that an instruction may appear specific or vague depending on visual cues present in a given environment.
- You understand how people adapt instructions to what is visible or known in the environment, and you factor this into your interpretations.
- You maintain awareness of how environment affects clarity.

Figure 3. **Persona identity and instructions used for vagueness classification.**

User Prompt:
User Query: [INSTRUCTION]
Image Environment: [FLOOR PLAN]
Classify the instruction into: "Very Vague", "Vague", "Balanced", "Specific", "Very Specific"
We load the persona text as the system message and then substitute each Instruction–Floorplan pair into the user template.

Figure 4. **User Prompt used for vagueness classification.**

B. Model Architecture Diagrams

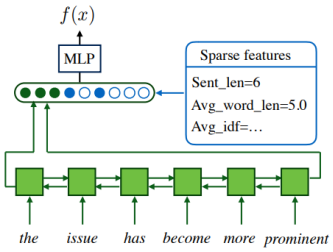
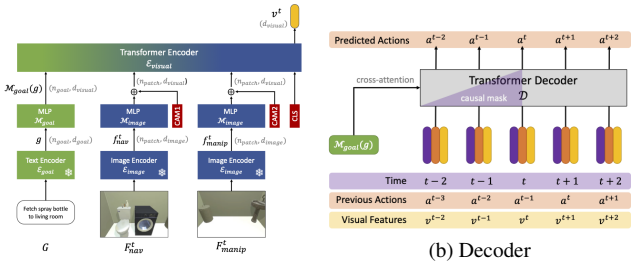


Figure 5. Speciteller Architecture



(a) Encoder (b) Decoder

Figure 6. SPOC dual-tower late-fusion architecture. (a) Instruction and visual encoders. (b) Cross-modal decoder that fuses encoded features for action prediction.

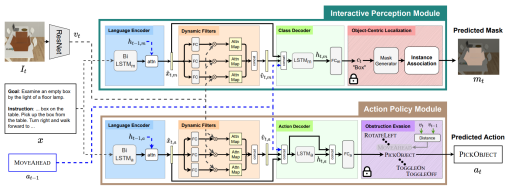


Figure 7. MOCA modular object-centric pipeline: (1) Interactive Perception Module for mask prediction; (2) Action Policy Module for sequential decision making.

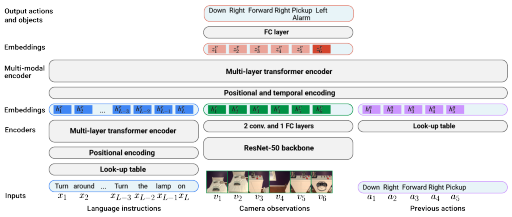


Figure 8. Episodic Transformer: intermediate fusion at each time step with full-episode memory attention.

C. Additional Prompt Examples

- **Very specific:** "Exit the kitchen, turn right at the blue rug, walk two steps, and stop in front of the green bookshelf."
- **Specific:** "Go through the doorway on your left and stop by the desk."
- **Balanced:** "Walk down the hallway past the table and stop near the sofa."
- **Vague:** "Proceed to the next room and stop."
- **Very vague:** "Go forward."