

Successfully Guiding Humans with Imperfect Instructions by Highlighting Potential Errors and Suggesting Corrections

♣Lingjun Zhao and ♣Nguyen X. Khanh and ♣Hal Daumé III

♣University of Maryland, College Park ♣University of California, Berkeley
lzhao123@umd.edu

Abstract

Language models will inevitably err in situations with which they are unfamiliar. However, by effectively communicating uncertainties, they can still guide humans toward making sound decisions in those contexts. We demonstrate this idea by developing HEAR, a system that can successfully guide humans in simulated residential environments despite generating potentially inaccurate instructions. Diverging from systems that provide users with only the instructions they generate, HEAR warns users of potential errors in its instructions and suggests corrections. This rich uncertainty information effectively prevents misguidance and reduces the search space for users. Evaluation with 80 users shows that HEAR achieves a 13% increase in success rate and a 29% reduction in final location error distance compared to only presenting instructions to users. Interestingly, we find that offering users possibilities to explore, HEAR motivates them to make more attempts at the task, ultimately leading to a higher success rate. To our best knowledge, this work is the first to show the practical benefits of uncertainty communication in a long-horizon sequential decision-making problem.¹

1 Introduction

Expecting language models to consistently make accurate predictions in a dynamic world is unrealistic (Kalai and Vempala, 2024; Xu et al., 2024). Evidence shows that these models often falter in unfamiliar situations (Wu et al., 2023; Dziri et al., 2024). Given the inherent fallibility of language models, an important research problem is to enable these models to successfully assist humans even when they make errors.

But how is it possible for a model to guide a human toward the right decisions when it cannot precisely specify what those decisions are? This

¹Our code and data for model and human evaluation are publicly released at <https://lingjunzhao.github.io/HEAR.html>.

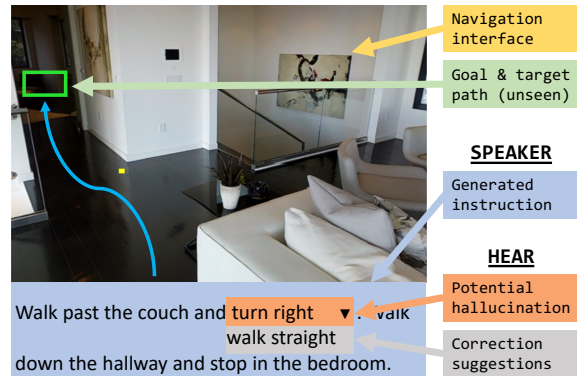


Figure 1: HEAR detects errors in a navigation instruction and suggests corrections. It enables humans to avoid being misled and efficiently search the environment, leading to improved performance.

work demonstrates the feasibility of tackling this problem in a language-guided visual navigation setting. Concretely, we develop HEAR (Hallucination DEtection And Remedy), a system that aids human navigation in 3D residential environments using potentially erroneous natural language instructions. The key to the success of HEAR is its ability to communicate various types of uncertainty information to users. Specifically, HEAR can identify and highlight potential errors in an instruction, and suggest possible corrections. This information prevents misdirection and narrows the search space for users, enabling them to navigate successfully even when given inaccurate instructions.

To our best knowledge, our work presents the first study on the effects of uncertainty communication on human decision making in a long-horizon task. Although uncertainty communication has been identified as crucial for AI systems, very few studies have investigated how uncertainty information impacts human decisions. Previous studies have primarily focused on classification tasks rather than long-horizon tasks, and on numerical uncertainty (i.e., probability) rather than verbal un-

certainties (Vodrahalli et al., 2022; Nizri et al.). By demonstrating that presenting uncertainties leads to a substantial performance boost in this navigation task, we provide strong evidence to support the development of these features in sequential decision-making AI agents.

To build HEAR, we tackle the problem of detecting and classifying hallucinated phrases in visually grounded instructions. This problem is particularly challenging in the environments we study because of the realism and diversity of the visual scenes. Our solution involves training two vision-language models: one for hallucination detection and the other for classification (i.e., deciding whether a phrase should be deleted or replaced). We combine these models to identify hallucinations in an instruction, as well as score and rank potential corrections. To train each model, we fine-tune a large vision-language model (Guhur et al., 2021) with synthetically created data to optimize for a contrastive learning objective. We introduce a practical methodology for generating synthetic data, combining rule-based approaches with large language models.

We conduct an evaluation with 80 human users to measure the effectiveness of HEAR. Our results demonstrate that incorporating HEAR improves user navigation outcomes. Specifically, HEAR increases the likelihood of a user successfully reaching their destination by 13% and reduces the average distance to the true destination by 29%. Analyzing human behavior reveals that by providing useful hints, HEAR motivates humans to put more effort into solving a task, leading to a higher success rate.

Interestingly, our results suggest that the uncertainty communication capabilities of a system do not need to be flawless to boost user performance. The components of HEAR are all imperfect: the error detection and correction, and the instruction generation capabilities are all of reasonable quality, but not faultless. However, because these capabilities *complement* one another, and complement the knowledge of the human user, they ultimately improve user decisions.

2 Related Work

Grounded instruction generation. Grounded instruction generation involves creating language instructions for navigation in situated environments, evolving from simple settings (Anderson et al.,

1991; Goeddel and Olson, 2012; Fried et al., 2018a) to more complex, photo-realistic simulations (Fried et al., 2018b; Kamath et al., 2023; Zhao et al., 2023a). Model-generated instructions can contain landmark errors (e.g., confusing a bathroom with a gym) and path errors (e.g., instructing a left turn instead of a right turn) (Wang et al., 2022). Zhao et al. (2023a) demonstrate a significant gap between the quality of model- and human-generated instructions. However, their work is not concerned with error detection.

Uncertainty communication for human-AI collaboration. As AI-assisted decision-making has become the norm, it is imperative to investigate the influence of human cognitive biases on their perception of model-generated information (Rastogi et al., 2022). Several studies have questioned the necessity of probabilistic calibration, showing that presenting *uncalibrated* probabilities may improve human decisions (Benz and Rodriguez, 2023; Vodrahalli et al., 2022; Nizri et al.). Other research proposes model designs to better calibrate human trust (Zhang et al., 2020; Ma et al., 2023; Buçinca et al., 2021). The experimental settings in all of these papers focus on classification tasks rather than long-horizon decision-making tasks, as explored in this work.

Regarding complementary performance in human-AI collaboration, Bansal et al. (2021) famously demonstrate that presenting model-generated explanations to humans does not enable human-AI teams to outperform individual entities. We present a contrasting result, showing that a complementary performance boost is possible with careful selection and presentation of model-generated information.

Hallucination Detection. Neural text generation models produce hallucinations in textual domains (Kalai and Vempala, 2024; Müller et al., 2020; Maynez et al., 2020; Durmus et al., 2020; Liu et al., 2022) as well as multimodal domains (Wiseman et al., 2017; Rohrbach et al., 2018; Liu et al., 2024; Chen et al., 2024). Hallucination detection has been explored, but primarily for machine translation (Dale et al., 2023; Xu et al., 2023; Wang and Sennrich, 2020; Zhou et al., 2021) or summarization (Falke et al., 2019; Kryscinski et al., 2020; Chen et al., 2021). Closest to our work is Zhao et al. (2023b), who study this problem in a similar visual navigation setting. However, their model cannot provide correction suggestions, nor do they

design user interfaces or perform evaluations with real human users.

3 Problem Setting

We consider the problem of generating language instructions to guide a human to follow an intended route in an environment. The concrete goal is to build a *speaker model* $S(\mathbf{w} \mid \mathbf{r})$, which takes an intended route \mathbf{r} as input and generates a corresponding language instruction \mathbf{w} as output (Figure 1). The instruction $\mathbf{w} = (w_1, \dots, w_n)$ is a sequence of words (e.g., “Walk past the couch and turn right. Walk down the hallway and stop in the bedroom.”). The route $\mathbf{r} = (\mathbf{o}_1, a_1, \dots, \mathbf{o}_l, a_l)$ is a sequence of observations and actions, where each observation is a collection of RGB images that capture the view at a location, and each action represents a transition from one location to another. The speaker is evaluated through an *instruction-following* task, in which a human user receives an instruction generated by the speaker and follows it in the corresponding environment. Success is achieved if the user reaches the final location along the intended route.

To simulate this problem, we employ the Matterport3D simulator and Room-to-Room (R2R) dataset (Anderson et al., 2018) for model training and human experiments. Matterport3D is a photo-realistic simulator that features images taken from various real residential buildings. The R2R dataset contains pairs of route and language instruction. The instructions contain more than 7,000 object and direction phrases.

We follow Zhao et al. (2023a) to train a T5-based (Raffel et al., 2020) speaker model. The instructions generated by this model often contain object or directional phrases that are inconsistent with the scenes along the intended route. We refer to such phrases as *hallucinations*. We categorize hallucinations into two types: *intrinsic hallucination* is a phrase that needs to be replaced because it inaccurately describes an observation or action (e.g., an instruction says “Walk past the reception desk and out the door on the **right**”, but on the intended route, the door is on the left); *extrinsic hallucination* is a phrase that needs to be removed because it does not have a correspondence on the input route (e.g., “Walk through the office and out of the office. **Walk into the hallway and turn left**”, where the second sentence describes a path that does not exist in the environment). Upon

inspecting 40 sample instructions generated by our speaker, we find that 67.5% of them have hallucinations, and that 20.9% of all the object and direction phrases are hallucinations.

4 HEAR: Hallucination Detection and Remedy

In this section, we introduce HEAR, which augments a speaker model by enabling it to (i) highlight potential hallucinations in an instruction and (ii) produce a list of plausible corrections for each hallucination. We expect that (i) would help a user avoid being misled into incorrect regions, while (ii) would reduce the effort required to locate the correct region. We build two models (§ 4.1, § 4.2, illustrated in Figure 2) to generate these pieces of information and design an interface to effectively convey them to users (§ 4.4).

4.1 Hallucination Detection

The hallucination detection model predicts hallucinations in an instruction. We adopt the model from Zhao et al. (2023b) but train it on a different training set so that it can detect *phrases* instead of just tokens as in the original work.

We frame the hallucination detection problem as a binary classification task: given an input $\mathbf{x} = (\mathbf{r}, \mathbf{w}, i, j)$ consisting of a route \mathbf{r} , an instruction \mathbf{w} , and token indices $i, j \in \{1, \dots, n\} (i \leq j)$, decide whether the phrase $\mathbf{w}_{i:j} = (w_i, w_{i+1}, \dots, w_j)$ is a hallucination (more specifically, whether it should be replaced or removed to make \mathbf{w} consistent with \mathbf{r}). For example, in the instruction shown in Figure 1, $\mathbf{w}_{6:7}$ is predicted to be a hallucination. We use a combination of a POS tagger² and GPT-3.5-turbo to identify the phrases to be classified.

Our model is a classifier $P_H(y = 1 \mid \mathbf{x} = (\mathbf{r}, \mathbf{w}, i, j))$ that is fine-tuned from the Airbert model (Guhur et al., 2021)—a vision-language model pre-trained on a large corpus of captioned household scenes collected from AirBnB.

For each instruction, we wrap the phrases to be classified between a pair of special tokens ([BH] and [EH]). For example, if $\mathbf{w}_{i:j}$ is classified, the instruction becomes $[w_1, \dots, [\text{BH}], w_i, \dots, w_j, [\text{EH}], \dots, w_n]$. The model takes as input this annotated instruction and the visual route and outputs a score $s(\mathbf{x})$. The hallucination confidence is calculated as $P_H(\mathbf{x}) = \sigma(s(\mathbf{x}))$, where σ is the sigmoid function. The

²<https://spacy.io>

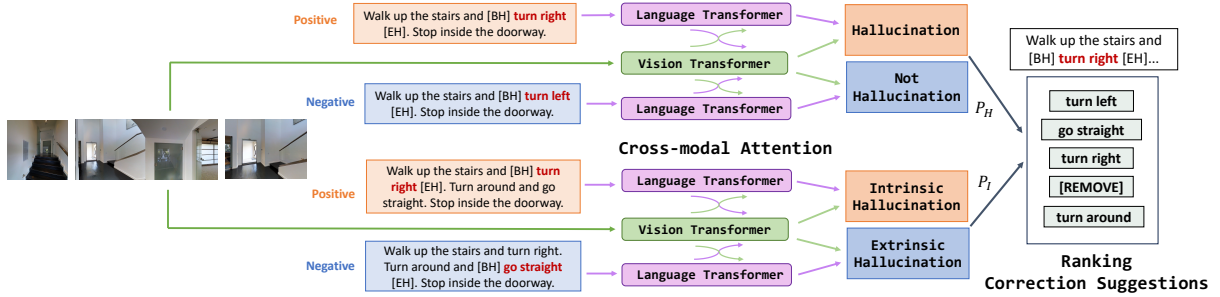


Figure 2: Our hallucination detection model (top) and hallucination type classification model (bottom). Each model takes a language instruction and a visual route as input and predicts a binary label. For hallucination detection, the label is whether a phrase is a hallucination. For hallucination-type classification, the label is whether a hallucination is extrinsic (needed to be replaced) or extrinsic (needed to be removed). Each model is built on top of a pre-trained vision-language model and is fine-tuned using contrastive learning. The first model is used to decide which phrases to highlight in an instruction, and the two models are combined to score and rank possible corrections.

model is trained with a contrastive objective (Majumdar et al., 2020) on pairs of positive and negative examples (described in §4.3).

4.2 Correction Suggestion

For each phrase $w_{i:j}$ classified as hallucination by P_H , we compute the top- K correction suggestions. To do so, we first generate a set of candidate corrections $\{\hat{w}_{i:j}^m\}_{m=1}^M$ (this procedure will be described in §4.3). For example, in Figure 1, $\{\hat{w}_{6:7}^m\}$ is $\{\text{turn right, walk straight}\}$. A special token [REMOVE] represents the deletion of the phrase. We train a hallucination-type classification model, which allows us to rank these candidates and choose the top K .

Ranking suggestions. As mentioned in §3, we categorize hallucinations into two types: *intrinsic* and *extrinsic*. Let z_x denote the hallucination type of a phrase x ; $z_x = 1$ if x is an intrinsic hallucination. We learn a binary classifier to estimate $P_I(z = 1 \mid x, y_x = 1)$ where $y_x = 1$ indicates that x is a hallucination. Let $x = (r, w, i, j)$ and \hat{x} be the corrected version of x obtained by replacing $w_{i:j}$ with a candidate correction $\hat{w}_{i:j}$. We compute a score $R(\hat{x})$ for every candidate (the higher is the better). We consider two cases. If \hat{x} indicates a replacement, we define $R(\hat{x})$ as:

$$P_I(z = 1 \mid x, y_x = 1) \cdot P_H(y = 1 \mid \hat{x}) \quad (1)$$

where the first term computes how likely x necessitates a replacement, while the second term captures how good the proposed replacement \hat{x} is. If \hat{x} indicates a deletion, we set $R(\hat{x}) = P_I(z = 0 \mid x, y_x = 1)$, which estimates the probability that x is an extrinsic hallucination (thus requiring deletion).

Hallucination type classification. The model P_I uses the same model architecture and is trained in a similar fashion as the hallucination model P_H . However, it solves a different problem: determining the type of a hallucination rather than identifying whether a phrase is a hallucination. This is achieved by training on a different dataset, as described in §4.3.

4.3 Dataset Creation

To train the models described in previous sections, we construct training datasets with positive and negative examples, defined by the specific classification problem. We also create a set of candidate corrections for each predicted hallucination. As human-labeled training data is costly to obtain, we synthetically create training data by taking human-generated instructions in the R2R training set and perturbing them using rule-based procedures and GPT models.

Training data for hallucination detection. For this problem, the negative examples are instructions from the R2R training set (Anderson et al., 2018), which are assumed to contain no hallucinations. To create a positive example from a negative example denoted by $x^- = (r, w^-, i, j)$, we perturb the instruction w^- in various ways. Following Zhao et al. (2023b), we focus on three types of intrinsic hallucinations: room, object, and direction. We create a room hallucinations by replacing a room phrase with another randomly chosen from a pre-composed list, and generate an object hallucination by replacing an object phrase with another that appears in the same instruction. For directions, since one can be expressed in various ways (e.g.

go straight is the same as proceed forward), we leverage GPT-3.5-turbo to modify them, using the following prompt (the few-shot examples are not shown for brevity; the full prompt is in §A.1):

SYSTEM: Find a directional word/phrase in the original instruction, and substitute it with a completely different directional word/phrase, so a person following the modified instruction would go in a different direction from the original instruction. Craft three modified instructions for each original instruction, and utilize the <s></s> tag pair to highlight the directional word/phrase you’ve modified in both the original and modified instructions.

Input: Walk out of the bedroom and turn left.

Output: <original1> walk <s> out of </s> the bedroom and turn left . </original1>
 <modified1> walk <s> around </s> the bedroom and turn left . </modified1>

Meanwhile, an extrinsic hallucination in an instruction is constructed by inserting a sentence taken from the same or a different instruction into a randomly selected beginning-of-sentence location within the instruction.

Multiple hallucinations are created within an instruction, but only one is wrapped by the [BH] [EH] tags for classification. We also add hallucinations to the negative example, but ensure that the span enclosed by [BH] [EH] is not a hallucination.

Training data for hallucination-type classification. For this dataset, both the positive and negative examples contain hallucinations, but the enclosed spans in the positive examples are intrinsic hallucinations, while those in the negative examples are extrinsic hallucinations. We apply the approach used in the detection problem to synthesize hallucinations.

Generating sets of candidate corrections. We generate a set of candidate corrections for each predicted hallucination. The candidate corrections for a room or an object hallucination are all the rooms and objects provided by the Matterport3D simulator. For directions, we ask GPT-4 to generate candidates, using the following prompt (the few-shot examples are not shown; the full prompt is in §A.1):

SYSTEM: Find directional words/phrases in the instruction and use <original> </original> tags to mark them, and list all the possible substitutions to change the meaning completely with <modified> </modified> tags, so that a person following the substituted instruction would go in a different direction from the original instruction. Use <sep> to separate each substitution, and do not mark the nouns.

Input: Walk out of the bedroom and turn left.

Output: walk <original1> out of </original1> <modified1> into <sep> around <sep> to the left of <sep> to the right of </modified1> the bedroom and <original2> turn left </original2> <modified2> go straight <sep> turn right <sep> turn around </modified2>

On average, we generate 47.6 candidates for each room or object hallucination and 5.9 candidates for each direction hallucination.

4.4 Designing Communication Interface

We build on top of the interface developed by [Ku et al. \(2021\)](#) and [Zhao et al. \(2023a\)](#) which allows a human to follow a language instruction to interact with a Matterport3D environment. We augment the interface to display highlights and suggestions for potential hallucinations. This section discusses our design principle; more details and a visualization of the interface are given in §A.5.

Our system generates a lot of information that can potentially be communicated to users. Deciding what piece of information to present and how to present it is vital to the success of the system. We choose not to present model probabilities to users because they can be miscalibrated and even if they are, different people might interpret them differently ([Vodrahalli et al., 2022](#)). Instead, we convey binary predictions of hallucinations through highlights. To do so, we select a decision threshold for the hallucination detection model to maximize its F-1 score on a manually annotated development set. If all phrases in a clause are highlighted, we simply highlight the entire clause and treat the clause as a single hallucination. For each instruction, we highlight at most three hallucinations predicted by the model, which is approximately the average number of hallucinations in an instruction detected by our human annotators.

For suggestions, because their presence can be overwhelming, we display them only when the user

deliberately seeks them out. Initially, the user sees only the instruction (potentially with hallucination highlights). We instruct them to click on a highlighted phrase if they also suspect it to be a hallucination and want to view possible corrections. If that happens, a drop-down menu will appear, displaying the top three suggestions in descending order by the score produced by our ranking models. The user can click on a suggestion to apply it to the instruction, which closes the drop-down menu. We explicitly instruct users to correct the instruction to encourage them to consider the suggestions.

A complication we encounter is to decide how much information about the true final location should be revealed to the users. If users do not know the true final locations, they cannot correct the instructions. However, if the location is completely revealed to them, the influence of the instructions on their behavior is significantly weakened, undermining the purpose of our study. To address this issue, we introduce a *Check* button, which enables the human to verify whether they have reached the final location. The button enables users to correct instructions while also retaining their reliance on instructions. In addition, analyzing user button usage uncovers interesting insights about their behavior.

5 Experiments

The questions that we aim to answer are:

- (Q1) Can HEAR reliably detect hallucinations and provide reasonable suggestions?
- (Q2) Does providing hallucination highlights and suggesting corrections improve human navigation performance?
- (Q3) What are the effects of highlights and suggestions on human behavior?

To answer Q1, we evaluate HEAR intrinsically with human-annotated data. To answer Q2 and Q3, we conduct a human evaluation with various systems, including ablated versions of HEAR and an oracle human-based system.

Data. To train the hallucination detection model, we synthetically generate a training set with 164,939 pairs of positive and negative examples (§4.3), which are created from the Room-to-Room (R2R) (Anderson et al., 2018) train set (4,675 routes, each route has 3 human-annotated instructions). To train the hallucination type classification model, we generate 117,357 pairs of positive and negative examples, created from the R2R train set.

For both evaluations, we first use a speaker model (§3) to generate instructions describing routes from the R2R validation seen split. For intrinsic evaluation and model selection, we randomly select and annotate 40 routes from the split as our *Dev Set*. For human evaluation, we use the 75 test routes from previous work (Zhao et al., 2023a,b) as our *Test Set*. There is no overlap between the Dev Set and the Test Set.

5.1 Intrinsic Evaluation: Hallucination Detection and Correction Suggestion

Annotation. We manually annotate hallucinations in the instructions generated by the speaker model, with mutual agreement from two of the authors. We also annotate corrections for those spans that we label hallucinations. In the end, we create intrinsic evaluation datasets consisting of 376 examples from the *Dev Set* for model selection; and 625 examples from the *Test Set* for testing, as well as used by the Oracle system for human evaluation (§5.2).

Systems. We implemented the following approaches (detailed hyperparameters in §A.3):

- (a) *HEAR* is our final system described in §4.1, §4.2, and §4.3.
- (b) *HEAR-SameEnvSwap* is similar to HEAR but the strategy to create room and object hallucinations is slightly different. Instead of following the procedure described in §4.3, we swap objects and rooms with those in the same environment (more details in §A.2).
- (c) *One-stage HEAR* combines hallucination detection and type classification into a single model (more details in §A.2). This model can directly score each correction suggestion.
- (d) *Random* samples a label uniformly at random among all possible labels, where the labels are {yes, no} for hallucination detection, and are the set of all possible 3-element subsets of the candidate set for correction suggestion.

Metrics. We compute *macro-averaged F-1* for hallucination detection and compute *Recall@3* for correction suggestion, which is the empirical chance that the gold correction appears in the top-3 suggestions ranked by a system.

Main results (Table 1). All the learned models substantially outperform the random baseline. In particular, the R@3 metrics of these models are in

System	Dev		Test	
	F-1	R@3	F-1	R@3
Random	42.6	47.8	43.8	50.4
HEAR-SameEnvSwap	64.8	75.0	69.1	78.7
One-stage HEAR	62.8	82.7	60.9	86.2
HEAR (final)	63.4	88.4	66.5	70.6

Table 1: Intrinsic evaluation of HEAR and our baseline systems. The decision threshold for each system is selected to maximize the F-1 score on the Dev Set. R@3 computes how often the top-3 correction suggestions contain the gold annotated correction.

the range of 70-90%, showing that they have a high potential to aid humans.

The results in hallucination detection show a clear trend, HEAR-SameEnvSwap is the best model in terms of F-1 score, followed by HEAR and finally one-stage HEAR. This indicates that the data-creation strategy in the HEAR-SameEnvSwap training set is beneficial. Meanwhile, the performance of one stage HEAR is low, possibly because it has twice as few parameters as the other two models. The results in correction suggestion recall are more nuanced: HEAR is best on Dev but one-stage HEAR is superior on Test. HEAR-SameEnvSwap outperforms others in hallucination detection, but its underperformance in correction suggestion indicates that the probabilities output by its hallucination detection module are not reliable.

Considering the average of F-1 and R@3, HEAR is the best performing model on the Dev set. Therefore, we select it for evaluation with human users.

5.2 Extrinsic Evaluation with Human Followers

Setup. We evaluate five systems:

- (a) *No communication* only tells the user that the instruction may be imperfect. It does not provide highlights and suggestions, and is similar to the system in Zhao et al. (2023a).
- (b) *HEAR (no suggestion)* tells the user that the instructions can be imperfect, highlights potential hallucinations, and tells the user that those phrases are potential errors. It does not provide suggestions. This system is similar to Zhao et al. (2023b).
- (c) *HEAR* is our final system, which adds to (b) the ability to suggest the top three corrections for each predicted highlight. We choose to present the top three suggestions to balance the system’s recall performance with user

mental load.

- (d) *Oracle (no suggestion)* is similar to (b) but highlights are annotated by the authors.
- (e) *Oracle* is similar to (c), but highlights and corrections are annotated by the authors. It displays two instead of three candidate suggestions: the original phrase and the gold correction.

We evaluate each system on 18 routes randomly chosen from the Test Set. For each route and each system, we recruit five human users using Amazon Mechanical Turk and ask them to follow the instruction generated by the system to describe the route. Users are paid \$4.10 for each session, which involves performing 7 navigation tasks and takes on average 19 minutes to complete. One of the tasks is a quality-control task that appears in every session. We analyze only sessions in which the user passes this task. After completing a session, users can provide feedback on the system. We ensure that each user encounters each route only once to prevent them from memorizing it. In total, we recruit 80 users and evaluate 525 navigation tasks.

Metrics. We evaluate navigation performance using standard metrics of the R2R task:

- (a) Success rate (*SR* ↑): fraction of examples in which the user’s final location is within 3m of the true goal;
 - (b) Navigation error (*DIST* ↓): distance between the user’s final location and the true goal.
- After a user has finished navigating, we ask for their subjective judgements about the route and the instruction, specifically:
- (a) Is the instruction easy to follow?
 - (b) Are you confident the path you followed is the intended path?
 - (c) Is the task mentally demanding?

For each question, we use 5-point Likert scale to ask for a rating on the affirmative statement (e.g., I am confident that I traversed the path that the AI system tried to describe).

HEAR enhances user navigation performance.

As seen in Figure 3, compared to no communication, simply highlighting potential errors using HEAR increases user success rate (+6.7%) and decreases navigation error (-1.9m). These results confirm that error highlights can effectively compensate for the deficiencies of the instruction generation model. A user described the effects of highlights as follows: “*highlights help me know if the instructions were going to be wrong. It made it*

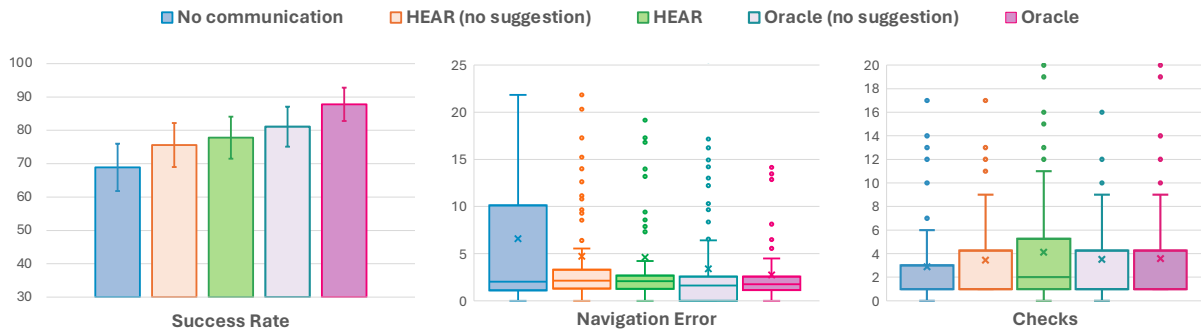
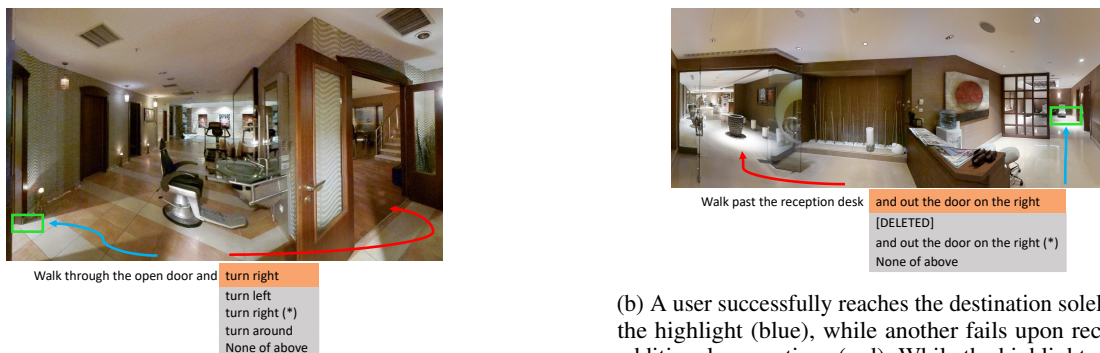


Figure 3: Performance measured by success rate (SR \uparrow) and navigation error (DIST \downarrow), and the number of check-button clicks recorded when human users perform navigation tasks with different assistant systems. HEAR improves user navigation performance and is competitive with the two Oracle systems. The error bars for SR represent 85% confidence intervals. For DIST and Checks, the “x” marks the mean, the line inside a bar marks the median, and the box represents the interquartile. Table 4 shows the corresponding results in table format.



(a) Highlights and suggestions directs a user to correctly make a left turn (blue). With only highlights, another user mistakenly turns right (red).

(b) A user successfully reaches the destination solely with the highlight (blue), while another fails upon receiving additional suggestions (red). While the highlight and the top suggestion ([delete]) are incorrect, they appear to reinforce each other, making the user believe that the highlight is correct and go in the alternative direction.

Figure 4: Example success and failure cases of HEAR (more in §A.7).

System	Easy to follow? \uparrow	Confident on actions? \uparrow	Mental burden? \downarrow
No communication	3.7	3.8	3.6
HEAR (no suggestion)	3.5	3.9	3.5
HEAR	4.0	4.2 \ddagger	3.5
Oracle (no suggestion)	3.9	3.8	3.6
Oracle	4.1 \dagger	4.1 \dagger	3.7

Table 2: User subjective ratings of systems after completing navigation sessions. The symbols \ddagger and \dagger indicate results that are significantly higher than those of the “No communication” system in the first row, with $p < 0.004$ (Bonferroni correction for 12 tests comparing 4 systems with “No communication”) and $p < 0.05$, respectively, as determined by a two-related-sample t-test.

easier to know where to go back to and retrace steps in order to go to the right place”. User performance is further improved with suggestions generated by HEAR (+2.2% in SR and -0.1m in DIST). Figure 4a shows an example where a user who is provided with both highlights and suggestions successfully reaches the target destination, whereas

another user who is shown only highlights does not.

Another notable pattern, shown in Figure 3 (middle), is that adding highlights and suggestions substantially decreases the variance of the navigation error. This indicates that highlights and suggestions effectively reduce the search space of the users.

HEAR receives favorable subjective ratings. As shown in Table 2, users find the instructions generated by HEAR (and Oracle systems) easier to follow and report greater confidence in their actions. Despite being asked to correct errors in the instructions, users do not report a significant increase in mental load.

HEAR improves user persistence in completing tasks. Figure 3 (rightmost) shows that users, on average, use the Check button more often when provided with highlights and suggestions. This result suggests that these features incentivize users to make more attempts to solve the task and consequently become more successful. We

hypothesize that by suggesting possibilities for exploration, users can avoid blind searches, making them more willing to invest effort. In contrast, without highlights and suggestions, users lack direction and may give up more quickly. They may perceive an entire instruction as incorrect and believe that the correct instruction could be entirely different from the current one, leading them to feel there is no hope in searching without further clues.

Better highlights and suggestions further improve user performance. Figure 3 shows that users benefit from a better hallucination detection model; they achieve a higher success rate (+5.5%) and a smaller navigation error (-1.3 m) when Oracle highlights are given, compared to when HEAR highlights are presented.

User performance is also enhanced when using an improved correction suggestion model: +10.0% in success rate and -1.9m in navigation error when using Oracle suggestions compared to when using HEAR suggestions. Figure 4b illustrates how a user is misled by incorrect highlights and suggestions.

6 Conclusion

We present a novel approach to enhance human task performance by effectively communicating model uncertainties. By encouraging users to refine AI-generated solutions, our approach offers an alternative to the conventional method that focuses on directly improving AI autonomous capabilities while overlooking human capabilities. To fully unlock the potentials of AI technologies, we advocate for viewing AI systems not as independent problem solvers, but as assistants and collaborators of humans.

While our research primarily addresses language-guided visual navigation, the insights gained are broadly applicable to other vision-language tasks. Specifically, we have demonstrated that: (i) it is feasible to generate meaningful error highlights and correction suggestions for vision-language models, and (ii) presenting these highlights and suggestions to human users can improve their decision-making. Moreover, our methods for creating synthetic errors and correction suggestions using rules and large language models are generalizable to various contexts.

Limitations

Due to cost constraints, the scale of our human evaluation is limited. We prioritize having

more annotators evaluate each route over having more routes. Furthermore, the assessment of cognitive load in the human evaluation study is not sufficiently robust; we plan to administer other schemes, such as the NASA Task Load Index (Hart, 2006), in future work.

Before using the navigation interface, users watch a video tutorial that explains the components of the interface and the associated questions. However, this could be improved by incorporating a warm-up practice session to help users become more familiar with the interface.

Another limitation of our human study is that we cannot determine how much of the performance improvement can be attributed to specific highlights and their associated correction suggestions, as task performance is assessed solely based on how close users are to the true final location. Additionally, we do not record the time when the Check button is pressed, which prevents us from analyzing the distribution of button presses throughout a navigation process.

Acknowledgements

We thank Hyemi Song, Yue Feng and Mingyang Xie for providing suggestions on improving human evaluation interface. We thank Eleftheria Briakou, Connor Baumler, Trista Cao, Navita Goyal and other group members for providing suggestions on human evaluation experimental design.

References

- Anne H Anderson, Miles Bader, Ellen Gurman Bard, Elizabeth Boyle, Gwyneth Doherty, Simon Garrod, Stephen Isard, Jacqueline Kowtko, Jan McAllister, Jim Miller, et al. 1991. The hrc map task corpus. *Language and speech*, 34(4):351–366.
- 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.
- Gagan Bansal, Tongshuang Wu, Joyce Zhou, Raymond Fok, Besmira Nushi, Ece Kamar, Marco Tulio Ribeiro, and Daniel Weld. 2021. Does the whole exceed its parts? the effect of ai explanations on complementary team performance. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–16.

- Nina L. Corvelo Benz and Manuel Gomez Rodriguez. 2023. [Human-aligned calibration for AI-assisted decision making](#). In *Thirty-seventh Conference on Neural Information Processing Systems*.
- Zana Bućinca, Maja Barbara Malaya, and Krzysztof Z Gajos. 2021. To trust or to think: cognitive forcing functions can reduce overreliance on ai in ai-assisted decision-making. *Proceedings of the ACM on Human-Computer Interaction*, 5(CSCW1):1–21.
- Sihao Chen, Fan Zhang, Kazuo Sone, and Dan Roth. 2021. [Improving faithfulness in abstractive summarization with contrast candidate generation and selection](#). In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5935–5941, Online. Association for Computational Linguistics.
- Xuwei Chen, Ziqiao Ma, Xuejun Zhang, Sihan Xu, Shengyi Qian, Jianing Yang, David F Fouhey, and Joyce Chai. 2024. Multi-object hallucination in vision-language models. *arXiv preprint arXiv:2407.06192*.
- David Dale, Elena Voita, Loic Barrault, and Marta R. Costa-jussà. 2023. [Detecting and mitigating hallucinations in machine translation: Model internal workings alone do well, sentence similarity Even better](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 36–50, Toronto, Canada. Association for Computational Linguistics.
- Esin Durmus, He He, and Mona Diab. 2020. [FEQA: A question answering evaluation framework for faithfulness assessment in abstractive summarization](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5055–5070, Online. Association for Computational Linguistics.
- Nouha Dziri, Ximing Lu, Melanie Sclar, Xiang Lorraine Li, Liwei Jiang, Bill Yuchen Lin, Sean Welleck, Peter West, Chandra Bhagavatula, Ronan Le Bras, et al. 2024. Faith and fate: Limits of transformers on compositionality. *Advances in Neural Information Processing Systems*, 36.
- Tobias Falke, Leonardo FR Ribeiro, Prasetya Ajie Utama, Ido Dagan, and Iryna Gurevych. 2019. Ranking generated summaries by correctness: An interesting but challenging application for natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2214–2220.
- Daniel Fried, Jacob Andreas, and Dan Klein. 2018a. [Unified pragmatic models for generating and following instructions](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1951–1963, New Orleans, Louisiana. Association for Computational Linguistics.
- Daniel Fried, Ronghang Hu, Volkan Cirik, Anna Rohrbach, Jacob Andreas, Louis-Philippe Morency, Taylor Berg-Kirkpatrick, Kate Saenko, Dan Klein, and Trevor Darrell. 2018b. Speaker-follower models for vision-and-language navigation. *Advances in Neural Information Processing Systems*, 31.
- Robert Goeddel and Edwin Olson. 2012. Dart: A particle-based method for generating easy-to-follow directions. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1213–1219. IEEE.
- Pierre-Louis Guhur, Makarand Tapaswi, Shizhe Chen, Ivan Laptev, and Cordelia Schmid. 2021. Airbert: In-domain pretraining for vision-and-language navigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1634–1643.
- Sandra G Hart. 2006. Nasa-task load index (nasa-tlx); 20 years later. In *Proceedings of the human factors and ergonomics society annual meeting*, volume 50, pages 904–908. Sage publications Sage CA: Los Angeles, CA.
- Adam Tauman Kalai and Santosh S Vempala. 2024. Calibrated language models must hallucinate. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing (STOC)*.
- Aishwarya Kamath, Peter Anderson, Su Wang, Jing Yu Koh, Alex Ku, Austin Waters, Yinfei Yang, Jason Baldridge, and Zarana Parekh. 2023. A new path: Scaling vision-and-language navigation with synthetic instructions and imitation learning. In *CVPR*.
- Wojciech Kryscinski, Bryan McCann, Caiming Xiong, and Richard Socher. 2020. [Evaluating the factual consistency of abstractive text summarization](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9332–9346, Online. Association for Computational Linguistics.
- Alex Ku, Peter Anderson, Jordi Pont-Tuset, and Jason Baldridge. 2021. [Pangea: The panoramic graph environment annotation toolkit](#).
- Hanchao Liu, Wenyuan Xue, Yifei Chen, Dapeng Chen, Xiutian Zhao, Ke Wang, Liping Hou, Rongjun Li, and Wei Peng. 2024. A survey on hallucination in large vision-language models. *arXiv preprint arXiv:2402.00253*.
- Tianyu Liu, Yizhe Zhang, Chris Brockett, Yi Mao, Zhifang Sui, Weizhu Chen, and Bill Dolan. 2022. [A token-level reference-free hallucination detection benchmark for free-form text generation](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6723–6737, Dublin, Ireland. Association for Computational Linguistics.
- Shuai Ma, Ying Lei, Xinru Wang, Chengbo Zheng, Chuhan Shi, Ming Yin, and Xiaojuan Ma. 2023. Who should i trust: Ai or myself? leveraging human and

- ai correctness likelihood to promote appropriate trust in ai-assisted decision-making. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, pages 1–19.
- Arjun Majumdar, Ayush Shrivastava, Stefan Lee, Peter Anderson, Devi Parikh, and Dhruv Batra. 2020. Improving vision-and-language navigation with image-text pairs from the web. In *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VI 16*, pages 259–274. Springer.
- Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. [On faithfulness and factuality in abstractive summarization](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1906–1919, Online. Association for Computational Linguistics.
- Mathias Müller, Annette Rios, and Rico Sennrich. 2020. [Domain robustness in neural machine translation](#). In *Proceedings of the 14th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Track)*, pages 151–164, Virtual. Association for Machine Translation in the Americas.
- Meir Nizri, Amos Azaria, Chirag Gupta, and Noam Hazon. Does calibration affect human actions?
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.
- Charvi Rastogi, Yunfeng Zhang, Dennis Wei, Kush R Varshney, Amit Dhurandhar, and Richard Tomsett. 2022. Deciding fast and slow: The role of cognitive biases in ai-assisted decision-making. *Proceedings of the ACM on Human-Computer Interaction*, 6(CSCW1):1–22.
- Anna Rohrbach, Lisa Anne Hendricks, Kaylee Burns, Trevor Darrell, and Kate Saenko. 2018. [Object hallucination in image captioning](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4035–4045, Brussels, Belgium. Association for Computational Linguistics.
- Kailas Vodrahalli, Tobias Gerstenberg, and James Y Zou. 2022. Uncalibrated models can improve human-ai collaboration. *Advances in Neural Information Processing Systems*, 35:4004–4016.
- Chaojun Wang and Rico Sennrich. 2020. [On exposure bias, hallucination and domain shift in neural machine translation](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3544–3552, Online. Association for Computational Linguistics.
- Su Wang, Ceslee Montgomery, Jordi Orbay, Vighnesh Birodkar, Aleksandra Faust, Izzeddin Gur, Natasha Jaques, Austin Waters, Jason Baldridge, and Peter Anderson. 2022. Less is more: Generating grounded navigation instructions from landmarks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15428–15438.
- Sam Wiseman, Stuart Shieber, and Alexander Rush. 2017. [Challenges in data-to-document generation](#). In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2253–2263, Copenhagen, Denmark. Association for Computational Linguistics.
- Zhaofeng Wu, Linlu Qiu, Alexis Ross, Ekin Akyürek, Boyuan Chen, Bailin Wang, Najoung Kim, Jacob Andreas, and Yoon Kim. 2023. Reasoning or reciting? exploring the capabilities and limitations of language models through counterfactual tasks. *arXiv preprint arXiv:2307.02477*.
- Weijia Xu, Sweta Agrawal, Eleftheria Briakou, Marianna J. Martindale, and Marine Carpuat. 2023. [Understanding and detecting hallucinations in neural machine translation via model introspection](#). *Transactions of the Association for Computational Linguistics*, 11:546–564.
- Ziwei Xu, Sanjay Jain, and Mohan Kankanhalli. 2024. Hallucination is inevitable: An innate limitation of large language models. *arXiv preprint arXiv:2401.11817*.
- Yunfeng Zhang, Q Vera Liao, and Rachel KE Bellamy. 2020. Effect of confidence and explanation on accuracy and trust calibration in ai-assisted decision making. In *Proceedings of the 2020 conference on fairness, accountability, and transparency*, pages 295–305.
- Lingjun Zhao, Khanh Nguyen, and Hal Daumé III. 2023a. Define, evaluate, and improve task-oriented cognitive capabilities for instruction generation models. In *Findings of ACL*.
- Lingjun Zhao, Khanh Nguyen, and Hal Daumé III. 2023b. [Hallucination detection for grounded instruction generation](#). In *Findings of the Empirical Methods in Natural Language Processing: EMNLP 2023*, Singapore. Association for Computational Linguistics.
- Chunting Zhou, Graham Neubig, Jiatao Gu, Mona Diab, Francisco Guzmán, Luke Zettlemoyer, and Marjan Ghazvininejad. 2021. [Detecting hallucinated content in conditional neural sequence generation](#). In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1393–1404, Online. Association for Computational Linguistics.

A Appendices

A.1 GPT for Dataset Creation

The following prompt is given to GPT-3.5-turbo to create direction hallucinations in instructions (§4.3):

Input: Walk out of the bedroom and turn left. Walk into the kitchen and stop by the counter.
Output: (1) <original1> walk out of the bedroom and <s>turn left</s> . walk into the kitchen and stop by the counter . </original1> <modified1> walk out of the bedroom and <s>turn right</s> . walk into the kitchen and stop by the counter . </modified1>
(2) <original2> walk <s>out of</s> the bedroom and turn left . walk into the kitchen and stop by the counter . </original2> <modified2> walk <s>around</s> the bedroom and turn left . walk into the kitchen and stop by the counter . </modified2>
(3) <original3> walk out of the bedroom and turn left . walk <s>into</s> the kitchen and stop by the counter . </original3> <modified3> walk out of the bedroom and turn left . walk <s>out of</s> the kitchen and stop by the counter . </modified3>
Input: Walk straight and turn left. Walk down the hallway and stop in the first doorway on your left.
Output: (1) <original1> walk straight and turn left . walk <s>down</s> the hallway and stop in the first doorway on your left . </original1> <modified1> walk straight and turn left . walk <s>up</s> the hallway and stop in the first doorway on your left . </modified1>
(2) <original2> walk straight and turn left . walk down the hallway and stop in the first doorway <s>on your left</s> . </original2> <modified2> walk straight and turn left . walk down the hallway and stop in the first doorway <s>to your right</s> . </modified2>
(3) <original3> walk straight and turn right . walk down the hallway and stop in the <s>first</s> doorway on your left . </original3> <modified3> walk straight and turn right . walk down the hallway and stop in the <s>second</s> doorway on your left . </modified3>
Input: Exit the bathroom. Walk forward and go down the stairs. Stop four steps from the bottom.
Output: (1) <original1> exit the bathroom . walk <s>forward</s> and go down the stairs . stop four steps from the bottom . </original1> <modified1> exit the bathroom . walk <s>backward</s> and go down the stairs . stop four steps from the bottom . </modified1>
(2) <original2> <s>exit</s> the bathroom . walk forward and go down the stairs . stop four steps from the bottom . </original2> <modified2> <s>enter</s> the bathroom . walk forward and go down the stairs . stop four steps from the bottom . </modified2>
(3) <original3> exit the bathroom . walk forward and go down the stairs . stop four steps from the <s>bottom</s> . </original3> <modified3> exit the bathroom . walk forward and go down the stairs . stop four steps from the <s>top</s> . </modified3>
Input: walk through open door, turn left, walk toward fireplace turn right, stop outside doorway.
Output: (1) <original1> walk through open door , turn left , walk toward fireplace turn right , stop <s>outside</s> doorway . </original1> <modified1> walk through open door , turn left , walk toward fireplace turn right , stop <s>inside</s> doorway . </modified1>
(2) <original2> walk through open door , <s>turn left</s> , walk toward fireplace turn right , stop outside doorway . </original2> <modified2> walk through open door , <s>go straight</s> , walk toward fireplace turn right , stop outside doorway . </modified2>
(3) <original3> walk through open door , turn left , walk <s>toward</s> fireplace turn right , stop outside doorway . </original3> <modified3> walk through open door , turn left , walk <s>away from</s> fireplace turn right , stop outside doorway . </modified3>

The following prompt is given to GPT-4 to generate candidate direction corrections (§4.3):

SYSTEM: Find directional words/phrases in the instruction and use <original> </original> tags to mark them, and list all the possible substitutions to change the meaning completely with <modified> </modified> tags, so that a person following the substituted instruction would go in a different direction from the original instruction. Use <sep> to separate each substitution, and do not mark the nouns.
Input: Walk out of the bedroom and turn left. Walk into the kitchen and stop by the counter.
Output: walk <original1> out of </original1> <modified1> into <sep> around <sep> to the left of <sep> to the right of </modified1> the bedroom and <original2> turn left </original2> <modified2> go straight <sep> turn right <sep> turn around </modified2> . walk <original3> into </original3> <modified3> out of <sep> pass </modified3> the kitchen and <original4> stop by </original4> <modified4> walk pass <sep> walk away from </modified4> the counter .
Input: Walk straight and turn left. Walk down the hallway and stop in the first doorway on your left.
Output: <original1> walk straight </original1> <modified1> turn left <sep> turn right <sep> turn around </modified1> and <original2> turn left </original2> <modified2> turn right <sep> go straight <sep> turn around </modified2> . <original3> walk down </original3> <modified3> stop in <sep> walk away from </modified3> the hallway and <original4> stop in </original4> <modified4> go into <sep> turn left at <sep> turn right at <sep> walk away from </modified4> the <original5> first </original5> <modified5> second <sep> third <sep> fourth <sep> last </modified5> doorway <original6> on your left </original6> <modified6> on your right <sep> straight ahead </modified6> .
Input: Exit the bathroom. Walk forward and go down the stairs. Stop four steps from the bottom.
Output: <original1> exit </original1> <modified1> enter </modified1> the bathroom . <original2> walk forward </original2> <modified2> go backward <sep> turn left <sep> turn right </modified2> and <original3> go down </original3> <modified3> go up <sep> stop by <sep> walk away from </modified3> the stairs . stop <original4> four </original4> <modified4> one <sep> two <sep> three </modified4> steps from the <original5> bottom </original5> <modified5> top </modified5> .
Input: walk through open door, turn left, walk toward fireplace turn right, stop outside doorway.
Output: <original1> walk through </original1> <modified1> walk past </modified1> open door , <original2> turn left </original2> <modified2> turn right <sep> turn around <sep> go straight </modified2> , <original3> walk toward </original3> <modified3> walk away from </modified3> fireplace <original4> turn right </original4> <modified4> turn left <sep> turn around <sep> go straight </modified4> , stop <original5> outside </original5> <modified5> inside </modified5> doorway .

Hyperparameter	Value
Learning rate	10^{-5}
Batch size	128
Optimizer	AdamW
Training iterations	5×10^5
Maximum instruction length	60
Image feature size	2048
Embedding dropout	0.1
Hidden size	768
Transformer layers	12
Transformer dropout rate	0.1
Number of parameters	250M
Computation and training time	RTX A4000: $\sim 72\text{h}$

Table 3: The hyperparameters of the hallucination detection and hallucination type classification models.

System	Success Rate \uparrow	Navigation Error \downarrow	Checks
No communication	68.9 ± 7.1	6.6 ± 1.6	2.9 ± 0.6
HEAR (no suggestion)	75.6 ± 6.6	4.7 ± 1.2	3.4 ± 0.7
HEAR	77.8 ± 6.3	4.6 ± 1.2	4.1 ± 0.8
Oracle (no suggestion)	$81.1 \pm 6.0^\dagger$	$3.4 \pm 0.9^\dagger$	3.5 ± 0.7
Oracle	$87.8 \pm 5.0^\ddagger$	$2.7 \pm 0.7^\ddagger$	3.6 ± 0.6

Table 4: Performance measured by success rate (SR \uparrow) and navigation error (DIST \downarrow), and the number of check-button clicks recorded when human users perform navigation tasks with different assistant systems. The error bars after \pm represent 85% confidence intervals. The symbols ‡ and † indicate results that are significantly higher than those of the “No communication” system in the first row, with $p < 0.004$ (Bonferroni correction) and $p < 0.05$, respectively, as determined by a two-related-sample t-test.

A.2 Model Variants

HEAR-SameEnvSwap. This system is identical to HEAR, but the synthetic hallucinations are created using different strategies. In the case of object hallucination, rather than swapping two objects within the same instruction, we replace an object in the instruction with another object randomly selected from those encountered along the described route. For room perturbation, instead of replacing a room mentioned in the instructions with another room from a list, we substitute it with another room that exists in the same environment.

One-stage HEAR. This underlying model of this system is similar to the hallucination detection model of HEAR. But its positive examples contain instructions with an empty token [REMOVE]. For example:

Positive: Go forward toward the windows. Exit [BH] [REMOVE] [EH] to living room.

Negative: Go forward toward the windows. Exit [BH] exercise room [EH] to living room.

Thus, instead of using two models as in HEAR, we can use this single model to score any correction, including deletion corrections. Concretely, with this model, we simply set the score function $R(\hat{x}) = 1 - P(y = 1 | \hat{x})$ where $P(y = 1 | \hat{x})$ is the probability output by the model. The training data of this model contain 216,323 pairs of positive and negative examples.

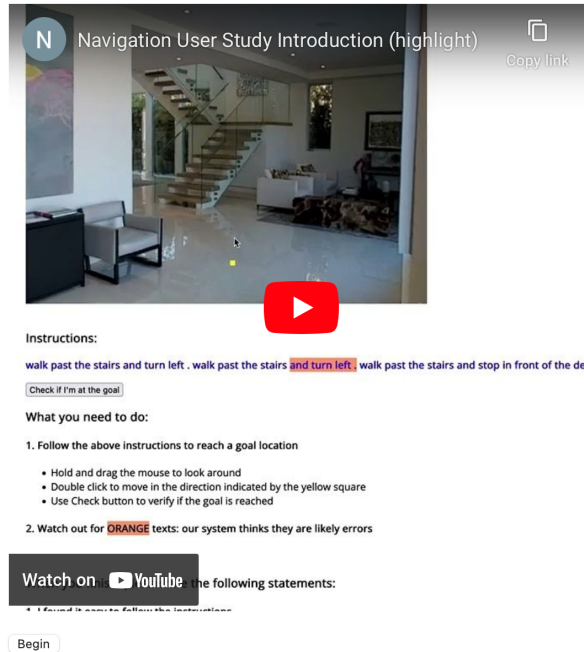
A.3 Hyperparameters and Tools

The hyperparameters and computation cost of the HEAR’s two models are listed in Table 3 (they have the same architecture and are trained in the same way). Other baseline models (§A.2) also have the same hyperparameters. We implement our models with Pytorch 1.7.1, Huggingface Transformers 4.5.1, NLTK 3.6.7, and use SciPy 1.6.0 for our result analyses.

Instructions

You will be helping us improve an AI system that assists humans with navigation. Before you start, please watch our **2-minute instructional video below** (if you haven't already). It is important that you watch the video to familiarize yourself with our goal and interface.

Instruction Video (click [CC] to turn on subtitles)



Navigation User Study Introduction (highlight) Copy link

Instructions:

walk past the stairs and turn left. walk past the stairs and turn left. walk past the stairs and stop in front of the desk

Check if I'm at the goal

What you need to do:

- Follow the above instructions to reach a goal location
 - Hold and drag the mouse to look around
 - Double click to move in the direction indicated by the yellow square
 - Use Check button to verify if the goal is reached
- Watch out for **ORANGE** texts: our system thinks they are likely errors

Watch on YouTube the following statements:

Begin

Figure 5: Introductory page of the human navigation task. A video instruction is provided.

A.4 Main Result Table

Table 4 shows human navigation performance when using different assistant systems, which corresponds to the charts in Figure 3.

A.5 Human Evaluation

Figure 6 shows the user interface of the *HEAR* and the *Oracle* systems. Figure 7 presents the interface of the *HEAR (no suggestion)* and *Oracle (no suggestion)* systems. Figure 8 is the interface of *No communication*. The interfaces are adapted from Zhao et al. (2023a) with the MIT License and Pangea³ with the Apache License v2.0. Before starting a task, we provide the user with a video instruction that shows them how to use the interface (Figure 5). After they complete the task, we record their route, the number of times they click on the *Check* button, and their subjective ratings. User participants must be at least 18 years old and speak English. The intended use of the system is first explained to them, and if they consent to perform the task, then they will be taken to the interface.

This study has been approved by the Institutional Review Board (IRB). For data anonymization, we removed the only PII information, the Amazon Mechanical Turk ID, after collecting the data. This information will also be removed in the future dataset release and replaced with serial numbers that do not reveal the identities of the participants. The dataset will be released under MIT license terms that are compatible with those of the tools used to create it and will be intended for research usage. We do not identify any potential risk to participants or the general public in releasing our dataset.

³<https://github.com/google-research/pangea>



Instructions:

walk down the stairs and turn right . walk past the couch and stop in front of the tv .

What you need to do:

1. Follow the above instructions to reach a goal location

- Hold and drag the mouse to look around
- Double click to move in the direction indicated by the yellow square
- Use Check button to verify if the goal is reached

2. Correct the instructions if you can't follow them

- Our system thinks **ORANGE** texts are likely errors
- Click and correct texts that are actual errors
- Select [DELETED] to remove a text
- Select [None of the above] if there is an error but no suggested correction is right
- (*) indicates the original phrase

When you finish, please rate the following statements:

1. I found it easy to follow the instructions.

Strongly Disagree Disagree Neutral Agree Strongly Agree

2. I am confident that I traversed the path that the AI system tried to describe.

Strongly Disagree Disagree Neutral Agree Strongly Agree

3. I found this navigation task mentally demanding.

Strongly Disagree Disagree Neutral Agree Strongly Agree

Figure 6: The interface used by the HEAR and Oracle systems.

A.6 Check Button Usage

In Figure 9, we show the number of checks when users succeed or fail. We observe that highlights and suggestions increase the number of checks in both cases.

A.7 Qualitative example (Figure 10)



Instructions:

walk down the stairs and turn right . walk past the couch and stop in front of the tv .

What you need to do:

1. Follow the above instructions to reach a goal location

- Hold and drag the mouse to look around
- Double click to move in the direction indicated by the yellow square
- Use Check button to verify if the goal is reached

2. Watch out for **ORANGE texts: our system thinks they are likely errors**

When you finish, please rate the following statements:

1. I found it easy to follow the instructions.

Strongly Disagree Disagree Neutral Agree Strongly Agree

2. I am confident that I traversed the path that the AI system tried to describe.

Strongly Disagree Disagree Neutral Agree Strongly Agree

3. I found this navigation task mentally demanding.

Strongly Disagree Disagree Neutral Agree Strongly Agree

Figure 7: The interface used by the HEAR and Oracle systems without correction suggestions.



Instructions:

walk down the stairs and turn right . walk past the couch and stop in front of the tv .

Check if I'm at the goal

What you need to do:

1. Follow the above instructions to reach a goal location

- Hold and drag the mouse to look around
- Double click to move in the direction indicated by the yellow square
- Use Check button to verify if the goal is reached

2. Our system may contain errors, use your judgement when the instruction is not followable

When you finish, please rate the following statements:

1. I found it easy to follow the instructions.

Strongly Disagree Disagree Neutral Agree Strongly Agree

2. I am confident that I traversed the path that the AI system tried to describe.

Strongly Disagree Disagree Neutral Agree Strongly Agree

3. I found this navigation task mentally demanding.

Strongly Disagree Disagree Neutral Agree Strongly Agree

Figure 8: The interface without highlights and suggestions (no communication).

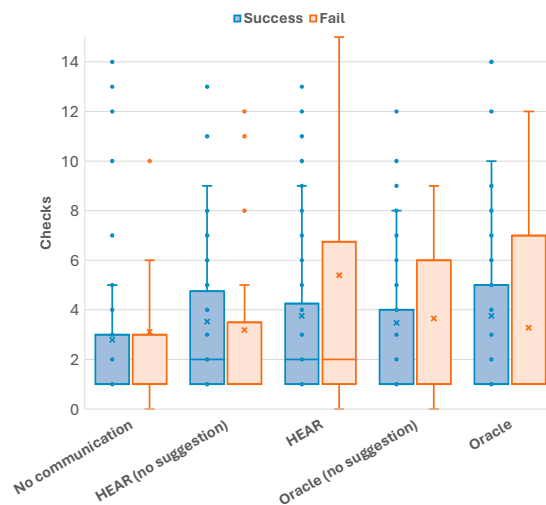
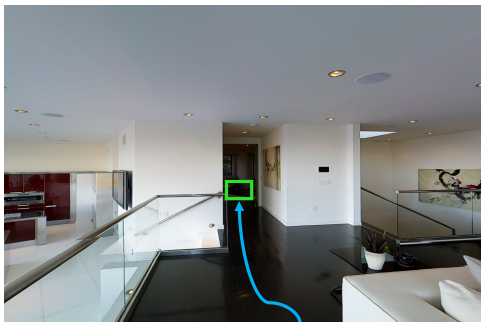


Figure 9: Number of check-button clicks when users succeed and fail on the task.



Walk forward and **turn left** . Walk forward and exit the building

(a) A qualitative example where our system accurately highlights a hallucinated direction and helps a user navigate successfully. Another user, who is not given the highlight, follows the instruction and takes the wrong turn.



walk past the couch and **turn right** . walk down the hallway and stop in the bedroom...
 turn left
 turn right (*)
 [DELETED]
 None of above

(b) Accurate highlights from our system help a user to correctly go straight. Although the suggestions are not accurate, it can still enable the user to make the right decision.



walk past the couch and stop **in front of** the tv .
 in front of (*)
 next to
 away from
 None of above

(c) In this case, the correct instruction is: walk past the couch and stop in front of the *bed*. Inaccurate highlight generated by our system leads the user to the wrong location.

Figure 10: Additional qualitative examples. The true route and the target destination are marked by a blue arrow and a green box, respectively. The user's route is indicated by a red arrow.