In *Conference on Robot Learning* (CoRL 2019),
Osaka, Japan, October 2019

Vision-and-Dialog Navigation

Jesse Thomason **Michael Murray** **Maya Cakmak** **Luke Zettlemoyer**
Paul G. Allen School of Computer Science and Engineering
University of Washington
{jdtho, mmurr, mcakmak, lsz}@cs.washington.edu

Abstract: Robots navigating in human environments should use language to ask for assistance and be able to understand human responses. To study this challenge, we introduce Cooperative Vision-and-Dialog Navigation, a dataset of over 2k embodied, human-human dialogs situated in simulated, photorealistic home environments. The *Navigator* asks questions to their partner, the *Oracle*, who has privileged access to the best next steps the *Navigator* should take according to a shortest path planner. To train agents that search an environment for a goal location, we define the Navigation from Dialog History task. An agent, given a target object and a dialog history between humans cooperating to find that object, must infer navigation actions towards the goal in unexplored environments. We establish an initial, multi-modal sequence-to-sequence model and demonstrate that looking farther back in the dialog history improves performance. Sourcecode and a live interface demo can be found at https://cvdn.dev/

1 Introduction

Dialog-enabled smart assistants, which communicate via natural language and occupy human homes, have seen widespread adoption in recent years. These systems can communicate information, but do not manipulate objects or actuate. By contrast, manipulation-capable and mobile robots are still largely deployed in industrial settings, but do not interact with human users. Dialog-enabled robots can bridge this gap, with natural language interfaces helping robots and non-experts collaborate to achieve their goals [1, 2, 3, 4, 5].

Navigating successfully from place to place is a fundamental need for a robot in a human environment and can be facilitated, as with smart assistants, through dialog. To study this challenge, we introduce Cooperative Vision-and-Dialog Navigation (CVDN), an English language dataset situated in the Matterport Room-2-Room (R2R) simulation environment [6, 7] (Figure 1). CVDN can be used to train navigation agents, such as language teleoperated home and office robots, that ask targeted questions about where to go next when unsure. Additionally, CVDN can be used to train agents that can answer such questions given expert knowledge of the environment to enable automated language guidance for humans in unfamiliar places (e.g., asking for directions in an office building). The photorealistic environment used in CVDN may enable agents trained in simulation to conduct and understand dialog from humans to transfer those skills to the real world. The dialogs in CVDN contain nearly three times as many words as R2R instructions, and cover average path lengths more than three times longer than paths in R2R.

In Section 2 we situate the Vision-and-Dialog Navigation paradigm. After introducing CVDN (Section 3), we create the Navigation from Dialog History (NDH) task with over 7k instances from CVDN dialogs (Section 4). We evaluate an initial, sequence-to-sequence model on this task (Section 5). The sequence-to-sequence model encodes the human-human dialog so far and uses it to infer navigation actions to get closer to a goal location. We find that agents perform better with more dialog history and when mixing human and planner supervision during training. We conclude with next directions for creating tasks from CVDN, such as two learning agents that must be trained cooperatively, and more nuanced models for NDH, where our initial sequence-to-sequence model leaves headroom between its performance and human-level performance (Section 6).

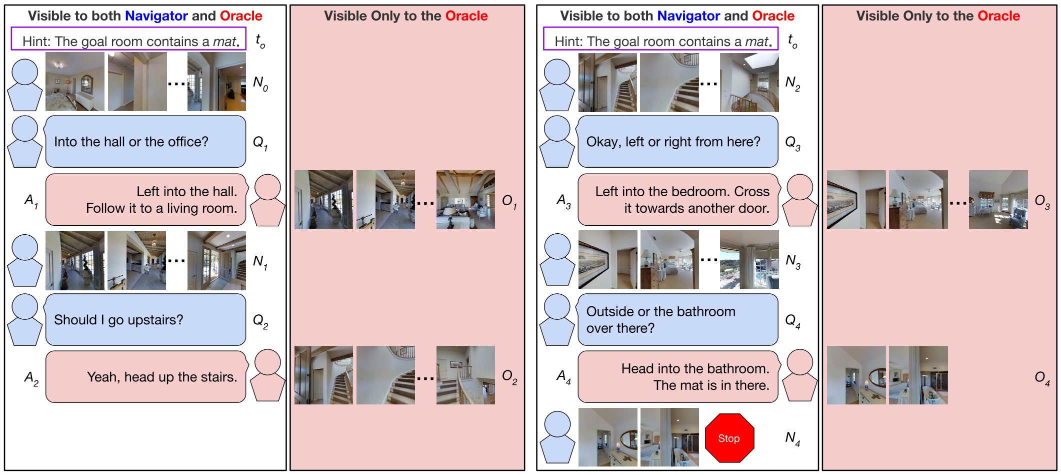
Visible to both Navigator and Oracle
Visible Only to the Oracle
Hint: The goal room contains a mat.
t_o
N_0
Into the hall or the office?
Q_1
A_1
Left into the hall. Follow it to a living room.
O_1
N_1
Should I go upstairs?
Q_2
A_2
Yeah, head up the stairs.
O_2
Visible to both Navigator and Oracle
Visible Only to the Oracle
Hint: The goal room contains a mat.
t_o
N_2
Okay, left or right from here?
Q_3
A_3
Left into the bedroom. Cross it towards another door.
O_3
N_3
Outside or the bathroom over there?
Q_4
A_4
Head into the bathroom. The mat is in there.
O_4
Stop
N_4

Figure 1: In Cooperative Vision-and-Dialog Navigation, two humans are given a hint about an object t_o in the goal room. The *Navigator* moves (N) through the simulated environment to find the goal room, and can stop at any time to type a question (Q) to the *Oracle*. The *Oracle* has a privileged view of the best next steps (O) according to a shortest path planner, and uses that information to answer (A) the question. The dialog continues until the *Navigator* stops in the goal room.

2 Related Work and Background

Dialogs in CVDN begin with an underspecified, ambiguous instruction analogous to what robots may encounter in a home environment (e.g., "Go to the room with the bed"). Dialogs include both navigation and question asking / answering to guide the search, akin to a robot agent asking for clarification when moving through a new environment. Table 1 summarizes how CVDN combines the strengths and difficulties of a subset of existing navigation and question answering tasks.

Vision-and-Language Navigation. Early, simulator-based Vision-and-Language Navigation (VLN) tasks use language instructions that are unambiguous—designed to uniquely describe the goal—and fully specified—describing the steps necessary to reach the goal [8, 9]. In a more recent setting, a simulated quadcopter drone uses low-level controls to follow a route described in natural language [10]. In photorealistic simulation environments, agents can navigate high-definition scans of indoor scenes [7] or large, outdoor city spaces [11]. In interactive question answering [12, 13] settings, the language context is a single question (e.g., "What color is the car?") that requires navigation to answer. The questions serve as underspecified instructions, but are unambiguous (e.g., there is only one car whose color can be asked about). These questions are generated from templates rather than human language. In CVDN, input is an underspecified hint about the goal location (e.g., "The goal room has a sink") requiring exploration and dialog to resolve. Rather than single instructions, CVDN includes two-sided, human-human dialogs.

Question Answering and Dialog. In Visual Question Answering (VQA), agents answer language questions about a static image. These tasks exist for templated language on rendered images [14] and human language on real-world images [15, 16, 17]. Later extensions feature two-sided dialog, where a series of question-answer pairs provide context for the next question [18, 19]. Question answering in natural language processing is a long-studied task for questions about static text documents (e.g., the Stanford QA Dataset [20]). Recently, this paradigm was extended to two-sided dialogs via human-human, question-answer pairs about a document [21, 22, 23]. Questions in these datasets are unambiguous: they have a right answer that can be inferred from the context. By contrast, CVDN conversations begin with a hint about the goal location that is always ambiguous and requires cooperation between participants. Contrasting VQA, because CVDN extends navigation the visual context is temporally dynamic—new visual observations arrive at each timestep.

Task-oriented Dialog. In human-robot collaboration, robot language requests for human help can be generated to elicit non-verbal human help (e.g, moving a table leg to be within reach for the robot) [1]. However, humans may use language to respond to robot requests for help in task-oriented

Dataset	**—Language Context—**				**—Visual Context—**		
	Human	Amb	UnderS	Temporal	Real-world	Temporal	Shared
MARCO[8, 9], DRIF[10]	✓	✗	✗	**1I**	✗	**Dynamic**	-
R2R[7], Touchdown[11]	✓	✗	✗	**1I**	✓	**Dynamic**	-
EQA[12], IQA[13]	✗	✗	✓	**1Q**	✗	**Dynamic**	-
CLEVR[14]	✗	✗	-	**1Q**	✗	**Static**	-
VQA[15, 16, 17]	✓	✗	-	**1Q**	✓	**Static**	-
CLEVR-Dialog[18]	✗	✗	-	**2D**	✗	**Static**	✓
VisDial[19]	✓	✗	-	**2D**	✓	**Static**	✓
VLNA[24], HANNA[25]	✗	✓	✓	**1D**	✓	**Dynamic**	✗
TtW[26]	✓	✗	✓	**2D**	✓	**Dynamic**	✗
CVDN	✓	✓	✓	**2D**	✓	**Dynamic**	✓

Table 1: Compared to existing datasets involving vision and language input for navigation and question answering, CVDN is the first to include two-sided dialogs held in natural language, with the initial navigation instruction being both ambiguous (*Amb*) and underspecified (*UnderS*), and situated in a photorealistic, visual navigation environment viewed by both speakers. For temporal language context, we note single navigation instructions (*1I*) and questions (*1Q*) versus 1-sided (*1D*) and 2-sided (*2D*) dialogs.

dialogs [3, 5, 27]. Recent work adds requesting navigation help as an action, but the response either comes in the form of templated language that encodes gold-standard planner action sequences [24] or as an automatic generation trained from human instructions and coupled with a visual goal frame as additional supervision [25]. Past work introduced Talk the Walk (TtW) [26], where two humans communicate to reach a goal location in a photorealistic, outdoor environment. In TtW, the guiding human does not have an egocentric view of the environment, but an abstracted semantic map, and so language grounding centers around semantic elements like “bank” and “restaurant” rather than visual features, and the target location is unambiguously shown to the guide from the start. In CVDN, a *Navigator* human generates language requests for help, and an *Oracle* human answers in language conditioned on higher-level, visual observations of what a shortest-path planner would do next, with both players observing the same, egocentric visual context. In some ways, CVDN echoes several older human-human, spoken dialog corpora like the HCRC Map Task [28], SCARE [29], and CReST [30], but these are substantially smaller and have fewer and less rich environments.

Background: Matterport Simulator and the Room-2-Room Task. We build on the R2R task [7] and train navigation agents using the same simulator and API. MatterPort contains 90 3D house scans, with each scan S divided into visual panoramas $p \in S$ (nodes which a navigation agent can occupy) accompanied by an adjacency matrix A_S. We differentiate between the *steps* and *distance* between p and q—*steps* represent the number of intervening nodes d_h, while *distance* is defined in meters as d_m. Step distance $d_h(p, q)$ is the number of hops through A_S to get from node p to node q. The distance in meters $d_m(p, q)$ is defined as physical distance if $A_S[p, q] = 1$ or the shortest route between p and q otherwise. On average, 1 step corresponds to 2.25 meters.

At each timestep, an agent emits a navigation action taken in the simulated environment. The actions are to turn *left* or *right*, tilt *up* or *down*, move *forward* to an adjacent node, or *stop*. After taking any action except *stop*, the agent receives a new visual observation from the environment. The *forward* action is only available if the agent is facing an adjacent node.

3 The Cooperative Vision-and-Dialog Navigation Dataset

We collect 2050 human-human navigation dialogs, comprising over 7k navigation trajectories punctuated by question-answer exchanges, across 83 MatterPort [6] houses.[1] We prompt with initial instructions that are both *ambiguous* and *underspecified*. An *ambiguous* navigation instruction is one that requires clarification because it can refer to more than one possible goal location. An *underspecified* navigation instruction is one that does not describe the route to the goal.

Dialog Prompts. A dialog prompt is a tuple of the house scan S, a target object t_o to be found, a starting position p_0, and a goal region G_j. We use the MatterPort object segmentations to get region

[1] A demonstration video of the data collection interface: https://youtu.be/BonlITv_PKw.

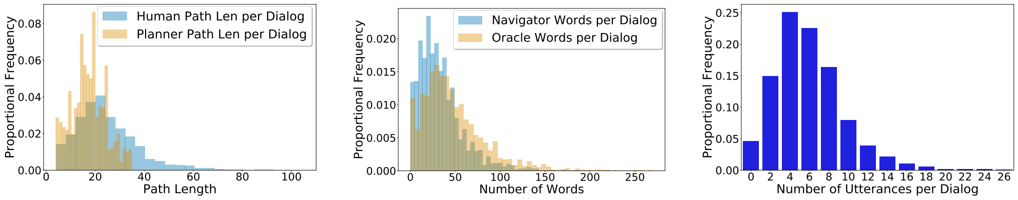

Figure 2: The distributions of steps taken by human *Navigator*s versus a shortest path planner (Left), the number of word tokens from the *Navigator* and the *Oracle* (Center), and the number of utterances in dialogs across the CVDN dataset.

locations for household objects, as in prior work [24]. We define a set of 81 unique object types that appear in at least 5 unique houses and appear between 2 and 4 times per such house.[2] Each dialog begins with a hint, such as "The goal room contains a *plant*," which by construction is both ambiguous (there are two to four rooms with a plant) and underspecified (the path to the room is not described by the hint).

Given a house scan S and a target object t_o, a dialog prompt is created for every goal region G_j in the house containing an instance of t_o. Goal regions are sets of nodes that occupy the same room in a house scan. The starting node p_0 is chosen to maximize the distance between p_0 and the goal regions $G_{0:|G|}$ containing t_o. Formally,

$$p_0 = \text{argmax}_{p \in S} \left(\sqrt{\sum_j \min_{p_i \in G_j} (d_h(p, p_i)^2)} \right).$$

Crowdsourced Data Collection. We gathered human-human dialogs through Amazon Mechanical Turk.[3] In each Human Intelligence Task (HIT), workers read about the roles of *Navigator* and *Oracle* and could practice using the navigation interface. Pairs of workers were connected to one another via a chat interface.

Every dialog was instantiated via a randomly chosen prompt (S, t_o, p_0, G_j), with the *Navigator* starting at panorama p_0 and both workers instructed via the text: "Hint: The goal room contains a t_o." The dialog begins with the *Navigator*'s turn. On the *Navigator*'s turn, they could navigate, type a natural language question to ask the *Oracle*, or guess that they had found the goal room. Incorrect guesses disabled further navigation and forced the *Navigator* to ask a question to the *Oracle*. Throughout navigation, the *Oracle* was shown the steps being taken as a mirror of the *Navigator*'s interface, so that both workers were always aware of the current visual frame. On the *Oracle*'s turn, they could view an animation depicting the next 5 hops through the navigation graph towards the goal room according to a shortest path planner and communicate back to the *Navigator* via natural language (Figure 1). Five hops was chosen because this is slightly shorter than the 6 hop average path in the R2R dataset, for which human annotators were able to provide reasonable language descriptions. Each HIT paid \$1.25 per worker, the entire dataset collection cost over \$7k.

After successfully locating the goal room, workers rated their partner's cooperativeness (from 1 to 5). Workers who failed to maintain a 4 or higher average peer rating were disallowed from taking more of our HITs. On average, dialog participants' mean peer rating is 4.52 out of 5 across CVDN.

Analysis. The CVDN dataset has longer routes and language contexts than the R2R task. The dialogs exhibit complex phenomena that require both dialog and navigation history to resolve.

Figure 2 shows the distributions of path lengths, word counts, and number of utterances across dialogs in the CVDN dataset. Human (25.0 ± 12.9) and planner (17.4 ± 7.0) path lengths are on average more than three times longer, and have higher variance, than the path lengths in R2R (6.0 ± 0.85). Average word counts for navigators (33.5) and oracles (48.1) sum to an average 81.6 words per dialog, again exceeding the Room-to-Room average of 29 words per instruction by nearly

[2] We also cut odd ("soffet") and non-specific ("wall") objects, and merge similar object names (e.g., "potted plant" and "plant") to cut down the initial 929 object types to these salient 81. Some houses do not have objects that meet our criteria, so CVDN represents only 83 of the 90 total MatterPort houses.

[3] `https://cvdn.dev/`. Connect with two tabs to start a dialog with yourself.

	Dia	***Nav***	***Ora***	**Example**
Ego	92.5	52.9	65.8	*Oracle*: Turn slightly to your right and go forward down the hallway
Needs Q	13.0	-	3.9	*Navigator*: Should I turn left down the hallway ahead? *Oracle*: ya
Needs Dialog History	3.5	0.4	1.0	*Oracle*: Through the lobby. So go through the door next to the green towel. Go to the left door next to the two yellow lights. Walk straight to the end of the hallway and stop . . . *Navigator*: Are these the yellow lights you were talking about?
Needs Nav History	14.0	1.5	3.4	*Oracle*: You were there briefly but left. There is a turntable behind you a bit. Enter the bedroom next to it.
Repair	12.5	1.6	3.4	*Oracle*: I am so sorry I meant for you to look over to the right not the left
Off-topic	3.0	5.4	5.1	*Navigator*: I am to the 'rear' of the zebra. Nice one. *Oracle*: Ok hold your nose and go to the left of the zebra, through the livingroom and kitchen and towards the bedroom you can see past that
Vacuous	6.0	22.7	2.3	*Navigator*: Ok, now where?

Table 2: The average percent of *Dialogs*, as well as individual *Navigator* and *Oracle* utterances, exhibiting each phenomena out of 100 hand-annotated dialogs. Two authors annotated each dialog and reached an agreement of Cohen's $\kappa = .738$ across all phenomena labels.

a factor of three. Dialogs average about 6 utterances each (3 question and answer exchanges), with a fraction being much longer—up to 26 utterances. Some dialogs have no exchanges (about 5%): the *Navigator* was able to find the goal location by intuition alone given the hint. Because more than one room always contains t_o, these are 'lucky' guesses.

We randomly sampled 100 dialogs with at least one QA exchange and annotated whether each utterance (out of 342 per speaker) exhibited certain phenomena (Table 2). Over half the utterances from both *Navigator* and *Oracle* roles, and over 90% of all dialogs, contain egocentric references requiring the agent's position and orientation to interpret. Some *Oracle* answers require the *Navigator* question to resolve (e.g., when the answer is just a confirmation). Some utterances need dialog history from previous exchanges or past visual navigation information. More than 10% of dialogs exhibit conversational repair, when speakers try to rectify mistakes. Speakers sometimes establish rapport with off-topic comments and jokes. Both speakers, especially those in the *Navigator* role, sometimes send vacuous communications, but this is limited to a smaller percentage of dialogs.

Models attempting to perform navigation, ask questions, or answer questions about an embodied environment must grapple with these types of phenomena. For example, an agent may need to attend not just to the last QA exchange, but to the entire dialog and navigation history in order to correctly follow instructions.

4 The Navigation from Dialog History Task

CVDN facilities training agents for navigation, question asking, and question answering. In this paper, we focus on navigation. The ability to navigate successfully given dialog history is key to any future work in the Vision-and-Dialog Navigation paradigm. Every dialog is a sequence of *Navigator* question and *Oracle* answer exchanges, with *Navigator* steps following each exchange. We use this structure to divide dialogs into Navigation from Dialog History (NDH) instances.

In particular, CVDN instances are each comprised of a repeating sequence $< N_0, Q_1, A_1, N_1, \ldots, Q_k, A_k, N_k >$ of navigation actions, N, questions asked by the *Navigator*, Q, and answers from the *Oracle*, A. Because sending a question or answer ends the worker's turn, every question Q_i and answer A_i is a single string of tokens. For each dialog with prompt (S, t_o, p_0, G_j), an NDH instance is created for each of $0 \leq i \leq k$. The input is t_o and a (possibly empty) history of questions and answers $(Q_{1:i}, A_{1:i})$. The task is to predict navigation actions that bring the agent closer to the goal location G_j, starting from the terminal node of N_{i-1} (or p_0, for N_0). We extract 7415 NDH instances from the 2050 navigation dialogs in CVDN.

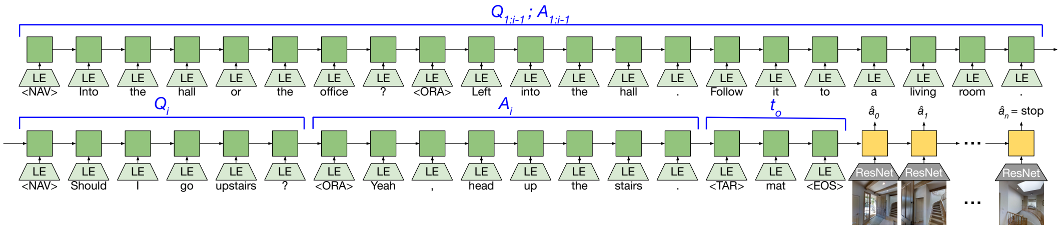
$Q_{1:i-1}$; $A_{1:i-1}$
LE
<NAV> Into the hall or the office ? <ORA> Left into the hall . Follow it to a living room .
Q_i
A_i
t_o
<NAV> Should I go upstairs ? <ORA> Yeah , head up the stairs . <TAR> mat <EOS>
$\hat{a}_0$
$\hat{a}_1$
$\hat{a}_n$ = stop
ResNet
...

Figure 3: We use a sequence-to-sequence model with an LSTM encoder that takes in learnable token embeddings (LE) of the dialog history. The encoder conditions an LSTM decoder for predicting navigation actions that takes in fixed ResNet embeddings of visual environment frames. Here, we demarcate subsequences in the input (e.g., t_o) compared during input ablations.

We divide these instances into training, validation, and test folds, preserving the R2R folds by house scan. This division is further done by dialog, such that for every dialog in CVDN the NDH instances created from it all belong to the same fold. As in R2R, we split the validation fold into *seen* and *unseen* house scans, depending on whether the scan is present in the training set. This results in 4742 training, 382 *seen* validation, 907 *unseen* validation, and 1384 *unseen* test instances.

We provide two forms of supervision for the NDH task: N_i, the navigation steps taken by the *Navigator* after question-answer exchange i, and O_i, the shortest-path steps shown to the *Oracle* and used as context to provide answer A_i. In each instance of the task, i indexes the QA exchange in the dialog from which the instance is drawn (with $i = 0$ an empty QA followed by initial navigation steps). Across NDH instances, the N_i steps range in length from 1 to 40 (average 6.63), and the O_i steps range in length from 0 to 5 (average 4.35). The *Navigator* often continues farther than what the *Oracle* describes, using their intuition about the house layout to seek the target object.

We evaluate performance on this task by measuring how much progress the agent makes towards G_j. Let $e(P)$ be the end node of path P, $b(P)$ the beginning, and $\hat{P}$ the path inferred by the navigation agent. Then the progress towards the goal is defined as the reduction (in meters) from the distance to the goal region G_j at $b(\hat{P})$ versus at $e(\hat{P})$. Because G_j is a set of nodes, we take the minimum distance $\min_{p \in G_j}(d_m(p, q))$ as the distance between q and region G_j. Note that this is a topological distance (e.g., we measure the distance around a wall, rather than straight through it).

5 Experiments

Anderson et al. [7] introduced a sequence-to-sequence model to serve as a learning baseline in the R2R task. We formulate a similar model to encode an entire dialog history, rather than a single navigation instruction, as an initial learning baseline for the NDH task. The dialog history is encoded using an LSTM and used to initialize the hidden state of an LSTM decoder whose observations are visual frames from the environment, and whose outputs are actions in the environment (Figure 3).

We replace words that occur fewer than 5 times with an `UNK` token. The resulting vocabulary sizes are 1042 language tokens in the training fold and 1181 tokens in the combined training and validation folds. We also use special `NAV` and `ORA` tokens to preface a speaker's tokens, `TAR` to preface the target object token, and `EOS` to indicate the end of the input sequence. During training, an embedding is learned for every token and given as input to the encoder LSTM. For visual features, we embed the visual frame as the penultimate layer of an Imagenet-pretrained ResNet-152 model [31].

When evaluating against the validation folds, we train only on the training fold. When evaluating against the test fold, we train on the union of the training and validation folds. We ablate the distance of dialog history encoded, and introduce a mixed planner and human supervision strategy at training time. We hypothesize both that encoding a longer dialog history and using mixed-supervision steps will increase the amount the agent progresses towards the goal.

Training. Given supervision from an end node $e(P^*)$, the agent infers navigation actions to form path $\hat{P}$. We train all agents with student-forcing for 20000 iterations of batch size 100, and evaluate validation performance every 100 iterations (see the Appendix for details). The best performance across all epochs is reported for validation folds. At each timestep the agent executes its inferred action $\hat{a}$, and is trained using cross entropy loss against the action a^* that is next along the shortest

		Seq-2-Seq Inputs					Goal Progress (m) ↑		
Fold		V	t_o	A_i	Q_i	$Q_{1:i-1}$ $A_{1:i-1}$	**Oracle**	**Navigator**	**Mixed**
Val (Seen)	Baselines	`Shortest Path Agent`					8.29	7.63	9.52
		`Random Agent`					0.42	0.42	0.42
							0.59	0.83	0.91
		✓					4.12	5.58	5.72
			✓	✓	✓	✓	1.41	1.43	1.58
	Ours	✓	✓				4.16	**5.71**	5.71
		✓	✓	✓			4.34	5.61	6.04
		✓	✓	✓	✓		4.28	5.58	**6.16**
		✓	✓	✓	✓	✓	**4.48**	5.67	5.92
Val (Unseen)	Baselines	`Shortest Path Agent`					8.36	7.99	9.58
		`Random Agent`					1.09	1.09	1.09
							0.69	1.32	1.07
		✓					0.85	1.38	1.15
			✓	✓	✓	✓	1.68	1.39	1.64
	Ours	✓	✓				0.74	1.33	1.29
		✓	✓	✓			1.14	1.62	2.05
		✓	✓	✓	✓		1.11	1.70	1.83
		✓	✓	✓	✓	✓	**1.23**	**1.98**	**2.10**
Test (Unseen)	Baselines	`Shortest Path Agent`					8.06	8.48	9.76
		`Random Agent`					0.83	0.83	0.83
							0.13	0.80	0.52
		✓					0.99	1.56	1.74
			✓	✓	✓	✓	1.51	1.20	1.40
	Ours	✓	✓				1.05	1.81	1.90
		✓	✓	✓			1.21	2.01	2.05
		✓	✓	✓	✓		**1.35**	1.78	2.27
		✓	✓	✓	✓	✓	1.25	**2.11**	**2.35**

Table 3: Average agent progress towards the goal location when trained using different path end nodes for supervision. Among sequence-to-sequence ablations, **bold** indicates most progress across available language input, and **blue** indicates most progress across supervision signals.

path to the end node $e(P^*)$. Using the whole navigation path, P^*, as supervision rather than only the end node has been considered in other work [32]. At test time, the agents are trained up to the epoch that achieved the best performance on the *unseen* validation fold and then evaluated (e.g., test fold evaluations are run only *once* per agent).

Recall that for each NDH instance, the path shown to the *Oracle* during QA exchange i, O_i, and the path taken by the *Navigator* after that exchange, N_i, are given. We define the mixed supervision path M_i as N_i when $e(O_i) \in N_i$, and O_i otherwise. This new form of supervision has parallels to previous works on learning from imperfect or adversarial human demonstrations. One common solution is to use imperfect human demonstrations to learn an initial policy which is then refined with Reinforcement Learning (RL) [33]. Learning performance can be improved by first assigning a confidence measure to the demonstrations and only including those demonstrations that pass a certain threshold [34]. While we leave the evaluation of more sophisticated RL methods to future work, the mixed supervision described above can be thought of as using a simple binary confidence heuristic to threshold the human demonstrations.

Baselines and Ablations. We compare the sequence-to-sequence agent to a full-state information shortest path agent, to a non-learning baseline, and to unimodal baselines. The `Shortest Path` agent takes the shortest path to the supervision goal at inference time, and represents the best a learning agent could do under a given form of supervision. The non-learning `Random` agent chooses a random heading and walks up to 5 steps forward (as in [7]). Random baselines can be outperformed by unimodal model ablations—agents that consider only visual input, only language input, or neither—on VLN tasks [35]. So, we also compare our agent to unimodal baselines where agents have zeroed out visual features in place of the V ResNet features at each decoder timestep (vision-less baseline) and/or empty language inputs to the encoder (language-less baseline). To examine the impact of dialog history, we consider agents with access to the target object t_o; the last *Oracle* answer A_i; the prefacing *Navigator* question Q_i; and the full dialog history (Figure 3).

Results. Table 3 shows agent performances given different forms of supervision. We ran paired t-tests between all model ablations within each supervision paradigm and across paradigms, and applied the Benjamini–Yekutieli procedure to control the false discovery rate (details in the Appendix).

Using all dialog history significantly outperforms unimodal ablations in *unseen* environments. The `Shortest Path` agent performance with *Navigator* supervision N_i approximates human performance on NDH, because $e(N_i)$ is the node reached by the human *Navigator* after QA exchange i during data collection. The sequence-to-sequence models establish an initial, multimodal baseline for NDH, with headroom remaining compared to human performance, especially in *unseen* environments. Using all dialog history, rather than just the last question or question-answer exchange, is needed to achieve statistically significantly better performance than using the target object alone in *unseen* test environments. This supports our hypothesis that dialog history is beneficial for understanding the context of the latest navigation instruction A_i. Models trained with mixed supervision always statistically significantly outperform those trained with oracle or navigator supervision. This supports our hypothesis that using human demonstrations only when they appear trustworthy increases agent progress towards the goal.

6 Conclusions and Future Work

We introduce Cooperative Vision-and-Dialog Navigation: 2050 human-human, situated navigation dialogs in a photorealistic, simulated environment. The dialogs contain complex phenomena that require egocentric visual grounding and referring to both dialog history and past navigation history for context. CVDN is a valuable resource for studying *in-situ* navigation interactions, and for training agents that both navigate human environments and ask questions when unsure, as well as those that provide verbal assistance to humans navigating in unfamiliar places.

We then define the Navigation from Dialog History task. Our evaluations show that dialog history is relevant for navigation agents to learn a mapping between dialog-based instructions and correct navigation actions. Further, we find that using a mixed form of both human and planner supervision combines the best of each: long-range exploration of an environment according to human intuition to find the goal, and short-range accuracy aligned with language input.

Limitations. The CVDN dataset builds on the Room-to-Room task in the MatterPort Simulator [7]. We would like to use CVDN to train real world agents for dialog and navigation. Simply fine-tuning on real world data may not be sufficient. Real-world robot navigation relies on laser scan depths, not just RGB information, and invokes lower quality egocentric vision, sensor noise, and localization issues. While the simulation provides photorealistic environments, it suffers from discrete, graph-based navigation, requiring a real world navigable environment to be mapped and divided into topological waypoints. Human-human dialogs collected in high-fidelity, continuous motion simulators (e.g., [36]) or using virtual reality technology may facilitate easier transfer to physical robot platforms. However, sharing a simulation environment with the existing R2R task means that models for dialog history tasks like NDH may benefit from pretraining on R2R.

Future Work. The sequence-to-sequence model used in our experiments serves as an initial learning baseline for the NDH task. Moving forward, by formulating NDH as a sequential decision process we can use RL to shape the agent's policy, as in recent VLN work [37]. Dialog analysis also suggests that there is relevant information in the historical navigation actions which are not considered by the initial model. Jointly conditioning dialog and navigation history may help resolve past reference instructions like "Go back to the stairwell and go up one flight of steps," and could involve cross-modal attention alignment.

The CVDN dataset also provides a scaffold for navigation-centered question asking and question answering tasks. In our future work, we will explore training two agents in tandem: one to navigate and ask questions when lost, and another to answer those questions. This will facilitate end-to-end evaluation on CVDN, and will differ from all existing VLN tasks by involving two, trained agents engaged in task-oriented dialog.

Acknowledgments

This research was supported in part by the ARO (W911NF-16-1-0121) and the NSF (IIS-1252835, IIS-1562364). We thank the authors of Anderson et al. [7] for creating an extensible base for further research in VLN using the MatterPort 3D simulator, and our coworkers Yonatan Bisk, Mohit Shridhar, Ramya Korlakai Vinayak, and Aaron Walsman for helpful discussions and comments.

References

[1] S. Tellex, R. Knepper, A. Li, D. Rus, and N. Roy. Asking for help using inverse semantics. In *RSS*, 2014.

[2] J. Y. Chai, Q. Gao, L. She, S. Yang, S. Saba-Sadiya, and G. Xu. Language to action: Towards interactive task learning with physical agents. In *IJCAI*, 2018.

[3] J. Thomason, A. Padmakumar, J. Sinapov, N. Walker, Y. Jiang, H. Yedidsion, J. Hart, P. Stone, and R. J. Mooney. Improving grounded natural language understanding through human-robot dialog. In *ICRA*, 2019.

[4] M. Murnane, M. Breitmeyer, F. Ferraro, C. Matuszek, and D. Engel. Learning from human-robot interactions in modeled scenes. In *SIGGRAPH*, 2019.

[5] T. Williams, F. Yazdani, P. Suresh, M. Scheutz, and M. Beetz. Dempster-shafer theoretic resolution of referential ambiguity. *Autonomous Robots*, 43(2):389–414, 2019.

[6] A. Chang, A. Dai, T. Funkhouser, M. Halber, M. Niessner, M. Savva, S. Song, A. Zeng, and Y. Zhang. Matterport3D: Learning from RGB-D data in indoor environments. *3DV*, 2017.

[7] P. Anderson, Q. Wu, D. Teney, J. Bruce, M. Johnson, N. Sünderhauf, I. Reid, S. Gould, and A. van den Hengel. Vision-and-Language Navigation: Interpreting visually-grounded navigation instructions in real environments. In *CVPR*, 2018.

[8] M. MacMahon, B. Stankiewicz, and B. Kuipers. Walk the talk: Connecting language, knowledge, and action in route instructions. In *AAAI*, 2006.

[9] D. L. Chen and R. J. Mooney. Learning to interpret natural language navigation instructions from observations. In *AAAI*, 2011.

[10] V. Blukis, D. Misra, R. A. Knepper, and Y. Artzi. Mapping navigation instructions to continuous control actions with position visitation prediction. In *CoRL*, 2018.

[11] H. Chen, A. Suhr, D. Misra, N. Snavely, and Y. Artzi. Touchdown: Natural language navigation and spatial reasoning in visual street environments. In *CVPR*, 2019.

[12] A. Das, S. Datta, G. Gkioxari, S. Lee, D. Parikh, and D. Batra. Embodied question answering. In *CVPR*, 2018.

[13] D. Gordon, A. Kembhavi, M. Rastegari, J. Redmon, D. Fox, and A. Farhadi. IQA: Visual question answering in interactive environments. In *CVPR*, 2018.

[14] J. Johnson, B. Hariharan, L. van der Maaten, L. Fei-Fei, C. Lawrence Zitnick, and R. Girshick. CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning. In *CVPR*, 2017.

[15] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh. VQA: Visual Question Answering. In *ICCV*, 2015.

[16] D. A. Hudson and C. D. Manning. GQA: A new dataset for real-world visual reasoning and compositional question answering. *CVPR*, 2019.

[17] R. Zellers, Y. Bisk, A. Farhadi, and Y. Choi. From recognition to cognition: Visual commonsense reasoning. In *CVPR*, 2019.

[18] S. Kottur, J. M. Moura, D. Parikh, D. Batra, and M. Rohrbach. CLEVR-Dialog: A diagnostic dataset for multi-round reasoning in visual dialog. In *NAACL*, 2019.

[19] A. Das, S. Kottur, K. Gupta, A. Singh, D. Yadav, J. M. Moura, D. Parikh, and D. Batra. Visual Dialog. In *CVPR*, 2017.

[20] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang. SQuAD: 100,000+ questions for machine comprehension of text. In *EMNLP*, 2016.

[21] E. Choi, H. He, M. Iyyer, M. Yatskar, S. Yih, Y. Choi, P. Liang, and L. Zettlemoyer. QuAC: Question answering in context. In *EMNLP*, 2018.

[22] M. Saeidi, M. Bartolo, P. Lewis, S. Singh, T. Rocktäschel, M. Sheldon, G. Bouchard, and S. Riedel. Interpretation of natural language rules in conversational machine reading. In *EMNLP*, 2018.

[23] S. Reddy, D. Chen, and C. D. Manning. CoQa: A conversational question answering challenge. *TACL*, 7, 2019.

[24] K. Nguyen, D. Dey, C. Brockett, and B. Dolan. Vision-based navigation with language-based assistance via imitation learning with indirect intervention. In *CVPR*, 2019.

[25] K. Nguyen and H. Daumé III. Help, Anna! Visual Navigation with Natural Multimodal Assistance via Retrospective Curiosity-Encouraging Imitation Learning. In *EMNLP*, 2019.

[26] H. de Vries, K. Shuster, D. Batra, D. Parikh, J. Weston, and D. Kiela. Talk the walk: Navigating new york city through grounded dialogue. *arXiv*, 2018.

[27] M. Marge, S. Nogar, C. Hayes, S. Lukin, J. Bloecker, E. Holder, and C. Voss. A research platform for multi-robot dialogue with humans. In *NAACL*, 2019.

[28] A. Anderson, M. Bader, E. Bard, Boyle, G. M. E., Doherty, S. Garrod, S. Isard, J. Kowtko, J. McAllister, J. Miller, C. Sotillo, H. S. Thompson, and R. Weinert. The HCRC Map Task Corpus. *Language and Speech*, 34:351–366, 1991.

[29] L. Stoia, D. M. Shockley, D. K. Byron, and E. Fosler-Lussier. SCARE: a Situated Corpus with Annotated Referring Expressions. In *LREC*, 2008.

[30] K. Eberhard, H. Nicholson, S. Kübler, S. Gunderson, and M. Scheutz. The Indiana "Cooperative Remote Search Task" (CReST) Corpus. In *LREC*, 2010.

[31] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[32] V. Jain, G. Magalhaes, A. Ku, A. Vaswani, E. Ie, and J. Baldridge. Stay on the path: Instruction fidelity in vision-and-language navigation. *ACL*, 2019.

[33] M. E. Taylor, H. B. Suay, and S. Chernova. Integrating reinforcement learning with human demonstrations of varying ability. In *AAMAS*, 2011.

[34] Z. Wang and M. E. Taylor. Improving reinforcement learning with confidence-based demonstrations. In *IJCAI*, 2017.

[35] J. Thomason, D. Gordon, and Y. Bisk. Shifting the Baseline: Single Modality Performance on Visual Navigation & QA. In *NAACL*, 2019.

[36] E. Kolve, R. Mottaghi, W. Han, E. VanderBilt, L. Weihs, A. Herrasti, D. Gordon, Y. Zhu, A. Gupta, and A. Farhadi. AI2-THOR: An Interactive 3D Environment for Visual AI. *arXiv*, 2017.

[37] H. Tan, L. Yu, and M. Bansal. Learning to navigate unseen environments: Back translation with environmental dropout. In *NAACL*, 2019.

7 Appendix

7.1 Additional CVDN Analysis

Figure 4 gives the distributions of target objects t_o across the dialogs in CVDN. The most frequent objects are those that are both frequent across houses and typically number between 2 and 4 per house, and often have a one-to-one correspondence with bedrooms and bathrooms.

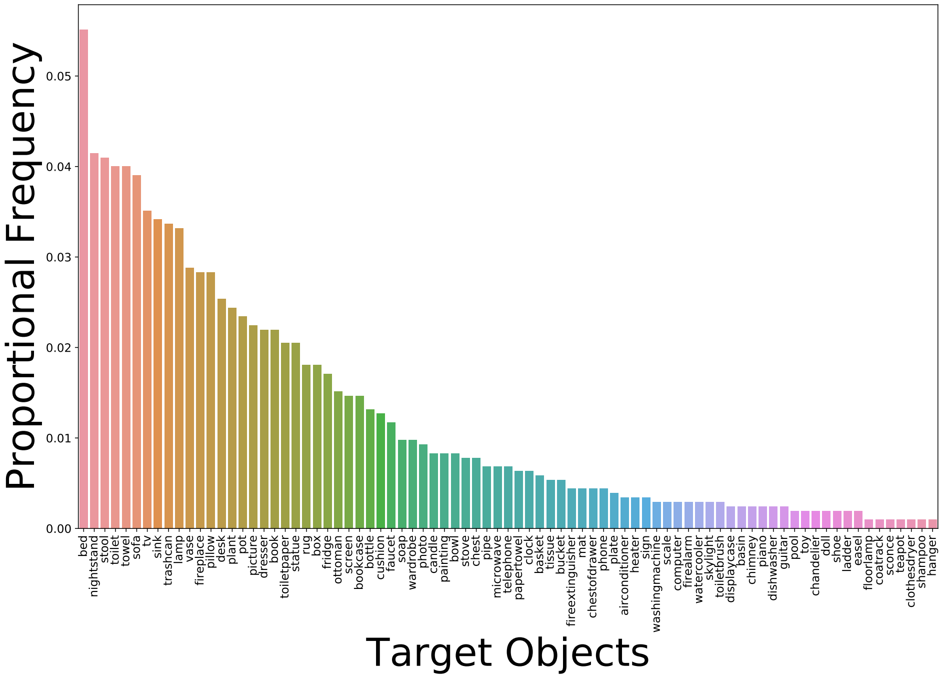

Figure 4: The distribution of the 81 target objects t_o in dialogs across CVDN.

Figure 5 gives the intersection-over-union (IoU) of paths in CVDN within the same scan, comparing them against those in R2R and human performance per-dialog. The average path IoU across a scan is the average number of navigation nodes in the intersection of two paths over the union of nodes in those paths, across all paths in the scan. Compared to R2R, the paths in the dialogs of CVDN share more navigation nodes per scan because of the way starting panoramas p_0 were chosen—to maximize the distance to potential goal regions. Many CVDN paths start at or near the same remote p_0 nodes in, e.g., basements, rooftops, and lawns. Per-dialog, we measure the IoU between human *Navigator* and shortest path planner trajectories and find that there is substantially more overlap than between two paths in the same scan, indicating that humans follow closer to the shortest path than to an average walk through the scan (e.g., they are not just memorizing previous dialog trajectories).

7.2 Additional NDH Analysis

Figure 6 gives path data for the NDH task. Compared to R2R, path lengths using shortest path supervision (O_i) are on average shorter than those in R2R, because paths shown to the *Oracle* were at most length 5. By contrast, human *Navigator* paths (N_i) are substantially longer than those seen in R2R. We also examine the distribution of the number of hops progressed towards the goal per NDH instance across *Oracle* shortest path, human navigator, and mixed supervision (M_i). While the planner always moves towards the goal (or stands still, if the *Navigator* is already in the goal region), human *Navigator*s sometimes move farther away from the goal, though in general make more progress than the planner. Using mixed supervision, fewer trajectories move "backwards"; the simple heuristic of whether a *Navigator* walked over the last node in the *Oracle*'s described shortest path shifts the distribution weight farther towards positive goal progress.

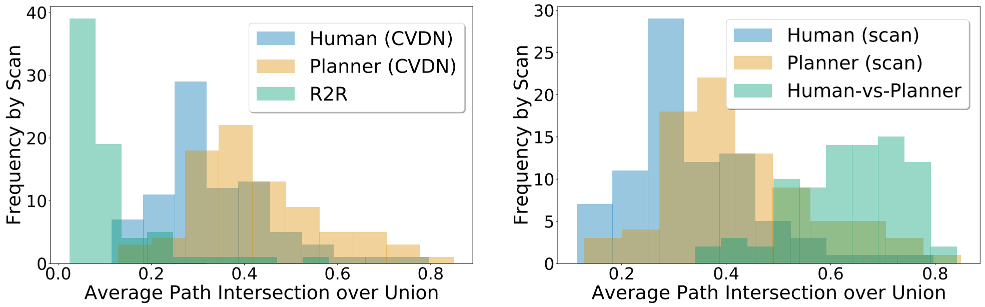

Figure 5: Left: The IoU of nodes in the paths of human *Navigator* and shortest path planner trajectories in CVDN versus those in R2R when comparing paths in the same scan. Right: The IoU of *Navigator* and shortest path planner trajectories in the same scan versus the IoU of player and shortest path planner trajectories across a dialog.

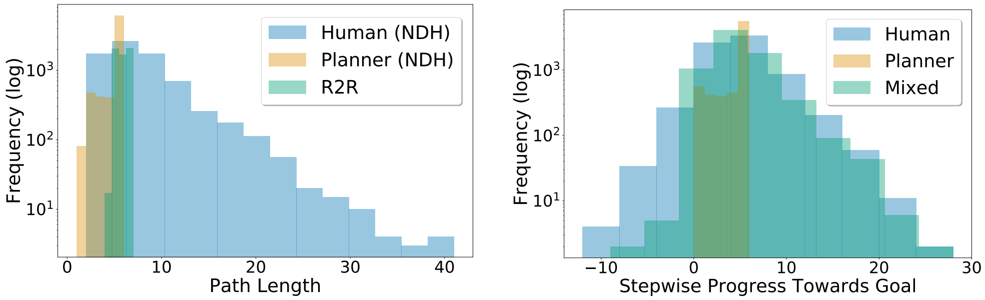

Figure 6: Left: The distributions of path lengths by human *Navigator* and the shortest path planner provided as supervision in NDH instances versus path lengths in R2R supervision. Right: The progress per NDH instance made towards the goal (in steps) by the human *Navigator*, the shortest path planner, and the mixed-supervision path.

7.3 NDH Model Performance Statistical Comparisons

We ran paired t-tests between all model ablations within each supervision paradigm (e.g., comparing all mixed supervision models to one another), and across paradigms (e.g., comparing the full dialog history model trained with mixed supervision to the one with navigator path supervision). Data pairs are NDH the distances progressed towards the goal on the same instance (i.e., dialog history and goal) between two conditions. This results in hundreds of comparison tests, so we apply a Benjamini–Yekutieli procedure to control the false discovery rate. Because the tests are not all independent, but some are, we estimate $c(m)$ under an arbitrary dependence assumption as $c(m) = \sum_{i+1}^{m} \frac{1}{i}$, where m is the number of tests run. We choose a significance threshold of $\alpha < 0.05$. Rather than report the hundreds of individual p-values, we highlight salient results below.

Different forms of supervision With one exception, in all environments (*seen* validation, *unseen* validation, and *unseen* test), across ablations of language context (i.e., full model using all history down to model using only the target object as dialog context), the differences in progress towards the goal under oracle, navigator, and mixed supervision are statistically significantly different. The only exception is the difference between oracle and navigator path supervision in *unseen* validation environments with the last answer only (i.e., row 16 of Table 3) ($p = 0.006$). Models trained with mixed supervision almost always achieve the most progress. For brevity, below we discuss further comparisons between models trained with mixed supervision.

Different amounts of dialog history In *unseen* validation and test environments, using all dialog history statistically significantly outperforms using only the target object, but not only the last answer ($p = 0.773$ in validation, $p = 0.035$ in test) or the last question-answer exchange ($p = 0.143$ in validation, $p = 0.560$ in test). Notably, in test environments, a statistically significant difference compared to using only the target object is observed *only* when using all dialog history. In *seen* validation houses, adding additional dialog history does not result in statistically significant gains, reflecting the representative power of the vision-only unimodal baseline.

Unimodal ablations In *unseen* validation and test environments, the model using all dialog history statistically significantly outperforms the unimodal baselines in all cases except the language-only unimodal model in the *unseen* validation houses ($p = 0.011$). In *seen* validation houses, this model statistically significantly outperforms the language-only and zero (no language, no vision) unimodal ablations. This result does not hold for the vision-only baseline, which is able to memorize the familiar houses for use at test time.

7.4 Sequence-to-Sequence Model Training

Hyperparameters. We use the training hyperparameters (optimizer, learning rate, hidden state sizes, etc.) presented in Anderson et al. [7] when training our sequence-to-sequence agents. We adjust the maximum input sequence length for language encoding based on the amount of dialog history available: 3 for t_o only (e.g., TAR tag, the target itself, and EOS); 70 for A_i; 120 for adding Q_i; and 720 (e.g., 120 times 6 turns of history) for $Q_{1:k}, A_{1:k}$. We increase the maximum episode length (e.g., the maximum number of navigation actions) depending on the supervision being used: 20 for oracle O_i (the same as in R2R) and 60 for navigator N_i and mixed M_i.

Teacher- versus Student-Forcing. We use student-forcing when training all of our sequence-to-sequence agents. Anderson et al. [7] found that student-forcing improved agent performance in unseen environments. Further, Thomason et al. [35] found that agents trained via teacher-forcing were outperformed by their unimodal ablations (i.e., they did not learn to incorporate both language and vision supervision, instead memorizing unimodal priors). Thus, we see no value in evaluating multi-modal agents trained via teacher-forcing in this setting.

Language Encoding. It is common in sequence-to-sequence architectures to reverse the input sequence of tokens during training, because the tokens relevant for the first decoding actions are likely also the first in the input sequence. Reversing the sequence means those relevant tokens have been seen more recently by the encoder, and this strategy was employed in prior work [7]. Following this intuition, we preserve the order of the dialog history during encoding, so that the most recent utterances are read just before decoding, but reverse the tokens at the utterance level (e.g., Q_i in Figure 3 is represented as sequence "<NAV> ? upstairs go I Should").

7.5 Naive Dialog History Encoding

We naively concatenated an encoded navigation history N_H (via an LSTM taking in ResNet embeddings of past navigation frames) to the encoded dialog history, then learned a feed-forward shrinking layer to initialize the decoder (Table 4). We hypothesize that there is some signal in this information, but we discover that naive concatenation does not improve performance in *seen* or *unseen* environments. We suspect that a modeling approach which learns an attention alignment between the navigation history and dialog history could make better use of the additional signal.

	Seq-2-Seq Inputs						**Goal Progress** (m) ↑		
Fold	N_H	V	t_o	A_i	Q_i	$Q_{1:i-1}$ $A_{1:i-1}$	**Oracle**	**Navigator**	**Mixed**
Val (Se)		Shortest Path Agent					8.29	7.63	9.52
		✓	✓	✓	✓	✓	4.48	5.67	5.92
	✓	✓	✓	✓	✓	✓	4.47	5.37	5.82
Val (Un)		Shortest Path Agent					8.36	7.99	9.58
		✓	✓	✓	✓	✓	1.23	1.98	2.10
	✓	✓	✓	✓	✓	✓	1.19	1.86	1.84

Table 4: Average sequence-to-sequence agent performance when the agent encodes the entire navigation history N_H compared against the Shortest Path upper bound and the agent encoding all dialog history across different supervision signals.