A Tasks and Data: Comparisons

Table 7 provides summary statistics comparing LANI and CHAI to existing related resources.

B Reward Function

LANI Following Misra et al. (2017), we use a shaped reward function that rewards the agent for moving towards the goal location. The reward for exampl i is:

$$R^{(i)}(s, a, s') = R_p^{(i)} + \phi^{(i)}(s) - \phi^{(i)}(s') \quad (1)$$

where s' is the origin state, a is the action, s is the target state, $R_p^{(i)}$ is the problem reward, and $\phi^{(i)}(s) - \phi^{(i)}$ is a shaping term. We use a potential-based shaping (Ng et al., 1999) that encourages the agent to both move and turn towards the goal. The potential function is:

$$\begin{aligned}\phi^{(i)}(s) = \quad & \delta \textsc{TurnDist}(s, s_g^{(i)}) \\ & +(1-\delta)\textsc{MoveDist}(s, s_g^{(i)}) \ ,\end{aligned}$$

where MOVEDIST is the euclidean distance to the goal normalized by the agent's forward movement distance, and TURNDIST is the angle the agent needs to turn to face the goal normalized by the agent's turn angle. We use δ as a gating term, which is 0 when the agent is near the goal and increases monotonically towards 1 the further the agent is from the goal. This decreases the sensitivity of the potential function to the TURNDIST term close to the goal. The problem reward $R_p^{(i)}$ provides a negative reward of up to -1 on collision with any object or boundary (based on the angle and magnitude of collision), a negative reward of -0.005 on every action to discourage long trajectories, a negative reward of -1 on an unsuccessful stop, when the distance to the goal location is greater than 5, and a positive reward of +1 on a successful stop.

CHAI We use a similar potential based reward function as LANI. Instead of rewarding the agent to move towards the final goal the model is rewarded for moving towards the next intermediate goal. We heuristically generate intermediate goals from the human demonstration by generating goals for objects to be interacted with, doors that the agent should enter, and the final position of the agent. The potential function is:

$$\begin{aligned}\phi^{(i)}(s) = \quad & \textsc{TurnDist}(s, s_{g,j}^{(i)}) + \\ & \textsc{MoveDist}(s, s_{g,j}^{(i)}) + \textsc{IntDist}(s, s_{g,j}^{(i)}) \ ,\end{aligned}$$

where $s_{g,j}^{(i)}$ is the next intermediate goal, TURNDIST rewards the agent for turning towards the goal, MOVEDIST rewards the agent for moving closer to the goal, and INTDIST rewards the agent for accomplishing the interaction in the intermediate goal. The goal is updated on being accomplished. Besides the potential term, we use a problem reward $R_p^{(i)}$ that gives a reward of 1 for stopping near a goal, -1 for colliding with obstacles, and -0.002 as a verbosity penalty for each step.

C Baseline Details

MISRA17 We use the model of Misra et al. (2017). The model uses a convolution neural network for encoding the visual observations, a recurrent neural network with LSTM units to encode the instruction, and a feed-forward network to generate actions using these encodings. The model is trained using policy gradient in a contextual bandit setting. We use the code provided by the authors.

CHAPLOT18 We use the gated attention architecture of Chaplot et al. (2018). The model is trained using policy gradient with generalized advantage estimation (Schulman et al., 2015). We use the code provided by the authors.

Our Approach with Joint Training We train the full model with policy gradient. We maximize the expected reward objective with entropy regularization. Given a sampled goal location $l_g \sim p(. \mid \bar{x}, \mathbf{I}_P)$ and a sampled action $a \sim p(. \mid l_g, (\mathbf{I}_1, p_1), \ldots, (\mathbf{I}_t, p_t))$, the update is:

$$\begin{aligned}\nabla J \quad \approx \quad & \{\nabla \log P(l_g \mid \bar{x}, \mathbf{I}_P) + \\ & \nabla \log P(a_t \mid l_g, (\mathbf{I}_1, p_1), \ldots, (\mathbf{I}_t, p_t))\} R(s_t, a) \\ & \lambda \nabla H(\pi(. \mid \tilde{s}_t) \ .\end{aligned}$$

We perform joint training with randomly initialized goal prediction and action generation models.

D Hyperparameters

For LANI experiments, we use 5% of the training data for tuning the hyperparameters and train on the remaining. For CHAI, we use the development set for tuning the hyperparameters. We train our models for 20 epochs and find the optimal stopping epoch using the tuning set. We use 32 dimensional embeddings for words and time. LSTM_x and LSTM_A are single layer LSTMs with 256 hidden units. The first layer of CNN_0 contains 128 8×8 kernels with a stride of 4 and padding 3, and the second layer contains 64 3×3 kernels with a stride of 1 and padding 1. The convolution layers in LINGUNET use 32 5×5 kernels with stride

Dataset	Num Instructions	Vocabulary Size	Mean Instruction Length	Num. Actions	Avg Trajectory Length	Partially Observed
Bisk et al. (2016)	16,767	1,426	15.27	81	15.4	No
MacMahon et al. (2006)	3,237	563	7.96	3	3.12	Yes
Matuszek et al. (2012b)	217	39	6.65	3	N/A	No
Misra et al. (2015)	469	775	48.7	>100	21.5	No
LANI	28,204	2,292	12.07	4	24.6	Yes
CHAI	13,729	1018	10.14	1028	54.5	Yes

Table 7: Comparison of LANI and CHAI to several existing natural language instructions corpora.

Category	Present	Absent	p-value
Spatial relations	2.56	1.77	.023
Location conjunction	3.85	1.93	.226
Temporal coordination	1.70	2.14	.164
Co-reference	1.98	1.98	.993

Table 8: Mean goal prediction error for CHAI instructions with and without the analysis categories we used in Table 2. The p-values are from two-sided t-tests comparing the means in each row.

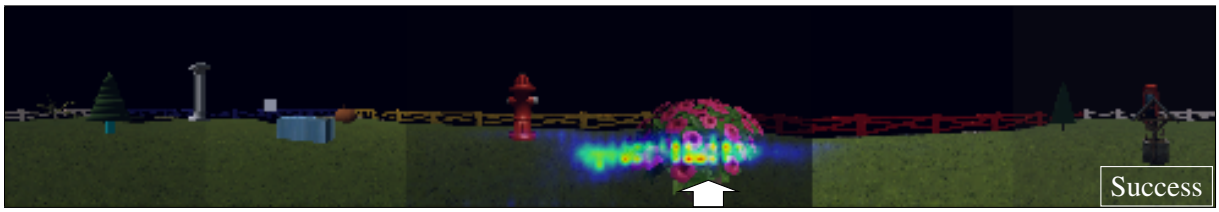
2. All deconvolutions except the final one, also use 32 5×5 kernels with stride 2. The dropout probability in LINGUNET is 0.5. The size of attention mask is 32×32 + 1. For both LANI and CHAI, we use a camera angle of 60° and create panoramas using 6 separate RGB images. Each image is of size 128×128. We use a learning rate of 0.00025 and entropy coefficient λ of 0.05.

E CHAI Error Analysis

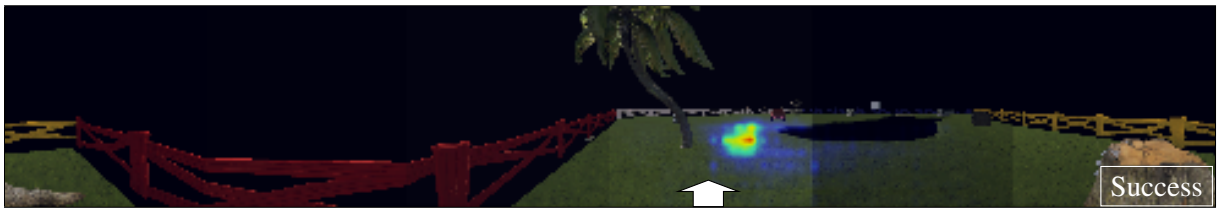
Table 8 provides the same kind of error analysis results here for the CHAI dataset as we produced for LANI, comparing performance of the model on samples of sentences with and without the analysis phenomena that occurred in CHAI.

F Examples of Generated Goal Prediction

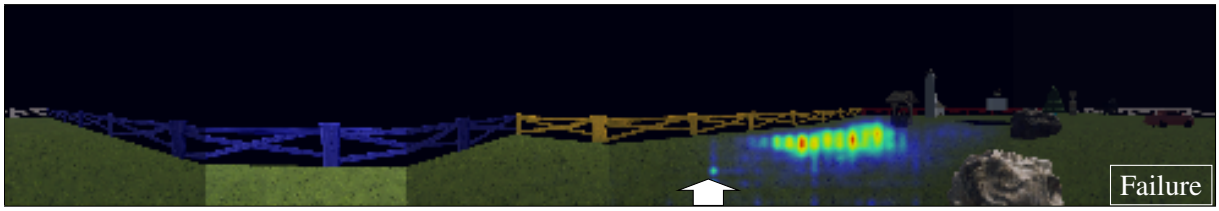
Figure 7 shows example goal predictions from the development sets. We found the predicted probability distributions to be reasonable even in many cases where the agent failed to successfully complete the task. We observed that often the evaluation metric is too strict for LANI instructions, especially in cases of instruction ambiguity.



go round the flowers



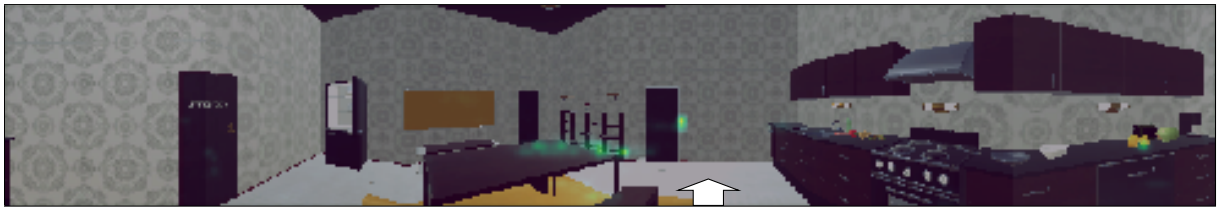
fly between the palm tree and pond



head toward the wishing well and keep it on your right .

move back to the kitchen .

then drop the tropicana onto the coffee table .



walk the cup to the table and set the cup on the table .

Figure 7: Goal prediction probability maps P_g overlaid on the corresponding observed panoramas $\mathbf{I}_P$. The top three examples show results from LANI, the bottom three from CHAI. The white arrow indicates the forward direction that the agent is facing. The success/failure in the LANI examples indicate if the task was completed accurately or not following the task completion (TC) metric.