TEACh: Task-driven Embodied Agents that Chat

Aishwarya Padmakumar 1, Jesse Thomason 2, Ayush Shrivastava 3, Patrick Lange 1, Anjali Narayan-Chen 1, Spandana Gella 1, Robinson Piramuthu 1, Gokhan Tur 1, Dilek-Hakkani Tur 1

1 Amazon Alexa AI
2 USC Viterbi Department of Computer Science, University of Southern California
3 Department of Electrical Engineering And Computer Science, University of Michigan

padmakua@amazon.com, jessedt@amazon.com, ayshrv@umich.edu, patlange@amazon.com, naraanja@amazon.com, sgella@amazon.com, robinpir@amazon.com, gokhatur@amazon.com, hakkanit@amazon.com

Abstract

Robots operating in human spaces must be able to engage in natural language interaction, both understanding and executing instructions, and using conversation to resolve ambiguity and correct mistakes. To study this, we introduce **TEACh**, a dataset of over 3,000 human–human, interactive dialogues to complete household tasks in simulation. A **Commander** with access to oracle information about a task communicates in natural language with a **Follower**. The **Follower** navigates through and interacts with the environment to complete tasks varying in complexity from **MAKE COFFEE** to **PREPARE BREAKFAST**, asking questions and getting additional information from the **Commander**. We propose three benchmarks using **TEACh** to study embodied intelligence challenges, and we evaluate initial models’ abilities in dialogue understanding, language grounding, and task execution.

1 Introduction

Many benchmarks for translating visual observations and an initial language instruction to actions assume no further language communication (Anderson et al. 2018; Shridhar et al. 2020). However, obtaining clarification via simulated interactions (Chi et al. 2020; Nguyen and Daumé III 2019) or learning from human-human dialogue (Thomason et al. 2019; Suhr et al. 2019) can improve embodied navigation. We hypothesize that dialogue has even more to offer for object-centric, hierarchical tasks.

We introduce **Task-driven Embodied Agents that Chat (TEACh)** to study how agents can learn to ground natural language (Harnad 1990; Bisk et al. 2020) to the visual world and actions, while considering long-term and intermediate goals, and using dialogue to communicate. **TEACh** contains over 3,000 human–human sessions interleaving utterances and environment actions where a **Commander** with oracle task and world knowledge and a **Follower** with the ability to interact with the world communicate in written English to complete household chores (Figure 1).

**TEACh** dialogues are unconstrained, not turn-based, yielding variation in instruction granularity, completeness, relevance, and overlap. Utterances include coreference with previously mentioned entities, past actions, and locations. Because **TEACh** sessions are human, rather than planner-based (Ghallab et al. 1998), **Follower** trajectories include mistakes and corresponding, language-guided correction.

We propose three benchmarks based on **TEACh** sessions to study the ability of learned models to achieve aspects of embodied intelligence: Execution from Dialog History (EDH), Trajectory from Dialog (TfD) and Two-Agent Task Completion (TATC) 1. We evaluate a baseline **Follower** agent for the EDH and TfD benchmarks based on the Episodic

---

*Authors contributed equally

Copyright © 2022, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

1 https://github.com/alexa/teach
Task demonstrations are created by the human Commander, who engages in a free-form, rather than turn-taking, dialogue with the human Follower. Compared to past dialogue datasets for visual tasks, TEACh contains many more individual dialogues.

The TEACh dataset contains many more individual dialogues.

### Table 1: TEACh is the first dataset where human-human, conversational dialogues were used to perform tasks involving object interaction, such as picking up a knife, and state changes, such as slicing bread, in a visual simulation environment. TEACh task demonstrations are created by the human Follower, who engages in a free-form, rather than turn-taking, dialogue with the human Commander. Compared to past dialogue datasets for visual tasks, TEACh contains many more individual dialogues.
Figure 2: To collect TEACh, the Commander knows the task to be completed and can query the simulator for object locations. Searched items are highlighted in green for the Commander; highlights blink to enable seeing the underlying true scene colors. The Commander has a top-down map of the scene, with the current camera position shown in red, the Follower position shown in blue, and the object search camera position shown in yellow. The Follower moves around in the environment and interacts with objects, such as placing a fork (middle). Target objects for each interaction action are highlighted.

criteria. For example, for a task to make coffee, we consider the environment to be in a successful state if there is a mug in the environment that is clean and filled with coffee.

Parameterized tasks such as PUT ALL X ON Y enable task variation. Parameters can be object classes, such as putting all forks on a countertop, or predefined abstract hypernyms, for example putting all silverware—forks, spoons, and knives—on the counter. TEACh task definitions are also hierarchical. For example, PREPARE BREAKFAST contains the subtasks MAKE COFFEE and MAKE PLATE OF TOAST. We incorporate determiners such as “a”, “all” and numbers such as 2 to enable easy definition of a wide range of tasks, such as N SLICES OF X IN Y. The TEACh TDL includes template-based language prompts to describe tasks and subtasks to Commanders (Figure 3).

3.2 Gameplay Session Collection

Annotators first completed a tutorial task demonstrating the interface to vet their understanding. For each session, two vetted crowdworkers were paired using a web interface and assigned to the Commander and Follower roles (Figure 2). The Commander is shown the task to be completed and the steps needed to achieve this given the current state of the environment, using template-based language prompts, none of which are accessible to the Follower. The Commander can additionally search for the location of objects, either by string name, such as “sink”, or by clicking a task-relevant object in the display (Figure 3). The Commander and Follower must use text chat to communicate the parameters of the task and clarify object locations. Only the Follower can interact with objects in the environment.

We obtained initial states for each parameterized task by randomizing AI2-THOR environments and retaining those that satisfied preconditions such as task-relevant objects being present and reachable. For each session, we store the initial simulator state \( S_i \), the sequence of actions \( A = (a_1, a_2, \ldots) \) taken, and the final simulator state \( S_f \). TEACh Follower actions are Forward, Backward, Turn Left, Turn Right, Look Up, Look Down, Strafe Left, Strafe Right, Pickup, Place, Open, Close, ToggleOn, ToggleOff, Slice, and Pour. Navigation actions move the agent in discrete steps. Object manipulation expects the agent to specify an object via a relative coordinate \((x, y)\) on Follower egocentric frame. The TEACh wrapper on the AI2-THOR simulator examines the ground truth segmentation mask of the agent’s egocentric image, selects an object in a 10x10 pixel patch around the coordinate if the desired action can be performed on it, and executes the action in AI2-THOR. The Commander can execute a Progress Check and SearchObject actions, demonstrated in Figure 3. TEACh Commander actions also allow navigation, but the Commander is a disembodied camera.

3.3 TEACh Statistics

TEACh is comprised of 3,047 successful gameplay sessions, each of which can be replayed using the AI2-THOR simulator for model training, feature extraction, or model evaluation. In total, 4,365 crowdsourced sessions were collected with a human-level success rate of 74.17% (3320 sessions) and total cost of $105k; more details in appendix. Some successful sessions were not included in the final split used in benchmarks due to replay issues. TEACh sessions span all 30 AI2-THOR kitchens, and include most of the 30 each AI2-THOR living rooms, bedrooms, and bathrooms.

Successful TEACh sessions consist of over 45k utterances, with an average of 8.40 Commander and 5.25 Follower utterances per session. The average Commander utterance length is 5.70 tokens and the average Follower utterance length is 3.80 tokens. The TEACh data has a vocabulary size of 3,429 unique tokens.\(^2\) Table 2 summarizes such metrics across the 12 task types in TEACh. Simple tasks like MAKE COFFEE require fewer dialogue acts and Follower actions on average than complex, composite tasks like PREPARE BREAKFAST which subsume those simpler tasks.

\(^2\)Using the spaCy tokenizer: https://pypi.org/project/spacy/
4 TEACH Benchmarks

We introduce three benchmarks based on TEACH sessions to train and evaluate the ability of embodied AI models to complete household tasks using natural language dialogue. **Execution from Dialogue History** and **Trajectory from Dialogue** require modeling the Follower. **Two-Agent Task Completion**, by contrast, requires modeling both the Commander and Follower agents to complete TEACH tasks end-to-end. For each benchmark, we define how we derive benchmark instances from TEACH gameplay sessions, and by what metrics we evaluate model performance.

Each session has an initial state $S_i$, the sequence of actions $A = (a_1, a_2, \ldots)$ taken by the Commander and Follower including dialogue and environment actions, and the final state $S_f$. We denote the subsequence of all dialogue actions as $A^D$, and of all navigation and interaction as $A^I$. Following ALFRED, we create validation and test splits in both seen and unseen environments (Table 3). Seen splits contain sessions based in AI2-THOR rooms that were seen during training, whereas unseen splits contain only sessions in rooms absent from the training set.

4.1 Execution from Dialogue History (EDH)

We segment TEACH sessions into EDH instances. We construct EDH instances $(S^E, A^I, A^D, F^E)$ where $S^E$ is the initial state of the EDH instance, $A^I$ is an action history, and the agent is tasked with predicting a sequence of actions that changes the environment state to $F^E$, using $A^D$ reference interaction actions taken in the session as supervision. We constrain instances to have $|A^D| > 0$ and at least one object interaction in $A^I$. Each EDH instance is punctuated by a dialogue act starting a new instance or the session end. We append a `Stop` action to each $A^R$. An example is included in Figure 4.

To evaluate inferred EDH action sequences, we compare the simulator state changes $\hat{E}$ at the end of inference with $F^E$ using similar evaluation criteria generalized from the ALFRED benchmark.

- Success $\{0, 1\}$: 1 if all expected state changes $F^E$ are present in $\hat{E}$, else 0. We average over all trajectories.
- Goal-Condition Success (GC) $(0, 1)$: The fraction of expected state changes in $F^E$ present in $\hat{E}$. We average over all trajectories.\(^3\)
- Trajectory Weighted Metrics: For a reference trajectory $A^R_i$ and inferred action sequence $A^I$, we calculate trajectory length weighted metric for metric value $m$

\[
TLW-m = \frac{m \ast |A^I|}{\max(|A^R_i|, |A^I|)}
\]

During inference, the learned Follower agent predicts actions until either it predicts the `Stop` action, hits a limit of 1000 steps, or hits a limit of 30 failed actions.

4.2 Trajectory from Dialogue (TID)

A Follower agent model is tasked with inferring the whole sequence of Follower environmental actions taken during the session conditioned on the dialogue history. A TID instance is $(S_i, A^D, A^R, S_f)$, where $A^D$ is all dialogue actions taken by both agents, and $A^R$ is all non-dialogue actions taken by the Follower. We append a `Stop` action to $A^R$. The agent does not observe dialogue actions in context, however, we use this task to test long horizon action

\(^3\)We follow ALFRED in using a macro-, rather than micro-average for Goal-Conditioned Success Rate.
parameters and can require more actions per session to finish. Composite tasks like P

\[ \hat{E} \]

challenges the TATC benchmark represents studying the “whole” set of model can communicate back via language generation. The TEACh Table 2: The 12 tasks represented in TEACh sessions vary in complexity. Tasks like PUT ALL X ON Y take object class parameters and can require more actions per session to finish. Composite tasks like PREPARE SALAD contain sub-tasks like N SLICES OF X IN Y. Per session data are averages with standard deviation across task types.

<table>
<thead>
<tr>
<th>Fold</th>
<th>Split</th>
<th># Sessions</th>
<th># EDH Instances</th>
</tr>
</thead>
<tbody>
<tr>
<td>Train</td>
<td>Seen</td>
<td>1482 (49%)</td>
<td>5758 (49%)</td>
</tr>
<tr>
<td></td>
<td>Unseen</td>
<td>614 (20%)</td>
<td>2188 (19%)</td>
</tr>
<tr>
<td>Val</td>
<td>Seen</td>
<td>181 (6%)</td>
<td>654 (5%)</td>
</tr>
<tr>
<td></td>
<td>Unseen</td>
<td>614 (20%)</td>
<td>2188 (19%)</td>
</tr>
<tr>
<td>Test</td>
<td>Seen</td>
<td>181 (6%)</td>
<td>654 (5%)</td>
</tr>
<tr>
<td></td>
<td>Unseen</td>
<td>589 (19%)</td>
<td>2370 (20%)</td>
</tr>
</tbody>
</table>

Table 3: Session and EDH instances in TEACh data splits.

Two-Agent Task Completion (TATC)

To explore modeling both a Commander and Follower agent, the TATC benchmark gives as input only environment observations to both agents. The Commander model must use the Progress Check action to receive task information, then synthesize that information piece by piece to the Follower agent via language generation. The Follower model can communicate back via language generation. The TATC benchmark represents studying the “whole” set of challenges the TEACh dataset provides. We calculate success and goal-conditioned success by comparing \( \hat{E} \) against state changes between \( S_i \) and \( S_f \).

4.3 Two-Agent Task Completion (TATC)

Model. We establish baseline performance for the EDH and TID tasks using the Episodic Transformer (E.T.) model (Paschevich, Schmid, and Sun 2021), designed for the ALFRED benchmark. The original E.T. model trains a transformer language encoder and uses a ResNet-50 backbone to encode visual observations. Two multimodal transformer layers are used to fuse information from the language, image, and action embeddings, followed by a fully connected layer to predict the next action and target object category for interaction actions. E.T. uses a MaskRRCNN (He et al. 2017) model pretrained on ALFRED images to predict a segmentation of the egocentric image for interactive actions, matching the predicted mask to the predicted object category. We convert the centroid of this mask to a relative coordinate specified to the TEACh API wrapper for AI2-THOR.

We modify E.T. by learning a new action prediction head to match TEACh Follower actions. Given an EDH or TID instance, we extract all dialogue utterances from the action history \( A_H^I \) and concatenate these with a separator between utterances to form the language input. The remaining actions \( A_H^I \) are fed in order as the past action input with associated image observations. Consequently, our adapted E.T. does not have temporal alignment between dialogue actions and environment actions.

Following the mechanism used in the original E.T. paper, we provide image observations from both actions in the history \( A_H^I \) and the reference actions \( A_R^I \), and task the model to predict the entire sequence of actions. The model parameters are optimized using cross entropy loss between the predicted action sequence and the ground truth action sequence. For EDH, we ablate a history loss \( H \) as cross entropy over the entire action sequence—actions in both \( A_H^I \) and \( A_R^I \) to compare against loss only against actions in \( A_H^I \). Note that in TID, \( |A_H^I| = 0 \).

We additionally experiment with initializing the model using weights trained on the ALFRED dataset. Note that since the language vocabulary and action space change,

5 Experiments and Results

We implement initial baseline models and establish the richness of TEACh data and difficulty of resulting benchmarks.

5.1 Follower Models for EDH and TID

We use a single model architecture to train and evaluate on the EDH and TID benchmark tasks.
some layers need to be retrained. For EDH, we experiment with initializing the model both with weights from the E.T. model trained only on base ALFRED annotations (A) and the model trained on ALFRED augmented with synthetic instructions (S) (from Pashevich, Schmid, and Sun (2021)). We also perform unimodal ablations of the E.T. model to determine whether the model is simply memorizing sequences from the training data (Thomason, Gordon, and Bisk 2018).

At inference time, the agent uses dialogue history as language input, and the environment actions in $A'_{IL}$ as past action input along with their associated visual observations. At each time step we execute the predicted action, with predicted object coordinate when applicable, in the simulator. The predicted action and resulting image observation are added to agent’s input for the next timestep. The appendix details model hyperparameters.

**Results.** Table 4 summarizes our adapted E.T. model performance on the EDH and TfD benchmarks.

We observe that all E.T. model conditions in EDH are significantly better than Random and Lang-Only condition on all splits on SR and GC, according to a paired two-sided Welch $t$-test with Bonferroni corrections. Compared to the Vision-Only baseline, the improvements of the E.T. models are statistically significant on unseen splits, but not on seen splits. Qualitatively, we observe that the Random baseline only succeeds on very short EDH instances that only include one object manipulation involving a large target object, for example placing an object on a countertop. The same is true of most of the successful trajectories of the Lang-Only baseline. The success rate of the Vision-Only baseline suggests that the E.T.-based models are not getting much purchase with language signal. Notably, E.T. performs well below its success rates on ALFRED, where it achieves 38.24% on the ALFRED test-seen split and 8.57% on the ALFRED test-unseen split. Additionally, although there appears to be a small benefit from initializing the E.T. model with pretrained weights from ALFRED, these differences are not statistically significant. TEACH language is more complex, involving multiple speakers, irrelevant phatic utterances, and dialogue anaphora.

E.T. model performance on TfD is poor but non-zero, unlike a Random baseline. We do not perform additional ablations for TfD given the low initial performance. Notably, in addition to the complexity of language, TfD instances have substantially longer average trajectory length ($\sim 130$) than those in ALFRED ($\sim 50$).

5.2 Rule-based Agents for TATC

In benchmarks like ALFRED, a PDDL (Ghallab et al. 1998) planner can be used to determine what actions are necessary to solve relatively simple tasks. In VLN, simple search algorithms yield the shortest paths to goals. Consequently, some language-guided visual task models build a semantic representation of the environment, then learn a hierarchical policy to execute such planner-style goals (Blukis et al. 2021).

Inspired by such planning-based solutions, we attempted to write a pair of rule-based Commander and Follower agents to tackle the TATC benchmark. In a loop, the rule-based Commander executes a Progress Check action, then forms a language utterance to the Follower consisting of navigation and object interaction actions needed to accomplish the next sub-goal in the response. Each sub-goal needs to be identified by the language template used to describe it, then a hand-crafted policy must be created for the rule-based Commander to reference. For example, for the put all X on Y task, all sub-goals of the form “X needs to be on some Y” for a particular instance of object X, and so a rule-based policy can be expressed as “navigate to the X instance, pick up the X instance, navigate to Y, put X down on Y.” Commander utterances are simplified to se-
it is clear that, unlike ALFRED and navigation-only tasks, success rates could certainly be increased by increasing sub-engineering work to hand-craft subgoal policies. While succeeded. The rule-based agents represent about 150 hours of agents perform \((x, y)\) screen click positions to select objects. The rule-based agents perform no learning.

Table 5 summarizes the success rate of these rule-based agents across task types. Note that for the tasks BOIL, POTATO, MAKE PLATE OF TOAST, MAKE SANDWICH, and BREAKFAST, sub-goal policies were not successfully developed. The rule-based agents represent about 150 hours of engineering work to hand-craft subgoal policies. While success rates could certainly be increased by increasing sub-goal policy coverage and handling simulation corner cases, it is clear that, unlike ALFRED and navigation-only tasks, a planner-based solution is not reasonable for TEACH data and the TATC benchmark. The appendix contains detailed implementation information about the rule-based agents.

### 6 Conclusions and Future Work

We introduce Task-driven Embodied Agents that Chat (TEACH), a dataset of over 3000 situated dialogues in which a human Commander and human Follower collaborate in natural language to complete household tasks in the AI2-THOR simulation environment. TEACH contains dialogue phenomena related to grounding dialogue in objects and actions in the environment, varying levels of instruction granularity, and interleaving of utterances between speakers in the absence of enforced turn taking. We also introduce a task definition language that is extensible to new tasks and even other simulators. We propose three benchmarks based on TEACH. To study Follower models, we define the Execution from Dialogue History (EDH) and Trajectory from Dialogue (TfD) benchmarks, and evaluate an adapted Episodic Transformer (Pashevich, Schmid, and Sun 2021) as an initial baseline. To study the potential of Commander and Follower models, we define the Two-Agent Task Completion benchmark, and explore the difficulty of defining rule-based agents from TEACH data.

In future, we will apply other ALFRED modeling approaches (Blukis et al. 2021; Kim et al. 2021; Zhang and Chai 2021; Suglia et al. 2021) to the EDH and TfD Follower model benchmarks. However, TEACH requires learning several different tasks, all of which are more complex than the simple tasks in ALFRED. Models enabling few shot generalization to new tasks will be critical for TEACH Follower agents. For Commander models, a starting point would be to train a Speaker model (Fried et al. 2018) on TEACH sessions. We are excited to explore human-in-the-loop evaluation of Commander and Follower models developed for TATC.
7 Acknowledgements

We would like to thank Ron Rezac, Shui Hu, Lucy Hu, Hangjie Shi for their assistance with the data and code release, and Sijia Liu for assistance with data cleaning. We would also like to thank Nicole Chartier, Savanna Staff, Ana Sanchez, Ben Kelk, Joel Sachar, Govind Thattai, Gaurav Sukhatme, Joel Chengottusseryil, Tony Bissell, Qiaozi Gao, Kaixiang Lin, Karthik Gopalakrishnan, Alexandros Papangelis, Yang Liu, Mahdi Namazifar, Behnam Hedayatnia, Di Jin, Seokhwan Kim and Nikko Strom for feedback and suggestions over the course of the project.

References


Kim, B.; Bhamibri, S.; Singh, K. P.; Mottaghi, R.; and Choi, J. 2021. Agent with the Big Picture: Perceiving Surroundings for Interactive Instruction Following. In Embodied AI Workshop CVPR.


