TEACh: Task-Driven Embodied Agents That Chat

Aishwarya Padmakumar* [1], **Jesse Thomason*** [1 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 ***T**ask-driven **E**mbodied **A**gents that **C**hat* (*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.

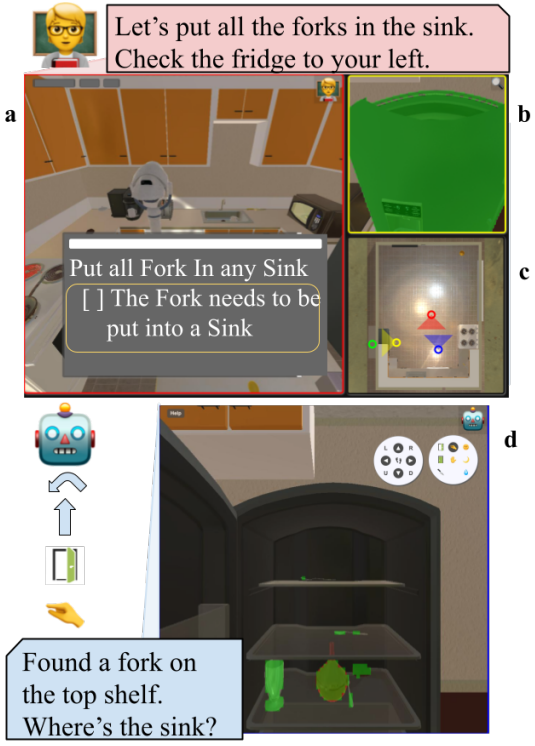

Figure 1: The *Commander* has oracle task details (a), object locations (b), a map (c), and egocentric views from both agents. The *Follower* carries out the task and asks questions (d). The agents can only communicate via language.

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

[1] https://github.com/alexa/teach

Dataset	— Object —		— Language —			Demonstrations
	Interaction	State Changes	Conversational	# Sessions	Freeform	
R2R (Anderson et al. 2018)	✗	✗	✗	-	-	Planner
CHAI (Misra et al. 2018)	✓	✓	✗	-	-	Human
CVDN (Thomason et al. 2019)	✗	✗	✓	2050	✗	Human
CerealBar (Suhr et al. 2019)	✓	✗	✓	1202	✗	Human
MDC (Narayan-Chen et al. 2019)	✓	✗	✓	509	✓	Human
ALFRED (Shridhar et al. 2020)	✓	✓	✗	-	-	Planner
III (Abramson et al. 2020)	✓	✗	✗	-	-	Human
TEACh	✓	✓	✓	3215	✓	Human

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.

Transformer (E.T.) model (Pashevich, Schmid, and Sun 2021) and demonstrate the difficulty of engineering rule-based solvers for end-to-end task completion.

The main contributions of this work are:

- *TEACh*, a dataset of over 3000 human-human dialogs simulating the experience of a user interacting with their robot to complete tasks in their home, that interleaves dialogue messages with actions taken in the environment.
- An extensible task definition framework (§3) that can be used to define and check completion status for a wide range of tasks in a simulated environment.
- Three benchmarks based on *TEACh* sessions and experiments demonstrating initial models for each.

2 Related Work

Table 1 situates *TEACh* with respect to other datasets involving natural language instructions for visual task completion.

Vision & **Language Navigation (VLN)** tasks agents with taking in language instructions and a visual observation to produce an action, such as turning or moving forward, to receive a new visual observation. VLN benchmarks have evolved from the use of symbolic environment representations (MacMahon, Stankiewicz, and Kuipers 2006; Chen and Mooney 2011; Mei, Bansal, and Walter 2016) to photorealistic indoor (Anderson et al. 2018) and outdoor environments (Chen et al. 2019), as well as the prediction of continuous control (Blukis et al. 2018). *TEACh* goes beyond navigation to object interactions for task completion, and beyond single instructions to dialogue.

Vision & **Language Task Completion** involves actions beyond navigation. Models have evolved from individual rule-based or learned components for language understanding, perception and action execution (Matuszek et al. 2013; Kollar et al. 2013), to end-to-end models in fully observable blocks worlds (Bisk et al. 2018; Misra et al. 2018). More complex tasks involve partially observable worlds (Kim et al. 2020) and object state changes (Misra et al. 2018; Puig et al. 2018; Shridhar et al. 2020). Some works use a planner to generate ideal demonstrations that are then labeled, while others first gather instructions and gather human demonstrations (Misra et al. 2018; Shah et al. 2021; Abramson et al. 2020). In *TEACh*, human instructions and demonstrations are gathered simultaneously.

Vision & **Dialogue Navigation and Task Completion** Agents that additionally engage in dialogue can be learned by combining individual rule-based or learned components (Tellex et al. 2016; Arumugam et al. 2018; Thomason et al. 2020). End-to-end VLN models can be improved by simulated clarification (Chi et al. 2020; Nguyen and Daumé III 2019) and incorporating human-human conversation history (Thomason et al. 2019; Zhu et al. 2020). Other works learn agent-agent policies for navigating and speaking (Roman et al. 2020; Shrivastava et al. 2021), and deploy individual agent policies for human-in-the-loop evaluation (Suhr et al. 2019). However, such models and underlying datasets are limited to navigation actions and turn-taking conversation. In contrast, *TEACh* involves *Follower* navigation and object interaction, as well as freeform dialogue acts with the *Commander*. The Minecraft Dialogue Corpus (MDC) (Narayan-Chen, Jayannavar, and Hockenmaier 2019) gives full dialogues between two humans for assembly tasks. MDC is similar in spirit to *TEACh*; we introduce a larger action space and resulting object state changes, such as slicing and toasting bread, as well as collecting many more human-human dialogues.

3 The *TEACh* Dataset

We collect 3,047 human–human *gameplay sessions* for completing household tasks in the AI2-THOR simulator (Kolve et al. 2017). Each session includes an initial environment state, *Commander* actions to access oracle information, utterances between the *Commander* and *Follower*, movement actions, and object interactions taken by the *Follower*. Figure 2 gives an overview of the annotation interface.

3.1 Household Tasks

We design a *task definition language* (TDL) to define household tasks in terms of object properties to satisfy, and implement a framework over AI2-THOR that evaluates these

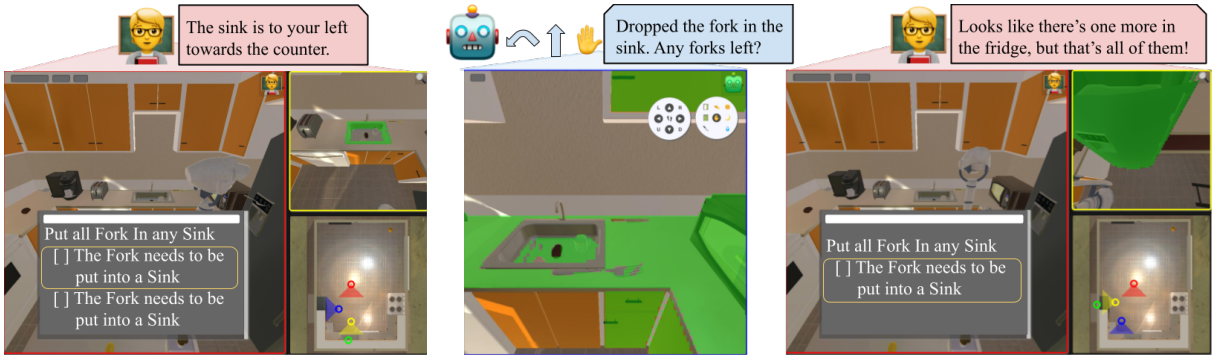

Figure 2: To collect *TEACh*, the *Commander* knows the task to be completed and can query the simulator for object locations. Searched items are highlighted in green for the *Commander*; highlights blink to enable seeing the underlying true scene colors. The *Commander* has a topdown map of the scene, 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 *Commander*s (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/

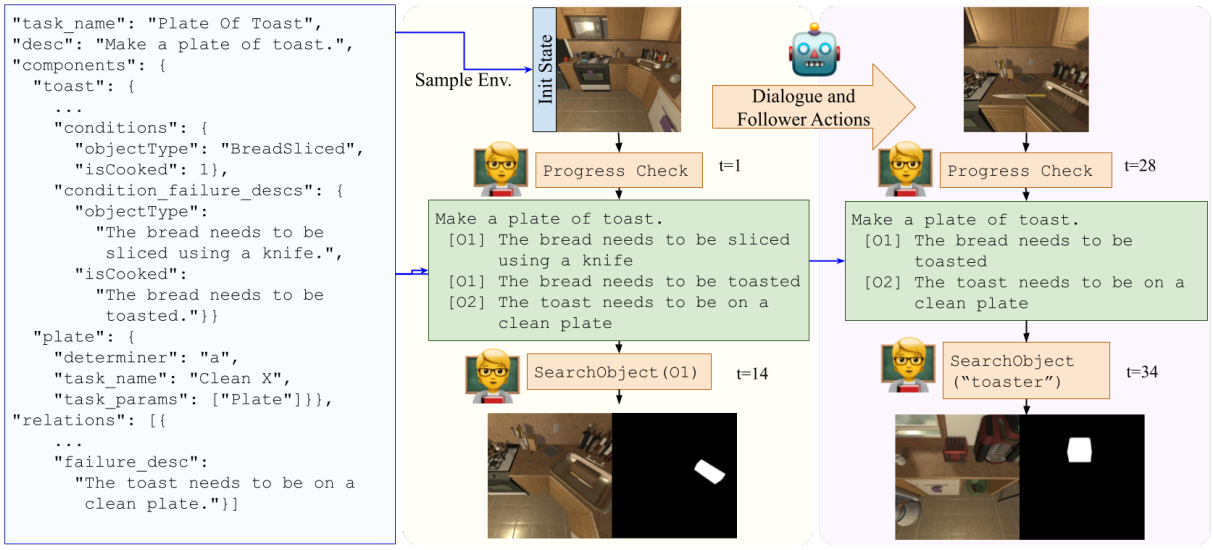

Figure 3: An example task definition from the *TEACh* task definition language (left) and how it informs the initial simulator state and the *Commander* `Progress Check` action. The *Commander* can `SearchObject` with a string query (right) or object instance (center) returned by the `Progress Check` task status, yielding a camera view, segmentation mask, and location.

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 the 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 $\left(S^E, A_H, A^I_R, F^E\right)$ where S^E is the initial state of the EDH instance, A_H 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^I_R reference interaction actions taken in the session as supervision. We constrain instances to have $|A^D_H| > 0$ and at least one object interaction in A^I_R. 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^I_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^I_R and inferred action sequence $\hat{A}^I$, we calculate trajectory length weighted metric for metric value m as

$$TLW\text{-}m = \frac{m * |A^I_R|}{max(|A^I_R|, |\hat{A}^I|)}.$$

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 (TfD)

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 TfD instance is $\left(S_i, A^D_H, A^I_R, S_f\right)$, where A^D_H is all dialogue actions taken by both agents, and A^I_R is all non-dialogue actions taken by the *Follower*. We append a `Stop` action to A^I_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.

	Parameter Variants	Unique Scenes	Total Sessions	Utterances per Session	*Follower* Actions/Session	All Actions/Session
WATER PLANT	1	10	176	6.37± 4.36	51.86± 30.71	67.93± 40.70
MAKE COFFEE	1	30	308	7.75± 5.08	55.25± 33.61	72.29± 50.85
CLEAN ALL X	19	52	336	9.65± 7.03	74.06± 59.66	96.92± 71.31
PUT ALL X ON Y	209	92	344	8.66± 5.82	82.13± 66.39	103.53± 80.97
BOIL POTATO	1	26	202	10.65± 7.61	104.66± 79.50	130.13± 94.80
MAKE PLATE OF TOAST	1	27	225	12.26± 8.51	108.30± 55.81	136.11± 70.73
N SLICES OF X IN Y	16	29	304	13.50±10.86	113.62± 94.25	146.23±113.96
PUT ALL X IN ONE Y	84	50	302	11.32± 7.03	115.74± 90.13	147.80±104.45
N COOKED X SLICES IN Y	10	30	240	14.94± 9.43	155.18± 75.17	189.26± 87.90
PREPARE SANDWICH	5	28	241	18.03± 9.96	195.93± 83.96	241.61±100.86
PREPARE SALAD	9	30	323	20.47±10.80	206.29±111.47	253.94±130.09
PREPARE BREAKFAST	80	30	308	27.67±14.73	295.06±138.76	359.90±162.33
***TEACh* Overall**	**438**	**109**	**3320**	**13.67**±10.81	**131.80**±109.68	**164.65**±130.89

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.

Fold	Split	# Sessions	# EDH Instances
Train		1482 (49%)	5758 (49%)
Val	Seen	181 (6%)	654 (5%)
	Unseen	614 (20%)	2188 (19%)
Test	Seen	181 (6%)	696 (6%)
	Unseen	589 (19%)	2370 (20%)

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

prediction with a block of instructions, analogous to ALFRED or TouchDown (Chen et al. 2019). 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)

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.

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 TfD

We use a single model architecture to train and evaluate on the EDH and TfD benchmark tasks.

Model. We establish baseline performance for the EDH and TfD tasks using the Episodic Transformer (E.T.) model (Pashevich, 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 MaskRCNN (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 TfD instance, we extract all dialogue utterances from the action history A_H^D 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_R^I. Note that in TfD, $|A_H^I| = 0$.

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

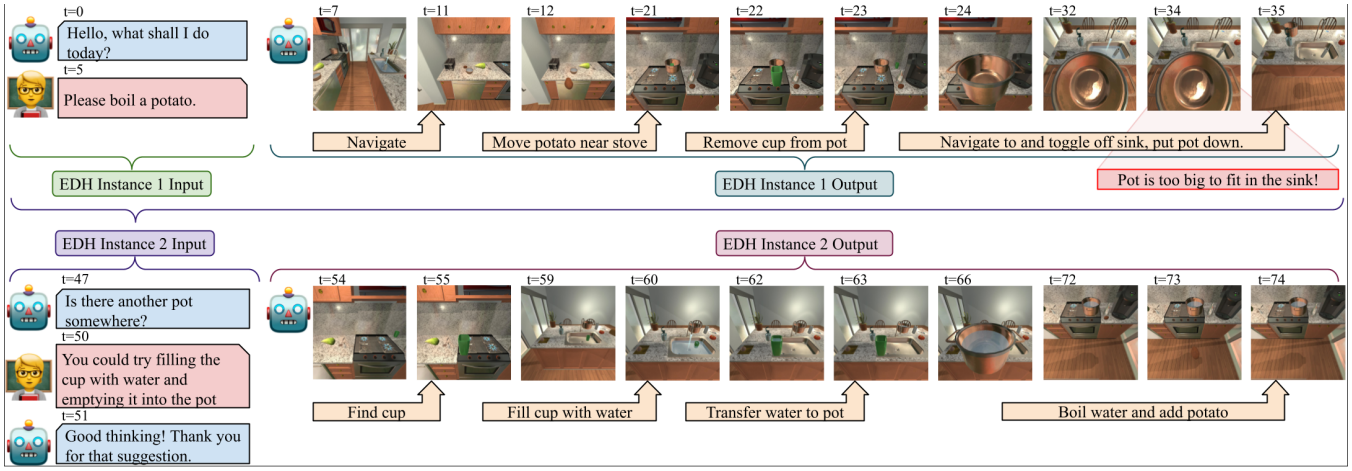

Figure 4: Two EDH instances are constructed from this real example from the *TEACh* data. The first instance input contains only dialogue actions. After inference on the first instance, the agent is evaluated based on whether it moved the potato, pot, and the items cleared out of the sink to their target destinations. In this example, the pot cannot fit into the sink. The second instance input has both dialogue and environment actions, and is evaluated at inference by whether the pot lands on the stove filled with water, and whether the potato is inside the pot and boiled.

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^I_H 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 (∼130) than those in ALFRED (∼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 are 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-

	EDH Validation				EDH Test			
	Seen		*Unseen*		*Seen*		*Unseen*	
Model	SR [TLW]	GC [TLW]	SR [TLW]	GC [TLW]	SR [TLW]	GC [TLW]	SR [TLW]	GC [TLW]
Random	0.82 [0.62]	0.17 [0.2]	1.54 [0.55]	0.04 [-0.16]	0.6 [0.09]	0.25 [0.24]	1.9 [0.94]	0.17 [0.06]
Lang	3.12 [0.27]	1.84 [1.25]	4.0 [1.19]	3.93 [4.34]	4.2 [1.0]	2.79 [2.71]	4.01 [0.63]	4.66 [4.06]
Vision	8.88 [0.89]	8.79 [2.24]	5.68 [1.07]	4.99 [3.91]	3.45 [0.79]	2.45 [1.82]	6.44 [0.87]	6.95 [4.2]
E.T.	**9.05 [1.2]**	**9.05 [4.17]**	**13.49 [3.69]**	**12.97 [12.15]**	**12.16 [2.48]**	**10.96 [6.41]**	**9.62 [2.52]**	**10.49 [7.64]**
+H	12.5 [1.78]	16.96 [5.61]	12.19 [2.9]	12.36 [10.57]	15.62 [1.56]	17.57 [5.66]	6.66 [0.46]	8.19 [3.9]
+A	8.88 [1.14]	9.1 [3.49]	14.01 [3.97]	13.35 [12.28]	10.06 [1.3]	9.21 [4.28]	8.82 [1.14]	9.68 [5.53]
+S	7.73 [0.93]	7.77 [3.41]	13.22 [3.67]	13.01 [11.91]	9.76 [0.95]	8.62 [3.73]	8.82 [1.06]	9.62 [5.52]
+H+A	9.38 [1.27]	9.93 [4.38]	13.45 [3.14]	13.42 [11.17]	10.36 [1.3]	8.45 [3.54]	8.16 [0.89]	7.7 [4.58]
+H+S	11.18 [0.97]	10.55 [4.48]	13.26 [2.97]	12.93 [10.59]	10.96 [1.78]	11.02 [4.98]	6.66 [1.02]	7.8 [4.2]
	TfD Validation				**TfD Test**			
Rand	0.00 [0.00]	0.00 [0.00]	0.00 [0.00]	0.00 [0.00]	0.00 [0.00]	0.00 [0.00]	0.00 [0.00]	0.00 [0.00]
E.T.	**1.02 [0.17]**	**1.42 [4.82]**	**0.48 [0.12]**	**0.35 [0.59]**	**0.51 [0.23]**	**1.60 [6.46]**	**0.17 [0.04]**	**0.67 [2.50]**

Table 4: E.T. outperforms random and unimodal baselines (**bold**). We ablate history loss (H), initializing with ALFRED (A), and initializing with ALFRED synthetic language (S). Metrics are success rate (SR) and goal condition success rate (GC). Trajectory length weighted metrics are included in [brackets]. All values are percentages. For all metrics, higher is better.

Task (Shrtnd)	Success Rate	Rule Agent Actions/Session	Human Actions/Session
PLANT	26.70	230.26± 54.65	67.93± 40.70
COFFEE	54.55	120.24± 66.55	72.29± 50.85
CLEAN	52.98	182.38± 79.84	96.92± 71.31
ALL X Y	52.91	126.82± 64.75	103.53± 80.97
BOIL	0.00	-	130.13± 94.80
TOAST	0.00	-	136.11± 70.73
N SLICES	22.51	248.77± 98.57	146.23±113.96
X ONE Y	50.98	150.09± 97.12	147.80±104.45
COOKED	1.67	424.25±135.57	189.26± 87.90
SNDWCH	0.00	-	241.61±100.86
SALAD	1.55	351.20± 82.09	253.94±130.09
BFAST	0.00	-	359.90±162.33
Overall	24.40	161.54± 92.00	164.65±130.89

Table 5: Rule-based agent policies were expansive enough to solve some simple tasks about half the time, while being unable to solve most compositional tasks at all. Note that TATC performance is not directly comparable to EDH or TfD due to two-agent modeling in TATC.

quences of action names with a one-to-one mapping to *Follower* actions to execute, with interaction actions including (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 subgoal 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 ***T**ask-driven **E**mbodied **A**gents that **C**hat* (*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.

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 Stiff, Ana Sanchez, Ben Kelk, Joel Sachar, Govind Thattai, Gaurav Sukhatme, Joel Chengottusseriyil, 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

Abramson, J.; Ahuja, A.; Brussee, A.; Carnevale, F.; Cassin, M.; Clark, S.; Dudzik, A.; Georgiev, P.; Guy, A.; Harley, T.; Hill, F.; Hung, A.; Kenton, Z.; Landon, J.; Lillicrap, T.; Mathewson, K.; Muldal, A.; Santoro, A.; Savinov, N.; Varma, V.; Wayne, G.; Wong, N.; Yan, C.; and Zhu, R. 2020. Imitating Interactive Intelligence. *arXiv*.

Anderson, P.; Wu, Q.; Teney, D.; Bruce, J.; Johnson, M.; Sünderhauf, N.; Reid, I.; Gould, S.; and van den Hengel, A. 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 (CVPR)*.

Arumugam, D.; Karamcheti, S.; Gopalan, N.; Williams, E. C.; Rhee, M.; Wong, L. L.; and Tellex, S. 2018. Grounding Natural Language Instructions to Semantic Goal Representations for Abstraction and Generalization. *Autonomous Robots*.

Bisk, Y.; Holtzman, A.; Thomason, J.; Andreas, J.; Bengio, Y.; Chai, J.; Lapata, M.; Lazaridou, A.; May, J.; Nisnevich, A.; Pinto, N.; and Turian, J. 2020. Experience Grounds Language. In *Empirical Methods in Natural Language Processing (EMNLP)*.

Bisk, Y.; Shih, K.; Choi, Y.; and Marcu, D. 2018. Learning Interpretable Spatial Operations in a Rich 3D Blocks World. In *Proceedings of the Thirty Second AAAI Conference on Artificial Intelligence (AAAI)*, volume 32.

Blukis, V.; Misra, D.; Knepper, R. A.; and Artzi, Y. 2018. Mapping Navigation Instructions to Continuous Control Actions with Position Visitation Prediction. In *Proceedings of the Conference on Robot Learning (CoRL)*.

Blukis, V.; Paxton, C.; Fox, D.; Garg, A.; and Artzi, Y. 2021. A Persistent Spatial Semantic Representation for High-level Natural Language Instruction Execution. *arXiv*.

Chen, D.; and Mooney, R. 2011. Learning to Interpret Natural Language Navigation Instructions from Observations. In *Proceedings of the Twenty Fifth AAAI Conference on Artificial Intelligence (AAAI)*, volume 25.

Chen, H.; Suhr, A.; Misra, D.; Snavely, N.; and Artzi, Y. 2019. Touchdown: Natural Language Navigation and Spatial Reasoning in Visual Street Environments. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 12538–12547.

Chi, T.-C.; Shen, M.; Eric, M.; Kim, S.; and Hakkani-Tür, D. 2020. Just Ask: An Interactive Learning Framework for Vision and Language Navigation. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Fried, D.; Hu, R.; Cirik, V.; Rohrbach, A.; Andreas, J.; Morency, L.-P.; Berg-Kirkpatrick, T.; Saenko, K.; Klein, D.; and Darrell, T. 2018. Speaker-Follower Models for Vision-and-Language Navigation. In *Neural Information Processing Systems (NeurIPS)*.

Ghallab, M.; Howe, A.; Knoblock, C.; McDermott, D.; Ram, A.; Veloso, M.; Weld, D.; and Wilkins, D. 1998. PDDL The Planning Domain Definition Language. *Yale Center for Computational Vision and Control*.

Harnad, S. 1990. The Symbol Grounding Problem. *Physica D: Nonlinear Phenomena*, 42(1-3): 335–346.

He, K.; Gkioxari, G.; Dollár, P.; and Girshick, R. B. 2017. Mask R-CNN. *International Conference on Computer Vision (ICCV)*.

Kim, B.; Bhambri, 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*.

Kim, H.; Zala, A.; Burri, G.; Tan, H.; and Bansal, M. 2020. ArraMon: A Joint Navigation-Assembly Instruction Interpretation Task in Dynamic Environments. In *Findings of the Association for Computational Linguistics: EMNLP 2020*.

Kollar, T.; Tellex, S.; Walter, M. R.; Huang, A.; Bachrach, A.; Hemachandra, S.; Brunskill, E.; Banerjee, A.; Roy, D.; Teller, S.; et al. 2013. Generalized Grounding Graphs: A Probabilistic Framework for Understanding Grounded Language. *Journal of Artificial Intelligence Research*.

Kolve, E.; Mottaghi, R.; Han, W.; VanderBilt, E.; Weihs, L.; Herrasti, A.; Gordon, D.; Zhu, Y.; Gupta, A.; and Farhadi, A. 2017. AI2-THOR: An Interactive 3D Environment for Visual AI. *arXiv preprint arXiv:1712.05474*.

MacMahon, M.; Stankiewicz, B.; and Kuipers, B. 2006. Walk the Talk: Connecting Language, Knowledge, and Action in Route Instructions. In *Proceedings of the Twentieth AAAI Conference on Artifial Intelligence (AAAI)*, volume 20.

Matuszek, C.; Herbst, E.; Zettlemoyer, L.; and Fox, D. 2013. Learning to Parse Natural Language Commands to a Robot Control System. In *Experimental Robotics*, 403–415. Springer.

Mei, H.; Bansal, M.; and Walter, M. 2016. Listen, attend, and walk: Neural mapping of navigational instructions to action sequences. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI)*, volume 30.

Misra, D. K.; Bennett, A.; Blukis, V.; Niklasson, E.; Shatkhin, M.; and Artzi, Y. 2018. Mapping Instructions to Actions in 3D Environments with Visual Goal Prediction. In Riloff, E.; Chiang, D.; Hockenmaier, J.; and Tsujii, J., eds., *Empirical Methods in Natural (EMNLP)*.

Narayan-Chen, A.; Jayannavar, P.; and Hockenmaier, J. 2019. Collaborative Dialogue in Minecraft. In *Association for Computational Linguistics (ACL)*.

Nguyen, K.; and Daumé III, H. 2019. Help, Anna! Visual Navigation with Natural Multimodal Assistance via Retrospective Curiosity-Encouraging Imitation Learning. In *Conference on Empirical Methods in Natural Language Processing and International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*.

Pashevich, A.; Schmid, C.; and Sun, C. 2021. Episodic Transformer for Vision-and-Language Navigation. *arXiv preprint arXiv:2105.06453*.

Puig, X.; Ra, K.; Boben, M.; Li, J.; Wang, T.; Fidler, S.; and Torralba, A. 2018. VirtualHome: Simulating Household Activities via Programs. In *Computer Vision and Pattern Recognition (CVPR)*.

Roman, H. R.; Bisk, Y.; Thomason, J.; Celikyilmaz, A.; and Gao, J. 2020. RMM: A Recursive Mental Model for Dialog Navigation. In *Findings of Empirical Methods in Natural Language Processing (EMNLP Findings)*.

Shah, R.; Wild, C.; Wang, S. H.; Alex, N.; Houghton, B.; Guss, W.; Mohanty, S.; Kanervisto, A.; Milani, S.; Topin, N.; et al. 2021. The MineRL BASALT Competition on Learning from Human Feedback. *arXiv preprint arXiv:2107.01969*.

Shridhar, M.; Thomason, J.; Gordon, D.; Bisk, Y.; Han, W.; Mottaghi, R.; Zettlemoyer, L.; and Fox, D. 2020. ALFRED: A Benchmark for Interpreting Grounded Instructions for Everyday Tasks. In *Computer Vision and Pattern Recognition (CVPR)*.

Shrivastava, A.; Gopalakrishnan, K.; Liu, Y.; Piramuthu, R.; Tür, G.; Parikh, D.; and Hakkani-Tür, D. 2021. VISITRON: Visual Semantics-Aligned Interactively Trained Object-Navigator. *arXiv preprint arXiv:2105.11589*.

Suglia, A.; Gao, Q.; Thomason, J.; Thattai, G.; and Sukhatme, G. 2021. Embodied BERT: A Transformer Model for Embodied, Language-guided Visual Task Completion. *arXiv preprint arXiv:2108.04927*.

Suhr, A.; Yan, C.; Schluger, J.; Yu, S.; Khader, H.; Mouallem, M.; Zhang, I.; and Artzi, Y. 2019. Executing Instructions in Situated Collaborative Interactions. In *Empirical Methods in Natural Language Processing and International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*.

Tellex, S.; Knepper, R. A.; Li, A.; Roy, N.; and Rus, D. 2016. Asking for Help Using Inverse Semantics. In *Robotics: Science and Systems Conference (RSS)*.

Thomason, J.; Gordon, D.; and Bisk, Y. 2018. Shifting the Baseline: Single Modality Performance on Visual Navigation & QA. *arXiv preprint arXiv:1811.00613*.

Thomason, J.; Murray, M.; Cakmak, M.; and Zettlemoyer, L. 2019. Vision-and-Dialog Navigation. In *Conference on Robot Learning (CoRL)*.

Thomason, J.; Padmakumar, A.; Sinapov, J.; Walker, N.; Jiang, Y.; Yedidsion, H.; Hart, J.; Stone, P.; and Mooney, R. 2020. Jointly improving parsing and perception for natural language commands through human-robot dialog. *Journal of Artificial Intelligence Research*, 67: 327–374.

Zhang, Y.; and Chai, J. 2021. Hierarchical Task Learning from Language Instructions with Unified Transformers and Self-Monitoring. *arXiv preprint arXiv:2106.03427*.

Zhu, W.; Hu, H.; Chen, J.; Deng, Z.; Jain, V.; Ie, E.; and Sha, F. 2020. BabyWalk: Going Farther in Vision-and-Language Navigation by Taking Baby Steps. In *Association for Computational Linguistics (ACL)*.