Supplementary Material for HandMeThat: Human-Robot Communication in Physical and Social Environments

Yanming Wan[*] IIIS, Tsinghua University

Jiayuan Mao[*] MIT CSAIL

Joshua B. Tenenbaum MIT BCS, CBMM, CSAIL

The generated episodes, the gym-style environment, and codes for reproducing baseline results are available on our website: http://handmethat.csail.mit.edu/. This supplementary material document is organized as the following. In Section A, we provide the detailed information for HandMeThat data generation and its textual interface. In Section B, we summarize the statistics of the dataset. In Section C we present the experimental details including the baseline implementations, the full results with error bars, and the performance of Offline-DRRN, an offline variant of the DRRN model. Section D discusses the possibility of introducing human-robot interaction to HandMeThat. Section E is the documentation for HandMeThat following Datasheets for Datasets [1] standards.

A Task Design

A.1 Object Space

Recall that HandMeThat uses an object-centric representation for states. Here, we present all the object categories, attributes, and how the object universe is sampled.

Object hierarchy. HandMeThat classifies all categories into 5 classes: location, receptacle, food, tool, and thing. "Location" consists of all non-movable entities. They are the positions that an agent can "move to." "Receptacle" consists of all movable containers. The latter three classes are ordinary (non-receptacle) movable physical entities.

Each class (except for "location") is composed of multiple subclasses, and each subclass contains several object categories. In total, there are 155 object categories. We list the hierarchies for all the five classes in Table 1. In a scene, there could be multiple objects within a same category, labeled with different IDs.

Attributes and relations. Each object category is also associated with several attributes. There are 2 static attributes and 8 non-static attributes. The static ones are *size* and *color*, which include {large, small} and {red, green, blue}, respectively. The non-static attributes are all Boolean-valued, including *open, cooked, frozen, dusty, stained, sliced, soaked, toggled*. Each attribute is associated with a meta-predicate: *has-X* or *X-able* (e.g., *has-size* and *openable*), indicating whether a object (of a certain category) can be assigned assigned with that attribute.

For relations, we only consider spatial relations between object pairs, i.e., "inside" and "on top of." Each movable object can be in/on a "location" object, and each non-receptacle movable object can be in/on a "receptacle" object. Note that we only explicitly represent *direct* spatial relations, that is, when *apple#1 is in plate#1* and *plate#1 is on table#1*, we do not explicitly represent *apple#1 is on table#1*. This representation brings us the advantage that when the plate is moved to another place, we do not need to manually re-assign the relationship between apple#1 and table#1. There are two additional meta-properties, associated with location and receptacle categories: "has-inside" and "has-ontop,", indicating whether the object of the corresponding category can hold another object inside or on top of it. We include all details in Table 2.

[*]indicates equal contribution. Correspondence to: jiayuanm@mit.edu.

36th Conference on Neural Information Processing Systems (NeurIPS 2022) Track on Datasets and Benchmarks.

Class	Subclass	Category
location	(has-ontop)	floor, countertop, sofa, bed, stove, table, shelf, toilet
	(has-inside)	cabinet, bathtub, microwave, oven, dishwasher, refrigerator, sink, pool
receptacle	furniture	highchair, chair, seat
	vessel	bottle, jar, kettle, caldron
	tableware	bowl, mug, plate, dish, cup
	utensil	saucepan, pan, casserole
	bag	duffel bag, sack, backpack, briefcase
	bucket tray basket box package ashcan Xmas stocking Xmas tree	bucket tray basket box package ashcan Xmas stocking Xmas tree
food	fruit	apple, banana, melon, grape, lemon, orange, peach, strawberry, raspberry, date, olive, chestnut
	vegetable	carrot, radish, tomato, broccoli, mushroom, onion, lettuce, pumpkin
	drink	pop, beer, juice, water, milk
	protein	beef, chicken, pork, fish, egg
	flavorer	catsup, sauce, parsley, tea bag, suger, vegetable oil
	baked food	cracker, bread, cookie, cake
	snack	chip, hamburger, sandwich, candy
	prepared food	oatmeal, sushi, salad, soup, pasta
tool	metal tool	carving knife, hammer, screwdriver, scraper, saw
	electric equipment	printer, scanner, facsimile, modem
	electrical device	calculator, headset, earphone, mouse, alarm
	toiletry	toothbrush, perfume, makeup
	writing tool	highlighter, marker, pen, pencil
	piece of cloth	dishtower, hand towel, rag
	cleaning tool	scrub brush, broom, vacuum
	cleansing	soap, shampoo, detergent, toothpaste
	cutlery	fork, spoon, knife
	illumination tool	lamp, candle
thing	decoration	necklace, bracelet, jewelry, bow, wreath, ribbon
	paper product	hardback, notebook, book, newspaper, painting, pad, document
	footwear	gym shoe, sandal, shoe, sock
	headwear	hat, sunglass
	clothing	shirt, sweater, underwear, apparel
	building materials	tile, plywood
	plaything	cube, ball

Table 1: Object hierarchy designed for HandMeThat based on BEHAVIOR-100.

Meta-Properties	Categories
has-inside	cabinet, bathtub, microwave, oven, dishwasher, refrigerator, sink, pool, vessel, tableware, utensil, bag, basket, box, package, ashcan, bucket, Xmas stocking
has-ontop	floor, countertop, sofa, bed, stove, table, shelf, toilet furniture, tray, Xmas tree
has-size	tableware tray box package ashcan
has-color	furniture vessel bag basket box package
openable	cabinet, microwave, oven, dishwasher, refrigerator, vessel, bag, box, package
toggleable	microwave, oven dishwasher, refrigerator, stove, sink, electric equipment
cookable	food
freezable	food
sliceable	fruit, vegetable, protein
dustyable	location, receptacle, thing
stainable	location
soakable	piece of cloth, clothing

Table 2: All meta-properties in HandMeThat and their corresponding categories. "has-inside" and "has-ontop" indicate whether the objects in this category can hold other objects in the corresponding relation; "has-size" and "has-color" indicate whether the objects in this category will be assigned with size or color; the remaining properties (X-able) indicate whether objects in this category can be manipulated through corresponding verb. The category list contains class/subclass names: all categories derived from the listed class/subclass are associated with the corresponding meta-property.

Valid actions. To interact with the world, an agent can perform a variety of actions. We present all the action schemas in Table 3 written in PDDL format [2]. Formally, each action should be defined as a collection of preconditions and effects, which represents the change in state variables. But here in the table we only introduce the intuitive definitions in Semantics column. Please refer to our code at http://handmethat.csail.mit.edu/ for a complete definition of STRIPS operators in HandMeThat. Note that some of the actions are only for robot agents. Additionally, there are constraints specified for a partial set of actions, as shown in Table 4.

Initial state sampling. The object universe is sampled in four steps. First, we create exactly one instance of all categories in the "location" category. Second, we sample the number of objects in each category (up to 3). We make sure that there is at least one object in each subclass. The objects that belong to the same category are labeled by different IDs. To this end, we have generated a massive number of objects.

The third step is to sample the positions for all movable objects. We list all the valid positions for categories in each subclass in Table 5, to reflect the distribution of real-world object placements. The position is uniformly sampled from all locations and receptacles, at the category-level. If there are multiple receptacles of the category, we use a second-stage uniform sampling to assign the object to one of the receptacle instance.

The last step is to assign attribute values for each object, based on the following rules:

- No objects are open, sliced, or soaked initially. Only the refrigerator is toggled on.
- Objects at ovens, stoves, and microwaves are cooked. Objects in refrigerators are frozen.
- Objects are stained (dusty) with probability $1/3$ if stainable (dustyable).
- The size and color properties of objects are uniformly sampled.

A.2 Goal Space

Since all object categories and states above are inherited from BEHAVIOR-100, we can design the goal space based on human-annotated household tasks in BEHAVIOR-100.

Compositional goal templates. We summarize the original 100 tasks (by removing duplicates and incompatible ones) into 69 goal templates, represented in first-order logic statements. All the templates are shown in Table 6 to Table 10. To instantiate from a template, we first sample a corresponding concrete object category to replace each subclass in brackets. For example, we

Action Name and Parameters	Semantics
Action: move **Args:** ?a - agent ?from - location ?to - location	Agent [?a] moves from [?from] to [?to].
Action: pick-up-at-loc **Args:** ?a - agent ?obj - object ?loc - location	Agent [?a] picks up [?obj] at [?loc].
Action: pick-up-from-rec-at-loc **Args:** ?a - agent ?obj - object ?rec - receptacle ?loc - location	Agent [?a] picks up [?obj] from [?rec] at [?loc].
Action: put-inside-loc **Args:** ?a - agent ?obj - object ?loc - location	Agent [?a] puts [?obj] into [?loc].
Action: put-ontop-loc **Args:** ?a - agent ?obj - object ?loc - location	Agent [?a] puts [?obj] onto [?loc].
Action: put-inside-rec-at-loc **Args:** ?a - agent ?obj - object ?rec - receptacle ?loc - location	Agent [?a] puts [?obj] into [?rec] at [?loc].
Action: put-ontop-loc-at-loc **Args:** ?a - agent ?obj - object ?rec - receptacle ?loc - location	Agent [?a] puts [?obj] onto [?rec] at [?loc].
Action: open-loc **Args:** ?a - agent ?loc - location	Agent [?a] opens [?loc].
Action: close-loc **Args:** ?a - agent ?loc - location	Agent [?a] closes [?loc].
Action: open-rec-at-loc **Args:** ?a - agent ?rec - receptacle ?loc - location	Agent [?a] opens [?rec] at [?loc].
Action: close-rec-at-loc **Args:** ?a - agent ?rec - receptacle ?loc - location	Agent [?a] closes [?rec] at [?loc].
Action: toggle-on-loc **Args:** ?a - agent ?loc - location	Agent [?a] toggles on [?loc].
Action: toggle-off-loc **Args:** ?a - agent ?loc - location	Agent [?a] toggles off [?loc].
Action: toggle-on-obj-at-loc **Args:** ?a - agent ?obj - object ?loc - location	Agent [?a] toggles on [?obj] at [?loc].
Action: toggle-off-obj-at-loc **Args:** ?a - agent ?obj - object ?loc - location	Agent [?a] toggles off [?obj] at [?loc].
Action: heat-obj **Args:** ?a - agent ?obj - object ?loc - location	Agent [?a] heats up [?obj] at [?loc].
Action: cool-obj **Args:** ?a - agent ?obj - object ?loc - location	Agent [?a] cools down [?obj] at [?loc].
Action: soak-obj **Args:** ?a - agent ?obj - object ?loc - location	Agent [?a] makes [?obj] soaked at [?loc].
Action: slice-obj **Args:** ?a - agent ?obj - object ?tool - tool ?loc - location	Agent [?a] slices up [?obj] with [?tool] at [?loc].
Action: clean-obj-at-loc **Args:** ?a - agent ?obj - object ?tool - tool ?loc - location	Agent [?a] cleans up [?obj] with [?tool] at [?loc].
Action: clean-loc **Args:** ?a - agent ?tool - tool ?loc - location	Agent [?a] cleans up [?loc] with [?tool].
Action: bring-to-human **Args:** ?r - robot ?obj - object ?h - human	Robot [?r] brings [?obj] to human [?h].
Action: take-from-human **Args:** ?r - robot ?obj - object ?h - human	Robot [?r] takes [?obj] from human [?h].

Table 3: All actions in HandMeThat. If the action schema has an argument of type "agent," the action can be executed by either the human or the robot.

Action Name	Constraints
pick-up put-inside	The receptacle/location must be open if it's openable.
heat cool soak	The agent can only heat things at microwave, oven and stove. The agent can only cool things at refrigerator. The agent can only soak things at sink.
slice	The agent can only slice things with knife.
clean	The agent can only clean things with suitable cleaning tool within rag, dishtowel, hand towel, scrub brush, vacuum, broom.

Table 4: Action constraints in HandMeThat.

Category	Valid Positions
furniture	floor
vessel	countertop table cabinet
tableware	countertop table cabinet dishwasher refrigerator sink
utensil	countertop table cabinet dishwasher refrigerator sink
bag	floor countertop table sofa bed
bucket	floor countertop table
tray	countertop table cabinet refrigerator
basket	floor countertop table shelf cabinet sofa bed
box	floor countertop table shelf cabinet sofa bed
package	floor countertop table shelf cabinet sofa bed
ashcan	floor
xmas tree	floor
xmas stocking	floor countertop table shelf cabinet sofa bed
fruit	table countertop refrigerator utensil
vegetable	table countertop refrigerator stove utensil
drink	table countertop refrigerator cabinet bag
protein	table countertop refrigerator stove utensil
flavorer	table countertop refrigerator cabinet bag
baked food	table countertop refrigerator oven tray
snack	table countertop refrigerator microwave tray
prepared food	table countertop refrigerator microwave tray
metal tool	countertop table cabinet shelf furniture
electric equipment	countertop table cabinet shelf furniture
electrical device	countertop table cabinet shelf furniture
toiletry	cabinet toilet bathtub sink pool bag
writing tool	countertop table cabinet shelf bag
piece of cloth	cabinet toilet bathtub sink pool bucket
cleaning tool	cabinet toilet bathtub sink pool bucket
cleansing	cabinet toilet bathtub sink pool bucket
cutlery	countertop table cabinet dishwasher refrigerator utensil
illumination tool	countertop table sofa bed shelf
decoration	cabinet sofa bed package
paper product	cabinet sofa bed package
footwear	cabinet floor
headwear	cabinet sofa bed package
clothing	cabinet sofa bed package
building materials	pool
plaything	cabinet sofa bed package

Table 5: Valid positions for each category when sampling the initial state.

can replace "?[fruit]" by "apple" and "?[vessel]" by "jar." As a result, each goal templates can be instantiated into a large number of concrete goals.

PDDL setting. We use PDDL (Planning Domain Definition Language) to set up the HandMeThat environment for both human and robot agent. All attributes and relations are represented by predicates with one or two arguments. All the actions are specified with particular preconditions (i.e., constraints) and effects (i.e., semantics). The goal state for each task is represented as the first-order logic statements defined above.

A.3 Quests

We translate the sampled utterance u natural language (English) following the templates in Table 11. Specifically, in the table we have a "Description" function that takes in a set of specifiers and outputs natural language description. The template for this function consists of four parts: static attributes, non-static attributes, object category and current position. These parts are connected coherently. Usually a description in the quest only specifies some of the four parts. Example descriptions are: "the large red box on the table," "the uncooked food," and "the sliced one." The pronoun "one" is used if the specifier does not contain any category information.

A.4 Cost Functions

To compute the utility of the human and to evaluate the performance of agents, we define the following cost functions.

Human's action cost. For simplicity, we assume all human actions are unit-cost. Thus, we use the length of action list as the total cost of a trajectory.

Human's language cost. We use the following cost function to evaluate the complexity of a quest (both meaning and utterance), which is intuitively the number of specifiers in the statement. The specifiers include: 1) category, 2) attributes and relations, 3) target position (only for "move-to"-typed quests). Each specifier in the latter two kinds has cost 1. For example, the specifier"large" costs 1 and "cooked, on-table" costs 2. For category information, each quest can contain at most one specifier of {class, subclass, category}. They cost 1, 2, and 3, respectively. We present more examples for this cost function in Table 12.

Robot's action cost. Similar to human's action costs, we assume all robot actions are unit-costs. There are two exceptions: "examine" and "inventory." Action "examine" gathers the information of objects at the current position. Action "inventory" gathers the information of the current holding object. The costs of these two actions are zero: the agent can obtain such information at no cost and at any time.

A.5 Hyperparameters

In our data generation process, we use the following hyperparameters.

- When generating the meaning m, we need two inverse temperature constants for the Boltzmann distribution: $\beta_1 = 3, \beta_2 = 1.5$.
- When generating the utterance u, we need three constants for the RSA model: $\alpha = 2, \alpha' = 1$ and $k = 10$.
- When evaluating the hardness level, we need to estimate the most probable meaning following the RSA model. In this paper, we use same values of α and k as in generation process.

The hyperparameter $\beta_1, \beta_2, \alpha, \alpha'$ in generating meaning and utterance are chosen to balance the generated data. Intuitively, these values describe the habit in asking for help and language use of a particular human. For example, when α' gets larger, the human tends to give use shorter phrase in utterance (like "hand me that"); otherwise, the human tends to use more detailed description in utterance. We choose the values so that different lengths of meaning and utterance can be generated, ensuring the diversity of data.

Goal Name	First-Order Logic Formula
assembling gift baskets	$\forall y, \exists x, ?[\text{illumination tool}](x) \wedge (\text{basket}(y) \Rightarrow \text{in}(x, y))$ $\forall y, \exists x, ?[\text{snack}](x) \wedge (\text{basket}(y) \Rightarrow \text{in}(x, y))$ $\forall y, \exists x, ?[\text{baked food}](x) \wedge (\text{basket}(y) \Rightarrow \text{in}(x, y))$ $\forall y, \exists x, ?[\text{decoration}](x) \wedge (\text{basket}(y) \Rightarrow \text{in}(x, y))$
bottling fruit	$\exists y_1, y_2, \text{jar}(y_1), \text{jar}(y_2), \ldots$ $\forall x, ?[\text{fruit}]_1(x) \Rightarrow \text{in}(x, y_1)$ $\forall x, ?[\text{fruit}]_2(x) \Rightarrow \text{in}(x, y_2)$
boxing books up for storage	$\exists y, \text{box}(y), \forall x, ?[\text{paper product}](x) \Rightarrow \text{in}(x, y)$
bringing in wood	$\exists y, \text{floor}(y), \forall x, ?[\text{building material}](x) \Rightarrow \text{on}(x, y)$
brushing lint off clothing	$\exists y, \text{bed}(y), \forall x, ?[\text{clothing}](x) \Rightarrow \text{on}(x, y) \wedge \neg\text{dusty}(x)$
chopping vegetables	$\exists y, ?[\text{tableware}](y), \forall x, ?[\text{fruit}]_1(x) \Rightarrow \text{in}(x, y) \wedge \text{sliced}(x)$ $\exists y, ?[\text{tableware}](y), \forall x, ?[\text{fruit}]_2(x) \Rightarrow \text{in}(x, y) \wedge \text{sliced}(x)$ $\exists y, ?[\text{tableware}](y), \forall x, ?[\text{vegetable}]_1(x) \Rightarrow \text{in}(x, y) \wedge \text{sliced}(x)$ $\exists y, ?[\text{tableware}](y), \forall x, ?[\text{vegetable}]_2(x) \Rightarrow \text{in}(x, y) \wedge \text{sliced}(x)$
cleaning bathrooms	$\forall x, \text{toilet}(x) \Rightarrow \neg\text{stained}(x)$ $\forall x, \text{bathtub}(x) \Rightarrow \neg\text{stained}(x)$ $\forall x, \text{sink}(x) \Rightarrow \neg\text{stained}(x)$ $\forall x, \text{floor}(x) \Rightarrow \neg\text{stained}(x)$ $\exists y, \text{bucket}(y), \exists x, \text{rag}(x) \wedge \text{in}(x, y) \wedge \text{soaked}(x)$
cleaning bedrooms	$\exists y, \text{cabinet}(y), \forall x, ?[\text{clothing}](x) \Rightarrow \text{in}(x, y) \wedge \neg\text{dusty}(y)$ $\exists y, \text{cabinet}(y), \forall x, ?[\text{decoration}](x) \Rightarrow \text{in}(x, y)$ $\exists y, \text{cabinet}(y), \forall x, ?[\text{toiletry}](x) \Rightarrow \text{in}(x, y)$ $\exists y, \text{cabinet}(y), \forall x, ?[\text{paper product}](x) \Rightarrow \text{in}(x, y)$
cleaning closet	$\exists y, \text{cabinet}(y), \forall x, ?[\text{decoration}](x) \Rightarrow \text{in}(x, y) \wedge \neg\text{dusty}(y)$ $\exists y, \text{shelf}(y), \forall x, ?[\text{headwear}](x) \Rightarrow \text{on}(x, y) \wedge \neg\text{dusty}(y)$ $\exists y, \text{floor}(y), \forall x, ?[\text{footwear}](x) \Rightarrow \text{on}(x, y) \wedge \neg\text{dusty}(y)$
cleaning floors	$\forall x, \text{floor}(x) \Rightarrow \neg\text{stained}(x) \wedge \neg\text{dusty}(x)$
cleaning garage	$\forall x, \text{floor}(x) \Rightarrow \neg\text{dusty}(x)$ $\forall x, \text{cabinet}(x) \Rightarrow \neg\text{stained}(x)$ $\forall x, \text{cabinet}(x) \Rightarrow \neg\text{dusty}(x)$ $\exists y, \text{ashcan}(y), \forall x, ?[\text{paper product}](x) \Rightarrow \text{in}(x, y)$ $\exists y, \text{table}(y), \forall x, ?[\text{vessel}](x) \Rightarrow \text{on}(x, y)$
cleaning high chair	$\forall x, ?[\text{furniture}](x) \Rightarrow \neg\text{dusty}(x)$
cleaning kitchen cupboard	$\forall x, \text{cabinet}(x) \Rightarrow \neg\text{dusty}(x)$ $\exists y, \text{cabinet}(y), \forall x, ?[\text{tableware}]_1(x) \Rightarrow \text{in}(x, y)$ $\exists y, \text{cabinet}(y), \forall x, ?[\text{tableware}]_2(x) \Rightarrow \text{in}(x, y)$
cleaning microwave oven	$\forall x, \text{microwave}(x) \Rightarrow \neg\text{stained}(x) \wedge \neg\text{dusty}(x)$
cleaning out drawers	$\exists y, \text{sink}(y), \forall x, ?[\text{piece of cloth}](x) \Rightarrow \text{in}(x, y)$ $\exists y, \text{sink}(y), \forall x, ?[\text{tableware}](x) \Rightarrow \text{in}(x, y)$ $\exists y, \text{sink}(y), \forall x, ?[\text{cutlery}](x) \Rightarrow \text{in}(x, y)$
cleaning oven	$\forall x, \text{oven}(x) \Rightarrow \neg\text{stained}(x)$ $\exists x, \text{rag}(x) \wedge \text{soaked}(x)$
cleaning shoes	$\exists y, \text{floor}(y), \forall x, ?[\text{footwear}](x) \Rightarrow \text{on}(x, y)$ $\exists y, \text{floor}(y), \forall x, ?[\text{footwear}](x) \Rightarrow \text{on}(x, y) \wedge \neg\text{dusty}(x)$
cleaning stove	$\forall x, \text{stove}(x) \Rightarrow \neg\text{stained}(x) \wedge \neg\text{dusty}(x)$ $\forall x, \text{rag}(x) \Rightarrow \text{soaked}(x)$ $\forall x, \text{dishtowel}(x) \Rightarrow \text{soaked}(x)$
cleaning table after clearing	$\forall x, \text{table}(x) \Rightarrow \neg\text{stained}(x)$
cleaning the pool	$\forall x, \text{pool}(x) \Rightarrow \neg\text{stained}(x)$ $\exists y, \text{shelf}(y), \forall x, \text{scrub brush}(x) \Rightarrow \text{on}(x, y)$ $\exists y, \text{floor}(y), \forall x, \text{cleansing}(x) \Rightarrow \text{on}(x, y)$

Table 6: The goal names are collected from original BEHAVIOR-100 tasks, and the formulas are the rewritten version for original goal predicates.

Goal Name	First-Order Logic Formula
cleaning toilet	$\forall x, \text{toilet}(x) \Rightarrow \neg\text{stained}(x)$ $\exists y, \text{floor}(y), \forall x, \text{scrub brush}(x) \Rightarrow \text{on}(x, y)$ $\exists y, \text{floor}(y), \forall x, \text{cleansing}(x) \Rightarrow \text{on}(x, y)$
cleaning up after a meal	$\forall x, \text{table}(x) \Rightarrow \neg\text{stained}(x)$ $\forall x, \text{floor}(x) \Rightarrow \neg\text{stained}(x)$ $\forall x, ?[\text{furniture}](x) \Rightarrow \neg\text{dusty}(x)$ $\forall x, ?[\text{tableware}]_1(x) \Rightarrow \neg\text{dusty}(x)$ $\forall x, ?[\text{tableware}]_2(x) \Rightarrow \neg\text{dusty}(x)$ $\forall x, ?[\text{tableware}]_3(x) \Rightarrow \neg\text{dusty}(x)$ $\exists y, ?[\text{bag}](y), \forall x, ?[\text{snack}](x) \Rightarrow \text{in}(x, y)$
cleaning up refrigerator	$\exists y, \text{sink}(y), \forall x, \text{rag}(x) \Rightarrow \text{in}(x, y)$ $\exists y, \text{sink}(y), \forall x, \text{cleansing}(x) \Rightarrow \text{in}(x, y)$ $\exists y, \text{refrigerator}(y), \forall x, \text{tray}(x) \Rightarrow \text{in}(x, y) \wedge \neg\text{dusty}(x) \wedge \neg\text{stained}(y)$ $\exists y, \text{sink}(y), \forall x, ?[\text{tableware}](x) \Rightarrow \text{in}(x, y) \wedge \neg\text{dusty}(x)$
cleaning up the kitchen only	$\exists y, \text{countertop}(y), \forall x, ?[\text{vessel}](x) \Rightarrow \text{on}(x, y)$ $\exists y, \text{sink}(y), \forall x, ?[\text{cleansing}](x) \Rightarrow \text{in}(x, y)$ $\exists y, \text{cabinet}(y), \forall x, ?[\text{flavorer}](x) \Rightarrow \text{in}(x, y)$ $\exists y, \text{cabinet}(y), \forall x, ?[\text{tableware}](x) \Rightarrow \text{in}(x, y) \wedge \neg\text{dusty}(x) \wedge \neg\text{dusty}(y)$ $\exists y, \text{sink}(y), \forall x, \text{rag}(x) \Rightarrow \text{in}(x, y)$ $\exists y, \text{refrigerator}(y), \forall x, ?[\text{utensil}](x) \Rightarrow \text{in}(x, y)$ $\exists y, \text{refrigerator}(y), \forall x, ?[\text{fruit}](x) \Rightarrow \text{in}(x, y)$
clearing the table after dinner	$\exists y_1, \text{bucket}(y_1), \forall x, ?[\text{cutlery}]_1(x) \Rightarrow \text{in}(x, y_1)$ $\exists y_2, \text{bucket}(y_2), \forall x, ?[\text{cutlery}]_2(x) \Rightarrow \text{in}(x, y_2)$ $\exists y, \text{bucket}(y), \forall x, ?[\text{flavorer}](x) \Rightarrow \text{in}(x, y)$
collect misplaced items	$\exists y, \text{table}(y), \forall x, ?[\text{footwear}](x) \Rightarrow \text{on}(x, y)$ $\exists y, \text{table}(y), \forall x, ?[\text{decoration}](x) \Rightarrow \text{on}(x, y)$ $\exists y, \text{table}(y), \forall x, ?[\text{paper product}](x) \Rightarrow \text{on}(x, y)$
collecting aluminum cans	$\exists y, \text{ashcan}(y), \forall x, ?[\text{drink}](x) \Rightarrow \text{in}(x, y)$
filling a Christmas stocking	$\forall y, \exists x, ?[\text{plaything}](x) \wedge (\text{Xmas stocking}(y) \Rightarrow \text{in}(x, y))$ $\forall y, \exists x, ?[\text{snack}](x) \wedge (\text{Xmas stocking}(y) \Rightarrow \text{in}(x, y))$ $\forall y, \exists x, ?[\text{writing tool}](x) \wedge (\text{Xmas stocking}(y) \Rightarrow \text{in}(x, y))$
filling a Easter basket	$\exists y, \text{countertop}(y), \forall x, ?[\text{basket}](x) \Rightarrow \text{on}(x, y)$ $\forall y, \exists x, ?[\text{protein}](x) \wedge (\text{basket}(y) \Rightarrow \text{in}(x, y))$ $\forall y, \exists x, ?[\text{snack}](x) \wedge (\text{basket}(y) \Rightarrow \text{in}(x, y))$ $\forall y, \exists x, ?[\text{paper product}](x) \wedge (\text{basket}(y) \Rightarrow \text{in}(x, y))$ $\forall y, \exists x, ?[\text{plaything}](x) \wedge (\text{basket}(y) \Rightarrow \text{in}(x, y))$ $\exists y, \text{basket}(y), \forall x, ?[\text{decoration}](x) \Rightarrow \text{in}(x, y)$
installing a fax machine	$\exists y, \text{table}(y), \forall x, ?[\text{electric equipment}](x) \Rightarrow \text{on}(x, y)) \wedge \text{toggled}(x)$
installing alarms	$\forall y, \exists x, ?[\text{electrical device}](x) \wedge (\text{table}(y) \Rightarrow \text{on}(x, y))$ $\forall y, \exists x, ?[\text{electrical device}](x) \wedge (\text{countertop}(y) \Rightarrow \text{on}(x, y))$ $\forall y, \exists x, ?[\text{electrical device}](x) \wedge (\text{sofa}(y) \Rightarrow \text{on}(x, y))$
laying tile floors	$\exists y, \text{floor}(y), \forall x, ?[\text{building material}](x) \Rightarrow \text{on}(x, y)$
loading the dishwasher	$\exists y, \text{sink}(y), \forall x, ?[\text{tableware}]_1(x) \Rightarrow \text{in}(x, y)$ $\exists y, \text{sink}(y), \forall x, ?[\text{tableware}]_2(x) \Rightarrow \text{in}(x, y)$ $\exists y, \text{sink}(y), \forall x, ?[\text{vessel}](x) \Rightarrow \text{in}(x, y)$
moving boxes to storage	$\exists y, \text{floor}(y), \forall x, \text{box}(x) \Rightarrow \text{on}(x, y)$
opening packages	$\forall x, \text{package}(x) \Rightarrow \text{open}(x)$
organizing boxes in garage	$\exists y, \text{floor}(y), \forall x, \text{box}(x) \Rightarrow \text{on}(x, y)$ $\exists y_1, y_2, y_3, \text{box}(y_1), \text{box}(y_2), \text{box}(y_3), \ldots$ $\forall x, ?[\text{plaything}](x) \Rightarrow \text{in}(x, y_1)$ $\forall x, ?[\text{cutlery}](x) \Rightarrow \text{in}(x, y_2)$ $\forall x, ?[\text{cleansing}](x) \Rightarrow \text{in}(x, y_3)$

Table 7: (Cont'd.) The goal names are collected from original BEHAVIOR-100 tasks, and the formulas are the rewritten version for original goal predicates.

Goal Name	First-Order Logic Formula
organizing file cabinet	$\exists y, \text{table}(y), \forall x, \text{?[writing tool]}(x) \Rightarrow \text{on}(x, y)$ $\exists y, \text{cabinet}(y), \forall x, \text{?[paper product]}(x) \Rightarrow \text{in}(x, y)$
organizing school stuff	$\exists y, \text{backpack}(y), \ldots$ $\exists x, \text{?[paper product]}(x) \wedge \text{in}(x, y)$ $\exists x, \text{?[writing tool]}_1(x) \wedge \text{in}(x, y)$ $\exists x, \text{?[writing tool]}_2(x) \wedge \text{in}(x, y)$ $\exists x, \text{?[electrical device]}(x) \wedge \text{in}(x, y)$
packing adult's bag	$\exists y, \text{backpack}(y), \ldots$ $\forall x, \text{?[decoration]}(x) \Rightarrow \text{in}(x, y)$ $\exists x, \text{?[toiletry]}_1(x) \wedge \text{in}(x, y)$ $\exists x, \text{?[toiletry]}_2(x) \wedge \text{in}(x, y)$ $\exists x, \text{?[electrical device]}(x) \wedge \text{in}(x, y)$
packing bags or suitcase	$\exists y, \text{briefcase}(y), \ldots$ $\forall x, \text{?[clothing]}(x) \Rightarrow \text{in}(x, y)$ $\exists x, \text{?[toiletry]}(x) \wedge \text{in}(x, y)$ $\exists x, \text{?[cleansing]}_1(x) \wedge \text{in}(x, y)$ $\exists x, \text{?[cleansing]}_2(x) \wedge \text{in}(x, y)$ $\exists x, \text{?[paper product]}(x) \wedge \text{in}(x, y)$
packing boxes for household move or trip	$\exists y_1, y_2, \text{box}(y_1), \text{box}(y_2), \ldots$ $\forall x, \text{?[cutlery]}(x) \Rightarrow \text{in}(x, y_1)$ $\exists x, \text{?[piece of cloth]}(x) \wedge \text{in}(x, y_1)$ $\forall x, \text{book}(x) \Rightarrow \text{in}(x, y_2)$ $\exists x, \text{?[clothing]}(x) \wedge \text{in}(x, y_2)$
packing child's bag	$\exists y, \text{backpack}(y), \ldots$ $\exists x, \text{?[headwear]}(x) \wedge \text{in}(x, y)$ $\exists x, \text{?[decoration]}(x) \wedge \text{in}(x, y)$ $\exists x, \text{?fruit]}(x) \wedge \text{in}(x, y)$ $\exists x, \text{?[paper product]}(x) \wedge \text{in}(x, y)$ $\exists x, \text{?[electrical device]}(x) \wedge \text{in}(x, y)$
packing box for work	$\exists y, \text{box}(y), \ldots$ $\exists x, \text{?[fruit]}(x) \wedge \text{in}(x, y)$ $\exists x, \text{?[drink]}(x) \wedge \text{in}(x, y)$ $\exists x, \text{?[snack]}_1(x) \wedge \text{in}(x, y)$ $\exists x, \text{?[snack]}_2(x) \wedge \text{in}(x, y)$
packing lunches	$\exists y_1, y_2, \text{box}(y_1), \text{box}(y_2), \ldots$ $\exists x, \text{?[snack]}_1(x) \wedge \text{in}(x, y_1)$ $\exists x, \text{?[snack]}_1(x) \wedge \text{in}(x, y_2)$ $\exists x, \text{?[baked food]}_1(x) \wedge \text{in}(x, y_1)$ $\exists x, \text{?[baked food]}_1(x) \wedge \text{in}(x, y_2)$ $\exists x, \text{?[prepared food]}(x) \wedge \text{in}(x, y_1)$ $\exists x, \text{?[snack]}_2(x) \wedge \text{in}(x, y_2)$ $\exists x, \text{?[drink]}_1(x) \wedge \text{in}(x, y_1)$ $\exists x, \text{?[drink]}_2(x) \wedge \text{in}(x, y_2)$ $\exists x, \text{?[fruit]}_1(x) \wedge \text{in}(x, y_1)$ $\exists x, \text{?[fruit]}_2(x) \wedge \text{in}(x, y_2)$
packing picnics	$\exists y_1, y_2, y_3, \text{box}(y_1), \text{box}(y_2), \text{box}(y_3), \ldots$ $\exists x, \text{?[snack]}_1(x) \wedge \text{in}(x, y_1)$ $\exists x, \text{?[snack]}_2(x) \wedge \text{in}(x, y_1)$ $\exists x, \text{?[fruit]}_1(x) \wedge \text{in}(x, y_2)$ $\exists x, \text{?[fruit]}_2(x) \wedge \text{in}(x, y_2)$ $\exists x, \text{?[fruit]}_3(x) \wedge \text{in}(x, y_2)$ $\exists x, \text{?[drink]}_1(x) \wedge \text{in}(x, y_3)$ $\exists x, \text{?[drink]}_2(x) \wedge \text{in}(x, y_3)$ $\exists x, \text{?[drink]}_3(x) \wedge \text{in}(x, y_3)$
picking up take out food	$\exists y, \text{box}(y), \forall x, \text{?[prepared food]}(x) \Rightarrow \text{in}(x, y)$ $\exists y, \text{box}(y), \forall x, \text{?[snack]}(x) \Rightarrow \text{in}(x, y)$ $\exists y, \text{floor}(y), \forall x, \text{?[box]}(x) \Rightarrow \text{on}(x, y) \wedge \text{open}(x)$

Table 8: (Cont'd.) The goal names are collected from original BEHAVIOR-100 tasks, and the formulas are the rewritten version for original goal predicates.

Goal Name	First-Order Logic Formula
picking up trash	$\exists y, \text{ashcan}(y), \forall x, \text{?[paper product]}(x) \Rightarrow \text{in}(x, y)$ $\exists y, \text{ashcan}(y), \forall x, \text{?[drink]}(x) \Rightarrow \text{in}(x, y)$
polishing furniture	$\forall x, \text{table}(x) \Rightarrow \neg\text{dusty}(x)$ $\forall x, \text{shelf}(x) \Rightarrow \neg\text{dusty}(x)$
polishing shoes	$\exists y, \text{sink}(y), \forall x, \text{rag}(x) \Rightarrow \text{in}(x, y) \wedge \text{soaked}(x)$ $\forall x, \text{?[footwear]}(x) \Rightarrow \neg\text{dusty}(x)$
polishing silver	$\exists y, \text{cabinet}(y), \forall x, \text{?[cutlery]}(x) \Rightarrow \text{in}(x, y) \wedge \neg\text{dusty}(x)$ $\exists y, \text{cabinet}(y), \forall x, \text{rag}(x) \Rightarrow \text{in}(x, y)$
preparing food	$\exists y_1, y_2, \text{plate}(y_1), \text{plate}(y_2), \ldots$ $\exists x, \text{?[vegetable]}_1(x) \Rightarrow \text{in}(x, y_1)$ $\exists x, \text{?[vegetable]}_1(x) \Rightarrow \text{in}(x, y_2)$ $\exists x, \text{?[vegetable]}_2(x) \Rightarrow \text{in}(x, y_1)$ $\exists x, \text{?[vegetable]}_2(x) \Rightarrow \text{in}(x, y_2)$ $\exists x, \text{?[vegetable]}_3(x) \Rightarrow \text{in}(x, y_1) \wedge \text{sliced}(x)$ $\exists x, \text{?[vegetable]}_3(x) \Rightarrow \text{in}(x, y_2) \wedge \text{sliced}(x)$ $\exists x, \text{?[fruit]}_1(x) \Rightarrow \text{in}(x, y_1) \wedge \text{sliced}(x)$ $\exists x, \text{?[fruit]}_1(x) \Rightarrow \text{in}(x, y_2) \wedge \text{sliced}(x)$
preserving food	$\exists y, \text{?[vessel]}(y), \forall x, \text{?[fruit]}(x) \Rightarrow \text{in}(x, y) \wedge \text{sliced}(x) \wedge \text{cooked}(x)$ $\exists y, \text{refrigerator}(y), \forall x, \text{?[protein]}(x) \Rightarrow \text{in}(x, y)$
putting away Christmas decorations	$\exists y, \text{cabinet}(y), \forall x, \text{?[decoration]}_1(x) \Rightarrow \text{in}(x, y)$ $\exists y, \text{cabinet}(y), \forall x, \text{?[decoration]}_2(x) \Rightarrow \text{in}(x, y)$ $\exists y, \text{cabinet}(y), \forall x, \text{?[decoration]}_3(x) \Rightarrow \text{in}(x, y)$
putting away Halloween decorations	$\exists y, \text{cabinet}(y), \forall x, \text{?[vegetable]}(x) \Rightarrow \text{in}(x, y)$ $\exists y, \text{cabinet}(y), \forall x, \text{?[illumination tool]}(x) \Rightarrow \text{in}(x, y)$ $\exists y, \text{table}(y), \forall x, \text{?[vessel]}(x) \Rightarrow \text{on}(x, y)$
putting away toys	$\exists y, \text{box}(y), \forall x, \text{?[plaything]}(x) \Rightarrow \text{in}(x, y) \wedge \neg\text{open}(y)$
putting dishes away after cleaning	$\exists y, \text{cabinet}(y), \forall x, \text{?[tableware]}(x) \Rightarrow \text{in}(x, y)$
putting up Christmas decorations inside	$\exists y, \text{table}(y), \forall x, \text{?[illumination tool]}(x) \Rightarrow \text{on}(x, y)$ $\exists y, \text{table}(y), \forall x, \text{?[decoration]}_1(x) \Rightarrow \text{on}(x, y)$ $\exists y, \text{table}(y), \forall x, \text{?[decoration]}_2(x) \Rightarrow \text{on}(x, y)$ $\exists y, \text{table}(y), \forall x, \text{?[decoration]}_3(x) \Rightarrow \text{on}(x, y)$
re-shelving library books	$\exists y, \text{shelf}(y), \forall x, \text{?[paper product]}(x) \Rightarrow \text{on}(x, y)$
serving a meal	$\exists y_1, y_2, y_3, \text{table}(y_1), \text{dish}(y_2), \text{dish}(y_3), \ldots$ $\forall x, \text{dish}(x) \wedge \text{on}(x, y_1)$ $\forall x, \text{bowl}(x) \wedge \text{on}(x, y_1)$ $\forall x, \text{?[cutlery]}_1(x) \wedge \text{on}(x, y_1)$ $\forall x, \text{?[cutlery]}_2(x) \wedge \text{on}(x, y_1)$ $\forall x, \text{?[drink]}(x) \wedge \text{on}(x, y_1)$ $\exists x, \text{?[protein]}(x) \wedge \text{in}(x, y_2)$ $\exists x, \text{?[protein]}(x) \wedge \text{in}(x, y_2)$ $\exists x, \text{?[prepared food]}(x) \wedge \text{in}(x, y_2)$ $\exists x, \text{?[prepared food]}(x) \wedge \text{in}(x, y_3)$ $\exists x, \text{?[baked food]}(x) \wedge \text{in}(x, y_2)$ $\exists x, \text{?[baked food]}(x) \wedge \text{in}(x, y_3)$
serving hors d'oeuvres	$\exists y, \text{table}(y), \forall x, \text{tray}(x) \Rightarrow \text{on}(x, y)$ $\exists y, \text{table}(y), \forall x, \text{?[baked food]}(x) \Rightarrow \text{on}(x, y)$ $\exists y, \text{table}(y), \forall x, \text{?[vegetable]}(x) \Rightarrow \text{on}(x, y)$ $\exists y, \text{table}(y), \forall x, \text{?[prepared food]}(x) \Rightarrow \text{on}(x, y)$
setting up candles	$\forall y, \exists x, \text{?[illumination tool]}(x) \wedge (\text{?[furniture]}(y) \Rightarrow \text{on}(x, y))$
sorting books	$\exists y, \text{shelf}(y), \forall x, \text{?[paper product]}_1(x) \Rightarrow \text{on}(x, y)$ $\exists y, \text{shelf}(y), \forall x, \text{?[paper product]}_2(x) \Rightarrow \text{on}(x, y)$

Table 9: (Cont'd.) The goal names are collected from original BEHAVIOR-100 tasks, and the formulas are the rewritten version for original goal predicates.

Goal Name	First-Order Logic Formula
storing food	$\exists y, \text{cabinet}(y), \forall x, \text{?[prepared food]}(x) \Rightarrow \text{in}(x, y)$ $\exists y, \text{cabinet}(y), \forall x, \text{?[snack]}(x) \Rightarrow \text{in}(x, y)$ $\exists y, \text{cabinet}(y), \forall x, \text{?[flavorer]}_1(x) \Rightarrow \text{in}(x, y)$ $\exists y, \text{cabinet}(y), \forall x, \text{?[flavorer]}_2(x) \Rightarrow \text{in}(x, y)$
storing the groceries	$\exists y, \text{refrigerator}(y), \forall x, \text{?[fruit]}(x) \Rightarrow \text{in}(x, y)$ $\exists y, \text{refrigerator}(y), \forall x, \text{?[protein]}(x) \Rightarrow \text{in}(x, y)$ $\exists y, \text{refrigerator}(y), \forall x, \text{?[vegetable]}_1(x) \Rightarrow \text{in}(x, y)$ $\exists y, \text{refrigerator}(y), \forall x, \text{?[vegetable]}_2(x) \Rightarrow \text{in}(x, y)$
thawing frozen food	$\exists y, \text{sink}(y), \forall x, \text{?[fruit]}(x) \Rightarrow \text{in}(x, y)$ $\exists y, \text{sink}(y), \forall x, \text{?[protein]}(x) \Rightarrow \text{in}(x, y)$ $\exists y, \text{sink}(y), \forall x, \text{?[vegetable]}(x) \Rightarrow \text{in}(x, y)$
throwing away leftovers	$\exists y, \text{ashcan}(y), \forall x, \text{?[snack]}(x) \Rightarrow \text{in}(x, y)$
washing dishes	$\forall x, \text{?[tableware]}_1(x) \Rightarrow \neg\text{dusty}(x)$ $\forall x, \text{?[tableware]}_2(x) \Rightarrow \neg\text{dusty}(x)$ $\forall x, \text{?[tableware]}_3(x) \Rightarrow \neg\text{dusty}(x)$
washing pots and pans	$\exists y, \text{cabinet}(y), \forall x, \text{?[utensil}(x) \Rightarrow \text{in}(x, y) \wedge \neg\text{dusty}(x)$ $\exists y, \text{cabinet}(y), \forall x, \text{?[vessel]}_1(x) \Rightarrow \text{in}(x, y) \wedge \neg\text{dusty}(x)$ $\exists y, \text{cabinet}(y), \forall x, \text{?[vessel]}_2(x) \Rightarrow \text{in}(x, y) \wedge \neg\text{dusty}(x)$

Table 10: (Cont'd.) The goal names are collected from original BEHAVIOR-100 tasks, and the formulas are the rewritten version for original goal predicates.

Utterance	Natural Language Instruction
bring-me(None)	Bring/Hand/Give me that.
bring-me(specifiers)	Bring/Hand/Give me [Description(specifiers)].
move-to(None, None)	Put it over there.
move-to(specifiers, None)	Put [Description(specifiers)] over there.
move-to(None, position)	Put/Move it to the [position].
move-to(specifiers, position)	Put/Move [Description(specifiers)] to the [position].
change-state(None, verb, prep)	[Verb] it [prep].
change-state(specifiers, verb, prep)	[Verb] [prep] [Description(specifiers)].

Table 11: Translation from utterance to natural language instruction. "Description(specifiers)" transfers a set of specifiers to a natural language description. The brackets should be replaced by corresponding values, and the words in parentheses should be removed if that attribute doesn't appear in the specifiers. Moreover, we randomly append "please," "can you" or nothing to the beginning of an instruction.

Quest	Cost
Bring me that.	0
Bring me the *food*.	1
Bring me the *apple* on the *table*.	3+1
Put it over there.	0
Put the *large receptacle* over there.	1+1
Move the *large*, *red box* to the *sofa*.	1+1+2+1
Slice it up.	0
Slice up the one on the *table*.	1
Slice up the *unfrozen apple* on the *table*.	1+3+1

Table 12: Examples for quest costs. For easier understanding, we represent the quests in natural language. Note that when we are generating data, we separately process three types of quests, so we only need to compare the costs within a same quest type.

Action	Observation
(Welcome Message)	Welcome to the world! In the room there is a countertop, a sofa, . . . , a pool. Now you are standing on the floor.
(Previous Trajectory)	The human agent has taken a list of actions towards a goal, which includes: Human moves to the [location]. Human opens [receptacle] at the [location]. Human picks up [object] at the [location]. . .
(Human's utterance)	Human stops and says, "[Natural Language Instruction]." Now it is your turn to help human to achieve the goal!
Examine/Look	You are at the [location]. You see the [state] [location]. In/On [location] you can see [state] [object] [ID], . . . In/On [receptacle] you can see [state] [object] [ID], . . . (First the [location] and then the [receptacle] here.)
Inventory	You are holding nothing / [state] [object] [ID]. Recall your task: [Previous Trajectory], [Human's Utterance].
Move to [location]	You move from [location] to [location].
Pick up [movable]	You pick up the [movable] from [location].
Pick up [movable] from [receptacle]	You pick up the [movable] from the [receptacle] at the [location].
put [movable] into/onto [location]	You put the [movable] into/onto the [location].
put [movable] into/onto [receptacle]	You put the [movable] into/onto the [receptacle] at the [location].
take [movable] from human	You take the [movable] from [location].
give [movable] to human	You give the [movable] to [location].
open/close [receptacle/location]	You open/close the [receptacle/location].
toggle on/off [toggleable]	You toggle the [toggleable] on/off.
heat [cookable]	You heat the [cookable] up with the [location].
cool [freezable]	You cool the [freezable] down with the [location].
soak [soakable]	You make the [soakable] soaked with the [location].
slice [sliceable] with [tool]	You slice up the [sliceable] with the [tool].
clean [cleanable] with [tool]	You clean up the [cleanable] with the [tool].

Table 13: This table presents the natural language templates for actions and corresponding observations in textual interface. Note that the initial observation is not related to player's command, including welcome message and task description.

The hyperparameter $k = 10$ in RSA model is not sensitive to our generation process. It is chosen to be non-trivial while easy to compute. This parameter is used to compute the distribution for utterance generation, and $k = 10$ is already close to the effect of $k \to \infty$.

A.6 Textual Interface

As claimed in the main text, we construct a textual interface for HandMeThat based on a gym-like environment. In this subsection, we present the templates for all valid commands and corresponding observations. The action and observation templates are listed in Table 13.

We have a set of commands for player to move, examine and manipulate objects in the world. The objects in a command should be specified through object identifiers instead of its category name, i.e., they should include IDs.

There is an initial observation when the game starts. It contains the welcome message, previous human actions, examine result at current location, and human's quest statement. Each time when the player enters a command, a new observation is generated, which includes the message of the command's effect. If the command is not valid, the observation will be "You can't do that" or "I can't understand." To reduce the processing cost of neural-network-based agents, we have shorten textual observations, especially for the "examine" results.

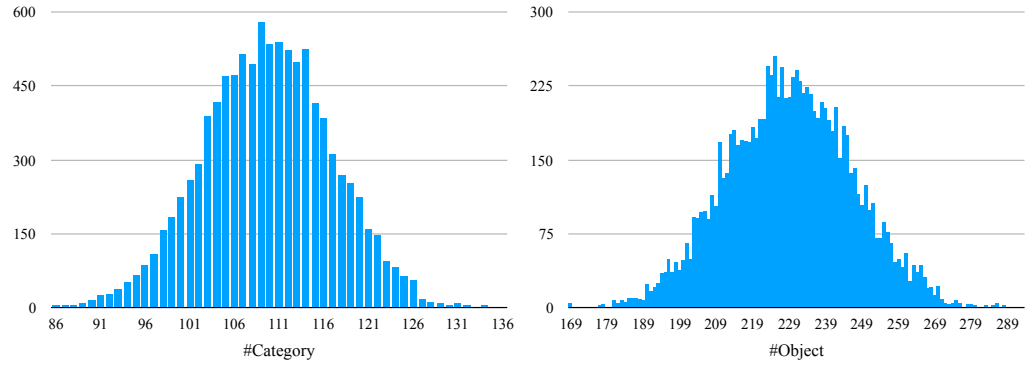

Figure 1: Number of categories and objects in the scene.

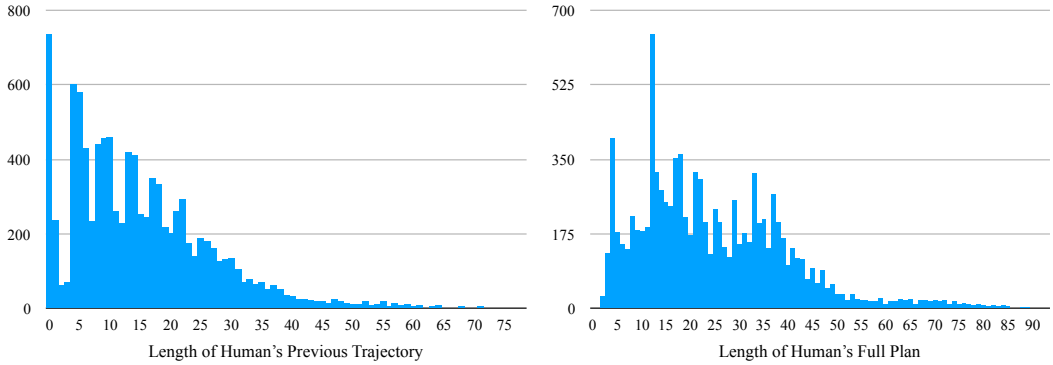

Figure 2: Lengths of human's previous trajectory and original full plan.

B Dataset Statistics

Scene. Fig. 1 shows the number of categories and objects in the sampled scenes. There are 230 objects and 110 categories on average in each scene.

Human actions. Fig. 2(left) shows the length distribution of human's action list before she asks for help, with an average length 15. Fig. 2(right) shows the length distribution of human's original full plan towards her goal, with an average length 25.

Demonstration. For each episode, we provide an expert demonstration that achieves the subgoal. The trajectories are generated by applying greedy best-first search algorithms with FF heuristic on each episode.

In Fig. 3 we show the distribution of lengths of expert trajectories. Most of the demonstrations have 4 or 5 steps since most subgoals follow the pick-and-place template. However, it is important to note that in the partially observable setting, the robot agent should take extra steps to search for the object being referred to, although our expert demonstration generator assumes full information of the environment.

Textual interface. Presented as a text adventure game, HandMeThat has a vocabulary of size 250. We present the distribution of lengths for both fully- and partially-observable setting in Fig. 4. The average length is 860 and 140 separately.

Data split. HandMeThat is composed of 10,000 episodes, and can be classified in three ways. The details are shown in Table 14. *Hardness:* Each hardness level contains 2,500 episodes. *Quest type:* Recall that there are 3 types of subgoals. The "bring-me" and "move-to" subgoals take the dominant part with 4,700 and 4,765 episodes separately, and the remaining 535 episodes are of the

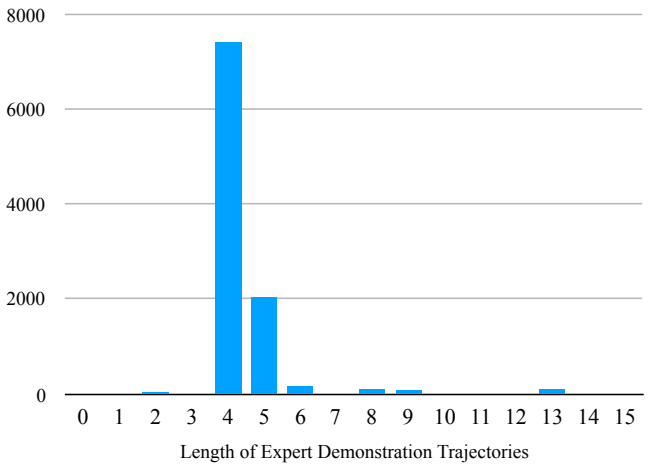

Figure 3: Lengths of trajectories given by expert demonstrations.

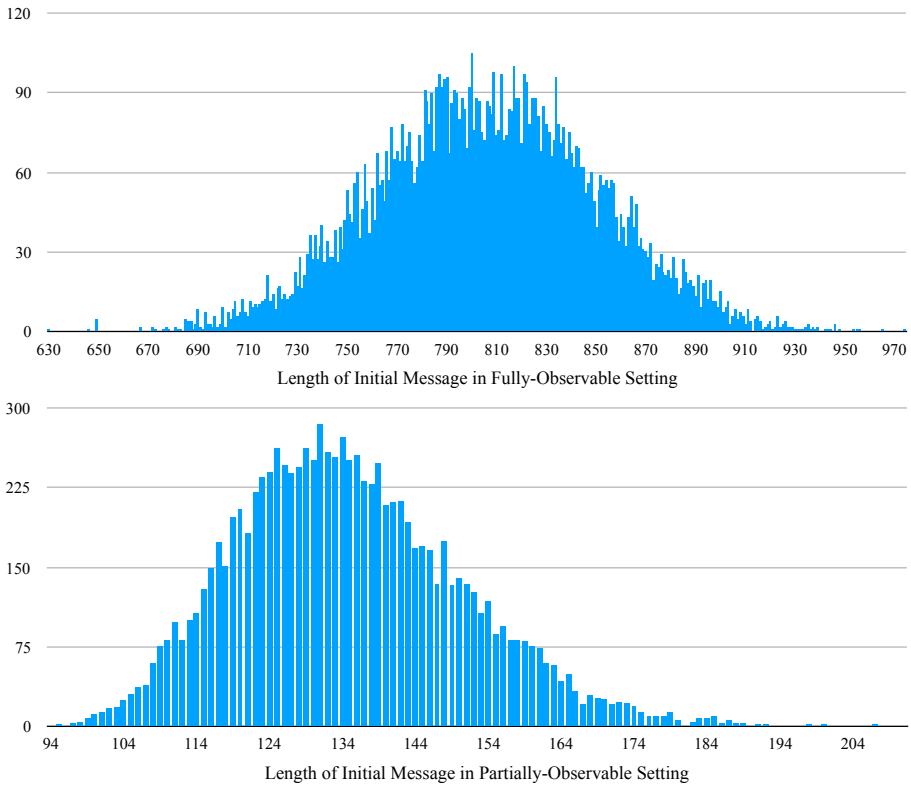

Figure 4: Number of words in the initial observation in both fully- and partially-observable setting.

		Level1	Level 2	Level 3	Level 4
train	bring-me	1351	711	728	965
	move-to	558	1061	1167	1037
	change-state	78	242	80	22
validation	bring-me	174	89	88	123
	move-to	73	117	167	111
	change-state	9	31	12	6
test	bring-me	175	100	84	112
	move-to	69	122	166	117
	change-state	13	27	8	7

Table 14: The 10,000 HandMeThat episodes classified in 3 ways.

"change-state" subgoal. *Training split:* We randomly split 10,000 episodes into three splits: 8,000 for training, 1,000 for validation and 1,000 for test. For each learning method, we train a series of models on the training set, choose the optimal one based on their performances on the validation set and finally, evaluate on the test set.

C Experimental Details

In this section, we explain the details of our baseline models and show the experiment results with error bars. We also present another baseline learning model and discuss potential improvements over the existing baselines.

C.1 Model Details

Seq2Seq model. The Seq2Seq model is based on the sequence-to-sequence model for language modeling, trained with behavior cloning on expert demonstrations [3]. Intuitively, this model learns to predict a list of tokens (action string) given the observation tokens of historical steps. We have largely followed the model design in the ALFworld environment [4].

Specifically, let obs_0 be the initial observation tokens, $a_0, obs_1, a_1 \dots a_{k-1}, obs_k$ be the trajectory of historical interactions. a_i and obs_i are the action command and the observation at step i, respectively. Here, obs_0 includes the welcome message and the initial "examine" results, i.e., all object states in the fully-observable setting, or objects at the current location of the robot in the partially-observable setting; obs_i includes the effect of command a_i (for the partially-observable setting, it also includes the examine results at the current location). There is also a sequence of tokens called *task_desc* which contains human's previous actions as well as the language instruction. At each step, the input to the model includes *task_desc* and the sequence "obs_0 [SEP] a_0 [SEP] $obs_1 \dots a_{k-1}$ [SEP] obs_k," where "[SEP]" is the separator symbol in the BERT Tokenizer [5]. The model encodes the two parts of the input and fuse the encoded feature. The output of the model is a_k, the next action command.

DRRN model. The DRRN model is a choice-based text-game agent based on Deep Reinforcement Relevance Network (DRRN), which learns a Q function for possible state-action pairs. Concretely, the model 1) uses a GNU to encode a valid action; 2) uses separate GNUs to encode last observation, current inventory, and current examine results; 3) concatenates the above parts as the input for network; 4) outputs an estimation of Q value. Note that we include the task description information (human's past actions and the utterance) in the "inventory" part. The model sequentially chooses the best action based on the estimated Q values.

The network is trained by a batch of transitions tuples $(obs, a, r, obs', A_{valid}(s'))$, where obs, obs' are the observations before and after action a, r is the reward, and $A_{valid}(s')$ is the set of valid actions for next state s'. The training objective is to minimize $\|r + \gamma \max_{a'} Q(obs', a') - Q(obs, a)\|$.

Offline-DRRN model. Here we present an offline variation of the DRRN model. Instead of actively collecting environmental trajectories based on the current policy (with epsilon-greedy exploration), we train the Q function network with expert demonstration trajectories. Intuitively, this offline

version only trains on the trajectories that can lead to positive rewards in a few steps. We show the performance of this model in the extended results below.

C.2 Results.

We evaluate all models on both fully-observable and partially-observable settings. We consider three evaluation metrics: 1) the average score of the model; 2) the success rate that the model achieves the goal within limited steps (40); 3) the average number of moves of successful episodes. All three scores are averaged on 1,000 episodes in the test splits.

Table 15 and Table 16 gives the experiment results for all baseline models with error bars. These results are the average values and standard deviation over three runs. Overall, the updated error bars do not change the conclusions and analysis in the main paper.

Analysis of the Heuristic model. The heuristic model leverages additional groundtruth information that all learning models do not have. Specifically, the heuristic model works on the symbolic state composed of object properties and relations. It also has a built-in planner to navigate to and manipulate objects. The only decision made by the model is to choose the proper objects to manipulate. The heuristic model performs well because the "repeating human's action" strategy is a good heuristic for using human history trajectory. To fairly compare it with the other baselines, we only allow the model to perform in a one-trial manner, i.e., the model could only guess once about the specified subgoal and then follow the plan towards it.

Analysis of the Offline-DRRN model. Unfortunately, the offline-DRRN model also fails on HandMeThat. Although the training loss decreases quickly, the model encounters a Q-value overestimation problem. Due to the generation procedure of expert demonstrations, all trajectories are successful. As a result, the model tends to attribute the success only to the last action in each trajectory, which is typically "put-down," "heat," "cool," etc. Empirically, these actions have high Q-values at all states. Thus, at performance time, the robot put the objects down immediately after picking up, and tends to move to fridge and microwave (in order to execute cooling and heating actions). These issues can be potentially addressed by considering uncertainty of unseen actions in expert demonstrations (e.g., through Conservative Q-Learning [6]), but are beyond the scope of this paper.

Reward shaping on DRRN model. As an extension, we perform additional reward shaping for the DRRN baseline based on expert demonstrations. Specifically, we assign positive rewards for the actions within the expert demonstrations. The i-th action in the sequence is assigned with extra reward $10i$. In the training phase, the robot can receive the corresponding reward if it has taken all the previous actions in the expert demonstration and then chooses the correct next action. Intuitively, this method encourages the RL agent to follow the optimal action sequence. However, training with this modified reward function does not improve the overall performance. That is, the overall success rate is still 0.00% for all the settings. When we visualize the predicted actions of the agent, we find that in many cases the models manage to predict the first or second action correctly, but they still fail to complete the whole task. We believe that a better design in integrating imitation learning and reinforcement learning [7] is needed to tackle with HandMeThat tasks.

D Extension: Human-Robot Interaction

When the provided information is limited (e.g., in the hardness level 4), it is possible to extend the current framework to allow robot agents to ask human questions. Specifically, we can allow robots to ask questions about additional specifiers of the referred objects. As a preliminary version, we list all the valid questions in Table 17. These questions query the value of a single attribute, and we can simulate the human agent response using the corresponding attribute value of the internal meaning representation m. When the attribute is not specified in the meaning m, the human agent may return "either is fine" or similar answers.

Note that there should be a balance between environmental attempts and question asking. As a formalization, we can set the question cost to be a non-zero constant per specifier in the textual interface. During evaluation, the number of asked questions can also be evaluated separately.

Due to the poor performances of all learning-based models on original HandMeThat setting, we do not describe this additional human-robot interaction interface in the main text. However, this is a

Model	Fully Observable			
	Level 1	Level 2	Level 3	Level 4
Human	N/A 92 4.4	N/A 80 4.8	N/A 36 4.3	N/A 16 4.5
Random	-40.0 0.0 N/A	-39.8 0.1 30.0	-40.0 0.0 N/A	-40.0 0.0 N/A
Heuristic (1-trial)	88.9 94.9 4.2	-1.9 28.1 4.4	-23.7 12.0 4.3	-15.9 17.8 4.4
Seq2Seq	-9.49 ± 0.47 22.44 ± 0.34 4.03 ± 0.01	-30.74 ± 0.63 6.83 ± 0.46 4.29 ± 0.04	-34.56 ± 0.17 4.01 ± 0.13 4.29 ± 0.01	-32.51 ± 0.00 5.51 ± 0.00 4.13 ± 0.02
Seq2Seq+goal	-15.71 ± 0.87 17.90 ± 0.64 4.30 ± 0.05	-25.83 ± 0.00 10.44 ± 0.00 4.31 ± 0.03	-30.54 ± 0.42 6.98 ± 0.31 4.38 ± 0.03	-29.07 ± 0.47 8.05 ± 0.34 4.04 ± 0.03
Seq2Seq+subgoal	-4.86 ± 0.22 25.88 ± 0.16 4.20 ± 0.00	-22.01 ± 0.00 13.25 ± 0.00 4.20 ± 0.01	-21.03 ± 0.00 13.95 ± 0.00 4.03 ± 0.00	-19.29 ± 0.00 15.25 ± 0.00 4.22 ± 0.00
DRRN	-39.82 ± 0.18 0.13 ± 0.13 5.00 ± 0.00	-39.83 ± 0.17 0.13 ± 0.13 10.00 ± 0.00	-39.83 ± 0.17 0.13 ± 0.13 9.00 ± 0.00	-40.00 ± 0.00 0.00 ± 0.00 N/A
offline-DRRN	-40.00 ± 0.00 0.00 ± 0.00 N/A	-40.00 ± 0.00 0.00 ± 0.00 N/A	-40.00 ± 0.00 0.00 ± 0.00 N/A	-40.00 ± 0.00 0.00 ± 0.00 N/A

Table 15: Experiment results in the fully observable setting. Each model is evaluated on 4 hardness levels with 3 metrics, presented as three values in each cell: 1) the average score, 2) the success rate (%), and 3) the average number of moves in successful episodes. The results for learning models are the mean and standard error values over three runs.

promising extension because the real-world environments are not restricted to one single episode and it also introduces new machine learning challenges including generating helpful questions to human and learning to trade-off between question asking and environmental explorations.

E Datasheets For Dataset

In this section, we present the documentation for HandMeThat following Datasheets for Datasets [1] standards.

E.1 Motivation

- **For what purpose was the dataset created?** To study the interpretation of human's natural language based on both physical and social contexts.
- **Who created the dataset?** The authors listed on this paper, which include researchers from from Tsinghua University and MIT.
- **Who funded the creation of the dataset?** This work is in part supported by ONR MURI N00014-16-1-2007, the Center for Brain, Minds, and Machines (CBMM, funded by NSF STC award CCF-1231216), the MIT Quest for Intelligence, and the MIT-IBM AI Lab. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of our sponsors.

Model	Partially Observable			
	Level 1	Level 2	Level 3	Level 4
Human	N/A 76 5.1	N/A 52 4.8	N/A 20 5.8	N/A 8 5.0
Random	-40.0 0.0 N/A	-39.8 0.1 16.0	-40.0 0.0 N/A	-40.0 0.0 N/A
Seq2Seq	-5.39 ± 0.21 25.49 ± 0.16 4.19 ± 0.01	-25.81 ± 0.45 10.44 ± 0.89 4.31 ± 0.02	-34.73 ± 0.43 3.88 ± 0.31 4.05 ± 0.04	-32.80 ± 0.71 5.30 ± 0.52 4.08 ± 0.01
Seq2Seq+goal	-11.45 ± 0.00 21.01 ± 0.00 4.13 ± 0.02	-28.80 ± 0.67 8.24 ± 0.49 4.05 ± 0.00	-30.01 ± 0.00 7.36 ± 0.00 4.32 ± 0.04	-32.51 ± 0.71 5.51 ± 0.52 4.04 ± 0.03
Seq2Seq+subgoal	-10.12 ± 0.65 21.99 ± 0.48 4.09 ± 0.00	-18.72 ± 0.45 15.66 ± 0.33 4.12 ± 0.02	-24.46 ± 0.22 11.44 ± 0.16 4.13 ± 0.00	-21.87 ± 0.24 13.35 ± 0.18 4.11 ± 0.01
DRRN	-40.00 ± 0.00 0.00 ± 0.00 N/A	-40.00 ± 0.00 0.00 ± 0.00 N/A	-40.00 ± 0.00 0.00 ± 0.00 N/A	-40.00 ± 0.00 0.00 ± 0.00 N/A
offline-DRRN	-40.00 ± 0.00 0.00 ± 0.00 N/A	-40.00 ± 0.00 0.00 ± 0.00 N/A	-40.00 ± 0.00 0.00 ± 0.00 N/A	-40.00 ± 0.00 0.00 ± 0.00 N/A

Table 16: Experiment results in the partially observable setting. Each model is evaluated on 4 hardness levels with 3 metrics, presented as three values in each cell: 1) the average score, 2) the success rate (%), and 3) the average number of moves in successful episodes. The results for learning models are the mean and standard error values over three runs.

Valid Questions	**Example for Answers**
Which type do you mean?	I mean the apple. I just want a fruit. I just want a food.
Which color do you like? Which size do you like?	The red one. The large one.
Where is the object you want? Where do you want to place it?	On top of the table. Put it on the countertop.
Do you want a dusty/cooked/frozen/ sliced/toggled/soaked/open one?	Yes, I mean the dusty one. No, I mean the no dusty one.
Can you say it clearly?	I mean the sliced apple on the table.
(No specified answer.)	Not any specific type/color/... Either is fine.

Table 17: All the valid questions in our textual interface and the corresponding answers.

E.2 Composition

- **What do the instances that comprise the dataset represent and how many instances are there?** Each instance in HandMeThat is an episode of a human-robot communication scene. Concretely, it provides both the physical information in the environment, and the social information which includes human's previous trajectory and her natural language instruction. There are 10,000 episodes in our dataset.
- **Does the dataset contain all possible instances or is it a sample (not necessarily random) of instances from a larger set?** It is a sampled subset.
- **Is any information missing from individual instances?** No.
- **Are relationships between individual instances made explicit?** There is no relationship between instances. Each episode is generated individually by code.
- **Are there recommended data splits?** We include the data splits for training, validation test. We also include the data splits for four hardness levels, and for three quest types.
- **Are there any errors, sources of noise, or redundancies in the dataset?** Because our generation process is based on code, we do not expect errors in the dataset.
- **Is the dataset self-contained, or does it link to or otherwise rely on external resources?** The dataset is self-contained.
- **Does the dataset contain data that might be considered confidential?** No.
- **Does the dataset contain data that, if viewed directly, might be offensive, insulting, threatening, or might otherwise cause anxiety?** No.
- **Does the dataset relate to people?** No.

E.3 Collection Process

- **How was the data associated with each instance acquired? What mechanisms or procedures were used to collect the data?** We design the object space and goal space based on BEHAVIOR-100, simplifying its object hierarchy and summarizing the goal templates. Then all the instances are collected through the code of generation process we introduce in the main paper.
- **Who was involved in the data collection process?** Students in the author list.
- **Over what timeframe was the data collected?** The data is collected in May 2022.
- **Were any ethical review processes conducted?** N/A.
- **Does the dataset relate to people?** No.

E.4 Preprocessing/cleaning/labeling

- **Was any preprocessing/cleaning/labeling of the data done?** N/A. Our dataset is synthetically generated.
- **Was the "raw" data saved in addition to the preprocessed/cleaned/labeled data?** N/A. Our dataset is synthetically generated.
- **Is the software used to preprocess/clean/label the instances available?** N/A. Our dataset is synthetically generated.

E.5 Uses

- **Has the dataset been used for any tasks already?** Yes, in our main paper we discuss the challenge of interpreting human's utterance based on accessible information.
- **Is there a repository that links to any or all papers or systems that use the dataset?** No other papers have used HandMeThat yet.
- **What (other) tasks could the dataset be used for?** The dataset can be used for simulating the situation that one agent is asking for help in collaborative tasks.

- **Is there anything about the composition of the dataset or the way it was collected and preprocessed/cleaned/labeled that might impact future uses?** No.
- **Are there tasks for which the dataset should not be used?** No.

E.6 Distribution

- **Will the dataset be distributed to third parties outside of the entity (e.g., company, institution, organization) on behalf of which the dataset was created?** Yes. It will be publicly available.
- **How will the dataset will be distributed (e.g., tarball on website, API, GitHub)?** We will distribute the data on GitHub.
- **When will the dataset be distributed?** The data will be publicly available along with the camera-ready version of the paper.
- **Will the dataset be distributed under a copyright or other intellectual property (IP) license, and/or under applicable terms of use (ToU)?** We bear all responsibility in case of violation of rights.
- **Have any third parties imposed IP-based or other restrictions on the data associated with the instances?** No.
- **Do any export controls or other regulatory restrictions apply to the dataset or to individual instances?** No.

E.7 Maintenance

- **Who will be supporting/hosting/maintaining the dataset?** The authors of the paper will be maintaining the dataset.
- **How can the owner/curator/manager of the dataset be contacted?** The correspondence author of the main paper can be contacted via email listed in the main paper.
- **Will the dataset be updated?** Yes. The authors will be maintaining and updating the dataset upon requests and for other research purposes.
- **If the dataset relates to people, are there applicable limits on the retention of the data associated with the instances?** N/A.
- **Will older versions of the dataset continue to be supported/hosted/maintained?** Yes.
- **If others want to extend/augment/build on/contribute to the dataset, is there a mechanism for them to do so?** Yes. Other contributors are welcomed to submit pull requests.

References

[1] Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumé III, and Kate Crawford. Datasheets for datasets. *Commun. ACM*, 64(12):86–92, nov 2021. 1, 17

[2] Richard E Fikes and Nils J Nilsson. Strips: A new approach to the application of theorem proving to problem solving. *AIJ*, 2(3-4):189–208, 1971. 3

[3] Michael Bain and Claude Sammut. A framework for behavioural cloning. In *Machine Intelligence*, pages 103–129, 1995. 15

[4] Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. ALFWorld: Aligning Text and Embodied Environments for Interactive Learning. In *ICLR*, 2021. 15

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, 2019. 15

[6] Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. In *NeurIPS*, 2020. 16

[7] Stéphane Ross, Geoffrey J. Gordon, and J. Andrew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *AISTATS*, 2011. 16