MANCHESTER
1824
The University of Manchester

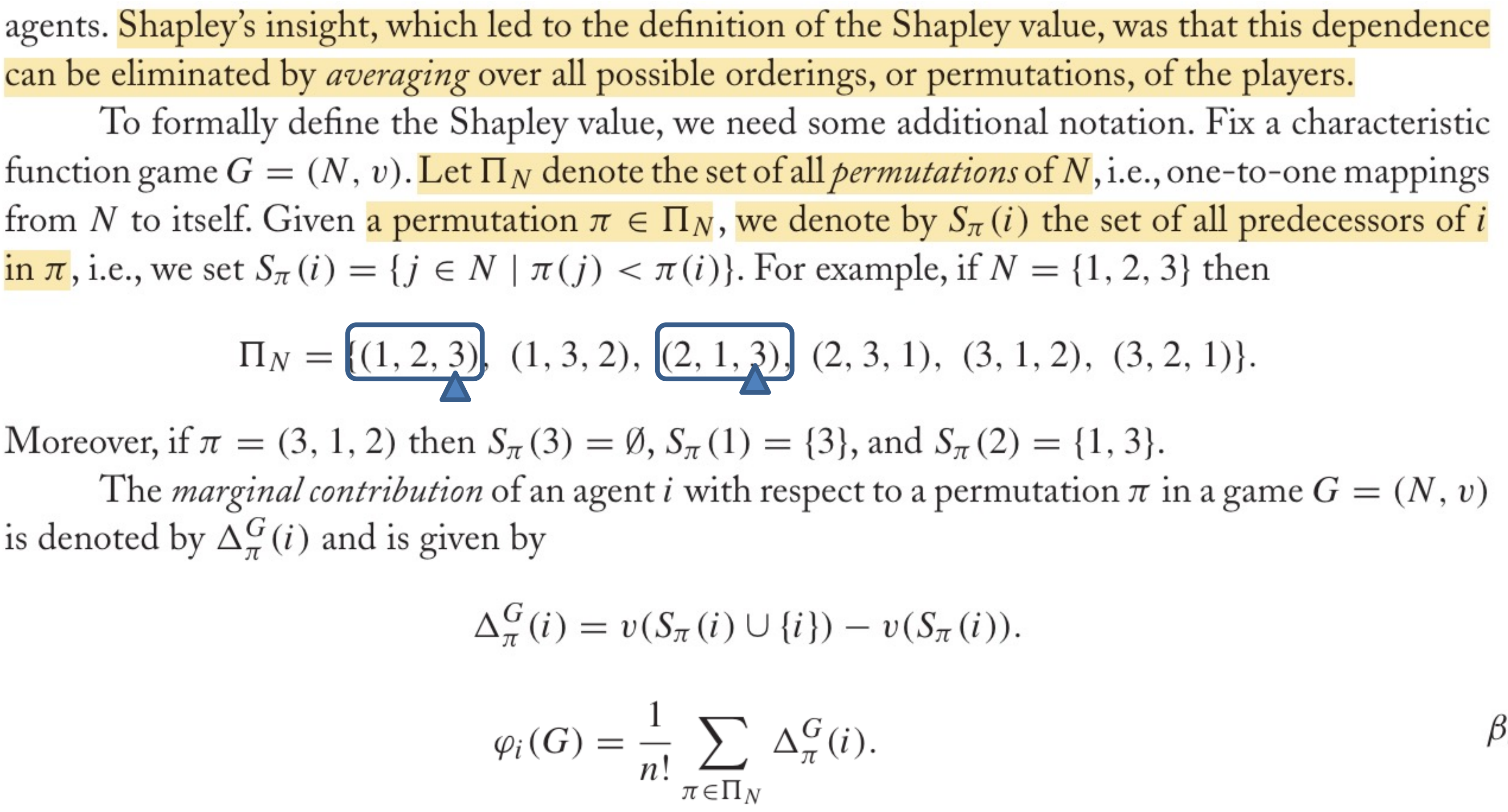
Robotics-LLM Reading Party

- “No, to the Right” – Online Language Corrections for Robotic Manipulation via Shared Autonomy
- Large Language Models as Zero-Shot Human Models for Human-Robot Interaction

Yang Li, PhD student
University of Manchester

1. The Shapley Value may not be the optimal theoretical framework for addressing this problem for several reasons:

- While the Shapley value distributes contributions throughout the grand coalition by considering all possible permutations of players, our problem doesn't prioritize the order of strategies. This may lead to decreased efficiency.



agents. Shapley's insight, which led to the definition of the Shapley value, was that this dependence can be eliminated by *averaging* over all possible orderings, or permutations, of the players.

To formally define the Shapley value, we need some additional notation. Fix a characteristic function game $G = (N, v)$. Let Π_N denote the set of all *permutations* of N, i.e., one-to-one mappings from N to itself. Given a permutation $\pi \in \Pi_N$, we denote by $S_\pi(i)$ the set of all predecessors of i in π, i.e., we set $S_\pi(i) = \{j \in N \mid \pi(j) < \pi(i)\}$. For example, if $N = \{1, 2, 3\}$ then

$$\Pi_N = \{\boxed{(1, 2, 3)}, (1, 3, 2), \boxed{(2, 1, 3)}, (2, 3, 1), (3, 1, 2), (3, 2, 1)\}.$$

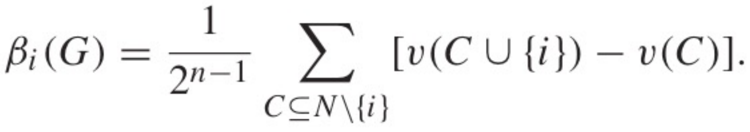
Moreover, if $\pi = (3, 1, 2)$ then $S_\pi(3) = \emptyset$, $S_\pi(1) = \{3\}$, and $S_\pi(2) = \{1, 3\}$.

The *marginal contribution* of an agent i with respect to a permutation π in a game $G = (N, v)$ is denoted by $\Delta_\pi^G(i)$ and is given by

$$\Delta_\pi^G(i) = v(S_\pi(i) \cup \{i\}) - v(S_\pi(i)).$$

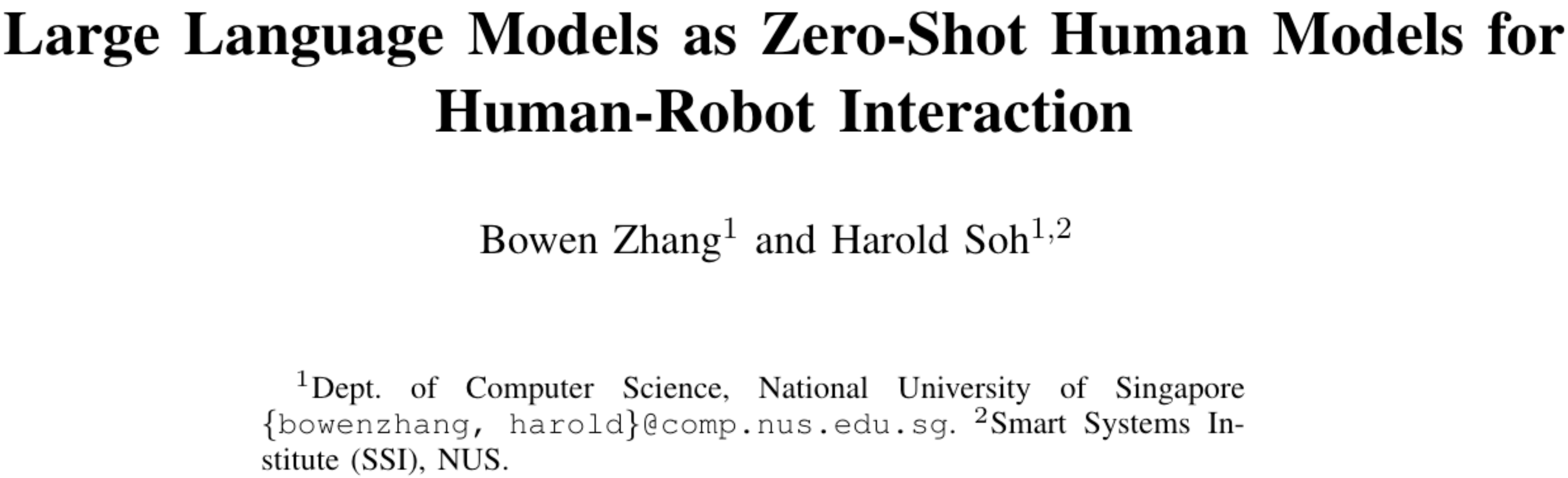
$$\varphi_i(G) = \frac{1}{n!} \sum_{\pi \in \Pi_N} \Delta_\pi^G(i).$$

Shapley Value



$$\beta_i(G) = \frac{1}{2^{n-1}} \sum_{C \subseteq N \setminus \{i\}} [v(C \cup \{i\}) - v(C)].$$

Banzhaf index



Large Language Models as Zero-Shot Human Models for Human-Robot Interaction

Bowen Zhang[1] and Harold Soh[1,2]

[1]Dept. of Computer Science, National University of Singapore {bowenzhang, harold}@comp.nus.edu.sg. [2]Smart Systems Institute (SSI), NUS.

[2303.03548] Large Language Models as Zero-Shot Human Models for Human-Robot Interaction (arxiv.org)

Background

Accurate human modelling remains a significant challenge for human-robot interaction:

- Handcrafted human model: strong assumptions → limit model flexibility and challenging to scale up to real-world settings.
- Non-parametric data-driven model: human interaction data

Can LLMs function effectively as human models for HRI?

- Authors first present an empirical study that shows that LLMs indeed well-capture human latent states and behavior (test 3 datasets on two SOTA LLMs)
- A deeper analysis of our results shows that the LLMs do not work well on tasks that require spatial and numerical reasoning, and are sensitive to prompt syntax.
- This can make application in real-world HRI scenarios challenging and **suggest that LLMs may be best used as "task-level" human models.**

Related Work

Human models for HRI:

1. Theory-of-mind (ToM): ToM models broadly refer to methods that incorporate a set of assumptions about human mental processing and behavior.
    - Bayesian ToM assumes humans behave rationally and update their beliefs in a Bayesian manner
2. Black-box data-driven models: make few assumptions and model human behavior in a data-driven manner
    - MIRROR: models a human based on a robot’s internal self-model (trained with RL) and the uses a small amount of human data to adapt the model to a particular individual.

LLMs as Zero-Shot Human Models

Aim: evaluate the performance of LLMs on a set of social inference tasks --- Given information about a context/scenario, aim to predict either a human's behavior or state.

Problem Statement: Seek to model the distribution $p(z_t^H | s_t, h_t)$, where. $z_t^H \in \mathcal{Z}^H$ is a specific property of one of the human agents $a^H \in \mathcal{A}$, h is history of interactions. More precisely, z is an abstract scenario-specific random variable.

Methods: Omit for too NLP

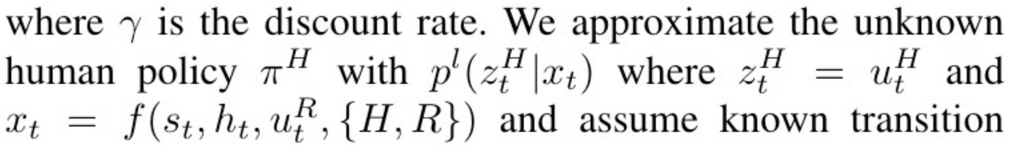
Results:

LLMs can be effective human models for HRI without further training or fine-tuning.
LLMs can perform poorly on HRI tasks that require spatial/physical/numerical reasoning
LLMs are sensitive to the prompt structure.
our results suggest that LLMs are better-suited as task-level (symbolic) human models, and alternative "low-level" models may be needed to account for geometry and motions in a continuous space.

Planning with LLM-Based Human models

Problem Statement:

$$\pi_*^R = \arg\max_{\pi_R} \mathop{\mathbb{E}}_{u_t^H \sim \pi^H, u_t^R \sim \pi^R} \sum_{t=0}^{\infty} \gamma^t r(s_t, u_t^R, u_t^R)$$



where γ is the discount rate. We approximate the unknown human policy π^H with $p^l(z_t^H|x_t)$ where $z_t^H = u_t^H$ and $x_t = f(s_t, h_t, u_t^R, \{H, R\})$ and assume known transition

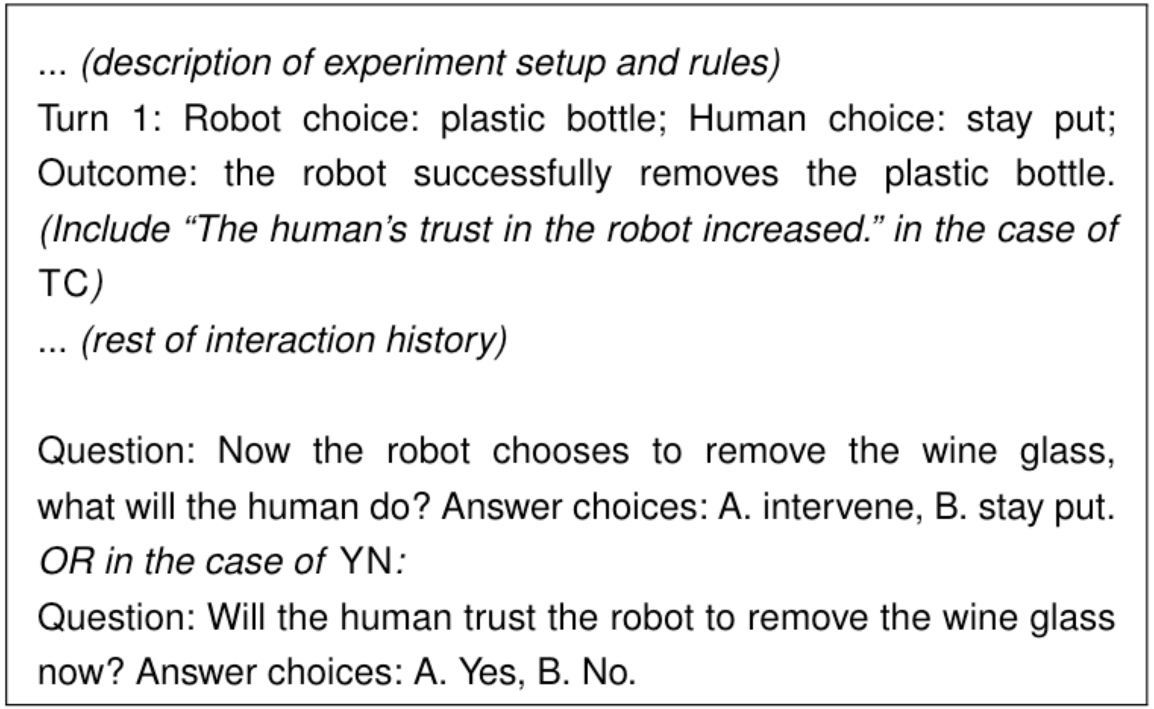
Experiment 1: table-clearing

a human and a robot collaborate to clear objects off a table.

- The objects include three water bottles, one fish can, and one wine glass.
- At each time step, the robot chooses one of the objects to remove
- The human then chooses whether to intervene and pick up the object, or stay put and let the robot remove the object by itself.
- If the human stays put and the robot succeeds, they will get a reward based on the object: 1 for plastic bottle, 2 for fish can, and 3 for wine glass.
- if they stay put and the robot fails, they will receive a penalty: no penalty for plastic bottle, 4 for fish can and 9 for wine glass.
- If they choose to intervene, they will receive no reward or penalty.
- It is assumed that the robot will never fail but this information is not revealed to the human participant.

Planning with LLM-Based Human models

Experiment 1: table-clearing



... *(description of experiment setup and rules)*

Turn 1: Robot choice: plastic bottle; Human choice: stay put; Outcome: the robot successfully removes the plastic bottle. *(Include "The human's trust in the robot increased." in the case of* TC*)*

... *(rest of interaction history)*

Question: Now the robot chooses to remove the wine glass, what will the human do? Answer choices: A. intervene, B. stay put.

OR in the case of YN*:*

Question: Will the human trust the robot to remove the wine glass now? Answer choices: A. Yes, B. No.

Fig. 4: Example prompt used in table-clearing experiment.

- TC: We explicitly include the Trust Change in each turn (the most likely post-observation trust-change predicted by the LLM model) using a multiple-choice question with options {increased, decreased, unchanged}
- YN: Instead of asking which action the human will take when the robot chooses to remove an object, the prompt asks a Yes-No question about whether the human will trust the robot to do so. We assume a deterministic relationship between trust and the human action, i.e., the human will intervene if they do not trust the robot to perform the task and stay-put otherwise.

Planning with LLM-Based Human models

Experiment 1: table-clearing

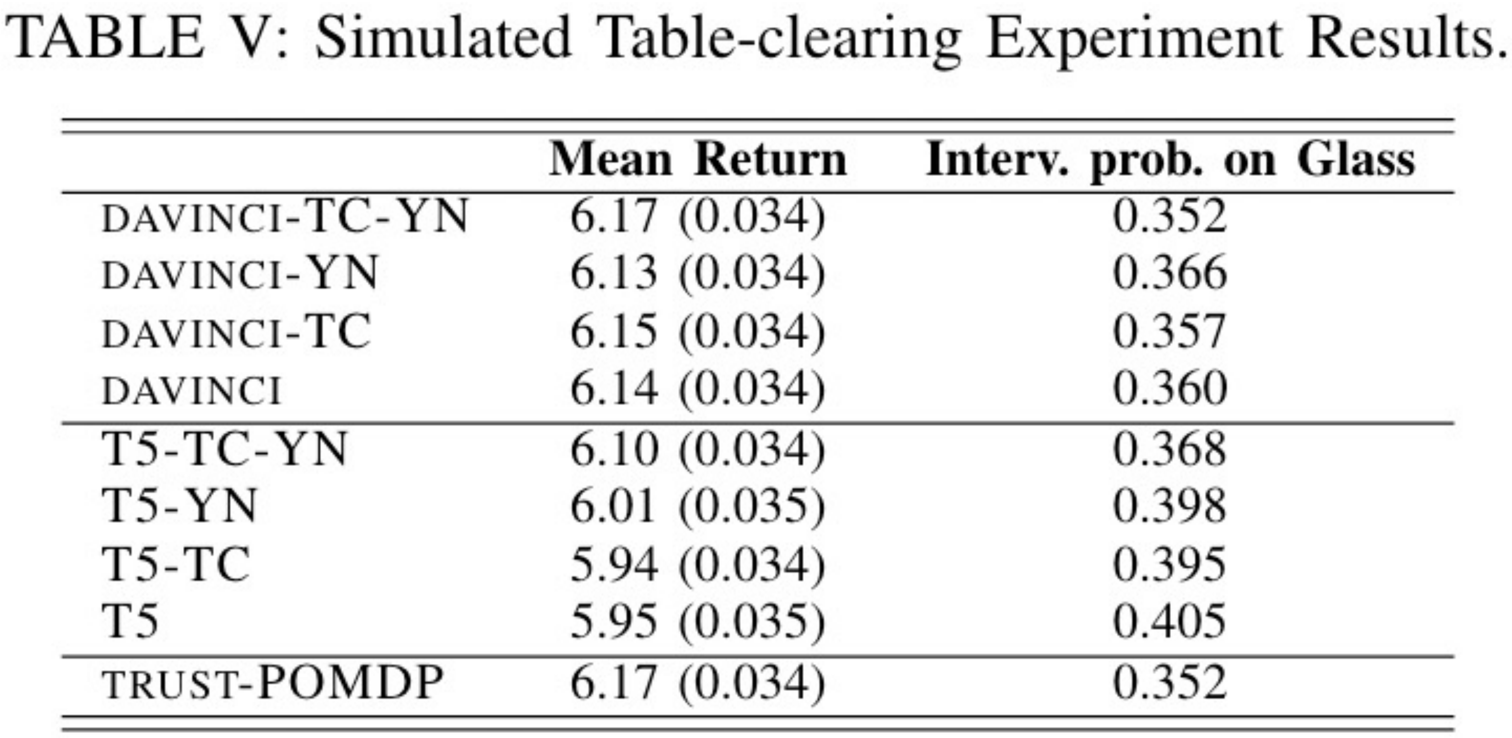


TABLE V: Simulated Table-clearing Experiment Results.

	Mean Return	Interv. prob. on Glass
DAVINCI-TC-YN	6.17 (0.034)	0.352
DAVINCI-YN	6.13 (0.034)	0.366
DAVINCI-TC	6.15 (0.034)	0.357
DAVINCI	6.14 (0.034)	0.360
T5-TC-YN	6.10 (0.034)	0.368
T5-YN	6.01 (0.035)	0.398
T5-TC	5.94 (0.034)	0.395
T5	5.95 (0.035)	0.405
TRUST-POMDP	6.17 (0.034)	0.352

Planning with LLM-Based Human models

Experiment 2: Utensil-passing Experiment

- In this scenario, a human is washing utensils in a kitchen and a robot (a Franka Emika Panda robot arm) is helping to pass dirty utensils to them.
- The objects include a spatula, an egg whisk, a pair of scissors, and a knife.
- At each time step, the robot chooses one of the objects to pass. The human then chooses between two actions:
  - (A) intervene and retrieve the object by themselves or
  - (B) stay put and wait for the robot to pass it.
- if the human stays put and the robot succeeds, they receive a reward of 1.
- Since the utensils are dirty, the handover is only considered successful if the robot passes the object in a manner that the human can easily grasp the clean handle.
- If the human stays put and the robot fails, they receive a penalty of −1.
- If the human chooses to intervene, they will receive no reward or penalty.
- The following information is not revealed to the participant: the robot is able to always succeed on the spatula, whisk, and scissors, but it may fail to properly hand over the knife and accidentally drop it.
  - If the knife is dropped, the experiment is terminated and a penalty of −10 is received.
- intentionally fail on all utensils (except the knife) by handing the wrong part to the human (so that the human can only grasp the dirty end. This intentional failure results in a penalty of −1

Planning with LLM-Based Human models

Experiment 2: Utensil-passing Experiment

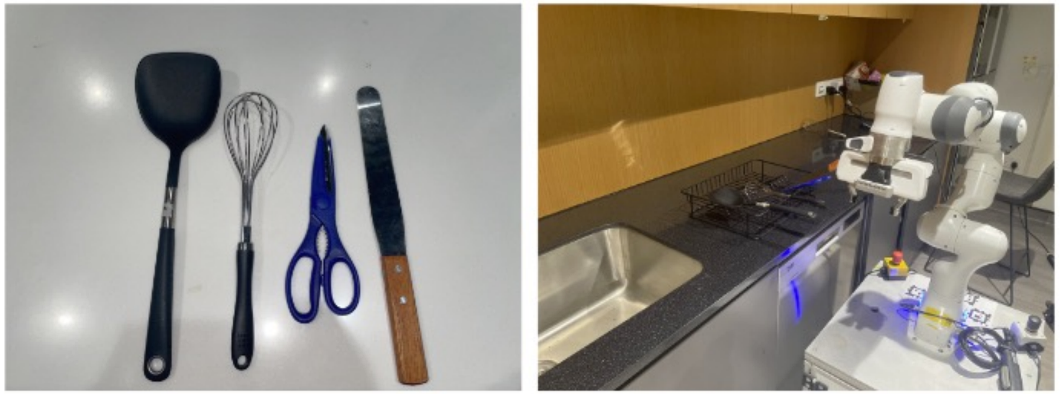


Fig. 5: (Left) Utensils used for the experiment: spatula, egg whisk, scissors and knife. (Right) The experiment environment that emulates a kitchen.

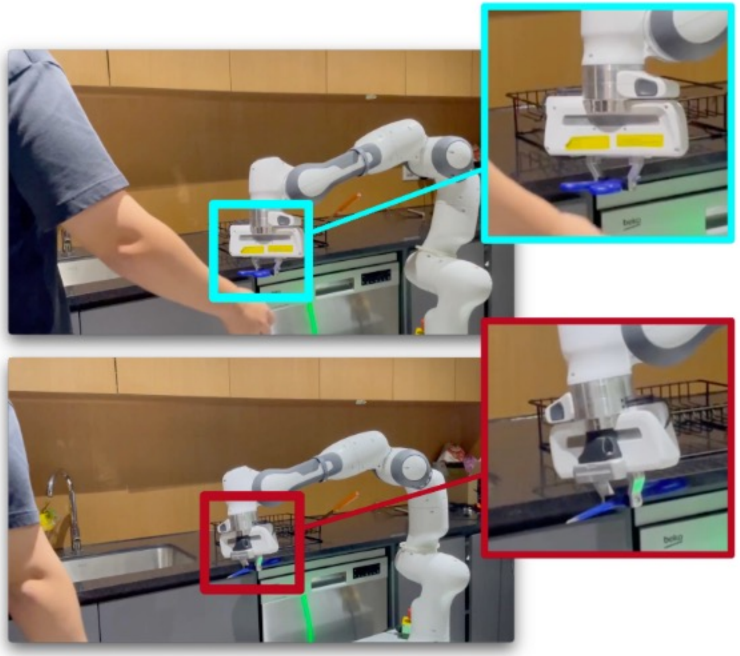
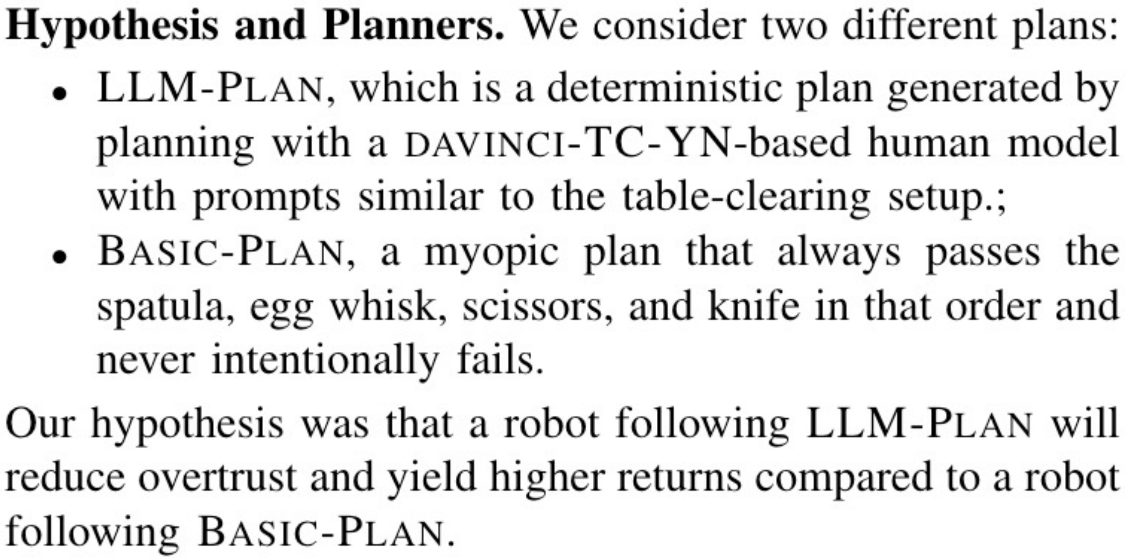


Fig. 6: Success condition (top) and intentional failure (bottom) while passing the scissors.

Planning with LLM-Based Human models

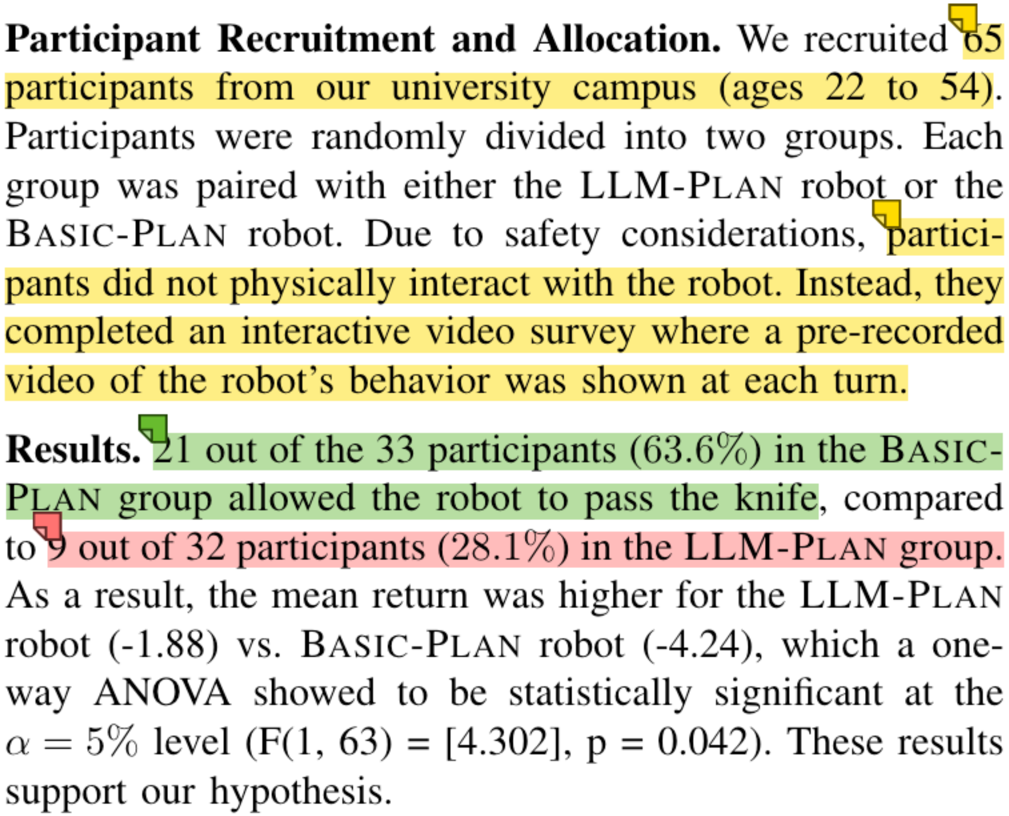
Experiment 2: Utensil-passing Experiment



Hypothesis and Planners. We consider two different plans:

- LLM-PLAN, which is a deterministic plan generated by planning with a DAVINCI-TC-YN-based human model with prompts similar to the table-clearing setup.;
- BASIC-PLAN, a myopic plan that always passes the spatula, egg whisk, scissors, and knife in that order and never intentionally fails.

Our hypothesis was that a robot following LLM-PLAN will reduce overtrust and yield higher returns compared to a robot following BASIC-PLAN.



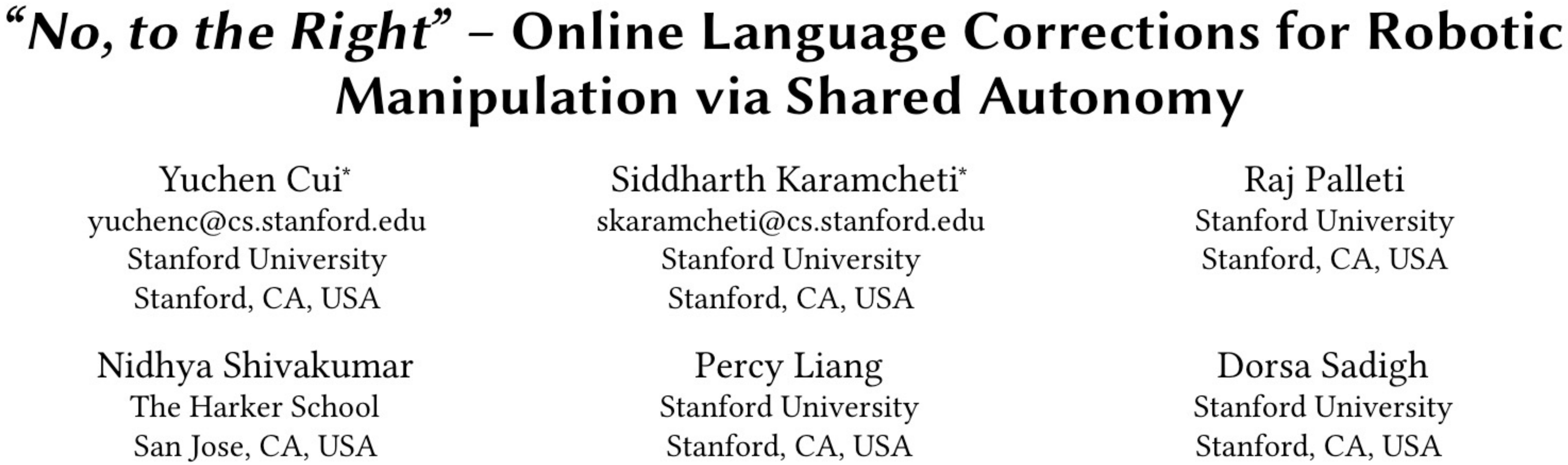
Participant Recruitment and Allocation. We recruited 65 participants from our university campus (ages 22 to 54). Participants were randomly divided into two groups. Each group was paired with either the LLM-PLAN robot or the BASIC-PLAN robot. Due to safety considerations, participants did not physically interact with the robot. Instead, they completed an interactive video survey where a pre-recorded video of the robot's behavior was shown at each turn.

Results. 21 out of the 33 participants (63.6%) in the BASIC-PLAN group allowed the robot to pass the knife, compared to 9 out of 32 participants (28.1%) in the LLM-PLAN group. As a result, the mean return was higher for the LLM-PLAN robot (-1.88) vs. BASIC-PLAN robot (-4.24), which a one-way ANOVA showed to be statistically significant at the $\alpha = 5\%$ level (F(1, 63) = [4.302], p = 0.042). These results support our hypothesis.

Summary and Discussion

- “a first study” into LLM-based zero-shot human models in HRI.
- key finding is that LLMs can be effective task-level human-models — they can model high-level human states and behavior.
- demonstrated that incorporating a LLM-based human model can yield reasonable plans in both trust-based HRI scenarios.

Discussion:
trust-based HRI scenarios are good testbeds for our influence human work?



"*No, to the Right*" – Online Language Corrections for Robotic Manipulation via Shared Autonomy

Yuchen Cui*
yuchenc@cs.stanford.edu
Stanford University
Stanford, CA, USA

Siddharth Karamcheti*
skaramcheti@cs.stanford.edu
Stanford University
Stanford, CA, USA

Raj Palleti
Stanford University
Stanford, CA, USA

Nidhya Shivakumar
The Harker School
San Jose, CA, USA

Percy Liang
Stanford University
Stanford, CA, USA

Dorsa Sadigh
Stanford University
Stanford, CA, USA

[2301.02555] "No, to the Right" -- Online Language Corrections for Robotic Manipulation via Shared Autonomy (arxiv.org)

Background

Research in natural language for robotics has focused on ***dyadic, turn-based*** interactions between humans and robots.

In this paradigm a human gives an instruction, then the robot executes autonomously – simultaneously resolving the human's goal as well as planning a course of actions to execute in the environment, without any additional user input.

This explicit division of agency between humans and robots places a tremendous burden on learning

- *inefficiency*
  - *existing systems either require large amounts of language-aligned demonstration data to learn policies*
  - *make other restrictive assumptions about known environment dynamics*
- *lack of adaptivity*

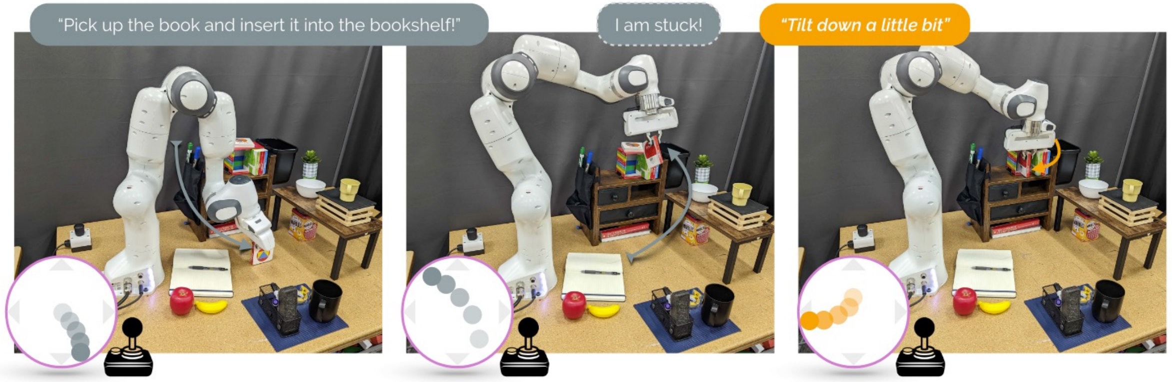
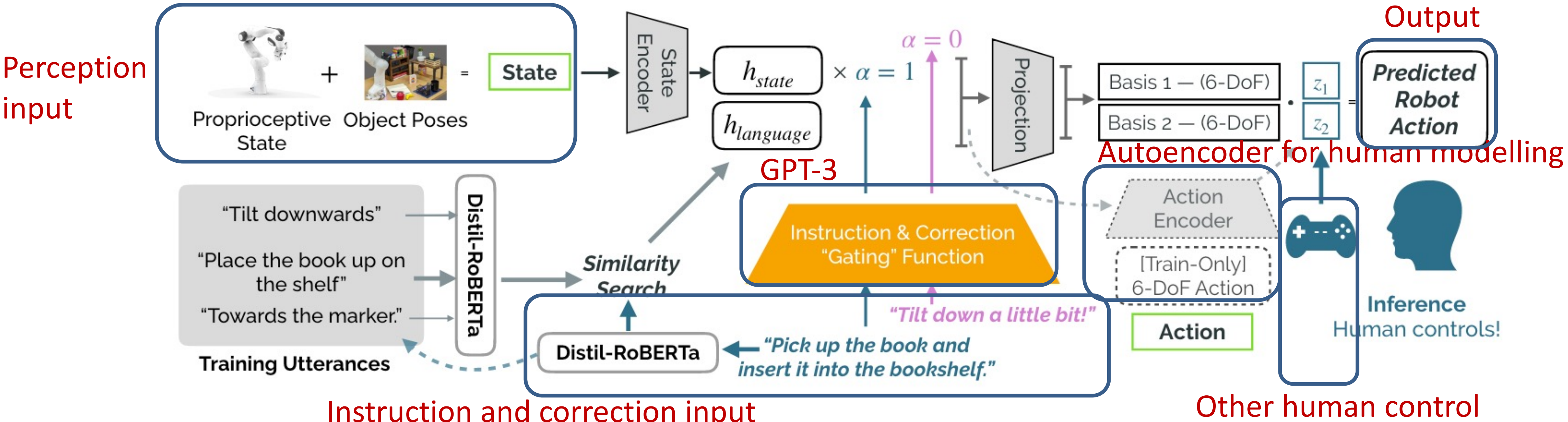


Figure 2: A user interacts with our system. [Left] The user utters "pick up the book and insert it into the bookshelf," inducing a low-dimensional controller (depicted with the joystick and shaded inputs). [Middle] This control space is state and language-conditioned: pressing down brings the end-effector close to the book, while holding up/left after grasping the book moves the end-effector towards the shelf. However, this *static* controller is not enough; the user gets stuck! [Right] Our approach allows users to provide real-time corrections ("tilt down a little bit") refining the control space so the user can complete the task.

MANCHESTER 1824
The University of Manchester

LILAC

LILAC: Language-Informed Latent Actions with Corrections – that presents a generalizable framework for adapting to ***online natural language corrections*** built within a ***shared autonomy paradigm*** for human-robot collaboration.



Perception input

Output

GPT-3

Autoencoder for human modelling

Instruction and correction input

Other human control

Figure 3: LILAC Overview – solid lines represent the inference pipeline, while dashed lines indicate training-only steps. Part of LILAC's ability to incorporate language corrections efficiently is the "gating" module (orange) which controls the amount of state-context for a given input – for example, grounding a correction such as "tilt down a little bit" requires no state context ($\alpha = 0$), whereas a high-level instruction such as "pick up the book and insert it into the bookshelf" does require context ($\alpha = 1$). We use GPT-3, a pretrained language model, to provide α (see §4.3 for discussion).

Problem Statement

$$(\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{U}, \mathcal{C}^*, \mathcal{Z})$$

u ∈ U denotes a high-level natural language instruction provided by the user

c ∈ C$*$ denotes the ordered (possibly empty) stack of natural language corrections the user has provided

a ∈ A ⊆ Rk denotes a robot's k-dimensional action
z ∈ Z ⊆ Rd where $d \ll k$ denotes a user-provided input via their low-dimensional control device (e.g., a 2-DoF joystick)

The goal of LILAC is to learn a function Fθ $(st$, zt , ut , ct) : S × Z × U × C$*$ → A

The corresponding low-DoF control manifold Ð zt ∈Z Fθ $(st$, zt , ut , ct) provides an intuitive interface for the user to maneuver the robot towards satisfying the task in question.

MANCHESTER 1824
The University of Manchester

LILAC

Address if the correction is completed

At each new timestep t + 1, a user can either provide a new language correction c' which is *"pushed" onto the stack, press a button to "pop" their latest correction off of the stack c_t signalling that their correction has been addressed*, or *provide a control input z_t that is mapped to the corresponding robot action a_t.*

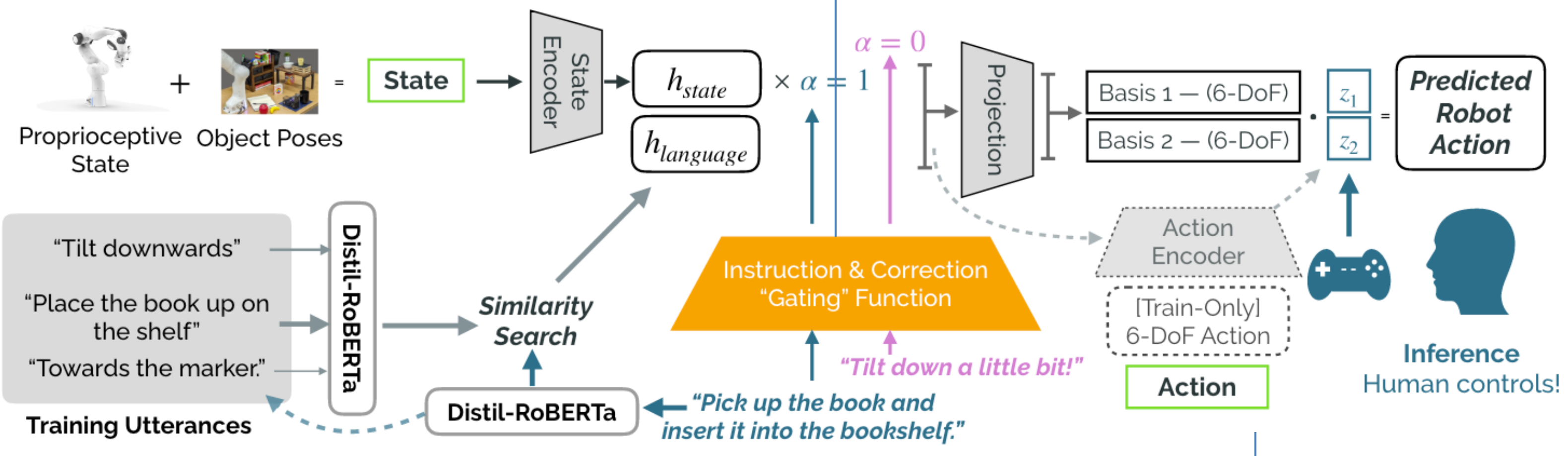
MANCHESTER 1824
The University of Manchester

LILAC

Address if the correction is completed

At each new timestep t + 1, a user can either provide a new language correction c ′ which is *“pushed” onto the stack, press a button to “pop” their latest correction off of the stack* ct *signalling that their correction has been addressed*, or *provide a control input* zt *that is mapped to the corresponding robot action* at .

Using GPT-3 to Identify Corrections



tions a. Crisply, we define $\hat{a} = \mathcal{F}_\theta(s, z, u, \mathbf{c})$ as:

$$h_{\text{state}} \in \mathbb{R}^m = \texttt{EncodeState}_\theta(s)$$
$$h_{\text{language}} \in \mathbb{R}^m = \texttt{EncodeLanguage}_\theta(u, \mathbf{c})$$
$$\alpha \in [0, 1] = \texttt{GPTGating}(u, \mathbf{c})$$
$$h_{\text{gated}} \in \mathbb{R}^m = \alpha \cdot h_{\text{state}} + (1 - \alpha) \cdot \text{bias}_\theta$$
$$h_{\text{fused}} \in \mathbb{R}^m = \texttt{FiLM}_\theta(h_{\text{gated}}, h_{\text{language}})$$
$$B_{\text{bases}} \in \mathbb{R}^{k \times d} = \text{Gram-Schmidt}(\texttt{Projection}_\theta(h_{\text{fused}}))$$
$$\hat{a} \in \mathbb{R}^k = B_{\text{bases}} \cdot z$$

Learning from Language & Demonstrations. To learn Fθ , we assume a dataset of (u = language, τ = trajectory) pairs,

$$z_{\text{compressed}} \in \mathbb{R}^d = \texttt{Compress}_\theta(a)$$
$$a_{\text{reconstruct}} \in \mathbb{R}^k = B_{\text{bases}} \cdot z_{\text{compressed}}$$
$$\mathcal{L}(\theta) = ||a - a_{\text{reconstruct}}||_2^2$$

we do not have access to “ground-truth” latent actions z for each given robot action a

learning a state-and-language conditional autoencode

```
PROMPT = (
    "I'm building a robot that can follow language commands. Tell me (YES or NO) if the
robot can execute "
    "the following language instructions without knowing any other information about its
environment.\n\n"
    "Input: move to the right\n"
    "Output: YES\n\n"
    "Input: rapidly twist to the front\n"
    "Output: YES\n\n"
    "Input: clean up the spilled coffee\n"
    "Output: NO\n\n"
    "Input: left\n"
    "Output: YES\n\n"
    "Input: move towards the bookshelf\n"
    "Output: NO\n"
    "Input: %s\n"
    "Output:"
)
```

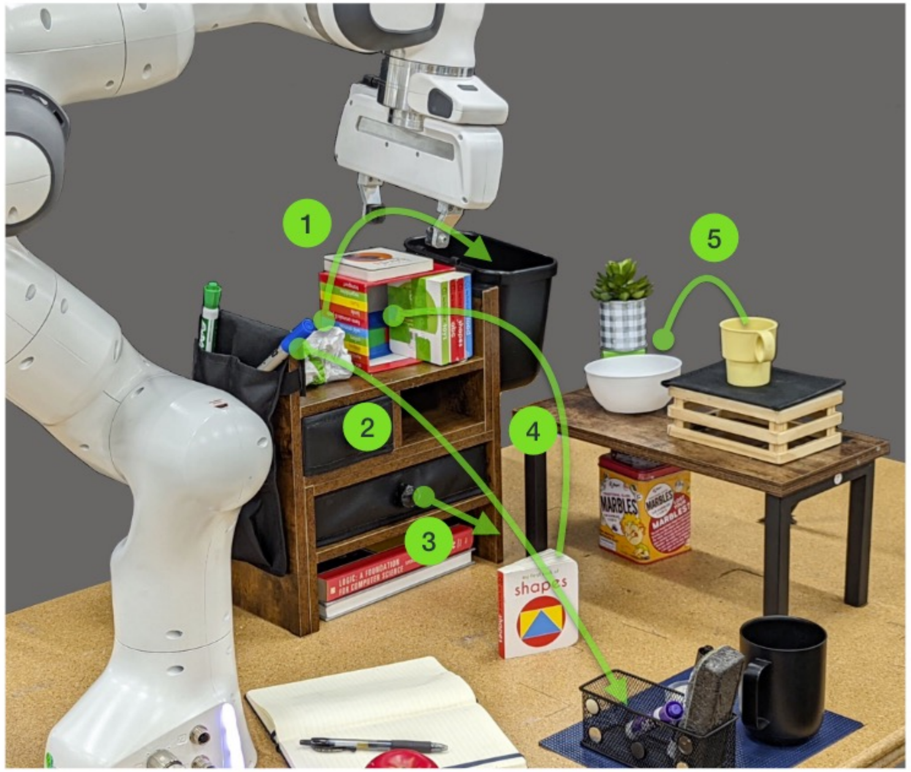


Figure 5: Setup of our tabletop manipulation environment with sketches of our high-level tasks (further details in §5).

Environment & Tasks. We consider a multi-task "desk" environment (Figure 5) with the following tasks listed by complexity:

(1) **`clean-trash`**: throw away a piece of crumpled paper (deformable) into the black trash bin.

(2) **`transfer-pen`**: transfer the blue marker (upper left of Figure 5) from the shelf into the metal tin holder (lower left).

(3) **`open-drawer`**: Open the bottom drawer on the shelf by grasping the small knob, and sliding out horizontally (requires fine-grained end-effector orientation control).

(4) **`insert-book`**: Pick up the book on the table by its spine, and insert it into the bookshelf (has only a few millimeters of clearance on either side).

(5) **`water-plant`**: Water the succulent (white bowl on the upper right of Figure 5) using the water in the yellow cup (rather than actual water, we use marbles for easy cleanup).

LILAC

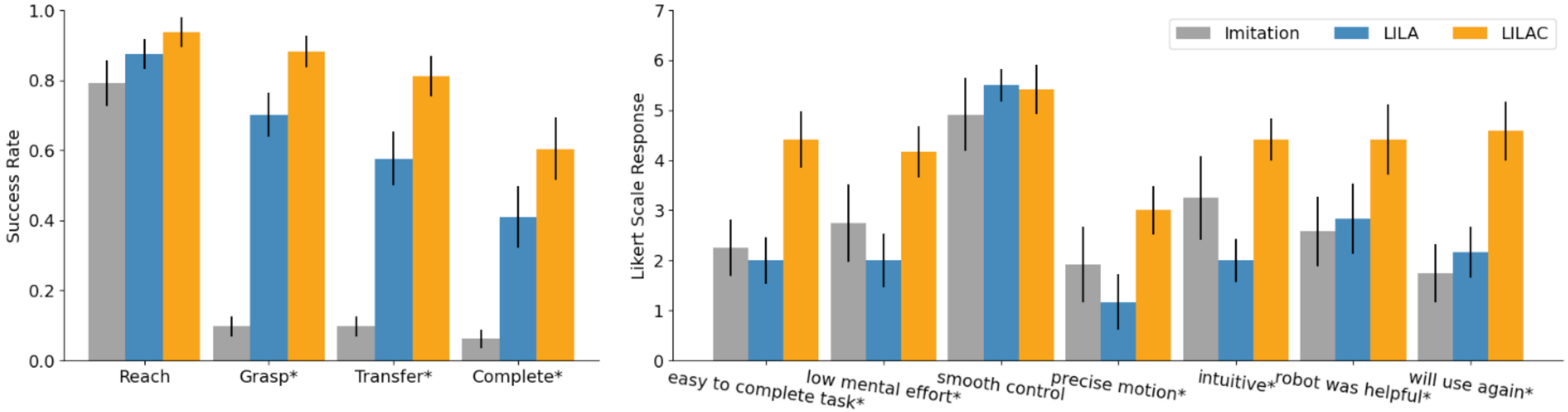


Figure 4: Results from our user study ($n = 12$) across three conditions: 1) Language-Conditioned Imitation Learning, 2) Language-Informed Latent Actions (LILA) – an instantiation of language-informed shared autonomy *without* online corrections, and 3) LILAC – our approach where users can provide online corrections at any point during robot execution.

LILAC

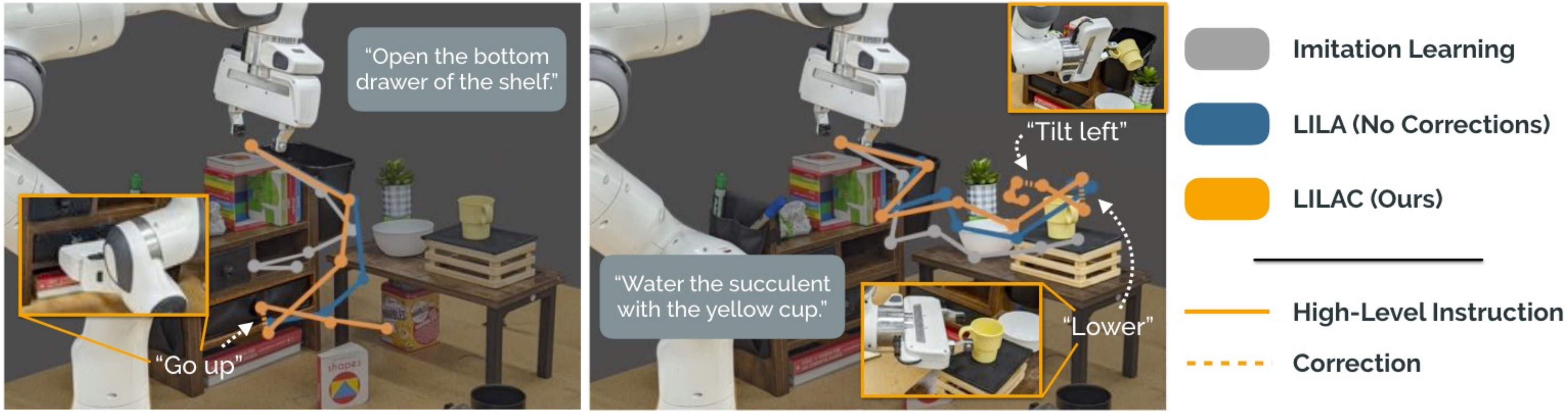


Figure 6: Qualitative trajectories across the different control strategies for the `open-drawer` and `water-plant` tasks. The fully autonomous imitation learning approach fails to make it beyond the first stage of the task, while LILA is able to reach the drawer as well as the cup but fails to precisely aim and grasp the object. LILAC gets stuck at the same place, but is able to recover as the user issues low-level corrections to precisely maneuver the end-effector and fully complete the tasks.

LILAC

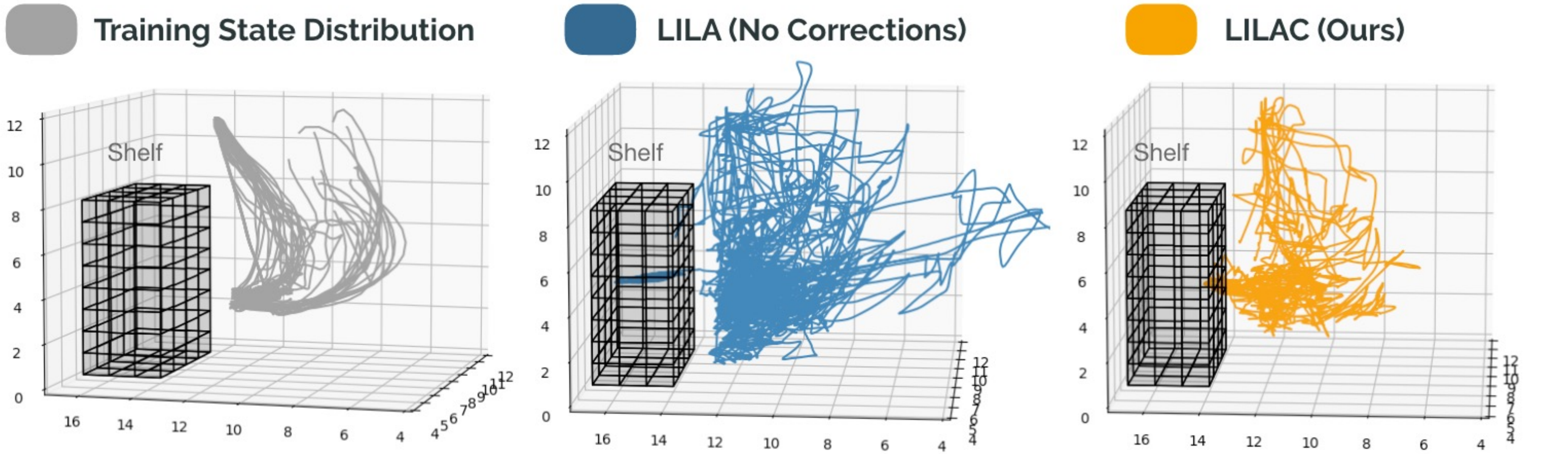


Figure 7: Observed trajectories for LILA and LILAC on the `open-drawer` task (with train trajectories shown on the left). While LILA deviates from the observed state distribution, states traversed with LILAC are close to those seen at training.

Summary and Discussion

LILAC is built within the shared autonomy paradigm whereby natural language utterances are mapped to meaningful, low-dimensional control spaces that humans can use to guide the robot, with each correction provided by the user working to refine the underlying control space, allowing for precise, targeted control.

Discussion