LaNMP: A Multifaceted Mobile Manipulation Benchmark for Robots

Ahmed Jaafar[†*], Shreyas Sundara Raman[†], Yichen Wei[†], Sofia Juliani[‡], Anneke Wernerfelt[§], Ifrah Idrees[†], Jason Xinyu Liu[†], Stefanie Tellex[†]
[†]Brown University [‡]Rutgers University [§]University of Pennsylvania
[*]ahmed_jaafar@brown.edu

***Abstract*—As robots that follow natural language become more capable and prevalent, we need a benchmark to holistically develop and evaluate their ability to solve long-horizon mobile manipulation tasks in large, diverse environments. To tackle this challenge, robots must use visual and language understanding, navigation, and manipulation capabilities. Existing datasets do not integrate all these aspects, restricting their efficacy as benchmarks. To address this gap, we present the *La*nguage, *N*avigation, *M*anipulation, *P*erception (*LaNMP*) dataset and demonstrate the benefits of integrating these four capabilities and various modalities. LaNMP comprises 574 trajectories across eight simulated and real-world environments for long-horizon room-to-room pick-and-place tasks specified by natural language. Trajectories consists of over 20 attributes, including RGB-D images, segmentations, and the poses of the robot body, end-effector, and grasped objects. We fine-tuned and tested two models in simulation to demonstrate the benchmark's efficacy in development and evaluation, as well as making models more sample efficient. The models performed suboptimally compared to humans across various metrics; however, showed promise in increasing model sample efficiency, indicating significant room for developing better multimodal mobile manipulation models using our benchmark.** [1]

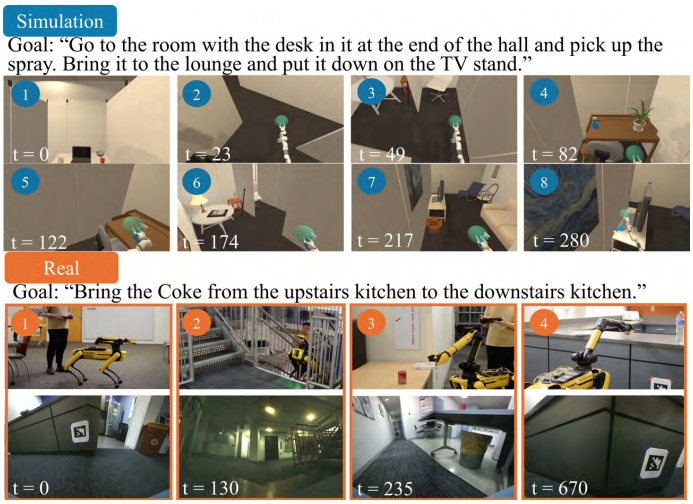


Fig. 1: Two natural language commands and their trajectories. The simulated is in blue, and the real is in orange. The top orange row shows the teleoperation of the Spot robot, and the bottom orange row depicts the robot's egocentric observation.

I. Introduction

Powered by large pretrained models, robots become more capable of understanding and executing natural language commands [17, 15, 27, 28, 41, 39]. However, language-conditioned mobile manipulation remains a major challenge. This is underscored by the best-performing system [31] of the NeurIPS 2023 Open Vocabulary Mobile Manipulation (OVMM) challenge [55] achieving a success rate of only 33%. One key reason is the lack of a comprehensive benchmark that aids the development and evaluation of a robotic system that can use multiple modalities to execute long-horizon tasks in diverse multiroom environments. For example, tasks like "Go to the kitchen and pour the boiling water into the teapot, then bring it to me in the living room" require the robot to use its language understanding, navigation, manipulation, and perception capabilities to satisfy. Specifically, the robot must ground the language command to the physical world, navigate between the kitchen and the living room, perceive (see) the boiling water, and manipulate the kettle and the teapot while ensuring the hot water does not spill.

Most existing datasets only contain a subset of language, navigation, manipulation, and perception data or are limited in ways, such as single-room environments, simulation only, and short-horizon language commands, as shown in Table I. This limits their ability to evaluate a robotic system's performance on long-horizon mobile manipulation tasks specified by complex language in multiroom environments with large numbers of objects. For example, RT-1 [6] introduces a dataset containing language, navigation, manipulation, and perception data across real and sim environments for kitchen tasks. However, it does not have multiroom tasks, limiting its task horizons. In a similar vein, Conq Hose [33] is a dataset of a quadruped moving a hose around following instructions. However, it does not include simulation data, limiting its diversity. Finally, QUARD [9] is a dataset that maps vision and language with navigation for quadrupeds, but it does not include manipulation, limiting its task-executing ability.

To address these problems, we present the **La**nguage, **N**avigation, **M**anipulation, **P**erception (**LaNMP**) dataset for language-conditioned mobile manipulation tasks. LaNMP contains 524 and 50 mobile manipulation tasks in five simulated and three real-world environments, respectively, that cover multiple rooms and floors. Each task is described by a natural language command and accompanied by a trajectory

[1]The dataset, models, and code are available at http://lanmpdataset.github.io.

collected from a human participant. Every trajectory consists of perception, navigation, and manipulation data of over 20 attributes, including RGB-D images, segmentation masks, and the poses of the robot body, end-effector, and grasped objects. To the best of our knowledge, LaNMP is the first dataset that contains long-horizon room-to-room mobile manipulation tasks integrating natural language, navigation, manipulation, and perception (LaNMP) data in both simulation and the real world, utilizing a quadruped mobile manipulator, Spot [4]. Example trajectories are illustrated in Figure 1.

To evaluate the applicability and strength of LaNMP as a training and evaluation benchmark, we fine-tuned and tested two recent models on various metrics. The models perform poorly in contrast to humans, scoring 0% on Success Rate metrics, indicating that current mobile manipulation models are not advanced enough to succeed on this difficult benchmark. Therefore, further model development is necessary to perform well on LaNMP and move one step closer to solving the mobile manipulation problem.

II. Related Work

Numerous datasets incorporate at least one of the four aspects, i.e., language, navigation, manipulation, and perception (LaNMP). Many focus solely on perception or natural language, which are not designed for embodied tasks. Some are for embodied tasks but only contain two aspects, limiting their capability. Some prominent ones include RoboTurk [29], SayTap [50], and TidyBot [54]. Thus, our discussion will focus on those encompassing at least three aspects, as these are most closely related to our work. While there may be several differences between the datasets discussed in this section and LaNMP, we primarily focus on the most significant variance, which pertains to the aspects each dataset lacks. A subset of the datasets is shown in Table I, and the full table, Table IV, is in Appendix A. Figure 2 shows how LaNMP is scored differently from other datasets.

A. Datasets of Language, Manipulation, and Perception

Many robot datasets encompass natural language, manipulation, and perception [43, 30, 22, 49, 3, 19, 18, 57, 20, 21, 45, 44, 52]. LaNMP is distinguished by incorporating navigation and these modalities within a closed-loop system. This enhancement extends the robot's general-purpose capabilities to mobile tasks, surpassing the limitations of stationary tasks like those performed on tabletop.

B. Datasets of Language, Navigation and Perception

A considerable body of work encompasses natural language, perception, and navigation but not manipulation. Room-to-Room [2], Room-Across-Room [24], ALFRED [46], CoNav [25], and TEACh [37] introduce datasets that map natural language instructions and visual data to navigation actions in household environments across multiple simulated platforms. Finally, QUARD [9] is a dataset that enhances quadruped robots' intelligence by integrating visual and natural language instructions into executable actions for tasks like navigation, terrain traversal, and manipulation. However, QUARD's manipulation refers to whole-body manipulation rather than object manipulation utilizing an arm. Our quadruped robot has an arm, meaning it collects object manipulation data. LaNMP's advantage over these datasets is that it includes manipulation data, enabling tasks that involve interacting with and manipulating objects on the go rather than merely navigating through environments.

C. Datasets of Navigation, Manipulation and Perception

There are significantly fewer papers with navigation, manipulation, and perception (NPM) but no natural language. Wong et al. [53] introduce the MoMaRT system, which allows for intuitive control of a mobile manipulator's arm and base through teleoperation. It focuses on collecting a multi-user demonstration dataset in simulated environments using MoMaRT, capturing long-horizon mobile manipulation tasks to support novel imitation learning with error detection methods. Mobile ALOHA [12] is a cost-effective physical system designed for imitating bimanual, whole-body mobile manipulation tasks, which is used as a teleoperation system for data collection. It was used to collect mobile manipulation NPM data, which then co-trained existing imitation learning models. BRMData [56] is a bimanual-mobile robot manipulation dataset for household tasks, featuring diverse manipulation scenarios and sensory inputs to advance robot learning and imitation from human demonstrations. Unlike these datasets, LaNMP centers its tasks around natural language, using it as the core modality upon which all other modalities are built. Incorporating natural language enhances user accessibility, facilitates intuitive human-robot interaction, and enables robots to execute a wider range of tasks based on semantic instructions.

D. Datasets of Language, Navigation, Manipulation and Perception

Few papers comprehensively cover natural language, perception, navigation, and manipulation, and they often have limitations in other areas. RT-1 (Robot Transformer) [6] is an approach utilizing a transformer that inputs visual and textual data and outputs both navigation and manipulation actions to complete mobile manipulation tasks. Simultaneously, they release a dataset collected using two robots from both simulation and the real world used to train the method. While this dataset encompasses LaNMP's modalities, LaNMP is different in the following ways: **1)** RT-1 is limited in scope, focusing solely on fetch and deliver tasks within single kitchen scenes. In contrast, LaNMP supports more complex, longer-horizon mobile manipulation tasks that span multiple rooms and floors in a diverse set of environments. **2)** RT-1 encompasses fewer data types than LaNMP. For instance, RT-1's perception data is limited to RGB, whereas LaNMP includes RGB, depth, and instance segmentations. **3)** The embodiments used in LaNMP, both in simulation and the real world, differ from those in RT-1. To collect data on physical robots, we used a quadruped mobile manipulator, while RT-1 used a wheeled mobile manipulation, so LaNMP can be used to evaluate the

	Real	Sim	Natural Language	Navigation	Manipulation	Perception	Multiroom Navigation
RT-1	✓	✓	✓	✓	✓	✓	✗
Conq Hose	✓	✗	✓	✓	✓	✓	✓
RoboCasa	✗	✓	✓	✓	✓	✓	✓
ALFRED	✗	✓	✓	✓	✗	✓	✗
QUARD	✓	✓	✓	✓	✗	✓	✗
Mobile ALOHA	✓	✗	✗	✓	✓	✓	✗
VIMA	✗	✓	✓	✗	✓	✓	✗
DROID	✓	✗	✓	✗	✓	✓	✗
LaNMP (Ours)	✓	✓	✓	✓	✓	✓	✓

TABLE I: Dataset comparison of the different aspects and modalities. Full table is in Appendix A, Table IV.

locomotion of both wheeled and quadruped robots. In addition, using a quadruped allows our data to expand to tasks in difficult-to-navigate areas, such as stairs in a house, a feat that RT-1's wheeled robots cannot accomplish. Conq Hose [33] is a mobile manipulation dataset utilizing a quadruped robot, Spot, to grab, lift, and drag a vacuum hose around in a real-world environment. The dataset is limited in size, with only 139 trajectories, capabilities, ability to perform only one task, and data types. LaNMP contains Spot body velocity, arm velocity, joint states, etc., while Conq Hose does not. It is also restricted to the real world, lacking simulation data. RoboCasa [34] is a simulation software that can be used to create datasets for training robots in everyday environments, leveraging LLMs to enhance diversity and realism. The created dataset does not contain real robot data, unlike LaNMP. Finally, Open X-Embodiment (OXE) [36] is a consolidated dataset combining many existing datasets, utilizing a multitude of real robots and a few simulated ones, aimed at exploring the potential for training generalist robotic policies that can be efficiently adapted to new robots, tasks, and environments. The authors also showcase RT-X models demonstrating the benefits of leveraging combined experiences across diverse robotic platforms. Relevant sub-datasets, such as RT-1, have already been discussed in this section. OXE is vastly composed of manipulation-only tabletop data, lacking a substantial amount of navigation data, meaning limited mobile manipulation. As a result, the vast majority of tasks are short-horizon, even in the few mobile manipulation sub-datasets. This also means that OXE can potentially suffer from the class imbalance problem. In addition, OXE includes a wide range of datasets, each with its own unique features but organized under a common structure. As a result, some data trajectories in OXE may lack certain attributes, leading to gaps and inconsistencies in the dataset. On the other hand, LaNMP ensures completeness in this aspect. It is important to understand that directly comparing OXE and LaNMP is not a straightforward evaluation because LaNMP is a single dataset, while OXE is a combination of multiple datasets. This difference suggests that LaNMP could be incorporated into the OXE framework, which would open up exciting possibilities for integration and advancement.

III. LANGUAGE, NAVIGATION, MANIPULATION AND PERCEPTION (LANMP) DATASET

LaNMP is a multimodal dataset that contains long-horizon mobile manipulation tasks specified by natural language in diverse multiroom simulated and real-world environments. Having both simulated and real data strengthens the diversity of the dataset and, as a result, the generalizability of models being trained on it. LaNMP encapsulates a broad spectrum of tasks typically performed at the home or workplace. Completing the tasks requires the robot to use its language understanding, navigation, manipulation, and perception capabilities. Throughout task execution, comprehensive trajectory data with over 20 attributes, e.g., RGB-D images, segmentation masks, and the poses of the robot body, end-effector, and grasped objects, is captured in a constant frequency of 3 Hz.

The relative scarcity of long-horizon data in existing datasets poses a significant challenge for developing versatile robotic systems capable of navigating and interacting with complex environments over extended periods of time. Our benchmark dataset uniquely enriches the landscape of long-horizon multi-modal multiroom data.

A. Simulation Dataset

Our simulation dataset comprises 524 trajectories over 20 rooms in five environments. We selected environments that ensured diversity in objects and room layouts, thus enhancing the generalizability of models trained on our dataset. We use the AI2THOR simulator [23]. Specifically, we use RoboTHOR [8] environments because they have multiple rooms, while the iTHOR [23] environments used by existing datasets, such as ALFRED, mainly have single rooms. In each environment, there is an average of 105 trajectories, as illustrated in Figure 3d. The average length of these trajectories is 172 steps, as detailed in Figure 3a.

We used the ManipulaTHOR [11] robot in the RoboTHOR environments since it has an arm with low-level pose data. In contrast, the iTHOR agent employed in existing datasets, such as ALFRED, only provides high-level skills, like "pick up" and "open", thus it cannot be used to train models with manipulation capabilities. We collected 13 attributes, such as RGB-D images, segmentations, and the poses of the robot body, end-effector, and grasped objects. Appendix A, specifically Figure 4, provides more details on the collected data.

We used Prolific[2] to collect natural language commands from 41 participants. To facilitate that, we developed a website that contains the task description, example commands, and the interactive RoboTHOR environments. Appendix B1 contains more

[2] https://www.prolific.com: a crowdsourcing platform that connects researchers with a wide pool of participants.

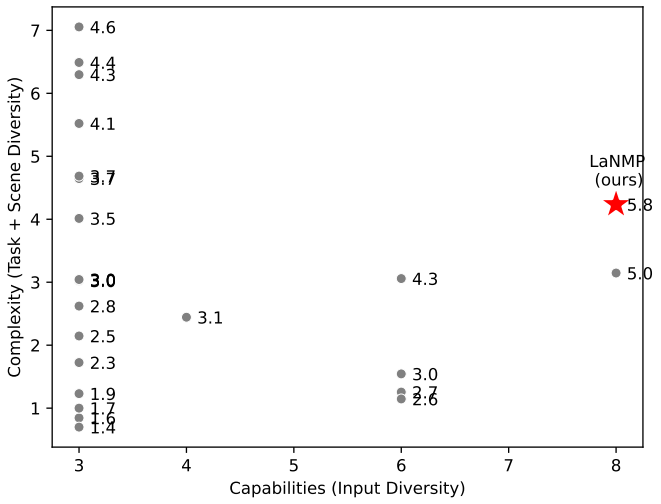


Fig. 2: LaNMP is differentiated from other datasets with diverse input features and environments. Quantitative scores show the geometric mean of the two axes, where LaNMP shows both high complexity and input diversity. Most datasets have low input diversity but a range of complexities. Scoring system details are in Appendix A1.

details about the website. Each participant first moved in the simulated environment and interacted with various objects, then provided 15 commands, three for each of the five environments. We collected a total of 615 natural language commands from the 41 participants. We then conducted a meticulous filtering process to select 524 high-quality commands that instruct the robot to perform room-to-room pick-and-place tasks by using all its navigation, manipulation, and perception capabilities.

Next, we recruited a different group of 15 participants to execute the commands in the simulator. We collected trajectories comprising navigation, manipulation, and perception data required to fulfill the commands. This teleoperation approach allowed us to gather precise ground-truth robot and environment data crucial for developing capable mobile manipulation systems. Appendices B2 and B3 contain more details about recruiting and simulation data collection, respectively.

B. Real-World Dataset

Our real-world dataset comprises 50 trajectories across 10 rooms in three environments. The first is a three-room laboratory, the second is a floor in a university building, and the last spans two adjacent floors connected by stairs in the same building. We picked these environments for their large size, multitude of rooms, inclusion of stairs, and object diversity. The two floors contain kitchens, furnished lobbies, a classroom, and a staircase. We recruited seven participants that provided 50 natural language commands. Specifically, 20, 15, and 15 for each of the three environments, respectively. Each command specifies long-horizon room-to-room pick-and-place tasks.

The collected commands were executed on a quadruped mobile manipulator, Spot. To collect the trajectory data, we first built dense 3D topological maps (shown in Figure 7) of the environments and then teleoperated the robot to follow the collected commands. Spot has more sensors than the virtual agent, allowing us to collect more diverse data types. The main data types include RGB-D, body and end-effector poses, body and arm velocities, joint states, and velocities. Appendix A, specifically Figure 5, lists all data types. The average trajectory length is 323, as detailed in Figure 3b.

IV. Evaluation

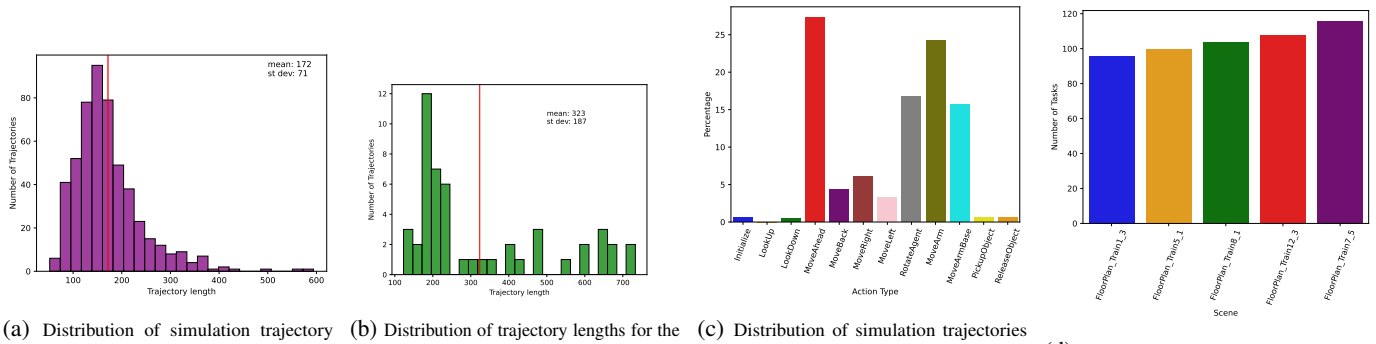
The LaNMP dataset can be used to benchmark different robotic paradigms such as imitation learning (IL) [38, 35, 1], reinforcement learning (RL) [48], skill learning, and providing in-context examples for planning. Since there has been increased interest and widespread adoption in IL approaches, such as behavior cloning (BC) [38] in the RT [6, 5, 36] models, we evaluated our dataset by using it to fine-tune two recent BC models.

To evaluate the applicability and strength of the LaNMP benchmark, we employed it to fine-tune two recent models, namely RT-1 [6] and ALFRED's Seq2Seq model [46], utilizing the simulation data from LaNMP. Both models take natural language commands and RGB images as input, while ALFRED's Seq2Seq also takes in previous actions. Both output a mix of high and low-level navigation and manipulation actions for the simulated and real robots. This selection of models was instrumental in conducting a thorough evaluation of LaNMP's benchmarking efficacy across a wide spectrum of model dimensions and initial performance benchmarks. RT-1 is a relatively large ($\sim$ 45M parameters) and high-performing model, while ALFRED's Seq2Seq is smaller ($\sim$ 35M parameters) and exhibited poor performance on the ALFRED benchmark. In the coming sections, we explain the models and evaluation metrics.

A. Models

RT-1: Robotics Transformer 1 (RT-1) [6] is a model designed for generalizing across large-scale, multi-task datasets with real-time inference capabilities. RT-1 leverages a Transformer architecture [51] to process images and natural language instructions to generate discretized actions for mobile manipulation. RT-1 is trained on a diverse dataset of approximately 130K episodes across more than 700 tasks collected using 13 robots. This enables RT-1 to learn through BC from human demonstrations annotated with detailed instructions. Although RT-1-X [36] demonstrates superior performance, it was trained on OXE, which is mainly manipulation-only, while RT-1 was trained on mobile manipulation data. This makes RT-1 more suitable for our mobile manipulation fine-tuning.

To fine-tune RT-1, we utilized the natural language commands with RGB image observations from LaNMP. Due to the contrasting embodiment and incompatible action space of ManipulaTHOR, we modified the self-attention head to output 7-dimensional action-tokens that predict the body state (`body position`, `body yaw`, `body camera pitch`), the end-effector state (`end-effector position`, `grasp signal`), control modes (between body and end-effector), and



(a) Distribution of simulation trajectory lengths

(b) Distribution of trajectory lengths for the real data.

(c) Distribution of simulation trajectories types of actions.

(d) Number of simulation tasks per scene.

Fig. 3: Dataset Statistics

episode termination. We find that predicting the difference of body and end-effector states between timesteps, rather than predicting absolute coordinates, resulted in more stable learning. We adopted the action tokenization approach used by RT-1 to categorize continuous action outputs into 1 of 256 uniformly distributed bins between the minimum and maximum. More specific RT-1 modification details can be found in Appendix C1.

ALFRED's Seq2Seq: The ALFRED paper introduces a Sequence-to-Sequence [47] model leveraging a CNN-LSTM architecture with an attention mechanism for task execution. It encodes visual inputs via ResNet-18 [14] and processes language through a bidirectional LSTM. A decoder processes these multimodal inputs along with historical action data to iteratively predict subsequent actions and generate pixelwise interaction masks, enhancing precise object manipulation capabilities within the given environment.

We utilized a subset of LaNMP's data types, RGB, natural language, and previous actions to fit the ALFRED model's input specifications. We modified the ALFRED model outputs to an 8-dimensional action vector tailored to our action space, encompassing modes (`stop`, `base`, `grasp-release`, `head`, `rotate`, `arm-base`, and `ee`), base movements (`NoOp`, `MoveAhead`, `MoveBack`, `MoveRight`, and `MoveLeft`), grasping actions (`NoOp`, `PickupObject`, `ReleaseObject`), head movements (`NoOp`, `LookUp`, and `LookDown`), rotational angles (`-359-359` and `NoOp`), and end-effector movements (specified ranges for x, y, z coordinates, including `NoOp`). The numerical actions, which are the rotation and end-effector, are converted from global coordinates to relative coordinates by taking the differences between the timesteps for more stable learning. Based on the findings from RT-1, we discretized the continuous rotation and end-effector relative actions into 256 bins. This structured action representation enabled precise predictions and executions of robotic actions within the simulator at each timestep. Furthermore, we changed it so that only the goal command is used rather than inputting specific instructions to the model at every timestep since the ideal situation is for a human to command a robot only once. Detailed descriptions of the model modifications are provided in Appendix C2.

B. Experiment Details

For both models, we performed fine-tuning utilizing 5-fold cross-validation to evaluate scene generalizability potential. Trajectories in each scene are designated to a fold; fine-tuning is performed using four scene folds and evaluated on the fifth held-out scene fold. Additionally, we performed another experiment focusing on task generalization instead by utilizing subsets of all scenes for training and testing. All experiments utilized cross entropy (CE) loss between the predicted action distributions and the ground-truth actions. Given the scope of our research, hyperparameter tuning was deemed unnecessary, and consequently, a validation split was not incorporated. Thus, for the task generalization experiment, we used an 85% train, 15% test split for tasks in every scene. Fine-tuning was performed on a single 24 GB NVIDIA 3090 GPU.

C. Evaluation Metrics

LaNMP assumes humans are logical agents with commonsense reasoning by collecting teleoperated trajectories for complex long-horizon tasks.

For robust evaluation, we considered two categories of metrics for cross-scene and task generalization experiments: "ground truth relative" (GTR) metrics that compare against LaNMP trajectories as standards and "ground truth independent" (GTI) metrics that evaluate a trajectory (ground-truth or predicted) on task understanding or smoothness. These metrics provide a multifaceted assessment framework:

- *Task Success* (GTR): a binary value measuring whether an agent achieves the goal/completes the task specified in the command.
- *Distance From Goal* (GTR): the spatial distance between the agent's final position after executing a learned trajectory and the designated ground-truth goal state.

$$d = 1/2 \left(\sqrt{x^2_{\text{gt\_body},n} - x^2_{\text{eval\_body},n}} + \sqrt{x^2_{\text{gt\_ee},n} - x^2_{\text{eval\_ee},n}} \right)$$

- *Grasp Success Rate* (GTR): the efficacy of the agent's attempts to grasp objects. Specifically, the percentage of attempts that result in successful object acquisition.
- *Average RMSE* (GTR): the average root-mean-square error. It is a weighted average of body and end-effector pose errors between the predicted and ground-truth trajectories,

Model	SR	Length	Grasp SR	RMSE v.s. GT	Weighted Δ_{xyz}	CLIP EMA Score	Dist. from Goal	CE Loss
Cross-Scene								
— ALFRED's Seq2Seq	0.0	655.09 ± 450.52	0.0	3.11 ± 0.63	0.0026 ± 0.0035	0.1614 ± 0.0120	12.42 ± 5.44	286.77 ± 20.31
— RT-1	0.0	205.03 ± 27.36	0.0	9.50 ± 0.27	1.3423 ± 0.1133	0.1521 ± 0.0065	12.56 ± 6.67	80.98 ± 4.68
Task Generalization								
— ALFRED's Seq2Seq	0.0	501.60 ± 578.62	0.0	3.01 ± 1.18	0.0008 ± 0.0014	0.1681 ± 0.0327	12.83 ± 11.12	286.66 ± 398.80
— RT-1	0.0	199.56 ± 106.11	0.0	9.74 ± 1.67	1.3980 ± 0.5834	0.1488 ± 0.0243	12.40 ± 12.20	82.61 ± 1.81
Ground Truth	1.0	171.69 ± 70.80	1.0	—	0.5576 ± 0.1751	0.2067 ± 0.0311	—	—

TABLE II: Quantitative performance of ALFRED and RT-1 on LaNMP's simulation dataset

normalized by their maximum lengths.

$$RMSE = \sum_{i=0}^{n} 1/2 \left(\sqrt{x^2_{\text{gt\_body},i} - x^2_{\text{eval\_body},i}} + \sqrt{x^2_{\text{gt\_ee},i} - x^2_{\text{eval\_ee},i}} \right)$$

This metric should not be interpreted in isolation but rather in conjunction with other metrics to comprehensively assess the agent's performance. This is because, in rare cases, the predicted trajectories could theoretically be more efficient than the ground-truth.

- *Average Number of Steps* (GTR): the total number of actions an agent takes. It serves to evaluate a model's ability to replicate efficient human navigation.
- *Mean and Standard Deviation in State Differences* (GTI): the standard deviation in positional differences between successive timesteps in a trajectory. It assesses the control smoothness exhibited by the agent to compare learned trajectories against the fluidity and naturalness of the ground-truth trajectories.

$$\Delta = \sum_{i=1}^{n} 1/2 \left(\sqrt{x^2_{\text{eval\_body},i} - x^2_{\text{eval\_body},(i-1)}} + \sqrt{x^2_{\text{eval\_ee},i} - x^2_{\text{eval\_ee},(i-1)}} \right)$$

- *CLIP Embedding Reward* (GTI): the exponential moving average (EMA) of CLIP [40] text-image correlation scores for all steps of a trajectory. Natural language task specification can be ambiguous and difficult to formulate into a structured goal condition. Inspired by previous works using CLIP for RL rewards [7, 42], we propose this metric to capture complex semantic correlations between the trajectory and task specification. It attempts to capture the agent's understanding, reasoning, and grounding of a task using the CLIP embedding space. Basically, it tries to provide a measure of the agent's task comprehension and execution fidelity.

$$EMA_i = \alpha EMA_{i-1} + (1-\alpha) r_i$$

where

$$r_i := CLIP(task, img_i)$$

V. Analysis

The objective of evaluating the models is to determine the dataset's applicability and assess its difficulty as a benchmark. Our findings highlight that the human teleoperation benchmark established within the LaNMP framework presents a challenging baseline for policy learning endeavors. Specifically, the comparative analysis reveals a pronounced disparity in performance between the fine-tuned algorithmic models and the human-teleoperated trajectories. To illustrate, in the context of unseen scenes and tasks, both computational models—RT-1 and ALFRED's Seq2Seq—demonstrated a success rate of 0%, starkly contrasted against the ground truth trajectories. This significant discrepancy underscores the infancy of current State-of-the-Art (SOTA) models in mirroring the proficiency of human counterparts. Consequently, it accentuates the importance of cultivating more comprehensive datasets, such as LaNMP, encompassing a broader spectrum of abilities and sensory modalities. Such benchmarks are critical for advancing mobile manipulation capabilities, illustrating a pressing need to integrate diverse and sophisticated datasets to bridge the current performance gap between AI-driven systems and human operators. Experimental results are summarized in Tables II.

A. Model Performance Results

Specifically, the cross-scene and task generalization model performances on LaNMP can be delineated as follows:

- The lower CE Loss showed that RT-1 learns better than ALFRED's Seq2Seq, which is unsurprising since RT-1 is a larger and more advanced model. More loss details are in Appendix D.
- ALFRED's Seq2Seq's RMSE is lower than RT-1's. This may be attributed to ALFRED's Seq2Seq frequently outputting `NoOP` commands, resulting in it often remaining stationary. This means the predicted base and end-effector positions could be closer to the ground-truth than RT-1, due to RT-1 exploring more, thus it potentially deviated further from the GT path.
- ALFRED's Seq2Seq stayed a few steps within the start most of the time. Though RT-1 explored more, shown by the higher average movement per time step, it did not ever reach the goal, likely explaining why both models' distances from the goal were roughly the same.
- RT-1's trajectory lengths were significantly shorter because ALFRED's Seq2Seq often predicted many `NoOp` until it reached the maximum action limit of 1500 whereas RT-1 usually stopped earlier at $\sim 300 - 400$ steps. Both were still less efficient than the human.
- The weighted movement between step positions indicates the smoothness of the agents' control, with smaller values representing better smoothness. Although the models show smaller values than humans, this is primarily due to agents, especially ALFRED's Seq2Seq, remaining stationary for large parts of the predicted trajectories.

Model	1 Scene (100)	2 Scenes (50/50)	3 Scenes (33/33/34)	4 Scenes (25/25/25/25)
ALFRED's Seq2Seq	582.33 ± 518.80	479.81 ± 473.80	509.85 ± 475.47	$\mathbf{405.99} \pm 475.64$
RT-1	88.23 ± 1.92	93.50 ± 1.89	$\mathbf{78.83} \pm 1.88$	89.63 ± 1.84

TABLE III: Cross entropy loss for different scenes, showing scene diversity vs. dataset scale.

- Regarding the CLIP score, higher values represent better performance, with humans scoring higher than both models. This disparity indicates that the models are not near human-level understanding, reasoning, and grounding.
- Since RT-1 is a more sophisticated model, one would expect its CLIP score would be higher than ALFRED's Seq2Seq. However, ALFRED's Seq2Seq is marginally higher, which can be considered negligible. Further testing is required to conclusively determine why the models had similar scores.
- All CLIP scores, including those of the humans, were likely low due to the lack of semantic correlation between observations and commands throughout most of a trajectory. This is attributed to the robot primarily looking downward towards the ground or into the distance while navigating to its target.
- Comparing scene generalization with task generalization, the latter performed better marginally on most metrics. This may suggest that more scenes should be included in the training process, as only four were used during cross-validation compared to hundreds of tasks. However, the marginal gains observed do not provide a definitive conclusion.

B. Sample Efficiency & Dataset Diversity

It is evident that these models are not sample efficient, as they were not able to learn enough from LaNMP's ~ 400 trajectories to generalize well as indicated by the 0% success rate. BC models can suffer from sample inefficiency [16, 13], and Transformers are known to require large amounts of training data [26], suggesting that future models may have to use different paradigms and architectures to reach human-level sample efficiency.

We propose an alternate solution to increase sample efficiency by using diverse data. Although dataset scale is essential for policy learning and generalization in modern models due to their sample inefficiency, we hypothesize that diversity in scene and modality data is equally, if not more, critical for robotics models.

Many previous works such as ALFRED and Prompter [32] performed ablations across different modalities, which demonstrated the performance improvement from using multimodal (visual-language) inputs are greater than the combined performance improvements from using individual (unimodal) inputs. This speaks to the importance of diverse input features for high-fidelity policy learning.

To assess the importance of scene diversity, we evaluated RT-1 and ALFRED's Seq2Seq on 100 trajectories evenly sampled from an increasing number of scenes (1, 2, 3, 4) – such that the number of trajectories per scene reduced (100, 50, 33, 25), but the total number of trajectories remained unchanged as 100. During this fine-tuning, the models' weights were unfrozen to control for the effect of useful pre-trained representations on generalization. All four ablations are tested on a fifth held-out test scene. Results are summarized in Table III and can be delineated as follows:

- ALFRED's Seq2Seq's CE loss decreased significantly as the number of scenes increased, showing the importance of diversity over scale in this context.
- The larger decrease in CE loss on ALFRED's Seq2Seq compared to RT-1 could be attributed to the fact that it is a larger model, so having diversity from just a few scenes likely is not enough to override as many pre-trained representations as on a smaller model like ALFRED's Seq2Seq, thus making a smaller impact on the loss decrease.
- Larger models (RT-1 with ~ 45M parameters) generalized better with smaller training sets than smaller models (ALFRED's Seq2Seq with ~ 35M parameters). This is expected, as RT-1 is a larger model, and already performed better on the CE loss, as shown in Table II.
- After training RT-1 on just 100 trajectories, the cross entropy loss is within 2σ of the loss after fine-tuning on ~ 400 trajectories for the cross-scene experiment. This suggests that fine-tuning on a diverse but smaller dataset leads to similar policy generalization as fine-tuning on a dataset with $4\times$ the scale on an unseen test set.
- The standard deviation in RT-1's test set CE losses is statistically significant given that losses lie $> 1\sigma$ from one another. This suggests that varying the dataset diversity has a measurable influence on the CE loss of an unseen test set, i.e., the generalizability of the learned policy.

The LaNMP benchmark diversity results highlight the crucial role of diversifying the dataset. This underscores diversity's significance, which matches or even exceeds that of scale. It suggests that models can train on datasets that, despite being limited in volume, are abundant in scene diversity, thereby augmenting their sample efficiency. Thus a potential solution to improving model sample efficiency is to increase the number of modalities and diversity of the data, rather than having to change paradigms and model architectures. Our benchmark has been meticulously curated to foster and quantitatively assess this aspect of model efficiency.

VI. Conclusion

We introduce LaNMP, a mobile manipulation benchmark comprised of simulated and real-world trajectories paired with their respective language commands specifying household tasks. The trajectories are long-horizon, spanning multiple rooms and

floors, consisting of navigation, manipulation, and perception data. We fine-tuned and evaluated two models on LaNMP to test its applicability and strength as a benchmark. Though the models reduce training and test-set losses, suggesting good generalization, they exhibit poor performance on metrics. This suggests that LaNMP could serve as a difficult benchmark for advancing the development of mobile manipulation models.

Currently, the real-world dataset is not evaluated, making real robot evaluation on LaNMP a future task. Extending the benchmark to include tasks involving more complex manipulation than pick-and-place will increase its applicability to a broader set of models. Furthermore, models with more input types such as depth should be benchmarked to evaluate the full capability of the dataset. Finally, during evaluation, the AI2THOR simulator did not produce the intended movements at times, so higher-fidelity and more advanced simulators could be utilized in the future.

ACKNOWLEDGMENTS

This work is supported by ONR under grant award numbers N00014-22-1-2592 and N00014-23-1-2794, NSF under grant award number CNS-2150184, and with support from Amazon Robotics. We also thank Aryan Singh, George Chemmala, Ziyi Yang, David Paulius, Ivy He, Lakshita Dodeja, Mingxi Jia, Benned Hedegaard, Thao Nguyen, Selena Williams, Benedict Quartey, Tuluhan Akbulut, and George Konidaris for their help in various phases of work.

REFERENCES

[1] Pieter Abbeel and Andrew Y. Ng. Apprenticeship learning via inverse reinforcement learning. In *Proceedings of the Twenty-First International Conference on Machine Learning*, ICML '04, page 1, New York, NY, USA, 2004. Association for Computing Machinery. ISBN 1581138385. doi: 10.1145/1015330.1015430. URL https://doi.org/10.1145/1015330.1015430.

[2] Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian D. Reid, Stephen Gould, and Anton van den Hengel. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3674–3683, 2017. URL https://api.semanticscholar.org/CorpusID:4673790.

[3] Homanga Bharadhwaj, Jay Vakil, Mohit Sharma, Abhinav Gupta, Shubham Tulsiani, and Vikash Kumar. Roboagent: Generalization and efficiency in robot manipulation via semantic augmentations and action chunking, 2023.

[4] Boston Dynamics. Spot® - the agile mobile robot. https://www.bostondynamics.com/products/spot.

[5] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, Pete Florence, Chuyuan Fu, Montse Gonzalez Arenas, Keerthana Gopalakrishnan, Kehang Han, Karol Hausman, Alex Herzog, Jasmine Hsu, Brian Ichter, Alex Irpan, Nikhil Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Lisa Lee, Tsang-Wei Edward Lee, Sergey Levine, Yao Lu, Henryk Michalewski, Igor Mordatch, Karl Pertsch, Kanishka Rao, Krista Reymann, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Pierre Sermanet, Jaspiar Singh, Anikait Singh, Radu Soricut, Huong Tran, Vincent Vanhoucke, Quan Vuong, Ayzaan Wahid, Stefan Welker, Paul Wohlhart, Jialin Wu, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. RT-2: Vision-language-action models transfer web knowledge to robotic control. In *arXiv preprint arXiv:2307.15818*, 2023.

[6] Anthony Brohan et al. RT-1: Robotics transformer for real-world control at scale, 2023.

[7] Xuzhe Dang, Stefan Edelkamp, and Nicolas Ribault. Clip-motion: Learning reward functions for robotic actions using consecutive observations, 2023.

[8] Matt Deitke, Winson Han, Alvaro Herrasti, Aniruddha Kembhavi, Eric Kolve, Roozbeh Mottaghi, Jordi Salvador, Dustin Schwenk, Eli VanderBilt, Matthew Wallingford, Luca Weihs, Mark Yatskar, and Ali Farhadi. RoboTHOR: An Open Simulation-to-Real Embodied AI Platform. In *CVPR*, 2020.

[9] Pengxiang Ding, Han Zhao, Zhitao Wang, Zhenyu Wei, Shangke Lyu, and Donglin Wang. Quar-vla: Vision-language-action model for quadruped robots, 2023.

[10] Frederik Ebert, Yanlai Yang, Karl Schmeckpeper, Bernadette Bucher, Georgios Georgakis, Kostas Daniilidis, Chelsea Finn, and Sergey Levine. Bridge data: Boosting generalization of robotic skills with cross-domain datasets, 2021.

[11] Kiana Ehsani, Winson Han, Alvaro Herrasti, Eli VanderBilt, Luca Weihs, Eric Kolve, Aniruddha Kembhavi, and Roozbeh Mottaghi. ManipulaTHOR: A Framework for Visual Object Manipulation. In *CVPR*, 2021.

[12] Zipeng Fu, Tony Z. Zhao, and Chelsea Finn. Mobile aloha: Learning bimanual mobile manipulation with low-cost whole-body teleoperation. In *arXiv*, 2024.

[13] Abraham George and Amir Barati Farimani. One act play: Single demonstration behavior cloning with action chunking transformers. *ArXiv*, abs/2309.10175, 2023. URL https://api.semanticscholar.org/CorpusID:262054598.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. doi: 10.1109/CVPR.2016.90.

[15] Wenlong Huang, Fei Xia, Ted Xiao, Harris Chan, Jacky Liang, Pete Florence, Andy Zeng, Jonathan Tompson, Igor Mordatch, Yevgen Chebotar, et al. Inner monologue: Embodied reasoning through planning with language models. *arXiv:2207.05608*, 2022.

[16] Ahmed Hussein, Mohamed Medhat Gaber, Eyad Elyan, and Chrisina Jayne. Imitation learning: A survey of learning methods. *ACM Comput. Surv.*, 50(2), apr 2017. ISSN 0360-0300. doi: 10.1145/3054912. URL https:

//doi.org/10.1145/3054912.

[17] Brian Ichter et al. Do as i can, not as i say: Grounding language in robotic affordances. In *Conference on Robot Learning (CoRL)*, 2022.

[18] Yunfan Jiang, Agrim Gupta, Zichen Zhang, Guanzhi Wang, Yongqiang Dou, Yanjun Chen, Li Fei-Fei, Anima Anandkumar, Yuke Zhu, and Linxi Fan. Vima: General robot manipulation with multimodal prompts. In *Fortieth International Conference on Machine Learning*, 2023.

[19] Chuhao Jin, Wenhui Tan, Jiange Yang, Bei Liu, Ruihua Song, Limin Wang, and Jianlong Fu. Alphablock: Embodied finetuning for vision-language reasoning in robot manipulation, 2023.

[20] Alexander Khazatsky et al. Droid: A large-scale in-the-wild robot manipulation dataset. 2024.

[21] Heecheol Kim, Yoshiyuki Ohmura, and Yasuo Kuniyoshi. Multi-task robot data for dual-arm fine manipulation, 2024.

[22] Junghyun Kim, Gi-Cheon Kang, Jaein Kim, Suyeon Shin, and Byoung-Tak Zhang. Gvcci: Lifelong learning of visual grounding for language-guided robotic manipulation, 2023.

[23] Eric Kolve, Roozbeh Mottaghi, Winson Han, Eli VanderBilt, Luca Weihs, Alvaro Herrasti, Daniel Gordon, Yuke Zhu, Abhinav Gupta, and Ali Farhadi. AI2-THOR: An Interactive 3D Environment for Visual AI. *arXiv*, 2017.

[24] Alexander Ku, Peter Anderson, Roma Patel, Eugene Ie, and Jason Baldridge. Room-across-room: Multilingual vision-and-language navigation with dense spatiotemporal grounding. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4392–4412, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.356. URL https://aclanthology.org/2020.emnlp-main.356.

[25] Changhao Li, Xinyu Sun, Peihao Chen, Jugang Fan, Zixu Wang, Yanxia Liu, Jinhui Zhu, Chuang Gan, and Mingkui Tan. Conav: A benchmark for human-centered collaborative navigation, 2024.

[26] Tianyang Lin, Yuxin Wang, Xiangyang Liu, and Xipeng Qiu. A survey of transformers. *AI Open*, 3:111–132, 2021. URL https://api.semanticscholar.org/CorpusID:235368340.

[27] Jason Xinyu Liu, Ziyi Yang, Ifrah Idrees, Sam Liang, Benjamin Schornstein, Stefanie Tellex, and Ankit Shah. Grounding complex natural language commands for temporal tasks in unseen environments. In *Conference on Robot Learning (CoRL)*, 2023.

[28] Peiqi Liu, Yaswanth Orru, Chris Paxton, Nur Muhammad Mahi Shafiullah, and Lerrel Pinto. Ok-robot: What really matters in integrating open-knowledge models for robotics. *arXiv preprint arXiv:2401.12202*, 2024.

[29] Ajay Mandlekar, Yuke Zhu, Animesh Garg, Jonathan Booher, Max Spero, Albert Tung, Julian Gao, John Emmons, Anchit Gupta, Emre Orbay, Silvio Savarese, and Li Fei-Fei. Roboturk: A crowdsourcing platform for robotic skill learning through imitation, 2018.

[30] Oier Mees, Lukas Hermann, Erick Rosete-Beas, and Wolfram Burgard. Calvin: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks. *IEEE Robotics and Automation Letters (RA-L)*, 7(3):7327–7334, 2022.

[31] Andrew Melnik, Michael Büttner, Leon Harz, Lyon Brown, Gora Chand Nandi, Arjun PS, Gaurav Kumar Yadav, Rahul Kala, and Robert Haschke. Uniteam: Open vocabulary mobile manipulation challenge, 2023.

[32] So Yeon Min, Devendra Singh Chaplot, Pradeep Ravikumar, Yonatan Bisk, and Ruslan Salakhutdinov. Film: Following instructions in language with modular methods, 2021.

[33] Peter Mitrano and Dmitry Berenson. Conq hose manipulation dataset, v1.15.0. https://sites.google.com/view/conq-hose-manipulation-dataset, 2024.

[34] Soroush Nasiriany, Abhiram Maddukuri, Lance Zhang, Adeet Parikh, Aaron Lo, Abhishek Joshi, Ajay Mandlekar, and Yuke Zhu. Robocasa: Large-scale simulation of everyday tasks for generalist robots, 2024.

[35] Andrew Y. Ng and Stuart J. Russell. Algorithms for inverse reinforcement learning. In *Proceedings of the Seventeenth International Conference on Machine Learning*, ICML '00, page 663–670, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc. ISBN 1558607072.

[36] Abhishek Padalkar, Acorn Pooley, Ajinkya Jain, Alex Bewley, Alex Herzog, Alex Irpan, Alexander Khazatsky, Anant Rai, Anikait Singh, Anthony Brohan, et al. Open X-Embodiment: Robotic learning datasets and RT-X models. *arXiv preprint arXiv:2310.08864*, 2023.

[37] Aishwarya Padmakumar, Jesse Thomason, Ayush Shrivastava, Patrick Lange, Anjali Narayan-Chen, Spandana Gella, Robinson Piramuthu, Gokhan Tur, and Dilek Hakkani-Tur. TEACh: Task-driven Embodied Agents that Chat. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 2017–2025, 2022.

[38] Dean Pomerleau. Alvinn: An autonomous land vehicle in a neural network. In D.S. Touretzky, editor, *Proceedings of (NeurIPS) Neural Information Processing Systems*, pages 305 – 313. Morgan Kaufmann, December 1989.

[39] Benedict Quartey, Eric Rosen, Stefanie Tellex, and George Konidaris. Verifiably following complex robot instructions with foundation models, 2024.

[40] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. *CoRR*, abs/2103.00020, 2021. URL https://arxiv.org/abs/2103.00020.

[41] Shreyas Sundara Raman, Vanya Cohen, Eric Rosen, Ifrah Idrees, David Paulius, and Stefanie Tellex. Planning with large language models via corrective re-prompting.

In *IEEE International Conference on Robotics and Automation (ICRA)*, 2024.

[42] Juan Rocamonde, Victoriano Montesinos, Elvis Nava, Ethan Perez, and David Lindner. Vision-language models are zero-shot reward models for reinforcement learning, 2024.

[43] Rosario Scalise, Shen Li, Henny Admoni, Stephanie Rosenthal, and Siddhartha S Srinivasa. Natural language instructions for human–robot collaborative manipulation. *The International Journal of Robotics Research*, 37(6): 558–565, 2018.

[44] Nur Muhammad Mahi Shafiullah, Anant Rai, Haritheja Etukuru, Yiqian Liu, Ishan Misra, Soumith Chintala, and Lerrel Pinto. On bringing robots home, 2023.

[45] Snehesh Shrestha, Yantian Zha, Saketh Banagiri, Ge Gao, Yiannis Aloimonos, and Cornelia Fermuller. Natsgd: A dataset with speech, gestures, and demonstrations for robot learning in natural human-robot interaction, 2024.

[46] Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox. ALFRED: A Benchmark for Interpreting Grounded Instructions for Everyday Tasks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. URL https://arxiv.org/abs/1912.01734.

[47] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014. URL https://proceedings.neurips.cc/paper_files/paper/2014/file/a14ac55a4f27472c5d894ec1c3c743d2-Paper.pdf.

[48] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, second edition, 2018. URL http://incompleteideas.net/book/the-book-2nd.html.

[49] Chao Tang, Dehao Huang, Wenqi Ge, Weiyu Liu, and Hong Zhang. Graspgpt: Leveraging semantic knowledge from a large language model for task-oriented grasping. *IEEE Robotics and Automation Letters*, 8(11):7551–7558, 2023. doi: 10.1109/LRA.2023.3320012.

[50] Yujin Tang, Wenhao Yu, Jie Tan, Heiga Zen, Aleksandra Faust, and Tatsuya Harada. Saytap: Language to quadrupedal locomotion. *arXiv preprint arXiv:2306.07580*, 2023.

[51] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[52] Homer Walke, Kevin Black, Abraham Lee, Moo Jin Kim, Max Du, Chongyi Zheng, Tony Zhao, Philippe Hansen-Estruch, Quan Vuong, Andre He, Vivek Myers, Kuan Fang, Chelsea Finn, and Sergey Levine. Bridgedata v2: A dataset for robot learning at scale. In *Conference on Robot Learning (CoRL)*, 2023.

[53] Josiah Wong, Albert Tung, Andrey Kurenkov, Ajay Mandlekar, Li Fei-Fei, Silvio Savarese, and Roberto Martín-Martín. Error-aware imitation learning from teleoperation data for mobile manipulation, 2021.

[54] Jimmy Wu, Rika Antonova, Adam Kan, Marion Lepert, Andy Zeng, Shuran Song, Jeannette Bohg, Szymon Rusinkiewicz, and Thomas Funkhouser. Tidybot: Personalized robot assistance with large language models. *Autonomous Robots*, 2023.

[55] Sriram Yenamandra, Arun Ramachandran, Mukul Khanna, Karmesh Yadav, Devendra Singh Chaplot, Gunjan Chhablani, Alexander Clegg, Theophile Gervet, Vidhi Jain, Ruslan Partsey, Ram Ramrakhya, Andrew Szot, Tsung-Yen Yang, Aaron Edsinger, Charlie Kemp, Binit Shah, Zsolt Kira, Dhruv Batra, Roozbeh Mottaghi, Yonatan Bisk, and Chris Paxton. The homerobot open vocab mobile manipulation challenge. In *Thirty-seventh Conference on Neural Information Processing Systems: Competition Track*, 2023. URL https://aihabitat.org/challenge/2023_homerobot_ovmm/.

[56] Tianle Zhang, Dongjiang Li, Yihang Li, Zecui Zeng, Lin Zhao, Lei Sun, Yue Chen, Xuelong Wei, Yibing Zhan, Lusong Li, and Xiaodong He. Empowering embodied manipulation: A bimanual-mobile robot manipulation dataset for household tasks, 2024.

[57] Minjie Zhu, Yichen Zhu, Jinming Li, Junjie Wen, Zhiyuan Xu, Zhengping Che, Chaomin Shen, Yaxin Peng, Dong Liu, Feifei Feng, et al. Language-conditioned robotic manipulation with fast and slow thinking. *arXiv preprint arXiv:2401.04181*, 2024.

APPENDIX

1) Appendix A: More Dataset Details
2) Appendix B: Data Collection
3) Appendix C: More Model Implementation Details
4) Appendix D: Cross Entropy Loss Details

A. More Dataset Details

This section provides the full table, Table IV, of related datasets mentioned in Section II. Additionally, all the data types collected for the dataset are displayed in JSON format in Figures 4 and 5. Finally, it explains how the scoring in Figure 2 is calculated.

1) Scoring System: We describe the scoring system highlighted in Figure 2 in more detail. Figure 2 compares our dataset and other datasets. The scoring system is divided into 2 axes: Capabilities and Complexity. The Capabilities axis measures the capabilities and modalities the dataset includes out of the four main categories: Language, Navigation, Manipulation, and Perception. The purpose is to see how diverse the dataset inputs are. We calculate the Capabilities score C_1 as follows:

$$\begin{aligned} L&: \text{Natural Language} \\ N&: \text{Navigation} \\ M&: \text{Manipulation} \\ P&: \text{Perception} \\ R&: \text{Real} \\ S&: \text{Sim} \end{aligned}$$

$$C_1 = (L + N + M + P)(R + S)$$

where each attribute is 1 if it's present, 0 otherwise. The maximum Capabilities score is 8. Conversely, the Complexity axis shows how difficult and diverse the tasks and environments are. We define the Complexity score C_2 as follows:

$$C_2 = \log_{10}(te)$$

where t and e are the number of trajectories and environments each dataset has, respectively. We take the logarithm of this product to account for large differences across datasets. To get the plotted score, we take the geometric mean G of the axes:

$$G = \sqrt{C_1 \cdot C_2}$$

2) Data Types: There are 24 unique data types across both the simulated and real-world data. Specifically, there are 4 simulated-only, 12 real-only, and 8 of both. All of the simulated ones are 13, and 20 are all of the real ones. Figures 4 and 5 show all of data types.

B. Data Collection

This section displays the maps that the simulation and real robots used during data collection in Figures 6 and 7. It also provides further details on how the data was collected.

1) Simulation Command Collection Website: We use Prolific, a crowdsourcing platform for researchers, to collect natural language commands for tasks a robot can do in the RoboTHOR simulator. The participants utilized a website we developed to watch a tutorial video, read instructions about the task, explore the five RoboTHOR environments to know what commands to give, and then input their commands. The website was hosted on AWS Elastic Beanstalk and the inputted commands were saved on an AWS S3 bucket. Screenshots of the website are displayed in Figure 8.

2) Crowdsourcing: The Prolific participants were paid an hourly wage of US$10, totaling US$380. Subsequently, the simulation teleoperation of those commands was done by a separate group of paid participants. They were recruited via Google Forms. The recruitment instructions are shown in Figure 15a. This group of participants was paid US$10/hr via Amazon gift cards, totaling US$630.

For the real-world data, a different group of seven was recruited to explore the environments and give natural language commands of tasks the robot can do, similar to the simulation. They were recruited via Google Forms. The recruitment instructions are shown in Figure 15b. The participants chose to volunteer for this task. Finally, the real robot teleoperation of those commands was done by one of the authors.

3) Real Robot Teleoperation: One person teleoperated the Spot robot via joysticks and a tablet. To collect, organize, and save the data, a laptop was mounted and connected via Ethernet to the Spot. The reason for mounting a laptop was so that the collection frequency of 3 Hz remains consistent, unlike using WiFi. 3 Hz in particular was an inspiration from the RT-1 paper [6].

C. More Model Implementation Details

The implementation details of the RT-1 and ALFRED Seq2Seq models are described in Section IV-A and are expanded upon in this section. The maximum action prediction limit is 1500.

1) RT-1: In addition to the modifications mentioned in Section IV-A, others were made as well:

- We utilized the Pytorch implementation of the RT-1.[3] This required us to pretrain the model before fine-tuning on LaNMP. We ran pretraining on the Bridge [10] dataset, which contains language annotations and ground-truth trajectories for diverse environments - tabletop, kitchen (also toy kitchen), and other household environments.
- Pretraining used subsample of 500 episodes per epoch from Bridge using a learning rate of 1×10^{-4}, batch size of 8, and a window size of 6 previous observations.
- For fine-tuning, we performed backpropagation and weight updates across all parameters of the RT-1 model, i.e. no parameters were frozen.
- For the Cross-Scene and Task Generalization experiments we fine-tuned on the training set over 2 epochs with a learning rate of 1×10^{-4} and with the 'Adam' optimizer.

[3] https://github.com/Rohan138/rt1-pytorch

	Real	Sim	Natural Language	Navigation	Manipulation	Perception	Multiroom Navigation
RT-1	✓	✓	✓	✓	✓	✓	✗
Collab	✗	✓	✓	✗	✓	✓	✗
CALVIN	✗	✓	✓	✗	✓	✓	✗
GVCCI	✓	✗	✓	✗	✓	✓	✗
LA-TaskGrasp	✓	✗	✓	✗	✓	✓	✗
RoboSet	✓	✗	✓	✗	✓	✓	✗
AlphaBlock	✓	✓	✓	✗	✓	✓	✗
VIMA	✗	✓	✓	✗	✓	✓	✗
RFST	✓	✓	✓	✗	✓	✓	✗
DROID	✓	✗	✓	✗	✓	✓	✗
Dobb-E	✓	✗	✓	✗	✓	✓	✗
DAA	✓	✗	✓	✗	✓	✓	✗
NatSGD	✓	✓	✓	✗	✓	✓	✗
ALFRED	✗	✓	✓	✓	✗	✓	✗
TEACh	✗	✓	✓	✓	✗	✓	✗
Room-to-Room	✗	✓	✓	✓	✗	✓	✓
Room-Across-Room	✗	✓	✓	✓	✗	✓	✓
QUARD	✓	✓	✓	✓	✗	✓	✗
MoMaRT	✗	✓	✗	✓	✓	✓	✗
BRMData	✓	✗	✗	✓	✓	✓	✗
Conq Hose	✓	✗	✓	✓	✓	✓	✓
RoboCasa	✗	✓	✓	✓	✓	✓	✓
CoNav	✗	✓	✓	✓	✗	✓	✓
BridgeData V2	✓	✗	✓	✗	✓	✓	✗
LaNMP (Ours)	✓	✓	✓	✓	✓	✓	✓

TABLE IV: Full table of the dataset comparison regarding the different aspects and modalities.

- For the scene diversity experiment, we randomly initialized RT-1 weights and trained directly on LaNMP to control for the influence of pretrained policies and leave only dataset diversity as the only independent variable. As we were training from initialization in this experiment, we used a learning rate of 1×10^{-4} and with the 'Adam' optimizer over 4 epochs.
- All fine-tuning experiments used a batch size of 8 with a window size of 6 previous image observations (specific to RT-1) to predict the next action token. When constructing each batch, 3 trajectories from the training dataset were bucketed using a window size of 6 and further grouped into batches of 8 before being input into RT-1.

As discussed in Section IV-A all attributes of the action space were discretized (using uniform discretization across the range of each attribute) across 256 bins. For the next step, the RT-1 model was fine-tuned against a cross entropy loss between the predicted discretized tokens and the ground-truth tokens. Furthermore, we fine-tuned RT-1 to predict *deltas* or *changes* in the action attribute values at each time step, rather than absolute values of each action attribute, as we empirically found this led to more stable learning.

The original model can be found in the RT-1 paper [6].

2) ALFRED Seq2Seq: We utilized a forked improved implementation of the ALFRED Seq2Seq model.[4] The model had to be modified to work for our data, robot, and environments. In addition to the modifications mentioned in Section IV-A, the following were also made:

- All weights except the final layer were frozen during fine-tuning.
- Unfrozen weights were initialized with random values from the ranges of the actions/states.
- The final layer was swapped with a fully connected layer outputting all of the classes, which is essentially the product of the number of bins and the number of actions.
- An adapter layer that resizes dimensions was added at the start of each LSTM cell from the second one onward, due to the action output from the first cell being different from the original.
- The model was fine-tuned to predict the discretized *deltas* or *changes* of the action/state values since it led to more stable learning.

Furthermore, we experimented with fine-tuning continuous and discrete action outputs. The continuous was regression utilizing Mean Squared Error (MSE) loss, while the discrete was classification utilized CE loss. As stated in Section IV-A, we settled on classification using 256 bins for discretization.

Details on the original model can be found in the ALFRED paper [46].

D. Cross Entropy Loss Details

This section illustrates the CE loss curves for all of the experiments conducted. The loss curves, including for both training and testing, decreased over the course of training, showing the models indeed learned from the LaNMP dataset. RT-1 losses are displayed in Figures 9, 10, and 11. ALFRED's Seq2Seq losses are displayed in Figures 12, 13, and 14.

[4] https://github.com/jiasenlu/alfred

Data Schema Adopted for LaNMP Dataset - Simulation Environment in AI2THOR

```
{
    "nl_command": "Find the pepper and put it on top of the green chair with a blue pillow on it.",
    "scene": "FloorPlan_Train8_1",
    "steps": [
        {
            "sim_time": 0.19645099341869354,
            "wall-clock_time": "15:49:37.334",
            "action": "Initialize",
            "state_body": [  # robot pose
                3.0,
                0.9009992480278015,
                -4.5,
                269.9995422363281
            ],
            "state_ee": [  # end-effector pose
                2.5999975204467773,
                0.8979992270469666,
                -4.171003341674805,
                -1.9440563492718068e-07,
                -1.2731799533306385,
                1.9440386333307377e-07
            ],
            "hand_sphere_radius": 0.05999999865889549
            "held_objs": [],
            "held_objs_state": {},
            "inst_det2D": {
                "keys": [  # identified instances in the environment
                    "Wall_4|0.98|1.298|-2.63",
                    "Wall_3|5.43|1.298|-5.218",
                    "RemoteControl|+01.15|+00.48|-04.24",
                    ...],
                "values": [  # bounding box coordinates of each instance
                    [418, 43, 1139, 220],
                    [315, 0, 417, 113],
                    [728, 715, 760, 719],
                    ...]
            },
            "rgb": "./rgb_0.npy",  # path of visual data for this timestep
            "depth": "./depth_0.npy",
            "inst_seg": "./inst_seg_0.npy",
        }
    ]
}
```

Fig. 4: Example JSON file from the simulation dataset that includes all the collected data types.

Data Schema Adopted for LaNMP Dataset - Robot Environment with Boston Dynamics Spot


```
{
    "language_command": "Take the cup from the table in the dining area which is closest to the stairs and bring it to the table near the
        couches in the corner of the big dining room besides the windows.",
    "scene_name": "upstairs",
    "wall_clock_time": "05:29:40.117",
    "left_fisheye_rgb": "./Trajectories/trajectories/data_33/folder_0.zip/left_fisheye_image_0.npy",  # path of visual data for this timestep
    "left_fisheye_depth": "./Trajectories/trajectories/data_33/folder_0.zip/left_fisheye_depth_0.npy",
    "right_fisheye_rgb": "./Trajectories/trajectories/data_33/folder_0.zip/right_fisheye_image_0.npy",
    "right_fisheye_depth": "./Trajectories/trajectories/data_33/folder_0.zip/right_fisheye_depth_0.npy",
    "gripper_rgb": "./Trajectories/trajectories/data_33/folder_0.zip/gripper_image_0.npy",
    "gripper_depth": "./Trajectories/trajectories/data_33/folder_0.zip/gripper_depth_0.npy",
    "left_fisheye_instance_seg": "./Trajectories/trajectories/data_33/folder_0.zip/left_fisheye_image_instance_seg_0.npy",
    "right_fisheye_instance_seg": "./Trajectories/trajectories/data_33/folder_0.zip/right_fisheye_image_instance_seg_0.npy",
    "gripper_fisheye_instance_seg": "./Trajectories/trajectories/data_33/folder_0.zip/gripper_image_instance_seg_0.npy",
    "body_state": {"x": 1.3496176111016192, "y": 0.005613277629761049, "z": 0.15747965011090911},
    "body_quaternion": {"w": 0.04275326839680784, "x": -0.0008884984706659231, "y": -0.00030123853590331847, "z": 0.999085220522855},
    "body_orientation": {"r": -0.003024387647253151, "p": 0.017297610440263775, "y": 3.05395206999625},
    "body_linear_velocity": {"x": 0.00015309476140765987, "y": 0.001022209848280799, "z": 0.0001717336942742603},
    "body_angular_velocity": {"x": 4.532841128101956e-05, "y": 0.003003578426140623, "z": -0.0046712267592016726},
    "arm_state_rel_body": {"x": 0.5535466074943542, "y": -0.00041040460928343236, "z": 0.2611726224422455},
    "arm_quaternion_rel_body": {"w": 0.9999685287475586, "x": -0.0011630485532805324, "y": 0.007775876671075821, "z": 0.007775876671075821},
    "arm_orientation_rel_body": {"x": -0.0023426745301198485, "y": 0.015549442728134426, "z": -0.0021046873064696214},
    "arm_state_global": {"x": 0.7976601233169699, "y": -0.00041040460928343236, "z": 0.2611726224422455},
    "arm_quaternion_global": {"w": 0.043804580215426665, "x": -0.008706641097541701, "y": -0.0011317045101892497, "z": 0.9990015291187636},
    "arm_orientation_global": {"x": -0.003024387647253151, "y": 0.017297610440263775, "z": 3.05395206999625},
    "arm_linear_velocity": {"x": 0.002919594927038712, "y": 0.004658882987521996, "z": 0.012878690074243797},
    "arm_angular_velocity": {"x": -0.01867944403436315, "y": 0.02911512882983833, "z": -0.008279345145765714},
    "arm_stowed": 1,  # Boolean
    "gripper_open_percentage": 0.39261579513549805,
    "object_held": 0,  # Boolean
    "feet_state_rel_body": [
        {"x": 0.3215886056423187, "y": 0.17115488648414612, "z": -0.5142754912376404},
        {"x": 0.32302412390708923, "y": -0.17028175294399261, "z": -0.5178792476654053},
        {"x": -0.27173668146133423, "y": 0.16949543356895447, "z": -0.5153297185897827},
        {"x": -0.2700275778770447, "y": -0.1685962975025177, "z": -0.5157276391983032}],
    "feet_state_global": [
        {"x": -0.3341075772867149, "y": -0.14278573670828154, "z": -0.5149532673227382},
        {"x": -0.3063631798978494, "y": 0.19752718640765313, "z": -0.518328069669068},
        {"x": 0.25719142551156154, "y": -0.19181889447285838, "z": -0.5149682779363334},
        {"x": 0.2843717159282008, "y": 0.1451830347804529, "z": -0.5151399962832868}],
    "all_joint_angles": {
        "fl.hx": 0.010119102895259857,
        "fl.hy": 0.7966763973236084,
        "fl.kn": -1.576759934425354, ...},
    "all_joint_velocities": {
        "fl.hx": -0.00440612155944109,
        "fl.hy": -0.004167056642472744,
        "fl.kn": -0.007508249022066593, ...}
}
```


Fig. 5: Example JSON file from the real-world dataset that includes all the collected data types.

FloorPlan_Train1_3

(a)

FloorPlan_Train5_1

(b)

FloorPlan_Train7_5

(c)

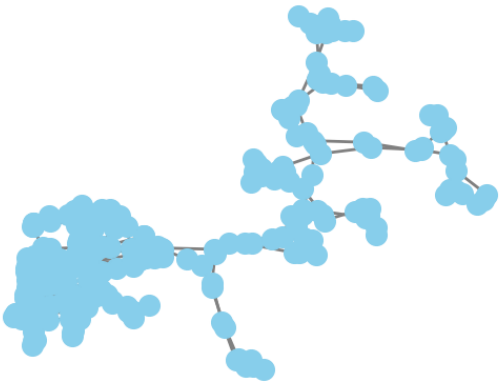
FloorPlan_Train8_1

(d)

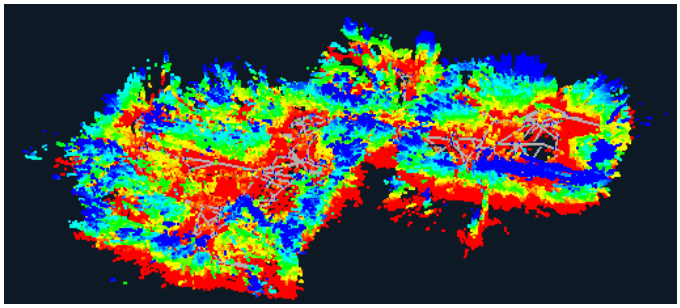
FloorPlan_Train12_3

(e)

Fig. 6: Simulated Environment Maps

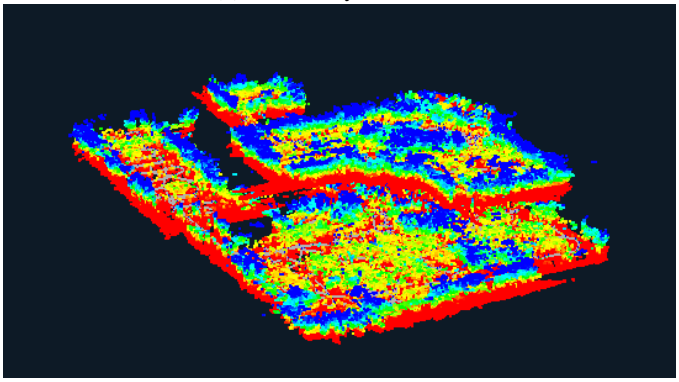


(a) Laboratory Graph



(b) Laboratory Point Cloud

(c) Multi-Floor Graph



(d) Multi-Floor Point Cloud

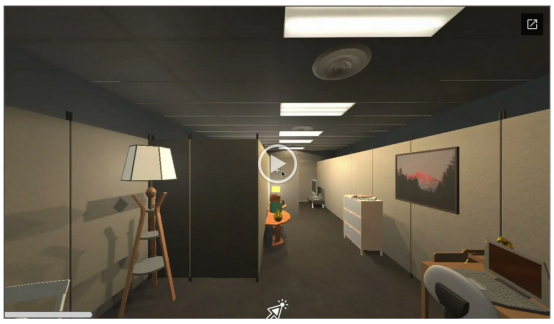
Fig. 7: Real-World Environment Robot Maps

Instructions

Welcome and thank you for participating in our data collection! On the next page, you will write **15** unique commands to a robot in 5 virtual household scenes, i.e., **3 commands per scene**.

A personal assistant robot can navigate the virtual environment, pick up and place objects. It perceives the environment with its camera. Please see the video below of how the robot operates in the environment.

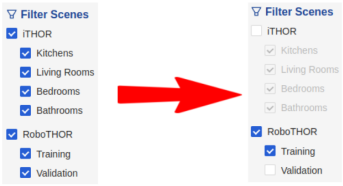
Video:



(a) Main instructions along with a video tutorial.

Commands

1. In the interactive section below, please wait for it to finish loading, **then** unselect the **Filter Scenes** checkboxes as indicated by the image below:



2. In the same interactive section, please scroll down (note there are two scrollers, one for the web page and one for the interactive section) to select **Floorplan_Train1_3** that looks like this (this will be scene 1):

(b) Detailed instructions on how to explore the environments.

For Scene 1, please select
Floorplan_Train1_3

Starting position: By laptop

Scene 1 Command 1:

Scene 1 Command 2:

Scene 1 Command 3:

For Scene 2, please select
Floorplan_Train5_1

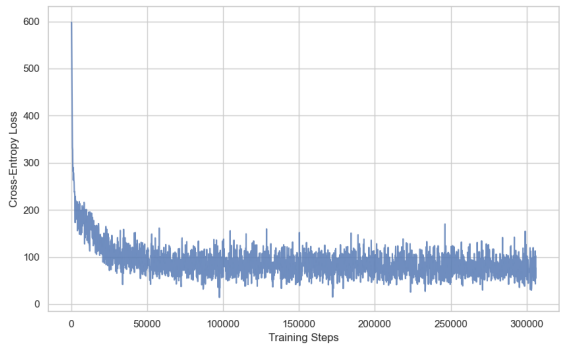
Starting position: By yellow couch

Scene 2 Command 1:

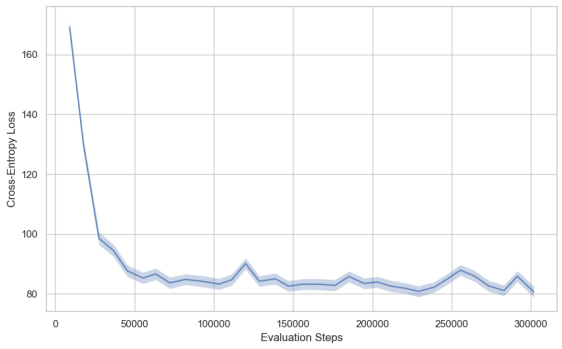
Scene 2 Command 2:

Scene 2 Command 3:

(c) Forms for users to input their commands.

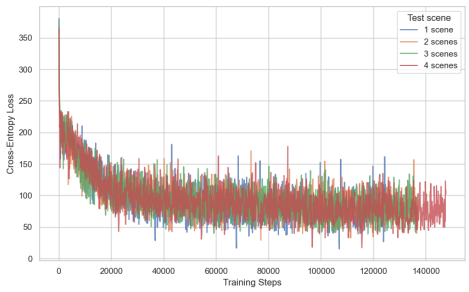


(a) Train

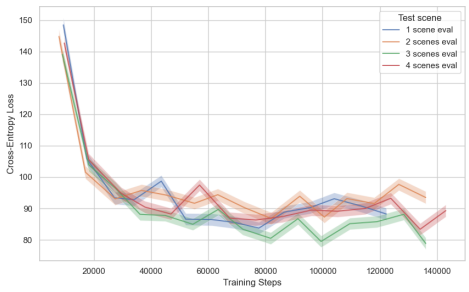


(b) Test

Fig. 9: RT-1 Task Generalization CE Loss Curves

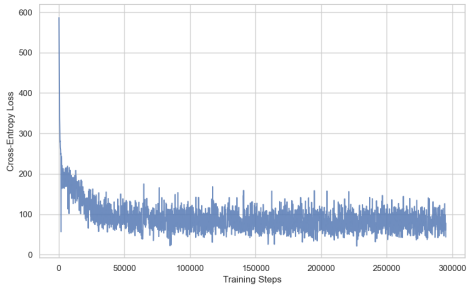


(a) Train

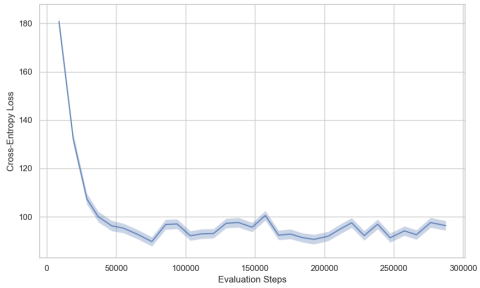


(b) Test

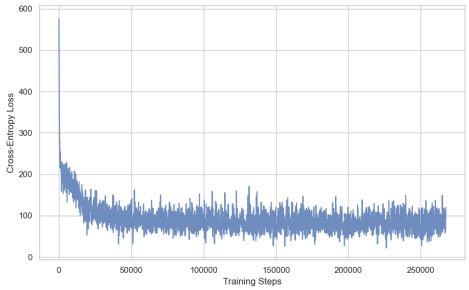
Fig. 10: RT-1 Scene Diversity CE Loss Curves



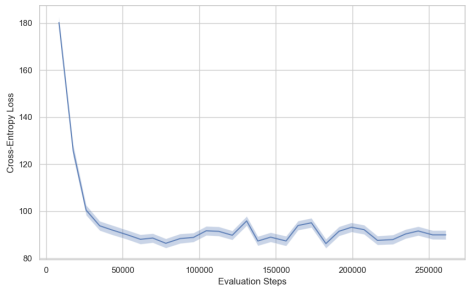
(a) Folds 2-5 Train



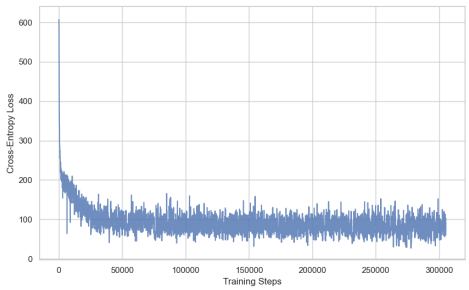
(b) Fold 1 Test



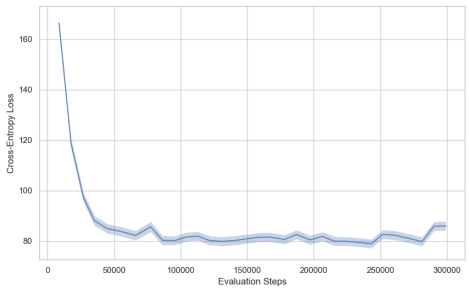
(c) Folds 1, 3-5 Train



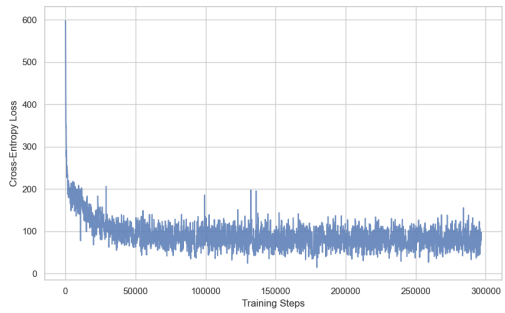
(d) Fold 2 Test



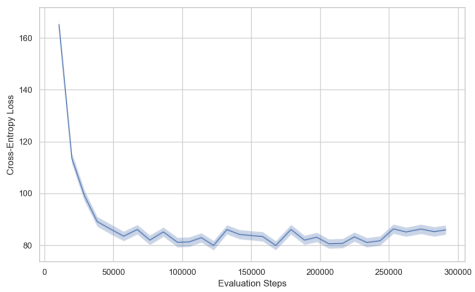
(e) Folds 1-2, 4-5 Train



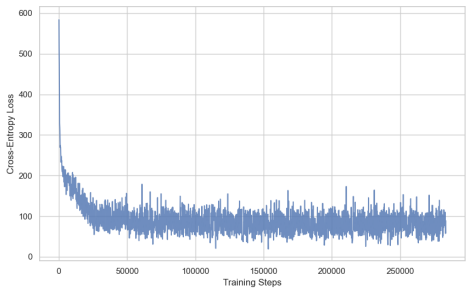
(f) Fold 3 Test



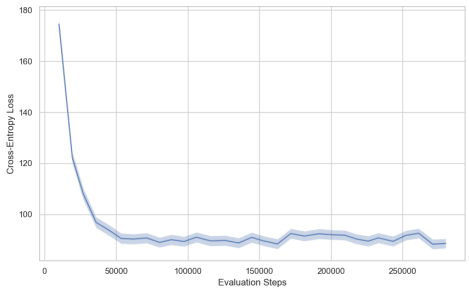
(g) Folds 1-3, 5 Train



(h) Fold 4 Test

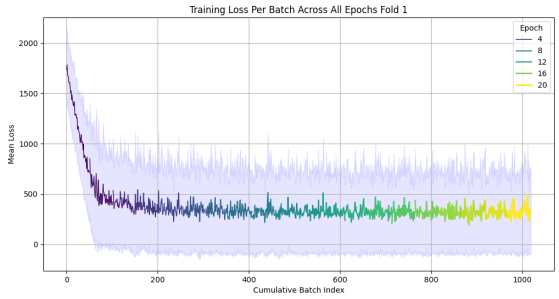


(i) Folds 1-4 Train

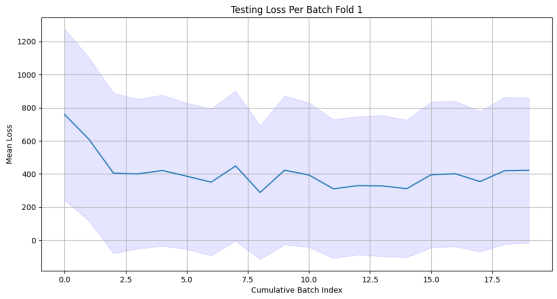


(j) Fold 5 Test

Fig. 11: RT-1 Cross-Validation CE Loss Curves

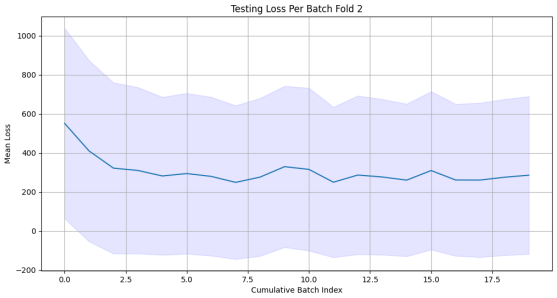


(a) Folds 2-5 Train



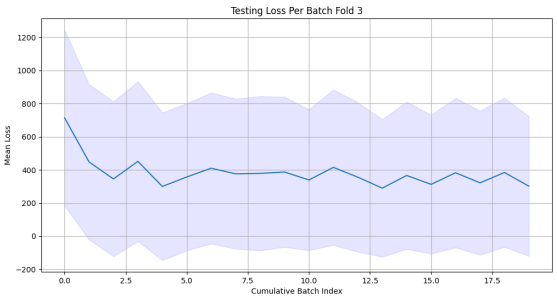
(b) Fold 1 Test

(c) Folds 1, 3-5 Train

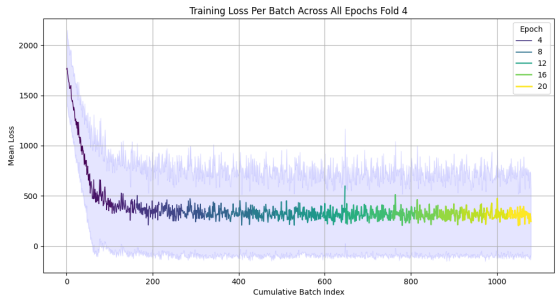


(d) Fold 2 Test

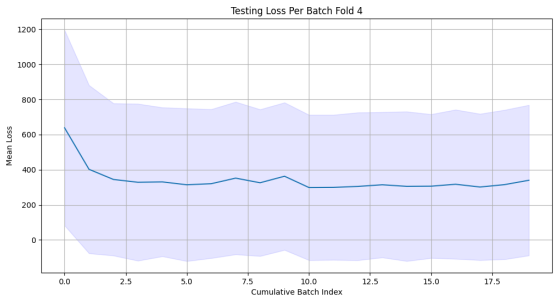
(e) Folds 1-2, 4-5 Train



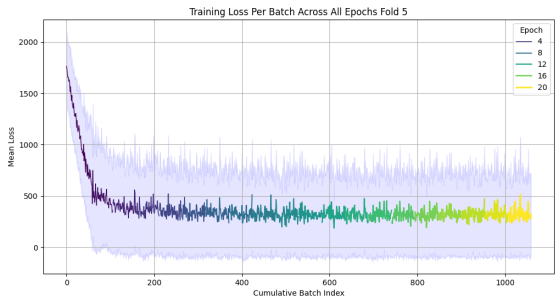
(f) Fold 3 Test



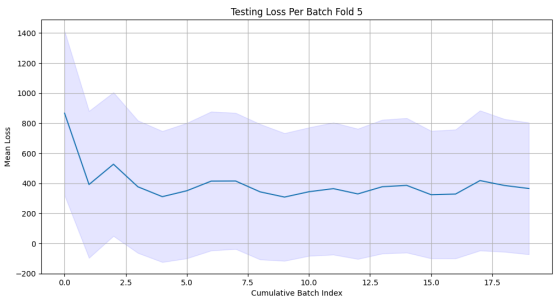
(g) Folds 1-3, 5 Train



(h) Fold 4 Test

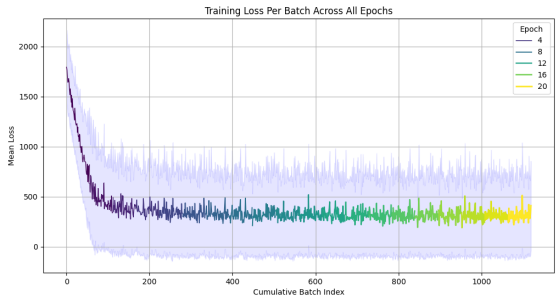


(i) Folds 1-4 Train

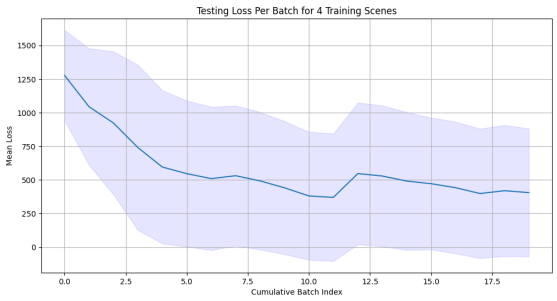


(j) Fold 5 Test

Fig. 12: ALFRED Seq2Seq Cross-Validation CE Loss Curves

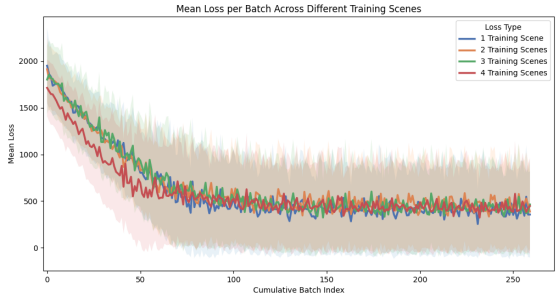


(a) Train

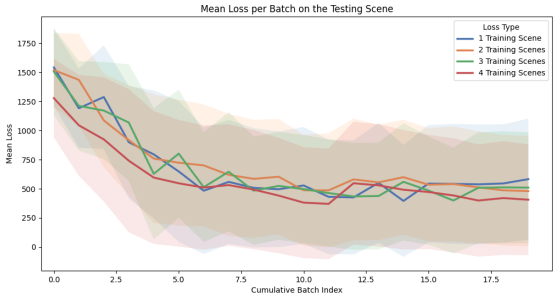


(b) Test

Fig. 13: ALFRED Seq2Seq Task Generalization CE Loss Curves

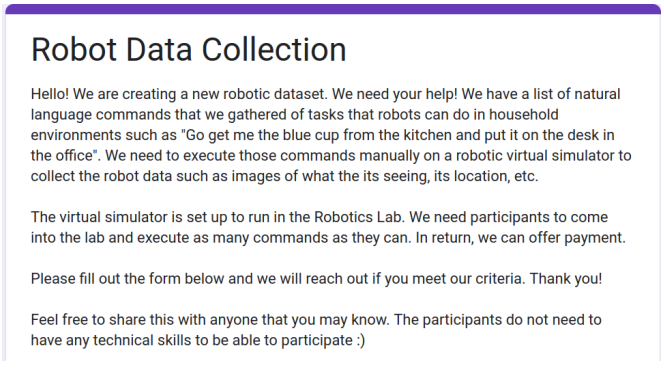


(a) Train



(b) Test

Fig. 14: ALFRED Seq2Seq Scene Diversity CE Loss Curves



Robot Data Collection

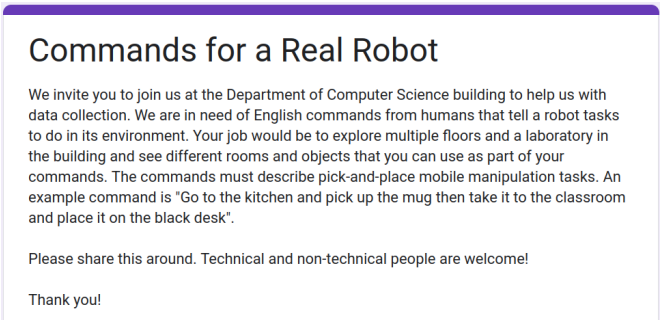
Hello! We are creating a new robotic dataset. We need your help! We have a list of natural language commands that we gathered of tasks that robots can do in household environments such as "Go get me the blue cup from the kitchen and put it on the desk in the office". We need to execute those commands manually on a robotic virtual simulator to collect the robot data such as images of what the its seeing, its location, etc.

The virtual simulator is set up to run in the Robotics Lab. We need participants to come into the lab and execute as many commands as they can. In return, we can offer payment.

Please fill out the form below and we will reach out if you meet our criteria. Thank you!

Feel free to share this with anyone that you may know. The participants do not need to have any technical skills to be able to participate :)

(a) Simulation Teleoperation



Commands for a Real Robot

We invite you to join us at the Department of Computer Science building to help us with data collection. We are in need of English commands from humans that tell a robot tasks to do in its environment. Your job would be to explore multiple floors and a laboratory in the building and see different rooms and objects that you can use as part of your commands. The commands must describe pick-and-place mobile manipulation tasks. An example command is "Go to the kitchen and pick up the mug then take it to the classroom and place it on the black desk".

Please share this around. Technical and non-technical people are welcome!

Thank you!

(b) Real Command Collection

Fig. 15: Google Forms