Dialog-based Skill and Task Learning for Robot

Weiwei Gu, Suresh Kondepudi, Lixiao Huang, Nakul Gopalan
Arizona State University
Email: weiweigu@asu.edu, nkondepu@asu.edu, Lixiao.Huang@asu.edu, ng@asu.edu

***Abstract*—Continual and interactive robot learning is a challenging problem as the robot is present with human users who expect the robot to learn novel skills to solve novel tasks perpetually with sample efficiency. In this work we present a framework for robots to query and learn visuo-motor robot skills and task relevant information via natural language dialog interactions with human users. Previous approaches either focus on improving the performance of instruction following agents, or passively learn novel skills or concepts. Instead, we used dialog combined with a language-skill grounding embedding to query or confirm skills and/or tasks requested by a user. To achieve this goal, we developed and integrated three different components for our agent. Firstly, we propose a novel visual-motor control policy ACT with Low Rank Adaptation (ACT-LoRA), which enables the existing state-of-the-art Action Chunking Transformer [28] model to perform few-shot continual learning. Secondly, we develop an alignment model that projects demonstrations across skill embodiments into a shared embedding allowing us to know when to ask questions and/or demonstrations from users. Finally, we integrated an existing Large Language Model (LLM) to interact with a human user to perform grounded interactive continual skill learning to solve a task. Our ACT-LoRA model learns novel fine-tuned skills with a 100% accuracy when trained with only five demonstrations for a novel skill while still maintaining a 74.75% accuracy on pre-trained skills in the RLBench dataset where other models fall significantly short.**

I. Introduction

Chai et al.2019 define natural interaction as an interaction between a human and a robot that resembles the way of natural communication between human beings such as dialogues, gestures, etc. without requiring the human to have prior expertise in robotics. The capability of learning tasks and acquiring new skills from natural interactions is desirable for robots as they need to perform unique tasks for different users. One direction of this interaction channel is well studied as instruction following [2, 4, 3], where the robot performs the tasks requested by the human via natural language. Our work focuses on the other side of this communication channel, where the robot starts the conversation with human when it needs their help. This reverse direction of communication plays an important role for robots to learn with non-expert human users as it enables robots to convey their lack of task knowledge to perform tasks in a way that non-expert users can understand. Furthermore, our framework can leverage the feedback from users and learn to perform the task.

Human-Robot interaction via language is a well studied problem [6, 4, 3, 12]. Robot agents have been able to interpret language instructions from the human users, and perform visual-motor policies to complete tasks [2, 4, 3]. These methods rely on the emergent behaviors of large models, and do not continually learn new skills or add to their task or skill knowledge. To address this issue, some works have proposed life-long learning for robot agents [24, 16, 12, 26]. Some recent works learn neural visuo-motor skills in a continual setting [26, 27, 18]. However, these approaches are passive and do not query the user for novel skills that the agent might need to complete given tasks.

We propose a framework that utilizes dialog to enable the robot agent to express its need for new skill or task information actively. When encountering a novel task, our robot agent starts a conversation with the human user to learn to execute the task. Throughout the interaction, the robot agent specifies the help that it needs from the human user via natural language, such as a human enacting the skill to find a feasible skill within the existing set of skills to perform the task or requesting multiple robot demonstrations to learn a completely novel skill for this specific task. Our contributions are as follows:

1) We compare ACT-LoRA against the baseline ACT model on few-shot continual learning on RLBench dataset. Our model demonstrates its strong adaptability by achieving 100% success rate on the tasks that it finetuned on with only 5 demonstrations. Furthermore, it achieves an average success rate of 74.75% on the tasks that it is pre-trained on, showing that our policy is effective in preventing catastrophic forgetting.
2) We present a model that can determine whether a pair of demonstrations of different embodiments, in our case human enactment of a skill or a robot demonstration, are performing the same task. Our alignment model achieves an overall accuracy of 91.4% on the RH20T dataset on aligning demonstrations from humans and robot.

II. Problem Formulation

We formulate a task solving problem where both the robot and the human agent can take actions on their turns. There is a joint physical state s of the world shared by both the human and the robot. In each turn, n, either the human or the robot acts, one after the other. Each turn can take longer than one time step, t, and continues until the robot or the human indicates a turn to be over. The actions can be physical actions represented by a_h, and a_r for the human and the robot actions respectively, or speech acts l_h and l_r for the human and the robot speech respectively for the human-robot grounded dialog. The problem has an initial state s_0 and a task θ specified by the human using a speech act l_h^0. Each of these actions updates the joint physical state s of the world,

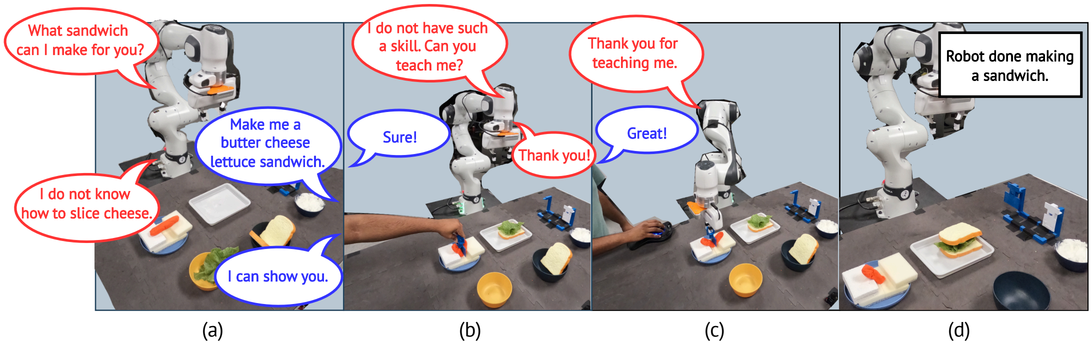
What sandwich can I make for you?
Make me a butter cheese lettuce sandwich.
I do not know how to slice cheese.
I can show you.
I do not have such a skill. Can you teach me?
Sure!
Thank you!
Thank you for teaching me.
Great!
Robot done making a sandwich.
(a)
(b)
(c)
(d)

Fig. 1. An example run of our framework in the user study. (a) The user asks the robot to make a sandwich , some of the tasks to make a sandwich are known but the robot does not know a dynamic skill to make the sandwich, slicing cheese. (b) So the human enacts cutting cheese with their own hands to show the robot the type of skill needed , but the robot has never seen such a skill before so it asks for help. (c) The user controls the robot to perform said skill. (d) The robot learns the novel skill from the human demonstration and is able to complete the entire sandwich on its own in the next interaction.

and internal dialog state s^d of the robot. The dialog state is hidden from the human user, but the human receives speech observations for the same. Over multiple turns and actions taken by the human and the robot these physical and robot states update over time. The objective of this turn taking problem is to complete the task θ. We measure the task completion rates for this interaction problem. Moreover, in our specific instance of the problem the human also teaches behaviors to the robot, we also measure the success of the individual learned behaviors within the task in simulation.

III. Methods

The goal of our framework is a robot agent that 1) actively generalizes its known skills to novel tasks when it is applicable; 2) queries the user for unknown skills; and 3) learns new skills with only a few instances. When encountered a task θ, the robot agent first searches for a learned skill using semantic representation, which comes from the language embedding of the linguistic description of the skills and tasks. This is a challenging question as the robot needs to know what it does not know. This work is performed by our queryable skill library. If the agent fails to find any usable skill for the task based on the semantic information, it attempts to search for a learned skill using skill representations, which come from human demonstrations and robot trajectories. We developed a novel sample efficient continual skill learning approach ACT-LoRA for this task. The robot agent can directly execute the task τ whenever it finds a learned skill that aligns with τ in either the semantic space or the skill space, and learns a novel skill to execute the task otherwise. We use an LLM to enable the robot agent to interact with the human user based on the information from the queryable skill library.

A. How to Know What the Robot Does not Know a.k.a. a Queryable Skill Library

The skill library consists of four parts - a text encoder E_{text}; a human demonstration encoder E_{human}; a robot trajectory encoder E_{robot}; and a set of learned skills $\mathcal{S} = \{S_1, \ldots, S_k\}$. Each learned skill S_i is a tuple of a linguistic description and a robot trajectory, denoted as $S_i = (l_i, \tau_i)$. The linguistic representation r_i^l and skill representation r_i^s of skill S_i can be obtained by encoding l_i and τ_i with the corresponding encoder, denoted as $r_i^l = E_{\text{text}}(l_i)$, and $r_i^s = E_{\text{robot}}(\tau_i)$ respectively.

Finding a usable skill from the skill library. The skill library is provided two inputs to find an appropriate skill to execute the task θ, the linguistic description l_θ and a human demonstration d_θ for the task. We obtain the linguistic or semantic representation and skill representation for the task by encoding the linguistic description and the human demonstration with the corresponding encoders. We then compute two sets of similarity scores between the task θ and any known skill S_i for both the linguistic representation and the skill representation. The state machine within the interaction module of the agent decides the skill to use to execute the task θ based on these scores.

We use a pre-trained CLIP as a text encoder E_{text}. For E_{robot} and E_{human}, we first extract features from each frame using a Resnet-18 [13], and then encode the sequence using a transformer encoder [25]. The robot trajectory encoder E_{robot} and the human demonstration encoder E_{human} are trained to encode the human demonstrations and robot trajectories into the same latent space. The two encoders are jointly trained with tuples (d, τ, y), where d denotes human demonstration videos, τ denotes robot trajectories, and y is the label of whether the human demonstration and the robot trajectory is in the same task. We use a cosine similarity loss to learn this embedding with a hyperparamter ψ to act as a margin to declare a human demonstration to be the same skill as a skill the robot knows. More details about learning this embedding space are in the Appendix A.

B. Interaction Module using a Large Language Model (LLM)

The dialog state s^d in our pipeline is maintained with an internal state machine. The state machine uses an LLM, ChatGPT 4 [1], as the natural language generator to produce speech acts for the robot agent. This state machine with the LLM has two major functionalities. Firstly, it interacts with the human user to asks for demonstrations or explanations based

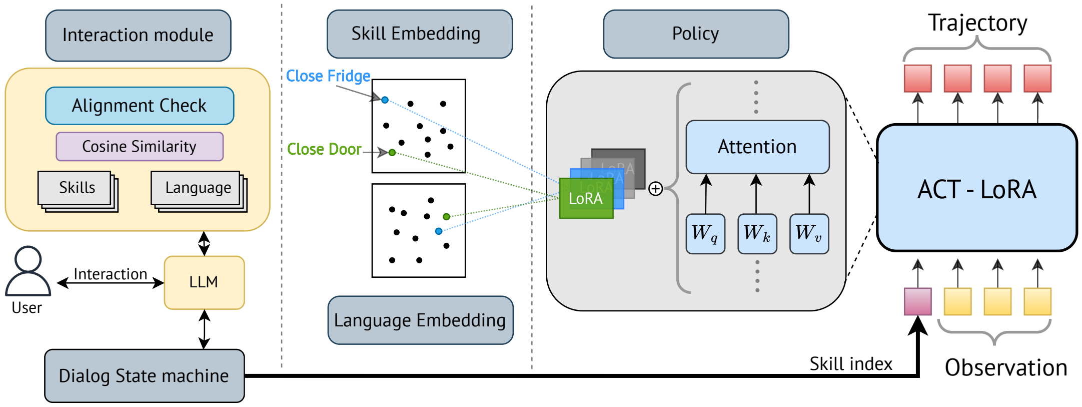

Fig. 2. Overview of our framework. The LLM serves as the interactive module and understands a user's feedback. The skill library provides representations for learned skills and novel demonstrations. The policy model executes the tasks based on the user's instructions. The agent searches for an executable skill by comparing the language representation and skill representation of the novel task with those of the known skills using a cosine similarity metric. We integrate Low-Rank Adaptor(LoRA) with the Action Chunking Transformer(ACT) model as our policy, which is capable of learning fine-grained skills and continually learning novel skills without catastrophic forgetting.

on the checks from our queryable skill library. Secondly, the interaction module also interprets the user's language feedback to update the dialog state s^d. The interaction module is given the autonomy to continue the dialog with the user until that it acquires the designated information for the agent. The module can also explain the dialog state s^d with language to the user explaining the robot's confusion.

C. ACT-LoRA as Visual-motor Policy

Combining Low-Rank Adaptor with Action Chunking. Adapter-based methods [14, 21, 11, 17] have exhibited promising capabilities of light-weight and data-efficient fine-tuning of neural networks across various domains such as NLP [14, 11], and computer vision [21]. Liu et al. [18] extend Low-Rank Adaptor(LoRA) into robotics with TAIL, enabling a simulated robot to continually adapt to novel tasks without forgetting the old ones. Unfortunately in our experiments TAIL [18] fails to provide high precision control on the robot leading to a lot of failures in even short skills. On the other hand, Action Chunking Transformer(ACT) [28] is capable of performing fine-grained tasks with high precision, but cannot be directly used for continual learning due to catastrophic forgetting. Therefore, we introduce LoRA adaptor to the ACT model, obtaining both the precision from action chunking and the capability of continual learning from the LoRA adaptor. We want to point out that we are using TAIL [18] as the baseline in this work as it is the closest continual learning agent to our approach.

Continual Imitation Learning. Our policy needs to continually learn new skills from demonstrations throughout the agent's lifespan. The robot agent is initially equipped with K skills $\{\mathcal{S}_1, \ldots, \mathcal{S}_K\}$. Whenever the robot agent encounters a task that requires a novel skill $\mathcal{S}_n, n > K$, it needs to adapt its existing policy π to the novel skill without forgetting any of the existing skills $\mathcal{S} \in \{\mathcal{S}_1, \ldots, \mathcal{S}_{n-1}\}$. Provided a number of demonstration trajectories for each skill, the continual learning policy of the robot agent can then be optimized with a behavior cloning loss, which in this case we use L_1 loss for action chunks following [28]. On top of the policy of the vanilla ACT model π_ϕ, the LoRA adaptor introduce a small set of additional low-rank parameters ϕ_i for each skill $\mathcal{S}_i$. During the pre-training phase, the additional parameters $\phi_1, \ldots, \phi_K$ for skills $\mathcal{S}_1, \ldots, \mathcal{S}_K$ are jointly trained with the model's parameter ϕ. When we are finetuning with a skill $\mathcal{S}_n, n > K$, we freeze the model's original parameters ϕ, and only allow gradient updates to the parameters from the task-specific adaptor ϕ_n. Such finetuning strategy prevents the policy from catastrophic forgetting the skills that it already possessed when adapting to novel skills.

Model	Pre-trained Skills(SR)	Few-shot Skills(SR)
ACT-LoRA	74.75	100.0
ACT	1.5	100.0
TAIL	0.25	5.0

TABLE I
EXPERIMENTAL RESULTS ON RLBENCH SIMULATOR. **PRE-TRAINED SKILLS(SR)** MEASURES THE POLICIES' AVERAGE SUCCESS RATE ON THE 8 SKILLS THAT POLICIES ARE PRE-TRAINED ON. **FEW-SHOT SKILLS(SR)** MEASURES THE POLICIES' AVERAGE SUCCESS RATE ON THE 6 NEW SKILLS THAT THEY ARE FINETUNED ON OVER 50 ROLLOUTS. DETAILED RESULTS OF EACH SKILL ARE IN APPENDIX A.

IV. Experimental Results

In this section, we present two sets of experimental results. Firstly, we present the results of our policy on few-shot continual imitation learning in the simulated RLBench environment [15]. These experiment results show that our behavior cloning model is able to continually to learn novel skills with only few demonstrations and avoid catastrophic forgetting. Then, we evaluate our demonstration alignment model on a subset of the RH20T dataset [10], and demonstrate that we are able to project demonstrations from different embodiments into the same latent space.

A. Experiments on Continual Imitation Learning

We evaluate our policy on few-shot continual imitation learning using the RLBench environment [15]. A total of 14 skills are chosen from the pre-defined skills of the environment, 8 for pre-training and 6 for continual training. We train the policy using 1000 trajectories for each of the pre-training skills, and finetune it using 5 demonstrations for each of the continual training skills. The SoTA visual policy model ACT [28] and SoTA continual policy learning model TAIL [18] were chosen as the baselines for comparison against our model. Our model learns novel skills with 100% accuracy while maintaining its pre-trained performance at 74.75% demonstrating its suitability for continual learning. We observed TAIL [18] to fail in tasks which require precision, and ACT fail to remember older skills.

B. Experiments with our alignment model

Our alignment model is evaluated on a subset of the RH20T dataset [10], which includes robot trajectories for diverse range of tasks and their corresponding human demonstration videos. Our alignment model achieves 91.4% in overall accuracy in distinguishing whether a pair of demonstrations are performing the same task. The detailed results are in Appendix A.

V. Related Work

Skill Discovery and Continual Learning. The area of visuo-motor continual learning is getting a lot of attention recently [26, 27, 18]. Wan et al. [26] discover new skills from segments of demonstrations by unsupervised incremental clustering. Xu et al. [27] learn the skill representation by aligning skills from different embodiments, and can re-compose the learned skills to complete a novel combination. Liu et al. [18] introduce task-specific *adapters* using low-rank adaptation techniques [14], preventing the agent from forgetting the learned skills when learning the new skills. However, these frameworks assume the presence of the demonstrations for the new tasks, and only discover skills in a passive fashion. Our proposed framework actively reasons and requests the human users for the demonstrations of the unseen skills while performing the ones it knows. This reasoning is done in two stages: first the human enacts the behavior, once the robot has seen this behavior it decides if it can perform the enacted behavior or not. After this reasoning the robot can choose to source demonstrations from the human using a joystick. This is a more natural setup for a language enabled continual learning agent in the real world. Furthermore, our agent requires less than ten demonstrations from the user to discover the new task without forgetting any of the learned skills which is an improvement over existing passive continual learning methods [18, 26].

Human-Robot Dialog. Human-Robot dialog is a mature problem [9, 22, 23, 5]. Traditional methods use statistical algorithms with a pre-defined grammar, such as semantic parsing [23, 22], to connect the semantics of the dialog to the environment's perceptual inputs. On the other hand, recent advancements in natural language processing (NLP) have led to Large Language Models (LLMs) that process natural language in free form. Grounded with perceptive inputs from the environment, these LLMs have been used in robotics research generate executable plans [2]. Furthermore, Ren et al. [20] and Dai et al. [9] use LLMs to ask for human feedback for the robot agents demonstrating the importance of dialog. However, these approaches leverage planning with LLMs where as we are attempting to learn continuous visuo-motor skills on the robot by asking for help.

Active Learning. Our work is related to active learning, where a learning agent actively improves its skills by asking a human for demonstrations [23, 19, 7, 8]. Defining an appropriate metric that triggers the request for assistance or information gathering becomes the key research problem in this domain. Thomason [23] measure the semantic similarity between a newly introduced concept and the known concepts to ask for classifier labels. Chernova and Veloso [7, 8] train a confidence classifier conditioned on the current state of the agent, and request expert demonstrations when the confidence score does not meet a pre-defined threshold. Maeda et al. [19] use the uncertainty of Gaussian Processes(GPs) as the metric to trigger the request for assistance. These existing methods reason over the semantic information in a task such as the goal condition or features of classifiers that identify the goal condition. We use a cosine distance metric to measure similarity for both the semantic information from language and the behavior information of a skill.

VI. Limitations

We present an approach to teach skills to robots using techniques from active learning and continual learning while using language as a modality to query and reason over the skills known to the agent. We acknowledge that we need to conduct a user study to showcase that our agent can function with non-expert users. The turn-taking in our framework is tightly controlled, and not dynamic. Our ACT-LoRA approach while being sample efficient has been observed to have issues with heterogeneous demonstrations. We also want to compare such continual learning approaches with pre-trained policy approaches such as RT [4] to scale up the policy learning approach while maintaining sample efficiency allowing for novice users to personalize skills for their robots.

VII. Conclusion

In conclusion, we present a novel framework for robot agents to learn task relevant knowledge and skills from interactions with human users. To the best of our knowledge this is the first work to demonstrate skill learning while querying a user with dialog to express doubt. By maintaining metrics in semantic and skill similarity, our agent can actively interact with human users and adapt its known skills to novel tasks. Moreover, our framework is able to learn a completely new skill (at 100%) with only a few robot demonstrations, without affecting the performance of any existing skills (at 74.75%)fulfilling continual learning requirements in robotics.

References

[1] ChatGPT. https://www.openai.com/chatgpt, 2024. Accessed: May 30, 2024.

[2] Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Daniel Ho, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Eric Jang, Rosario Jauregui Ruano, Kyle Jeffrey, Sally Jesmonth, Nikhil J Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Kuang-Huei Lee, Sergey Levine, Yao Lu, Linda Luu, Carolina Parada, Peter Pastor, Jornell Quiambao, Kanishka Rao, Jarek Rettinghouse, Diego Reyes, Pierre Sermanet, Nicolas Sievers, Clayton Tan, Alexander Toshev, Vincent Vanhoucke, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Mengyuan Yan, and Andy Zeng. Do as i can, not as i say: Grounding language in robotic affordances, 2022.

[3] 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, Alexander 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, 2023.

[4] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, Julian Ibarz, Brian Ichter, Alex Irpan, Tomas Jackson, Sally Jesmonth, Nikhil J Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Kuang-Huei Lee, Sergey Levine, Yao Lu, Utsav Malla, Deeksha Manjunath, Igor Mordatch, Ofir Nachum, Carolina Parada, Jodilyn Peralta, Emily Perez, Karl Pertsch, Jornell Quiambao, Kanishka Rao, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Kevin Sayed, Jaspiar Singh, Sumedh Sontakke, Austin Stone, Clayton Tan, Huong Tran, Vincent Vanhoucke, Steve Vega, Quan Vuong, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. Rt-1: Robotics transformer for real-world control at scale, 2023.

[5] Joyce Y. Chai, Qiaozi Gao, Lanbo She, Shaohua Yang, Sari Saba-Sadiya, and Guangyue Xu. Language to action: Towards interactive task learning with physical agents. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 2–9. International Joint Conferences on Artificial Intelligence Organization, 7 2018. doi: 10.24963/ijcai.2018/1. URL https://doi.org/10.24963/ijcai.2018/1.

[6] Joyce Yue Chai, Maya Cakmak, and Candace L. Sidner. Teaching robots new tasks through natural interaction. *Interactive Task Learning*, 2019. URL https://api.semanticscholar.org/CorpusID:160030141.

[7] S. Chernova and M. Veloso. Interactive policy learning through confidence-based autonomy. *Journal of Artificial Intelligence Research*, 34:1–25, January 2009. ISSN 1076-9757. doi: 10.1613/jair.2584. URL http://dx.doi.org/10.1613/jair.2584.

[8] Sonia Chernova and Manuela Veloso. Confidence-based policy learning from demonstration using gaussian mixture models. page 233, 05 2007. doi: 10.1145/1329125.1329407.

[9] Yinpei Dai, Run Peng, Sikai Li, and Joyce Chai. Think, act, and ask: Open-world interactive personalized robot navigation, 2024.

[10] Hao-Shu Fang, Hongjie Fang, Zhenyu Tang, Jirong Liu, Chenxi Wang, Junbo Wang, Haoyi Zhu, and Cewu Lu. Rh20t: A comprehensive robotic dataset for learning diverse skills in one-shot, 2023.

[11] Peng Gao, Jiaming Han, Renrui Zhang, Ziyi Lin, Shijie Geng, Aojun Zhou, Wei Zhang, Pan Lu, Conghui He, Xiangyu Yue, Hongsheng Li, and Yu Qiao. Llama-adapter v2: Parameter-efficient visual instruction model, 2023.

[12] Weiwei Gu, Anant Sah, and Nakul Gopalan. Interactive visual task learning for robots. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 10297–10305, 2024.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[14] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021.

[15] Stephen James, Zicong Ma, David Rovick Arrojo, and Andrew J. Davison. Rlbench: The robot learning benchmark & learning environment, 2019.

[16] Timothée Lesort, Vincenzo Lomonaco, Andrei Stoian, Davide Maltoni, David Filliat, and Natalia Díaz-Rodríguez. Continual learning for robotics: Definition, framework, learning strategies, opportunities and challenges. *Information fusion*, 58:52–68, 2020.

[17] Anthony Liang, Ishika Singh, Karl Pertsch, and Jesse Thomason. Transformer adapters for robot learning. In *CoRL 2022 Workshop on Pre-training Robot Learning*, 2022. URL https://openreview.net/forum?id=H--wvRYBmF.

[18] Zuxin Liu, Jesse Zhang, Kavosh Asadi, Yao Liu, Ding Zhao, Shoham Sabach, and Rasool Fakoor. Tail: Task-specific adapters for imitation learning with large pretrained models, 2024.

[19] Guilherme Maeda, Marco Ewerton, Takayuki Osa, Bap-

tiste Busch, and Jan Peters. Active incremental learning of robot movement primitives. In Sergey Levine, Vincent Vanhoucke, and Ken Goldberg, editors, *Proceedings of the 1st Annual Conference on Robot Learning*, volume 78 of *Proceedings of Machine Learning Research*, pages 37–46. PMLR, 13–15 Nov 2017. URL https://proceedings.mlr.press/v78/maeda17a.html.

[20] Allen Z. Ren, Anushri Dixit, Alexandra Bodrova, Sumeet Singh, Stephen Tu, Noah Brown, Peng Xu, Leila Takayama, Fei Xia, Jake Varley, Zhenjia Xu, Dorsa Sadigh, Andy Zeng, and Anirudha Majumdar. Robots that ask for help: Uncertainty alignment for large language model planners, 2023.

[21] Yi-Lin Sung, Jaemin Cho, and Mohit Bansal. Vl-adapter: Parameter-efficient transfer learning for vision-and-language tasks, 2022.

[22] Stefanie Tellex, Ross A. Knepper, Adrian Li, Daniela Rus, and Nicholas Roy. Asking for help using inverse semantics. In *Robotics: Science and Systems*, 2014. URL https://api.semanticscholar.org/CorpusID:3020962.

[23] Jesse Thomason. Jointly improving parsing and perception for natural language commands through human-robot dialog. 2020. URL https://api.semanticscholar.org/CorpusID:261975571.

[24] Sebastian Thrun and Tom M Mitchell. Lifelong robot learning. *Robotics and autonomous systems*, 15(1-2):25–46, 1995.

[25] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023.

[26] Weikang Wan, Yifeng Zhu, Rutav Shah, and Yuke Zhu. Lotus: Continual imitation learning for robot manipulation through unsupervised skill discovery, 2024.

[27] Mengda Xu, Zhenjia Xu, Cheng Chi, Manuela Veloso, and Shuran Song. Xskill: Cross embodiment skill discovery, 2023.

[28] Tony Z. Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware, 2023.

Appendix

A. Experiment Details

Model	Precision	Recall	F_1	Accuracy
Resnet + Transformer	88.0 ± 2.2	95.9 ± 2.0	91.8 ± 2.0	91.4 ± 2.11

TABLE II
EXPERIMENTAL RESULTS OF OUR ALIGNMENT MODEL ON ALIGNING HUMAN VIDEOS WITH ROBOT TRAJECTORIES ON SUBSET OF THE RH20T DATASET [10].

Detailed results of the alignment model on RH20T. We present the detailed results of the alignment model in Table II. We conduct five-split evaluation on the dataset, and report the mean score and standard deviation of each metric. Each model is trained on 80% of the trajectories and evaluated on the other 20%. In total, we use 1240 robot trajectories and 1193 human demonstrations across 98 tasks of the RH20T dataset configuration 5. As shown in Table II, our model achieves 91.8% on the F_1 metric, and 91.4% on the overall accuracy metric. This strong performance of the alignment model enables the robot agent to actively adapt learned skills to perform novel task, or to understand that it needs to learn a novel skill from seeing a single human demonstration.

Detailed results of the continual learning policy on each task of the RLBench. We present the per-task success rate of the policies in the RLBench simulator. Table III shows the performance of the three policies on each pre-trained task after fine-tuning, and Table IV demonstrates the performance of the policies on the tasks that they are finetuned on. All the three models are trained to predict joint positions for the same number of gradient steps. In the pre-train phase, each model is trained with 1000 robot demonstrations from each pre-train task for 1000 epochs. In the finetune phase, each model is trained with 5 robot demonstrations from each finetune task for 20000 epochs. Notice that due to the limitation of the visual-motor policies, we use a static location for all the finetune tasks during both training and evaluations. However, for all the pre-train tasks, we use randomized locations during both training and evaluation. As presented in the tables, TAIL achieves a near 0% success rate on majority of the tasks except for *close fridge*. This is because that *close fridge* is a relatively easier task in the environment, and the agent has a non-trivial chance to accidentally hit the fridge door and close it even if it is doing random behaviors. On the other hand, the baseline ACT model achieves a strong 100% success rate on the tasks that it is fine-tuned on, demonstrating its strong capability of learning fine-grained control. However, it also achieves a extremely low success rate on all the pre-train tasks after fine-tuning. This shows that ACT suffers from catastrophic forgetting and can no longer perform the pre-train tasks after fine-tuning. In comparison, ACT-LoRA achieves a 100% success rate on the fine-tune tasks, while still being able to perform on all the pre-train tasks with an overall success rate of 74.5%. This experiment result demonstrates that ACT-LoRA inherited the capability of fine-grained controls from ACT, and the ability to prevent catastrophic forgetting from the additional Low-Rank Adaptors, and hence is suitable for the use case where fine-grained control and continual learning are needed.

B. Implementation details of the dialogue state machine

We describe the details of the implementation of the dialogue state machine. Algorithm 1 is the pseudo code of the dialogue state machine. The robot agent first initializes the conversation with the human user, and repeatively asks questions until it obtains a clear list of instructions from this initial conversation. Then, the agent attempts to execute the list of actions sequentially until all the instructions are finished.

During execution of each task, if the agent finds that the task can be executed with one of the known skills, the agent directly executes the task with the corresponding policy. If the robot agent fails to directly find an executable skill for the task, it first searches for a usable skill in the semantic space. If it finds a skill that has a higher similarity score than the threshold in the semantic space, it proposes to the human user to use this skill to execute the task, and proceeds after obtaining the agreement from the human user. Otherwise, if the agent fails to find a usable skill, or the human user rejects the agent's proposed skill, the agent asks the human user for a human demonstration, and attempts to find a usable skill in the skill space based on the human demonstration. The skill search in the skill space is similar to that of the semantic space. If the agent finds a skill that has a higher similarity score than the threshold in the skill space, it proposes the skill to the human user. If the human user agrees with such skill proposal, the agent learns that a known skill can be adapted to the new task and executes the task. Otherwise, if the agent fails to find an aligned skill or its proposal is rejected by the human user, it realizes that it doesn't possess the skill to execute the task, and will ask for several robot demonstrations to train a completely new skill for the task.

The LLM serves as the interface between the robot agent and the human user. Whenever the robot agent is in a state that it needs inputs from the human user, it prompts the LLM with the current state of the agent and the information needed from the human user. The LLM then initiates a dialogue with the human user, and continues the dialogue until it retrieves the information needed by the robot agent. Such share autonomy between the state machine has more reliability than fully relying on the LLM, and can fully exploit the linguistic capability of the LLM.

C. Implementation details of the alignment model

We describe the details of the implementation of the alignment model. Following the notation in the main paper, we use E_{robot}, and E_{human} to denote the robot trajectory encoder and human demonstration encoder respectively. We also use ϵ_t and ϵ_v to denote the different thresholds for training and validation. To reduce the computational cost, we downsample all the human demonstrations and robot trajectories to 100 timesteps uniformly, and use image inputs from a single camera for both

Model	close box	open microwave	meat on grill	open door	push button	phone on base	toilet seat up	water plants
ACT-LoRA	**82.0**	**32.0**	**86.0**	**96.0**	**84.0**	**72.0**	**66.0**	**80.0**
ACT	0.0	0.0	8.0	0.0	0.0	0.0	0.0	4.0
TAIL	0.0	0.0	2.0	0.0	0.0	0.0	0.0	0.0

TABLE III
THE SUCCESS RATE OF ACT-LORA, ACT, AND TAIL ON EACH PRE-TRAINED TASK IN THE RLBENCH ENVIRONMENT.

Model	open box	close fridge	meat off grill	toilet seat down	take lid of sauce pan	close microwave
ACT-LoRA	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**
ACT	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**	**100.0**
TAIL	2.0	26.0	0.0	0.0	2.0	0.0

TABLE IV
THE SUCCESS RATE OF ACT-LORA, ACT, AND TAIL ON EACH FINE-TUNE TASK IN THE RLBENCH ENVIRONMENT, WHERE EACH MODEL IS ONLY FINETUNED ON 5 DEMONSTRATIONS FROM EACH TASK ON THE LIST.

the human demonstrations and robot trajectories. We use a 6-layer transformer encoder with 8 heads for both the human demonstration encoder and the robot trajectory encoder. Both encoders use a resnet-18 feature extractor to extract features from the raw image inputs. The robot trajectory encoder also takes in proprioceptive inputs from each time-step. During training, we minimize the cosine embedding loss between the human demonstration and robot trajectory with the training threshold ψ_t, denoted as following:

$$L(d,\tau,y) = \begin{cases} 1 - \cos(E_{\text{human}}(d), E_{\text{robot}}(\tau)) & \text{if } y = 1, \\ \max(0, \cos(E_{\text{human}}(d), E_{\text{robot}}(\tau)) - \epsilon_t) & \text{if } y = -1. \end{cases}$$

During inference, two trajectories are said to be the same skill if their cosine similarity is above the threshold ϵ_v. For the experiment of RH20T, we use $\epsilon_t = 0.5$ and $\epsilon_v = 0.7$, and train the alignment model for 10000 gradient step with a batch size of 16.

D. Implementation details for ACT-LoRA

We describe the details of our implementation of the ACT-LoRA policy. Following Zhao et al. [28], we train with a CVAE architecture and discard the additional encoder during inference. For both the CVAE encoder and the state encoder, we use a 4-layer transformer encoder with 8 heads. We extract features from raw image inputs from multiple cameras using resnet-18. These visual features are fed to the transformer encoder along with the proprioceptive inputs. For the decoder side, we use 6-layer transformer decoder with trainable embeddings. We also use a chunk size of 100 as it gives the best performance empirically [28]. The same configuration is also used for the baseline ACT model. As for the configuration of the low-rank adaptors, we follow TAIL [18] and use a rank size of 8. Each skill is associated with a set of unique adaptor weights.

E. Implementation details for TAIL

As there is no publicly available source code for TAIL [18], we tried our best attempt to re-implement TAIL for a fair comparison. To reduce the computation cost for the original TAIL model, we use a transformer encoder in replacement to the GPT-2 temporal decoder to speed up the training process. Furthermore, due to the limited time, the LoRA weights are only introduced to the transformer encoder, but not to any pre-trained feature extractors, including the CLIP text encoder and CLIP image encoder. Apart from these changes, we choose hyperparameters as close as possible to the original TAIL paper [18]. The TAIL model takes in linguistic task descriptions, image observations, and proprioceptive inputs over history timesteps. We first extract the feature of the raw image inputs and the linguistic task descriptions using the pretrained CLIP image and text encoder. Then, we use a FiLM layer to inject the linguistic features into the image features and the proprioceptive inputs. These inputs are treated as the input tokens of the transformer encoder. Then, we use an MLP layer to project the encoded token into parameters for a Gaussian Mixture Model(GMM). During training, the model is optimized by minimizing the negative log-likelihood loss of the ground truth actions. During inference, we sample from the distribution of the GMM predicted by the model.

Algorithm 1 The Algorithm for the Dialogue State Machine

```
Input:
        $\mathcal{O}_0$: The initial observation of the agent
        $\mathcal{S} = \{S_1, \dots, s_K\}$: The initial skill library of the agent
        $\pi_\psi, \psi = \{\psi_0, \psi_1, \dots, \psi_K\}$: Policy $\pi$ parameterized by $\psi$, composed of shared weights $\psi_0$ and skill specific weights
        $\{\psi_1, \dots, \psi_K\}$
        $\epsilon_{\text{text}}$: The threshold to determine whether the two skills are the same in the semantic space
        $\epsilon_{\text{skill}}$: The threshold to determine whether the two skills are the same in the skill space
 1: $\mathcal{A} \leftarrow$ GetListOfActionsFromDialogue()
 2: while $\mathcal{A}$ is not empty do
 3:     $a \leftarrow \mathcal{A}[0]$
 4:     if $a \in \mathcal{S}$ then
 5:         ExecuteTask($a$)
 6:     else
 7:         $S_i, s \leftarrow$ SearchSkillLibraryWithSemanticSimilarity($a$)
 8:         if $s \geq \epsilon_{\text{text}}$ then
 9:             response $\leftarrow$ ProposeSkillToHuman($S_i$)
10:             if response=agree then
11:                 ExecuteTask($S_i$)
12:                 Continue                                          ▷ skip line 13 to line 20
13:         $d$ ←AskForHumanDemonstration($a$)
14:         $S_j, s' \leftarrow$ SearchSkillLibraryWithSkillSimilarity($d$)
15:         if $s' \geq \epsilon_{\text{skill}}$ then ProposeSkillToHuman($S_j$)
16:             if response=agree then
17:                 ExecuteTask($S_j$)
18:                 Continue                                          ▷ skip line 19, 20
19:         $r$ ←AskForRobotDemonstration($a$)
20:         FinetunePolicyForNewSkill($\pi_\psi$,$r$)
21:     $\mathcal{A} \leftarrow \mathcal{A}[1:]$
```