Continual Robot Skill and Task Learning via Dialogue

Weiwei Gu[1], Suresh Kondepudi[1], Anmol Gupta[1], Lixiao Huang[1], Nakul Gopalan[1]

***Abstract*— 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 *continually 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 task learning with language, or passively learn novel skills or concepts. Instead, we have developed a robot agent that queries unknown skills from real human users, and continually learns these novel skills using only a few robot demonstrations provided by the users. To achieve this goal, we developed a novel continual learning policy Action Chunking Transformer [1] with Low Rank Adaptation (ACT-LoRA), and 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 policy consistently outperforms GMM-LoRA on continual learning by achieving 40% improvements in the RLBench dataset and 30% improvements in LIBERO dataset on fine-tuned skills. Additionally, we performed an IRB approved human-subjects study in a sandwich making domain to demonstrate that our framework is able to learn novel dynamic skills from non-expert human users and complete tasks using dialog interactions. Our framework achieved an overall 87.5% task completion rate of making novel sandwiches, and a 100% success rate on performing the novel skills learned from human users during the test phase of the study. This result illustrates the promise of a continual learning robot that saves time in the future for users once it has been taught tasks compared to non-learning agents.**

I. Introduction

Chai et al.[2] 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 [3], [4], [5], 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.

[1]School of Computation and AI, ASU, Tempe {weiweigu, nkondepu, anmolgupta, lixiao.huang ng}@asu.edu

Human-Robot interaction via language is a well studied problem [2], [4], [5], [6]. Robot agents have been able to interpret language instructions from the human users, and perform visual-motor policies to complete tasks [3], [4], [5]. 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 [7], [8], [6], [9]. Some recent works learn neural visuo-motor skills in a continual setting [9], [10], [11]. 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 novel framework that learns task abstractions and novel skills from dialog interactions from human users. Our agent learns the high-level plan by converting the dialog with the users into a sequence of skills. When encountering a novel task, our robot agent starts a conversation with the human user and requests the human user to provide several robot demonstrations for the novel skill. Previous methods in continual learning using human robot dialog have used Dynamic Movement Primitives with visual keypoints [12], where our method is completely end-to-end allowing our method to scale and function in a dynamic visual environment. Our closest comparison in continual learning is TAIL [11] which also user low-rank matrix (LoRA) in a continual learning setting, which heavily relies on a large scale of data for both pre-training and fine-tunining. To the best of our knowledge we present the first dialog aided continual dynamic end-to-end visuo-motor skill learning robot agent. Our contributions are as follows:

1) We develop a Continual Learning Aided by Dialog Agent (COLADA) that uses dialog and human demonstrations to keep improving over time by learning novel skill groundings and novel visuo-motor skills over interactions.
2) We developed a sample efficient Continual Learning algorithm for robots - ACT-LoRA as part of COLADA. We show that ACT-LoRA achieves the most robust performance in continual learning on RLBench and LIBERO dataset when compared against baslines.
3) Finally, we conduct a IRB approved human-subjects experiment to show that our system is able to learn to reason over and perform novel skills from non-expert human users using grounded dialog. Our agent achieved a overall success rate of 87.5% in task completion, and a success rate of 100% on the skills that are taught by the human users.

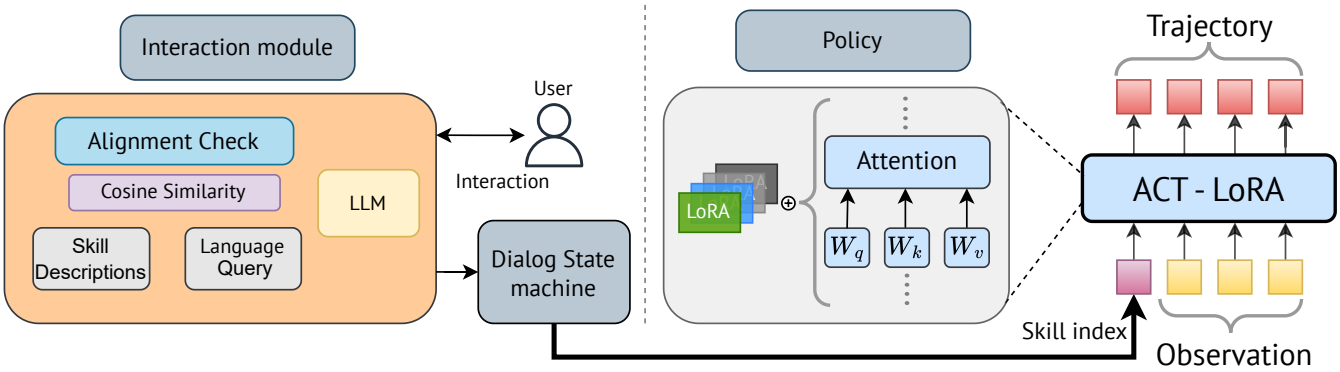


Fig. 1: Overview of our COLADA framework. The LLM serves as the interactive module and understands a user's feedback. The skill library provides representations for learned skills and novel demonstrations. The ACT-LORA policy executes the tasks based on the user's instructions. The agent searches for an executable skill by comparing the language representation of he queried skill and language representations of existing skills using a cosine similarity metric.

II. Problem Formulation

We formulate a task solving problem where both the robot and the human agent can take actions on their turns. 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, 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) learns high-level plans from dialog interactions with users; 2) queries the user for unknown skills; and 3) learns new skills with only a few instances. Our robot agent can learn a high level plan from the dialog interactions with the human users. We use a language model planner to map the high-level task θ specified by the users' utterances into a sequence of low-level motor skills τ. When needed to perform a motor skill τ, 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. 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. The robot agent can directly perform the skill τ whenever it finds a learned skill that aligns with τ in the semantic space. If τ is too far in the semantic space COLADA has to learn this skill. COLADA actively tries to learn this novel skill by requesting few robot demonstrations from the human users. To learn these novel visuo-motor skills from a few robot demonstrations, we developed a novel sample efficient continual skill learning approach ACT-LoRA for this task. Throughout the interaction with the users, our framework not only learns the novel visuo-motor skill, but also learns to ground language tokens to the skills. This enables our agent to perform the same skill when encountering the same language query at test time.

A. Interaction Module with a Large Language Model (LLM)

The dialog state s^d in our pipeline is maintained with an internal state machine, which is described in Algorithm 1 in Appendix F. The state machine uses an LLM to produce speech acts for the robot agent [13]. This state machine with the LLM has two major functionalities. Firstly, it tracks the dialog state to know what the user has explained previously or stated as a preference already as part of the dialog state s^d. Secondly, it interacts with the human user to ask for explanations and/or demonstrations based on the checks from our queryable skill library. If a skill τ is too far in the semantic space from any existing skill, COLADA has to learn this skill and the LLM produces the speech act to express this mismatch. For the semantic space we use is a CLIP text embedding [14]. The distance threshold for distances was hand-designed during the pilots and was chosen to be 0.95 in a unit normal space. The high threshold implies that COLADA has to be confidenct about a skill match before executing it. The interaction module is given the autonomy to continue the dialogue with the user until 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.

B. ACT-LoRA as Visual-motor Policy

Combining Low-Rank Adaptor with Action Chunking. Adapter-based methods [15], [16], [17], [18] have exhibited promising capabilities of light-weight and data-efficient fine-tuning of neural networks across various domains such as NLP [15], [17], and computer vision [16]. Liu et al. [11] extend Low-Rank Adaptor(LoRA) into robotics with TAIL, enabling a simulated robot to continually adapt to novel tasks without forgetting the old ones. Inspired by these methods, we take one step further and use the LoRA framework to learn to perform dynamic and contact-rich tasks such as cutting and butter applying for robots. On the other hand, Action Chunking Transformer(ACT) [1] 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.

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 $\{S_1, \ldots, S_K\}$. 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 $S \in \{S_1, \ldots, S_{n-1}\}$. 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 [1]. On top of the policy of the vanilla ACT model π_ϕ, the Low Rank Adaptor introduces a small set of additional low-rank parameters ϕ_i for each skill S_i. During the pre-training phase, the additional parameters $\phi_1, \ldots, \phi_K$ for skills $S_1, \ldots, S_K$ are jointly trained with the model's parameter ϕ. When we are finetuning with a skill $S_n, n > K$, 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.

IV. Experimental Results

In this section, we present the results of our policy on continual imitation learning in the simulated RLBench environment [19] and on three suites of the LIBERO environment. 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. Results for the human-subjects study can be found in Appendix A.

We chose two visuo-motor policies, ACT [1] and GMM-LoRA, as baselines to compare against our model on continual learning. ACT [1] is a SoTA visuo-motor policy that is able to perform dynamic tasks that require high precision, and GMM-LoRA resembles a scaled down version of TAIL [11], which is a SoTA continual learning policy. For our simulation study results in Table I and II, we present3 metrics, including **Pre-trained skills**, and **Fine-tuned skills(n trajectories)** and **Overall Success Rate(n trajectories)**. **Pre-trained skills** measures the policies' average success rate on the skills that policies are pre-trained on. **Fine-tuned skills(n trajectories)** and **Overall Success Rate(n trajectories)** measure the policies' average success rate on the new skills and the average success rate across both the pre-trained and fine-tuned skills respectively, where the policies are fine-tuned with n trajectories.

We first present our experiments on RLBench environment [19]. A total of 15 skills are chosen from the pre-defined skills of the environment. We then separate these skills into 5 different splits, and perform a five split validation on these skills. We report the statistics from the five-split validation in Table I, and more details for the experiment setup in Appendix E. Our model achieves a 59.4% of overall success rate after being fine-tuned with 1000 trajectories per fine-tuning skill, and a 64.13% overall success rate after being fine-tuned with 5 trajectories per fine-tuning skill. More specifically, ACT-LoRA achieves a success rate of 54% and 77.67% on the fine-tune skills after fine-tuning, while maintaining a 60.75% success rate on the pre-train skills. We observed GMM-LoRA to fail in skills which require precision, and ACT fail to remember older skills.

We also conduct simulation experiments on three suites of the LIBERO dataset [20], the spatial suite, the object suite, and the goal suite. Similarly, for each task suite we split the 10 skills into 5 different splits of skills and perform a five-split validation. We report the experiment results and statistics from the five-split validation in Table II. Further details can be found in Appendix F.

Although GMM-LoRA out-performs ACT-LoRA at pre-trained skills in the LIBERO-Object task suite by a small margin, ACT-LoRA consistently out-performs GMM-LoRA on fine-tune skills across all the three task suites, while maintaining a comparable performance in all the overall success rate metrics and the pre-trained skills success rate on the other two task suites. Furthermore, ACT-LoRA achieves better or comparable performance with the ACT model that goes through full fine-tuning on fine-tune skills, and consistently out-performs the ACT model in pre-trained skills and overall success rate metrics. These demonstrate that ACT-LoRA has the overall most stable performance across all the three models.

V. Related Work

Skill Discovery and Continual Learning. The area of visuo-motor continual learning is getting a lot of attention recently [9], [10], [11]. Wan et al. [9] discover new skills from segments of demonstrations by unsupervised incremental clustering. Xu et al. [10] learn the skill representation by aligning skills from different embodiments. Liu et al. [11] introduce task-specific *adapters* using low-rank adaptation techniques [15], preventing the agent from forgetting the learned skills when learning the new skills. These frameworks assume the presence of the demonstrations for the new tasks, and only discover skills in a passive fashion, while our agent can query for demonstrations to learn the new skills with language and does not rely on pre-existing skills.

Model	Pre-trained Skills	Fine-tune Skills(1000 traj.)	Overall Success Rate(1000 traj.)	Fine-tune Skills(5 traj.)	Overall Success Rate(5 traj.)
ACT-LoRA	$\mathbf{60.75 \pm 2.40}$	$54.00 \pm 9.73^*$	$\mathbf{59.40 \pm 1.52}$	77.67 ± 9.36	$\mathbf{64.13 \pm 1.80}$
GMM-LoRA	26.08 ± 4.02	13.33 ± 4.50	23.53 ± 2.99	16.67 ± 4.92	24.20 ± 3.72
ACT	9.25 ± 2.51	$62.00 \pm 8.84^*$	19.80 ± 1.69	$\mathbf{95.00 \pm 4.22}$	26.40 ± 2.45

TABLE I: Experimental results on RLBench dataset. $*$ indicates that two models have a similar best performance.

Model	Pre-trained Skills	Fine-tune Skills(50 traj.)	Overall Success Rate(50 traj.)	Fine-tune Skills(5 traj.)	Overall Success Rate(5 traj.)
			LIBERO-Spatial		
ACT-LoRA	$65.38 \pm 4.51^*$	40.50 ± 6.09	$\mathbf{60.40 \pm 4.20}$	35.50 ± 8.27	$59.40 \pm 4.40^*$
GMM-LoRA	$64.75 \pm 2.49^*$	9.00 ± 5.16	53.60 ± 1.70	6.00 ± 2.92	$53.0 \pm 2.21^*$
ACT	0.03 ± 0.02	$\mathbf{68.50 \pm 6.50}$	13.90 ± 1.31	$\mathbf{55.00 \pm 7.66}$	11.20 ± 1.43
			LIBERO-Object		
ACT-LoRA	67.00 ± 2.20	$68.00 \pm 8.57^*$	$67.20 \pm 1.50^*$	$48.00 \pm 10.23^*$	$63.20 \pm 1.60^*$
GMM-LoRA	$\mathbf{77.75 \pm 1.90}$	15.00 ± 5.65	$65.20 \pm 2.15^*$	14.00 ± 5.89	$65.00 \pm 1.08^*$
ACT	12.88 ± 2.78	$63.00 \pm 9.33^*$	22.90 ± 2.45	$35.50 \pm 7.92^*$	17.40 ± 3.45
			LIBERO-Goal		
ACT-LoRA	$73.63 \pm 2.96^*$	$\mathbf{49.00 \pm 8.54}$	$\mathbf{68.70 \pm 3.70}$	$23.00 \pm 8.57^*$	$63.50 \pm 4.00^*$
GMM-LoRA	$75.38 \pm 1.63^*$	10.50 ± 5.61	62.40 ± 1.39	3.5 ± 2.92	$61.00 \pm 1.72^*$
ACT	0.00 ± 0.00	19.50 ± 3.66	3.90 ± 0.73	$10.50 \pm 4.57^*$	2.10 ± 0.91

TABLE II: Experimental results on three suites of LIBERO dataset. An asterisk $*$ indicates that two models have a similar best performance.

Human-Robot Dialogue. Human-Robot dialog is a mature problem [21], [22], [23], [24]. Traditional methods use statistical algorithms with a pre-defined grammar, such as semantic parsing [23], [22], to connect the semantics of the dialogue to the environment's perceptual inputs. 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 [3]. Furthermore, Ren et al. [25] and Dai et al. [21] use LLMs to ask for human feedback for the robot agents demonstrating the importance of dialog. Recently, Grannen et al. [12] demonstrated a dialog based skill learning approach. However, these approaches either learn static visuo-motor pick and place tasks or learn dynamic skill with dynamic movement primitives (DMPs), while our framework learns a continual library of end-to-end neural visuo-motor skills from user data in a few shot setting, which allows better generalization to a dynamic scene.

Active Learning. Our work is related to active learning, where a learning agent actively improves its skills by asking a human for demonstrations [23], [26], [27], [28]. Defining an appropriate metric that triggers the request for assistance or information gathering becomes the key research problem in this domain. Thomason et al.[23] measure the semantic similarity between a newly introduced concept and the known concepts to ask for classifier labels. Chernova et al.[27], [28] 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.[26] use the uncertainty of Gaussian Processes(GPs) as the metric to trigger the request for assistance. Our method is similar to these approaches in that we use a cosine distance metric to measure similarity from the semantic information present in the language descriptions of skills without any strict labels.

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 need to conduct a wider user study with a larger number of skills and cooking tasks using our approach. 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 removed dynamic tasks such as chopping and cutting from our study because the robot's collision model would not allow it to continue even though the formalism is capable of learning these behaviors. We also want to compare such continual learning approaches with pre-trained policy approaches such as Robotics Transformer [4] to scale up our policy learning approach.

VII. Conclusion

In conclusion, we present a novel robot agents to learn task relevant knowledge and skills from dialogue interactions with human users. To the best of our knowledge this is the first work to demonstrate end-to-end dynamic visuo-motor skill learning while querying a user with dialog to express doubt. Our ACT-LoRA policy outperforms the existing continual learning baseline of GMM-LoRA in two separate simulated continual learning domains. Finally, we conducted a human-subject study, and demonstrated our framework is able to learn a completely new visual-motor skill from human and perform the tasks, with an overall task success rate of 87.5% and a success rate of 100% on the completely novel skill.

REFERENCES

[1] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn, "Learning fine-grained bimanual manipulation with low-cost hardware," 2023.

[2] J. Y. Chai, M. Cakmak, and C. L. Sidner, "Teaching robots new tasks through natural interaction," *Interactive Task Learning*, 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:160030141

[3] M. Ahn, A. Brohan, N. Brown, Y. Chebotar, O. Cortes, B. David, C. Finn, C. Fu, K. Gopalakrishnan, K. Hausman, A. Herzog, D. Ho, J. Hsu, J. Ibarz, B. Ichter, A. Irpan, E. Jang, R. J. Ruano, K. Jeffrey, S. Jesmonth, N. J. Joshi, R. Julian, D. Kalashnikov, Y. Kuang, K.-H. Lee, S. Levine, Y. Lu, L. Luu, C. Parada, P. Pastor, J. Quiambao, K. Rao, J. Rettinghouse, D. Reyes, P. Sermanet, N. Sievers, C. Tan, A. Toshev, V. Vanhoucke, F. Xia, T. Xiao, P. Xu, S. Xu, M. Yan, and A. Zeng, "Do as i can, not as i say: Grounding language in robotic affordances," 2022.

[4] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, J. Hsu, J. Ibarz, B. Ichter, A. Irpan, T. Jackson, S. Jesmonth, N. J. Joshi, R. Julian, D. Kalashnikov, Y. Kuang, I. Leal, K.-H. Lee, S. Levine, Y. Lu, U. Malla, D. Manjunath, I. Mordatch, O. Nachum, C. Parada, J. Peralta, E. Perez, K. Pertsch, J. Quiambao, K. Rao, M. Ryoo, G. Salazar, P. Sanketi, K. Sayed, J. Singh, S. Sontakke, A. Stone, C. Tan, H. Tran, V. Vanhoucke, S. Vega, Q. Vuong, F. Xia, T. Xiao, P. Xu, S. Xu, T. Yu, and B. Zitkovich, "Rt-1: Robotics transformer for real-world control at scale," 2023.

[5] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, X. Chen, K. Choromanski, T. Ding, D. Driess, A. Dubey, C. Finn, P. Florence, C. Fu, M. G. Arenas, K. Gopalakrishnan, K. Han, K. Hausman, A. Herzog, J. Hsu, B. Ichter, A. Irpan, N. Joshi, R. Julian, D. Kalashnikov, Y. Kuang, I. Leal, L. Lee, T.-W. E. Lee, S. Levine, Y. Lu, H. Michalewski, I. Mordatch, K. Pertsch, K. Rao, K. Reymann, M. Ryoo, G. Salazar, P. Sanketi, P. Sermanet, J. Singh, A. Singh, R. Soricut, H. Tran, V. Vanhoucke, Q. Vuong, A. Wahid, S. Welker, P. Wohlhart, J. Wu, F. Xia, T. Xiao, P. Xu, S. Xu, T. Yu, and B. Zitkovich, "Rt-2: Vision-language-action models transfer web knowledge to robotic control," 2023.

[6] W. Gu, A. Sah, and N. Gopalan, "Interactive visual task learning for robots," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 9, 2024, pp. 10 297–10 305.

[7] S. Thrun and T. M. Mitchell, "Lifelong robot learning," *Robotics and autonomous systems*, vol. 15, no. 1-2, pp. 25–46, 1995.

[8] T. Lesort, V. Lomonaco, A. Stoian, D. Maltoni, D. Filliat, and N. Díaz-Rodríguez, "Continual learning for robotics: Definition, framework, learning strategies, opportunities and challenges," *Information fusion*, vol. 58, pp. 52–68, 2020.

[9] W. Wan, Y. Zhu, R. Shah, and Y. Zhu, "Lotus: Continual imitation learning for robot manipulation through unsupervised skill discovery," 2024.

[10] M. Xu, Z. Xu, C. Chi, M. Veloso, and S. Song, "Xskill: Cross embodiment skill discovery," 2023.

[11] Z. Liu, J. Zhang, K. Asadi, Y. Liu, D. Zhao, S. Sabach, and R. Fakoor, "Tail: Task-specific adapters for imitation learning with large pre-trained models," 2024.

[12] J. Grannen, S. Karamcheti, S. Mirchandani, P. Liang, and D. Sadigh, "Vocal sandbox: Continual learning and adaptation for situated human-robot collaboration," 2024. [Online]. Available: https://arxiv.org/abs/2411.02599

[13] "ChatGPT," https://www.openai.com/chatgpt, 2024, accessed: May 30, 2024.

[14] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*. PMLR, 2021, pp. 8748–8763.

[15] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "Lora: Low-rank adaptation of large language models," 2021.

[16] Y.-L. Sung, J. Cho, and M. Bansal, "Vl-adapter: Parameter-efficient transfer learning for vision-and-language tasks," 2022.

[17] P. Gao, J. Han, R. Zhang, Z. Lin, S. Geng, A. Zhou, W. Zhang, P. Lu, C. He, X. Yue, H. Li, and Y. Qiao, "Llama-adapter v2: Parameter-efficient visual instruction model," 2023.

[18] A. Liang, I. Singh, K. Pertsch, and J. Thomason, "Transformer adapters for robot learning," in *CoRL 2022 Workshop on Pre-training Robot Learning*, 2022. [Online]. Available: https://openreview.net/forum?id=H--wvRYBmF

[19] S. James, Z. Ma, D. R. Arrojo, and A. J. Davison, "Rlbench: The robot learning benchmark & learning environment," 2019.

[20] B. Liu, Y. Zhu, C. Gao, Y. Feng, Q. Liu, Y. Zhu, and P. Stone, "Libero: Benchmarking knowledge transfer for lifelong robot learning," 2023. [Online]. Available: https://arxiv.org/abs/2306.03310

[21] Y. Dai, R. Peng, S. Li, and J. Chai, "Think, act, and ask: Open-world interactive personalized robot navigation," 2024.

[22] S. Tellex, R. A. Knepper, A. Li, D. Rus, and N. Roy, "Asking for help using inverse semantics," in *Robotics: Science and Systems*, 2014. [Online]. Available: https://api.semanticscholar.org/CorpusID:3020962

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

[24] J. Y. Chai, Q. Gao, L. She, S. Yang, S. Saba-Sadiya, and G. 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*. International Joint Conferences on Artificial Intelligence Organization, 7 2018, pp. 2–9. [Online]. Available: https://doi.org/10.24963/ijcai.2018/1

[25] A. Z. Ren, A. Dixit, A. Bodrova, S. Singh, S. Tu, N. Brown, P. Xu, L. Takayama, F. Xia, J. Varley, Z. Xu, D. Sadigh, A. Zeng, and A. Majumdar, "Robots that ask for help: Uncertainty alignment for large language model planners," 2023.

[26] G. Maeda, M. Ewerton, T. Osa, B. Busch, and J. Peters, "Active incremental learning of robot movement primitives," in *Proceedings of the 1st Annual Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, S. Levine, V. Vanhoucke, and K. Goldberg, Eds., vol. 78. PMLR, 13–15 Nov 2017, pp. 37–46. [Online]. Available: https://proceedings.mlr.press/v78/maeda17a.html

[27] S. Chernova and M. Veloso, "Interactive policy learning through confidence-based autonomy," *Journal of Artificial Intelligence Research*, vol. 34, p. 1–25, Jan. 2009. [Online]. Available: http://dx.doi.org/10.1613/jair.2584

[28] ——, "Confidence-based policy learning from demonstration using gaussian mixture models," 05 2007, p. 233.

[29] C. Bartneck, D. Kulić, E. A. Croft, and S. Zoghbi, "Measurement instruments for the anthropomorphism, animacy, likeability, perceived intelligence, and perceived safety of robots," *International Journal of Social Robotics*, vol. 1, pp. 71–81, 2009. [Online]. Available: https://api.semanticscholar.org/CorpusID:17967380

[30] J. Brooke, "Sus: A quick and dirty usability scale," *Usability Eval. Ind.*, vol. 189, 11 1995.

[31] S. G. Hart and L. E. Staveland, "Development of nasa-tlx (task load index): Results of empirical and theoretical research," *Human mental workload*, vol. 1, no. 3, pp. 139–183, 1988.

[32] H. Wang, K. Kedia, J. Ren, R. Abdullah, A. Bhardwaj, A. Chao, K. Y. Chen, N. Chin, P. Dan, X. Fan, G. Gonzalez-Pumariega, A. Kompella, M. A. Pace, Y. Sharma, X. Sun, N. Sunkara, and S. Choudhury, "Mosaic: A modular system for assistive and interactive cooking," 2024. [Online]. Available: https://arxiv.org/abs/2402.18796

APPENDIX

A. Human-Subjects Experiment

1) Robotics Domain for Sandwich Making: Our human subjects' experiment was on a customized sandwich making robot domain where the robot does not know all the skills required to make sandwiches for the users. More specifically, we design two different sandwich configurations, including a veggie sandwich and a lettuce sandwich with butter, where users need to teach the robot to pour pepper and apply butter respectively. The robotic setup includes a Franka FR3 Robot and three Realsense D435 cameras. We set up our cameras to provide a frontal view, a top-down view, and a wrist-mounted camera for a view from the robot's perspective. The workspace includes a table with items curated for the system. We designed 3D-printed tools tailored to support our task requirements as an attachment for the Franka Robot. These tools include a knife for cutting task and a spatula for spreading task. This configuration allows us to capture dense and diverse features for training our policy. Our data collection pipeline includes a 6D Spacemouse from 3DConnexion, which dictates the motion of the robot end effector. This facilitates the collection of dense data. Although limited by the data collection rate, this setup allows users to control the robot in the task space with relative ease because of the intuitive nature of the Spacemouse. Throughout the system's operation, picking and placing robot tools is done by pre-specified waypoints because grasping a tool is not our focus. Figure 2 demonstrates our sandwich making domain. We used the sandwich-making task for two reasons. Firstly, the sandwich-making task includes a lot of contact-rich and dynamic sub-tasks, such as applying butter and slicing cheese. Secondly, sandwich-making is a multi-step process, allowing the robot agent and the participants to have multiple rounds of conversations. Fake food was used as our ingredients for environmental reasons. These fake food includes play-doh, vegetable shortening, and other toy food made with plastics, such as bread, onions, cucumbers, and strawberries.

2) Baselines: : We compared against two baselines – An **inarticulate agent** that keeps solving a task even though some skills for a task might be unknown. This is similar to an agent that cannot reason about skills it knows vs skills it needs semantically. Secondly, an **inverse semantics agent** that knows which skills it does not know but asks for human help every time it reaches an unknown skill. This baseline is inspired from prior work where robots asked for human help when stuck [22].

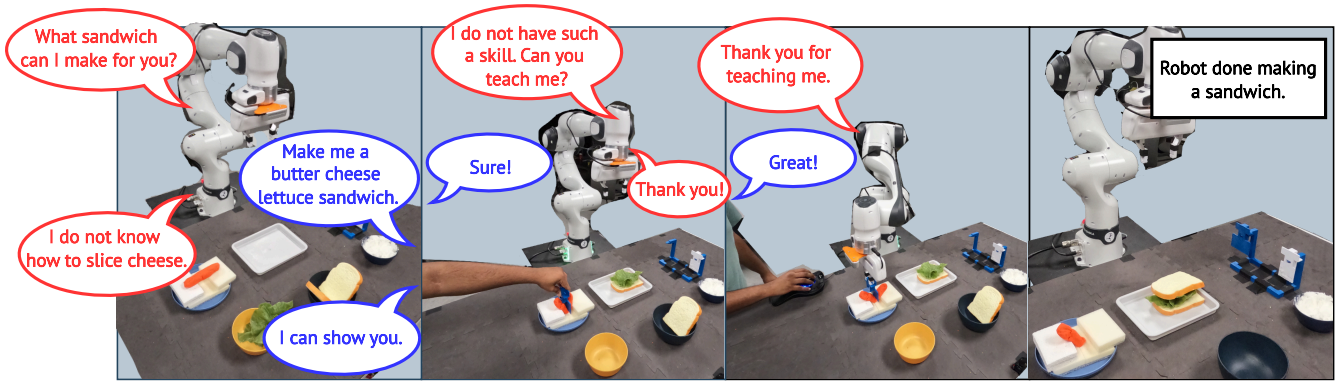
3) Study Design and Measures: Participants interacted with the robot in two phases. During the first phase - the interaction phase, participants interact with the robot and teach the robot novel skills and task knowledge as they interact with it, this includes dialogue, human demonstration, and robot demonstration. In the second phase - the evaluation phase, participants request the robot to perform the same tasks and evaluate the performance of the robot agent. We needed a two-phase study because we wanted to collect data for skill learning in the first phase and then run a learned policy on the agent in the second phase. In each phase the participants were expected to write emails as an auxiliary task as a realistic chore a user might have to do while the robot cooks. The participants came in for another session at least one day apart, allowing $5+$ hours of time to train novel skills using user demonstrations. This makes our study two separate 1×3 within-subjects experiment to measure our framework's ability to learn novel skills and task knowledge by interacting with non-expert human users. We do not compare subjective metrics for any tasks between the two phases as they were performed on different days and these subjective metrics might be different depending on the subjects' memory of the experience.

The objective metrics we used for the human-subjects experiment are as follows. We measured the overall success rate (SR) of completing the entire sandwich and the success rate for completing each independent sub-task. We measured time spent teaching the robot in each phase and time spent interacting with the robot to help solve the task in each phase. We measured words and emails written for the auxiliary task when working with each agent in both phases.

We make a distinction in the evaluation phase for skills that were taught by the participant vs pre-existing skills in our skill library. This demonstrates that we can add new skill without loss of performance to our existing skills. In the post-study survey, we administered the Godspeed Likability sub-scale [29], System Usability Scale (SUS)[30], and the NASA TLX [31].

4) Procedure: The procedure of the study is as follows. Participants first filled out the consent form and a pre-study survey. Then, we handed out a general introduction of the experiment and administered the two phases sequentially. Before each phase, a demonstration video and the instructions for the corresponding phase were provided to the participant. The anonymized instruction manual and videos are provided in the supplementary materials.

The interaction phase:- Here the participant requested the robot agent to make one of two sandwiches. During the process, COLADA asked the participant for task knowledge or robot demonstrations using dialog. The participant answered task knowledge-relevant questions directly with their language responses and provided robot demonstrations on request. We recorded all replies from the participants in audio and converted them to text using audio-to-text tools. As an auxiliary task the participants also wrote emails for various tasks on a computer while the robot made the sandwich. At the end of the interaction phase, the participant was asked to fill out a survey to evaluate the subjective experience with the robot. The three conditions that the participants observed here were: 1) *Inverse Semantics Agent* - the participant performed the skill that the robot does not know but the robot understands which skills are missing; 2) *Learning Agent* - The participant provides a demonstration to the robot so it can learn the skill for the evaluation phase; 3) *Inarticulate Agent*- The dialog state machine acts randomly when it encounters the mention of an unknown skill. This



(a) (b) (c) (d)

Fig. 2: 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.

agent is not guaranteed to complete a sandwich as it lacks the ability to know which skills it does not know and can take incorrect actions. The participants observe all these conditions in random order.

The evaluation phase:- The participant comes back after the next day or later and asks the robot agent to make the exact same sandwich as the one requested in the interaction phase. The participant again writes emails as an auxiliary task while the robot finishes the task. In certain conditions the participant might still need to help the robot in this phase to perform unknown skills. Finally they fill a survey to evaluate the robot's subjective performance. The three conditions observed in this phase are - 1) Inverse Semantics Agent - the participant performed the skill that the robot does not know; 2) Learning Agent with ACT-LoRA - the robot makes the whole sandwich based on the participants data from the interaction phase with our ACT-LoRA model; 3) Inarticulate Agent - where the robot still performs random actions when it reaches an unknown skill.

5) Research Questions: We investigate the following research questions (RQs) with our human-subjects study.

RQ1: *Can COLADA learn novel skills from the demonstrations provided by novice human users?* We investigate whether COLADA can learn a novel skill from 3 demonstrations provided by each participant, and then perform the learned skill correctly.

RQ2: *How does COLADA perform in completing the tasks requested by the users?* We investigate whether COLADA is able to complete the requested sandwich from the participants, and compare COLADA's performance against an inverse semantics agent that serves as an upper bound and also uses ACT-LoRA policy for pre-trained skills.

RQ3: *Is it actually efficient for users to teach our robot novel skills?* We investigate whether COLADA requires less interaction time from the participants after learning. Moreover, we hypothesize that participants have a higher ratio of time spent on the distraction task with COLADA than the inverse semantics agent after learning.

RQ4: *Do participants consider the Inarticulate agent worse than the other agents that ask intelligent questions?* We hypothesize that participants prefer COLADA and the inverse semantics agent over the Inarticulate agent on subjective metrics, including system usability,anthropomorphism, likeability, animacy, and perceived intelligence.

RQ5: *Do participants prefer COLADA over the inverse semantics agent?* We investigate whether participants consider COLADA better than the inverse semantics agent on subjective metrics including workload, system usability,anthropomorphism, likeability, animacy, and perceived intelligence.

B. Human Subject Study Results on a Franka FR3 Robot

We conducted an IRB approved study with 16 participants and 20 pilot subjects. We had 8 female subjects (50.00% of the user study). The age demographic of our participants is 23.44 ± 0.51. Our participants have 0.5 ± 0.32 years of experience in the field of robotics, and 4.69 ± 0.66 years of experience in computer science. The subjects spent 120 minutes in the interaction phase and then another 75 minutes for the evaluation phase. They were compensated with a $35 Amazon gift card. We present our objective success rate results in Tables III, IV, and objective metrics on the distraction tasks in Table V. For each metric, we perform normality test using Shapiro-Wilk test. If the data from such metric passes the normality test($p > 0.05$), we apply a parametric statistical test. Otherwise, we report the results of a non-parametric statistical test. Further details will be discussed later.

RQ1: *Can COLADA learn novel skills from the demonstrations provided by novice human users?* We find that COLADA is able to learn novel visuo-motor

skills from novice human users with just 3 tele-operation demonstrations. As shown in Table IV, our ACT-LoRA policy learns the novel skills with the success rate of (100%) and performs existing skills with the success rate of 88.89%.

RQ2: *How does COLADA perform in completing the tasks requested by the users?* Results from Table III and IV suggest that COLADA can complete the requested sandwich from the participants in both phases of the study. Our COLADA achieves 93.75% and 81.25% sandwich completion rate in the two phases of the study respectively. This is comparable to the performance of the inverse semantics agent, which relies on help from the users and cannot finish the task independently. This does demonstrate that there are challenging problems with making sandwiches such as picking objects that are not rigid and dynamic skills like applying butter. We noticed the most failures in the skills of picking up the bread on top as the bread can slip off from the bowl and can be unreachable for the robot after falling.

RQ3: *Is it actually efficient for users to teach our robot novel skills?* We expect subjects to spend more time teaching COLADA in the first phase and then using almost no time helping the agent in the second phase. As the time to interact with agents and write emails can vary drastically we measure the ratio of the time that participants spend on the distraction email writing task to the total time that they spend with the agent in Table V. According to a paired t-test, we find that COLADA allows participants to spend more time on finishing the distraction email writing in phase two than in phase one with significance($p < 0.001, t = 38.69$). Wilcoxon Signed-Rank test suggests that such trend also holds for the inarticulate agent($p = 0.006, Z = 3.21$), but not for the inverse semantics agent($p = 0.052, Z = 1.623$). This is because while all the three agents no longer require the human users to teach the sandwich through dialog in the test phase hence allowing the users to spend more time on the distraction task, the inverse semantics agent still requires the human users to pause on the distraction task and perform the unknown skill for the agent. More importantly, we also find that COLADA allows participants to spend a higher ratio of time writing emails than both the inverse semantics agent ($p < 0.001, Z = 3.61$) and the inarticulate agent($p < 0.001, Z = 4.17$) in phase two with significance, as indicated by Wilcoxon Signed-Rank test. We have also measured other objective metrics for the distraction task, including word count and number of emails completed. However, no significance was found in those metrics due to the high variance introduced by the participants, such as typing speed and the actual time each participant that actually spend on doing the agent interactions. These results suggest that the ability of learning enables the agent to complete the tasks autonomously, and eventually improve the time efficiency for users.

RQ4: *Do participants consider the inarticulate agent worse than the other agents that ask intelligent questions?* Our results from Table VII suggest that our participants consider that both COLADA and the inverse semantics agent better than the inarticulate agent in SUS, the Likeability, Animacy, Perceived Intelligence, and Anthropomorphism sub-scales from the Godspeed Questionnaire Series, and our customized comparative survey with significance. We discuss the details of the statistics test on these subjective metrics later. This indicates that even knowing that a skill is unknown is sufficient to demonstrate intelligence and be more useful to a human user.

RQ5: *Do participants prefer COLADA over the inverse semantics agent?* We investigated subjective metrics for COLADA and the inverse semantics agent from the post-surveys for both phases. In the post-surveys, we administer NASA Task Load Index(NASA TLX) [31], the System Usability Scale(SUS) [30], the Perceived Intelligence, Likeability, Animacy, and Anthropomorphism sub-scales from the Godspeed Questionnaire Series [29], and a customized comparative survey that rank the performance of the three agents. We find that participants rate COLADA to be more usable with SUS compared to the inverse semantics agent($p = 0.044, t = 1.83$). This is because the COLADA agent is able to complete the sandwich making task autonomously without disturbing the human users, whereas the inverse semantics agent still requires the human's help on the unknown skill. For other subjective metrics, we failed to find any significance between COLADA and the inverse semantics agent. We hypothesize that the long duration of the study and the distraction email writing task can be the reasons of the noise. For example, the workload from the email writing tasks can be a confounding factor for the workload relevant questions. Furthermore, the inverse semantics agent still asks intelligent questions to the participants in phase two, which can make the participants consider the inverse semantics agent more likeable or more intelligent.

Overall, we notice users being able to teach visuo-motor tasks few-shot to the robot and having a higher percentage of time to their auxiliary tasks such as writing emails using COLADA. We provide a demonstration video as a supplement to show how COLADA learns tasks and skills from its first interaction with a user, and performs the tasks fully autonomously in the second interaction.

C. Detailed results of the human-subjects study

We describe the details of the human-subjects study. Our human-subjects study is approved by the Institutional Review Board(IRB) of the university. We tested the study with 20 pilots before conducting the experiments on the participants. We fixed the issues of unclear instructions, short execution times for the learned skills and ambiguous phrases when the LLM was asking questions. We had to fine-tune the prompts of the LLMs a lot so the robot asked questions pertinant to the task of sandwich making. We also adjusted the configurations for the sandwiches, because some tasks can be very difficult for the novice users to teach the robot, such as picking up a deformable object. Additionally, we made the

Agent	Sandwich SR	Pre-trained Skill SR
COLADA	93.75%(15/16)	97.92%(47/48)
Inverse Semantics	81.25%(13/16)	93.75%(45/48)
Inarticulate	0.00%(0/16)	93.75%(15/16)

TABLE III: The phase one objective evaluations of the three agents on the human subject study, including the success rate of the entire sandwich, and the success rate of the robot performing on the skill that it was trained on.

Agent	Sandwich SR	Few-shot SR	Pre-train SR
COLADA	81.25%(13/16)	100.00%(16/16)	91.67%(44/48)
Inverse Semantics	87.50%(14/16)	N/A	91.67%(44/48)
Inarticulate	0.00%(0/16)	0.00%(0/16)	87.50%(14/16)

TABLE IV: The phase two objective evaluations of the three agents on the human subject study, including the success rate of the entire sandwich, and the success rate on the few-shot skill, and the success rate of the pre-train skill. The success rate of few-shot skill for the inverse semantics agent is not possible, because the inverse semantics agent always asks for help to skip the few-shot skill.

Agent	Interface Time Ratio(Phase one)	Interface Time Ratio(Phase two)
COLADA	47.78 ± 1.19	$\mathbf{95.41 \pm 0.38}$
Inverse Semantics	80.27 ± 2.06	85.13 ± 2.46
Inarticulate	80.88 ± 2.27	86.82 ± 1.46

TABLE V: The ratio of the interface time for participants. This metric measures how many percent of the time the users spend on the distraction task of writing emails.

interface of the distraction email writing task more intuitive for the participants, and created an instructional video for the email writing interface. All the instructional materials we used for the study can be found in the supplemental materials.

For the actual study a total of 16 participants were recruited through campus advertisements. The study is composed of two separate phases, the interaction phase that takes 120 minutes and the evaluation phase that takes 60 minutes, with a voluntary participation. The participants, including the pilots, are compensated with $35 Amazon gift card for their time. We designed the two-phase study for two major reasons. Firstly, our COLADA agent requires five hours to train for the novel skill. Secondly, we want to demonstrate a thorough comparison for the workload and objective metrics on the distraction task between our COLADA agent and the inverse semantics agent in the two phases. COLADA requires the users to remotely control the robot arm to perform the task in the interaction phase, and is fully automated in the evaluation phase, whereas the inverse semantics agent behaves the same in both phases by requesting the users to directly perform the task that it does not know.

We hypothesize that the users experience higher workload for COLADA than the inverse agent in the interaction phase, and a lower workload for the COLADA than the inverse semantics in the evaluation phase because we consider that for remotely controlling the robot arm to complete the task requires higher workload than directly completing the task themselves for the users, and the fully automated robot agent requests the least workload. We reject our hypothesis and accept the null hypothesis of – there is no difference in the users's perception of workload between COLADA and the inverse semantics agent. We consider that the workload from the distraction email writing task can be the major confounding factor to the workload metrics. From the users' perspective, even though COLADA saves their time by finishing the sandwich autonomously, they still need to work longer on the distraction tasks as the robot takes longer time to finish the same task than taking the users' help. As a result, the users might not perceive that the fully automated COLADA agent invokes less workload than the inverse semantics agent, and we did not find any significance in the subjective workload metric. However, our objective metrics that measure the ratio of time that users spend on the distraction tasks indicate that our COLADA agent allows user to use more of their time on the distraction time than the inverse semantics agent in phase two($p < 0.001, Z = 3.61$). This shows that a fully automated learning agent is more efficient for the users. Additionally, we observed that COLADA achieves a higher ratings than the inverse semantics agent with significance($p = 0.04, t = 1.83$) in the System Usability Scales(SUS). This demonstrates that a learning system is considered more useful than a system that relies on humans' help by the users.

1) Detailed procedure: We describe the detailed procedure for the study as follows.

Interaction Phase. Participants first filled out the consent form and a pre-study survey. Then, we handed out a general introduction of the experiment. The participants were then asked to read the instructions for the interaction phase, and watch a demonstration video. The demonstration video introduces how the robot agent requests for different types of help differently, and how to answer different requests from the robot agent. We use a completely different domain(Placing a block in the box) as example in the demonstration video. The instruction introduces domain relevant information, such as the configuration of the robot's workspace, the sandwich to make, and the steps to make the sandwich. The participants then watch another demonstration video that introduces how to use the email writing interface. The anonymized instructions and videos can be found in the supplementary material, and Fig. 3 shows our email writing interface. Then, the participants interacted with the three agents, the inarticulate agent, the inverse semantics agent, and the COLADA agent, in a random order. The inarticulate agent never interacts with the users except for getting the initial instruction set from the user. The inverse semantics agent always asks the human users for help when it encounters any task that it is uncertain with. The COLADA agent interacts with the human users by asking task-relevant question, asking for human help, and asking for robot demonstrations. The users then work on the distraction email writing tasks while these robot agents make the sandwich, and provide the required help from the agent when needed. After interacting with each system, the

Agent	Interruption Count	Normalized Completed Email Count	Normalized Word Count	Total Time	Task Time
			Phase One		
COLADA	2.13 ± 0.13	0.27 ± 0.03	0.24 ± 0.01	2176.67 ± 57.06	1035.21 ± 26.10
Inverse Semantics	1.13 ± 0.09	0.16 ± 0.02	0.20 ± 0.01	943.93 ± 32.41	753.21 ± 25.85
Inarticulate	0.00 ± 0.00	0.07 ± 0.02	0.08 ± 0.01	493.01 ± 58.62	412.98 ± 56.69
			Phase Two		
COLADA	0.00 ± 0.00	0.25 ± 0.03	0.23 ± 0.02	1083.42 ± 27.28	1033.70 ± 26.32
Inverse Semantics	1.00 ± 0.00	0.17 ± 0.02	0.17 ± 0.01	870.77 ± 26.26	738.27 ± 24.02
Inarticulate	0.00 ± 0.00	0.08 ± 0.01	0.07 ± 0.01	426.94 ± 51.85	376.78 ± 48.74

TABLE VI: The objective metrics of the human users on the distraction tasks of the study. The interruption count measures how many times each agent interrupt the users during the entire evaluation phase. The normalized email completion count measures the number of emails completed by the users while the agent is performing the task, normalized by the total number of emails completed by each user. The normalized word count measures the total number of words the users input when the agent is executing the tasks, normalized by the total number of words of each user for all agents. Total time measure the total amount of execution time in seconds of each agent, including the time that the agent interacts with the users and the time that the agent perform skills autonomously. Task time measures the amount of time in seconds for users to complete the distraction task, which is also the time that the agent performs skills autonomously.

Metrics	SUS(↑)	Anthropomorphism(↑)	Likability(↑)	Animacy(↑)	Perceived Intelligence(↑)	Comparative(↑)
			Phase One			
COLADA	8.06 ± 1.61	14.75 ± 0.89	20.38 ± 0.77	19.06 ± 0.85	36.13 ± 0.92	N/A
Inverse Semantics	11.13 ± 1.38	16.38 ± 0.98	20.13 ± 0.69	21.31 ± 0.98	37.31 ± 0.89	N/A
Inarticulate	4.06 ± 2.64	12.25 ± 1.05	17.13 ± 1.17	16.00 ± 1.34	29.31 ± 1.72	N/A
			Phase Two			
COLADA	12.50 ± 2.49	15.94 ± 1.04	20.19 ± 1.19	20.63 ± 1.32	35.69 ± 1.67	0.44 ± 0.87
Inverse Semantics	9.31 ± 2.55	15.31 ± 1.20	19.94 ± 1.07	20.75 ± 1.16	36.00 ± 1.47	-0.63 ± 0.68
Inarticulate	1.19 ± 2.62	12.00 ± 0.94	17.25 ± 1.27	15.50 ± 1.34	29.25 ± 1.95	-5.81 ± 0.86

TABLE VII: The subjective metrics for the interaction phase. We use the same ACT-LoRA policy as the policy for all the three agents.

participants were asked to fill-out a post-survey, including questions from NASA-TLX [31], SUS [30], and 4 sub-scales from the GodSpeed Questionnaire Series [29](Likability, Animacy, Natural, Perceived Intelligence). After the participants finished the interaction phase, we fine-tuned the ACT-LoRA policy the robot demonstrations collected from the users for COLADA.

Evaluation Phase. Participants came back to the lab. We handed the same instructions to the participants for them to ask the robot to make the same sandwich. The participants interacted with the same three robot agents, the inarticulate agent,the inverse semantics agent, and the COLADA agent. All the three agents remember the instructions to make the sandwich provided by the participants from the interaction phase. The inverse semantics agent and the inarticulate did not learn from the robot demonstrations from the interaction phase. This means that the inverse semantics agent still asked for help from the users for the same skill, and the inarticulate agent still failed to perform the same skill. The COLADA learned the novel skill from the demonstration in the interaction phase, and did not interact with the human users except for the initial interactions. After watching each agent, the participants were asked to fill out the same post-survey for the system. After watching all the three systems, the participants were asked to rank the three systems on 7 different description(helpful, useful, efficient, competent, uncooperative, inefficient, incompetent).

2) Detailed study results: The objective results on the task completion and skill success rates are presented in Table III, IV, and the objective results on distraction tasks are presented in Table V,VI. We also present results on subjective metrics for both phases in Table VII.

Based on our analysis, we found that COLADA is more efficient in time for our participants in phase two than in phase one. Additionally, COLADA is more time efficient for the user than the inverse semantics agent in phase two. For subjective metrics, no significance was found for the workload metrics between any agent pair. Both agents that can ask intelligent questions(COLADA and the inverse semantics agent) are considered better than the inarticulate agent in the sub-scales of system usability, anthropomorphism, likeability, animacy, perceived intelligence, and the comparative survey. Additionally, COLADA is considered better than the inverse semantics agent in the system usability sub-scale. We perform a normality test with Shapiro-Wilk test for each metric. If the data from such metric passes the normality test($p > 0.05$), we apply a parametric statistical test. Otherwise, we report the results of a non-parametric statistical test. The detailed results are described as follows.

Users' ratio of time on distraction task. Results from Shapiro-Wilk test suggest that conditions for normality were met for the data points to run a parametric statistical test($p = 0.18, W = 0.92$). Hence, we compare the time ratio metric between phase one and phase two for COLADA using paired t-test. Results from paired t-test suggest that COLADA allows users to spend more of their time on the email writing distraction task in phase two than in phase one($p < 0.001, t = 38.69$).

Results from Shapiro-Wilk test suggest that conditions for normality were not met for the data points to run a parametric statistical test ($p = 0.005, W = 0.82$). Hence, we compare the time ratio metric between COLADA and the inverse semantics agent using Wilcoxon Signed-Rank test. Results from Wilcoxon Signed-Rank test suggest that user can spend more time on the email writing task working with COLADA than the inverse semantics agent in phase two($p < 0.001, Z = 4.17$).

SUS. Results from Shapiro-Wilk test suggest that conditions for normality were met for the data points to run a parametric statistical test($p = 0.49, W = 0.96$). Hence, we conduct a paired t-test to compare the system usability metric of COLADA with the inarticulate agent. Results from paired t-test suggest that COLADA is considered better than the inarticulate agent in the system usability sub-scale in phase two($p = 0.002, t = 1.83$).

Results from Shapiro-Wilk test suggest that conditions for normality were met for the data points to run a parametric statistical test($p = 0.07, W = 0.90$). Hence, we conduct a paired t-test to compare the system usability metric of the inverse semantics agent with the inarticulate agent. Results from paired t-test suggest that the inverse semantics agent is considered better than the inarticulate agent in the system usability sub-scale in phase two($p = 0.006, t = 2.82$).

Results from Shapiro-Wilk test suggest that conditions for normality were met for the data points to run a parametric statistical test($p = 0.49, W = 0.95$). Hence, we conduct a paired t-test to compare the system usability metric of COLADA with the inverse semantics agent. Results from paired t-test suggest that COLADA is considered better than the inverse semantics agent in the system usability sub-scale in phase two($p = 0.04, t = 1.83$).

Anthropomorphism. Results from Shapiro-Wilk test suggest that conditions for normality were met for the data points to run a parametric statistical test($p = 0.34, W = 0.94$). Hence, we conduct a paired t-test to compare the anthropomorphism metric of COLADA with the inarticulate agent. Results from paired t-test suggest that COLADA is considered better than the inarticulate agent in the anthropomorphism metric in phase two($p = 0.003, t = 3.18$).

Results from Shapiro-Wilk test suggest that conditions for normality were met for the data points to run a parametric statistical test($p = 0.78, W = 0.97$). Hence, we conduct a paired t-test to compare the anthropomorphism metric of the inverse semantics agent with the inarticulate agent. Results from paired t-test suggest that the inverse semantics agent is considered better than the inarticulate agent in the anthropomorphism metric in phase two($p = 0.01, t = 2.54$).

Likability. Results from Shapiro-Wilk test suggest that conditions for normality were met for the data points to run a parametric statistical test($p = 0.57, W = 0.95$). Hence, we conduct a paired t-test to compare the likeability metric of COLADA with the inarticulate agent. Results from paired t-test suggest that COLADA is considered better than the inarticulate agent in the likeability metric in phase two(($p = 0.04, t = 1.86$).

Results from Shapiro-Wilk test suggest that conditions for normality were met for the data points to run a parametric statistical test($p = 0.59, W = 0.96$). Hence, we conduct a paired t-test to compare the likeability metric of the inverse semantics agent with the inarticulate agent. Results from paired t-test suggest that the inverse semantics agent is considered better than the inarticulate agent in the likeability metric in phase two($p = 0.03, t = 2.00$).

Animacy. Results from Shapiro-Wilk test suggest that our data in the animacy metric does not satisfy the condition for a parametric test($p = 0.03, W = 0.87$). Hence, we conduct a Wilcoxon Signed-Rank test to compare the animacy metric of COLADA with the inarticulate agent. Results from the Wilcoxon Signed-Rank test suggest that COLADA is considered better than inarticulate agent by users in the animacy metric with significance in phase two($p < 0.001, t = 3.10$).

Results from Shapiro-Wilk test suggest that our data in the animacy metric satisfies the condition for a parametric test($p = 0.07, W = 0.90$). Results from paired t-test suggest that the inverse semantics agent is considered better than inarticulate agent by users in the animacy metric with significance in phase two($p < 0.001, t = 3.87$).

Perceived Intelligence. Results from Shapiro-Wilk test suggest that conditions for normality were met for the data points to run a parametric statistical test($p = 0.07, W = 0.90$). Hence, we conduct a paired t-test to compare the perceived intelligence metric of COLADA with the inarticulate agent. Results from paired t-test suggest that COLADA is considered better than the inarticulate agent in the perceived intelligence metric in phase two($p = 0.01, t = 2.58$).

Results from Shapiro-Wilk test suggest that conditions for normality were met for the data points to run a parametric statistical test($p = 0.12, W = 0.91$). Hence, we conduct a paired t-test to compare the perceived intelligence metric of the inverse semantics agent with the inarticulate agent. Results from paired t-test suggest that the inverse semantics agent is considered better than the inarticulate agent in the perceived intelligence metric in phase two($p = 0.003, t = 3.09$).

Comparative. Conditions for normality were not met for the data points to run a parametric statistical test($p = 0.028, W = 0.871$). Hence, we conducted a Wilcoxon Signed-Rank test to compare COLADA with the inarticulate agent in the comparative metric. Results from Wilcoxon Signed-Rank test suggest that COLADA is preferred by user in the direct comparison with the inarticulate agent with significance($p = 0.004, Z = 2.61$).

Conditions for normality were met for the data points to

run a parametric statistical test($p = 0.24, W = 0.93$). Hence, we applied a paired t-test to compare the inverse semantics agent against the inarticulate agent in the comparative metric. Results from Wilcoxon Signed-Rank test suggest that the inverse semantics agent is preferred by user in the direct comparison with the inarticulate agent with significance($p = 0.001, Z = 3.02$).

3) Limitation of the study: There are two major limitations on the human-subjects study. Firstly, we need to increase the scale of the study to better understand the robustness of COLADA and ACT-LoRA. Currently, limited by the scale of data, we only conducted the study with two different sandwich configurations on 8 different tasks. A scaled-up version of the study with more tasks, more data, and more users will be necessary to test the robustness of our framework. Secondly, the demographic of the study is limited to university students. More subjects with wider demographic distribution will be needed to show that COLADA can work with the general population.

D. Discussion

In our analysis, we have demonstrated that COLADA can continually learn tasks and skills from dialog interactions with novice human users with our end-to-end neural network policy ACT-LoRA, and complete the requested tasks automatically with a task completion rate that is comparable to an upper bound of a human aided inverse semantics agent. This showcases that ACT-LoRA model is robust as a continual learning policy under low data regimes, whereas other SoTA continual learning policies such as TAIL [11] rely on fine-tuning large vision backbone on environment data and also require large scale of data for training, which is not accessible most of the time in real-world robot applications.

Additionally, we have also demonstrated that our end-to-end continual learning method can learn contact rich dynamic skills from novice users, such as applying butter and pouring pepper. We believe such methods have more potential to scale compared to existing DMP and keypoint based approaches [12]. We also want to state that even an agent that can just communicate its inadequacies such as our inverse semantics agent or other baselines like MOSAIC [32] has high satisfaction rates among users. This indicates the need for agents can indicate their confusions or lack of confidence in a task urgently. An observation we have made in our pilots and studies is that simple cold meals have a lot of contact based dynamic tasks that are challenging for users to demonstrate even if the platform can learn them. For example, it is challenging to present a tele-operation demo for the task of picking up bread from a flat plate. We believe more fundamental robotics research and user interface design is needed for such tasks. We first describe the details of our simulation experiments. We compared our approach to two baselines, ACT and GMM-LoRA because ACT represents the base architecture of ACT-LoRA, and GMM-LoRA is a continual learning policy baseline that resembles TAIL [11]. GMM-LoRA is a scaled-down version of TAIL [11], as the baseline in this work, as TAIL is a GMM-based policy augmented with LoRA weights and a larger vision backbone. TAIL itself failed to train in our experiments because we did not have enough visual data to pre-train TAIL's policy when we have only $10 - 20$ tasks. TAIL is promising when the set of pre-trained tasks is large enough to train its GPT-2 and CLIP layers which is challenging with robot data. The full TAIL model was not learning skills in our simulation domains within RLBench. We then confirmed the correctness of our implementation with the original authors of TAIL to confirm data scale issues. We then scaled the model down with a smaller visual backbone which we call GMM-LoRA that we can train with few samples to perform an equivalent comparison [20]. In all of the simulation domains, all of the three models go through a pre-train, fine-tune training schema. During fine-tuning, we only train the weights from the Low Rank Adaptor for ACT-LoRA and GMM-LoRA, while ACT is trained with all the weights.

E. Detailed Results on RLBench

For the RLBench simulation environment, all the three models are pre-trained with 1000 demonstrations for each of the pre-train skills for 100 epochs. To study the sample efficiency of these models, we experiment with fine-tuning with 5, 100, and 1000 trajectories. Each models is evaluated for 20 rollouts on each of the 15 skills to measure the success rate. We present the complete experimental results of the three policies in the RLBench simulator. We perform five-fold validation on 15 selected tasks from the RLBench simulator, and present the results in Table VIII. Detailed performance of each skill is presented in Table IX. All the three models are trained to predict joint positions in RLBench, and went through the same pre-trained, fine-tuned training schema. During the pre-train phase, each model is trained with 1000 robot demonstrations from each pre-train task for 5 epochs. In the fine-tuning phase, we only train the weights introduced by the Low-Rank Adaptor for ACT-LoRA and GMM-LoRA, while the ACT model is fine-tuned with all its weights. We fine-tuned models for $10, 100$ and 1000 epochs when using $1000, 100$ and 5 trajectories for fine-tune skills respectively. Notice that due to the limitation of the visual-motor policies, we use a static location to evaluate the fine-tune tasks when we fine-tune with 5 robot trajectories for all models. For the pre-trained skills and fine-tuned skills trained with more trajectories, we use a randomized initial configuration in evaluation.

As shown in Table VIII and Table IX, the full fine-tuned ACT model achieves a strong performance on fine-tuned skills, demonstrating its strong capability of learning fine-grained control. However, it suffers a near zero success rate for most of the pre-trained skills after fine-tuning. This shows that ACT suffers from catastrophic forgetting and can no longer perform the pre-train tasks after fine-tuning. On the contrary, our ACT-LoRA model not only achieves a comparable performance on fine-tuned skills as the ACT model, but also outperforms other baselines in pre-trained skills and overall success rate. This demonstrates that our ACT-LoRA model can continually learn novel skills without

Model	Pre-trained Skills	Fine-tune Skills(1000 traj.)	Overall Success Rate(1000 traj.)	Fine-tune Skills(100 traj.)	Overall Success Rate(100 traj.)	Fine-tune Skills(5 traj.)	Overall Success Rate(5 traj.)
ACT-LoRA	$\mathbf{60.75 \pm 2.40}$	$54.00 \pm 9.73^*$	$\mathbf{59.40 \pm 1.52}$	$47.67 \pm 10.24^*$	$\mathbf{58.87 \pm 1.55}$	77.67 ± 9.36	$\mathbf{64.13 \pm 1.80}$
GMM-LoRA	26.08 ± 4.02	13.33 ± 4.50	23.53 ± 2.99	11.00 ± 4.07	23.73 ± 3.15	16.67 ± 4.92	24.20 ± 3.72
ACT	9.25 ± 2.51	$62.00 \pm 8.84^*$	19.80 ± 1.69	$63.33 \pm 9.90^*$	20.60 ± 1.11	$\mathbf{95.00 \pm 4.22}$	26.40 ± 2.45

TABLE VIII: Complete experimental results on RLBench dataset. $*$ indicates that two models have a similar best performance. **Pre-trained skills** measures the policies' average success rate on the 12 skills that policies are pre-trained on. **Fine-tuned skills(1000 trajectories)** and **Fine-tuned skills(100 trajectories)** measure the policies' average success rate on the 3 new skills, where the policies are fine-tuned using 1000 trajectories and 100 trajectories for each fine-tuned skill. **Overall Success Rate(1000 trajectories)** and **Overall Success Rate(100 trajectories)** measure the average success rate across the pre-trained and fine-tuned skills of the same policies. **Fine-tuned skills(5 trajectories)** measures the policies' average success rate on the 3 new skills under a fixed and static initial configuration, where the policies are finetuned with 5 trajectories from each fine-tuned skill, and **Overall Success Rate(5 trajectories)** measures the average success rate across the pre-trained and fine-tuned skills for the same policies. ACT-LoRA out performs ACT and GMM-LoRA in the overall success rates and has fewer issues with forgetting pre-trained skills. GMM-LoRA is based on SOTA TAIL [11] model with a smaller visual backbone which can be fine-tuned for a smaller set of tasks.

Model	close door	close fridge	meat off grill	meat on grill	open box	open door	open window	phone on base	put money in safe	put rubbish in bin	slide block to target	take lid off sauce pan	toilet seat down	turn tap	water plants
	Pre-trained														
ACT-LoRA	1.25 ± 1.25	96.25 ± 2.39	72.50 ± 24.28	71.25 ± 22.11	63.75 ± 22.49	78.75 ± 16.38	73.75 ± 24.61	58.75 ± 19.83	61.25 ± 20.65	52.50 ± 17.85	46.25 ± 16.63	71.25 ± 23.84	93.75 ± 6.25	45.00 ± 15.41	25.00 ± 7.36
GMM-LoRA	1.25 ± 1.25	80.00 ± 10.61	6.25 ± 3.15	12.50 ± 7.77	37.50 ± 15.34	36.25 ± 9.66	41.25 ± 16.63	3.75 ± 3.75	28.75 ± 13.29	1.25 ± 1.25	0.00 ± 0.00	28.75 ± 12.81	70.00 ± 7.36	16.25 ± 7.18	27.50 ± 7.77
ACT	0.00 ± 0.00	72.50 ± 14.22	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	1.25 ± 1.25	0.00 ± 0.00	1.25 ± 1.25	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	50.00 ± 13.39	12.50 ± 9.46	1.25 ± 1.25
	Fine-tuned(1000 Traj.)														
ACT-LoRA	5.00	90.00	75.00	90.00	45.00	80.00	65.00	25.00	65.00	10.00	5.00	95.00	85.00	25.00	50.00
GMM-LoRA	0.00	70.00	0.00	15.00	0.00	25.00	0.00	5.00	0.00	0.00	0.00	0.00	65.00	20.00	0.00
ACT	5.00	95.00	90.00	90.00	65.00	85.00	15.00	80.00	45.00	85.00	15.00	90.00	100.00	50.00	20.00
	Fine-tuned(100 Traj.)														
ACT-LoRA	0.00	100.00	85.00	75.00	55.00	85.00	30.00	15.00	50.00	5.00	5.00	90.00	100.00	20.00	0.00
GMM-LoRA	0.00	80.00	0.00	5.00	5.00	20.00	0.00	0.00	0.00	0.00	0.00	0.00	25.00	25.00	5.00
ACT	15.00	95.00	90.00	75.00	65.00	65.00	60.00	70.00	65.00	55.00	20.00	90.00	100.00	55.00	30.00
	Fine-tuned(5 Traj., Static evaluation)														
ACT-LoRA	0.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	0.00	60.00	100.00	100.00	100.00	100.00	5.00
GMM-LoRA	0.00	0.00	0.00	0.00	0.00	15.00	0.00	0.00	0.00	5.00	35.00	25.00	85.00	80.00	5.00
ACT	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	35.00	100.00	100.00	100.00	100.00	100.00	90.00

TABLE IX: Experimental results on each skill of the RLBench dataset. We report success rate of each skill under pre-trained and fine-tuned with different number of trajectories. As we perform a five-fold validation on the skills, the statistics of the pre-trained skills come from 4 models, whereas the success rates of the fine-tuned skills come from the evaluation of a single model. For each model, we evaluate each skill by rolling out the skill in the simulator for 20 times.

suffering from catastrophic forgetting.

GMM-LoRA model performs the worst in both pre-trained skills and fine-tune skills on RLBench dataset. This is to our surprise as GMM-based model has demonstrated a strong performance in controlling robot manipulators on LIBERO dataset [20], [11]. We suspect that the reason for the poor performance is that GMM-based model suffers from joint-position controls, but further investigations are needed to verify this hypothesis.

F. Detailed Results on LIBERO

For our experiments on the LIBERO dataset, all the three models are pre-trained with 50 demonstrations for each of the pre-trained skills. To study the sample efficiency of these models, we experiment fine-tuning with 5 and 50 trajectories and report the results. Following Liu et al. [20], we report the success rate of models with 20 rollouts for each model on each skill.

We present the major results on the three task suites of the LIBERO dataset in Table II. Additionally, we present the detailed performance of each skill from three suites of the LIBERO dataset in Table X,XI,XII. For each of the three suite of the LIBERO dataset, we apply the same training schema and perform a five-fold validation on the 10 tasks of the task suite. All the three models are trained with robot trajectories in the operational control space(OCS), and went through the same pre-trained, fine-tuned training schema. During the pre-train phase, each model is trained with 50 robot demonstrations from each pre-train task for 100 epochs. In the fine-tuning phase, we only train the weights introduced by the Low-Rank Adaptor for ACT-LoRA and GMM-LoRA, while the ACT model is fine-tuned with all its weights. To study the models' performance with different data scales, we fine-tuned models for 100 and 1000 epochs when using 5 and 50 trajectories for fine-tune skills respectively.

As shown in Table II, we can observe that ACT-LoRA achieves the most stable performance across the three policies. In overall success rate, ACT-LoRA is either comparable to or better than a strong GMM-LoRA baseline. Additionally, although GMM-LoRA achieves the best performance in pre-trained skills in all the three task suites, ACT-LoRA outperforms GMM-LoRA on fine-tuned skills under all configurations without compromising much in the performance on the pre-trained skills. This demonstrates that ACT-LoRA is more suitable for continual learning than GMM-LoRA. On the other hand, ACT-LoRA shares the best performance in majority metrics on fine-tuned skills with an ACT model that undergoes full fine-tuning. However, ACT-LoRA achieves a significantly better performance than ACT in pre-trained skills and overall success rate metrics across all the three task suites. This demonstrates that ACT-LoRA is the most stable policy for continual learning when compared to the other strong baselines.

Model	0	1	2	3	4	5	6	7	8	9
	Pre-trained									
ACT-LoRA	70.00 ± 7.07	48.75 ± 18.53	80.00 ± 3.54	63.75 ± 21.35	57.50 ± 10.90	65.00 ± 21.89	95.00 ± 2.04	65.00 ± 22.27	47.50 ± 6.61	61.25 ± 4.73
GMM-LoRA	72.50 ± 10.90	38.75 ± 10.08	95.00 ± 2.04	60.00 ± 20.00	53.75 ± 5.54	36.25 ± 15.86	82.50 ± 2.50	63.75 ± 21.93	75.00 ± 2.04	70.00 ± 4.08
ACT	0.00 ± 0.00	0.00 ± 0.00	1.25 ± 1.25	1.25 ± 1.25	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
	Fine-tuned(50 Traj.)									
ACT-LoRA	40.00	65.00	45.00	65.00	15.00	65.00	40.00	25.00	25.00	20.00
GMM-LoRA	0.00	0.00	0.00	50.00	0.00	5.00	0.00	35.00	0.00	0.00
ACT	65.00	45.00	80.00	90.00	55.00	75.00	85.00	80.00	35.00	75.00
	Fine-tuned(5 Traj.)									
ACT-LoRA	35.00	85.00	5.00	55.00	10.00	45.00	60.00	35.00	5.00	20.00
GMM-LoRA	0.00	10.00	0.00	30.00	0.00	20.00	0.00	0.00	0.00	0.00
ACT	60.00	85.00	70.00	75.00	40.00	65.00	45.00	40.00	30.00	40.00

TABLE X: Experimental results on each skill of LIBERO-spatial dataset. We report success rate of each skill under pre-trained and fine-tuned with different number of trajectories. As we perform a five-fold validation on the skills, the statistics of the pre-trained skills come from 4 models, whereas the success rates of the fine-tuned skills come from the evaluation of a single model. For each model, we evaluate each skill by rolling out the skill in the simulator for 20 times.

Model	0	1	2	3	4	5	6	7	8	9
	Pre-trained									
ACT-LoRA	85.00 ± 4.56	37.50 ± 13.62	87.50 ± 6.61	37.50 ± 13.62	86.25 ± 4.27	31.25 ± 13.90	92.50 ± 3.23	65.00 ± 22.08	82.50 ± 7.22	65.00 ± 11.37
GMM-LoRA	93.75 ± 3.15	63.75 ± 17.84	96.25 ± 1.25	61.25 ± 20.55	88.75 ± 5.15	62.50 ± 21.07	88.75 ± 5.54	52.50 ± 14.79	87.50 ± 5.20	82.50 ± 7.77
ACT	2.50 ± 2.50	0.00 ± 0.00	1.25 ± 1.25	0.00 ± 0.00	5.00 ± 3.54	13.75 ± 13.75	37.50 ± 21.65	7.50 ± 4.33	47.50 ± 27.50	13.75 ± 9.44
	Fine-tuned(50 Traj.)									
ACT-LoRA	75.00	25.00	65.00	35.00	70.00	65.00	100.00	90.00	100.00	55.00
GMM-LoRA	0.00	50.00	0.00	5.00	0.00	45.00	0.00	50.00	0.00	0.00
ACT	50.00	25.00	90.00	35.00	70.00	40.00	95.00	95.00	60.00	70.00
	Fine-tuned(5 Traj.)									
ACT-LoRA	10.00	80.00	45.00	25.00	30.00	55.00	95.00	80.00	60.00	0.00
GMM-LoRA	0.00	15.00	0.00	65.00	0.00	15.00	0.00	45.00	0.00	0.00
ACT	75.00	25.00	60.00	55.00	15.00	15.00	35.00	25.00	45.00	5.00

TABLE XI: Experimental results on each skill of LIBERO-object dataset. We report success rate of each skill under pre-trained and fine-tuned with different number of trajectories. As we perform a five-fold validation on the skills, the statistics of the pre-trained skills come from 4 models, whereas the success rates of the fine-tuned skills come from the evaluation of a single model. For each model, we evaluate each skill by rolling out the skill in the simulator for 20 times.

Model	0	1	2	3	4	5	6	7	8	9
	Pre-trained									
ACT-LoRA	78.75 ± 4.73	72.50 ± 24.28	91.25 ± 3.75	31.25 ± 11.25	90.00 ± 4.08	63.75 ± 21.93	78.75 ± 3.15	70.00 ± 23.80	78.75 ± 5.15	81.25 ± 6.25
GMM-LoRA	92.50 ± 3.23	71.25 ± 22.21	98.75 ± 1.25	26.25 ± 8.75	93.75 ± 1.25	61.25 ± 21.45	72.50 ± 1.44	72.50 ± 20.97	82.50 ± 6.29	82.50 ± 4.33
ACT	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00
	Fine-tuned(50 Traj.)									
ACT-LoRA	65.00	95.00	55.00	25.00	85.00	0.00	60.00	0.00	45.00	60.00
GMM-LoRA	0.00	40.00	0.00	0.00	0.00	10.00	0.00	55.00	0.00	0.00
ACT	15.00	15.00	15.00	35.00	45.00	10.00	40.00	5.00	15.00	0.00
	Fine-tuned(5 Traj.)									
ACT-LoRA	10.00	90.00	15.00	0.00	60.00	10.00	0.00	10.00	30.00	5.00
GMM-LoRA	0.00	30.00	0.00	0.00	0.00	5.00	0.00	0.00	0.00	0.00
ACT	0.00	20.00	5.00	0.00	5.00	0.00	5.00	50.00	20.00	0.00

TABLE XII: Experimental results on each skill of LIBERO-goal dataset. We report success rate of each skill under pre-trained and fine-tuned with different number of trajectories. As we perform a five-fold validation on the skills, the statistics of the pre-trained skills come from 4 models, whereas the success rates of the fine-tuned skills come from the evaluation of a single model. For each model, we evaluate each skill by rolling out the skill in the simulator for 20 times.

G. Implementation details for ACT-LoRA

We describe the details of our implementation of the ACT-LoRA policy. Following zhao2023learning, we train with a CVAE architecture and discard the additional encoder during inference. We adjust the number of parameters for different experiments accordingly. For all of our experiments, we use a 4-layer transformer encoder both the CVAE encoder and the state encoder. For the RLBench experiments and the real-world experiments, we use a hidden dimension of 2048 and attention layers with 6 heads. For the LIBERO experiment, we use a hidden dimension of 256 and attention layers 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 for the real-world experiments and the RLBench experiments, and 4-layer transformer decoder for the LIBERO experiments. Trainable embeddings are used

Algorithm 1 The Algorithm for the Dialogue State Machine

Input:

$\mathcal{O}_0$: The initial observation of the agent

$\mathcal{S} = \{S_1, \ldots, S_K\}$: The initial skill library of the agent

$\pi_\psi, \psi = \{\psi_0, \psi_1, \ldots, \psi_K\}$: Policy π parameterized by ψ, composed of shared weights ψ_0 and skill specific weights $\{\psi_1, \ldots, \psi_K\}$

ϵ_{text}: The threshold to determine whether the two skills are the same in the semantic space

1: $\mathcal{A} \leftarrow$ GetListOfActionsFromDialogue()
2: **while** $\mathcal{A}$ is not empty **do**
3: $\quad \tau \leftarrow \mathcal{A}[0]$
4: $\quad$ **if** $\tau \in \mathcal{S}$ **then**
5: $\quad\quad$ ExecuteSkill(τ, π_ψ)
6: $\quad$ **else**
7: $\quad\quad S_i, s \leftarrow$ SearchSkillLibrary(τ)
8: $\quad\quad$ **if** $s \geq \epsilon_{\text{text}}$ **then**
9: $\quad\quad\quad$ response $\leftarrow$ ProposeSkillToHuman(S_i)
10: $\quad\quad\quad$ **if** response=agree **then**
11: $\quad\quad\quad\quad$ ExecuteSkill(S_i, π_ψ)
12: $\quad\quad\quad\quad$ Continue $\quad$ ▷ *skip line 13, 14*
13: $\quad\quad r \leftarrow$AskForRobotDemonstration(a)
14: $\quad\quad$ FinetunePolicyForNewSkill(π_ψ,r)
15: $\quad \mathcal{A} \leftarrow \mathcal{A}[1:]$

for all experiments. We also use a chunk size of 100 as it gives the best performance empirically [1]. The same configuration is also used for the baseline ACT model. As for the configuration of the low-rank adaptors, we follow TAIL [11] and use a rank size of 8 for all experiments. For both the simulation experiments and the human subject study, each skill is associated with a set of unique adaptor weights.

H. Implementation details for GMM-LoRA

We re-implemented GMM-LoRA with the help from the authors of TAIL [11] and the reference to the transformer-GMM policy from the LIBERO paper [20]. To reduce the computation cost for the original TAIL model, we use a transformer encoder in replacement to the GPT-2 temporal decoder. We also replace the CLIP image encoder with a resnet-18 model. For a fair comparison, we adjust the scale of the GMM-LoRA model to be similar to that of the ACT-LoRA model for each experiment. The GMM-LoRA 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 resnet-18 vision backbone and a frozen BERT 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 temporal encoder. Then, we use an MLP layer to project the encoded tokens into parameters for Gaussian Mixture Models(GMM). During training, the model is optimized by minimizing the negative log-likelihood loss of the ground truth actions over multiple time-steps. During inference, we sample only one action from the distribution of the GMM predicted by the model. Following TAIL [11], our GMM-LoRA model predicts an action chunk of size 10. For fair comparison, we use a GMM-LoRA of similar scale to that of the ACT-LoRA. For the LIBERO experiments, we use 8-layer of transformer encoder with 6 heads, with a hidden dimension of 256. For the RLBench experiments, we use 10-layer of transformer encoder with 8 heads, with a hidden dimension of 2048. We use a rank size of 8 for all the adaptor weights introduced by the low-rank weights.

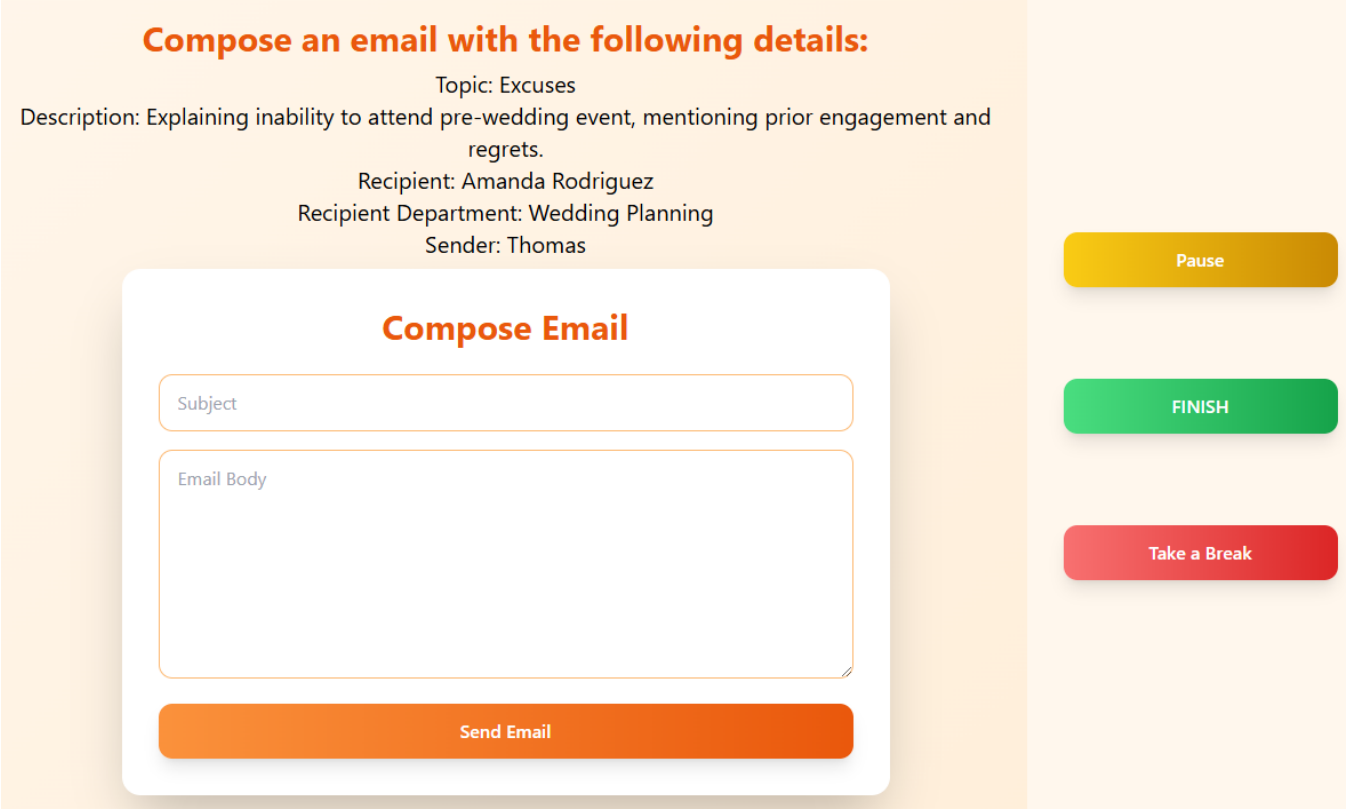


Fig. 3: A screen shot of our email writing interface, where users compose emails with synthetic topics and recipient, such as excuses, device maintenance, inquiries to local attractions, etc. These synthetic email information are generated with ChatGPT and does not include any real information.