A Implementing COLADA

Implementing the COLADA on a real robot for a user study requires more than the learning components such as the LLM interaction module and the continual learning policy. Hardware integrations are also needed for the robot to perform the tasks during the human-subjects study. Furthermore, we also need to integrate other components such as real time speech recognition and text-to-speech to facilitate the dialog interactions between the users and the robot. In this section, we describe how we integrated these additional components in our system. We then provide additional details for our LLM interaction module, and provide guidelines on how to implement our COLADA agent on a different domain.

A.1 Hardware Configuration

Our 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. 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.

Our workspace for the human-subjects study 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, and are picked and placed using pre-specified waypoints.

A.2 System Architecture

Instead of asking users to provide textual inputs through keyboard, we use spoken language as the channel of communication between users and robots. This is because that the distribution of spoken language and textual inputs can be different, and spoken language is a more natural form of interactions for applications like household robots. We use Whisper [32] from OpenAI API endpoint to transcribe users' utterances into text inputs for the LLM. Off-the-shelf real time text-to-speech model from OpenAI TTS API is used to enable our robot agents to perform vocal dialog interactions with the human users. The turns-takings in the dialog interactions are tightly controlled. We inform the human users when the robot agents are taking their utterances as inputs.

A.3 Details on Continual Learning Policies

Implementation details for ACT-LoRA. We describe the details of our implementation of the ACT-LoRA policy. Following Zhao et al. [1], 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 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.

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 [21]. 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.

Observations on GMM-based architecture. In our simulation experiments and pilot studies, we observed that GMM-based architecture appears to struggle with joint-position based control. This limitation is not studied in previous work that used GMM-based architecture [21, 11]. We tested GMM-LoRA with joint-position control in the RLBench environment, and with operational space control(OSC) in the LIBERO environment. As demonstrated in Table 2,1, while GMM-LoRA achieves a reasonable performance in LIBERO that is comparable to the GMM transformer policy [21], it struggles in all metrics in the RLBench environment. Additionally, GMM-LoRA was unstable in our pilot study on the sandwich-making domain, where joint-position control is used. It crashed the robot into the table twice, the predicted trajectories by GMM-LoRA for the real robot were not smooth. Therefore, we suspect that GMM-LoRA model struggles with joint-position controls, and further studies are needed to verify this hypothesis.

A.4 Details on LLM Interaction Module

We use GPT-4.0 Turbo with function calling as the base for our LLM interaction module. As described in the main paper, the interaction module has two major functionalities. We describe our prompting strategies at a high level, and the complete prompts will be released as a part of the code when the paper is accepted. Firstly, we use the LLM to convert the initial dialog with the users into a sequence of skills. The LLM is prompted to continue the dialog interactions with users until it gets the confirmation from the users that the complete sequence of skills has been described. This dialog history is then fed back into the LLM and converted into a sequence of task tokens, which will be used to compare semantic similarity. Here, we prompt several examples of parsing initial dialog into skill tokens. We also use chain-of-thoughts prompting [33] using a style inspired by previous work [24]. These examples are from simulated scenarios of making sandwiches that are not used in our human-subjects study. Secondly, when the COLADA agent and the inverse semantics agent encounter an unknown skill, we use the LLM to start dialog interactions with human users. For both the COLADA agent and the inverse-semantics agent, the LLM is prompted with the skill unknown to the robot agents to ask the human users to perform the corresponding skill. After that, the COLADA agent is additionally prompted to request for robot demonstrations from the human users to learn the unknown skill. Similarly, we provide examples of these interactions in the prompt, and use chain-of-thoughts prompting [33].

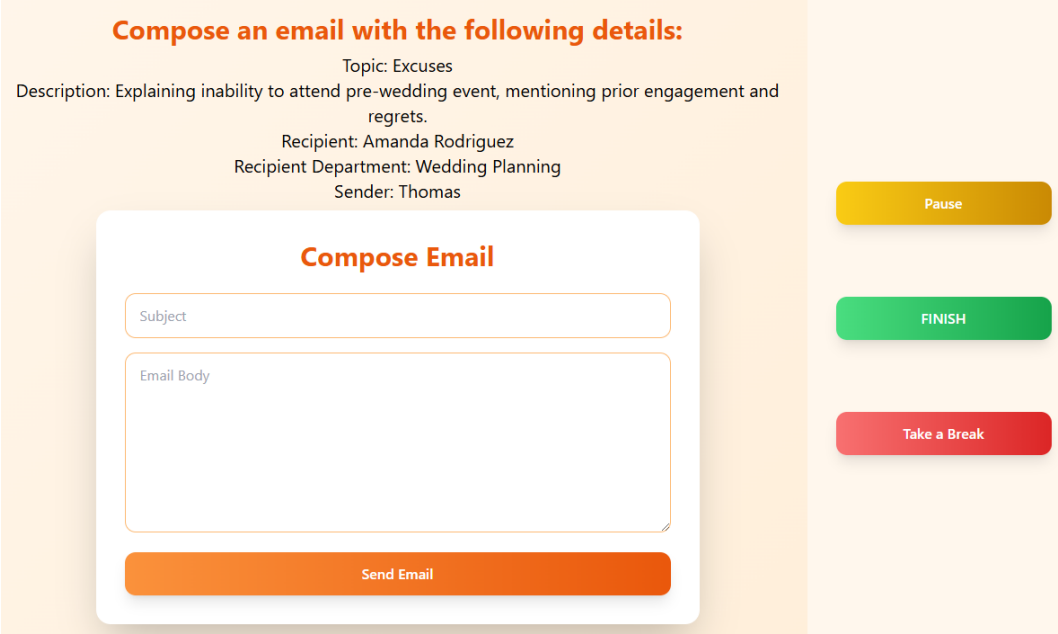

Figure 3: A screen shot of our email writing interface, where users compose emails with synthetic topics and recipient, such as excuses, device maintenance, inquiries to local attractions, etc. These synthetic email information are generated with ChatGPT and does not include any real information.

A.5 Implementing COLADA for a Different Domain

With all the implementation details, we will close this section with a discussion on what it takes to implement the COLADA agent in a completely different domain, for example, a machine-shop assistant robot.

Domain-specific dialog state machine. Firstly, the COLADA agent needs a dialog state machine that is designed specifically for the domain because dialog interactions can be different and more complicated in a different domain. In our sandwich-making domain, the COLADA agent only asks for task specification in the initial dialog interaction, and the other dialog interactions are limited to asking for help with unknown skills. However, the dialog interactions can be different and more complicated for a different domain, as multiple rounds of specifications and clarifications might be needed, even if the robot possesses the skill to perform a task. Consider the case where the COLADA agent is used as a machine-shop assistant robot, and the customer requires the robot to cut metal into certain shape. In this scenario, the robot agent also needs to use dialog to ask for specifications and clarifications for the task, such as querying the desired shape and size from the customer. To handle these different dialog interactions, the COLADA agent needs a dialog state machine that is designed for the domain. More specifically, the agent needs to have the knowledge of what information is missing to execute a task, and how to ask for that information via dialog interactions. This knowledge is domain-specific and needs to be incorporated into the dialog state machine. In our example of the machine-shop assistant robot, the LLM needs to be prompted to understand specifications, and query corresponding clarification questions such as"What shape do you want to cut the metal into?" or "What is the color of the material?" when specifications are not sufficient.

Pre-train policy with domain-specific skills. Secondly, the ACT-LoRA policy needs to be pre-trained on some common tasks in the domain. The ACT-LoRA policy can achieve its best performance if the pre-trained skills and the novel skills share the same domain. The weights of the base architecture have much more parameters($\sim 98\%$) than the weights of the task-specific adapters ($\sim 2\%$) [11]. Pre-training the policy with skills from the same domain can greatly boost the per-

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 4: 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 5: The subjective metrics for the interaction phase. We use the same ACT-LoRA policy as the policy for all the three agents.

formance of the policy on the novel skills as this exploits the domain knowledge stored in the base architecture weights learned from pre-training in the same domain. Continuing with the example of the machine-shop assistant robot, the robot should be pre-trained with necessary commonly-used basic skills, such as cutting metals and plastics, fastening screws, and drilling holes on woods. With the basic commonly used skills in this domain learned, the policy can then continually learn other skills with few demonstrations such as installing specific attachments or nailing.

Design necessary tools for robots. Lastly, tools to support the corresponding tasks might be needed as the existing tools are designed for humans and the robot might not be able to use them. For the sandwich making domain, we customized a blade and a butter brush for the robot because our robot does not have the dexterity to pick up a knife and use it. In the case of a machine-shop assistant robot, one might need to design specific tools such as hammers, glue guns, and screw drivers for robots, as the most common parallel gripper does not allow the robot to used these tools in a similar fashion to humans.

B Details 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 interface of the distraction email writing task more intuitive for

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 6: 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.

the participants, and created an instructional video for the email writing interface. Our email writing interface is demonstrated as Fig. 3. All the instructional materials we used for the study can be found in the supplemental materials and the associated webpage.

Challenges with GMM-LoRA on real robot. We did not use GMM-LoRA model as a baseline policy in our human-subjects study for two major reasons. Firstly, we use joint-position control in the real-world sandwich making domain that is used for our human-subjects study, and GMM-LoRA appears to struggle in joint-position based control in our simulation experiments. We evaluated GMM-LoRA on two simulated environments that use different controls for the robot:- RLBench that uses joint-position control and LIBERO that uses operational space control. As shown in Table 2,1, GMM-LoRA achieves a reasonable performance in the LIBERO environment but struggles in all metrics in the RLBench environment. We therefore hypothesize that GMM-LoRA model struggles with joint-position controls. However, further study is needed to verify this hypothesis. Secondly, we tested GMM-LoRA in our pilot studies, and does not have a stable performance to be used as a baseline in a human-subjects study. More specifically, the policy crashed the robot into the table when it tried to pick up a bowl. We then examined GMM-LoRA's performance on other tasks in the real-world domain, and we observed that the policy tends to move the robot at a high speed, and the trajectories predicted by the policy were not smooth. As a result, we consider the GMM-LoRA model is unsafe to be used in a human-subjects study.

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.

B.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 participants were asked to fill-out a post-survey, including questions from NASA-TLX [19], SUS [18], and 4 sub-scales from the GodSpeed Questionnaire Series [17](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).

B.2 Additional Results and Statistic Tests

The objective results on the task completion and skill success rates are presented in Table 3, and the objective results on distraction tasks are presented in Table 6,4. We also present results on subjective metrics for both phases in Table 5.

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($p = 0.001, Z = 3.02$).

B.3 Additional Findings and Analysis

We hypothesized 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.

We hypothesized that COLADA allows users to spend less time interacting with the COLADA agent in the test phase than in the interaction phase. According to a paired t-test, we find that COLADA allows participants to spend more time on finishing the distraction email writing in the test phase than in the interaction phase with significance($p < 0.001, t = 38.69$). This is because that the subjects need to spend more time teaching unknown skills to the COLADA agent in the interaction phase, and a learned COLADA agent in the test phase barely requires any time from the users. These results further suggest that the efficiency of a learning agent can be continually improved over time and experiences.

We also hypothesized that the inarticulate agent is considered worse by the users than the other agents that ask intelligent questions. Our results from Table 5 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. This indicates that even knowing that a skill is unknown is sufficient to demonstrate intelligence and be more useful to a human user.

Model	Pre-trained Skills(1000 traj.)	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 7: Complete experimental results on RLBench dataset. $*$ indicates that two models have a similar best performance. 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(1000 traj.)														
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 8: 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.

B.4 Limitations 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. Lastly, our human-subjects study does not establish the efficiency of the different policy learning algorithms; this comparison was only done in the simulation experiments where we will demonstrate the efficacy of ACT-LoRA. Partly this was also because of the stability of existing algorithms, but maybe with more data from users this issue could have been fixed.

C Details of Simulation Experiments

C.1 Detailed Results on RLBench

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 7. Detailed performance of each skill is presented in Table 8. Column **Pre-trained skills(x traj.)** measures the policies' average success rate on the skills that policies are pre-trained on after fine-tuning on x demonstration trajectories. Columns **Fine-tuned skills(x traj.)** and **Overall Success Rate(x traj.)** measure the policies' average success rate on the new skills, and the average success rate for both the pre-trained and fine-tuned skills respectively.

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-

Model	0	1	2	3	4	5	6	7	8	9
	Pre-trained(50 Traj.)									
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 9: 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(50 Traj.)									
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 10: 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.

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 7 and Table 8, 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 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 [21, 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.

C.2 Detailed Results on LIBERO

We present the major results on the three task suites of the LIBERO dataset in Table 2. Additionally, we present the detailed performance of each skill from three suites of the LIBERO dataset in Table 9,10,11. Column **Pre-trained skills(x traj.)** measures the policies' average success rate on the skills that policies are pre-trained on after fine-tuning on x demonstration trajectories. Columns **Fine-tuned skills(x traj.)** and **Overall Success Rate(x traj.)** measure the policies' average suc-

Model	0	1	2	3	4	5	6	7	8	9
	Pre-trained(50 Traj.)									
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 11: 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.

cess rate on the new skills, and the average success rate for both the pre-trained and fine-tuned skills respectively.

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 space control(OSC), 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 1000 and 100 epochs when using 5 and 50 trajectories for fine-tune skills respectively.

As shown in Table 2, 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.