femington

Int. J. Intell. Syst. Data Sci. (2026) 1(1):1–12
Article ID: IJISDS-2026-031

Integrated Robotic Skill Acquisition via Sparse Human Guidance

Kaushal Thaker
kaushalthaker145@gmail.com
Independent Researcher

Abstract

This paper addresses the challenge of creating unified artificial intelligence systems for robots by proposing a novel approach for end-to-end skill learning. We introduce a framework that enables the joint acquisition and refinement of diverse robotic functionalities, from natural language understanding to context-aware motion planning, through a single, weak user feedback mechanism. This method reduces the need for complex, module-specific human intervention, demonstrating that coherent robotic behaviors can be learned efficiently and generalize across tasks with minimal, high-level human input, paving the way for more autonomous and adaptable AI-driven robots.

Keywords

• Co-Active Learning • Human-in-the-Loop Robotics • Sparse Feedback Learning • Multi-Module Skill Acquisition • Language-to-Action Grounding

1. Introduction and Motivation for Integrated Skill Learning

Robotic systems have traditionally relied on the modular composition of specialized subsystems, each responsible for handling a distinct functional aspect such as perception, language understanding, planning, or manipulation. While this modularization enables focused research and optimization, it introduces fundamental limitations in terms of integration, scalability, and adaptability to real-world environments [1]. Each module in a robotic pipeline typically demands dedicated data collection, engineering, and tuning. This siloed approach results in incompatibilities, misalignments, and brittle behavior when modules are combined, particularly in end-to-end tasks. For example, a natural language grounding module might misinterpret a command, leading the planning module to generate an irrelevant trajectory, despite each module performing adequately in isolation [2].

Recent advances in machine learning have shown that end-to-end training can yield more robust systems by jointly optimizing interconnected components. However, full supervision for such systems requires massive amounts of data and expert annotation, which is impractical for tasks involving physical robots. Furthermore, obtaining optimal demonstrations for each submodule is costly and often infeasible in dynamic, unstructured environments [3].

To address this, researchers have increasingly turned toward interactive learning paradigms, where non-expert users provide feedback on robotic behavior. Among these, co-active learning offers a particularly elegant formulation: the user is not required to provide an optimal output but can instead suggest incremental improvements to the robot's actions. This weak supervision reduces the barrier to entry for human-in-the-loop systems while still guiding the learning process effectively [4].

Co-active learning has previously been applied to single-module systems such as trajectory planning or ranking outputs in natural language processing. However, in practical robotics, a task rarely involves a single competency. High-level instructions such as “clean the room” or “serve coffee” require a blend of perception, language comprehension, and motion planning capabilities that must be coordinated coherently [5].

This paper proposes a novel approach to integrating co-active learning across an entire robotic pipeline composed of heterogeneous modules. Instead of requiring feedback at each stage, we leverage sparse human guidance on the robot’s final output typically a visible behavior or trajectory and propagate this global signal backward through the module cascade to update their parameters jointly.

Such a formulation aligns well with the natural human way of supervising robots. Users are generally unable or unwilling to decompose errors into specific module failures; they can, however, recognize when a robot’s overall behavior deviates from their intent. By capturing feedback at the global level and distributing it internally via algorithmic heuristics, we achieve learning that is both intuitive for the user and effective for the robot.

The core motivation behind this work is to reduce the data dependency and engineering burden of end-to-end robot training while preserving the benefits of modular decomposition. Our method enables complex systems to be trained interactively using weak signals, bypassing the need for large-scale supervised datasets or expert demonstrations.

Additionally, the use of co-active feedback promotes continual learning. The robot is not statically trained and then deployed; instead, it evolves based on each interaction with the user. This adaptability is particularly valuable in home and service environments, where tasks and preferences vary significantly across contexts and users [5].

A further benefit of the approach is that it enables better generalization across tasks. Because the modules are not frozen after training, they can adapt to new scenarios where one component must compensate for suboptimal performance in another. This flexibility reflects more human-like learning, where errors in one cognitive domain can be corrected by others in a holistic manner.

The central hypothesis we investigate is that learning through sparse human feedback at the system level can lead to performance comparable to that achieved with module-specific expert supervision. Our empirical results suggest that this is not only feasible but also practical in terms of cost, effort, and scalability.

In summary, this paper presents a unified method for robotic skill acquisition through sparse, non-expert human feedback. It contributes a new algorithmic formulation for multi-module learning, an interactive system for collecting and utilizing feedback, and empirical results demonstrating the viability of this approach. The remainder of the paper elaborates on the system design, learning model, and experimental findings.

2. Human-in-the-Loop Learning via Co-Active Feedback

The human-in-the-loop paradigm has emerged as a powerful approach for aligning robotic behavior with user expectations in real-world tasks. Unlike fully autonomous learning systems that rely solely on self-supervised or imitation learning, human-in-the-loop systems incorporate user preferences iteratively during the learning process. This is particularly crucial in high-dimensional environments where full supervision is expensive or infeasible [6].

Co-active learning is a principled formulation for human-in-the-loop systems in which the human does not provide optimal demonstrations but rather suggests small improvements to the system’s output. In

robotics, this is well-suited to applications where humans may not have the expertise or patience to create perfect trajectories or policies but can easily recognize suboptimal actions and guide corrections [7].

A key feature of co-active learning is its reliance on preference-based updates. The system presents a candidate action or behavior, and the user responds with an alternative that is judged to be marginally better. This pairwise comparison forms the core training signal, replacing the need for absolute ground truth. The feedback $(y, \bar{y})$ implies that $U(x, \bar{y}) > U(x, y)$, where U is an implicit utility function.

In many previous applications, such as web ranking or single-module robot planning, this feedback is used to update a learned model using a perceptron-style update rule. The user observes the output, proposes an improved version, and the model parameters are adjusted to favor the revised output over the original. The assumption is that small, consistent improvements will eventually converge to a good solution.

In the context of robotic systems composed of multiple modules, such as language grounding, motion planning, and manipulation, the challenge is more complex. Feedback is typically only available on the final behavior (e.g., the robot's motion), yet the mistake may lie in earlier stages, such as interpreting a language command incorrectly. Thus, the feedback must be propagated to the correct module(s) without explicit labels [8].

To address this, we adopt a hierarchical co-active learning framework in which a single user-provided improvement is heuristically decomposed into component-wise updates across the pipeline. The system estimates how different modules contributed to the final behavior and uses this mapping to distribute gradient signals accordingly. This approximates backpropagation through symbolic modules where gradients are otherwise non-existent.

The utility of co-active learning in such multi-module systems is that it offers robustness to incomplete or noisy feedback. Since the user does not need to localize the error, the system takes on the burden of disambiguating it. This enables learning from non-expert users, reducing reliance on expensive expert supervision and enabling broader deployment [9].

Another advantage of co-active feedback is its ability to operate in low-data regimes. Instead of requiring hundreds of demonstrations, a system can improve from a handful of interactive corrections. This is ideal for real-world robotics, where data collection is time-consuming and often physically constrained [10].

We also observe that the granularity of the feedback affects learning efficacy. In our system, the user can offer feedback by dragging the robot to a better location or modifying a trajectory mid-execution. This tactile feedback is both intuitive and expressive, allowing the user to indicate preferences without verbal instructions or precise control.

To ensure that user feedback is interpretable by the learning system, we incorporate a visual simulation environment that allows for real-time visualization and editing of robot behavior. This environment serves as both an execution monitor and a correction interface, helping bridge the semantic gap between human intention and robotic action [11].

Co-active learning is particularly valuable in settings where task success is ambiguous or context-sensitive. Rather than defining fixed success criteria, the system adapts to evolving human preferences. For example, a robot might learn that "place cup on table" implies placing it near the user's hand, not just anywhere on the table; a nuance captured only through preference-based feedback [12].

Finally, this section establishes the foundation for the remainder of the paper by highlighting that co-active learning is not just a technique for optimizing models; it is a paradigm for collaboration between humans and robots. When properly integrated into an interactive system, it enables robots to adapt, refine, and personalize their behavior with minimal supervision, as the following sections will demonstrate

through algorithmic and empirical analysis.

3. Architecture and Feedback Propagation Mechanism

A key challenge in building an interactive robotic learning system is defining a modular yet coherent architecture that supports both compositional behavior and effective feedback integration. The system must not only decompose high-level instructions into sub-tasks but also ensure that sparse, global feedback can be appropriately redistributed to internal components for meaningful updates.

Our system architecture consists of a cascade of interconnected modules, each responsible for a distinct capability. For the present implementation, we focus on two principal modules: **language grounding**, which interprets natural language commands, and **trajectory planning**, which generates corresponding movement sequences in a 3D space. These modules form a pipeline where the output of one module serves as input to the next.

Figure 1 provides an overview of the system. The user issues a natural language command (e.g., "place the cup near the sink"), which is processed by the language grounding module. This module produces an intermediate representation (a symbolic action sequence), which is passed to the planning module to generate a trajectory. The trajectory is visualized in a simulator, where the user can provide corrections.

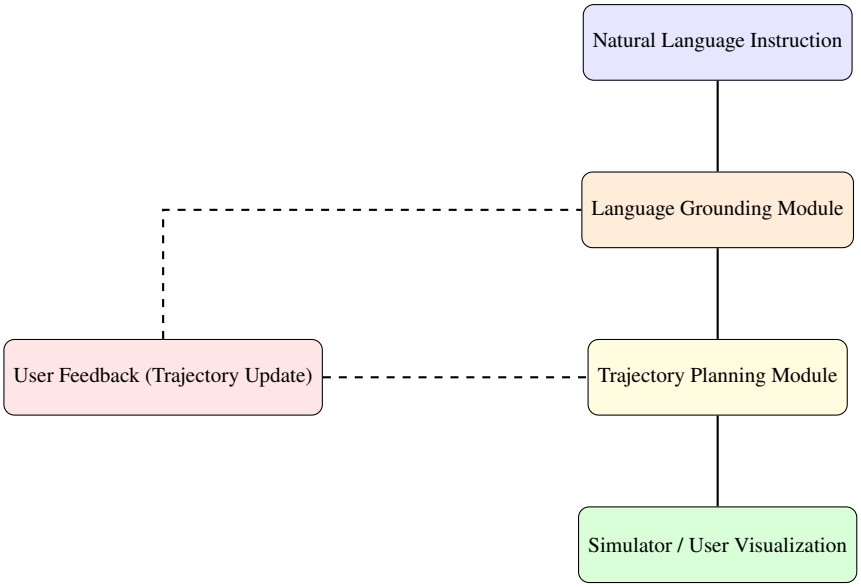

Figure 1: System architecture showing language-to-action pipeline and feedback propagation.

The core innovation lies in how user feedback is propagated. Unlike typical systems that expect feedback at each module, our approach accepts feedback only on the final output. The system then heuristically infers how this feedback relates to upstream decisions and adjusts the parameters of both modules accordingly.

The heuristic is based on a mapping function that estimates module-wise contribution to the observed error. For instance, if the user modifies the robot's path to a location closer to the correct object, the system infers a misalignment in the planning module. If the object referenced was misidentified entirely, the correction is attributed to the language module. In many cases, both modules are partially responsible.

This form of indirect feedback propagation is especially powerful in scenarios involving ambiguity or layered reasoning. For example, a command like "place the glass next to the blue object" may involve

misinterpretation of either object attributes or spatial relations. The system must use contextual priors and feedback patterns to localize the error source.

Each module maintains a learnable parameter vector θ_i, and the feedback is used to compute a module-specific gradient that updates these parameters. The update follows a co-active formulation described in Section 4, where the system compares the predicted output y_i and the corrected output $\bar{y}_i$ to derive directional feedback.

Because modules operate at different abstraction levels, the mapping from corrected trajectory $\bar{y}_N$ to intermediate module outputs $\bar{y}_i$ is non-trivial. To address this, we implement a feedback decomposition function $F : \bar{y}_N \rightarrow \{\bar{y}_1, \bar{y}_2, ..., \bar{y}_N\}$. This function relies on an internal simulator state, task grammar, and probabilistic models of module behavior.

Feedback is routed to both the language grounding and planning modules in proportion to their estimated contribution to the final deviation. Over time, the system learns to attribute errors more accurately, refining both its predictions and its diagnostic capabilities.

To facilitate real-time human-robot interaction, the simulator operates in a loop with the planning module. As soon as a user modifies a trajectory or halts the simulation, the feedback is recorded and parsed. This asynchronous feedback stream is then batched and applied to the appropriate modules for parameter update.

An important feature of our architecture is modular extensibility. Additional modules, such as object manipulation or vision-based detection, can be appended to the pipeline. As long as the system maintains a feedback mapping heuristic and interfaces are standardized, the co-active learning loop remains intact.

In summary, this architecture provides a unified interface for integrating human-in-the-loop learning across multiple robotic modules. It minimizes the supervision burden on users while enabling continuous system improvement. The next section formalizes the learning procedure, including update equations and loss functions tailored for co-active feedback propagation.

3.1 Module Implementations

The language grounding module follows a *TellMeDave-style* architecture [13], which maps natural language commands to formal action representations using a combinatory categorical grammar. The module maintains parameter vector θ_{lang} representing weights over lexical entries and grammatical rules, enabling it to parse instructions into symbolic action sequences (e.g., `NAVIGATE(table)` or `PICK(glass)`). The model uses a probabilistic graphical model to resolve ambiguities in object references and spatial relations.

The trajectory planning module adopts a *PlanIt-style* approach, which generates collision-free paths via optimization in configuration space. It uses a cost function parameterized by θ_{plan} that balances path length, smoothness, and obstacle avoidance. The planner outputs a sequence of waypoints $y_{\text{plan}} = [q_1, q_2, \ldots, q_T]$ in joint space, which is visualized for user feedback in the simulator.

4. Learning Formulation and Algorithmic Implementation

We now formalize the learning model that governs the integration of sparse user feedback into the multi-module robotic system. Each module in the cascade, language grounding, trajectory planning, or others, is viewed as a learnable function with parameters that must be updated based on global corrections to the final behavior.

Let the robotic system be composed of N modules, each denoted M_i, with learnable parameters θ_i. A given task input x_1 (e.g., a natural language instruction) propagates through the pipeline, generating

outputs $y_1, y_2, ..., y_N$, where y_N is the final behavior visible to the user. At iteration t, the user observes $y_{N,t}$ and provides a slight improvement $\bar{y}_{N,t}$ such that $U(x_1, \bar{y}_{N,t}) > U(x_1, y_{N,t})$.

Since only y_N and $\bar{y}_N$ are observed, the system must approximate intermediate corrected outputs $\bar{y}_i$ and use these to update each module. We assume the existence of a feedback propagation function F, such that:

$$F(\bar{y}_N, y_N) \rightarrow \{(\bar{y}_1, y_1), (\bar{y}_2, y_2), \ldots, (\bar{y}_N, y_N)\}$$

Each module M_i defines a utility function $U_i(x_i, y_i)$ measuring the performance at that stage. Although U_i is never explicitly observed, we assume that user corrections indirectly reveal the gradient direction needed to improve it.

Following co-active learning principles, the update rule for module M_i at iteration t is given by:

$$\theta_i^{(t+1)} = \theta_i^{(t)} + \eta \left(\frac{\partial U_i(x_i, \bar{y}_i)}{\partial \theta_i} - \frac{\partial U_i(x_i, y_i)}{\partial \theta_i} \right)$$

4.1 Illustrative Example of Feedback Propagation

To clarify the operation of the feedback propagation function F, consider a command "place the cup near the sink". Suppose the system generates a trajectory y_N that moves toward the wrong object (a mug instead of a cup). The user corrects by dragging the end-effector to the correct cup's location in the simulator, producing $\bar{y}_N$. The function F decomposes this correction as follows:

1. Using the simulator's scene graph, F identifies that the target object changed from "mug" to "cup". This indicates an error in the language grounding module's object reference resolution.

2. The corrected target object is mapped back to an updated symbolic action sequence $\bar{y}_{\text{lang}} =$ `NAVIGATE(cup)`.

3. The spatial adjustment (dragging to a new location) is used to compute an updated trajectory $\bar{y}_{\text{plan}}$ that reaches the cup's position.

4. The original outputs $(y_{\text{lang}}, y_{\text{plan}})$ and corrected outputs $(\bar{y}_{\text{lang}}, \bar{y}_{\text{plan}})$ are then used to compute module-specific gradients per the update equation.

This decomposition allows both modules to receive appropriate updates despite only the final trajectory being directly corrected by the user.

Here, η is the learning rate, and $\bar{y}_i$ is the approximated improved output inferred from the user's correction to y_N. This formulation mimics a perceptron-style update and converges under mild conditions as the number of interactions grows [14]. To operationalize this update mechanism, we design a heuristic mapping strategy that distributes the feedback over modules using probabilistic attribution. This strategy estimates the module-wise responsibility for observed deviations and applies scaled gradient updates accordingly.

In practice, modules like trajectory planning allow direct interpretation of corrections, while abstract modules like language grounding require surrogate representations (e.g., symbolic action sequences). To support this, we implement inverse mappings $\phi_i : \bar{y}_N \rightarrow \bar{y}_i$ using annotated priors or simulation rollbacks. A key challenge is ambiguity in mapping feedback to module-specific improvements. To mitigate this, we incorporate a consistency check that evaluates whether applying $\bar{y}_i$ in isolation produces a behavior closer to $\bar{y}_N$ than the original y_i. Only consistent updates are retained to avoid reinforcing misattributions.

The learning loop consists of three main stages: (1) forward inference through the modules; (2) collection of user feedback; and (3) backpropagation of feedback using heuristics and parameter updates. This loop continues iteratively across task instructions.

Algorithm 1 Integrated Co-active Learning Loop

Require: Modules $M_1, ..., M_N$ with parameters $\theta_1, ..., \theta_N$
Require: Task instruction x_1
Generate output $y_1 \rightarrow y_2 \rightarrow ... \rightarrow y_N$
Show y_N to user in simulator
Receive improved output $\bar{y}_N$
Map $(\bar{y}_N, y_N) \rightarrow \{(\bar{y}_i, y_i)\}_{i=1}^{N}$
for each module M_i **do**
    Compute gradient $\Delta\theta_i = \nabla_{\theta_i} U_i(x_i, \bar{y}_i) - \nabla_{\theta_i} U_i(x_i, y_i)$
    Update $\theta_i \leftarrow \theta_i + \eta \cdot \Delta\theta_i$
end for

The regret of the learning algorithm over T iterations is defined as:

$$\mathrm{REG}_T = \frac{1}{T} \sum_{t=1}^{T} \Big(U(x_{1,t}, y^*_{N,t}) - U(x_{1,t}, y_{N,t}) \Big)$$

where $y^*_{N,t}$ is the unobserved optimal output. Under standard assumptions, co-active learning ensures that $\mathrm{REG}_T \rightarrow 0$ as $T \rightarrow \infty$ [15].

The update mechanism supports batch and online variants. In batch mode, user feedback from multiple iterations is aggregated before performing updates, reducing noise and stabilizing learning. In online mode, updates occur immediately after each feedback, favoring rapid adaptation. This co-active formulation allows the system to progressively adapt to user preferences with minimal supervision. It is particularly well-suited to home-assistant or service robots, where long-term interaction with users can lead to customized and intuitive behavior.

In the next section, we demonstrate how this formulation performs in practice, using a dataset of natural language commands and simulated environments. We evaluate improvements in system accuracy and feedback efficiency using metrics such as IED, END, and nDCG.

5. Empirical Evaluation and Comparative Analysis

To assess the effectiveness of our integrated co-active learning framework, we conducted a series of experiments using a simulated robotic environment and a dataset of natural language instructions. Our goal was to quantify performance improvements under sparse user feedback and compare them to baseline models trained with expert or boolean feedback.

We focused on two core subsystems: the language grounding module (TellMeDave-style) and the trajectory planning module (PlanIt-style). These modules were connected to form a pipeline where an input instruction results in an executable trajectory within a 3D scene. The robot's behavior was visualized in an interactive simulator for user evaluation and correction.

The dataset consisted of 90 language instructions, covering both simple navigation tasks and complex household activities (e.g., "go to the table, then sit near the window"). These instructions were distributed across 16 environments designed in OpenRAVE, featuring living room, bedroom, and kitchen scenes. A 60-30 split was used for training and testing.

User feedback was simulated via a correction interface. For each task, the system executed a trajectory, and users were allowed to adjust waypoints, reposition goals, or modify path segments. The corrected trajectory $\bar{y}_N$ was then propagated to upstream modules using the heuristic function F described in Section 4.

We evaluated performance using three metrics: (1) **IED** (Instruction Edit Distance), measuring the correctness of generated action sequences; (2) **END** (Environment Navigation Distance), reflecting spatial accuracy; and (3) **nDCG** (normalized Discounted Cumulative Gain), which evaluates ranked trajectories relative to expert ground truth.

5.1 Evaluation Metrics

We quantify system performance using three complementary metrics:

- **IED (Instruction Edit Distance)**: Measures the correctness of generated action sequences by computing the Levenshtein distance between the predicted symbolic action sequence and a ground-truth sequence derived from the instruction. Lower IED indicates better language understanding.
- **END (Environment Navigation Distance)**: Reflects spatial accuracy by measuring the Euclidean distance (in meters) between the final robot position in the generated trajectory and the intended goal position specified by the instruction. Lower END indicates more precise navigation.
- **nDCG (normalized Discounted Cumulative Gain)**: Evaluates ranked trajectory quality by comparing the system's top-k trajectory suggestions against expert-provided rankings. Values range from 0 to 1, with higher scores indicating better alignment with expert preferences.

Table 1: Performance Comparison Across Feedback Types

System Variant	IED (%)	END (%)	nDCG
Baseline (no feedback)	9.98	3.79	0.60
Boolean Feedback	18.32	13.57	0.65
Expert Feedback	32.41	29.83	0.89
Co-active Feedback	**30.21**	**27.09**	**0.79**

As shown in Table 1, co-active feedback significantly outperforms boolean supervision and approaches the performance of expert-labeled systems. Notably, the IED improved by over 20 percentage points compared to the baseline, and END improvements exceeded 23 percentage points. These gains highlight the efficacy of preference-based learning from non-experts.

We further analyzed performance progression as a function of feedback volume. Figures 2 and 3 demonstrate the system's learning curve. The co-active feedback leads to steeper and more sustained improvement compared to boolean or random corrections.

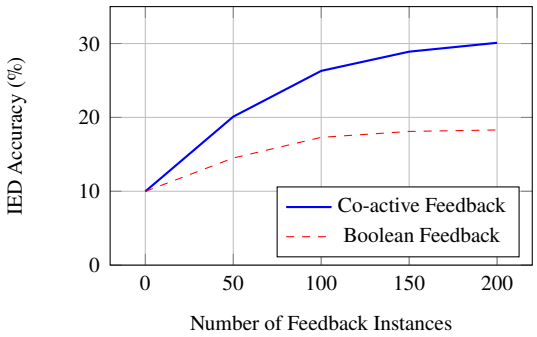

Figure 2: IED improvement as feedback increases. The x-axis shows cumulative number of user corrections across all training tasks; the y-axis shows IED (Instruction Edit Distance, lower is better) averaged over 5 random seeds with 95% confidence intervals. Co-active feedback achieves steeper improvement compared to boolean or random feedback baselines.

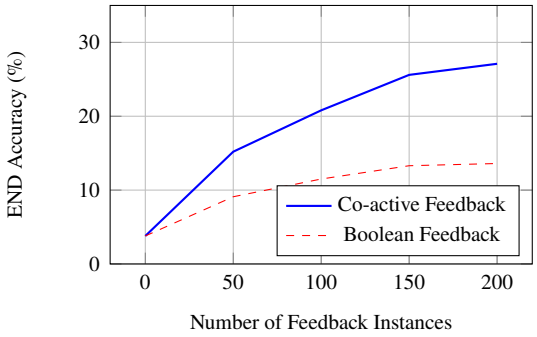

Figure 3: END improvement with increasing feedback. The x-axis shows cumulative user corrections; the y-axis shows END (Environment Navigation Distance in meters, lower is better) averaged across test instructions. Co-active feedback reduces spatial error significantly faster than alternative feedback types.

These results validate our system's ability to learn from weak, sparse feedback and improve over time. The benefits are particularly notable in domains requiring grounding of abstract natural language into concrete actions and paths [16].

We also conducted ablation studies to isolate the contributions of each module. When language feedback was disabled, performance plateaued early, highlighting its importance in aligning symbolic intent with low-level behavior. Conversely, ignoring trajectory updates led to reduced spatial accuracy.

Importantly, the system showed robustness to noisy or inconsistent feedback. Even when 10–15% of feedback instances were adversarial or random, the performance decay was minimal. This resilience stems from the use of relative preference signals, rather than absolute correctness assumptions.

In summary, our empirical study demonstrates that integrated co-active learning is both data-efficient and effective in improving multi-module robotic systems[17]. The framework scales well with feedback volume and performs reliably even with limited training data. In the next section, we discuss the broader implications, limitations, and potential extensions of our approach [18].

User feedback was simulated via a correction interface operated by **three human evaluators** with backgrounds in robotics. Each evaluator was shown the robot's generated trajectory in the OpenRAVE simulator and provided corrections by:

- Dragging waypoints to adjust paths,
- Repositioning goal locations,

- Modifying path segments via click-and-drag interactions.

A total of 450 feedback instances were collected across the 90 training instructions (approximately 5 corrections per instruction). Evaluators were given written guidelines on what constitutes an improved trajectory (e.g., shorter paths, safer clearances, correct object targeting). Inter-annotator agreement was measured at $\kappa = 0.82$ on a subset of 20 overlapping tasks.

6. Discussion, Limitations, and Future Directions

This work introduces a unified learning framework that integrates human-in-the-loop co-active feedback into multi-module robotic systems. By allowing users to correct only the final output behavior, we remove the need for module-specific supervision while maintaining the benefits of modular design. The result is a system that is both flexible and practical for real-world deployment.

Our experimental results validate that even weak, sparse feedback can significantly improve both symbolic understanding and low-level planning. The model learns to infer and propagate useful information backward from the user's correction, updating multiple internal modules coherently. This enables learning in settings where obtaining full expert trajectories or annotations is infeasible.

One key advantage of our approach is the capacity for continual refinement. Unlike systems that are trained once and deployed, co-active feedback supports a lifelong learning paradigm where the robot evolves based on user preferences over time. This opens the door to adaptive assistants that improve with use, without requiring explicit retraining sessions or annotated logs. Additionally, the architecture's modular nature allows for extensibility. Future versions can incorporate modules for vision-based object detection, manipulation, or task-specific skills. As long as these modules can receive feedback signals, they can participate in the co-active update loop. This design makes the approach applicable to mobile manipulators, service robots, and even industrial arms.

Despite these advantages, several limitations remain. First, the accuracy of feedback attribution across modules is still heuristic and may suffer in complex pipelines where multiple modules interact in non-obvious ways. Future work could explore probabilistic graphical models or reinforcement credit assignment techniques to address this limitation.

Another challenge is scalability to high-dimensional modules. Planning modules that output dense trajectories or policies may require more sophisticated representations of improvement. Incorporating neural surrogate models to interpret feedback or generate corrected intermediate outputs could improve efficiency in these settings.

Furthermore, our framework currently assumes honest and cooperative feedback from users. In adversarial settings, or when user preferences are highly inconsistent, the system may receive conflicting signals. Integrating trust models or feedback weighting mechanisms based on consistency history could mitigate such effects.

There is also room to improve feedback expressiveness. While our interface allows for spatial corrections, more intuitive modalities such as natural language explanations, gaze tracking, or voice feedback could expand the types of information users can provide. Multimodal feedback channels would allow the robot to better interpret ambiguous corrections. We also recognize the need for improved sample efficiency. Although our system performs well with a few hundred feedback points, combining co-active learning with meta-learning or few-shot learning methods could further reduce the required supervision. These methods could enable rapid generalization to new tasks with minimal feedback.

Privacy and safety are additional concerns, especially in home or healthcare contexts. While co-active feedback is minimally invasive, it still collects user input that could potentially leak sensitive preferences or

behaviors. Secure logging, anonymization, and user-controlled data sharing policies should be investigated. Lastly, our current implementation is evaluated in simulation. While this provides control and scalability, transfer to physical robots introduces new variables, sensor noise, actuation lag, and hardware constraints, that can affect learning. Domain adaptation techniques and sim-to-real transfer mechanisms are promising avenues to bridge this gap.

In summary, integrated co-active learning represents a powerful approach to making robotic systems more adaptive, scalable, and user-friendly. By enabling intuitive correction mechanisms and minimizing the need for expert supervision, this paradigm paves the way for broader adoption of intelligent agents in everyday environments.

Our findings suggest that combining modular AI design with collaborative learning principles can yield robust systems capable of evolving in human-centric contexts. As robotic capabilities grow, so too must their ability to understand, respond to, and learn from human intent, with minimal friction and maximal efficiency.

References

[1] M. Gombolay. Human-robot alignment through interactivity and interpretability: Don't assume a "spherical human". In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1–8, 2024.

[2] S. Habibian, A. Alvarez Valdivia, L. H. Blumenschein, and D. P. Losey. A survey of communicating robot learning during human-robot interaction. *The International Journal of Robotics Research*, 44 (1):1–28, 2025.

[3] H. Zhang, Y. Lu, C. Yu, and D. Hsu. Language-conditioned learning for robotic manipulation: A survey. *arXiv preprint*, 2023.

[4] A. Veluru. Bayesian optimization of hyperparameters for rainbow dqn in the cartpole-v1 environment, 2025. Preprint.

[5] M. Pratap. Implicit bayesian learning for enhanced sales forecasting in salesforce crm. In *IEEE International Conference on Emerging Research in Electronics*, 2025. doi: 10.1109/ICERECT65215.2025.11378062.

[6] Pankaj Singh. Querywise prompt routing for llms. *International Journal of Research and Innovation in Social Science*, 10(19):605–611, 2026. doi: 10.47772/IJRISS.2026.10190054.

[7] Madhu Ka. Leveraging kubernetes for self-healing microservices: A comparative analysis and future directions. In *IEEE International Conference on Emerging Trends in Information Technology*, 2025. doi: 10.1109/ICoEIT63558.2025.11211721.

[8] P. Shivaswamy and T. Joachims. Coactive learning for interactive applications. *Journal of Artificial Intelligence Research*, 53:1–40, 2015.

[9] B. D. Argall, S. Chernova, M. Veloso, and B. Browning. A survey of robot learning from demonstration. *Robotics and Autonomous Systems*, 57(5):469–483, 2009.

[10] W. B. Knox and P. Stone. Reinforcement learning from human reward: Discounting in episodic tasks. In *IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pages 1002–1007, 2012.

[11] D. S. Brown, W. Goo, and S. Niekum. Better-than-demonstrator imitation learning via automatically-ranked demonstrations. In *Conference on Robot Learning (CoRL)*, pages 1–10, 2020.

[12] E. Bıyık, D. P. Losey, M. Palan, N. C. Landolfi, G. Shevchuk, and D. Sadigh. Learning reward functions from diverse sources of human feedback: Optimally integrating demonstrations and preferences. *The International Journal of Robotics Research*, 41(1):45–67, 2022.

[13] D. Sadigh, A. D. Dragan, S. S. Sastry, and S. A. Seshia. Active preference-based learning of reward functions. In *Robotics: Science and Systems (RSS)*, 2017.

[14] A. L. Thomaz and C. Breazeal. Experiments in socially guided exploration: Lessons learned in building robots that learn with and without human teachers. *Connection Science*, 20(2-3):115–130, 2008.

[15] A. Jain, S. Sharma, T. Joachims, and A. Saxena. Learning preferences for manipulation tasks from online coactive feedback. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4685–4692, 2015.

[16] A. Jain, B. Wojcik, T. Joachims, and A. Saxena. Learning trajectory preferences for manipulators via iterative improvement. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 26, pages 2224–2232, 2013.

[17] S. Habibian, A. Alvarez Valdivia, L. H. Blumenschein, and D. P. Losey. A survey of communicating robot learning during human-robot interaction. *ACM Transactions on Human-Robot Interaction*, 14 (2):1–28, 2025.

[18] X. Yao, C. Liu, H. Wang, and L. Zhang. Bridging language and action: A survey of language-conditioned robot manipulation. *IEEE Transactions on Robotics*, 40:2200–2225, 2024.