WASP SE Assignment

Ermanno Bartoli

1 Introduction

Research Area and Topic. Long-term human-robot interaction (HRI) involves a robot that progressively evolves and adapts its behavior through continuous learning from human interactions over time Silver 2011. In real-world scenarios, the information from which a robot learns is often not provided in a structured or pre-defined manner. Instead, data arrives as a continuous, uncontrolled stream and the robot cannot revisit past experiences to refine its previously acquired knowledge Lesort et al. 2020.

Household environments often contain multiple objects, tasks, and occupants. Consequently, users expect the robot to either already be equipped with or acquire the ability to perform a variety of tasks Beer et al. 2012; Gross et al. 2015a. This necessitates that the robot not only learns to execute different tasks from different individuals but also personalizes these tasks to accommodate individual preferences. Furthermore, the robot should recognize the appropriate context for executing each task, with the aim of enabling them to seamlessly integrate into the dynamic routines and interactions of a busy household Wilson et al. 2019; Gross et al. 2015b.

For instance, a robot designed to interact with multiple family members will encounter unbalanced and varied training data due to the unique habits and routines of each individual Patel and Chernova 2022. As the robot often interacts with only one person at a time, it will be exposed to tasks and behaviors specific to that individual's habits at that moment. This can lead to skewed or incomplete learning, as the tasks a robot learns from one household member may differ significantly from those it encounters with another, resulting in a highly dynamic and uneven distribution of training data over time. Moreover, each individual's routine can change over time, meaning that the robot should also be equipped to handle changes within users as well as between users.

Deep Neural Networks, regardless of their size or complexity, if trained sequentially on multiple tasks, are prone to catastrophic forgetting, where learning a new task disrupts and overwrites knowledge gained from previous tasks Kirkpatrick et al. 2017. The challenge of learning incrementally from non-independent and identically distributed (non-i.i.d.) data, while retaining previously acquired knowledge without forgetting, is known as Continual Learning (CL) Zenke, Poole, and Ganguli 2017. CL is a relatively recent area of research that has primarily been explored in the context of computer vision or standard classification tasks as proof-of-concept studies Wang et al. 2024. However, one of the most compelling applications of CL is in scenarios where a robot interacts with humans and learns continuously over time Lesort et al. 2020. Real-world human-robot interaction scenarios exemplify the exact conditions where non-i.i.d. data naturally emerges, making CL especially critical in these environments. In our work, we aim to address the problem of incremental learning in household environments, where the robot is expected to engage in long-term interactions with humans, continuously learning new concepts over time without forgetting previously acquired knowledge. Specifically, we seek to answer the following research questions:

- (***RQ1***) How can a robot learn new tasks over time from real-world data and for real-world applications without experiencing catastrophic forgetting?
- (***RQ2***) How can we sustain real-time human-robot interaction by ensuring that the robot aligns its task execution to meet human expectations?

2 Lecture Principles

Building a robust robotic system that learns continuously is a software engineering challenge that often requires a rigorous engineering attention. This section discusses how two concepts from the lecture on verification vs. validation and the lecture on test oracle can have impact on my research methodology.

2.1 Verification vs. Validation in Continual Learning for HRI

The distinction between verification ("building the product right") and validation ("building the right product") is central to my research, providing a formal framework for my two research questions.

Verification directly addresses **RQ1**. It involves technically evaluating the Continual Learning (CL) algorithm against established benchmarks and metrics. Success is quantified by measuring stability (retaining old knowledge) and plasticity (learning new knowledge). For instance, a verification test would train the robot on a sequence of tasks and then measure its accuracy on earlier tasks to quantify catastrophic forgetting. This process confirms that the underlying learning mechanism is being built correctly according to CL theory, which is a necessary but insufficient condition for success.

Validation, conversely, addresses **RQ2**. A robot could be successfully verified, i.e. demonstrating zero forgetting on a technical benchmark, but still utterly fail validation in a real home. For instance, if it executes tasks at socially inappropriate times or for the wrong person, it is not the right product for seamless household integration. A system that is verified but not validated represents a technical success but a practical failure, wasting resources and failing to deliver user value. Building on that, in my methodology I treat these as distinct activities: verification is handled with offline, quantitative benchmarks, while validation relies on long-term, in-the-wild user studies to gather the qualitative feedback needed to assess true fitness-for-purpose.

2.2 The Challenge of the Test Oracle in Long-Term HRI

A test oracle determines if a test has passed or failed. In my HRI research, the oracle is the human's dynamic, subjective, and often unstated expectation. There is no predefined script of "correct" behavior. In this contect, the oracle is very complex because it is: personalized, context-dependent, non-stationary (user preferences evolve), and often implicit (conveyed through subtle cues, not explicit commands). The core difficulty of **RQ2** is that the robot cannot perfectly query this oracle.

The engineering challenge, therefore, is to build a system that can approximate and interact with this oracle. My approach is to integrate testing into the live interaction loop. The robot will be designed with mechanisms to solicit and interpret feedback (e.g., from verbal corrections or non-verbal cues), which serves as the oracle's judgment. A negative reaction is treated as a failed test case, and the feedback becomes a continual learning signal to update the robot's internal model. This transforms testing from a discrete, offline evaluation step into the core, online engine of adaptation and alignment.

3 Guest-Lecture Principles

The guest lecture on Requirements Engineering (RE) provides essential methodologies for complex, socio-technical systems like a household robot. This section applies two RE principles, i.e. the problem-space/solution-space distinction and stakeholder elicitation, into my research.

3.1 Distinguishing the Problem-Space from the Solution-Space

The RE lecture stressed the importance of understanding the Problem-Space ('why' and 'what') before defining the Solution-Space ('how'). My research risks focusing prematurely on the solution-

space of **RQ1** (e.g., "which CL algorithm is best?"). This is a classic pitfall where engineering effort is invested in a solution before the problem is fully understood. The true challenge lies in the problem-space of **RQ2**: "what does successful alignment with human expectations mean?" Without this, even a perfect CL algorithm is a solution applied to the wrong problem. For instance, a solution-first approach might lead to an algorithm that requires explicit, labeled corrections from the user, ignoring the problem-space reality that humans prefer implicit, seamless interaction. Applying this principle means my research must prioritize formative user studies to define "successful alignment," ensuring that technical work is always grounded in solving a real human problem.

3.2 Stakeholder and Goal Elicitation

RE provides techniques for identifying stakeholders and their often-conflicting goals. In my research, the "user" is not a unique concept; on the contrary a household contains distinct stakeholders (e.g., 'Parent A', 'Child') with unique goals. The Child's usage goal to "play a game" may conflict with a Parent's system goal for the "robot to operate silently." A robot learning from a raw interaction stream will struggle to negotiate these conflicts, satisfying one person at the expense of another.

By applying goal modeling, I can create personas for each stakeholder and formally map their goals and potential conflicts. This transforms the vague objective of "meeting expectations" into a concrete multi-objective optimization problem: how can the robot learn actions that best satisfy a network of competing goals? This is a significant research challenge in itself, requiring the robot's architecture to explicitly model and reason about these competing interests. It provides a structured, engineering-based approach to designing and evaluating the robot's social intelligence.

4 Data Scientists versus Software Engineers

The "Machine Learning in Production" book highlights the distinct roles of data scientists (exploration) and software engineers (production). This section reflects on these differences and their future.

4.1 Agreement on Essential Differences

I fully agree with the distinction between the data scientist's focus on exploration and the software engineer's on production. My research embodies both roles. The **Data Scientist** role involves experimenting with CL algorithms in simulations to maximize model quality (accuracy, forgetting rate). The **Software Engineer** role involves integrating the best algorithm onto a physical robot, a systems-building challenge focused on system quality (reliability, latency, safety). A 99% accurate model is useless if its implementation crashes the robot's OS. The book's claim that the model is a small part of the production system resonates strongly; my CL algorithm is the "brain," but the SE effort provides the "body" needed to function in the real world.

4.2 The Future of Roles: A Necessary Convergence

I believe a possible next step for researchers in my field lies in a convergence of these roles, not further specialization. The model of a data scientist "throwing a model over the wall" is too inefficient for complex systems like HRI, where the feedback loop between system and model is the basis for learning. A data scientist unaware of the robot's real-time constraints will design unusable models, and an engineer unaware of the model's failure modes cannot build effective feedback mechanisms. Success in my project requires a hybrid "AI Engineer" perspective.

This convergence is already being formalized via MLOps and the rise of the T-shaped professional: an expert in one domain with a robust, practical understanding of the other. Building these hybrid skills is a challenge for both academia and industry, requiring new curricula and

training programs. However, this integrated expertise is the foundation of true AI Engineering and is essential for building the next generation of intelligent systems.

5 Paper Analysis

This section analyzes two CAIN papers, connecting them to my research and situating them within the broader field of AI Engineering.

5.1 Paper 1: Continuous Human-LLM Co-Programming (COPMA)

Core Idea: Song et al. introduce COPMA, a structured methodology for building multi-agent systems by balancing flexible LLM-driven agents with deterministic code blocks. Its key contribution is a set of patterns for refactoring functionality between the "LLM world" (for adaptability and rapid prototyping) and the "code world" (for reliability, cost, and efficiency). This provides an engineering discipline to manage LLM unpredictability, a central challenge in modern AI development.

Relation to Research: COPMA's design-time refactoring is a powerful analogy for the run-time learning in my HRI project. An emerging user preference in my robot can be seen as a flexible "agent," while a stable, well-learned routine can be considered a hardened "code block." The process of a robot's learned behavior solidifying into a habit is a run-time version of COPMA's refactoring.

Integration & Adaptation of Research: In a larger (fictional) "Aether Home OS" project, COPMA would be the development methodology for the entire system, while my research would provide the robot's real-time personalization mechanism. This paper inspires several long-term adaptations to my research, moving it toward a more sophisticated form of knowledge management.

A primary adaptation is to incorporate **dynamic run-time refactoring**. My current CL approach updates a single, monolithic policy network. A more advanced architecture would enable the robot to automatically decide how a learned behavior should be represented. For a new or evolving task, the robot would use a flexible, computationally intensive policy network (the "agent"). However, once a task has been performed consistently and successfully (e.g., with a low variance in positive user feedback over N interactions), the robot could compile this learned policy into a more lightweight, deterministic script or decision tree (the "code block"). This on-robot knowledge distillation presents a significant research challenge: how can a complex neural policy be simplified into a verifiable script without losing the essential nuances of the interaction? Success here would dramatically enhance the robot's efficiency and predictability, a key component of building user trust (**RQ2**).

This leads to the idea of a **certainty-based behavior model**. The robot could maintain a "certainty score" for every learned task and preference, possibly using Bayesian methods to quantify confidence. Low-certainty behaviors would be handled by its adaptable CL model, which actively seeks feedback. High-certainty behaviors would be executed by the compiled, efficient scripts. The core of my CL research would thus evolve to not only update task knowledge but also to manage this certainty metric, deciding when a behavior is "mature" enough to be refactored. This also has implications for explainability (XAI), as the robot could articulate its certainty to a user (e.g., "I'm still learning this, please correct me" vs. "I've learned that you like this").

Finally, this framework would be completed by introducing **human-in-the-loop refactoring confirmation**. Making the process transparent solidifies the collaborative nature of the interaction. The robot might ask: "I have noticed that you always prefer your morning coffee black. Should I make this my standard procedure for you?" This is a dialogue-based request to approve the refactoring of a learned preference into a stable habit. This explicitly involves the user in the engineering of the robot's behavior, transforming my research from a focus on autonomous learning to a study of human-robot co-creation of reliable behaviors.

5.2 Paper 2: Continuous Experiment-driven MLOps (ExtremeXP)

Core Idea: Rajenthiram et al. propose ExtremeXP, an MLOps framework that reframes ML development as a series of structured experiments. Its key innovation is a Knowledge Repository (a Knowledge Graph) that captures a detailed history of all past experiments (data, workflows, intent) to guide and optimize future work. This brings rigor, traceability, and reusability to the often-chaotic ML development process, moving it from an "art" to an engineering discipline.

Relation to Research: The ExtremeXP framework provides a powerful conceptual model for the micro-level learning in my HRI project. The robot's long-term interaction with a user is effectively a personalized, N-of-1 experiment where the robot is the data scientist and the user is the supervisor. The robot's memory of past interactions serves as its local, implicit Knowledge Repository. My research challenge (**RQ2**) is to make this experimental process more efficient and effective.

Integration & Adaptation of Research: In the "Aether Home OS" project, ExtremeXP would be the central MLOps backbone, with my robot acting as a critical edge node that feeds real-time interaction data back to the central Knowledge Repository. This paper inspires me to adapt my research from reactive learning to proactive, structured experimentation.

First, my robot's learning mechanism could become an **active, goal-directed experimenter**. Instead of passively learning, it could form hypotheses about user preferences and design minimal, non-intrusive "tests" to confirm them. For instance, if uncertain about a preference, it could systematically try two different approaches over several days and measure implicit feedback. This requires developing safe exploration policies to ensure these experiments are never disruptive or unsafe, a key software engineering challenge. This active learning approach would make the process in **RQ1** more data-efficient.

Second, this experimental approach would be supported by an **onboard, local Knowledge Graph**. Rather than a simple replay buffer, the robot would build a structured, semantic representation of its interaction history, modeling entities like 'Stakeholder', 'Task', 'Context', and 'UserFeedback'. This would enable much richer, hybrid reasoning. For example, an engineer could query the KG to debug a failed interaction, asking "Show me all interactions with the 'Child' stakeholder in the 'Evening' context that resulted in negative feedback." This provides a level of interpretability and debuggability that is impossible with a purely black-box model.

Finally, on a longer timescale, my research could explore **meta-learning for experimentation strategies**. The robot could learn not just "what" the user prefers, but "how" that specific user prefers to be learned from. It might discover that User A responds well to direct questions, while User B prefers the robot to learn through quiet observation. The robot would thus be personalizing its own learning strategy to maximize effectiveness and minimize user friction. This would transform my project's focus from Continual Learning to a more holistic framework of "Continual Personalized Experimentation in HRI," directly embodying the core principles of the ExtremeXP framework.

6 Research Ethics & Synthesis Reflection

6.1 Search and Screening Process

- **Search Venue:** I used the provided link to the CAIN conference series proceedings.
- **Screening Method:** I scanned titles for keywords related to my research (e.g., 'continual learning', 'HRI'). I then read the abstracts to select papers with a strong AI/Software Engineering focus, not just algorithmic contributions.
- **Selection Rationale:** The COPMA paper was chosen for its design-time perspective on managing adaptable agents, while the ExtremeXP paper was chosen for its MLOps framework that treats development as a continuous experiment, providing two complementary lenses for my research.

6.2 Pitfalls and Mitigations

- **Pitfall:** Initial keyword searches often led to purely algorithmic papers irrelevant to the broader AI Engineering themes of the assignment.
- **Mitigation:** I adjusted my strategy to search for more abstract themes like 'human feedback' and 'system evolution', which yielded more suitable papers.

6.3 Ethical Considerations

- **Use of AI Tools:** No LLMs or other generative AI tools were used in the writing of this essay.
- **Ensuring Originality:** The description of my research was adapted from my previous working papers to ensure methodological precision. The analysis and synthesis in Sections 2 through 5 are entirely new and original work produced for this assignment.

References

Beer, Jenay M et al. (2012). "The domesticated robot: design guidelines for assisting older adults to age in place". In: *Proceedings of the seventh annual ACM/IEEE international conference on Human-Robot Interaction*, pp. 335–342.

Gross, Horst-Michael et al. (2015a). "Robot companion for domestic health assistance: Implementation, test and case study under everyday conditions in private apartments". In: *2015 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, pp. 5992–5999.

— (2015b). "Robot companion for domestic health assistance: Implementation, test and case study under everyday conditions in private apartments". In: *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5992–5999. DOI: 10.1109/IROS.2015.7354230.

Kirkpatrick, James et al. (2017). "Overcoming catastrophic forgetting in neural networks". In: *Proceedings of the national academy of sciences* 114.13, pp. 3521–3526.

Lesort, Timothée et al. (2020). "Continual learning for robotics: Definition, framework, learning strategies, opportunities and challenges". In: *Information fusion* 58, pp. 52–68.

Patel, Maithili and Sonia Chernova (2022). "Proactive robot assistance via spatio-temporal object modeling". In: *arXiv preprint arXiv:2211.15501*.

Silver, Daniel L (2011). "Machine lifelong learning: challenges and benefits for artificial general intelligence". In: *Artificial General Intelligence: 4th International Conference, AGI 2011, Mountain View, CA, USA, August 3-6, 2011. Proceedings 4*. Springer, pp. 370–375.

Wang, Liyuan et al. (2024). "A comprehensive survey of continual learning: theory, method and application". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Wilson, Garrett et al. (2019). "Robot-enabled support of daily activities in smart home environments". In: *Cognitive Systems Research* 54, pp. 258–272. ISSN: 1389-0417. DOI: https://doi.org/10.1016/j.cogsys.2018.10.032. URL: https://www.sciencedirect.com/science/article/pii/S1389041718302651.

Zenke, Friedemann, Ben Poole, and Surya Ganguli (June 2017). "Continual Learning Through Synaptic Intelligence". In: *Proceedings of the 34th International Conference on Machine Learning*. Ed. by Doina Precup and Yee Whye Teh. Vol. 70. Proceedings of Machine Learning Research. PMLR, pp. 3987–3995. URL: https://proceedings.mlr.press/v70/zenke17a.html.