Robotics: Science and Systems 2025
Los Angeles, CA, USA, June 21–25, 2025

FEAST: A Flexible Mealtime-Assistance System Towards In-the-Wild Personalization

Rajat Kumar Jenamani[1], Tom Silver[1], Ben Dodson[1], Shiqin Tong[1], Anthony Song[1], Yuting Yang[2], Ziang Liu[1], Benjamin Howe[3], Aimee Whitneck[3], Tapomayukh Bhattacharjee[1]
[1]Cornell University, [2]University of Michigan, [3]Independent researcher

Fig. 1: Informed by a formative study with 19 care recipients and 2 community researchers, we propose FEAST: a flexible mealtime-assistance system grounded in adaptability, transparency, and safety. Through in-home evaluations, we demonstrate that FEAST can personalize to the needs of two care recipients, assisting each with three meals in diverse in-the-wild contexts.

***Abstract*—Physical caregiving robots hold promise for improving the quality of life of millions worldwide who require assistance with feeding. However, in-home meal assistance remains challenging due to the diversity of activities (e.g., eating, drinking, mouth wiping), contexts (e.g., socializing, watching TV), food items, and user preferences that arise during deployment. In this work, we propose FEAST, a flexible mealtime-assistance system that can be personalized in-the-wild to meet the unique needs of individual care recipients. Developed in collaboration with two community researchers and informed by a formative study with a diverse group of care recipients, our system is guided by three key tenets for in-the-wild personalization: adaptability, transparency, and safety. FEAST embodies these principles through: (i) modular hardware that enables switching between assisted feeding, drinking, and mouth-wiping, (ii) diverse interaction methods, including a web interface, head gestures, and physical buttons, to accommodate diverse functional abilities and preferences, and (iii) parameterized behavior trees that can be safely and transparently adapted using a large language model. We evaluate our system based on the personalization requirements identified in our formative study, demonstrating that FEAST offers a wide range of transparent and safe adaptations and outperforms a state-of-the-art baseline limited to fixed customizations. To demonstrate real-world applicability, we conduct an in-home user study with two care recipients (who are community researchers), feeding them three meals each across three diverse scenarios. We further assess FEAST's ecological validity by evaluating with an Occupational Therapist previously unfamiliar with the system. In all cases, users successfully personalize FEAST to meet their individual needs and preferences. Supplementary materials and videos can be found at: emprise.cs.cornell.edu/feast.**

***Index Terms*—Assistive, Entertainment and Service Robots, Human-Robot Interaction, Robot Learning: Foundation Models**

I. Introduction

Eating is a fundamental part of human life, deeply intertwined with identity and social interaction [1]. The inability to self-feed has been associated with profound emotional impacts, including feelings of shame, diminished self-esteem, and heightened anxiety or fear [2–4]. Unfortunately, millions worldwide require assistance with feeding due to spinal cord injuries, strokes, cerebral palsy, old age, and other health conditions [5]. For caregivers, feeding is one of the most time-consuming Activities of Daily Living (ADLs) [6], contributing significantly to their already substantial workload [7, 8].

Robot mealtime-assistance systems have the potential to assist care recipients and improve their quality of life [9] while decreasing the physical workload on caregivers [10, 11]. Recent advancements have significantly improved various aspects of mealtime assistance, including food manipulation [12–19], skill sequencing [20–22], and bite transfer [11, 23–26]. Although further progress is needed before these systems can operate long-term without expert supervision, these recent works significantly enhance their robustness and autonomy.

Building upon these efforts, this work considers the need for personalization in mealtime assistance. A one-size-fits-all mealtime assistance system is not enough to address the wide range of preferences, functional abilities, behaviors, and environmental contexts that vary between care recipients and between meals [27]. For example, consider the two community researchers (CRs) shown in Figure 1, who are both co-authors on this paper. CR1, who has Multiple Sclerosis, prefers to lean forward to take a bite. CR2, who has a C4-C6 Spinal Cord Injury and very limited head and neck mobility, requires inside-mouth bite transfer [11]. In a social dining scenario, CR1 prefers that the robot retract after bite transfer so they can better see their companion. In another social scenario, CR2 prefers to control the robot using a custom *long-continuous-open-mouth* gesture that they create themselves through the user interface (a standard open-mouth gesture would be falsely triggered in conversation). These are a few examples among many underscoring the need for a mealtime assistance system that can be personalized in-the-wild by the users themselves.

To better understand the nature of personalization in mealtime assistance, we start our work with a formative user study (Section III). We use speculative videos to guide in-depth conversations with 21 care recipients who have diverse medical conditions and mobility limitations. This study has two outcomes. The first is a collection of specific personalization requests that we use to develop our system (Section V-A). The second is the realization of three key tenets crucial for personalization in mealtime assistance: adaptability, transparency, and safety (Section V). Care recipients voice the need for systems that can adapt to their needs and preferences—not just once, but from meal to meal, and over time as their needs change. They also express that any adaptations should be transparent so that they are able to understand and predict system behavior. Finally, they underscore the importance of safety, especially when the system changes.

With these key tenets in mind, we propose FEAST, a flexible mealtime-assistance system towards in-the-wild personalization (Section IV). FEAST is designed for personalization at both the hardware and software level. The hardware features a modular tool-change apparatus so that a single robot arm can assist with feeding, drinking, and mouth wiping. The system also features accessible buttons, status LEDs, cameras, microphones, and speakers to enable a wide range of customizable interactions with the user. We also propose a novel feeding utensil that increases the robot's workspace and decreases obstructions to the user's view.

On the software side, to strike a balance between adaptability, transparency, and safety, we propose to sequence together parameterized behavior-tree-based skills to achieve user-specified goals. The parameterized behavior trees provide a mechanism for personalization: user requests, formulated in natural language, are translated with a large language model (LLM) into structured updates to the behavior trees, which can then be statically validated for safety. This structured skill representation can also be analyzed by an LLM in response to user transparency requests. Finally, using the code synthesis capabilities of LLMs, we enable users to create their own custom head gestures that can be added to the behavior trees and used to interact with the robot. Users engage with the robot through a flexible web-based user interface.

We develop FEAST using community-based participatory research [28] in collaboration with two CRs (Figure 1). This approach, involving co-design and in-depth evaluation with one or two CRs, is well-established in assistive technology research [29–31] and increasingly common in assistive robotics [1, 32–36]. Our collaborations began with introductory video calls in November 2022 and have continued through regular meetings—both virtual and in-person at the CRs' homes. These sessions have been instrumental in gathering feedback on system design, brainstorming studies to identify personalization needs (Section III), and piloting developments.

This iterative process led to a five-day in-home evaluation in January 2025 (Section VI), where CRs fed themselves six meals across three distinct contexts: personal, watching TV, and social (Figure 1). Results show that CRs successfully completed meals with few researcher interventions and personalized the system on the fly to their preferences and abilities while reporting low cognitive workload, as indicated by NASA-TLX surveys [37]. CRs also rated FEAST highly for real-world applicability, as reflected in Technology Acceptance Model [38] survey results. Furthermore, both CRs reported that FEAST provided greater control over their meals and a stronger sense of independence compared to their human caregiver, with one noting that conveying preferences was easier with our system. Finally, to further assess ecological validity, we evaluated FEAST with an Occupational Therapist unfamiliar with the system, who confirms its merits over a no-personalization baseline (Section VII).

Overall, our contributions include:

- FEAST: A flexible **mealtime-assistance system** built with **community-based participatory research**, tackling in-the-wild scenarios by integrating diverse skills, custom tools, a flexible interface, and user personalization.
- A user study involving 21 care recipients, which identifies diverse **personalization needs** for mealtime assistance.
- A **personalization framework** built on three key tenets: (i) adaptability via LLM-based code synthesis, (ii) trans-

TABLE I: Comparison between FEAST and other mealtime assistance systems. Adaptability is assessed based on ability to handle open-ended user requests, safety by self-reported adherence to ISO 13482 [39], and transparency by Levels 1-5 of the IEEE Transparency Standard [40].

System	Adaptability	Safety	Transparency	Interface	Autonomous Bite Acquisition			Autonomous Bite Transfer		Minimal	Workspace	Other
					Seq.	Pre-acq.	Acq.	Outside	Inside	Occlusion	Reachability	Tasks
Obi [41]	-	✓	-	✓	-	-	-	-	-	✓	-	-
Neater Eater [42]	-	✓	-	✓	-	-	✓	-	-	✓	-	-
Park et al. [25]	-	-	-	✓	-	-	✓	✓	✓	-	✓	-
Bhattacharjee et al. [10]	-	-	-	✓	-	-	✓	-	✓	-	-	-
Feel the Bite [11]	-	-	-	-	-	-	-	-	✓	-	-	-
FLAIR [20]	-	-	-	-	✓	✓	✓	✓	✓	-	-	-
REPEAT [21]	-	-	-	-	-	✓	✓	-	-	-	-	-
Nanavati et al. [36]	-	✓	-	✓	-	-	✓	✓	-	-	-	-
FEAST (ours)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

● for commercial systems, ● for academic systems

parency (Levels 1-5) through LLM summarization, and (iii) safety aligned with ISO/TS 13482 safety principles.

- A **five-day in-the-wild system evaluation** with two community researchers, spanning six realistic meals across three distinct environmental contexts, showcasing FEAST's real-world applicability.
- An **evaluation with an Occupational Therapist** unfamiliar with our system, assessing ecological validity and demonstrating improved performance over a non-personalized baseline.

II. Related Work

A. Mealtime Assistance

Research on mealtime-assistance systems dates back to the 1970s, with early examples such as the Morewood Spoon Lifter [43]. Over the years, several commercial systems have been proposed, such as Winsford Feeder, My Spoon, Neater Eater, Bestic Arm, Meal Buddy, and Obi [44]. However, their reliance on manually programmed, fixed food acquisition and transfer actions has limited user adoption, and as of now, only Obi and Neater Eater remain commercially available. To address these limitations, recent research focuses on using sensors, such as in-hand cameras, to perceive the environment, and plan and execute autonomous motions.

Bite Acquisition and Transfer. Recent works have explored autonomous strategies for skewering solid bite-sized foods [12–15, 45], scooping soft food items [16–18], twirling and grouping noodle-like dishes [19], and cutting [46]. Building on these works, FLAIR [20] introduces a bite acquisition framework consisting of a library of vision-parameterized food manipulation skills that use a fork-based utensil. This framework leverages the commonsense reasoning and few-shot learning capabilities of foundation models to appropriately sequence these skills, enabling the feeding of complete meals while adhering to bite ordering preferences of the user. Various works also propose autonomous bite transfer methods [11, 23–26, 47], which can be broadly categorized into two types. Outside-mouth bite transfer methods [23, 24, 47] bring food close to the care recipient's mouth, requiring them to lean forward to take a bite. For individuals with severe mobility limitations who cannot lean forward, inside-mouth bite transfer methods [11, 25, 26] place food directly inside their mouth.

Our system builds on FLAIR [20] for bite acquisition and Gallenberger et al. [23] and Feel the Bite [11] for bite transfer, enabling it to feed realistic dishes to users with severe mobility limitations. However, recent studies show that feeding robots often exhibit obtrusive motion [1, 24], limiting their use in social settings, and have restricted workspace [36], constraining plate and user placement. These limitations hinder personalization, as robots must adapt to social contexts with non-obtrusive motion and allow flexible user and plate positioning. Towards addressing these challenges, and based on feedback from community researchers, FEAST adapts these approaches to a novel feeding utensil (Section IV-A) that improves the robot's workspace and decreases obstruction.

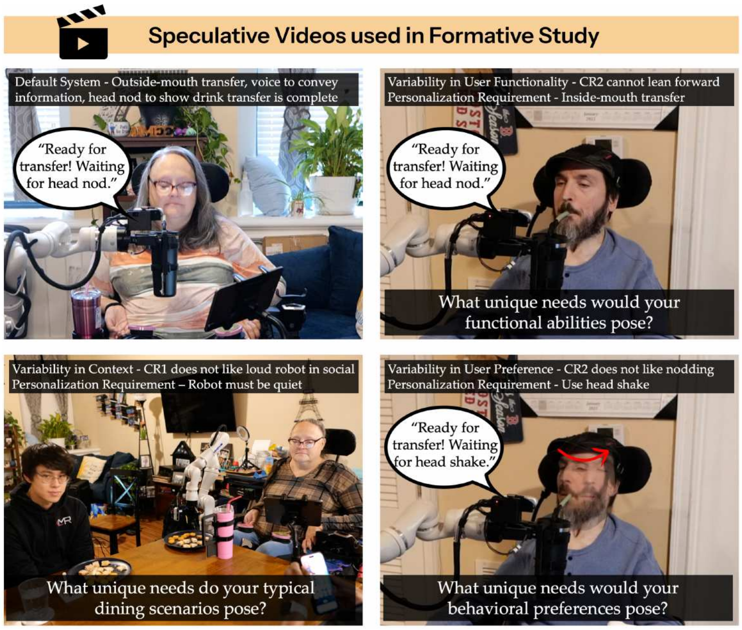
Other Essential Mealtime Tasks. Several works have explored assistance with drinking [48, 49], focusing on grasping a cup and having users drink from the rim. We instead focus on drinking from a cup with a straw, aligning with our community researchers' routines. Unlike prior methods, which assume specific cup colors [48] or pre-grasped cups [49], FEAST emphasizes real-world flexibility, using an adaptable handle that fits various cups without strict shape or color constraints. Some works also propose methods for mouth wiping [50, 51]. Similar to these, FEAST can pick up a custom wiping tool and position it near the user's mouth, ready for wiping.

Mealtime-Assistance Systems. Most relevant to our work are autonomous systems that demonstrate integration of various components for feeding a complete meal [20, 23, 25, 36] (see Table I). Unlike Park et al. [25], which focuses on feeding yogurt, and Bhattacharjee et al. [23] and Nanavati et al. [36], which limit bite acquisition to skewering, our system employs a range of skills—skewering, scooping, twirling, grouping, and more—to pick up diverse food items and sequence them over a long horizon for feeding an entire meal. Our system differs from FLAIR [20] (which integrates with Feel the Bite [11]) by adapting it to a novel feeding utensil and a web interface for user interaction. More significantly, our system is the first to integrate feeding with other essential mealtime tasks, including drinking and mouth wiping, while also automating user personalization. No prior system has explored feeding in diverse in-the-wild scenarios—except Nanavati et al. [36], which offers customization of limited system parameters.

B. Personalization in Assistive Robotics

Various formative studies in assistive robotics emphasize the importance of user personalization based on factors such as mobility limitations, behavior, and context [1, 52–54].

One approach to achieving personalization is through implicit adaptation, where limited test-time examples guide the



Demographics of Study Participants

ID	Age	Gender	Impairment	Onset Time
P1	46	Male	Arthrogryposis	Since birth
P2	32	Male	Arthrogryposis	Since birth
P3	28	Female	Schizencephaly	Since birth
P4	32	Male	Spinal Muscular Atrophy	Since birth
P5	43	Male	Spinal Cord Injury C2	Since birth
P6	43	Male	Cerebral Palsy	Since birth
P7	20	Male	Cerebral Palsy	Since birth
P8	51	Male	Multiple Sclerosis	Since 30 years
P9	28	Female	Muscular Dystrophy	Since birth
P10	30	Female	Fibromatosis Scoliosis	Since birth
P11	29	Female	Quadriplegic	Since 19 years
P12	42	Female	Spinal Muscular Atrophy	Since birth
P13	62	Female	Spinal Cord Injury C5-C6	Since 21 years
P14	50	Female	Charcot-Marie-Tooth	Since birth
P15	43	Male	Spinal Cord Injury C4	Since 9 years
P16	28	Male	Spinal Muscular Atrophy	Since birth
P17	42	Male	Spinal Muscular Atrophy	Since birth
P18	46	Female	Multiple Sclerosis	27 years
P19	34	Male	Cerebral Palsy	Since birth
CR1	47	Female	Multiple Sclerosis	Since 17 years
CR2	46	Male	Spinal Cord Injury C4-C6	Since 27 years

Fig. 2: We use speculative videos recorded with community researchers (left) to conduct a formative study with 19 care recipients and the 2 community researchers (demographics shown on the right), to identify user personalization requirements.

system to a specific preference [55–59]. FEAST aligns with this by generating personalized gesture detectors using a small number of examples from the user. Unlike prior black-box approaches to gesture personalization [60, 61], FEAST takes a more transparent approach by composing functions from our perception library to generate customized detectors. While some studies evaluate gesture detection for care recipients [62, 63], FEAST is the first to consider personalized gesture detection for those with mobility limitations.

Another approach involves explicitly identifying preferences through direct user input. Canal et al. [64] use post-hoc scoring for task planning, but this limits applicability in safety-critical tasks. Nanavati et al. [36] provide predefined customizations for a mealtime-assistance robot through a web interface, while FLAIR [20] uniquely explores bite-ordering preferences using language commands. FEAST leverages FLAIR for bite-ordering preferences and significantly extends personalization to other system components by enabling open-ended customization through language inputs.

C. *Large Language Models for Human-Robot Interaction*

Recent advancements in large language models (LLMs) [65, 66] enable the use of language as an interface for human-robot interaction. Some approaches utilize LLMs for high-level planning, allowing systems to generate plans over a fixed library of skills based on user language commands [67–73]. Others focus on generating low-level skills by training on large vision-language-action datasets [74–77], synthesizing rewards [78, 79], or inferring skills via code generation [80, 81]. FEAST represents a recent category of approaches that enable users to edit both high-level plans and parameters of low-level skills using LLMs [82–84]. However, given the physical nature of human-robot interactions in mealtime assistance, FEAST restricts edits to carefully selected operations to ensure safety.

III. Identifying Personalization Needs from Care Recipients with Diverse Medical Conditions

To guide the development of our flexible mealtime-assistance system, we conducted a formative user study with 21 care recipients (including 2 CRs) with diverse mobility limitations. Our objectives in this study were two-fold: (1) to collect specific personalization requests; and (2) to identify key tenets of personalization in the context of mealtime assistance.

Participants. We recruited 19 care recipients with diverse medical conditions such as Cerebral Palsy, Multiple Sclerosis, Muscular Dystrophy, Spinal Cord Injury, etc. (Figure 2). There were 8 female and 11 male participants who ranged in age from 20 to 62 years old and who were located throughout the United States. The participants were also diverse in terms of their prior experience with mealtime-assistance systems: some had no experience; others had participated in academic studies; and some had extensive experience (e.g., P6 noted *"I had a Winsford Self-Feeder for over 20 years until the second one I had stopped working 4 months ago"*).

Design Materials. Previous work [27] has shown that caregiving scenarios vary depending upon (i) care recipient functionality, (ii) care recipient behavior, and (iii) environment. Working with our community researchers, we identified realistic mealtime assistance scenarios where each of these three components vary. We then recorded speculative videos of the scenarios using our robot mealtime-assistance system (see Section IV for system details). The videos were organized into a slideshow and supplemented with conversation prompts (Figure 2). For example, one video showed the robot successfully feeding a care recipient with outside-mouth bite transfer; the next video showed the robot failing to feed a different care recipient who has very limited neck mobility; a final video in the sequence showed the robot successfully feeding the latter care recipient with inside-mouth bite transfer. After these three

Fig. 3: We extract and categorize personalization requests from our formative study with 21 care recipients. See Appendix A for details.

videos, we posed the question: "What unique needs might your own functional abilities pose for the robot?" Other speculative videos preceded discussion questions including: "What unique needs might your own behavioral preferences pose for the robot?" and "In what scenarios do you typically eat? What unique needs might these scenarios pose for the robot?"

Study Design. To broaden participation in the study, we conducted conversations virtually. Caregivers were also invited to join and actively contribute to the discussion. Each session consisted of the following steps: (1) introductions and agenda; (2) a brief review of recent progress in robot mealtime-assistance systems and an extension of gratitude to the study participants for facilitating this progress; (3) speculative videos and discussion questions; (4) final open-ended discussion and conclusion. Care recipients were compensated $30 USD for their participation in the 90-minute study.

Outcomes. This formative study successfully accomplished its two objectives. First, we collected specific personalization requests, which we group together into various categories and visualize in Figure 3. We detail each of these requests in Appendix A. These diverse requests highlight the need for personalization in a mealtime-assistance system and suggest where we should focus our efforts to meet the needs of users.

Our second objective was to identify key tenets for personalization. From our conversations, we identified three themes: adaptability, transparency, and safety.

1) Adaptability: The first clear tenet of personalization is adaptability: changing system behavior in response to user requests. Study participants highlighted the need for adaptability beyond one-time system setup. For example, they requested the ability to customize the system depending on the feeding scenario—when watching TV, controlling the robot with a button may be preferable to verbal commands; or when dining socially, the robot should retract to a resting position immediately after bite transfer. Study participants also described how their preferred interaction with the robot may change throughout the day, depending on their energy level, and over time, as their medical conditions change.
2) Transparency: Study participants also emphasized the importance of being able to understand the behavior of the robot, especially if that behavior may change. For example, describing their ideal relationship with the robot, P8 said: *"It would be a symbiotic relationship; I would like to understand the robot capabilities and the robot would understand my requirements."*
3) Safety: A final key tenet of personalization that emerged from the formative study was safety. For example, P12 said: *"Safety is the most important part. If I turn my head to look somewhere the robot should not try to feed me at that position which might be unsafe."*

These specific examples and key tenets informed the design of our mealtime-assistance system, which we describe next.

IV. FEAST: A Flexible Mealtime-Assistance System Towards In-the-Wild Personalization

In this section, we present FEAST, a mealtime-assistance system that enables users to personalize to in-the-wild eating scenarios commonly encountered in real-world settings. All hardware and software components of FEAST are open-sourced on our website. In the following subsections, we describe our system hardware (Section IV-A), software (Section IV-B), and user interface (Section IV-C), explaining how each component can be personalized while adhering to the tenets of adaptability, transparency, and safety.

A. System Hardware

FEAST (see Figure 4) uses a Kinova Gen3 7-DoF robot arm [85] and a Robotiq 2F-85 gripper [86]. It can be flexibly mounted either on the user's ROVI wheelchair [87], powered by the wheelchair's battery, or on a movable Vention stand [88], powered by a wall outlet.

Tool-Change Apparatus. FEAST employs three custom tools. First, a novel feeding utensil with integrated motors provides wrist-like degrees of freedom, enabling tasks such as twirling, scooping, and maintaining an upright orientation when holding food. The utensil's fork, made of compliant silicone, is connected to a 6-axis Nano25 ATI force/torque sensor [89]. This default fork is detachable, allowing users to exchange it with a metal fork if desired. To power and control the utensil without dangling wires, the robot's gripper fingertips are replaced with custom fingertips featuring magnetic electrical connections that engage with complementary connectors on the utensil when grasped. Second, for drinking, FEAST uses an adaptable mug handle inspired by adaptable mug holders [90], which accommodates various cup shapes without strict dimensional constraints and features an ArUco marker [91] for autonomous grasping. Third, for mouth wiping, a custom tool with a removable microfiber cloth provides gentle cleaning. Each tool is mounted and dismounted by the robot opening and closing its finger tips (see Figure 4).

Novel Utensil Orientation and Camera Mount. Previous state-of-the-art feeding systems use forward-facing in-hand cameras and similarly oriented utensils [11, 20, 23, 36]. However, this design restricts the robot's workspace, requires large movements to transition between acquisition and transfer, and can obstruct the user's view during feeding. Based on feedback from CRs, FEAST introduces a simple and effective solution: positioning tools at a perpendicular angle, mimicking natural human wrist movements during eating. This change, however, requires the camera to align accordingly. To address this, FEAST employs a custom camera mount with an RGB-D Intel RealSense Camera [92], oriented perpendicularly away.

Compute and Networking. FEAST employs custom controllers, such as a task-space compliant controller operating at approximately 1 kHz. This necessitates a dedicated real-time control system, for which we use an Intel NUC [93]. The primary computing platform is a Lenovo Legion Pro 7i laptop equipped with a 16GB RTX 4090 GPU [94]. Communication between the main compute unit, the NUC, the robot, and the web interface is managed via a Nighthawk RAX43 router [95]. When FEAST is mounted on a wheelchair, these components can also be securely mounted to ensure they move along.

Accessible Buttons. FEAST utilizes three accessible buttons [96]. Two are intended for the user: one customizable through our personalization pipeline for interactions, and another serving as an emergency stop. The third button functions as an experimenter emergency stop.

Status LED. FEAST features a status LED within the camera mount to alert users when their attention is needed, based on feedback from CRs. This is especially useful for users multitasking, such as watching TV while eating.

B. System Software

FEAST sequences together skills to accomplish goals given by the user. For example, if the user requests a drink while the robot is holding the utensil, the robot would invoke skills to (1) dismount the utensil; (2) mount the drink; and (3) transfer the drink to the user. We next describe how skills are generally implemented, adapted, and sequenced together, before detailing the specific skills used in this work.

Skills as Parameterized Behavior Trees. We implement skills as behavior trees [97]. As an extension of standard behavior trees, we expose node parameters that can be adapted in response to user requests (see below). For example, the behavior tree for bite acquisition includes three parameters: `Speed`, `TimeToWaitBeforeAutocontinue`, and `AskUserForConfirmation`. Every parameter is associated with a domain of possible values. For example:

- `Speed` $\in \{\texttt{low}, \texttt{medium}, \texttt{high}\}$
- `TimeToWaitBeforeAutocontinue` $\in [5, 100]$
- `AskUserForConfirmation` $\in \{\texttt{true}, \texttt{false}\}$.

All nodes and parameters are given human-readable names and descriptions to facilitate LLM-based adaptations.

Personalizing Skills from Natural Language. FEAST users can make personalization requests through spoken or typed natural language. Our pipeline for processing these requests is as follows:

1) The natural language request is converted into structured behavior tree updates using an LLM. (adaptability)
2) Each potential update is checked for safety. If the updates are deemed safe, the behavior trees are updated. Otherwise, a failure is reported. (safety)
3) The updates and outcomes are briefly summarized with an LLM and reported back to the user. (transparency)

We now describe these steps in more detail.

1) Language $\rightarrow$ Structured Updates. We use an LLM (GPT-4o [98]) to translate natural language personalization requests into structured behavior tree updates. The LLM is prompted with a brief explanation of the scenario, the personalization request, the behavior trees, and the form of an expected response.[1] The output of the LLM is statically checked to ensure that the structured requests are in the expected form and the names of behavior tree nodes and parameters are valid. If any of these checks fail, the LLM is automatically re-prompted (up to 3 times in experiments) with failure information. It is important to note that one personalization request can elicit multiple updates. For example, if the user requests for the robot to "always move as fast as possible," multiple behavior trees with speed parameters would be updated.

2) Safety Checks. If the structured behavior tree requests pass static checks, we pass them through another round of safety checks. We consider two types of updates: node additions and parameter changes. In this work, to guarantee safety, we permit only three types of node additions: "pause", "wait for gesture", and "retract." We further restrict "retract" node additions to ensure that the robot only retracts from a limited set of configurations that have been empirically tested. For parameter changes, we ensure that the proposed values lie within the parameter domains. These domains are carefully

[1]We also found that prompting the LLM to rephrase the original request into a "specific setting that should be changed in the robot's software" helped with certain requests such as "The robot is too slow right now."

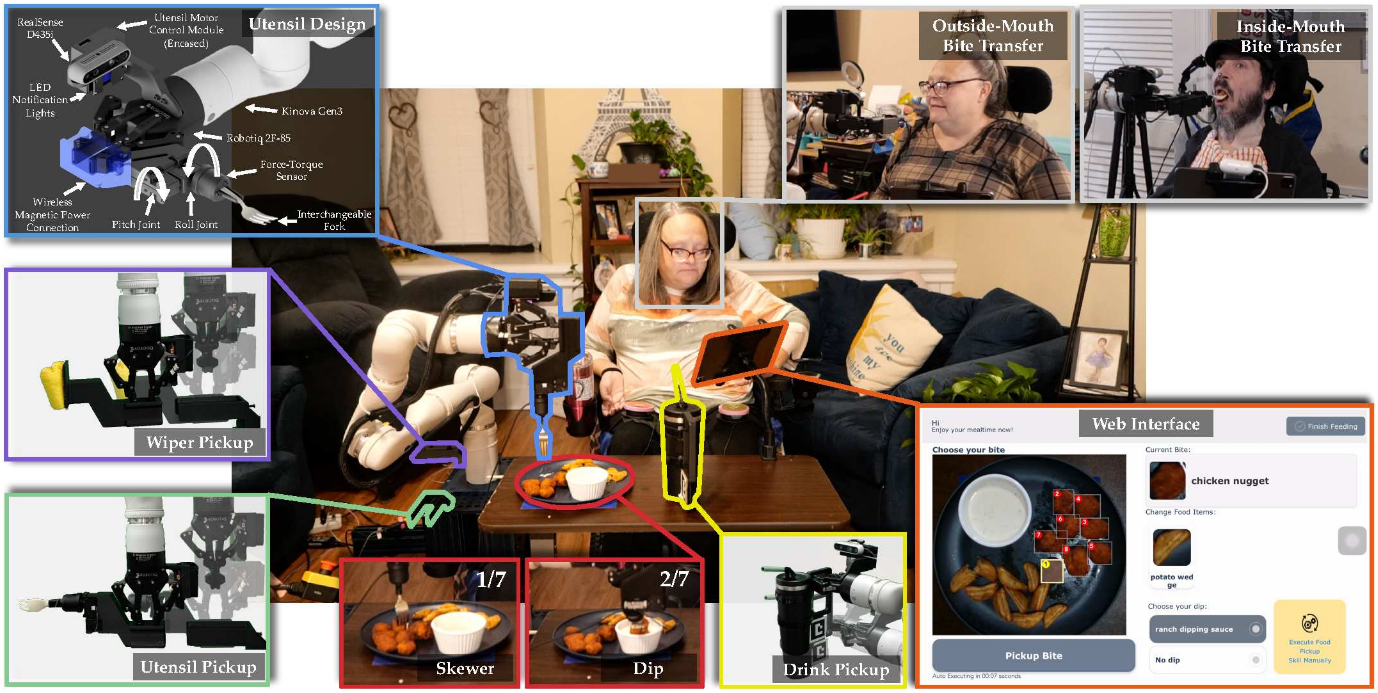


Fig. 4: FEAST features diverse mealtime-assistance skills such as feeding, drinking and mouth wiping, custom tools, and a flexible web interface, all of which can be personalized in-the-wild by end users.

chosen to ensure that safety is guaranteed, while also giving the user enough flexibility to meet their personalization needs.

3) Update Results → Language. Given the history of structured behavior tree updates and the check outcomes, we again use an LLM to generate a brief summary. The LLM is prompted with a brief explanation of the context, the original user request, the structured updates, and the outcomes, and asked to generate a short summary that a non-technical end user would understand. This summary is displayed on the user interface (Section IV-C).

This process—converting natural language requests into structured behavior tree updates that can be checked for safety and summarized back to the user—addresses our three key tenets of personalization. By exposing parameters and allowing for certain node additions, the system is adaptable; in summarizing the process, the system is transparent; and by restricting and checking the adaptations, the system is safe.

Synthesizing New Gestures with LLMs. As an additional step towards open-ended personalization, we allow the user to synthesize new gestures that can then be integrated into the behavior tree—for example, if the user wishes to use a custom head shaking motion to indicate that they are ready for bite transfer. Through the interface (Section IV-C), the user is guided through the process of collecting a few positive and negative examples of their new gesture. The user also gives the gesture a name (e.g., "head shake") and a brief description (e.g., "shaking my head from right to left"). FEAST then uses LLM-based program synthesis to generate a programmatic gesture detector that takes a head tracking module as input. Our program synthesis pipeline is also able to propose and optimize hyperparameters, e.g., `NumberHeadShakes` or `NoiseTolerance`. See Appendix B for details.

Sequencing Skills with PDDL. Towards generating a sequence of skills to achieve a user's goal, we associate each skill with a PDDL operator [99]. For example:

```
(:action PickTool
  :parameters (?tool - tool)
  :precondition (and (GripperFree) (Reachable ?tool))
  :effect (and (Holding ?tool) (not (GripperFree))
               (not (Reachable ?tool)))
)
```

where `?tool` can be grounded with `utensil`, `mug`, or `wiper`. See McDermott et al. [99] for a formal introduction to PDDL. We use an optimal task planner (FastDownward [100] with alias `seq-opt-lmcut`) to sequence the skills together given a known initial propositional state and a propositional goal derived from the user's input. For example, when the user requests a bite, FastDownward generates a three-step plan: `PickTool(utensil)`, `AcquireBite(utensil)`, `TransferTool(utensil)`. The plan is executed open-loop and the propositional state is updated using the operator effects. If the user subsequently requests a drink after requesting a bite, the second plan would be: `PlaceTool(utensil)`, `PickTool(mug)`, `TransferTool(mug)`.

Skills Included in FEAST. We now detail the specific skills that are currently implemented in FEAST.

- `PickTool`: Grasping routine for *utensil*, *mug*, or *wiper*. The utensil and wiper are placed in a fixed tool mount relative to the robot's base, enabling a predefined pickup motion. For the mug, the robot detects the ArUco marker on the adaptable handle and executes an appropriate grasp.
- `PlaceTool`: Returns the tool to its last known location. For utensils or wipers, this is a predefined position on the tool mount; for the mug, it is the original pickup position.
- `AcquireBite`: Picks up food from the plate. FEAST

builds on the bite acquisition framework proposed by FLAIR [20], adapting it to our novel utensil. This framework includes a vision-parameterized food manipulation skill library with four pickup actions—skewering, scooping, twirling, and dipping—and three pre-acquisition actions—pushing, grouping, and cutting—to prepare for future pickups. We incorporate force thresholds via the force-torque sensor to ensure safe operation. For each new bite, FEAST processes the plate image using vision-language foundation models to identify plate contents and predict the required skill sequence for each food type. These predictions, along with the user's bite order preferences and bite history, are provided to an LLM, which determines the next bite while balancing preference adherence and efficiency. The robot then executes the predicted skills to pickup the bite.
- `TransferTool`: By default, FEAST performs outside-mouth transfer [23], where it detects the user's head pose once and moves to a predefined distance outside the mouth. Upon user request, FEAST can switch to inside-mouth transfer [11], using a task-space compliant controller to continuously track the user's head pose and bring the tool inside the mouth. After detecting that the user has finished taking a bite, sip, or completed mouth wiping, it returns to its initial position. Both transfer methods utilize the head perception pipeline proposed by Feel the Bite [11].
- `EmulateTransfer`: Moves the gripper to a configuration just in-front of the user's mouth. Useful for recording and testing new gestures as we detail in the next section.
- `Retract`: Moves the robot to the retract configuration with an empty gripper.

C. User Interface

FEAST features a web-based interface accessible on personal devices, such as tablets or phones, enabling integration with assistive technologies that care recipients already use. For example, CR2 uses a tracker that follows a reflective dot on their nose [101] (Figure 4). We refine the interface design through multiple iterations with CRs to address their needs; they highlight that the interface should have big buttons that are easy to click, and should support speech to text.

We implement the interface using Vue.js [102], an open-source front-end JavaScript framework, while ROS Noetic [103] facilitates communication with the mealtime-assistance skills. FEAST's web interface displays pages on demand from the robot, automatically updating to reflect system behavior as users personalize it to meet their evolving needs. The interface supports system adaptability and transparency via a personalization webpage, as we detail in Section IV-D.

Beyond fully autonomous workflows that can continuously feed users without intervention, FEAST's interface allows on-demand user control. Users can override the robot's autonomous predictions, for instance, by manually selecting the next bite or pinpointing key points on a displayed plate image as parameters to bite acquisition skills. Such overrides help users recover from autonomous errors, increasing the robustness of the system in in-the-wild scenarios [104, 105].

D. Default Mealtime-Assistance Procedure

Mealtime Assistance. At the start of a new meal, the robot begins in the "retract configuration" without holding any tool. The user is presented with the New Meal Page on the web interface, where they specify the food items on the plate and their preferred bite order (e.g., "Feed me all of X first, then Y"). The interface then transitions to the Task Selection Page, allowing the user to choose from three primary actions: requesting a bite, taking a sip, or wiping their mouth, which sends `TransferTool(utensil)`, `TransferTool(mug)`, and `TransferTool(wipe)` commands respectively. If the robot is not holding the appropriate tool, the task planner directs it to place the current tool and pick up the correct one. Moreover, for a bite, the task planner invokes AcquireBite, which looks at the plate contents and predicts the next bite in adherence to the user's ordering preferences. The interface subsequently opens a Bite Acquisition Page, where the user can:

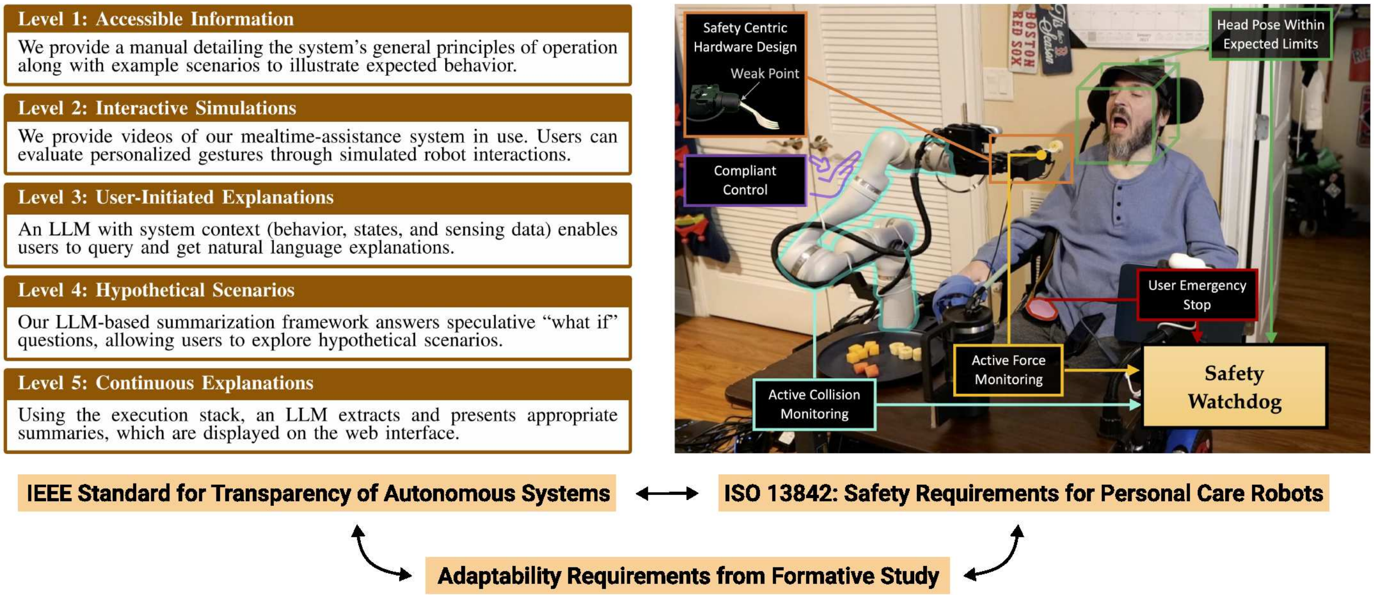
1) Switch to a different bite of the same or another food type.
2) Switch to a Manual Bite Acquisition Page to choose a specific skill (e.g., skewering) and set keypoint parameters (e.g., the exact skewering point on the plate image).

If the user clicks "acquire bite" or does not press any button within 10 seconds, the robot proceeds with bite acquisition. For `TransferTool`, the interface prompts the user to confirm readiness for transfer. Once confirmed, the robot moves to a ready-to-transfer configuration and announces, "Please open your mouth when ready." It waits for the user to open their mouth and then, moves the fork, straw, or wiping tool tip near the user's mouth, saying, "Ready for transfer." After detecting a bite via the force/torque sensor, or a completed sip or wipe through a head nod, the robot retracts to the ready-to-transfer position. It then returns to the Next Task Selection Page, with a 10-second auto-continue countdown to repeat the last task if it was a bite or sip. At any point during the meal, the user can click a "finish eating" button that triggers the Retract skill.

Personalization. Before and during a meal, users can access the Personalization Page, where they can send:

1) Adaptability Requests: Users can input commands to personalize components of the mealtime-assistance system to meet their needs (details in Section V-A).
2) Transparency Requests: Users can ask about the robot's behavior, and the system responds through textual answers (details in Section V-B).

The page also allows users to transition to the Gestures Page, which focuses on adding personalized gestures to the system's library. To add a new gesture, users provide a language label and description. The interface then triggers `EmulateTransfer`, moving the robot to a position just outside the user's mouth without holding a utensil to record positive and negative gesture examples. Once the recordings are complete, the robot returns to the ready-to-transfer configuration, and the interface displays the accuracy of the newly trained gesture detector. Users can also test gesture detectors by having the robot move to the position just outside their mouth (without holding a utensil) while the interface displays whether the detector



	Mealtime Tasks								Interaction											Hardware				Acquisition								Transfer						Miscellaneous								
	Feeding	Drinking	Static Wiping	Dynamic Wiping	Manual Task Selection	Auto-Execute Last Task	Auto-Execute Wiping	Meal Heating	Web Interface	Gesture-Based Control	Personalized Gestures	Button-Based Control	Personalized Clicks	Eye Gaze	Voice	Web Interface	Voice Feedback	Audible Beeps	LED Indicator	User-Preferable Tools	Auto Tool Change	Detachable Tool Tips	Flexible Mounting	Bite Prediction	Auto-Continue Bite	Manual Bite Selection	Diverse Fork-Based Skills	Finger Food Acq. Skills	Bite Size	Skewer Axis Selection	Confirmation After Acq.	Outside-Mouth Distance	Inside-Mouth Placement	Convey Ready to Initiate	Detect User Ready	Convey User can Bite	Detect Complete	Speed Control	Non-Obtrusive	Social-Aware Motion	Retract From Configs	Robot Voice Prompts	Preference Learning	Adjust Fixed Configs	User-Robot-Plate Setup	Scene Selection
FEAST (ours)	✓	✓	✓	✗	✓	✓	✗	✗	✓	✓	✓	✓	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓	✗	✗	✓	✗
Nanavati et al. [30]	✓	✗	✗	✗	✗	✗	✗	✗	✓	✓	✗	✗	✗	✗	✓	✓	✓	✗	✗	✗	✗	✗	✓	✗	✗	✓	✗	✗	✗	✗	✓	✓	✗	✓	✓	✓	✗	✓	✗	✗	✗	✗	✗	✓	✓	✓
FLAIR [20]	✓	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗	✓	✗	✗	✓	✗	✓	✓	✓	✓	✗	✓	✗	✗	✗	✗	✗	✓	✓	✓	✓	✓	✗	✗	✗	✗	✗	✗	✗	✓	✗

Fig. 5: FEAST proposes a personalization framework for mealtime assistance built on three key tenets: (i) adaptability towards tackling the various request types identified through our formative study, (ii) transparency towards Levels 1-5 on the IEEE Transparency Standard, and (iii) safety aligned with ISO/TS 13482 safety principles.

is triggered by their movements. When switching between webpages, the interface displays a white screen with dynamic text, continuously explaining its current actions. Together, these personalization features enhance adaptability and align with the IEEE Transparency framework (Section V-B).

V. FEAST Achieves the Three Tenets of Personalization

Our formative study affirmed the importance of adaptability, transparency, and safety for a mealtime assistance system that is personalized in-the-wild. In this section, we detail how the system design of FEAST addresses these three tenets.

A. Adaptability

In our formative user study, care recipients voiced 46 types of adaptability requests across 7 categories (Figure 3). Requests of the same type often differed from user to user. For example, P1 said: *"I would want the robot to be fast, as fast as it could go maybe up until those very last moments of moving towards my head,"* while P9 said, *"I have a trach and have to eat slowly."* P10 said: *"It would be useful to give the robot more autonomy during those social situations, to pick the bite or whatever, rather than me specifying something exactly,"* and P5 said, *"This might be just a personal thing, but the idea of hitting a switch and then waiting for something to happen makes me then feel like I'm no longer in control."* Users also voiced the importance of adapting over time: for example, P8 mentioned, *"I'm able to do that right now. But these things might change over time. So you might want to keep updating the robot with the progression of the disease."*

In Figure 5, we compare FEAST to two state-of-the-art feeding systems, Nanavati et al. [36] and FLAIR [20], based on their ability to address the 46 adaptability request types identified in our study. FEAST covers 36, compared to 17 for Nanavati et al. and 15 for FLAIR. Appendix A provides detailed justifications for each system. Appendix B presents an ablation study demonstrating that our LLM-based personalized gesture synthesis outperforms a no-personalization baseline.

B. Transparency

The IEEE Standard for Transparency of Autonomous Systems [40, 106], which explicitly includes care robots within its scope, defines five levels of transparency requirements, ranging from 1 to 5 (Figure 5). FEAST aims to meet all these levels.

Level 1 specifies that users must have access to information such as example scenarios and general principles of operation. FEAST satisfies this requirement with an instruction manual provided to new users. Level 2 requires interactive training material, enabling users to rehearse their interactions with the system in virtual scenarios. To address this, we provide demonstration videos of the system across various use cases. Moreover, after adding personalized gestures, users can test them directly on the real robot in isolation.

Levels 3 and 4 focus on providing user-initiated explanations. Level 3 requires explanations of the system's most recent activity, while Level 4 demands answers to "what if" queries about system behavior. FEAST uses LLM summarization to fulfill these requirements. We define the following to encapsulate the current system state:

1) Current Behavior: Current behavior tree encodings for mealtime-assistance skills, with text descriptions for each node and its parameters.
2) Node Execution History: A real-time log of behavior tree nodes in-execution and completed.
3) Perception Log: A real-time log of perceived data, such as plate contents, drink pose, user head pose, and gestures.
4) Safety Log: A real-time log of safety checks, including any invalidated specifications.

On the Personalization Page of the web interface, users can submit language queries. The LLM processes these queries–along with the current system state, detailed descriptions of system operations (as outlined in Section IV), and prior user queries and the corresponding LLM's responses–to generate concise, understandable explanations for non-expert users.

Level 5 transparency requires the system to provide continuous explanations of its behavior. FEAST aims to achieve this using LLM summarization. At fixed-time intervals, the system checks for changes in the system state as defined above. If a change is detected, the LLM is queried with detailed descriptions of system operations, the previous state, and the current state, prompting it to generate an explanation of what changed during that period. This explanation is displayed on the web interface whenever no other page is active.

C. Safety

FEAST fits within the scope of ISO 13842: Robots and Robotic Devices–Safety Requirements for Personal Care Robots [39], categorized as a "physical assistant robot", with the following hardware and software checks (Figure 5):

Pose filtering. Our pose detection methods (food, drink, and head pose) use predefined zones based on expected plate, drink, and head positions, initialized at the start of each mealtime assistance scenario. Head pose limits are tailored to the user's range of motion [107]. If detections occur outside these zones—such as mistakenly identifying a caregiver's head pose behind the user during feeding (a scenario that can arise in real-world settings)—the robot transitions to a safety state.

Active collision monitoring. While our robot verifies its motion plans for collisions with known obstacles, it also uses active collision monitoring to handle unexpected collisions when not using compliant control. This is achieved by comparing torque feedback from the joint F/T sensors with torque predictions from a nominal model-based control law [108].

Compliant control. For inside-mouth transfer, the most physical human-robot interaction intensive aspect of mealtime assistance, our robot switches to a custom Cartesian-space compliant controller that leverages Damped Least Squares [109].

Safety-centric hardware design. Within the feeding utensil, the fork is attached to a custom-designed holder with a mechanical weak point that yields under excessive force, breaking the utensil into two pieces. This safety feature ensures that if excessive force is applied while the utensil is in the user's mouth, only the fork tip remains, halting all physical interaction with the robot. For safety, we restrict outside-mouth bite transfers to the default silicone fork. Similarly, the drinking utensil uses a silicone straw, and the mouth wiper features a soft tip.

Emergency Stops. At all times, both the user and an experimenter have access to physical emergency stop buttons, which immediately transition the robot into a safety state.

Watchdog. A watchdog continuously monitors the functionality of all robot sensors and emergency stops, transitioning the robot to a safety state if an issue is detected.

We provide implementation details for the above and further discuss alignment with ISO 13842 guidelines in Appendix C.

VI. Five-Day In-the-Wild Evaluation

To showcase the real-world applicability of FEAST, we evaluate our system with two care recipients (CR1 and CR2) in a five-day in-home study spanning six realistic meals across three distinct environmental contexts:

1) **Personal context** where they focus solely on eating.
2) **Watching television** while they are eating.
3) **Social context** where they eat with another researcher.

This study asks the question: **can end-users effectively personalize FEAST to meet their requirements across various mealtime scenarios?**

A. Study Procedure

Introductory Meals and System Familiarization. Before any evaluation meals, we conduct one introductory meal with each care recipient on Day 1 of the study to familiarize them with the system. During these training meals, we explain the components of the mealtime-assistance system, demonstrate its various personalization features, and answer their questions.

Evaluation Meals. Over the course of the next 4 days, we feed 3 meals each to the two CRs, in their own homes, in the following chronological order:

- **Meal ID 1** (on Day 2): CR1 Dinner in Personal Context – buffalo chicken bites, potato wedges, and ranch dressing (home cooked meal), served with water.
- **Meal ID 2** (on Day 2): CR1 After-dinner dessert while Watching TV - strawberries with whipped cream (store-bought meal), served with water.
- **Meal ID 3** (on Day 3): CR2 Dinner in Personal Context - chicken nuggets, apple slices, and ketchup purchased from McDonald's, served with water.
- **Meal ID 4** (on Day 4): CR2 Lunch in Social Context - protein shake (CR2's usual lunch). Mug is cleaned and refilled with water after the shake is finished upon request.

Meal ID 1: Buffalo Chicken Bites, Potato Wedges, Ranch (home cooked)

Meal ID 2: Strawberry, Whipped Cream (store bought)

Meal ID 3: Chicken Nuggets, Apple Slices, Ketchup (from McDonalds)

Meal ID 5: General Tso's Chicken, Broccoli (home cooked)

Meal ID 6: Chicken Breasts, Hash Browns (home cooked)

Fig. 6: Meals fed to care recipients. The images, captured by the robot's in-hand camera, highlight the variability in lighting conditions. Each meal also included water. Meal ID 4 consisted only of a protein shake, which CR2 usually has for lunch.

- **Meal ID 5** (on Day 5): CR2 Dinner while Watching TV - General Tso's Chicken and Broccoli (home cooked meal), served with water.
- **Meal ID 6** (on Day 5): CR1 Dinner in Social Context - chicken breast strips and hash browns (home cooked meal), served with water.

We select the meals for the study based on the care recipient's eating habits, the robot's capabilities, and the need for variation. While we validate our system with similar meal types during study preparation, the exact meals naturally vary based on how they are prepared and what is available. Many adjustments were made on the fly; for example, hash browns from CR1's fridge were added to Meal ID 6 at the last minute.

Environmental and User-Specific Variability. Care recipients live in different homes and have unique preferences for their environment setup. This leads to variations both across and within the same context (see Figure 1), including:

- **Tools:** CR1 uses a metal fork for all meals and a pink mug for the last meal. CR2 uses a silicone fork (safer for inside-mouth transfer) and a black mug for all meals.
- **Web Interface Interaction:** Both use iPads. CR1, with limited limb mobility, interacts using their left arm, while CR2, with very limited limb mobility, uses a mouse tracker that follows a reflective dot on their nose.
- **Eating Tables:** Different table types include multiple overbed tables, a coffee table, and a dining table.
- **Seating Arrangements:** While watching TV, both CR1 and CR2 face the screen. In a social setting, CR1's social partner sits to their right, whereas CR2's social partner sits directly in front.

Meal Setup and Scene Configuration. For each new scene, experimenters update scene parameters used for safety checks, such as table height (ensuring a lower limit for acquisition actions) and the user's head pose (used for filtering head perception). They also adjust the fixed configurations that the robot moves to before detecting the plate, drink, and user's head to ensure their visibility. After the CRs personalize the robot for their personal contexts in their first evaluation meals, we initialize their other contexts with those settings.

B. Evaluation Metrics

Before the study, participants rate their experiences with human caregivers during mealtime on a 5-point Likert scale, which is later compared to FEAST. To evaluate system performance, we log experimenter interventions for each meal, categorized into hardware, web interface, skills, and personalization. We also track experimenter explanations, recording the number of questions participants ask. Skill success rates are measured for autonomous actions, which rely on perception, such as bite acquisition, bite transfer, drink pickup, drink transfer, and wipe transfer. Fixed-motion actions, such as utensil and mouth wipe pickup from the tool mount and subsequent stow, always succeed. After each meal, participants complete a NASA-TLX [37] survey (7-point Likert scale) to assess cognitive workload. At the end of the study, a Technology Acceptance Model (TAM) [38] survey evaluates overall satisfaction and usability, along with additional questions assessing key personalization tenets: adaptability, transparency, and safety. Full questionnaires are in Appendix D.

C. Results

For each meal, CR1 and CR2 reported either fully clearing their plate or eating until they were full. Table II summarizes objective system performance per meal, whereas Figure 7 shows subjective metrics. Details on each of the interventions, explanations, and personalization requests made during the study can be found in Appendix E.

Interventions. A total of 18 experimenter interventions occurred across six meals ($\mathbf{3 \pm 2}$). Four hardware interventions in the first meal were due to a damaged ArUco marker on the mug handle, which was later replaced. All but one web interface intervention resulted from a recurring network issue that intermittently blocked commands, requiring a refresh.

Explanations. Users asked experimenters 20 questions across six meals ($\mathbf{3.33 \pm 2.69}$). Many could have been directed to the robot, but the transparency page was only accessible from the task selection page, underscoring the need for transparency throughout the eating process.

Skill Success Rates. FEAST demonstrates robustness to different food items, with a bite acquisition success rate of $\mathbf{89.27 \pm 9.24\%}$ and a bite transfer success rate of $\mathbf{93.07 \pm 7.70\%}$. Prior work suggests that an 80% acquisition rate is sufficient for use [10]. While drink acquisition success was low for Meal ID 1 due to a damaged marker on the drink handle during transit, it improved significantly after

TABLE II: Per-meal system performance, including experimenter interventions, explanations, and skill success rates.

Meal	Interventions				Explanations	Skill Success Rate				
	Hardware	Web Interface	Skill	Personalization		Bite Acq.	Bite Transfer	Drink Acq.	Drink Transfer	Wipe Transfer
#1: CR1 Personal	4	2	1	0	8	24/25	14/17	3/7	2/3	1/1
#2: CR1 TC	1	1	0	0	1	22/27	11/11	1/1	1/1	-
#3: CR2 Personal	1	1	0	0	6	29/37	15/17	3/3	3/3	1/1
#4: CR2 Social	1	0	1	0	2	-	-	2/2	5/5	-
#5: CR2 TV	0	4	0	0	1	19/21	18/19	4/4	2/2	2/2
#6: CR1 Social	0	0	0	1	2	12/12	12/12	2/2	2/2	1/1

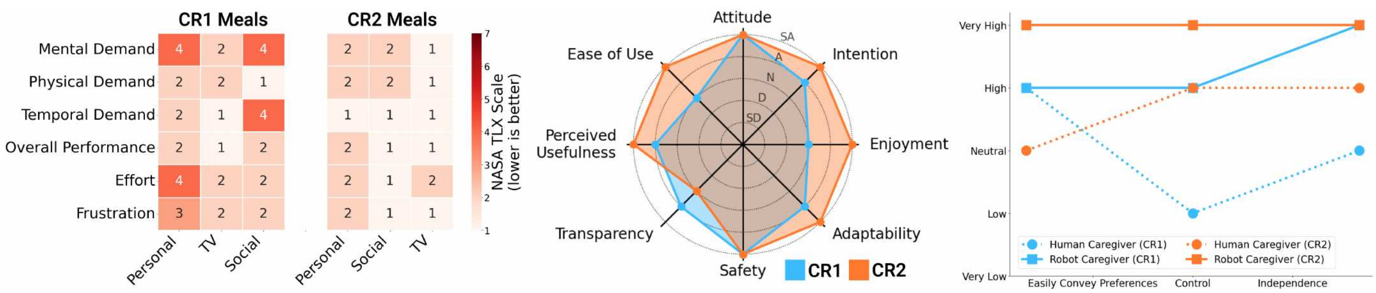


Fig. 7: Left: Per-meal NASA-TLX survey results, with CR1 and CR2 meals shown in chronological order. Center: TAM results, including questions on personalization tenets. Right: Comparison between human caregivers and FEAST.

replacement, averaging $\mathbf{100.0 \pm 0.0\%}$ across Meal IDs 2-5. Drink and wipe transfers were also highly successful at $\mathbf{94.44 \pm 13.61\%}$ and $\mathbf{100.0 \pm 0.0\%}$ respectively.

Cognitive Workload Per Meal. The average workload imposed by FEAST was relatively low. On a 0–100 NASA-TLX scale (where lower scores indicate lower workload), CR1 had a mean of $\mathbf{22.22 \pm 10.02}$, while CR2 had a mean of $\mathbf{7.41 \pm 5.79}$, compared to a baseline of 37 [1, 110]. Both care recipients rated their success in achieving mealtime goals very positively: the inverted Performance metric (0–100, lower = better) is $\mathbf{11.11 \pm 9.62}$ for CR1 and $\mathbf{5.56 \pm 9.62}$ for CR2.

Comparison with Human Caregiver. Both CRs rated FEAST as providing greater control over their meals and a greater sense of independence compared to their professional caregivers. CR2 also reported less effort in conveying preferences to the robot than to their human caregiver.

TAMs and Personalization Tenets. Results from the TAM survey indicate that both CRs rated FEAST highly across all metrics, with a mean score of $\geq 4/5$ for Perceived Usefulness, Ease of Use, Attitude, Intention, and Enjoyment. They also rated the system highly for adaptability and safety but noted that transparency could be improved. This may be because the transparency page was only accessible from the task selection page on the web interface, which was not always available, requiring them to ask experimenters for explanations.

D. Lessons Learned

Lesson 1

Significant variability exists across in-home eating scenarios, and in-the-wild personalization allows users to uniquely adapt the system to these variations.

Across different meals, both CR1 and CR2 made a range of personalization requests. These included adjusting robot speed (*"Feed me as fast as you can,"* Meal ID 1), modifying skill parameters (*"Dip the strawberry deeper into the whipped cream,"* Meal ID 2), changing web interface workflows (*"Don't show continue pages on the web interface,"* Meal ID 3), and customizing interactions between the robot and user, such as muting the robot (*"Be quiet and do not talk at all,"* Meal ID 5) or changing how transfers are confirmed (*"Use the button to complete a transfer only when taking a sip,"* Meal ID 1).

Even under similar contexts, the CRs had different personalization needs. For example, CR2 added a *"continuous mouth open"* personalized gesture detector for initiating transfers in a social setting, to differentiate from when they are talking. On the other hand, CR1 realized the default mouth-open detector required opening their mouth very wide. This design allowed CR1 to talk without inadvertently triggering the robot, and then open their mouth wide when ready to receive a bite. Thus, CR1 continued using it in social settings.

The spatial arrangement of people and objects also influenced personalization preferences. In CR2's social context, the social partner sat across the table, so CR2 rarely felt obstructed by the robot: *"I completely forgot about the robot sometimes and had to remember that I had to take a drink."* Conversely, in CR1's social context, the social partner was seated to the right—directly across from the robot—prompting CR1 to request the robot retract after every bite (*"Move to retract position after every bite"*) to avoid interference.

Lesson 2

Transparency helps users iteratively refine the system to meet their preferences, even when adaptability commands are not always effective.

FEAST's adaptability leverages an LLM to process language commands and update the robot's behavior. However, LLMs can sometimes hallucinate or make mistakes. For example, when CR1 tried to switch to a button for wipe transfers with the command, *"Use button when completing a transfer* *when taking a sip,"* the LLM mistakenly applied this change to all tools, responding, *"The robot-assisted feeding system has been updated to use a button to complete actions when transferring drinks, utensils, and wipes."* Despite the error, the system's transparency features allowed the user to recover. CR1 utilized transparency to verify the transfer completion method by asking, *"What is the default action to complete a transfer?"* and *"What other ways can I end a transfer besides pushing the button?"* After several transparency-adaptability iterations, CR1 submitted a detailed adaptability request: *"Use sensors to end transactions with utensil use, the button to end transactions when taking a sip, and the button to end transactions for face wipes."* Through these transparent interactions, CR1 was able to iteratively refine the robot's behavior to match their preferences, even when sometimes the initial adaptability command did not function as intended.

Lesson 3

Providing multiple interfaces is essential for transparency, as users may not always be able to interact with a single interface due to situational and environmental constraints.

Over multiple days of evaluating a flexible system, users often forgot the robot's settings and expectations (which gesture it is waiting to detect). FEAST provides transparency through voice prompts and a web interface, with multiple options proving crucial. In social settings, CR1 and CR2 switched off voice prompts and relied on the web interface for guidance. However, in cases where neither was accessible—such as when CR2 used a personalized gesture detector for *"keep head still"* during mouth wiping—the wiper's position right at their mouth prevented them from checking the interface. As CR2 noted, *"A small display on the gripper would help. I can't check the web interface while the fork or another tool is at my mouth and the robot expects something. Or maybe I could ask directly, like 'Hey robot, what am I supposed to do right now?'"*

Lesson 4

Cognitive workload generally decreases as users become more familiar with the system, but it also depends on the context and specific settings they choose.

A flexible system allows users to tinker and personalize their experience, but it can also impose cognitive workload. While NASA-TLX surveys indicated that our system did not impose significant cognitive workload during evaluation meals, this may be because introductory meals helped users acclimate. Over three meals, cognitive workload generally trended downward (Figure 7) as users became more familiar with the system, except for Meal ID 6 (a social meal for CR1). CR1 noted that the mental and temporal demand was not caused by the personalization process itself but rather by the specific settings they had chosen, which were not optimal for the context. *"I set the [auto-continue task selection] timer to 100 seconds, but that wasn't enough. I kept watching the countdown while trying to keep up with the conversation, making sure I was ready for my next bite before time ran out. Next time, I'd set it much longer."* Despite this challenge, they acknowledged that familiarity would make the process easier over time: *"The more I eat in this setting, the easier it'll get. I'll get used to what the robot can do and figure out which settings actually work for me."*

Lesson 5

System failures can occur in-the-wild for various reasons, but system flexibility and keeping the user in the loop improves the robot's ability to recover.

During our in-home evaluation, several unexpected skill failures occurred. For example, varying lighting conditions affected food perception, making it difficult for the robot to detect food items on the plate. However, the system provided a manual skill execution option on the web interface, allowing users to pick up undetected bites by selecting the skewering skill and a keypoint on a plate image. In cases where bite transfer failed and the robot was unaware, it would return to the plate to reacquire food, even though a food item was already on the fork. To recover, users switched to manual skill selection, skewering an empty space on the plate to retain the same food item on the fork and retry bite transfer.

Similarly, the default head nod detector—used to detect when a user was finished drinking or wiping their mouth—did not work well for CR1. When leaning forward to take a sip or wipe, they inadvertently leaned downward, which the system misinterpreted as a nod, causing the robot to move away too soon. However, because the system was flexible, CR1 was able to switch from gesture detection to a button for this task.

These examples highlight how system flexibility, combined with user involvement, allows the robot to recover from difficult to predict failures that arise in the real world.

VII. Assessing Ecological Validity with an Occupational Therapist

To assess FEAST's performance beyond the two community researchers, we evaluate our system with an Occupational Therapist (OT) not involved in system development procedure. Since OTs work with diverse care recipients with varying mobility limitations, their evaluation offers a comprehensive measure of the system's ecological validity. The OT was introduced to FEAST as in our in-the-wild evaluations—through the training manual and a training meal. After reviewing personalization categories from our formative study, the OT independently created 10 realistic personalization scenarios. For each scenario, they personalized the default FEAST system to their requirements using adaptability and transparency requests. We report: (1) user-reported personalization success rated at $5/5$, averaged across scenarios; (2) user-reported ease of personalization rated at 4.6 ± 0.16 out of 5, averaged across scenarios; (3) a preference for the personalized system over the default (no-personalization) system in all 10 scenarios; (4) a total of 13 adaptability and 2 transparency requests across all scenarios; and (5) a total of 2 experimenter interventions and 1 experimenter explanation throughout the evaluation. User ratings are on a 1–5 Likert scale (1 = Strongly Disagree, 5 = Strongly Agree). Appendix F details the personalization scenarios, including the adaptability and transparency requests made and any experimenter interventions or explanations. The OT also offered feedback on the interface design, stating: "Visually, it looks really good; the contrast is good, it should work for most care recipients... having speech-to-text is helpful because that is how many communicate with their devices."

VIII. Limitations and Future Work

Open-Set Food Detection. FEAST receives user input about food items on the plate and uses an open-set object detector to locate them. However, the detector often mislabels items, particularly when plates contain multiple food types (e.g., chicken nuggets and potato wedges). During evaluation, additional qualifiers like "yellow potato wedge piece" or "cut-up chicken nugget piece" were required. While users could potentially provide such qualifiers to improve detection accuracy, the effectiveness of this approach remains to be evaluated. In the future, more accurate open-set detectors are needed to address this limitation.

Adaptability Based on Plate Context: Some user adaptability requests required the system to have additional context about plate contents and selectively adjust behavior tree parameters. For instance, in Meal ID 1, CR1 requested more ranch for potato wedges and less for chicken bites. However, since FEAST currently uses a single dipping parameter, this required constant adjustments between bites. Future work should explore adaptability mechanisms that dynamically adjust behavior based on real-time plate contents.

LLM Hallucinations. FEAST leverages LLMs to interpret user requests and modify robot behavior, but hallucinations can lead to incorrect or incomplete adaptations, causing user confusion. While FEAST's transparency features help users identify and correct such errors, these features themselves rely on LLMs, which may also hallucinate. Future research should focus on reducing LLM-induced errors and exploring fallback mechanisms to ensure reliability.

Long-term Preference Learning: CR1 noted that generating a full set of personalization requests from scratch for a new setting could impose unnecessary cognitive workload. They suggested allowing users to describe their dining environment so that the system could initialize a relevant configuration, from which they could make refinements. Achieving this would require long-horizon preference learning, where FEAST analyzes past user preferences to propose suitable initial settings for different contexts.

Evaluation Beyond Community Researchers. Our evaluation involved two care recipients who are also community researchers. Since they were familiar with FEAST's development, they may have had a lower cognitive workload than other users. Future studies should evaluate FEAST with care recipients who are not community researchers to better understand how unfamiliar users perceive the system's usability.

Open-Loop Skill Limitations. FEAST employs open-loop skills for bite acquisition, which can fail when food items slip. Moreover, while users can adjust skill parameters to control bite size—such as scooping more or less—these settings often require frequent adjustments within a single meal as food properties change over time. For example, the same scooping distance may yield smaller portions as food consistency shifts. Incorporating closed-loop approaches to bite acquisition could improve robustness, allowing the system to adapt dynamically to food slippage and changing food properties.

IX. Acknowledgement

All data collection and user studies described in this paper were approved by the Cornell University Institutional Review Board. This work was partly funded by the National Science Foundation under grants IIS #2132846 and CAREER #2238792. We are deeply grateful to Aimee Whitneck (CR1) and Benjamin Howe (CR2) for their involvement throughout this work. We also thank Amal Nanavati for assisting with participant recruitment for the formative user study, and all formative study participants for their valuable feedback, which guided system development. Finally, we thank the anonymous reviewers for their insightful feedback.

References

[1] A. Nanavati, P. Alves-Oliveira, T. Schrenk, E. K. Gordon, M. Cakmak, and S. S. Srinivasa, "Design principles for robot-assisted feeding in social contexts," in Proceedings of the 2023 ACM/IEEE International Conference on Human-Robot Interaction, 2023, pp. 24–33.

[2] A. Mandy, T. Sims, G. Stew, and D. Onions, "Manual feeding device experiences of people with a neurodisability," The American Journal of Occupational Therapy, vol. 72, no. 3, pp. 7 203 345 010p1–7 203 345 010p5, 2018.

[3] C. Jacobsson, K. Axelsson, P. O. Österlind, and A. Norberg, "How people with stroke and healthy older people experience the eating process," Journal of Clinical Nursing, vol. 9, no. 2, pp. 255–264, 2000. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1046/j.1365-2702.2000.00355.x

[4] S. E. Shune, "An altered eating experience: Attitudes toward feeding assistance among younger and older adults," Rehabilitation nursing : the official journal of the Association of Rehabilitation Nurses, 2020.

[5] D. M. Taylor, "Americans with disabilities: 2014," US Census Bureau, pp. 1–32, 2018.

[6] A. Chiò, A. Gauthier, A. Vignola, A. Calvo, P. Ghiglione, E. Cavallo, A. Terreni, and R. Mutani, "Caregiver time use in als," Neurology, vol. 67, no. 5, pp. 902–904, 2006.

[7] K. C. Fleming, J. M. Evans, and D. S. Chutka, "Caregiver and clinician shortages in an aging nation," in Mayo Clinic Proceedings, vol. 78, no. 8. Elsevier, 2003, pp. 1026–1040.

[8] R. Zenker, C. Pardell, A. Gilmore-Bykovskyi, A. J. Kind, and N. Werner, "Exploring workload among informal caregivers of persons with dementia," in

proceedings of the human factors and ergonomics society annual meeting, vol. 61, no. 1. SAGE Publications Sage CA: Los Angeles, CA, 2017, pp. 1297–1297.
[9] S. W. Brose, D. J. Weber, B. A. Salatin, G. G. Grindle, H. Wang, J. J. Vazquez, and R. A. Cooper, "The role of assistive robotics in the lives of persons with disability," American Journal of Physical Medicine & Rehabilitation, vol. 89, no. 6, pp. 509–521, 2010.
[10] T. Bhattacharjee, E. K. Gordon, R. Scalise, M. E. Cabrera, A. Caspi, M. Cakmak, and S. S. Srinivasa, "Is more autonomy always better? exploring preferences of users with mobility impairments in robot-assisted feeding," in 2020 15th ACM/IEEE International Conference on Human-Robot Interaction (HRI). IEEE, 2020, pp. 181–190.
[11] R. K. Jenamani, D. Stabile, Z. Liu, A. Anwar, K. Dimitropoulou, and T. Bhattacharjee, "Feel the bite: Robot-assisted inside-mouth bite transfer using robust mouth perception and physical interaction-aware control," in Proceedings of the 2024 ACM/IEEE International Conference on Human-Robot Interaction, 2024, pp. 313–322.
[12] R. Feng, Y. Kim, G. Lee, E. K. Gordon, M. Schmittle, S. Kumar, T. Bhattacharjee, and S. S. Srinivasa, "Robot-assisted feeding: Generalizing skewering strategies across food items on a plate," in The International Symposium of Robotics Research. Springer, 2019, pp. 427–442.
[13] E. K. Gordon, X. Meng, T. Bhattacharjee, M. Barnes, and S. S. Srinivasa, "Adaptive robot-assisted feeding: An online learning framework for acquiring previously unseen food items," in 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2020, pp. 9659–9666.
[14] E. K. Gordon, S. Roychowdhury, T. Bhattacharjee, K. Jamieson, and S. S. Srinivasa, "Leveraging post hoc context for faster learning in bandit settings with applications in robot-assisted feeding," in 2021 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2021, pp. 10 528–10 535.
[15] E. K. Gordon, A. Nanavati, R. Challa, B. H. Zhu, T. A. K. Faulkner, and S. Srinivasa, "Towards general single-utensil food acquisition with human-informed actions," in Conference on Robot Learning. PMLR, 2023, pp. 2414–2428.
[16] J. Grannen, Y. Wu, S. Belkhale, and D. Sadigh, "Learning bimanual scooping policies for food acquisition," in 6th Annual Conference on Robot Learning, 2022. [Online]. Available: https://openreview.net/forum?id=qDtbMK67PJG
[17] Y.-L. Tai, Y. C. Chiu, Y.-W. Chao, and Y.-T. Chen, "Scone: A food scooping robot learning framework with active perception," in Conference on Robot Learning. PMLR, 2023, pp. 849–865.
[18] R. Liu, Z. Mahammad, A. Bhaskar, and P. Tokekar, "Imrl: Integrating visual, physical, temporal, and geometric representations for enhanced food acquisition," arXiv preprint arXiv:2409.12092, 2024.
[19] P. Sundaresan, J. Wu, and D. Sadigh, "Learning sequential acquisition policies for robot-assisted feeding," in Conference on Robot Learning. PMLR, 2023, pp. 1282–1299.
[20] R. K. Jenamani, P. Sundaresan, M. Sakr, T. Bhattacharjee, and D. Sadigh, "Flair: Feeding via long-horizon acquisition of realistic dishes," arXiv preprint arXiv:2407.07561, 2024.
[21] N. Ha, R. Ye, Z. Liu, S. Sinha, and T. Bhattacharjee, "Repeat: A real2sim2real approach for pre-acquisition of soft food items in robot-assisted feeding," IEEE International Conference on Intelligent Robots and Systems (IROS), 2024.
[22] A. Bhaskar, R. Liu, V. D. Sharma, G. Shi, and P. Tokekar, "Lava: Long-horizon visual action based food acquisition," arXiv preprint arXiv:2403.12876, 2024.
[23] D. Gallenberger, T. Bhattacharjee, Y. Kim, and S. S. Srinivasa, "Transfer depends on acquisition: Analyzing manipulation strategies for robotic feeding," in 2019 14th ACM/IEEE International Conference on Human-Robot Interaction (HRI). IEEE, 2019, pp. 267–276.
[24] S. Belkhale, E. K. Gordon, Y. Chen, S. Srinivasa, T. Bhattacharjee, and D. Sadigh, "Balancing efficiency and comfort in robot-assisted bite transfer," in 2022 International Conference on Robotics and Automation (ICRA). IEEE, 2022, pp. 4757–4763.
[25] D. Park, Y. Hoshi, H. P. Mahajan, H. K. Kim, Z. Erickson, W. A. Rogers, and C. C. Kemp, "Active robot-assisted feeding with a general-purpose mobile manipulator: Design, evaluation, and lessons learned," Robotics and Autonomous Systems, vol. 124, p. 103344, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0921889018307061
[26] L. Shaikewitz, Y. Wu, S. Belkhale, J. Grannen, P. Sundaresan, and D. Sadigh, "In-mouth robotic bite transfer with visual and haptic sensing," arXiv preprint arXiv:2211.12705, 2022.
[27] R. Madan, R. K. Jenamani, V. T. Nguyen, A. Moustafa, X. Hu, K. Dimitropoulou, and T. Bhattacharjee, "Sparcs: Structuring physically assistive robotics for caregiving with stakeholders-in-the-loop," in 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2022, pp. 641–648.
[28] K. Hacker, Community-based participatory research. Sage publications, 2013.
[29] R. S. Kushalnagar and C. Vogler, "Teleconference accessibility and guidelines for deaf and hard of hearing users," in Proceedings of the 22nd International ACM SIGACCESS Conference on Computers and Accessibility, 2020, pp. 1–6.

[30] C. Haidenhofer, L. Sitbon, C. P. Beaumont, M. Hoogstrate, and J. L. Korte, "From research participant to co-researcher: Chloe's story on co-designing inclusive technologies with people with intellectual disability," in Proceedings of the 26th International ACM SIGACCESS Conference on Computers and Accessibility, 2024, pp. 1–6.
[31] F. Fussenegger and K. Spiel, "Depending on independence an autoethnographic account of daily use of assistive technologies," in Proceedings of the 24th International ACM SIGACCESS Conference on Computers and Accessibility, 2022, pp. 1–6.
[32] T. L. Chen, M. Ciocarlie, S. Cousins, P. Grice, K. Hawkins, K. Hsiao, C. C. Kemp, C.-H. King, D. A. Lazewatsky, A. Leeper et al., "Robots for humanity: User-centered design for assistive mobile manipulation," in 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, 2012, pp. 5434–5435.
[33] S. Moharana, A. E. Panduro, H. R. Lee, and L. D. Riek, "Robots for joy, robots for sorrow: community based robot design for dementia caregivers," in 2019 14th ACM/IEEE International Conference on Human-Robot Interaction (HRI). IEEE, 2019, pp. 458–467.
[34] A. Padmanabha, J. Gupta, C. Chen, J. Yang, V. Nguyen, D. J. Weber, C. Majidi, and Z. Erickson, "Independence in the home: A wearable interface for a person with quadriplegia to teleoperate a mobile manipulator," in Proceedings of the 2024 ACM/IEEE International Conference on Human-Robot Interaction, 2024, pp. 542–551.
[35] V. Ranganeni, V. Nguyen, H. Evans, J. Evans, J. Mehu, S. Olatunji, W. Rogers, A. Edsinger, C. Kemp, and M. Cakmak, "Robots for humanity: In-home deployment of stretch re2," in Companion of the 2024 ACM/IEEE International Conference on Human-Robot Interaction, 2024, pp. 1299–1301.
[36] A. Nanavati, E. K. Gordon, T. A. Kessler Faulkner, Y. R. Song, J. Ko, T. Schrenk, V. Nguyen, B. H. Zhu, H. Bolotski, A. Kashyap, S. Kutty, R. Karim, L. Rainbolt, R. Scalise, H. Song, R. Qu, M. Cakmak, and S. S. Srinivasa, "Lessons learned from designing and evaluating a robot-assisted feeding system for out-of-lab use," in Proceedings of the 2025 ACM/IEEE International Conference on Human-Robot Interaction, 2025.
[37] S. Hart, "Development of nasa-tlx (task load index): Results of empirical and theoretical research," Human mental workload/Elsevier, 1988.
[38] F. D. Davis, R. P. Bagozzi, and P. R. Warshaw, "User acceptance of computer technology: A comparison of two theoretical models," Management science, vol. 35, no. 8, pp. 982–1003, 1989.
[39] I. O. for Standardization, Robots and Robotic Devices: Safety Requirements for Personal Care Robots, 2014.
[40] I. T. S. C. of the IEEE Vehicular Technology Society, the Standing Committee for Standards of the IEEE Robotics, and A. Society, "Ieee standard for transparency of autonomous systems," IEEE Std 7001-2021, 2022.
[41] "Meet Obi," https://meetobi.com/, [Online; accessed 1-Jan-2025].
[42] "Neater eater robot," 2024, (Accessed: 1st January, 2025). [Online]. Available: https://www.neater.co.uk/neater-eater-robotic
[43] A. Staros and E. Peizer, "Veterans administration prosthetics center research report," Bulletin of Prosthetics Research, 1977.
[44] I. Naotunna, C. J. Perera, C. Sandaruwan, R. Gopura, and T. D. Lalitharatne, "Meal assistance robots: A review on current status, challenges and future directions," in IEEE/SICE International Symposium on System Integration, 2015, pp. 211–216.
[45] P. Sundaresan, S. Belkhale, and D. Sadigh, "Learning visuo-haptic skewering strategies for robot-assisted feeding," in 6th Annual Conference on Robot Learning, 2022. [Online]. Available: https://openreview.net/forum?id=lLq09gVoaTE
[46] K. Zhang, M. Sharma, M. Veloso, and O. Kroemer, "Leveraging multimodal haptic sensory data for robust cutting," in 2019 IEEE-RAS 19th International Conference on Humanoid Robots (Humanoids), 2019.
[47] A. Candeias, T. Rhodes, M. Marques, M. Veloso et al., "Vision augmented robot feeding," in Proceedings of the European Conference on Computer Vision (ECCV) Workshops, 2018, pp. 0–0.
[48] F. F. Goldau, T. K. Shastha, M. Kyrarini, and A. Gräser, "Autonomous multi-sensory robotic assistant for a drinking task," in 2019 IEEE 16th International Conference on Rehabilitation Robotics (ICORR). IEEE, 2019, pp. 210–216.
[49] S. Schröer, I. Killmann, B. Frank, M. Völker, L. Fiederer, T. Ball, and W. Burgard, "An autonomous robotic assistant for drinking," in 2015 IEEE international conference on robotics and automation (ICRA). IEEE, 2015, pp. 6482–6487.
[50] T. Ikeda, T. Makino, S. Furuno, and F. Nagata, "Wiping the mouth using the robot arm with information of rgb-d sensor," Artificial Life and Robotics, vol. 24, pp. 360–367, 2019.
[51] K. P. Hawkins, C.-H. King, T. L. Chen, and C. C. Kemp, "Informing assistive robots with models of contact forces from able-bodied face wiping and shaving," in 2012 IEEE RO-MAN: The 21st IEEE International Symposium on Robot and Human Interactive Communication. IEEE, 2012, pp. 251–258.
[52] T. Bhattacharjee, M. E. Cabrera, A. Caspi, M. Cakmak, and S. S. Srinivasa, "A community-centered design framework for robot-assisted feeding systems," in Proceedings of the 21st International ACM SIGACCESS Conference on Computers and Accessibility, 2019, pp. 482–494.

[53] G. Canal, C. Torras, and G. Alenyà, "Are preferences useful for better assistance? a physically assistive robotics user study," THRI, 2021.

[54] A. Nanavati, V. Ranganeni, and M. Cakmak, "Physically assistive robots: A systematic review of mobile and manipulator robots that physically assist people with disabilities," Annual Review of Control, Robotics, and Autonomous Systems, vol. 7, 2023.

[55] G. Yang, S. Wang, J. Yang, and P. Shi, "Desire-driven reasoning considering personalized care preferences," Transactions on Systems, Man, and Cybernetics: Systems, 2021.

[56] J. Z.-Y. He, Z. Erickson, D. S. Brown, A. Raghunathan, and A. Dragan, "Learning representations that enable generalization in assistive tasks," in Conference on Robot Learning. PMLR, 2023, pp. 2105–2114.

[57] Y. Gao, H. J. Chang, and Y. Demiris, "Grace: Generalizing robot-assisted caregiving with user functionality embeddings," in 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2015, pp. 1840–1845.

[58] J. Wu, R. Antonova, A. Kan, M. Lepert, A. Zeng, S. Song, J. Bohg, S. Rusinkiewicz, and T. Funkhouser, "Tidybot: Personalized robot assistance with large language models," arXiv preprint arXiv:2305.05658, 2023.

[59] Z. Liu, Y. Ju, Y. Da, T. Silver, P. N. Thakkar, J. Li, J. Guo, K. Dimitropoulou, and T. Bhattacharjee, "Grace: Generalizing robot-assisted caregiving with user functionality embeddings," arXiv preprint arXiv:2501.17855, 2025.

[60] A. Joshi, S. Ghosh, M. Betke, S. Sclaroff, and H. Pfister, "Personalizing gesture recognition using hierarchical bayesian neural networks," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 6513–6522.

[61] H. Brock and R. Gomez, "Personalization of human-robot gestural communication through voice interaction grounding," in 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2021, pp. 846–853.

[62] A. Taheri and Z. Weissman, "Designing a first person shooter game for quadriplegics," in Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems, 2021, pp. 1–6.

[63] A. Jackowski, M. Gebhard, and R. Thietje, "Head motion and head gesture-based robot control: A usability study," IEEE Transactions on Neural Systems and Rehabilitation Engineering, vol. 26, no. 1, pp. 161–170, 2017.

[64] G. Canal, G. Alenyà, and C. Torras, "Adapting robot task planning to user preferences: an assistive shoe dressing example," Autonomous Robots, 2019.

[65] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat et al., "Gpt-4 technical report," arXiv preprint arXiv:2303.08774, 2023.

[66] A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H. W. Chung, C. Sutton, S. Gehrmann et al., "Palm: Scaling language modeling with pathways," Journal of Machine Learning Research, vol. 24, no. 240, pp. 1–113, 2023.

[67] S. Kambhampati, "Can large language models reason and plan?" Annals of the New York Academy of Sciences, 2024.

[68] A. Padmanabha, J. Yuan, J. Gupta, Z. Karachiwalla, C. Majidi, H. Admoni, and Z. Erickson, "Voicepilot: Harnessing llms as speech interfaces for physically assistive robots," arXiv preprint arXiv:2404.04066, 2024.

[69] Y. Chen, J. Arkin, Y. Zhang, N. Roy, and C. Fan, "Autotamp: Autoregressive task and motion planning with llms as translators and checkers," arXiv preprint arXiv:2306.06531, 2023.

[70] W. Huang, F. Xia, T. Xiao, H. Chan, J. Liang, P. Florence, A. Zeng, J. Tompson, I. Mordatch, Y. Chebotar et al., "Inner monologue: Embodied reasoning through planning with language models," arXiv preprint arXiv:2207.05608, 2022.

[71] C. Agia, T. Migimatsu, J. Wu, and J. Bohg, "Taps: Task-agnostic policy sequencing," arXiv preprint arXiv:2210.12250, 2022.

[72] K. Lin, C. Agia, T. Migimatsu, M. Pavone, and J. Bohg, "Text2motion: From natural language instructions to feasible plans," Autonomous Robots, vol. 47, no. 8, pp. 1345–1365, 2023.

[73] I. Singh, V. Blukis, A. Mousavian, A. Goyal, D. Xu, J. Tremblay, D. Fox, J. Thomason, and A. Garg, "Progprompt: Generating situated robot task plans using large language models," in 2023 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2023, pp. 11 523–11 530.

[74] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, J. Hsu et al., "Rt-1: Robotics transformer for real-world control at scale," arXiv preprint arXiv:2212.06817, 2022.

[75] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, X. Chen, K. Choromanski, T. Ding, D. Driess, A. Dubey, C. Finn et al., "Rt-2: Vision-language-action models transfer web knowledge to robotic control," arXiv preprint arXiv:2307.15818, 2023.

[76] A. Padalkar, A. Pooley, A. Jain, A. Bewley, A. Herzog, A. Irpan, A. Khazatsky, A. Rai, A. Singh, A. Brohan et al., "Open x-embodiment: Robotic learning datasets and rt-x models," arXiv preprint arXiv:2310.08864, 2023.

[77] M. J. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. Foster, G. Lam, P. Sanketi et al., "Openvla: An open-source vision-language-action model," arXiv preprint arXiv:2406.09246, 2024.

[78] W. Yu, N. Gileadi, C. Fu, S. Kirmani, K.-H. Lee, M. G. Arenas, H.-T. L. Chiang, T. Erez, L. Hasenclever, J. Humplik et al., "Language to rewards for robotic skill

synthesis," arXiv preprint arXiv:2306.08647, 2023.
[79] Y. J. Ma, W. Liang, G. Wang, D.-A. Huang, O. Bastani, D. Jayaraman, Y. Zhu, L. Fan, and A. Anandkumar, "Eureka: Human-level reward design via coding large language models," arXiv preprint arXiv:2310.12931, 2023.
[80] J. Liang, W. Huang, F. Xia, P. Xu, K. Hausman, B. Ichter, P. Florence, and A. Zeng, "Code as policies: Language model programs for embodied control," in 2023 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2023, pp. 9493–9500.
[81] M. G. Arenas, T. Xiao, S. Singh, V. Jain, A. Ren, Q. Vuong, J. Varley, A. Herzog, I. Leal, S. Kirmani et al., "How to prompt your robot: A promptbook for manipulation skills with code as policies," in 2024 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2024, pp. 4340–4348.
[82] J. Grannen, S. Karamcheti, S. Mirchandani, P. Liang, and D. Sadigh, "Vocal sandbox: Continual learning and adaptation for situated human-robot collaboration," arXiv preprint arXiv:2411.02599, 2024.
[83] S. Belkhale, T. Ding, T. Xiao, P. Sermanet, Q. Vuong, J. Tompson, Y. Chebotar, D. Dwibedi, and D. Sadigh, "Rt-h: Action hierarchies using language," arXiv preprint arXiv:2403.01823, 2024.
[84] L. X. Shi, Z. Hu, T. Z. Zhao, A. Sharma, K. Pertsch, J. Luo, S. Levine, and C. Finn, "Yell at your robot: Improving on-the-fly from language corrections," arXiv preprint arXiv:2403.12910, 2024.
[85] "Kinova gen3 7dof robotic arm," 2025, (Accessed: 1st January, 2025). [Online]. Available: https://www.kinovarobotics.com/uploads/User-Guide-Gen3-R07.pdf
[86] "Robotic 2f-85 gripper," 2025, (Accessed: 1st January, 2025). [Online]. Available: https://robotiq.com/products/2f85-140-adaptive-robot-gripper
[87] "Rovi wheelchair," 2025, (Accessed: 1st January, 2025). [Online]. Available: https://www.rovimobility.com/
[88] "Vention stand," 2025, (Accessed: 1st January, 2025). [Online]. Available: https://vention.io/designs/heavy-duty-robot-pedestal-mobile-130889
[89] "Ati f/t sensor," 2025, (Accessed: 1st January, 2025). [Online]. Available: https://www.ati-ia.com/products/ft/sensors.aspx
[90] "Anti slip adjustable mug holder," 2025, (Accessed: 1st January, 2025). [Online]. Available: https://www.amazon.com/YOUCOX-Anti-Slip-Adjustable-Tumbler-SUNWILL/dp/B0936XNR2C
[91] S. Garrido-Jurado, R. Muñoz-Salinas, F. J. Madrid-Cuevas, and M. J. Marín-Jiménez, "Automatic generation and detection of highly reliable fiducial markers under occlusion," Pattern Recognition, vol. 47, no. 6, pp. 2280–2292, 2014.
[92] "Intel realsense d435i," 2025, (Accessed: 1st January, 2025). [Online]. Available: https://www.intelrealsense.com/depth-camera-d435i/
[93] "Intel nuc 12 (nuc12wshi7)," 2025, (Accessed: 1st January, 2025). [Online]. Available: IntelNUC12ProMiniPCKitNUC12WSHi7
[94] "Legion pro 7i gen 9 intel (16) with rtxtm 4090 (83de000aus)," 2025, (Accessed: 1st January, 2025). [Online]. Available: https://www.lenovo.com/us/en/p/laptops/legion-laptops/legion-pro-series/lenovo-legion-pro-7i-gen-9-16-inch-intel/83de000aus
[95] "Netgear nighthawk wifi router (rax43)," 2025, (Accessed: 1st January, 2025). [Online]. Available: https://www.amazon.com/NETGEAR-Nighthawk-5-Stream-Router-RAX43/dp/B08JY6RMG9/
[96] "Glassouse press switch," 2025, (Accessed: 1st January, 2025). [Online]. Available: https://glassouse.com/product/press-switch-100gr-glassouse-assistive-device/
[97] M. Colledanchise and P. Ögren, Behavior trees in robotics and AI: An introduction. CRC Press, 2018.
[98] A. Hurst, A. Lerer, A. P. Goucher, A. Perelman, A. Ramesh, A. Clark, A. Ostrow, A. Welihinda, A. Hayes, A. Radford et al., "Gpt-4o system card," arXiv preprint arXiv:2410.21276, 2024.
[99] D. M. McDermott, "The 1998 ai planning systems competition," AI magazine, vol. 21, no. 2, pp. 35–35, 2000.
[100] M. Helmert, "The fast downward planning system," Journal of Artificial Intelligence Research, vol. 26, pp. 191–246, 2006.
[101] "Tracker pro 2," 2025, (Accessed: 1st January, 2025). [Online]. Available: https://www.schoolhealth.com/tracker-pro-2
[102] E. You, "Vue.js," 2025, (Accessed: 1st January, 2025). [Online]. Available: https://vuejs.org
[103] "Ros noetic," May Accessed 2022. [Online]. Available: https://wiki.ros.org/noetic
[104] N. Hawes, C. Burbridge, F. Jovan, L. Kunze, B. Lacerda, L. Mudrova, J. Young, J. Wyatt, D. Hebesberger, T. Kortner et al., "The strands project: Long-term autonomy in everyday environments," IEEE Robotics & Automation Magazine, vol. 24, no. 3, pp. 146–156, 2017.
[105] S. Wang and H. I. Christensen, "Tritonbot: First lessons learned from deployment of a long-term autonomy tour guide robot," in 2018 27th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN). IEEE, 2018, pp. 158–165.
[106] A. F. Winfield, S. Booth, L. A. Dennis, T. Egawa, H. Hastie, N. Jacobs, R. I. Muttram, J. I. Olszewska, F. Rajabiyazdi, A. Theodorou et al., "Ieee p7001: A proposed standard on transparency," Frontiers in Robotics and AI, vol. 8, p. 665729, 2021.
[107] N. B. Reese and W. D. Bandy, Joint range of motion and muscle length testing-E-book. Elsevier Health Sciences, 2016.
[108] S. Haddadin, A. De Luca, and A. Albu-Schäffer, "Robot

collisions: A survey on detection, isolation, and identification," IEEE Transactions on Robotics, vol. 33, no. 6, pp. 1292–1312, 2017.
[109] Y. Nakamura and H. Hanafusa, "Inverse kinematic solutions with singularity robustness for robot manipulator control," Journal of dynamic systems, measurement, and control, vol. 108, no. 3, pp. 163–171, 1986.
[110] M. Hertzum, "Reference values and subscale patterns for the task load index (tlx): a meta-analytic review," Ergonomics, vol. 64, no. 7, pp. 869–878, 2021.