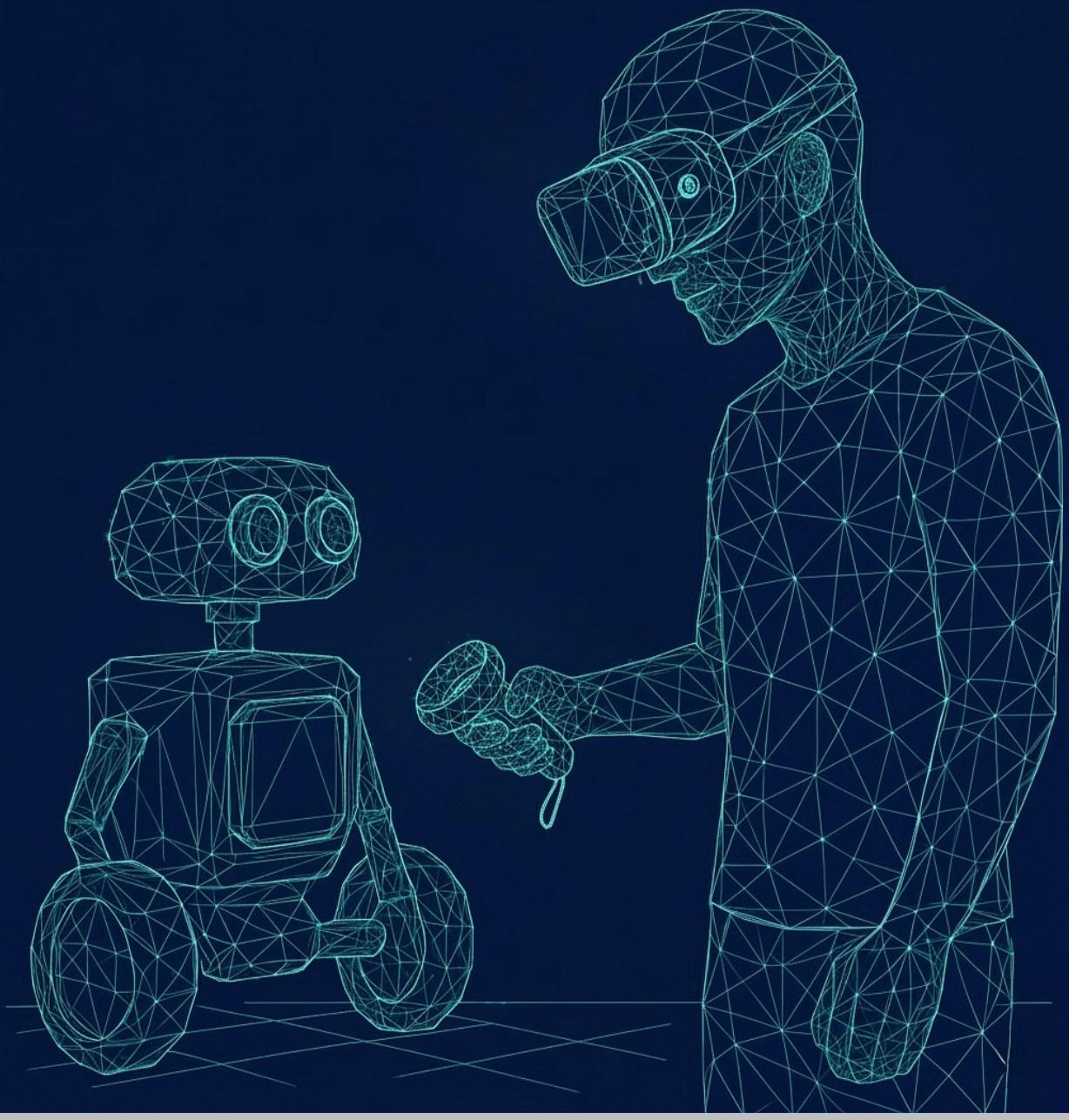
LEARNING PERSONALIZED AND HUMAN-AWARE ROBOT NAVIGATION

JORGE DE HEUVEL



UNI BONN

DISSERTATION

Bonn 2025

HRL

Learning Personalized and Human-Aware Robot Navigation

Dissertation

zur

Erlangung des Doktorgrades (*Dr. rer. nat.*)

der

Mathematisch-Naturwissenschaftlichen Fakultät

der

Rheinischen Friedrich-Wilhelms-Universität Bonn

vorgelegt von

Jorge de Heuvel

aus

Neuss

Bonn, Juli 2025

Angefertigt mit Genehmigung der Mathematisch-Naturwissenschaftlichen
Fakultät der Rheinischen Friedrich-Wilhelms-Universität Bonn

Gutachterin & Betreuerin:	Prof. Dr. Maren Bennewitz Rheinische Friedrich-Wilhelms-Universität Bonn
Gutachter:	Prof. Dr. Abhinav Valada Albert-Ludwigs-Universität Freiburg
Tag der Promotion:	30. Oktober 2025
Erscheinungsjahr:	2026

Acknowledgements

Without the support of many people, this dissertation would not have been possible in its current form.

First, I would like to sincerely thank my advisor and doctoral supervisor, Maren Bennewitz, for her long-standing mentorship at the Humanoid Robots Lab and her unwavering trust. I particularly appreciated the freedom she gave me in choosing research topics, which allowed me to pursue my own scientific ideas and grow professionally.

I also thank Abhinav Valada for serving as the second reviewer, and Aimee van Wynsberghe for her commitment to the examination committee. My thanks go as well to Matthias Hullin for taking on the role of committee chair.

I was fortunate to be part of an inspiring and collegial research environment at the Humanoid Robots Lab. Therefore, I am grateful to all colleagues and friends who made the institute a second home over the course of nearly four and a half years. I would also like to highlight the diverse and rewarding collaborations with my co-authors, through which I was able to grow both scientifically and personally, and for which I am deeply thankful.

I would especially like to thank my Master's students Xiangyu Zeng, Weixian Shi, Florian Seiler, and Hendrik Surmann, whose work made significant contributions to publications that are, in part, included in this dissertation. I also thank Nathan B. Corral, Tharun Sethuraman, and Aftab Akhtar for their reliable and dedicated work as student assistants on various projects, and for their resilience in the face of the unusual hours often required before submission deadlines. For enabling a research stay at KTH Stockholm in the summer of 2024, I would like to thank Iolanda Leite and all others involved.

For proofreading this dissertation, I would like to thank Marvin Klingner and Marlene Wessels for their support.

My heartfelt thanks go to my partner Marlene Wessels for her unwavering support throughout the entire PhD journey, especially for her patience when I spent weekends, odd hours, or even vacations working on publications instead of enjoying our well-deserved time together. I thank my mother, Ulrike de Heuvel, and my father, Ulrich Freier, for their constant support and their trust in my academic path.

Furthermore, I would like to thank Thomas Schmidt for his dependable IT support and Petra Zitzmann for her assistance with administrative matters.

For funding my research and conference travels, I would like to thank both the German Research Foundation (FOR 2535 "Anticipating Human Behavior," Grant BE 4420/2-2) and the Lamarr Institute for Machine Learning and Artificial Intelligence.

Abstract

Robots are increasingly moving from industrial applications into everyday human environments such as healthcare, households, and public spaces. In these interactive and personal contexts, successful human-robot interaction (HRI) critically depends on robots' abilities to interpret, reflect, and adapt to individual human preferences. Yet traditional robot navigation methods, though reliable in structured environments, generally fail to capture and reflect nuanced user preferences, resulting in suboptimal user experience, reduced trust, and limited acceptance.

To address these shortcomings, this thesis presents a comprehensive approach toward personalized, learning-based robot navigation. It specifically focuses on four critical aspects: (1) efficient and intuitive collection of human preferences, (2) balancing user preference reflection with robot navigation goals, (3) deriving expressive sensor representations suitable for dynamic environments, and (4) ensuring adaptability and transparency in HRI once deployed on a robot.

First, user preferences are captured using intuitive interfaces and efficient learning frameworks. We introduce a virtual reality (VR) demonstration interface, enabling users to sketch robot navigation trajectories with high intuitiveness. The VR interface is complemented with a hybrid reinforcement learning (RL) and behavioral cloning (BC) framework that requires only few demonstrations. We confirm through a user study that the personalized controller outperforms non-aligned baseline approaches, with users reporting that their preferences were better reflected. Besides demonstration-based approaches, we also optimize the preference collection through RL from human feedback (RLHF). We introduce the novel query generation approach "EnQuery" based on policy ensembles, maximizing the information gain in low-query regimes, while providing trajectory options with common start and goal reference points. EnQuery subsequently drives a user study that compares immersive VR and conventional 2D video interfaces for preference collection. Here, we find effects of the interface modality on user experience, preference consistency, and policy alignment.

Second, the thesis develops and validates learning architectures that balance the trade-off between user preference reflection and robot task completion. The proposed hybrid RL+BC learning framework internalizes user preferences while preserving goal-directed performance. To quantify the quality of preference reflection in navigation trajectories, we introduce a new metric that is based on the Fréchet Distance.

Third, we address the challenge of sensor representation for robust navigation in dynamic, human-populated environments. A depth vision-based perception pipeline employing a variational autoencoder and motion prediction compresses sensor observations into latent states, capturing both scene details and the user for effective personalized policy learning. In parallel, a spatiotemporal attention mechanism paired with a novel 2D lidar state representation improves obstacle avoidance and foresight in dy-

namic human environments over state-of-the-art baselines.

Fourth, the thesis advances the adaptability and transparency of learning-based robot navigation policies. To accommodate adaptability to evolving user preferences, a multi-objective RL framework facilitates principled post-deployment tuning of demonstration reflection and other navigation objectives. For improved transparency between robot and user, an explainable artificial intelligence (AI) interface in VR is developed, visually grounding navigation policy attribution scores semantically in scene context. The approach communicates internal decision-making of black-box neural network policies in an intuitive manner and thereby improves non-expert users' objective and subjective understanding of robot behavior.

These contributions are validated through extensive simulation studies, user experiments, and real-robot deployments. The findings demonstrate that preference-reflecting, learning-based navigation is achievable, robust, and perceived as superior to classical approaches by users. The insights regarding interface modality, interaction sample efficiency, sensor abstraction, and explainability inform the design of future user-centric robotic systems. In summary, this thesis establishes principled methods for navigation preference collection, learning, and behavior explanation, advancing the state-of-the-art towards seamless, preference-aware HRI in daily life.

Table of Contents

1 Introduction

The field of robotics is developing rapidly, expanding from primarily human-free industrial automation settings into human-centric domains such as public spaces, healthcare, and households. Our society will increasingly rely on robots not only due to societal workforce demand through population aging [1], but also because of the robots' economic and comfort benefits in assisting with daily tasks [2]. So, with the foreseeable emergence of robots as everyday helpers in human-shared household environments and healthcare roles, which represent intimate and interactive contexts, the robots' success and acceptance depend critically on their ability to engage appropriately and intuitively with human users [3].

Human-robot interaction (HRI) has become a critical research area for ensuring the robot's practical utility and societal acceptance. As opposed to classical robotics that optimizes for accuracy and efficiency, HRI requires robots to interpret and respond dynamically to human social norms, expectations, and individual preferences [4]. Human-aware navigation represents a core sub-field of HRI, as spatial behavior is a very salient aspect of mobile robots through which humans assess a robot's social competence. For HRI and the general acceptance among users, the robot's ability to address people's preferences will become an increasingly important success criterion [4], [5], [6], [7].

To understand how robots might achieve such preference-reflecting behavior, we can take inspiration from natural human interactions. The seamless coexistence between people interacting and sharing space with one another relies on the human ability to account for the preferences of others. Given a specific (joint) task or goal, humans align their actions accordingly, aiming to achieve optimal coexistence with those involved [8], [9]. To do so, they draw on past experiences and knowledge of others' preferences and mannerisms, which allows them to adapt their behavior based on context, situation, and necessity [10]. For instance, a person's comfort distance with someone may be smaller when jointly cleaning a kitchen as part of a team effort, but larger when encountering the same person while relaxing in a living room.

Transferring the insights from human-human interactions to HRI, a fundamental challenge in achieving comfortable interaction is the alignment of robot navigation strategies with subjective human preferences [11], [12]. These subjective preferences may manifest in individual social norms, personal space preferences or proxemics [13], and context- and scene-related behavior expectations towards the robot. Yet, user preferences might be influenced by personality traits [14], cultural background, past experiences, attitudes, and current intentions [4]. Even if the robot completes its task technically correct, failure to adapt to user preferences may result in discomfort [15], disruption [16], reduced trust among users [17]. Therefore, a one-fits-all approach is potentially too short-sighted and navigation behavior that works well for one person might be inappropriate or uncomfortable for another. For instance, while one person

may prefer the robot to take the shortest path and pass directly in front of them, another one may expect it to move aside, finding a close frontal passage uncomfortable or intrusive.

At the same time, users generally wish to have a certain degree of control over the technology they interact with [12]. Especially in HRI scenarios with repeated interactions, adaptation to the user is a key factor for long-term acceptance [14], [18]. Traditional robot navigation algorithms, while capable of obstacle avoidance and path optimization, fail to adequately reflect user-specific preferences, leading to potentially suboptimal user experiences [19], [20]. It is therefore essential to design navigation systems that capture, learn, and integrate human preferences. Advancing from traditional navigation approaches to preference-reflecting policies, navigation becomes a complex multi-objective optimization problem, where social appropriateness, preference reflection, efficiency, and safety must be jointly balanced. As an illustration, a robot may need to decide whether to take a longer, less efficient trajectory to avoid annoying a seated person in its vicinity and to respect their preference for minimal distraction, despite a shorter path being available.

In repeated interactions, humans construct mental models of others, allowing them to anticipate behaviors and adapt accordingly. These models are formed through a combination of verbal and non-verbal communication, observational learning, and context-sensitive inference [21]. Humans can recognize subtle feedback signals and dynamically adjust their behaviors to accommodate or anticipate others' needs. For a robot, however, interpreting subtle feedback signals from users is an inherently challenging task [22], [23], as it requires not only refined multimodal perception capabilities, but also profound emotional intelligence [7].

While we can take inspiration from human interpersonal interactions for robotic systems, they must rely on interaction modalities that are both informative and technically practical. Building accurate internal representations of user preferences becomes more straightforward when leveraging explicit modalities, such as direct human feedback [24], [25], [26], [27], demonstrations [15], [24], [25], [27], [28], and verbal instructions [29], [30].

This thesis addresses the challenge of how robotic systems can capture, interpret, and adapt to user preferences in the context of navigation tasks carried out in the immediate vicinity of the user. Addressing this challenge requires consideration of several aspects: First, the design of interfaces through which users can express their preferences, including mechanisms that maximize the associated information gain. Second, the creation of data-efficient learning methods that integrate preference information while preserving the robot's task-specific objectives. Third, the derivation of information-dense sensor representations that capture the dynamic robot environment with obstacles and the user. Fourth, investigating how to improve HRI further by policies that remain adaptable to changing user preferences, and enhancing the users' perception and understanding of the robot behavior for improved HRI.

1.1 Towards Personalized Robot Navigation

Learning-based robot navigation is the central topic addressed in all the publications included in this thesis. But why are learning-based navigation and personalization so tightly connected? To approach this question, this section highlights the evolution and recent developments in robot navigation that enable the personalization of robot behavior to human preferences.

1.1.1 Traditional Approaches

Traditional non-learning-based robot navigation typically involves global path planning, paired with local obstacle avoidance to account for unknown and dynamic obstacles. The global path planning module has the task of providing high-level navigation cues or waypoints to a desired goal using a map of the environment. For occupancy grid maps [31], established planning algorithms include A* [32] and Rapidly-exploring Random Trees (RRT) [33], [34]. To follow these planned paths reliably, the robot must accurately estimate its position relative to the map, typically achieved via Monte Carlo Localization (MCL) [35].

The local obstacle avoidance relies on high-level guidance from the global planner and generates collision-free trajectories in the robot's immediate vicinity. Established methods include the Dynamic Window Approach (DWA) [36] or potential field methods [37], [38]. This navigation skill is crucial in human-populated or cluttered environments, where unknown static (e.g., non-fixed objects, clutter) or dynamic (e.g., walking humans, navigating robots) obstacles require real-time sensing and rapid decision-making to ensure safe and effective navigation.

The approaches presented in this thesis are not traditional but learning-based, yet use methods such as A* for high-level guidance to local learning-based obstacle avoidance (Chapter 4), Adaptive MCL for localization (Chapter 4), or DWA as a non-personalized obstacle avoidance baseline (Chapter 2).

1.1.2 Human-Aware and Social Robot Navigation

While the traditional methods offer reliable performance in static or moderately dynamic settings, they are often inadequate for navigation in human-populated environments where social context and motion prediction are critical [39], [40]. In such scenarios, pedestrians represent dynamic agents with individual goals, preferences, and social norms, which challenge the assumptions of traditional obstacle avoidance frameworks, because they treat all obstacles as non-interactive. This has led to the development of social navigation approaches that explicitly incorporate models of human behavior, typically by integrating pedestrian trajectory prediction, proxemics, or interaction-aware planning into the robot's control loop. Methods such as social force models [41], social cost maps [42], (reciprocal) velocity obstacles [43], [44], [45] with human-aware constraints, and optimization-based planners such as elastic band [46], [47] have been pro-

posed to ensure safe and socially acceptable navigation in shared spaces. Another social navigation paradigm is learning to reduce the influence on nearby humans through the robot [48]. These approaches typically rely on explicit pose and kinematics data of pedestrians, but accurately estimating this from the robot's sensors is a challenge on its own.

Human-aware navigation is central to this thesis, with most works focusing on personalized navigation by designing policies that interact with and adapt to a single, preference-expressing user. Additionally, Chapter 2 employs a social cost approach as a baseline.

1.1.3 Advent of Reinforcement Learning Policies

While classical methods offer robust performance in structured environments, they often struggle to capture the nuanced and context-dependent behaviors required for safe and socially compliant navigation in human-populated spaces [39]. To address these limitations, recent research has increasingly focused on reinforcement learning-based (RL) navigation controllers [49], [50], which allow robots to learn navigation behaviors directly from experience via reward signals and demonstration [51], [52], [53]. Deep learning and scalable computation have made RL practical, with stable algorithms and simulators enabling efficient, reproducible training [54], [55]. The policies of RL are typically implemented as deep neural networks (hence the term deep RL, short DRL), which support flexible, generalizable behaviors across diverse scenarios [54]. By directly coupling sensory observations with policy learning, these controllers operate in an end-to-end manner [56], eliminating the need for hand-crafted intermediate representations. This is particularly advantageous in partially observable, dynamic environments, where conventional rule-based systems lack the adaptability to handle uncertainty and interactive agents. The aforementioned characteristics of DRL offer great potential for human-aware navigation, as they enable robots to learn context-sensitive behaviors in an end-to-end manner. Leveraging this capacity, all approaches presented throughout this thesis integrate DRL policies.

1.1.4 Learning Human Preferences

While traditional RL methods typically rely on low-dimensional, hand-crafted state representations and discrete action spaces, they struggle to capture the complexity and variability of human preferences. In contrast, the capacity of RL policies and neural networks in general makes them ideal candidates to learn and interpret nuanced human preferences [24], [57], [58]. Rather than relying on hand-tuned cost functions, methods such as inverse reinforcement learning (IRL) [59], RL from human feedback (RLHF) [60], and RL combined with behavioral cloning (BC) [61] enable implicit preference learning.

In short, IRL infers reward structures from expert demonstrations, capturing latent intent behind observed behavior. IRL requires high-quality demonstrations and can suffer from ambiguity in inferred rewards but provides a generalizable and interpretable

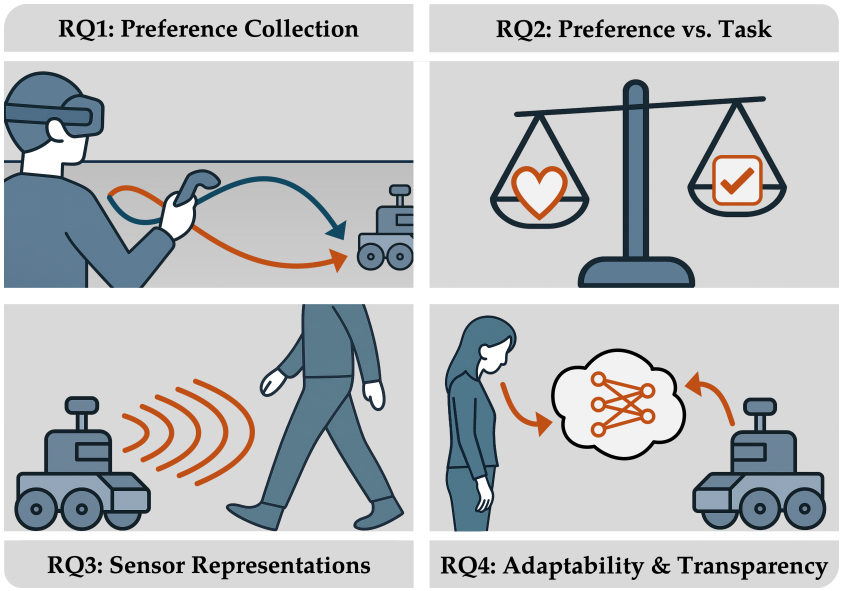


Figure 1.1: Visual summary of the four overarching research questions (RQs) addressed throughout this thesis. RQ1 focuses on the efficient collection of human preferences via intuitive interfaces. RQ2 investigates the balance between adherence to user preferences and task efficiency in robot navigation. RQ3 addresses the design of compact and expressive sensor state representations for RL-based navigation in dynamic environments. RQ4 examines adaptability to changing user preferences and user-transparent explanations to enhance long-term usability and user understanding in HRI. See Section 1.2 and Table 1.1 for details and chapter-wise mapping.

reward model. In contrast, RLHF leverages comparative feedback to align policy behavior with subjective human judgments. It avoids the need for demonstrations but relies on noisy and potentially inconsistent human feedback. Lastly, RL+BC enhances the learning process by supporting initial exploration with demonstrated trajectories while maintaining generalization to unseen states through reinforcement. Its effectiveness depends on demonstration coverage and careful tuning between exploration and imitation.

All three methods allow robot policies to internalize preferences directly from data, making them adaptable to individual users and contexts. In other words, preferences are not just modeled but actively integrated into the learning process.

In this thesis, all three preference learning techniques are addressed. Chapters 2 and 3 utilize RL+BC, Chapters 7 and 6 deal with RLHF for personalization of robot navigation behavior, and Chapter 5 leverages IRL.

1.2 Overarching Research Questions

To better understand the research challenges addressed in this thesis, four publication-overarching research questions (RQs) are discussed in the subsequent sections, see Figure 1.1. In short, they cover 1) efficient and intuitive collection of human preferences, 2) balancing user preference reflection with robot navigation goals, 3) deriving expres-

sive sensor representations suitable for dynamic environments, and 4) ensuring user-centric adaptability and transparency in HRI. The linkage of each individual publication to these RQs is briefly outlined, also compare Table 1.1. Since not all publications address each question equally, only the relevant ones are referenced accordingly. Note that additionally, the final summary of key contributions will in Section 1.3 will refer to the RQs sorted by publications.

	RQ1 Efficient Preference Collection	RQ2 Task/ Preference Balancing	RQ3 Sensor Representations	RQ4 Adaptability & Transparency
Chapter 2	✓	✓		
Chapter 3	✓	✓	✓	
Chapter 4			✓	
Chapter 5		✓		✓
Chapter 6	✓			
Chapter 7	✓			
Chapter 8				✓
Chapter 9	✓	✓		✓

Table 1.1: Mapping of thesis chapters to the overarching research questions.

1.2.1 RQ1: Efficient Human Preference Collection

A key requirement for personalized navigation is the efficient collection of human preferences via some sort of interface between the robot's control system and the user. In robotics, humans express preferences by demonstrating how a specific task should be performed, by providing feedback on the robot's actions, and lastly, by verbal explanation. This thesis directly tackles demonstration and feedback, while verbal preference expression is discussed as an outlook in context of the recent advent of large language models (LLMs) in Chapter 9.

While collecting preferences on real robots is feasible [15], it poses challenges such as limited repeatability, high time and resource costs, and reduced experimental control. A promising solution is computer-based interfaces that provide a high degree of experimental control over the collection process at larger repetition numbers [62] also in conjunction with high fidelity robotic simulators [63], [64].

However, perspective gaps between preference expression and how the final robot behavior is perceived and evaluated by the user, especially in demonstration or feedback tasks, can reduce preference accuracy [65], [66]. For example, a user might teleoperate the robot using a game controller while watching a robot-centric video stream. Although the controller may feel intuitive, the video stream shows a viewpoint different from the user's own, and depth perception is limited. This mismatch may impair the user's understanding of the scene, potentially leading to demonstrations that do not accurately reflect their preferences.

Immersive virtual reality (VR) technology can overcome the gap between the demon-

stration and evaluation perspective between robot and user. Chapters 2 and 3 employ VR to collect user preferences via demonstration. Specifically, non-expert users draw desired robot trajectories onto the floor of a VR scene by pointing with a handheld controller. Chapter 7 compares user preferences collected via feedback in immersive VR with those from a 2D video interface.

Another challenge of the preference collection process is optimizing the information gain through each interaction with the user [52], [67], thereby reducing the number of interactions required and therefore user fatigue. To maximize the utility of demonstration-based preferences [52], Chapters 2 and 3 apply data augmentation to improve learning from only a few demonstrations. Chapter 6, on the other hand, introduces a querying approach for RLHF, addressing the open challenge of optimal query generation [12], [67], [68].

RQ1 summarizes to how preference collection interfaces can be designed to maximize intuitiveness, efficiency, and information gain.

1.2.2 RQ2: Preference Reflection vs. Task Efficiency: A Balancing Act

When personalizing the robot's behavior for a given task, a balance between task execution and personalization is required [69], a balancing act between multiple, possibly conflicting objectives. Ultimately, the robot is expected to fulfill its original task, or in terms of navigation, reach its original goal while reflecting preferences.

For example, a user might prefer longer paths, such as having the robot follow walls closely. While the policy can learn this from demonstrations, it must also generalize to unseen scenarios. How can we design a navigation system that fulfills the task while respecting preferences wherever possible?

To tackle this problem, Chapters 2 and 3 employ a hybrid RL+BC framework, along with a specifically designed reward system to balance general task execution with personalized behavior. To also quantify the quality of preference reflection, Chapter 3 introduces a novel metric that helps in evaluating the balance between preference adherence and goal-directed navigation.

While the aforementioned approaches strike a one-shot balance at personalization, Chapter 5 tackles the balancing act of personalization explicitly as a multiobjective learning problem. This approach allows us to fluently interpolate a single policy between goal pursuit and demonstration reflection.

RQ2 summarizes as how to design and evaluate robot policies that balance task completion with adherence to individual user preferences, especially when these objectives are in conflict.

1.2.3 RQ3: Sensor Representations for RL-Based Navigation in Dynamic Environments

For RL-based robot navigation systems, effective sensor data representations are crucial, especially in dynamic environments [70]. While vision sensors such as depth cameras

offer rich spatial information, they also pose challenges for RL due to high dimensionality and occlusions. Depth-based navigation must compress complex visuals into a compact, expressive representation that captures scene dynamics and supports preference anchoring. Otherwise, performance degrades, especially near moving humans.

To address this challenge, Chapter 3 presents a perception pipeline employing a variational autoencoder coupled with a motion predictor for dynamic navigation scenarios, such that depth images are effectively compressed into latent representations conducive to learning personalized navigation policies. For non-personalized, foresighted navigation among humans, Chapter 4 leverages a spatiotemporal attention mechanism on 2D lidar sensor data.

RQ3 summarizes how to design and evaluate compact yet expressive sensor state representations that enable RL-based navigation to robustly handle dynamic obstacles.

1.2.4 RQ4: Adaptability and Transparency in Robot Decision-Making

The final research question concerns the interaction between a (personalized) navigation policy and the user once the policy is deployed. What measures improve HRI at this stage? Firstly, user preferences can be dynamic and change over time [14], [71], or demonstrations provided in one context may not be applicable to all situations. Personalized policies thus require some form of adaptability after deployment. Chapter 5 addresses this with a multi-objective RL (MORL) approach, which enables post-training policy adaptation.

Second, transparency is essential for HRI, as users were found to have a clear interest in robots capable of explaining their navigational decisions [72]. However, the black-box nature of end-to-end neural policies [73] can hinder user understanding, impair users' mental model formation, and reduce user trust. To address this challenge, explainability methods can enhance transparency of policy behavior for users. For non-expert users, those explanations need to be communicated in a user-friendly and cognitively accessible manner. Chapter 8 introduces a VR interface that augments the robot's perception and reasoning by conveying explainability attribution scores and lidar sensor perception to non-expert users.

RQ4 summarizes as how to enable adaptability to changing user preferences and user-digestible behavior explanations for the robot policies to improve long-term usability and user understanding in HRI.

1.3 Key Contributions

Chapter 2 introduces a novel intuitive VR demonstration interface combined with a RL framework, enabling non-expert users to intuitively specify personalized navigation preferences through limited demonstration data. In a user study, the resulting personalized navigation controller demonstrates superior comfort and alignment with subjective user preferences, significantly outperforming traditional navigation methods, and

seamlessly transfers from simulation to real robotic platforms. (RQ1, RQ2)

Chapter 3 proposes a personalized depth vision-based robot navigation method learned from VR demonstrations in dynamic environments. The approach leverages a perception pipeline consisting of a variational autoencoder (VAE) and a motion prediction module to compress depth observations into latent state representations for a personalized DRL policy. A new metric based on the Fréchet Distance quantitatively evaluates the degree of user preference reflection, complemented by extensive qualitative and quantitative analyses validating the method's capability to capture and generalize user-specific navigation behaviors effectively from depth vision. (RQ1, RQ2, RQ3)

Chapter 4 develops a lightweight, lidar-based robot navigation controller enhanced by a novel spatiotemporal attention mechanism. The introduced lidar state representation, temporal accumulation group descriptor (TAGD), reveals dynamic obstacles over static ones, and improves DRL-based local obstacle avoidance without explicit obstacle tracking. The spatiotemporal attention mechanism selectively processes sensory data, significantly enhancing navigation robustness and foresight in dynamic pedestrian-rich environments. (RQ3)

Chapter 5 presents a demonstration-enhanced MORL framework enabling robot navigation policies to adapt dynamically to changing user preferences post-training without additional retraining. Demonstrations are incorporated as a tunable objective, facilitating continuous policy refinement in response to evolving preferences. The approach is rigorously validated through simulation experiments, which demonstrates adaptability, robustness, and generalization, and leads to successful real-world deployment. (RQ2, RQ4)

Chapter 6 introduces EnQuery, a novel query-generation method employing ensembles of policies to generate behaviorally diverse navigation trajectory queries for RLHF. Through a regularization term ensuring diversity, EnQuery enhances preference alignment efficiency, achieving superior alignment performance even with minimal user queries compared to state-of-the-art baselines. Additionally, a novel visualization scheme comprehensively captures learned navigation behaviors across the scene from a top-down perspective. (RQ1)

Chapter 7 systematically evaluates the influence of VR and conventional 2D interfaces on the collection of human navigation preferences for robot policy alignment. Through the dataset collection and analysis of over 2,000 preference queries from a user study, the chapter identifies significant interface modality-driven differences in user experience, preference consistency, and resulting navigation policy effectiveness. This comparative analysis underscores critical trade-offs between interface immersion, perception fidelity, and reliability of captured user preferences, guiding future interface selection decisions in preference-based robot learning. (RQ1)

Chapter 8 presents an immersive virtual reality interface integrating semantic explainable AI (XAI) projections with lidar visualizations to enhance transparency and trust in robot navigation. By grounding abstract neural network attribution scores se-

mantically within scene contexts, the interface significantly improves non-expert users' objective understanding and subjective predictability of robot behaviors, as empirically confirmed by a detailed user study. This novel integration of semantic XAI and sensor visualization effectively bridges the gap between abstract policy explanations and human interpretability, advancing user-centric explainability in HRI. (RQ4)

Chapter 9 provides a structured overview of the current advancements in the field of robotics through foundation models, specifically the possibilities of large (vision) language model interfaces and policies for personalization.

Finally, **Chapter 10** reflects on the findings of Chapter 2 to 8 and contextualizes them against the background of recent advancements in foundation models and other developments relevant to robot personalization.

1.4 Publications

Parts of this thesis have been published in international journals and conference proceedings, and the list below provides a chapter overview of the individual publications. The chapters represent slightly revised versions of the published papers. Where appropriate, citations and contextual discussions were updated to reflect recently published literature, including works that have appeared after the original publication dates.

- Chapter 2
  [74] **J. de Heuvel**, N. Corral, L. Bruckschen, and M. Bennewitz, "Learning Personalized Human-Aware Robot Navigation Using Virtual Reality Demonstrations from a User Study," in *Proceedings of the IEEE International Conference on Human & Robot Interactive Communication (RO-MAN)*, 2022.

- Chapter 3
  [75] **J. de Heuvel**, N. Corral, B. Kreis, J. Conradi, A. Driemel, and M. Bennewitz, "Learning Depth Vision-Based Personalized Robot Navigation From Dynamic Demonstrations in Virtual Reality," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023.

- Chapter 4
  [76] **J. de Heuvel**, X. Zeng, W. Shi, T. Sethuraman, and M. Bennewitz, "Spatiotemporal Attention Enhances Lidar-Based Robot Navigation in Dynamic Environments," in *IEEE Research and Automation Letters*, 2024.

- Chapter 5
  [77] **J. de Heuvel**, T. Sethuraman, and M. Bennewitz, "Demonstration-Enhanced Adaptable Multi-Objective Robot Navigation," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2025.

- Chapter 6
  [78] **J. de Heuvel**, F. Seiler, and M. Bennewitz, "EnQuery: Ensemble Policies for

Diverse Query-Generation in Preference Alignment of Robot Navigation," in *Proceedings of the IEEE International Conference on Human & Robot Interactive Communication (RO-MAN)*, 2024.

- Chapter 7
  [79] **J. de Heuvel**, D. Marta, S. Holk, I. Leite, and M. Bennewitz, "The Impact of VR and 2D Interfaces on Human Feedback in Preference-Based Robot Learning," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2025.

- Chapter 8
  [80] **J. de Heuvel**, S. Müller, M. Wessels, A. Akhtar, C. Bauckhage, and M. Bennewitz, "Immersive Explainability: Visualizing Robot Navigation Decisions through XAI Semantic Scene Projections in Virtual Reality," in *Proceedings of the IEEE International Conference on Human & Robot Interactive Communication (RO-MAN)*, 2025.

Supplemental material is available for selected publications. A comprehensive list can be found in the supplemental material section of the thesis appendix.

Publications Not Covered By This Thesis

The following publications were (co-)authored during the period of employment as a research associate. However, they are not included within the scope of this thesis.

- **J. de Heuvel**, W. Shi, X. Zeng, and M. Bennewitz, "Subgoal-Driven Navigation in Dynamic Environments Using Attention-Based Deep Reinforcement Learning," in *Proceedings of the IEEE/RSJ International Conference on Advanced Robotics (ICAR)*, 2023.

- M. Dawood, N. Dengler, **J. de Heuvel**, and M. Bennewitz, "Handling Sparse Rewards in Reinforcement Learning Using Model Predictive Control," in *Proceedings of the IEEE International Conference on Robotics & Automation (ICRA)*, 2023.

- B. Kreis, R. Menon, B. K. Adinarayan, **J. de Heuvel**, and M. Bennewitz, "Reactive Correction of Object Placement Errors for Robotic Arrangement Tasks," in *Proceedings of the International Conference on Intelligent Autonomous Systems (IAS)*, 2023.

- B. Kreis, N. Dengler, **J. de Heuvel**, R. Menon, H. D. Perur, and M. Bennewitz, "Compact Multi-Object Placement Using Adjacency-Aware Reinforcement Learning," in *Proceedings of the IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, 2024.

- S. Agrawal, M. Wessels, **J. de Heuvel**, J. Kraus, and M. Bennewitz, "Sound Matters: Auditory Detectability of Mobile Robots," in *Proceedings of the IEEE International Conference on Human & Robot Interactive Communication (RO-MAN)*, 2024.

- E. Schlachhoff, N. Dengler, L. Van Holland, P. Stotko, **J. de Heuvel**, R. Klein, and M. Bennewitz, "RHINO-VR Experience: Teaching Mobile Robotics Concepts in an Interactive Museum Exhibit," in *Proceedings of the IEEE International Conference on Human & Robot Interactive Communication (RO-MAN)*, 2024.
- H. Surmann, **J. de Heuvel**, and M. Bennewitz, "Multi-Objective Reinforcement Learning for Adaptable Personalized Autonomous Driving," in *Proceedings of the 12th European Conference on Mobile Robots (ECMR)*, 2025.
- M. Wessels, **J. de Heuvel**, L. Müller, A. L. Maier, M. Bennewitz, and J. Kraus, "Auditory Localization and Assessment of Consequential Robot Sounds: A Multi-Method Study in Virtual Reality," in *Proceedings of the IEEE International Conference on Human & Robot Interactive Communication (RO-MAN)*, 2025.

1.5 Collaborations

This thesis includes work conducted in collaboration with other researchers and institutions. The following outlines the contributions and responsibilities of all parties involved.

- Chapter 2: The broad idea of a VR-based interface originated the DFG proposal for the Research Unit 2535 "Anticipating Human Behavior", co-authored by Lilli Bruckschen. I extended this idea into a demonstration platform for personalized robot navigation, developed the learning architecture, and led the project execution. Nathan B. Corral supported with data collection.
- Chapter 3: I conceived the project, executed all core work, and wrote the paper. The "Deviation-Aware Fréchet Distance" metric was co-developed with Jacobus Conradi. Benedikt Kreis supported the literature review. Nathan B. Corral assisted with the iGibson setup.
- Chapter 4: As a student assistant under my supervision, Xiangyu Zeng conceptualized the TAGD lidar representation and conducted initial experiments. The PyBullet-based simulation framework was a joint development by her and Weixian Shi within the scope of their master's theses, both of which I supervised. I conducted all subsequent evaluations and real-world experiments and authored the manuscript.
- Chapter 6: The EnQuery approach is the result of a master's thesis project by Florian Seiler under my supervision. We jointly refined the evaluation for publication; I finally authored the publication.
- Chapter 5: I conceived and implemented the demonstration-infused MORL approach for adaptable preference modeling. Tharun Sethuraman assisted with real-world experiments.

- Chapter 7: This collaboration involved the Humanoid Robots Lab (Bonn) and the Division of Robotics, Perception and Learning (KTH). I initiated the idea of comparing interface modalities and led the user study design and execution. Together with Daniel S. Marta and Simon Holk, the project idea was refined. Prof. Iolanda Leite facilitated the pilot study at KTH. Tharun Sethuraman supported the interface implementation. Simon Holk and Daniel S. Marta contributed reward model code and RLHF-related sections.
- Chapter 8: This work was a collaboration between three research groups. I developed the VR interface concept for scene-semantic XAI projections and coordinated the project. Sebastian Müller refined the XAI method and contributed to related work and methodology section. Marlene Wessels assisted with the user study and statistical analysis. Aftab Akhtar implemented the Unity interface and supported data collection.

2 Learning Personalized Human-Aware Robot Navigation Using Virtual Reality Demonstrations from a User Study

Abstract

For the most comfortable, human-aware robot navigation, subjective user preferences need to be taken into account. These preferences need to be collected in an efficient and user-intuitive manner, to subsequently shape the navigation policy of the robot. This chapter presents a novel reinforcement learning framework to train a personalized navigation controller along with a virtual reality demonstration interface. Using the immersive interface, users can draw robot trajectories on the floor using a handheld controller, enabling spatially grounded demonstration without requiring expert knowledge. The conducted user study provides evidence that our personalized approach significantly outperforms classical approaches with more comfortable human-robot experiences. We achieve these results using only a few demonstration trajectories from non-expert users, who predominantly appreciate the intuitive demonstration setup. As we show in the experiments, the learned controller generalizes well to states not covered in the demonstration data, while still reflecting user preferences during navigation. Finally, we transfer the navigation controller without loss of performance to a real robot.

2.1 Introduction

Robot personalization to specific user preferences will become a key factor for comfortable and satisfying human-robot interactions, as robots find their way into our everyday lives. Hence, the number one goal should be a naturally collaborative experience between users and the robot. Harmonic human-robot interactions build trust and satisfaction with the user [4], whereas negative interaction experiences can quickly lead to frustration [81]. As users might have personal preferences about specific aspects of the robot's behavior that define the personal gold standard of interaction, falling short of preference reflection could lead to such negative interaction experiences.

Where mobile household robots navigate in the vicinity of a human, basic obstacle avoidance approaches fail to capture individual user preferences. While collision avoidance is undoubtedly crucial during navigation, the navigation policy should furthermore be human-aware and take into account user preferences regarding proxemics [81] and privacy, compare Figure 2.1 (bottom). Subjective preferences may vary depending on the environment and social context, e.g., navigation preferences could reflect in the

This chapter is a revised and updated version of the peer-reviewed publication [74]. Refer to Section 1.4 for details.

robot's approaching behavior, or always driving in front or behind the human. In addition, following a certain speed profile and maintaining a certain distance from humans and other obstacles in the environment might play a role. The resulting navigation objective for the robot is to reach the navigation goal, not necessarily by only following the shortest path, but also by taking personal robot navigation preferences into account.

Recent advances in learning socially aware navigation behavior from human demonstrations have been made with inverse reinforcement learning, where the parameters of a proxemics-encoding reward function were inferred [15]. Influenced by the initial shaping of the reward function [82], such approaches lack the ability for navigation style personalization beyond the scope of the reward function. For smooth navigation, reinforcement learning (RL) based continuous control has led to promising results on mobile robots [56], [83]. Furthermore, off-policy RL methods can be complemented with demonstration data to greatly improve learning speed on a given task, even outperforming the resourcefulness of the original demonstrations [53]. However, RL robot navigation policies learn the most efficient trajectories to the goal. These trajectories do not necessarily reflect the original demonstration behavior, which contains user preferences. To more precisely imitate behavior from demonstrations, behavioral cloning (BC) can be used [84]. However, the final policy is limited by the quality and amount of demonstration data [52]. The dataset would need to cover most of the state space to generalize fluently in unseen environments. This poses a problem, as human demonstrators can only provide limited amounts of demonstration data due to their finite patience [85]. With regard to our overarching RQ1 (cf. Chapter 1.2.1) the question crystallizes, how do we efficiently record personal preferences and teach them to the robot, without being limited by the quality and quantity of demonstrations.

In order to solve the aforementioned challenges, this chapter proposes a novel navigation learning approach together with an immersive virtual reality (VR) interface to intuitively demonstrate robot navigation preferences by drawing trajectories onto the floor with a handheld controller, see Figure 2.1. Importantly, the interface does not require expert-level knowledge of robotics, facilitating personalized navigation for a wide range of users. Our demonstration process is time-efficient, as only few demonstrations are required. The demonstrations are leveraged to successfully train a personalized human-aware navigation controller, by combining deep reinforcement learning and behavioral cloning. We show that our navigation policy closely reflects user preferences from only a few demonstrations. But at the same time, it generalizes to unseen states. In an extensive user study, we evaluate the personalized navigation behavior against classical navigation approaches both in VR and on a real robot.

The threefold **main contributions** of our study are:

- A VR demonstration interface for teaching navigation preferences to robots intuitively.
- Learning a user-personalized, context-based navigation policy based on the com-

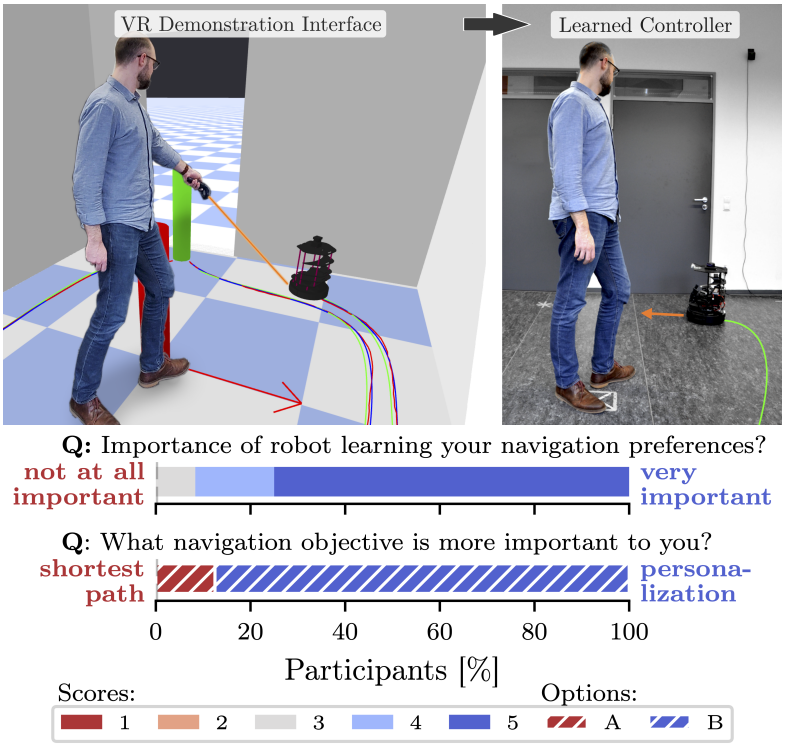


Figure 2.1: **Top:** We propose a virtual reality (VR) interface to intuitively demonstrate robot navigation preferences by drawing trajectories on the floor with a handheld controller. **Bottom:** User study survey results on the importance of personalized navigation behavior. Participants strongly expressed their preference for personalization of robot navigation behavior, even at a possible cost of longer trajectories.

bination of RL and BC.

- An interactive user study recording user-specific navigation preferences, evaluating both the presented interface and learned personalized navigation policies.

2.2 Related Work

Extensive research has been done on both human-aware navigation [40] and on robot personalization [4], [86], but only a few works can be found at the intersection of both disciplines. Wilde *et al.* align robot path planning in a known environment to user preferences, by letting users rank different path candidates. While their approach operates in the global path planning domain, our work will focus on local human-aware navigation.

Various studies adapt human-aware navigation behavior either by learning or inferring cost-maps [15], [87], [88]. These cost-maps usually encode proxemics or environmental characteristics. To improve navigation in human-robot interaction based on context, Bruckschen *et al.* [89] leveraged previously observed human-object interactions to predict human navigation goals, which in turn enables foresighted robot navigation and assistance. Other studies aimed to distinguish between different environment types

as context in order to automatically adjust the robot's navigation behavior [90], [91]. In our work, we consider different environment scenarios as context.

Posture and gait can serve as informative non-verbal cues for robot navigation, as the following works show. Luber *et al.* [92] studied the angle of approach between two individuals to improve human-aware navigation. Also, head orientation and gaze can represent predictors for social navigation [23]. Recently, Narayanan *et al.* [93] leveraged the human gait posture as social cues for foresighted robot navigation by predicting the human's navigation intent and emotion. To build upon the aforementioned findings, we as well take the human orientation into account.

To learn personal navigation preferences in a human-robot collaboration scenario from demonstrations, Kollmitz *et al.* [15] learned the parameters of a navigation reward function from physical human-robot interaction via inverse reinforcement learning. More specifically, the navigation reward function was learned from a user pushing the robot away to a desired distance. A limitation of this approach is the state space represented by a 2D grid map of the environment, making the approach unsuitable for larger and unknown environments. To overcome this limitation, our state space is robot-centric and continuous, focusing on the vicinity to the human and obstacles.

Xiao *et al.* [90] proposed using teleoperation demonstrations to learn context-based parameters of a conventional planner. Here, the reproduction of demonstration trajectories during navigation is limited by the capabilities of the conventional planner. To ensure a more distinct preference reproduction including certain trajectory profiles, we chose a deep learning-based controller.

To efficiently train a deep learning-based navigation controller for robot navigation via reinforcement learning, Pfeiffer *et al.* [83] utilized demonstration navigation data gathered from an expert planner algorithm. The demonstration data was used to pre-train the agent via imitation learning, followed by the reinforcement learning. In our work, we use a similar architecture for continuous control learning, but in contrast, we focus on human demonstrations of robot trajectories.

Virtual reality environments have been successfully deployed to simulate human-robot interactions [87], [94], offering a tool for realistic demonstration and evaluation. As a result, we chose to develop a VR interface that interactively records the user-demonstrated trajectories of a robot. These demonstrated trajectories provide the data required to learn user-specific robot navigation preferences. The VR interface enables a first-person experience of the navigating robot during demonstration, ensuring a realistic perception of proxemic aspects. In these regards, a clear benefit over, e.g., real world robot teleoperation is the easy separation of the demonstration and re-evaluation experience in simulation, enabling interactive replay of scenarios.

Since the publication of our original study, related approaches in the areas of virtual and augmented reality (VR/AR) and robot personalization have been presented. Immersive interfaces have recently been adopted for data collection pipelines in robotics. For instance, Moletta *et al.* collected user demonstrations of cloth folding to learn

garment-specific folding plans for a robot [95], and Zhang *et al.* used VR for the demonstration of socially acceptable navigation behavior to an autonomous agent [96]. VR and AR have also been employed for learning user preferences in navigation tasks. Nakaoka *et al.* leveraged VR to align a social force navigation model to user preferences [97], and Nigro *et al.* used augmented reality (AR) to collect proxemic preferences of elderly people for an approaching mobile robot[98]. In parallel, other works have introduced personalized robot navigation approaches interacting with users through natural language interfaces powered by large language models [99], [100], [101]. Finally, Wang *et al.* optimize the amount of preference data required for preference-reflecting navigation [102], a challenge also tackled in our work. Their learning framework is also hybrid, fusing explorations and human demonstrations obtained through a keyboard interface in a common buffer. While their demonstrations are reward-labeled through another feedback loop, we directly run them through a behavioral cloning loss. While differing in implementation, these contributions underline both the applicability of immersive technologies for safe and user-friendly data collection, and the alignment of robot behavior to individual user preferences as an emerging research trend.

2.3 Problem Definition and Assumptions

In this work, we consider a differential wheeled robot that has a local navigation goal and navigates in the vicinity of a single human. Our goal is to create a personalized robot navigation controller that adapts to user preferences by learning from demonstrations of robot trajectories that include a velocity profile. Hereby, we focus on local human avoidance taking into account user-specific preference. Both human and robot are interacting in the same room, which serves as context for the navigation behavior. We assume that the positions and orientations of the human, the robot, and all obstacles are known. All parameters above can play a role for the robot navigation preferences of the user and need to be reflected in the robot-centric state space.

2.4 Reinforcement Learning from Demonstrations

We adapted a twin-delayed deep deterministic policy gradient (TD3) architecture consisting of an actor and two critic networks [103]. TD3 was chosen for two reasons: i) It has a continuous action space allowing smooth robot control and ii) it is off-policy, thus is a perfect candidate for use with demonstration data. The actor network outputs two continuous robot control commands, i.e., forward and angular velocity. We introduce two modifications to classic TD3, similar to Nair *et al.* [104]: i) a behavioral cloning loss on the actor network and ii) a separate buffer to integrate demonstration data. The introduction of the behavioral cloning loss makes our approach a hybrid of reinforcement and imitation learning. Figure 2.2 depicts a schematic overview of our approach.

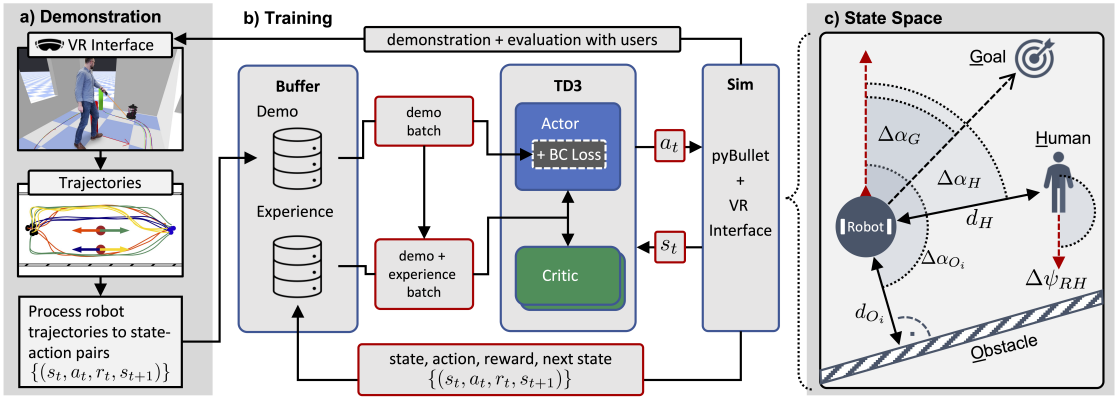


Figure 2.2: Schematic representation of the proposed architecture. **a)** Demonstration trajectories are drawn by the user and fed into the demonstration buffer. **b)** A TD3 reinforcement learning architecture with an additional behavioral cloning (BC) loss on the actor trains a personalized navigation policy for human-robot interaction with continuous control. The learned policy is then evaluated in VR and subsequently transferred to a real robot. **c)** The robot-centric state space captures the vicinity and orientation of the human and the obstacles as well as the goal direction.

2.4.1 Twin-Delayed Deep Deterministic Policy Gradient

Reinforcement learning describes the optimization of transitions from state $s_t \rightarrow s_{t+1}$ following a Markov Decision Process that result in a reward $r_t = r(s_t, a_t)$, by taking an action $a_t = \pi_\phi(s_t)$ at time step t with respect to a policy π_ϕ [105]. The tuples (s_t, a_t, r_t, s_{t+1}) are referred to as state-action pairs. The optimization objective is to maximize the cumulative return $R = \sum_{i=t}^{T} \gamma^{(i-t)} r_t$ of the γ-discounted rewards, onward from t. With TD3, we optimize the expected return

$$y_t = r_t + \gamma \min_{i=1,2} Q_{\theta_i^*} \left(s_{t+1}, \pi_\phi \left(s_{t+1}\right) + \epsilon_{\theta_i}\right), \tag{2.1}$$

while using the minimum of two critics $(Q_{\theta_1}, Q_{\theta_2})$ to prevent value overestimation. θ_i denotes the (network) parameters of critic i and ϕ those of the actor. The clipped Gaussian policy noise ϵ_{θ_i} stabilizes the Q-value estimation over similar state-action pairs and is controlled by the standard deviation $\sigma_{\epsilon_{\theta_i}}$

To ensure sufficient exploration, we add Gaussian noise from a process $\mathcal{N}$ with standard deviation σ_{ϵ_π} to the actions drawn from the actor, so that $a_t = \pi_\phi(s_t) + \mathcal{N}(0, \sigma_{\epsilon_\pi})$.

To update the critic θ_i, TD3 optimizes the loss

$$\mathcal{L}_{\theta_i} = \frac{1}{b} \sum_{j}^{b} \left(y_j - Q_i \left(s_j, a_j | \theta_i\right)\right)^2 \tag{2.2}$$

over all state-action pairs j in the batch of size b. The actor network parameters ϕ_π are

updated using the policy gradient:

$$\nabla_\phi J = \frac{1}{b}\sum_{j}^{b} \nabla_a Q_{\min}(s, a|\theta)|_{s=s_j, a=\pi(s)} \nabla_\phi \pi(s|\phi)|_{s_j} \tag{2.3}$$

For further details on the learning algorithm, please refer to [103] and [106].

The actor and critic networks share a feed-forward three-layered perceptron architecture with 256 neurons each. We normalize both the input (observation space for actor and critic) and output (action space for actor) of the networks, respectively.

2.4.2 Replay and Demonstration Buffer

In addition to TD3's standard experience replay buffer of size B_E, we introduce a second replay buffer to solely hold demonstration data, called the demonstration buffer. As the demonstration data is collected before training begins, its main difference to the experience buffer is that it is not updated during training and thus holds the demonstration data for the entire training duration. Its size B_D is equivalent to the number of demonstration state-action pairs.

We uniformly sample both from the experience replay buffer and the demonstration buffer with batch size $b_E = b_D = 64$. As both batches are merged, the actor and critic networks are optimized both with the demonstration and the latest experience data at every training step.

2.4.3 Behavioral Cloning

Similar to [104], we introduce a behavioral cloning loss $\mathcal{L}_{\text{BC}}$ on the actor network as an auxiliary learning task:

$$\mathcal{L}_{\text{BC}} = \sum_{i=1}^{b_D} ||\pi(s_i|\phi) - a_i||^2 \tag{2.4}$$

Only the batch fraction originating from the demonstration replay buffer is processed on the behavioral cloning loss. The resulting gradient of the actor network is

$$\nabla_\phi J_{\text{total}} = \lambda_{\text{RL}} \nabla_\phi J - \lambda_{\text{BC}} \nabla_\phi \mathcal{L}_{\text{BC}}. \tag{2.5}$$

Leveling both gradients against each other using $\lambda_{\text{BC/RL}}$ is important to achieve a balance where the navigation policy reproduces demonstration-like behavior around known states (in demonstration data), but also learns to handle unknown states correctly.

2.4.4 State Space

A visualization of our robot-centric state space is shown in Figure 2.2c. The state space is kept as minimalist as possible to ensure a fast and reliable training performance. The functionality of our approach is proven for a single human in the vicinity of the robot.

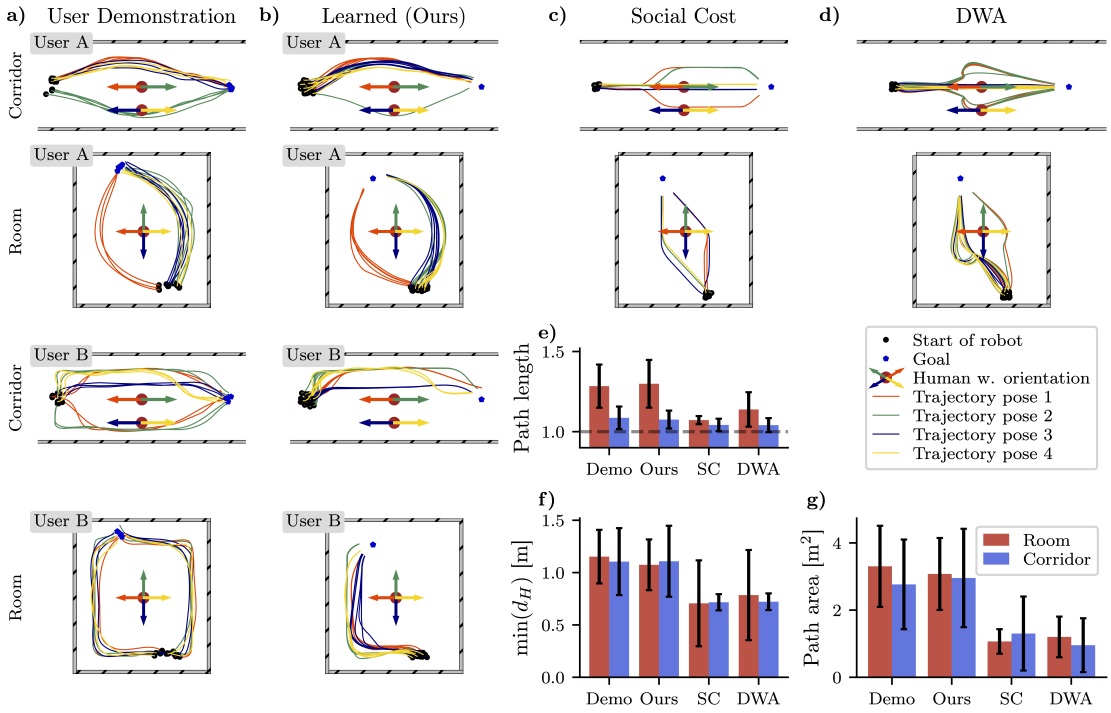


Figure 2.3: **a)** The demonstrated robot navigation preference trajectories of two participants, A and B, are shown for different human position-orientation pairs (color-coded). Note the wall-following preference of user B, whereas user A prefers a smooth curve navigation style. **b)** The personalized controller successfully learned to reflect the individual user preferences. Note that when no specific side preference is given, as in the demonstrations in the corridor, the controller reproduces trajectories mainly on one side. We evaluated our approach against **c)** the social cost model and **d)** the Dynamic Window Approach. A quantitative comparison of the different approaches in both environments reveals **e)** a higher relative path length (normalized by linear distance) and **f)** a higher preferred minimum distance. **g)** The increased path area for our controller (between the learned trajectory and linear distance) also points to a general preference for earlier deviation from the shortest path in favor of more comfortable trajectories.

The state vector contains the person's distance d_H to the robot's position and relative angle $\Delta\alpha_H$ to its orientation, facilitating human awareness. Furthermore, the relative angle to the navigation goal $\Delta\alpha_G$ is provided. To increase awareness for the human's field of view, the person's body orientation relative to the orientation of the robot $\Delta\psi_{RH}$ is included. It indicates whether a person faces the robot or not. To deal with obstacles, we include the closest distance d_{O_i} and relative angle $\Delta\alpha_{O_i}$ from the robot's pose to all environment obstacles O_i.

2.4.5 Reward

The reward function is designed to avoid collisions and ensure goal-oriented navigation behavior. We aim to teach user-specific navigation preferences not by complex reward shaping, but only via demonstration data. Consequently, we keep the reward as sparse as possible, besides basic collision penalties and goal rewards. More specifically, the

reward function is defined as

$$r = r_{\text{collision}} + r_{\text{goal}} + r_{\text{timeout}}. \tag{2.6}$$

We introduce a scaling factor for the reward $c_{\text{rew}} = 5$ that is used throughout the reward definition below. When the robot collides with the human or an obstacle during navigation, we penalize with

$$r_{\text{collision}} = \begin{cases} -c_{\text{rew}} & \text{if collision} \\ 0 & \text{else.} \end{cases} \tag{2.7}$$

The goal-reaching reward is provided to the agent if the robot is located closer than a certain distance to the goal position:

$$r_{\text{goal}} = \begin{cases} +c_{\text{rew}} & \text{if goal reached in demonstration data} \\ 0 & \text{if goal reached during training} \\ 0 & \text{else} \end{cases} \tag{2.8}$$

Note that we give a detailed explanation on the goal-reaching reward in Section 2.5.5. Finally, the timeout reward encourages the agent to avoid inefficient actions by penalizing behavior where the goal is not reached by the agent after a certain number of steps N_{ep}:

$$r_{\text{timeout}} = \begin{cases} -\frac{c_{\text{rew}}}{2} & \text{if episode timeout } (n > N_{\text{ep}}) \\ 0 & \text{else} \end{cases} \tag{2.9}$$

All three conditions above (goal reached, collision, timeout) are end criteria for an episode.

2.5 Demonstration and Training Environment

We propose a novel VR demonstration setup where the user teaches the robot personal navigation preferences in a virtual reality environment, see Figure 2.2a. The user can see the robot and its navigation goal (green cone). Intuitively, the person uses the handheld controller emitting a beam of light to draw preferred trajectories onto the floor in VR. The trigger on the backside of the controller allows the user to dynamically select the robot speed along the drawn trajectory. The robot executes the demonstrated trajectory right away for reevaluation, allowing the user to either keep or redo it. After the demonstrations have been collected, the training process begins. Finally, the personalized navigation controller is evaluated in VR, before being transferred to the real robot. For the user study conducted, we chose a corridor and a room environment, see Figure 2.4.

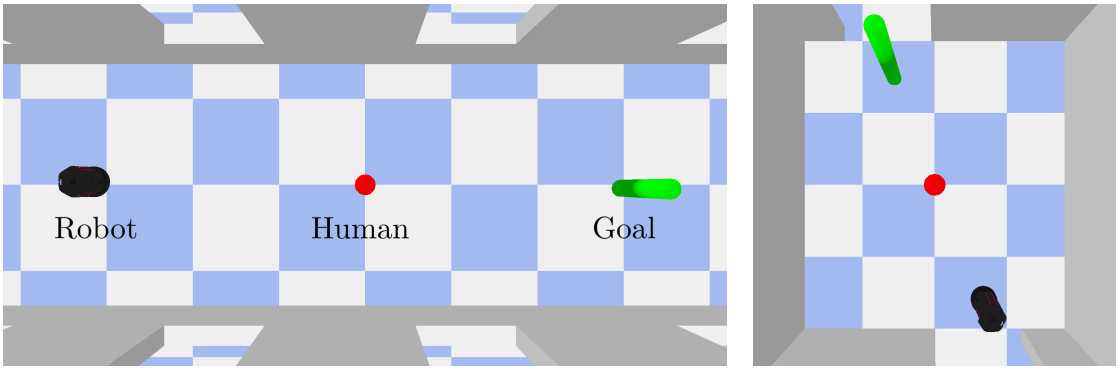


Figure 2.4: Top-view of both demonstration environment configurations: corridor **(left)** and room **(right)** of the VR interface for the user study. The human needs to be avoided by the robot, which is navigating to the goal.

2.5.1 Simulator and Robot

Our robotic platform is the Kobuki *TurtleBot 2*. As a VR and physics simulator we use PyBullet [107], the VR system is an HTC Vive Pro Eye.

A key challenge in using demonstrations for reinforcement learning is bridging the gap between the agent's and the demonstrator's state space. To do so, we analytically calculate action commands along a demonstration trajectory, so that the robot follows the trajectory by executing successive actions calculated at the control frequency f. The kinematics of a differential wheeled robot are

$$\begin{aligned} v &= \frac{K}{2}\left(u_r + u_l\right) \\ \omega &= \frac{K}{L}\left(u_r - u_l\right), \end{aligned} \tag{2.10}$$

where K is the wheel radius, L the distance between both wheels, and v the forward velocity. The rotation speeds of the left and right wheel are u_l and u_r. By integrating v and ω over time t, we find a relation for the finite distance $\Delta d = v\Delta t$ traveled forward and the change in robot orientation $\Delta\alpha = \omega\Delta t$ within a certain time period Δt:

$$\frac{v}{\omega} = \frac{\Delta d}{\Delta \alpha} \tag{2.11}$$

The time period Δt is determined by our chosen control frequency $f = \frac{1}{\Delta t} = 5\,\mathrm{Hz}$ of the robot. Now, given a desired forward velocity v, one can analytically calculate the matching angular control command ω to follow a discrete segment $(\Delta d, \Delta\alpha)$ along a trajectory.

2.5.2 Collecting and Processing Demonstration Trajectories

We use the following steps to process raw demonstration trajectories into state-action pairs contained in the demonstration buffer:

1. In VR, a user draws a trajectory using the handheld controller. The analogue trig-

ger on the controller backside allows us to control the robot speed linearly in the range from $v_{\text{min}} = 0.1\,\text{m/s}$ to $v_{\text{max}} = 0.25\,\text{m/s}$ at the drawing location.

2. The drawn trajectory is interpolated and smoothed with a 2D spline, parameterized by $k \in [0, 1]$. Also, the speed information is spline-interpolated.

3. The robot is supposed to follow the demonstrated trajectory. Based on the speed along the spline $v(k)$, we consecutively extract the locations on the spline at which the robot receives a new control command, using $\Delta d = v(k)\Delta t$.

4. Inserting $v(k)$ for all control command locations into Equation (2.11), the corresponding angular velocities ω are calculated.

5. The robot is placed and oriented according to the trajectory's starting point.

6. Successively, the control command tuples $a_t = (v_t, \omega_t)$ are executed and the robot follows the trajectory.

7. Before and after the execution of each action a_t, we record the corresponding states s_t, s_{t+1} and the reward r_{t+1}.

8. Finally, all state-action-reward pairs (s_t, a_t, r_t, s_{t+1}) are stored in the demonstration buffer.

Each demonstration trajectory is checked against possible collisions with the environment.

2.5.3 Data Augmentation

We use data augmentation to increase the data output from a single demonstration trajectory. More specifically, the robot's initial placement is shifted linearly by $\frac{\Delta d}{N_{\text{aug}}}$ within the distance $\Delta d = v(k_0)\Delta t$ along the spline $N_{\text{aug}} = 15$ times, where k_0 refers to the trajectory spline start. The result is a slightly shifted execution of the trajectory, while the original characteristic of the trajectory is preserved ($\max(\Delta d) = 5\,\text{cm} \ll$ environment scale). Steps 5) to 8) are repeated for each data augmentation.

2.5.4 Successful Demonstrations

Reinforcement learning with demonstrations works best when demonstrations are successful, i.e., lead to the goal state. Thus, we end each demonstration trajectory with the goal state and thus a positive reward. Even if the goal position is not at the exact end of the trajectory, the goal is retroactively moved to the end of the demonstration trajectory.

2.5.5 Value of Demonstration Data

To boost the value of demonstration-like behavior for the critics during learning, we exclusively provide the goal-reaching reward on the demonstration part of the batch, see Equation (2.8). The motivation behind this is that the agent should navigate on states

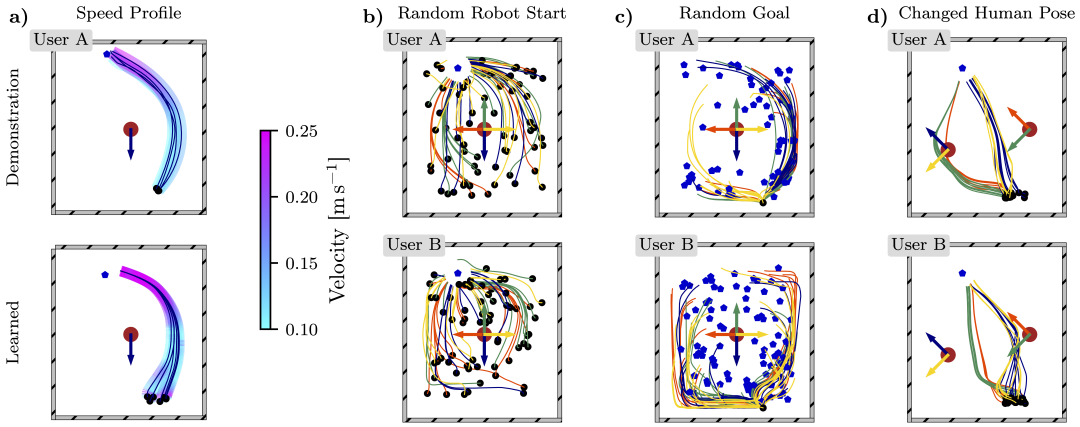


Figure 2.5: **a)** User A demonstrated a distinct speed profile **(top)** when facing the robot start position in the room environment. It was successfully adapted by the learned controller **(bottom)**. Furthermore, we tested the ability for generalization of the learned controller threefold by showcasing state configurations not covered by the demonstration data: **b)** When the robot starts at a random position in the environment, its navigation behavior still reflects the characteristics of the trajectory from the user demonstrations (cf. Figure 2.3a). **c)** Even when its goal is randomly placed in the room, the robot exhibits the distinct user preferences. **d)** The user's position and orientation were altered to non-demonstration configurations. When the human is obstructing the robot's path while facing the wall, the robot traverses on a straight path behind the human. In all other cases, a distinct distance is kept to the human, as demonstrated by both users. This clearly shows how the navigation agent improved beyond the limits of the demonstration data provided. For a legend, please refer to Figure 2.3.

s_t that are as similar as possible to the states of the demonstrated trajectories, ideally recovering to those whenever useful. To maximize return, however, the agent generally tries to navigate towards the goal with as few state transitions as possible (due to the discount value γ), possibly disregarding demonstrated user preferences. The resulting behavior corresponds to shortest-path trajectories with maximum speed while barely avoiding the human, promising a faster and higher return R. The demonstration state value boost counteracts this unwanted effect, since the agent is encouraged to follow state transitions from the demonstration data due to their always *higher* return.

2.5.6 Training

We initialize the robot, human, and goal position either around the position from the demonstration configuration with probability p_{env} or randomly in the environment to explore the entire state space with probability $(1 - p_{\text{env}})$. Training starts with pre-initialization of the experience buffer by executing 5×10^4 randomly sampled actions. Subsequently, we train for 800 epochs. Each epoch consists of 5000 environment interactions, while the actor and critic networks are updated every 5 interactions. Each epoch ends with 10 evaluation episodes. An episode denotes the trajectory roll-out from initial robot placement until one of the end criteria is satisfied. We train for each user and environment individually to learn context-sensitive controllers.

An overview of all relevant and experimentally obtained training parameters can be found in Table 2.1. We found it beneficial for the training performance to adjust the balancing factors to $\lambda_{\text{RL}} = \lambda_{\text{BC}} = 0.5 \times 10^1$ at epoch 350, and reduce the actors learning rate to $l_a = 1 \times 10^{-5}$ at epoch 650.

2.6 Experimental Evaluation

This section highlights the results of our user study and provides a qualitative and quantitative analysis of the learned personalized navigation controller.

2.6.1 User Study

We conducted a user study with 24 non-expert participants (13 male, 11 female) to i) record individual navigation preferences (demonstration data), ii) evaluate the navigation behavior learned by our personalized controller, and iii) evaluate the presented VR demonstration interface. Participants attended two appointments, the first being the demonstration session and the second being the evaluation session. In the user study section, the values in brackets refer to the mean survey-scores (1-5) and their standard deviation.

2.6.1.1 Demonstration Session

During the demonstration session, trajectories in both environments (corridor and room) were recorded, see Figure 2.3a. Each environment featured four position-orientation pairs (color-coded) for the participant. For each pair, between three and five trajectories were recorded. The total time investment was about 20 min. After the recording session, the participants were asked about their experience with the VR demonstration interface. The survey questions and results are shown in Figure 2.6a). Participants predominantly experienced comfortable interactions with the simulated robot (4.6 ± 0.1) and found drawing trajectories with our interface very intuitive (4.5 ± 0.1). Also, no participants disliked our demonstration environments while

Table 2.1: Notations and training settings.

Notation	Value	Description
p_{env}	0.25	Placement probability: room vs. start position
n_{ep}	300	Maximum number of steps per episode
B_E	1×10^6	Experience replay buffer size
l_a	1×10^{-4}	Learning rate of actor
l_c	8×10^{-4}	Learning rate of critic
γ	0.99	Discount factor
σ_{ϵ_π}	0.2	Std. deviation of exploration noise ϵ_π
σ_{ϵ_θ}	0.05	Std. deviation of target policy noise ϵ_θ
λ_{RL}	10/3	Weighting factor of RL gradient on actor
λ_{BC}	20/3	Weighting factor of BC loss gradient on actor

the majority liked it *very much* (4.6 ± 0.1).

2.6.1.2 Evaluation Session

During the second session, our personalized navigation approach was evaluated against two approaches in virtual reality: The Dynamic Window Approach (DWA) [36] using the ROS *move_base* package [108] in combination with a 2D lidar sensor, and a social cost model (SC) based on the configuration of [42]. Each navigation approach was shown in VR (order: SC $\rightarrow$ DWA $\rightarrow$ Ours) in both environments for all four position-orientation pairs (cf. Figure 2.3b-d), followed by an evaluation survey (cf. Figure 2.6b). Potential ordering effects cannot be completely ruled out. Participants were unaware of presented approach types. Pairwise Bonferroni-corrected Wilcoxon signed-rank tests indicated that our personalized approached significantly outperformed both the SC and DWA navigation on all three measures comfort (Q1), unpleasant closeness (Q2) and preference (Q3) (see Table 2.2). No significant differences were measured between SC and DWA.

2.6.1.3 Real Robot Evaluation

Our personalized controller was demonstrated on the real robot (room environment) to investigate the participants transition experience from the simulated to the real robot. The real robot evaluation was also complemented by a survey, see Figure 2.6c). As in VR, the navigation of the real robot was predominantly experienced comfortable (4.5 ± 0.1) and participants saw their preferences mostly reflected (4.3 ± 0.1). Furthermore, the transition from the simulated robot experience in VR to the real robot was mostly experienced as *very natural* (4.5 ± 0.1).

2.6.2 Qualitative Navigation Analysis

Figure 2.3a shows demonstration data from two participants in both environments. In the room environment, the preference of participant A is a smooth curve around their position, while the robot drives in their field of view when approaching from either side. Interestingly, participant B's preference is a wall-following robot that navigates at a higher distance from the human compared to participant A.

Figure 2.3b shows trajectories of the learned navigation behavior. The learned policy clearly reflects the characteristics of the demonstration trajectories. Furthermore, the robot adjusts its navigation trajectory according to the human orientation. For user A,

Table 2.2: Wilcoxon signed-rank tests on mean scores of all approaches

Question	Ours - SC	Ours - DWA	SC - DWA
Q1: comfort	$z = -4.17^*$	$z = -4.01^*$	$z = -1.81$
Q2: closeness	$z = -4.29^*$	$z = -4.2^*$	$z = -3.61$
Q3: preference	$z = -4.01^*$	$z = -4.06^*$	$z = -1.97$

Note that statistical significance was always $p < 0.001$, as marked with *. All other comparisons did not reach statistical significance.

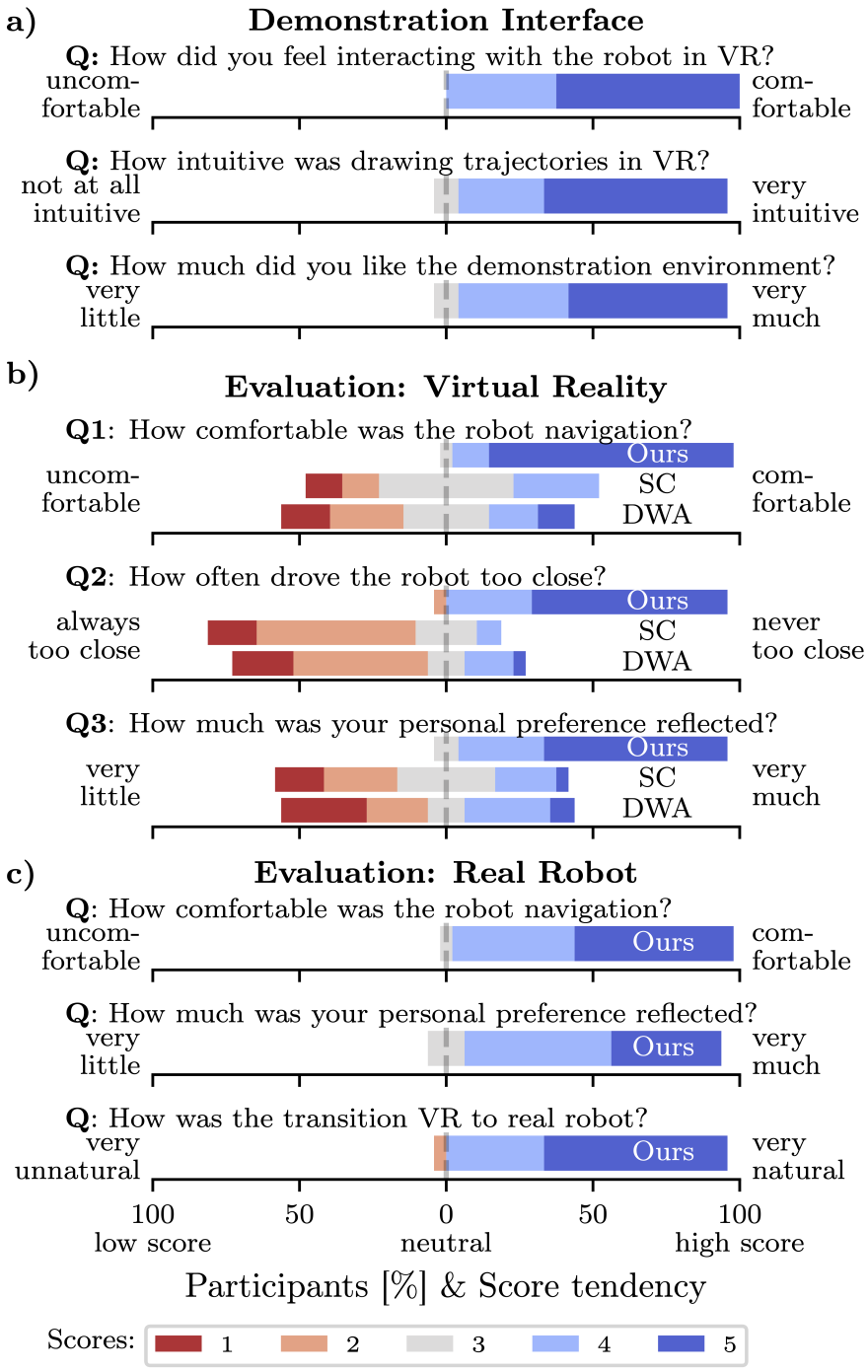


Figure 2.6: User study survey results of both the demonstration and evaluation session. **a)** The demonstration interface was predominantly appreciated and experienced as intuitive by the participants. **b)** Evaluation: In virtual reality, both the Dynamic Window Approach and the social cost model were outperformed by our personalized controller in various aspects. **c)** On the real robot, our novel personalized controller was perceived predominantly positively as well. The positions of the plot bars are aligned to the neutral score (3) to indicate overall rating.

it learned to traverse in the user's field of view, compare yellow orientation and trajectories. In participant B's demonstration, trajectories from a single position-orientation pair traverse both in front and behind the participant. Here, no specific side preference is given, and the controller reproduces trajectories mainly on one side.

Besides trajectory shape, users demonstrated speed profiles along the demonstration trajectories. As an example, Figure 2.5a depicts how user A demonstrated a distinct speed profile when directly facing the robot start position in the room environment. After the robot slowly approached and passed by, it was allowed to accelerate. As can be seen, the behavior is picked up by the controller during training.

2.6.3 Quantitative Navigation Analysis

Figure 2.3e-g compare quantitative properties of all three evaluation approaches and demonstrations from all 24 study participants. The personalized navigation trajectories are on average longer than those by DWA or SC, while maintaining a higher minimal distance from the human. Interestingly, the mean preferred minimum human distance gathered from the user demonstrations is similar in both environments, averaged at $\overline{d_H} = 1.1 \pm 0.2\,\mathrm{m}$. The path area is calculated between the trajectory and linear distance from start to goal. A higher path area reveals earlier deviation from the linear path in favor of personalization, as it is the case for our personalized controller, compare Figure 2.3g. This clearly indicates that users prefer personalized navigation trajectories over shortest path navigation. Furthermore, the large standard deviation of the path area indicates a high trajectory shape variability among the participants.

2.6.4 Generalization

Finally, we tested the ability for generalization of the learned navigation policy, see Figure 2.5b-d. First, the robot started at random positions in the environment not covered by the demonstrations. As can be seen, the controller still reflects the user preferences in the driving style (cf. Figure 2.5b and Figure 2.3a) by either approaching demonstration-like states or reproducing demonstration-like navigation patterns at slightly different positions in the environment. When appropriate, the robot drives straight to the goal. Second, we tested random goal positions in the environment (cf. Figure 2.5c). Interestingly, only after driving in accordance with preferences, the robot turns towards the goal when in direct vicinity. Finally, we tested altered human positions (cf. Figure 2.5d). Human position-orientation pairs not covered in the demonstration data encourage the controller to still keep a preference-like distance.

As demonstrated with these results, our framework can successfully learn a personalized navigation controller that improves beyond the limits of a few demonstration trajectories.

2.7 Conclusion

To summarize, we presented both a learning framework and an intuitive virtual reality interface to teach navigation preferences to a mobile robot. From a few demonstration trajectories, our context-based navigation controller successfully learns to reflect user preferences, generalizes successfully to non-demonstrated states, and furthermore transfers smoothly to a real robot. The conducted user study provides evidence that our personalized approach significantly surpasses standard navigation approaches in terms of perceived comfort. Furthermore, the study verifies the demand for personalized robot navigation among the participants. Also, our findings prove the suitability of the applied methodologies, and represent a first important step towards personalized robot navigation, made possible by our intuitive interface and comprehensive user study. Regarding the overarching research questions of this thesis, the development and validation of the VR interface for intuitive and efficient preference demonstration directly address RQ1 (Section 1.2.1), and the RL+BC learning framework balancing user preferences and goal-directed behavior directly addresses RQ2, compare Section 1.2.2. As a next logical step, we will transfer the framework to more complex and diverse environments.

While the approach presented in this chapter assumed static human poses and known obstacle positions in the minimalistic scenarios, the following Chapter 3 explicitly addresses preference-reflecting navigation in more complex household environments and interactions with dynamically moving users. Thus, we advance both our VR-based demonstration interface, and our hybrid learning framework: The challenge on the VR interface side is how to pair the floor-drawn demonstration trajectories for the robot with the motion patterns of a walking user. On the side of the learning framework, the demonstrations need to be anchored both around the dynamic user and in the complex indoor environment, requiring a suitable perception pipeline for the DRL policy.

3 Learning Depth Vision-Based Personalized Robot Navigation From Dynamic Demonstrations in Virtual Reality

Abstract

In this chapter, we present a learning framework complemented by a perception pipeline to train a depth vision-based, personalized navigation controller from user demonstrations. Our virtual reality interface enables the demonstration of robot navigation trajectories under motion of the user for dynamic interaction scenarios. The novel perception pipeline employs a variational autoencoder in combination with a motion predictor. It compresses the perceived depth images to a latent state representation to enable efficient reasoning of the learning agent about the robot's dynamic environment. In a detailed analysis and ablation study, we evaluate different configurations of the perception pipeline. To further quantify the navigation controller's quality of personalization, we develop and apply a novel metric to measure preference reflection based on the Fréchet distance. We discuss the robot's navigation performance in various virtual scenes and demonstrate the first personalized robot navigation controller that solely relies on depth images.

3.1 Introduction

Where humans share the same environment with a mobile robot, the robot's navigation behavior significantly influences the comfort of interaction [109]. As we learned in the preceding chapter, especially the personalization of robot navigation behavior in the direct vicinity of the user is a core factor for user comfort. We furthermore found that basic obstacle avoidance approaches are insufficient to address individual preferences regarding proxemics, trajectory shape, or area of navigation in a given environment, while being a key component to successful navigation without question. Instead, a robot's navigation policy should be aware of humans [40] and reflect the users' personal preferences.

In Chapter 2, we demonstrated that pairing a virtual reality (VR) interface with a reinforcement learning (RL) framework enables the demonstration and training of highly customizable navigation behaviors. The resulting navigation controller outperformed non-personalized controllers in terms of perceived comfort and interaction experience. However, a key assumption in the previous work is an always-present, static human of known pose in a predefined environment with pose-encoded obstacles, which results in a low-dimensional, information-dense state space, a clear benefit for the learning process. To overcome these assumptions and advance the approach to navigate in more

This chapter is a revised and updated version of the peer-reviewed publication [75]. Refer to Section 1.4 for details.

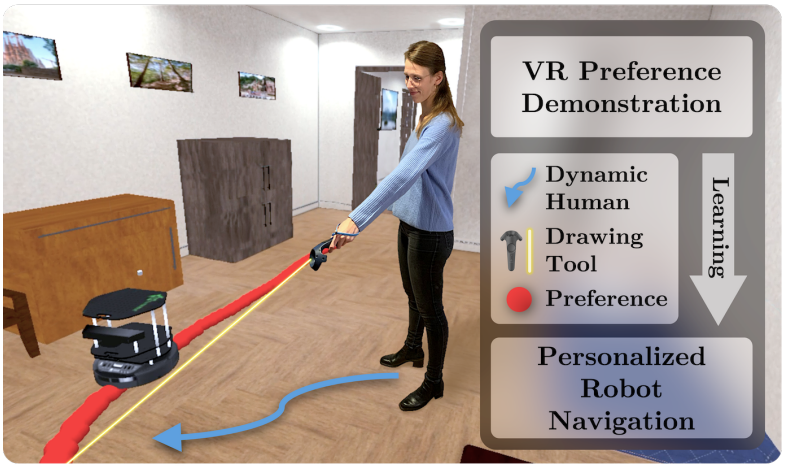


Figure 3.1: Our virtual reality (VR) interface allows the demonstration of robot navigation preferences by drawing trajectories intuitively onto the floor. By applying a learning-based framework, we achieve personalized navigation using a depth vision-based perception pipeline.

complex dynamic indoor environments, employing a depth vision camera to sense both human and obstacles is a possible solution [110]. However, depth vision cameras come at the cost of high-dimensional, complex, and redundant output. Learning from such high-dimensional data on dynamic scenes is a challenging task [111]. The question crystallizes, how do we capture and teach preferences of moving users in realistic environments, while relying on state-of-the art sensor modalities?

To solve the challenges above, we first advance the VR demonstration interface for demonstrations with users in motion, and second, introduce a depth vision-based perception pipeline that is both lightweight, human-aware and, most importantly, provides the robot with a low-dimensional representation of the dynamic scene. This perception pipeline i) perceives both the human and obstacles, ii) compresses the perceived depth information, and iii) enables efficient reasoning about the robot's dynamic environment for the learning framework. Our new system is able to learn personalized navigation preferences from a VR interface and learning framework for dynamic scenes in which both robot and human move.

In summary, the **main contributions** of our work are:

- Learning a preference-reflecting navigation controller that relies solely on depth vision.
- A VR demonstration framework to record navigation preferences for a dynamic human-robot scenario.
- The introduction and application of a novel metric to quantify the quality of navigation preference reflection.
- An extensive qualitative and quantitative analysis of different perception configurations for personalized navigation.

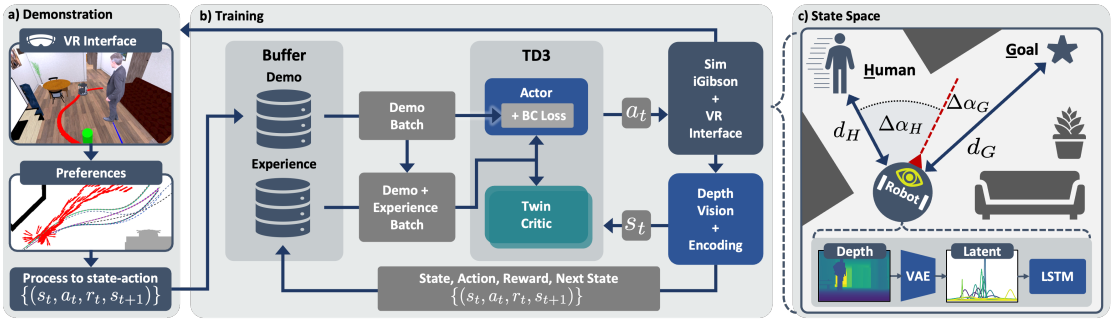


Figure 3.2: Schematic representation of our architecture. **a)** Demonstration trajectories are drawn by the user in VR on the floor using the handheld controller. Subsequently, the trajectories are fed into the demonstration buffer. **b)** Our TD3 reinforcement learning architecture with an additional behavioral cloning (BC) loss on the actor trains a personalized navigation policy that outputs linear and angular velocities. **c)** The robot-centric state space relies on a depth vision perception pipeline, capturing the vicinity of the human and the obstacles in the environment, as well as the relative goal position. A variational autoencoder (VAE) compresses the raw images to a latent state representation, while a predictor (LSTM) provides subsequent state predictions.

3.2 Related Work

Adjusting or learning the navigation behavior of a robot based on feedback or demonstration has been the focus of various studies [15], [69], [112]. Especially, deep learning-based approaches shine by their ability to learn from subtle and implicit features in their environment [83], [113], [114]. This is an ideal motivation to use a deep RL architecture for our personalized navigation controller.

Fusing the potential of user demonstrations with a learning architecture led to promising results in the field of robotic manipulation tasks [104]. Therefore, this is a key concept for our learning architecture and has successfully been applied to the personalize robot navigation in the previous chapter.

Vision-based sensor modalities for navigation appeal due to their cost-efficiency. For human-aware navigation, the detection and explicit localization of pedestrians enabled socially conforming navigation controllers [110], [115]. Other works learn navigation directly through monocular RGB vision [116].

Recent advances in the field of depth vision-based navigation in combination with RL have been made by Hoeller *et al.* [117], who study a state representation of depth images to efficiently learn navigation in dynamic environments. They employ a variational autoencoder to compress the high-dimensional depth image into a feature-rich representation for the agent. Our proposed perception pipeline is built upon their successful architecture.

Furthermore, a navigating agent benefits from dynamic scene understanding. Predicting the movement of surrounding pedestrians and obstacles with Long Short-Term Memory (LSTM) models has led to promising results [117], [118], [119]. Therefore, we will integrate an LSTM architecture into our perception pipeline.

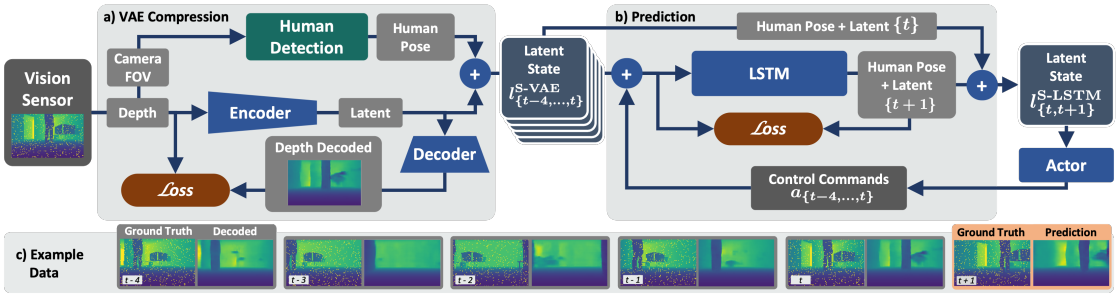


Figure 3.3: Schematic representation of the perception pipeline. **a)** The vision sensor's depth frames are encoded using a VAE to a latent space of dimensionality 32. In parallel, we check for human presence in the sensor's field of view (FOV), in which case we provide the human position relative to the robot. The merged latent state S-VAE provides our first state representation for learning. **b)** After the last five states $l^{\text{S-VAE}}_{\{t-4,\dots,t\}}$ are merged with the robot control commands $a_{\{t-4,\dots,t\}}$, the LSTM predicts both the next latent and the next human pose $l^{\text{S-LSTM}}_{\{t+1\}}$. Only the human pose prediction $(d_H^{t+1}, \Delta\alpha_H^{t+1})$ is merged with the previous latent l_t and human pose $(d_H^t, \Delta\alpha_H^t)$ to the state S-LSTM. Both state versions S-VAE and S-LSTM are used separately for training. **c)** Visualization of the trained VAE and LSTM model with ground truth depth data before encoding (left in box) and the decoder's reconstruction (right in box). The LSTM predicts the next latent and human pose, where the latent reconstruction is shown (orange box).

Since the publication of our original study, related advancements have been presented in literature. For instance, the combination of depth vision-based perception and LSTMs has been applied to navigation in crowded environments [120]. While their approach the emphasizes sim-to-real transfer of the policy, our focus lies in learning user preferences, their spatial anchoring in the indoor environment, and their explicit reflection in the navigation behavior. Regarding robot perception based on depth vision, Xu *et al.* have presented a principled framework for detecting and tracking dynamic obstacles from depth images [121]. While our work assumes that the human position can be obtained from the robots front-facing camera, thus treating it as given, their approach could serve as a suitable component for enabling a future sim-to-real transfer of the policy developed in this chapter.

In the previous chapter we presented one of the first approaches at the intersection of navigation and robot personalization, we now enhance the system by allowing the user to demonstrate navigation trajectories under dynamic motions and using only depth vision as controller input.

3.3 Our Approach

In this work, we consider a robot navigating in the same room as a single human user. The user has personal preferences about the way the robot circumnavigates them while pursuing a local goal in the same room. Such preferences could lie in the approaching behavior or the robot's trajectory. We assume the robot to be provided a local goal from a global planner. The local goal could be a door on the opposite side of the current

room to be traversed, or a location of interest in the same room. Using such sparse local goals several meters apart, we provide the controller with the spatial and temporal freedom to navigate towards the goal in a user-preferred, personalized manner. The human shares the navigation space with the robot, whether being dynamic by walking through the room, or resting static. To achieve preference-aligned and collision-free navigation behavior, the robot relies only on a depth vision camera to sense the distance from the human as well as obstacles. We formulate personalized navigation as a learning task in which the robot learns a personalized controller outputting linear and angular velocity from VR demonstrations of the user.

3.3.1 Learning Architecture

The learning approach presented in this section is a hybrid of reinforcement learning and behavior cloning.

RL refers to the optimization of environment interactions, leading from state $s_t \rightarrow s_{t+1}$ that obey a Markov Decision Process [105]. The interacting agent receives a reward $r_t = r(s_t, a_t)$ for taking an action $a_t = \pi_\phi(s_t)$ at time step t with respect to a policy π_ϕ. The tuples (s_t, a_t, r_t, s_{t+1}) are referred to as state-action pairs. The optimization goal is to maximize the overall return $R = \sum_{i=t}^{T} \gamma^{(i-t)} r_t$ of the γ-discounted rewards, onward from time step t.

Figure 3.2 depicts a schematic overview of our approach. We employ an off-policy twin-delayed deep deterministic policy gradient (TD3) reinforcement learning architecture [103]. In short, two critic networks learn to estimate the value of the state-action distribution, the actor network learns a policy $\pi(s_t) = a_t$ ideally leading to the highest expected return R. All three networks are standard multi-layer perceptrons (MLP) and share the same architecture. For policy updates, batches of training data b_E are sampled from the experience buffer. TD3's continuous action space ensures smooth robot control, as the actor network outputs linear and angular velocities as control commands.

An additional modification to the standard TD3 is a behavioral cloning loss $\mathcal{L}_{\text{BC}} = \sum_{i=1}^{b_D} ||\pi_\phi(s_i) - a_i||^2$ on the actor network provided with demonstration data in batches b_D [104] from a separate static buffer containing navigation preferences collected in VR, see Figure 3.2a-b. The extended and $\lambda_{\text{BC/RL}}$-balanced loss on the actor is $\nabla_\phi J_{\text{total}} = \lambda_{\text{RL}} \nabla_\phi J - \lambda_{\text{BC}} \nabla_\phi \mathcal{L}_{\text{BC}}$ with the actor's original policy gradient $\nabla_\phi J$.

By continuously sampling data from both buffers and applying the BC loss throughout the training, a navigation policy is learned that exhibits demonstration-like behavior whenever the navigation scenario allows. At the same time, the policy generalizes to unknown states not covered by the demonstration data.

3.3.2 Representation Learning

This section provides implementation and training details on our perception pipeline depicted in Figure 3.3.

3.3.2.1 Variational Autoencoder

Reinforcement learning on raw high-dimensional vision data is unfeasible. Ideally, a dimensionality-reduced state representation is used [117]. Thus, we compress the depth data to a latent representation l using a β-variational autoencoder (VAE) with six ReLU-activated convolutional layers, see Figure 3.3a. The dimensionality reduction is a factor of 320 from a 128 x 80 pixel depth image to a latent space of dimensionality 32. To make the model robust against sensor noise that a depth camera would exhibit, we apply a $5\ \%$ dropout noise to the depth frames during VAE training. The VAE learns to filter the noise, as the VAE's reconstruction loss is computed between the decoded and the noise-free depth frame. A visualization of the VAE's performance is depicted in Figure 3.3c.

3.3.2.2 Predictor

Originating from single depth frames, the latent space alone fails to capture dynamic scene information such as motion or human movement. To leverage dynamic scene information such as the human motion for the navigation controller, a predictor is introduced, see Figure 3.3b. The predictor receives the last five human poses, control commands, and latent frames $(d_H^i, \Delta\alpha_H^i, a_i, l_i)_{i\in\{t-4,\dots,t\}}$ as input. We predict the next human pose $(d_H^{t+1}, \Delta\alpha_H^{t+1})$ and the latent of the next time-step l_{t+1}. The model consists of two LSTM layers with 64 units each, followed two linearly activated MLPs which output both mean μ_{t+1} and variance σ_{t+1} as in the VAE, from which the latent prediction l_{t+1} is sampled. The human pose prediction is performed by a two-layer MLP from the LSTM-layer's output. A visualization of the predictor's performance is depicted in Figure 3.3c.

3.3.2.3 Training Data

To train the autoencoder and predictor, we generated an extensive dataset of depth frames in the iGibson simulator [122]. Here, we used the scene setup described in Section 3.4.3 with a static or dynamic human. The robot's navigation policy for the dataset generation was a simple obstacle avoidance controller trained with TD3 RL. Furthermore, the dataset contains ground-truth data about the human pose and the human's presence in the RGB-D camera's field of view (FOV).

3.3.3 State and Action Space

Our robot-centric state space consists of three main parts, compare Figure 3.2c: 1) The relative goal position $(d_G, \Delta\alpha_G)$, 2) the human position $(d_H, \Delta\alpha_H)$ and presence $k_H \in \{0, 1\}$ in the robot's FOV, and 3) the latent representation of the depth data. The human state corresponds to the current time step t for VAE-only configurations, and to a $t+1$ prediction concatenated with the t-human state for the VAE+LSTM. Thus, the human is both implicitly encoded as an obstacle in the latent-encoded depth image, but also explicitly. All positions are given in robot-centric polar coordinates. When no

human is observed in the FOV, then $d_H^* = -1$ m and $\Delta\alpha_H^* = 0$ rad. The actor's action space is composed of forward and angular velocity (v, ω), which are used as control commands.

3.3.4 Reward

We aim to teach user-specific navigation preferences not by complex reward shaping, but only via demonstration data. Consequently, we keep the reward as sparse as possible besides basic collision penalties and goal rewards

$$r = r_{\text{collision}} + r_{\text{goal}} + r_{\text{timeout}}. \tag{3.1}$$

The scaling factor $c_{\text{rew}} = 10$ is used throughout the reward definition below. Upon collision with either an obstacle or a human, we penalize with $r_{\text{collision}} = -\frac{1}{2}c_{\text{rew}}$. When the robot reaches the goal location, a positive reward is provided:

$$r_{\text{goal}} = \begin{cases} +c_{\text{rew}} & \text{if goal reached in demonstration data} \\ +\frac{c_{\text{rew}}}{2} & \text{if goal reached during training} \\ 0 & \text{else} \end{cases} \tag{3.2}$$

Note the explicitly higher reward of the demonstration data to boost the value of demonstration-like behavior for the critics during learning. This is further complemented by an additional $+\frac{c_{\text{rew}}}{100}$ on each demonstration state reward. In short, a higher value of demonstration-like behavior encourages user-preference-like navigation whenever possible, while preventing the agent from taking more efficient, shorter trajectories to achieve the faster and higher return R.

To overcome navigation behavior that does not lead to the goal on the long run, upon timeout when $n > N_{\text{ep}}$ we penalize with $r_{\text{timeout}} = -\frac{c_{\text{rew}}}{4}$. In all other cases the reward is zero. An episode denotes the trajectory roll-out from initial robot placement until one of the termination criteria is satisfied. All three reward criteria are also episode termination criteria.

3.4 Demonstration and Training Environment

We first introduce the advances on the VR interface, in which a human user teaches personal navigation preferences to a robot, now under dynamic motion. Subsequently, the learning environment and navigation task are presented.

3.4.1 Simulator and Robot

To teach and train our navigation controller in a more realistic environment with RL, we use the iGibson simulator [122] that provides a set of interactive indoor scenes and a VR interface that we used for immersive demonstration. iGibson renders the robot's

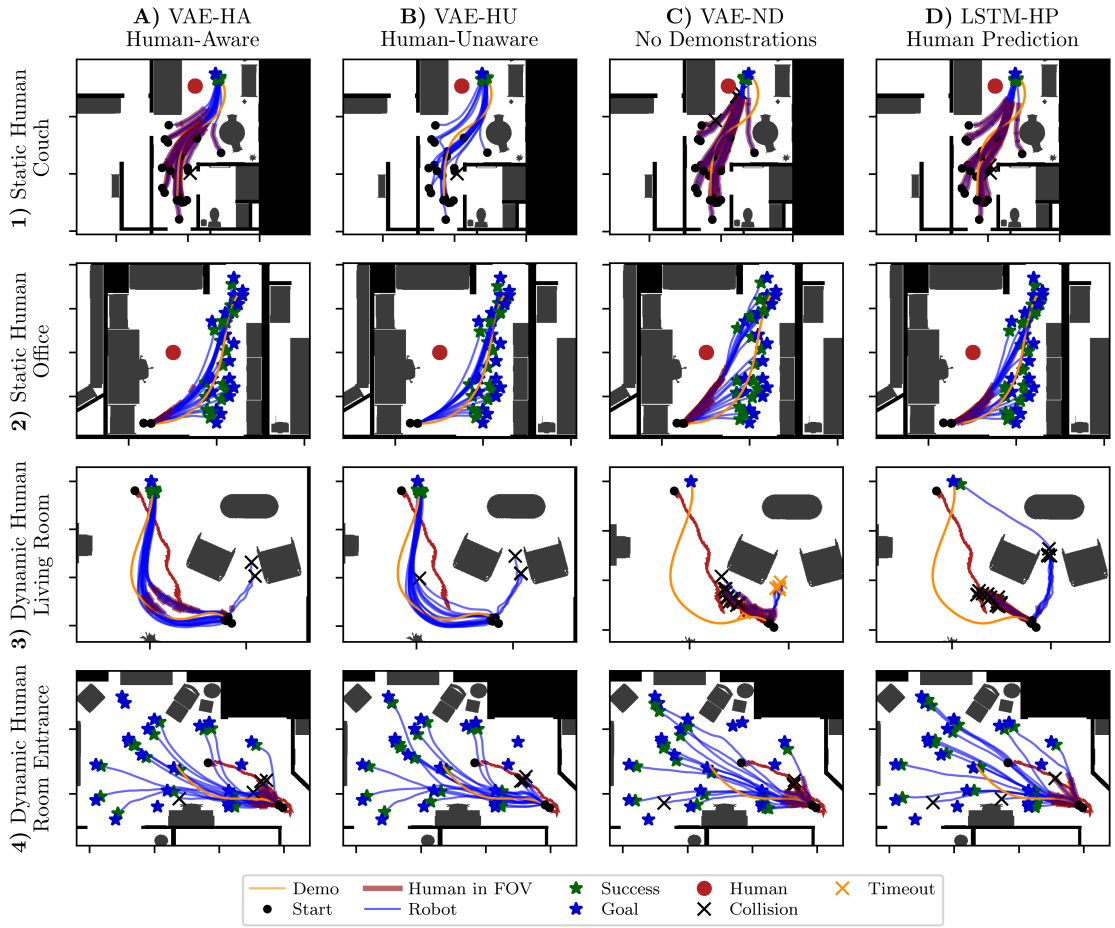


Figure 3.4: The robot's learned navigation behavior (blue lines) for scenes, where preferences were demonstrated (**rows 1-4**) and perception as well as learning configurations (**columns A-D**) are depicted. For all scenes, one demonstrated navigation preference is shown (orange). The human (red) is either static (red circle) or moving through the scene (red arrow). The goal (blue star) and start location (black dot) are either taken from the demonstration trajectory or sampled equivalently in the room across all configurations. Whenever the human is in the RGB-D camera's FOV, the robot trajectory is shaded in red. In short, the VAE-HA approach (A) exhibits navigation behavior which is the closest to the demonstrated preferences. As the human detection is turned off on the VAE-HU (B) and no human pose is provided to the controller, the robot performs less pronounced avoidance (B1 vs. A1). In contrast, the VAE-ND controller trained without any demonstration data (C) rather reflects a shortest path driving behavior. In the most challenging scene (3) LSTM-HP (D) and VAE-ND (C) fail, where the VAE-HA approach (A3) shines.

vision sensors, which serve as input to our perception pipeline during training. Its underlying physics engine is PyBullet [107]. We focus on the differential-wheeled robot Kobuki TurtleBot 2. Generally, our approach is applicable to other robots with similar control modalities. The TurtleBot's control limits lie at $v \in [0, 0.5]$ m/s forward and $\omega \in [-\pi, +\pi]$ rad/s angular. Inspired by the Intel RealSense D455 depth camera, the robot features a forward facing depth-camera with a $87°$ horizontal FOV. The sensing range is limited to 6 m, which is equivalent to a temporal foresight of 12 s at the TurtleBot's maximum forward velocity. As there is no sensor facing backward to sense rear

obstacles, the TurtleBot is not allowed to drive backward.

3.4.2 Collecting and Processing Demonstration Trajectories

As an extension of the VR demonstration interface compared to Chapter 2, the demonstrating user can now move and teach dynamic situations. To demonstrate, the user firstly familiarizes themselves with the environment in VR. Subsequently, he/she demonstrates a trajectory for the robot by drawing it onto the floor using the beam-emitting handheld controller, see Figure 3.1. There is no preset goal for the robot in the demonstration scene, so the user can demonstrate preferences in any direction. The goal location will automatically be set to the end of the demonstration trajectory. As the robot executes the trajectory from analytically computed action commands, the evolution of the human position and orientation is recorded from the wireless head-mounted display's location. So to complement the demonstration with his/her movement, the user can walk freely in the scene while the robot navigates. Just like the robot's trajectory, the human trajectory is also converted into a spline representation to be replayed during training and when the state-action pairs for the demonstration buffer are subsequently recorded. In a last step, the user can step aside and observe the moving robot and human 3D mesh from a third-person perspective. The demonstrations are double-checked for any collisions that would result in negative rewards upon replay to the demonstration buffer to ensure their quality for the learning process.

The conducted user study in the previous chapter has shown great acceptance of the VR interface and perceived navigation comfort of the learned controller. In this work, we focus on the development and evaluation of a depth vision-based perception pipeline. For this study, we recorded dynamic and static navigation scenarios by ourselves. The dataset contains nine scene configurations, with around three demonstration trajectories each.

3.4.3 Navigation Task and Training

We train our navigation controller on a set of interactive iGibson scenes and demonstration scenarios. Start and goal locations of the robot are randomly sampled in the same room, while ensuring a goal distance d_G between $1.5\,\mathrm{m} < d < 6\,\mathrm{m}$, equivalent to the depth sensing range.

To simulate the human in the scene, four different behavior modes are sampled: 1) Human walks in the opposite direction from the robot's goal to its start on an A* path, thus encountering the robot. 2) Random human start and goal locations. 3) The human is static. 4) No human in scene. 5) The human moves according to recorded demonstrations. For modes 1+2, the human speed is sampled from a standard distribution $\mathcal{N}(\mu = 0.5\,\mathrm{m/s}, \sigma = 0.3\,\mathrm{m/s})$.

Lastly, we randomize over a set of iGibson scenes during training and change scenes every 50 episodes.

Before training begins, the experience buffer is initialized with 5×10^3 samples by

executing randomly sampled actions. An overview of all relevant and experimentally obtained training parameters can be found in Table 3.1.

3.5 Experimental Evaluation

This section highlights the performance of our learned preference-reflecting navigation controller under different configurations. A qualitative analysis in Section 3.5.2 showcases and discusses the navigation behavior on a set of selected scenes. This is followed by a quantitative analysis targeting the robustness with success metrics in Section 3.5.3. Lastly in Section 3.5.4, we introduce a customized Fréchet similarity metric to quantify the quality of preference-reflecting navigation behavior with respect to the demonstrations.[1]

3.5.1 Perception Pipeline Configurations

We first evaluate different perception pipeline and learning configurations against each other, compare Figure 3.4.A-D and Figure 3.7.A-C. Their key differences lie in the state space as input to the RL policy.

The standard **h**uman-**a**ware VAE-HA (Figure 3.4A) state space configuration S-VAE contains the current latent depth encoding, goal position, the human presence binary and human position: $s_t^{\text{VAE-HA}} = (l_t, d_G, \Delta\alpha_G, k_H^t, d_H^t, \Delta\alpha_H^t)$.

The **h**uman-**u**naware VAE-HU (Figure 3.4B) is the same controller as the VAE-HA, but the human detection in the robot's field of view is disabled during evaluation.

The **n**o-**d**emonstration VAE-ND controller does not rely on the learning architecture as shown in Figure 3.2. It has neither a demonstration buffer, nor a behavioral cloning loss, making it a standard TD3 architecture. Therefore, it has learned its navigation behavior without user demonstrations.

[1] A video of the demonstration procedure and navigation performance is linked in the supplemental material section in the appendix.

Table 3.1: Notations and training settings.

Notation	Value	Description
β	3	Weighting factor of the VAE's KL-divergence
N_{ep}	150	Maximum number of steps per episode
B_E	2×10^5	Experience replay buffer size
$b_{E/D}$	64	Batch size of experience/demo data
l_a	1×10^{-4}	Learning rate of actor
l_c	8×10^{-4}	Learning rate of critic
γ	0.99	Discount factor
σ_{ϵ_π}	0.2	Std. deviation of exploration noise ϵ_π
σ_{ϵ_θ}	0.05	Std. deviation of target policy noise ϵ_θ
λ_{RL}	30/4	Weighting factor of RL gradient on actor
λ_{BC}	10/4	Weighting factor of BC loss gradient on actor

The **h**uman-**p**rediction LSTM-HP (Figure 3.4D) state space configuration S-LSTM is similar to S-VAE, except for the additional prediction of the next human position: $s_t^{\text{LSTM-HP}} = (s_t^{\text{VAE-HA}}, d_H^{t+1}, \Delta\alpha_H^{t+1})$. Therefore, S-LSTM provides dynamic scene information by predicting the human movement.

Our ablation study introduces two more configurations, see Section 3.5.5: VAE-FOV-120 implements a widened FOV at $120°$ over the standard $87°$, as it can be found on wide-angle depth cameras such as the Microsoft Azure Kinect. VAE-NG discards the goal distance d_G from the state space.

3.5.2 Qualitative Navigation Analysis

Figure 3.4 shows the learned navigation behavior of our controller and highlights resulting differences between the perception pipeline configurations introduced above.

In Figure 3.4.1, the human is static and located at the couch. The robot's start location is randomized, while keeping the goal at the end of the demonstration trajectory. As the robot traverses the living room, it shall navigate on the opposite side of the room close to the dining table and along the cupboard. With VAE-HA, the robot learned to navigate closely to the demonstrated preference. It exhibits a similar, smooth, S-shaped curve while passing by the couch. Interestingly, only little difference in the robot's overall trajectory shape can be observed between VAE-HA and VAE-HU (Figure 3.4.A1+B1). Here, a few trajectories traverse closer to the human (red dot). So even though the human is not explicitly observed in the state space of VAE-HU, its overall approaching behavior to the human still reflects demonstration patterns. A possible explanation is the agent's anchoring of behavior to the overall scene layout, rather than the human position. Note that the robot trajectories are shaded in red in Figure 3.4, whenever the human is observed in the FOV.

Located at the desk in Figure 3.4.2, the static human prefers the robot to take a wide turn as it leaves the corner next to the desk. In this scenario, the navigation behavior among all configurations except VAE-ND is mostly reflecting the wide turn, where VAE-ND cuts short on the wide turn as expected.

In Figure 3.4.3, the moving human encounters the robot with an opposite direction of travel at the living room's suite. As a preference, the robot should take a wide turn of avoidance around the armchair to make space for the approaching human. Among all controllers but VAE-HA, the navigation of the situation is challenging, leading to collisions around the armchair's corner. While LSTM-HP fails to exhibit the demonstrated behavior in this scenario, VAE-HA and -HU display successful preference-like behavior in most cases.

As the human walks out of the room in Figure 3.4.4, the robot enters. Upon detection of the approaching human, the robot shall take a left turn and make room for the human to pass. Afterward, the robot can continue traversing the living room to its goal. In this scenario, the effect of demonstration trajectories strikes: The VAE-ND controller without access to demonstrations mostly exhibits direct goal-oriented, straight-path navigation.

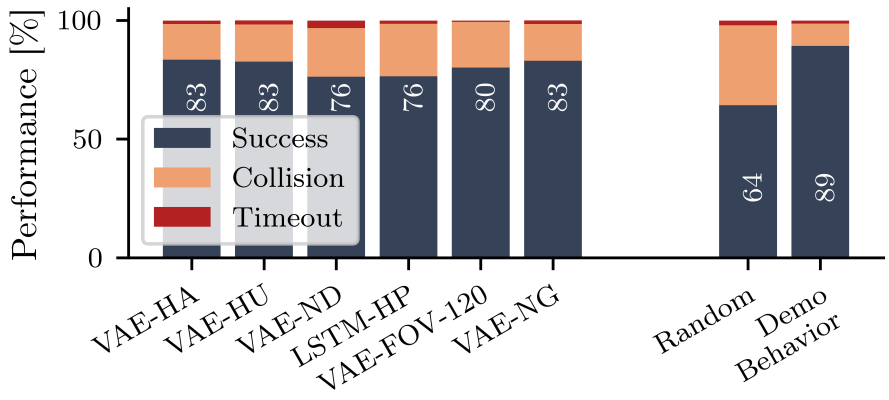


Figure 3.5: The performance of the different controllers has been averaged over all demonstration scenarios and other scenes. For each combination of scene, human behavior mode, and demonstration preference (if available), 50 trajectories were generated. "Random behavior" refers to behavior modes 1-4, while "demo behavior" refers to mode 5, both evaluated with controller VAE-HA. The success rates are shown in the plotted bars.

Interestingly, the same applies for the LSTM-HP controller, while it exhibits superior collision avoidance towards the approaching human over VAE-HA, though with some reduction in preference reflection.

Qualitatively speaking, the VAE-HA configuration results in the best-performing personalized robot navigation controller. Interestingly, the LSTM-HP configuration does not seem to provide a significant improvement compared to VAE-HA in most cases, but presumably at the cost of weaker preference reflection, compare Section 3.5.4.

3.5.3 Quantitative Analysis: Robustness

Figure 3.5 shows the performance of our different controller setups and human behavior modes (see Section 3.4.3) in terms of success rate, collision rate, and timeout rate. We determine the demonstration-aware VAE architectures (VAE-HA, -HU, -NG) most capable of avoiding collisions with scene objects and the dynamic user. Both the VAE-ND without demonstration access and the LSTM-HP controller perform worse than the demonstration-based VAE architectures. Regarding different human behavior sampling modes (Section 3.4.3), as expected, the demonstration-related mode 5 performs best. We can also conclude from VAE-FOV-120 that increasing the RGB-D camera's field of view, e.g., for better perception of pedestrians approaching from the side, does not lead to better collision avoidance. Generally, we observe more collisions than timeout events. This could be a consequence of the agent being encouraged to drive by the behavioral cloning loss from demonstration data.

3.5.4 Quantitative Analysis: Preference Reflection

To quantify how closely the individual controllers reproduce the demonstrated preferences, we use the Fréchet distance between the navigation and demonstration trajectories. The Fréchet distance $F(A, B)$ measures the similarity of two trajectories A and

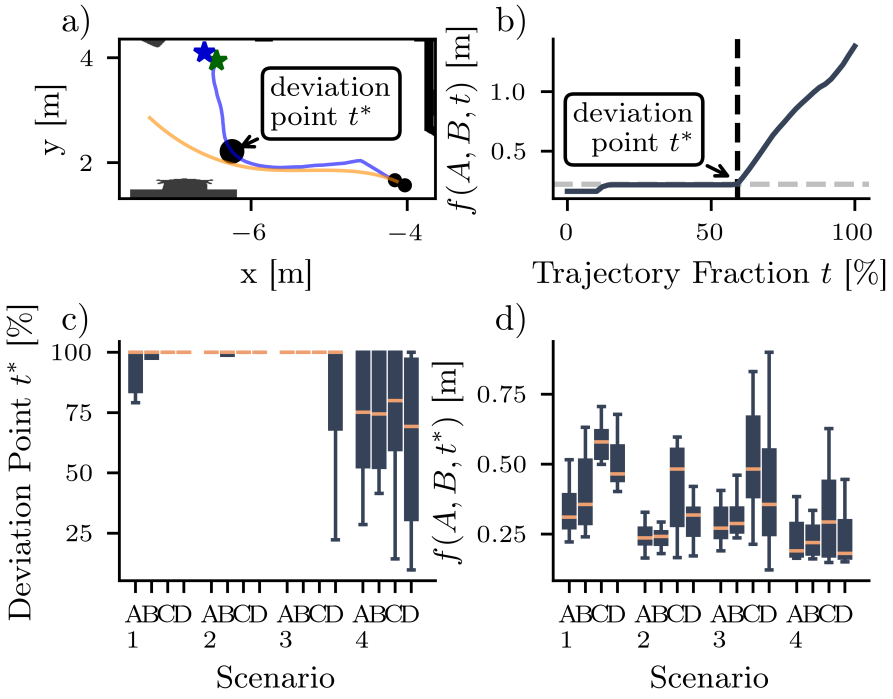


Figure 3.6: Visualization of our deviation-aware Fréchet metric $f(A, B, t^*)$. **a)** The robot follows the demonstrated path up to the deviation point t^*. Only up to this point we can reasonably compute a similarity between both trajectories. **b)** The deviation point t^* is determined by the sudden increase in the Fréchet distance between demonstration and t-partially considered navigation trajectory via a cost function. **c)** With regard to all scenarios and trajectories in Figure 3.4, the distribution of t^* is shown. **d)** Consequently, the deviation-aware Fréchet metric $f(A, B, t^*)$ is computed, pointing towards the best and worst preference reflecting controller, VAE-HA (A) and LSTM-HP (D), respectively.

B [123], by calculating the minimum value of the maximum distance between points on two curves or ordered points as $F(A, B) = \inf_{\alpha,\beta} \max_{t\in[0,1]} \|A(\alpha(t)) - B(\beta(t))\|$. The order of points is taken into account with all possible reparameterizations α and β of the curves, respectively. We leverage the Fréchet distance not only to compute the similarity, but also to estimate and quantify the point along the robot trajectory, where the robot significantly starts to deviate from the preference trajectory, as described below. An example of this procedure is shown in Figure 3.6a and 3.6b on a given set of trajectories. Firstly, the Fréchet distance is computed as a function of the considered fraction $t \in [0, 1]$ of the partial robot navigation trajectory $A[0, t]$ as

$$f(A, B, t) = \inf_{t'\in[0,1]} F(A[0, t], B[0, t']). \tag{3.3}$$

Secondly, a trade-off cost $C_\varphi(t)$ between $f(A, B, t)$ and t is computed as $C_\varphi(t) = \cos(\varphi)F(A, B, t) + \sin(\varphi)t$, where $\varphi = \frac{3}{4}\pi$. Thirdly, we define the deviation point t^* on trajectory A, where $\min_{t\in[0,1]} C_\varphi(t)$. In other words, we estimate the point along the robot trajectory t^*, when $f(A, B, t)$ starts to continuously increase as the robot leaves

the demonstrated path to pursue a goal aside the preference path. Finally, we can determine how closely the robot navigated along the demonstration trajectory up to the deviation point t^*, by evaluating $f(A, B, t^*)$. We call $f(A, B, t^*)$ the deviation-aware Fréchet distance. In Figure 3.6a+b the deviation point is marked in both plots.

By applying our metric, we solve the problem of either non-matching start or goal point between navigation and demonstration trajectory for a classical Fréchet analysis. For those cases, it would be pointless to quantify the similarity of both full-length trajectories with the plain Fréchet distance, as the deviation either at the end (same start) or at the beginning (same goal) would overshadow any measurable similarity. Our deviation-aware Fréchet metric $f(A, B, t^*)$ calculates the Fréchet distance in an isolated manner on trajectory segments, among which similarity can be expected. When the end points are close instead of the start points such as in Figure 3.4.1, the metric is applied on the reversed trajectories A and B.

We apply our deviation-point Fréchet metric to all navigation scenarios in Figure 3.4. On the one hand, we evaluated the deviation point t^* in Figure 3.6c and the corresponding deviation-aware Fréchet distance $f(A, B, t^*)$ in Figure 3.6d. We find the majority of navigation scenarios in Figure 3.4.1-3 to fully follow the demonstration trajectory, which manifests in a deviation point t^* close to $100\ \%$. This is especially true for Figure 3.4.3, where the start and goal of navigation and demonstration overlap. For the dynamic room entrance (Figure 3.4.4), the robot's deviations from the demonstration path in favor for aside or further in the room positioned goals reflect in a lower distribution of t^*, see Figure 3.6c.4A-D. However, no obvious difference in t^* can be observed when comparing the controller configurations, see Figure 3.6c.A-D. Interestingly, a clear difference in the deviation-aware Fréchet distance for the controller configurations can be found, see Figure 3.6d. Over all four navigation scenarios, controller VAE-HA (A in Figure 3.6d and Figure 3.4) exhibits the smallest deviation-aware Fréchet distance between preference and the resulting navigation. As expected, the worst preference reflection can be assigned to the plain TD3 architecture without demonstration access VAE-ND (C in Figure 3.6d and Figure 3.4).

3.5.5 Ablation Study

Finally, we perform an ablation study to investigate the effects of an increased camera field of view (Figure 3.7.B) and the removal of the goal distance from the state space (Figure 3.7.C). In the given scenarios, VAE-FOV-120 rather deteriorates the collision avoidance capabilities. This is in line with the obtained overall performance results, see Figure 3.5. Removing the goal distance (VAE-NG) interestingly does not deteriorate the performance, but also results in robust and preference-reflecting navigation.

Demonstrating the ability for generalization, in Figure 3.7.D we showcase a scenario where humans follow an A* path in the opposite direction from the robot (compare behavior mode 1 in Section 3.4.3). In most cases, the robot intuitively gives way to the approaching human.

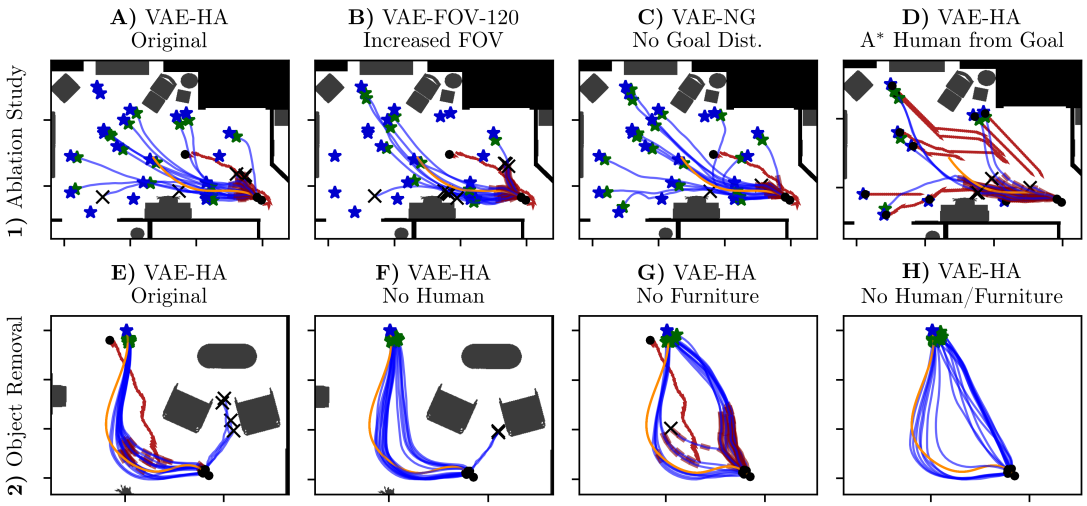


Figure 3.7: **1)** In our ablation study, we investigate (**B**) the effect of increased camera field of view with VAE-FOV-120, (**C**) the removal of the goal distance from the state space (VAE-NG) in comparison to the original approach (**A**). **2)** To learn about relevant environment features for the agent, the human (**F**), the furniture (**G**), or both (**H**) was removed from the scene, compared to the original setup (**E**). For a legend, please refer to Figure 3.4.

To learn which features of the environment the agent uses for navigation and preference reproduction, we removed either the human, furniture, or both from the scene, see Figure 3.7.E-H. Interestingly, as no human approaches from behind the armchair (Figure 3.7), the robot navigates closer to the chair with a similar trajectory shape. As all furniture is removed from the scene (Figure 3.7.G), the robot either exhibits preference navigation or a shorter path on the other side of the approaching human. With everything removed (Figure 3.7.H), the small deviation around the human collapses to a shortest path on most trajectories. Here, the deciding factor might be the initial orientation. But also when neither human nor furniture is part of the scene, the robot is able to reflect preferences. We attribute this behavior to the walls and room layout that are still observable for the robot, or the learned guidance by relative goal position in the state space.

3.6 Conclusion

To summarize, we presented a learning approach to personalized navigation based on depth vision. As demonstrated with our results, we successfully learned a personalized navigation controller that reflects user preferences from a few VR demonstrations in dynamic human-robot navigation scenarios. These dynamic demonstrations are a contribution of this work and directly contribute to this thesis' RQ1 on preference collection interfaces, compare Section 1.2.1. For the perception pipeline, various configurations have been tested and the extensive analysis points towards a pure VAE perception architecture for the best results, contributing insights to the overarching RQ3 on effective

sensor representations in dynamic environments, as motivated in Section 1.2.3. Interestingly, including the motion predictor did not significantly improve the navigation performance or preference reflection. Alongside the analysis, we have also developed and successfully applied a new metric that allows us to quantify the quality of preference reflection during navigation. As it measures the personalization quality of trajectories but also when the robot deviates from demonstrated behavior, its development directly contributes to RQ2 on preference task-balancing, compare Section 1.2.2. In conclusion, our research has demonstrated the feasibility of personalized robot navigation utilizing depth vision sensors and presents a promising avenue for further development, especially for preference anchoring in feature-rich environments.

A central assumption of the approaches presented in this and the previous chapter is that the user located in the robot's immediate vicinity is the same individual whose preferences are to be reflected. An interesting direction for future work involves extending these approaches to scenarios with multiple users, each potentially exhibiting distinct preferences and interacting with the robot simultaneously. This extension, however, would require the robot to reliably identify and distinguish between individuals, a challenge that lies beyond the scope of this thesis. Moreover, as the number of people in the robot's environment increases, the problem naturally shifts from single-user personalization towards a broader social navigation challenge. In these settings, the question can be raised of how much perceptual richness is truly required for reliable, socially aware behavior. Building on these findings, the next chapter investigates how sensor representations can be optimized for foresighted navigation in dynamic environments, shifting from depth vision modalities to 2D lidar-based perception. In particular, we explore robustness learning-based navigation in dynamic human environments with a lidar-based perception pipeline that eliminates the need for explicit pedestrian tracking.

4 Spatiotemporal Attention Enhances Lidar-Based Robot Navigation in Dynamic Environments

Abstract

Foresighted robot navigation in dynamic indoor environments with cost-efficient hardware necessitates the use of a lightweight yet dependable controller. So, inferring the scene dynamics from sensor readings without explicit object tracking is a pivotal aspect of foresighted navigation among pedestrians. In this chapter, we introduce a spatiotemporal attention pipeline for enhanced navigation based on 2D lidar sensor readings. This pipeline is complemented by a novel lidar-state representation that emphasizes dynamic obstacles over static ones. Subsequently, the attention mechanism enables selective scene perception across both space and time, resulting in improved overall navigation performance within dynamic scenarios. We thoroughly evaluated the approach across different scenarios and simulators, finding excellent generalization to unseen environments. The results demonstrate outstanding performance compared to state-of-the-art methods, thereby enabling the seamless deployment of the learned controller on a real robot.

4.1 Introduction

As demonstrated in the previous chapters on the application of behavior personalization, DRL-based robot controllers have the potential to learn nuanced human-aware navigation, also in dynamic environments. While our previous works (Chapter 2 to 3) focus on user-centric navigation around one individual, this chapter transitions into the domain of human-shared dynamic spaces. In these social navigation settings that involve more than one human, a key performance requirement for learning-based controllers is usually an information-dense representation of the dynamic scene, e.g., with explicitly tracked pedestrians [124]. However, when transitioning away from training and evaluation simulation frameworks to the real robot, complex fusion from multiple sensors and hardware-heavy post-processing steps are required to achieve such information-dense dynamics representations [125], [126], [127]. Here, also feature-rich but costly 3D lidar sensors are appealing [128], [129]. On the other side of the spectrum, many studies focus on learning-based navigation among dynamic obstacles of known position to avoid sensor-based pedestrian tracking [114], [130]. These approaches suffer from a reality gap that hinders generalization to the real world [131], [132]. Following the demand for improved reactive local planners, as recently emphasized by Xiao *et al.* [49], the need for sensor-based lightweight but reliable perception and navigation pipelines emerges that redundantizes explicit obstacle tracking. This necessity for per-

This chapter is a revised and updated version of the peer-reviewed publication [76]. Refer to Section 1.4 for details.

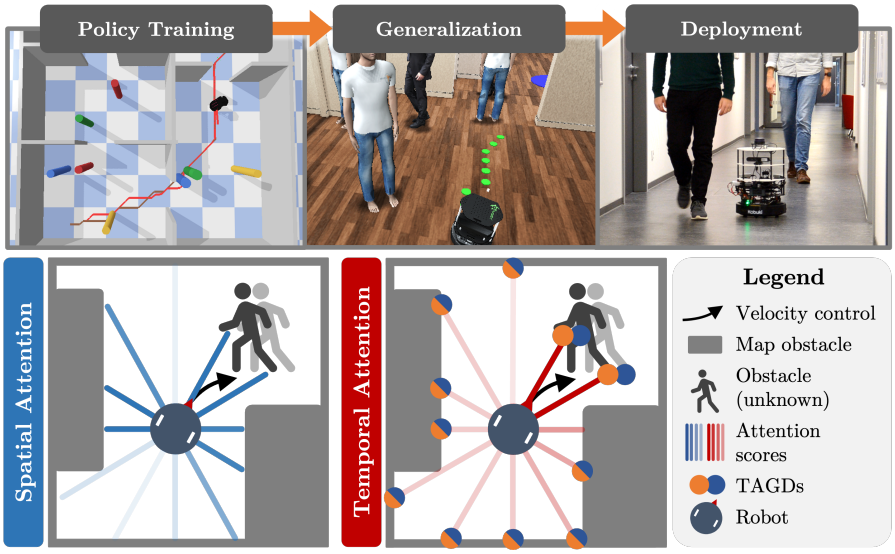


Figure 4.1: Our pipeline for learning a robot navigation controller based on lidar. Two attention mechanisms reason about the importance of individual lidar sectors with respect to known and unknown dynamic obstacles. Our Temporal Accumulation Group Descriptors (TAGD) reveal moving obstacles from subsequent lidar scans affected by robot self-motion.

formant sensor representations in dynamic environments ties directly to RQ3 of this thesis.

A possible solution is the use of 2D lidar sensors that provide accurate obstacle information within the moving plane of mobile robots [56]. They operate independently of lighting conditions, enabling both day and night operation. But without data such as colors or contours, explicitly tracking object instances like pedestrians only by their leg profiles from 2D lidar readings is a hard task [133], [134]. Furthermore, the robot's self-movement makes static objects appear dynamic between lidar scans.

While most current methods leverage convolutional neural networks (CNNs) to process and extract features from lidar data [70], [135], a recent appealing idea to tackle these sensor-implicit obstacle representations is selective attention on a collision-relevant subsectors of the lidar data [136]. Especially when a temporal observation sequence provides dynamic scene information, selective attention on moving obstacles can be beneficial.

To address this, we introduce a novel feature extraction technique tailored for 2D point clouds, incorporating both spatial and temporal attention across the sensor readings. This approach distills critical navigational information, offering a more robust solution for learning-based navigation in dynamic indoor environments. We demonstrate better than state-of-the-art generalization to unseen navigation scenarios and enable a smooth sim-to-real transfer of the learned policy, as we will be able to demonstrate in the experiments.

In summary, the main contributions of our work are:

- A deep reinforcement learning-based (DRL) navigation controller that learns dy-

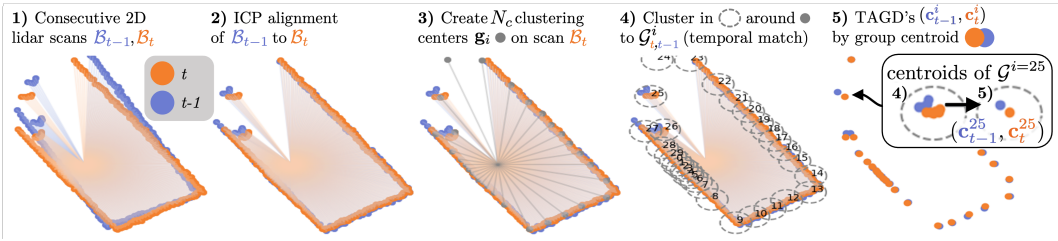


Figure 4.2: Schematic of the TAGD generation process. The ICP alignment of two subsequent lidar scans (1) in 2D Cartesian coordinates reduces the effect of robot self-motion (2). This allows better differentiation between dynamic obstacles and static obstacles. The aligned scan is grouped and clustered around ray-cast centers (3). From the clustered points (4), the position difference of the centroid from both time steps reveals a moving obstacle (5).

namic obstacle avoidance implicitly from 2D lidar readings only.

- A spatiotemporal attention module that infers the relative importance of different observation sectors with respect to proximity and obstacle motion trends.
- A novel 2D lidar observation representation highlighting dynamic obstacles over the robot's self-motion called temporal accumulation group descriptor (TAGD).

4.2 Related Work

Where mobile robot navigation decomposes into global path planning and local obstacle avoidance, the latter can be tackled with traditional and learning-based approaches. While traditional approaches such as the popular dynamic window approach (DWA) [36], [137] have been advanced with motion prediction [138], they come with the difficulties to avoid C-shaped or dynamic obstacles, or the necessity for re-tuning in different environments [90].

4.2.1 Learning-based navigation

Deep learning-based methods [139] appeal with decent generalization performance and less tedious fine-tuning as compared to hard-coded controllers.

Especially reinforcement learning-based (RL) methods have successfully been applied to motion planning [56], [140], [141], [142], as further demonstrated by Chapter 2 and Chapter 3. These works, however, do not embed dynamic scene understanding, thus limiting the agent's capability around walking pedestrians. Methods like SARL [114] or MP-RGL [130] capture interactions between robot and humans with excellent results, but rely on the known velocity of humans. Others infer or forecast human behavior by predicting their long-term goals, or by predicting their future motion and activities [143], [144], often by employing 3D lidar or RGB(-D) cameras [70], [116], [145], [146], [147]. Recently, learning mapless navigation from 2D lidar has shown promising results [70], [135], [148]. The challenge arises with our aim to learn time-series motion

trends for scene-dynamics aware navigation from 2D lidar readings in an end-to-end manner.

4.2.2 Point Cloud Feature Extraction

For the feature extraction in a deep RL navigation task, the spatial nature of lidar point cloud data suggests convolutional neural networks (CNN) as natural fit [70], [135]. Here, reducing the input dimensionality into a sparse encoding is a pivotal step. Taking into account the temporal dimension for scene dynamics understanding, individual lidar scans may be CNN-processed separately, followed by a multilayer perceptron (MLP) for joint extraction [70]. With PointNet [149], a high-performing network architecture for 3D point cloud registration has been proposed that was recently put to test in a short-horizon RL-based robotic manipulation task [150]. For obstacle pose and dynamics estimation, using a point cloud segmentation approach represents a viable avenue [129]. Looking at the non-learning-based domain, obstacle tracking from point cloud data has been presented before [151], [152]. With the advent of transformer models, the self-attention operator's invariance to cardinality and permutation of input data has proven to be a useful property for point cloud inference [153]. Building on this foundation, our work leverages attention-based 2D lidar feature extraction. This approach enhances deep RL-based local-obstacle avoidance, while integrating high-level guidance from a conventional path planner.

4.2.3 Recent Works

Since the publication of our original study, several works have leveraged attention mechanisms and transformer architectures for learning-based robotic tasks, as outlined below. Similar to our approach, Kazemi *et al.* employ both spatial and temporal attention modules to improve autonomous marine navigation in flow-affected waterways [154]. A graph attention module encodes obstacles and other state information, while a transformer block models the temporal evolution of the agent's environment, resulting in improved navigation performance. Although their work addresses a different application domain, their learning architecture is closely related to ours, as both follow an end-to-end reinforcement learning paradigm. For mobile robot navigation in dynamic environments, Zhang *et al.* propose a gated attention mechanism in a similar 2D lidar-based end-to-end RL policy [155]. While their method directly processes raw lidar input, our approach first applies the TAGD representation to expose dynamic obstacles more effectively before passing the data to the policy. An attention mechanism comparable to our implementation has also been applied in a recent learning-based robot manipulation task [156]. These recent developments highlight the growing potential of attention-based architectures in learning-based robot control.

4.3 Problem Statement and Assumptions

In this work, we consider a differential-wheeled robot pursuing a global goal in a cluttered and dynamic indoor environment, compare Figure 4.3a). A map of the empty environment is available for global path planning via A*. Static or dynamic pedestrians, however, are unknown obstacles to the robot. Also, the pedestrians at different speeds move rigorously without avoiding the robot in their motion, in contrast to other social navigation studies [135]. Therefore, smart and foresighted local collision avoidance is entirely up to the robot. The controlling agent has access to subsequent 2D lidar readings and upcoming path waypoints as observations, which it maps to linear and angular velocity commands. We formulate the task as in a learning-based manner and apply off-policy DRL. In summary, the proposed controller should be able to achieve two tasks: 1) Pursue the global goal through guidance of the computed path and 2) effectively avoid dynamic obstacles on a local scale.

4.4 Our Approach

This section explains our novel temporal accumulation group descriptor for lidar readings and subsequently the learning framework.

4.4.1 Temporal Accumulation Group Descriptor (TAGD)

It is inherently difficult to capture motion trends of moving obstacles from consecutive 2D lidar readings when the robot is in motion. To reveal moving obstacles over static ones without explicit obstacle tracking, we introduce our novel TAGD. We assume lidar scans to be recorded at a constant frequency of $1/\Delta t$ with a range of d_{max}. Our approach is described in Algorithm 1 with a visualization of all major steps is shown in Figure 4.2. We start with the min-pooled 2D lidar points $\mathcal{B}_{t-1}, \mathcal{B}_t$ with N points each. To eliminate the impact of robot rotation and translation, ICP [157] aligns $\mathcal{B}_{t-1}$ to $\mathcal{B}_t$ in the transformed point set $\mathcal{B}'_{t-1}$ (Figure 4.2.1). For static obstacles, the points now match up while their positions misalign for dynamic obstacles (Figure 4.2.2). For spatial clustering and subsequent temporal matching, clustering group centers g^i are formed along N_c uniformly cast rays by determining the robot-closest point within an angular threshold θ_{thresh} (Figure 4.2.3). For temporal matching and dimensionality reduction, the points in $\mathcal{B}'_{t-1}$ and $\mathcal{B}_t$ are assigned to clustering groups $\mathcal{G}^i_{t-1}$ and $\mathcal{G}^i_t$. This assignment is based on the Euclidean distance to their clustering center $\mathbf{g}^i$, within a fixed threshold $d_{\text{thresh}} = 0.25\text{m}$ (Figure 4.2.4). Note that d_{thresh} is a static parameter and chosen with a safety margin based on the relation between maximum expected obstacle speed and the inference time step as $d_{\text{thresh}} > v_{\text{max}}\Delta t$. The 2D centroids $\mathbf{c}^i$ of each group $\mathcal{G}^i_{t-1}$ and $\mathcal{G}^i_t$ counteract sensor noise and finally represent a single TAGD $(\mathbf{c}^i_t, \mathbf{c}^i_{t-1})$ (Figure 4.2.5).

A TAGD represents the center of data points across two consecutive lidar scans close to the nearest obstacle within an angular zone, such that even small obstacles hit by only

a single ray are successfully represented by a TAGD. Note that it is possible for a single dynamic obstacle to be represented in more than one TAGD, depending on the positions of clustering centroids, e.g., see TAGDs 26 and 27 in Figure 4.2.4). However, such double representations did not hinder the performance in the context of the learned controller. With regard to real-world pedestrians and their leg motion pattern, the influence of faster-than-body moving single legs on the TAGD displacement and therefore body speed estimation cannot be entirely ruled out. However, the group centroid calculation within d_{thresh} supports averaging out the effects of sensor noise or displacement and speed of individual legs, even though leg walking patterns are not explicitly considered or simulated in this work. It is worth noticing that a consistent inference timing between the lidar scans is key to correctly represent a given obstacle velocity with TAGDs, also with regards to a sim-to-real transfer. Here, this is directly based on the reinforcement learning control time step $\Delta t = 0.2\text{s}$. We have not used odometry or IMU data for enhanced ICP alignment, but solely rely on the observed static obstacles in the scene. While posing a limitation, this is a defensible assumption for indoor environments. However, we will evaluate the reliance and performance dependency of the navigating RL agent on correct ICP alignment in two ways, 1) without static obstacles in open space among dynamic obstacles, and 2) with ICP alignment artificially turned off. In summary, TAGDs reveal obstacle motion and will therefore be used as input to the temporal attention module of our pipeline.

Algorithm 1 Temporal Accumulation Group Descriptors

Require: Lidar readings $\mathcal{B}_{t-1}, \mathcal{B}_t$, $d_{\text{thresh}}, d_{\text{max}}, N_c$
$\theta_{\text{thresh}} \leftarrow \pi / N_c$
$\mathcal{B}'_{t-1} \leftarrow \text{ICP}(\mathcal{B}_{t-1}, \mathcal{B}_t)$
Initialize TAGD list $\mathcal{C}_t = \{\}$
for $i = 0$ to N_c **do**
  $\theta_{\text{ref}} \leftarrow 2\pi i / N_c$
  $T \leftarrow \{\mathbf{b} = (r, \theta) \in \mathcal{B}_t \mid |\theta - \theta_{\text{ref}}| \leq \theta_{\text{thresh}}\}$
  $r_{\text{min}} \leftarrow \min_{(r,\theta)\in T}(r, d_{\text{max}})$
  $\mathbf{g}_i \leftarrow (r_{\text{min}}, \theta_{\text{ref}})$
  $\mathcal{G}_t^i \leftarrow \{\mathbf{b} \in \mathcal{B}_t \mid \text{dist}(\mathbf{b}, \mathbf{g}_i) \leq d_{\text{thresh}}\}$
  $\mathcal{G}_{t-1}^i \leftarrow \{\mathbf{b}' \in \mathcal{B}'_{t-1} \mid \text{dist}(\mathbf{b}', \mathbf{g}_i) \leq d_{\text{thresh}}\}$
  TAGD $(\mathbf{c}_t^i, \mathbf{c}_{t-1}^i) \leftarrow (\text{centroid}(\mathcal{G}_t^i), \text{centroid}(\mathcal{G}_{t-1}^i))$
  $\mathcal{C}_t \leftarrow \mathcal{C}_t \cup \{(\mathbf{c}_t^i, \mathbf{c}_{t-1}^i)\}$

4.4.2 Deep Reinforcement Learning for Navigation

We choose a deep deterministic policy gradient (DDPG) architecture consisting of an actor and a critic, modeled by neural networks [158]. DDPG features a continuous action space, allowing for smooth robot control. The actor network outputs linear and angular velocities for the robot. The RL framework is based on the Markov Decision Process: An agent in state s_t at time step t decides upon an action a_t based on a policy $\pi(s_t) = a_t$ [105]. Upon reaching the next state s_{t+1}, it receives a reward r_t. The optimization

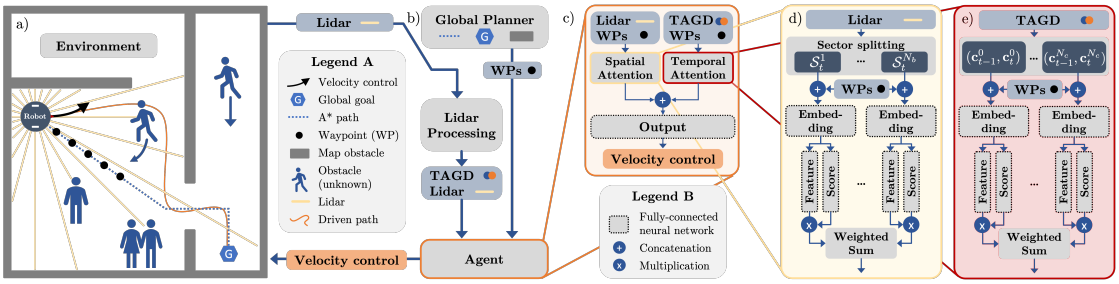


Figure 4.3: Illustration of our architecture. **a)** The indoor environment provides lidar readings to the deep reinforcement learning agent that drives a differential-wheeled robot via linear and angular velocity commands. **b)** From subsequent lidar readings, the TAGDs are computed. Merged with the five upcoming waypoints of the global path and the raw lidar readings as observations, they are **c)** processed by the agent in a separate spatial and temporal stream. Both streams feature an attention block to weigh the importance of **d)** individual lidar sectors (spatial) or **e)** the TAGDs (temporal), with respect to the upcoming waypoints. After feature extraction, both streams are concatenated for further processing in the final output network of the actor-critic agent.

objective is the maximization of the γ-discounted cumulative return $R = \sum_{i=t}^{T} \gamma^{i-t} r_t$, where $\gamma = 0.98$. As DDPG is an off-policy RL algorithm, the state-action pairs are stored in an experience replay buffer of length $N_{\text{RP}} = 2,000,000$ and sampled in batches for policy updates.

4.4.3 State and Action Space

The state space defines the observations we provide to the agent. As can be seen in Figure 4.3b, the agent has access to 2D lidar sensor data, the lidar-derived TAGDs, and upcoming waypoints for global guidance. The $N = 180$ ray min-pooled lidar scan $\mathcal{B}_t$ is represented as a set of robot-centric Cartesian 2D points. Focusing on local obstacles, the lidar scanning range is limited to $d_{\text{max}} = 3.5\text{m}$. To sense dynamic obstacles, we provide the TAGDs $\mathcal{C}_t = \{(\mathbf{c}_t^i, \mathbf{c}_{t-1}^i) | 0 \leq i < N_c\}$ as described earlier, where the number of TAGD's $N_c = 30$ is equal to the number of spatial sectors $N_b = 30$. The value $N_c = 30$ was heuristically chosen so that the clustering groups would jointly provide a cohesive circular coverage mid-range of the lidar distance around $d = 2.5\text{m}$, and the TAGD clustering circles defined by the radius d_{thresh} start to overlap ($N_c \approx 2\pi d/(2d_{\text{thresh}})$). From the robot nearest waypoint $\mathbf{p}^c$ on path $\mathcal{P}$, we sample $N_f = 5$ waypoints spaced at $\Delta\mathbf{p}^i = 0.3\text{m}$ towards the goal. These are converted to robot-centric Cartesian coordinates and input to the agent as $\mathcal{P}_t^f = \{\mathbf{p}^i | c \leq i < c + N_f\}$.

The continuous action space of the agent consists of linear and angular velocities (v, w), with a range of $v \in [0, 1.0]\text{m/s}$ and $w \in [-\pi, \pi]\text{rad/s}$. The robot is not allowed to drive backwards to foster foresighted navigation.

4.4.4 Reward

The navigating agent's overall objective is to navigate collision-free along a given path among unknown dynamic obstacles. The reward r_t is therefore a weighted sum:

$$r_t = \alpha_1 r_t^{\text{collision}} + \alpha_2 r_t^{\text{guide}} + \alpha_3 r_t^{\text{prox}} \tag{4.1}$$

$\alpha_1 = 10$, $\alpha_2 = 0.2$ and $\alpha_3 = 3$ are the experimentally determined weighting factors.

To encourage collision-free navigation, we penalize with $r^{\text{collision}} = -1$ upon collision of the robot with any obstacles.

A natural guidance along the global path is beneficial as it encourages the agent to drive towards the goal. From the current closest waypoint $\mathbf{p}^c$ on the path to the robot, we interpolate 0.6m forward along the path to obtain the guidance point $\mathbf{p}^g$. The distance between $\mathbf{p}^g$ and the robot's position $\mathbf{p}^r$ are penalized with $r^{\text{guide}} = -||\mathbf{p}^g - \mathbf{p}^r||$. By design and due to the update at every time step, $\mathbf{p}^g$ cannot be reached, thus providing a continuous penalty that increases when the robot deviates from the path and encourages the robot to drive back to the path in a forward-leading manner.

The concept of r^{prox} aligns with the sparse collision reward, but does not terminate the episode for easier learning. Instead it alerts the agent in vicinity to obstacles about a higher risk of collisions, or in other words encourages the agent to keep clear of obstacles. When the minimum distance d between robot and any lidar-scanned obstacle falls below $d_{\text{prox}} = 0.5\text{m}$, a linearly growing penalty is computed as $r^{\text{prox}} = -1 \times |d_{\text{prox}} - \min(d, d_{\text{prox}})|$, else $r^{\text{prox}} = 0$.

4.4.5 Network Architecture

As shown in Figure 4.3c, our agent's architecture is constructed around two data streams. The individual streams extract spatial and temporal features via an attention mechanism, respectively. Note that both the down-sampled lidar input of the spatial, and the TAGD input of the temporal steam contain partially redundant information due to their origin in the raw distance readings.

4.4.5.1 Temporal and Spatial Data Stream

With the individual TAGDs and the possibly attention-relevant, therefore redundantly represented upcoming waypoints, we construct N_c individual vectors $\mathbf{U}_{temp} = \{[\mathbf{c}_{t-1}^i, \mathbf{c}_t^i, \mathcal{P}_t^f] | 0 \leq i < N_c\}$, to be passed on to the temporal attention module as $\mathbf{y}_{temp} = Att_{temp}(\mathbf{U}_{temp})$, see Figure 4.3e. In the spatial data stream, the lidar scan $\mathcal{B}_t$ of N rays at time step t is split into N_b angular sectors $\mathcal{S}_t^i = \{\mathbf{b}_t^j \in \mathcal{B}_t | iN_b \leq j < (i+1)N_b\}$ with N/N_b rays each. Again, each sector-vector is concatenated with next path segment forming N_b individual vectors $\mathbf{U}_{spat} = \{[\mathcal{S}_t^i, \mathcal{P}_t^f] | 0 \leq i < N_b\}$, jointly passed on to the spatial attention module as $\mathbf{y}_{spat} = Att_{spat}(\mathbf{U}_{spat})$. After both data streams have been processed by their attention modules, respectively, they are concatenated and jointly processed by an output module $O(\cdot)$ for joint feature extraction $\mathbf{o} = O(\mathbf{y}_{temp}, \mathbf{y}_{spat})$.

Two separate modules of this pipeline form the actor and critic.

4.4.5.2 Attention module

Both temporal and spatial attention modules $Att(\cdot)$ share a similar network architecture, but no parameters. A visualization of our lightweight attention module can be found in Figure 4.3d-e. It is constructed with an embedding, a score and a feature network, inspired by Chen *et al.* [114] and [136]. The embedding module $E(\cdot)$ encodes the input vectors individually along the attention dimension to $\mathbf{e}_i = E(\mathbf{u}_i)$. The embedding $\mathbf{e}_i$ is fed into the score module $S(\cdot)$ that outputs the attention scores $\mathbf{s}_i = S(\mathbf{e}_i)$. All attention scores are Softmax-normalized to obtain the final importance weight. In parallel the embedding is also fed into the feature module $F(\cdot)$ that generates the feature representations as $\mathbf{f}_i = F(\mathbf{e}_i)$. Finally, the feature vectors are scaled by their importance in a weighted sum.

$$\mathbf{y} = Att(\mathbf{U}) = \textstyle\sum_i \text{Softmax}(\mathbf{s}_i) \cdot \mathbf{f}_i \tag{4.2}$$

$$= \textstyle\sum_i \text{Softmax}(S(\mathbf{e}_i)) \cdot F(\mathbf{e}_i) \tag{4.3}$$

Note that the due to the lightweight implementation of our attention scheme, the dimensionality along the attention axis reduces from N_b or N_c vectors to one in the output. In other words, the individually embedded lidar sectors or TAGDs do not attend to each other, but the attention scales their impact in the weighted sum, respectively. This form of attention is also referred to as *location-based* attention [159], [160]. All networks described above are constructed as ReLU-activated multi-layer perceptrons (MLP)[2].

4.4.6 Indoor Training Environments

To train our navigation agent, we use the PyBullet [107] physics engine. We use the minimalistic but well-randomizing indoor environments from de Heuvel *et al.* [136] featuring dynamic cuboid obstacles that represent pedestrians, with three different types of scenarios, see Figure 4.4: Corridors, intersections, and offices. The randomization of wall density and placement provides varying levels of scene complexity. The corridor environment is long and narrow with a length between $[6\text{m}, 8\text{m}]$ and a width between $[2.0\text{m}, 2.5\text{m}]$. The robot encounters pedestrians moving in opposite directions. The intersection environment is cross-shaped featuring hallway widths between 2.0m and 2.5m, and includes corners that create blind spots for sudden pedestrian appearances. The office environment features a fixed outer size with randomized interconnected rooms and introduces doorway encounters where the robot waits for pedestrian clearance before proceeding. Our room types cover typical encounters suggested for social navigation tasks [109], as found also in other related studies [70], [135]. While our rectangular en-

[2]Layer sizes (hidden nodes): embedding: $256 \times 128 \times 64$, score: $60 \times 50 \times 1$, feature: $80 \times 50 \times 30$, output: $128 \times 64 \times 64 \times \{1, 2\}$ (critic/actor)

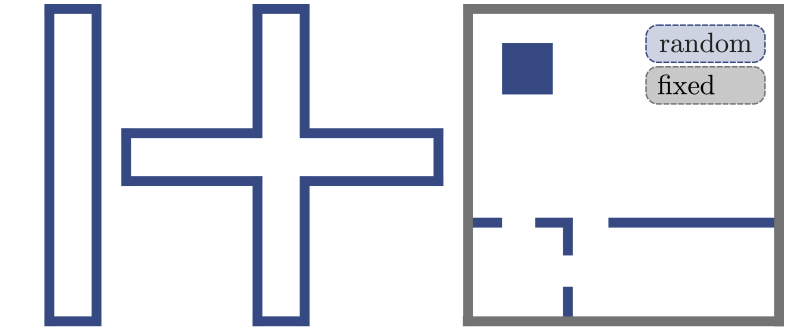


Figure 4.4: The PyBullet-based environments from [136] are used for training. **a)** In the corridor and **b)** intersection environments, the wall distances are randomized (blue). **c)** In the office environment, the outer walls are fixed with randomized inner wall placement for diverse room setups.

vironments generate variety through architectural randomization, other works achieve variety through larger but static, non-rectilinear scenes [146]. The robot's start and goal locations are sampled in the corners or dead ends of the scenes, respectively.

4.4.6.1 Obstacle simulation

Dynamic and static pedestrians represented by cuboids move back and forth through the environments along A* paths with randomized quantity ($N_{\text{dyn}} \in [1, N_{\text{dyn}}^{\text{max}}]$, $N_{\text{stat}} \in [1, 2]$), speed ($v_{\text{ped}} \in [0.5, 1.0]\text{m/s}$), start, and goal position. Note that the pedestrian speed can exceed the robot's maximum velocity. The maximum dynamic obstacle number $N_{\text{dyn}}^{\text{max}} \in \{2, 4, 8\}$ follows a curriculum scheme (three levels) and is increased over the course of training, whenever the evaluation success rate exceeds 70%. For the purpose of increasing the obstacle encounter likelihood with the robot, start and goal locations of the first pedestrian are sampled around the robot path. All other pedestrians will cross the robot path eventually. Note that the A*-following pedestrians do not take into account each other or the robot position, but rigorously move forward. Collision avoidance is therefore entirely up to the robot, similar to [70], [146] This can lead to highly challenging navigation encounters, especially for larger obstacle numbers. This is in contrast to other studies [135], [161] that simulate the pedestrians motion based on Optimal Reciprocal Collision Avoidance (ORCA), where the pedestrians avoid each other. Notably, also the robot is actively avoided by the pedestrians, easing the collision-free navigation task for the RL agent. Other works have employed the social force model for crowd navigation [162]. Though our more basal dynamic obstacle simulation leads to occasional pedestrian mesh overlaps and occasionally non-passable situations, our selection of an only path-based model is justified by our study's primary focus on feature extraction for RL-driven dynamic obstacle avoidance, rather than on crowd navigation.

4.4.7 Robot Model

We employ a differential-wheeled robot, more precisely, the Kobuki TurtleBot 2. The TurtleBot performs angular turns with a speed difference between both wheels. A Slamtec RPlidar A3 2D lidar sensor is mounted on top of the TurtleBot, emitting 1,440 beams. In simulation, we add sensor noise to the distance readings with an amplitude of 2.5cm.

4.5 Experiments

In the following we present the training and evaluation details, followed by an ablation and baseline study. After evaluating the domain shift to the iGibson simulator, the section is rounded up by the real-robot deployment.

4.5.1 Training Setup

An episode denotes one navigation run of the robot from start until one of the termination criteria is reached: Collision with other obstacles, timeout after $T_{\text{timeout}} = 150 \equiv 30\text{s}$ steps, or goal-reaching upon vicinity of 0.2m to the global goal. To foster generalization abilities, for each episode a randomly generated environment is set up, as described in Section 4.4.6. The inference and control time step of the agent is set to $\Delta t = 0.2\text{s}$, which also represents the time difference between subsequent lidar scans for the temporal processing. The learning rates for both actor and critic is 1×10^{-4}. All agents presented are trained for 300,000 episodes and evaluated regularly. The best performing model checkpoint of the highest curriculum level is selected for all approaches.

4.5.2 Quantitative Performance

We evaluated our trained models with respect to success rate, collision rate, timeout rate, and navigation time over 1,000 episodes. For comparability, the 1,000 episodes were set up identically among all approaches. The flagship approach presented in this study is denoted with OUR. Generally, with challenging environment complexity due to increased obstacle velocities (Figure 4.5a), or increased number of dynamic obstacles (Figure 4.5b), the success rate stagnates.

4.5.2.1 Ablation Study

We did an ablation study with respect to OUR approach described above to evaluate the contribution of each module to the results, see Table 4.1a and Figure 4.5.

A1 NO-SPATIAL: As OUR, but removing the spatial attention stream, leaving only TAGD and waypoint processing.

A2 NO-TEMPORAL: As OUR, but with no temporal stream or TAGD input, leaving only the spatial single time step attention stream and waypoint processing.

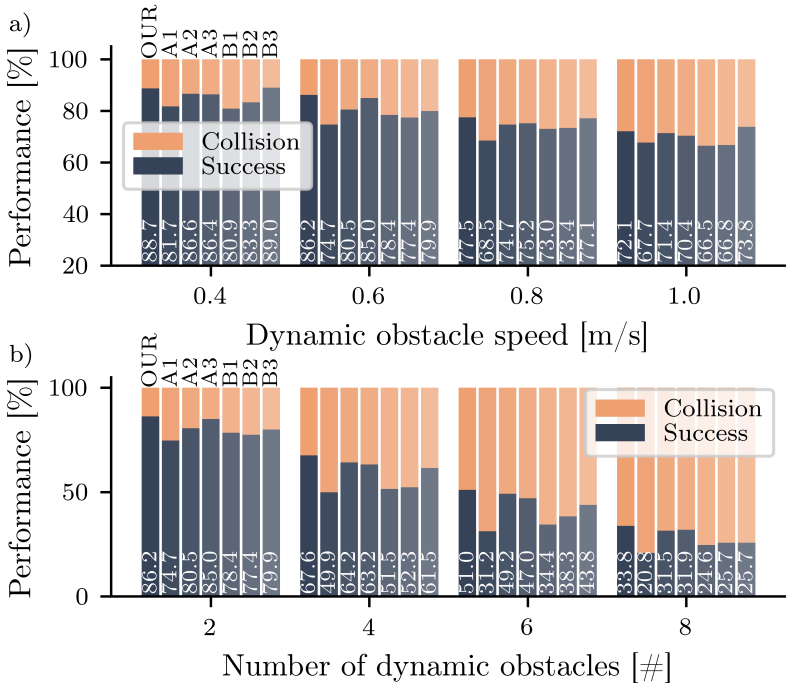


Figure 4.5: Performance overview for all approaches averaged over 1,000 episodes with identical scene setups in all three PyBullet environments for **a)** increasing obstacle speeds, with two dynamic and one static pedestrian, and **b)** increasing numbers of obstacles, with a fixed pedestrian speed $0.6\mathrm{m/s}$.

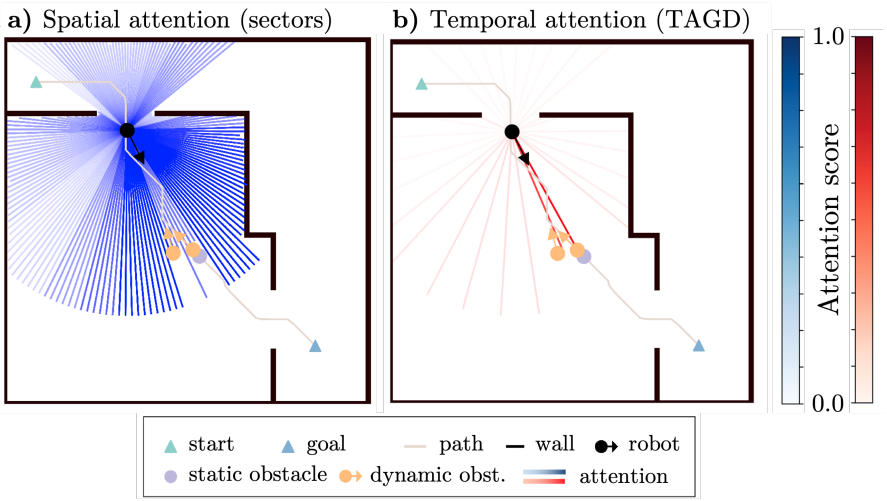


Figure 4.6: Exemplary visualization of the **a)** spatial (blue) and **b)** temporal attention (red) for a given navigation scene. The attention scores were color-mapped onto the lidar beam sectors for the spatial and on the beams pointing towards the TAGDs for the temporal attention, respectively. Increased spatial attention towards the forward-facing lidar sectors, as well as increased temporal attention towards the oncoming dynamic obstacle can be observed.

A3 NO-TAGD: As OUR, but without TAGD preprocessing. The network structure implements the spatial attention stream twice with separate network parameters, each processing one of the consecutive lidar scans, respectively.

As can be seen from Table 4.1a and Figure 4.5, with all ablations the performance deteriorates. The joint contribution of spatial and temporal attention emerges with A2 NO-TEMPORAL having a lower success rate compared to OUR, as it relies only on single-time step spatial information.

4.5.3 Baselines

To identify the contribution of our feature extraction approach, we compared it against two baseline architectures. All baselines leverage 2D lidar ($360°$) for learning-based mobile robot navigation and were trained in the same environment and training parameters as our approach. The baseline-related modifications lie in the state space content and processing network architectures.

4.5.3.1 Liang *et al.* - B1

A highly related state-of-the-art approach has been presented in [70]. Similarly to ours, it is an end-to-end obstacle avoidance algorithm originally trained with Proximal Policy Optimization. The authors use 2D lidar and a depth camera to perceive the environment, while the controller outputs velocity commands. From both perception modalities, we solely implement the lidar-related preprocessing and network architecture to replace our attention blocks, which is a 1D CNN taking in three consecutive scans. Precisely, this module is composed of two 1D CNN layers followed by a fully connected MLP. In contrast to our approach with 2D Cartesian point lidar representation, single-value lidar distance readings are used. The state space still contains five upcoming waypoints, which in contrast to OUR are processed by a separate MLP. Without convergence and therefore not included, we have also tested a closer-to-the-original implementation (512 lidar rays, no waypoints, only goal position).

a)	**Ablation**	SR↑	CR↓	TR↓	Nav. time↓
	OUR	**86.2**	**13.8**	0.0	**17.7** s
	A1: NO-SPATIAL	74.7	25.3	0.0	18.1 s
	A2: NO-TEMPORAL	80.5	19.5	0.0	17.9 s
	A3: NO-TAGD	85.0	15.0	0.0	17.8 s
b)	**Baseline**				
	B1: Liang *et al.* [70]	78.4	21.6	0.0	18.9 s
	B2: Pérez-D. *et al.* [135]	77.4	22.6	0.0	18.6 s
	B3: Pérez-D. *et al.* [135]	79.9	20.1	0.0	18.4 s
c)	**Generalization**				
	iGibson [163]	79.2	18.6	2.2	19.0 s

Table 4.1: Performance rates in [%] with respect to success (SR), collision (CR), and timeout (TR) and average navigation time for successful episodes of **a)** ablation and **b)** baseline study averaged over 1,000 episodes, with 2 dynamic pedestrians ($0.6\mathrm{m/s}$) and 1 static pedestrian. The **c)** generalization evaluation reveals slightly decreased performance for the post-training domain shift to the iGibson simulator on similar navigation tasks in more complex environments.

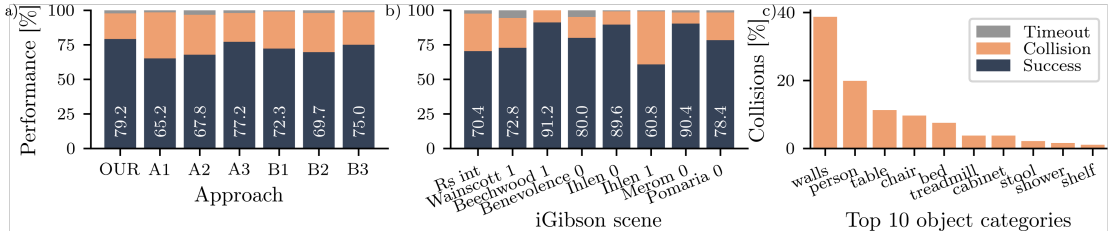


Figure 4.7: Results of the generalization study using the iGibson simulator over eight scenes with 125 episodes each. **a)** OUR controller demonstrates the best generalization capabilities in the sim-to-sim transfer of all approaches. **b)** Breakdown into the different scenes shows a scene dependency of the controller performance for OUR controller. **c)** Collision object category analysis within iGibson: The most-collided-with objects are the walls.

4.5.3.2 Pérez-D'Arpino *et al.* - B2/3

In the end-to-end lidar navigation approach of [135], no temporal information but only the current lidar reading is processed. Similar to Liang *et al.* [70], the authors employ a lidar-processing 1D CNN but with three layers followed by a fully connected layer. Furthermore, $N = 128$ single-value lidar distance readings are used. Additionally, the global goal position and next upcoming waypoints of the A* path ($\Delta\mathbf{p}^i = 1.0\text{m}$) are part of state space. In B2, we employ their state space and replace our attention block with their lidar-processing CNN and waypoint-processing MLP modules. A sub-version (B3) uses only their CNN architecture but our original state space with regards to waypoints and lidar resolution.

4.5.3.3 de Heuvel *et al.*

In initial tests, we compared against [136], outputting collision-free subgoals instead of velocity commands from single-time step lidar data with similar spatial attention. Direct comparisons with our current method are not viable due to later changes in the training settings. Despite this, the comparison showed a 5.3% performance boost by incorporating TAGDs and temporal attention, motivating our current work.

As can be seen in Table 4.1b) and Figure 4.5, for our setup, the CNN-related baselines B1-B3 struggle more with an increased number of obstacles. In almost all cases, our approach outperforms all baselines in terms of success rate.

4.5.4 Qualitative Attention Analysis

Figure 4.6 visualizes the learned spatial (a) and temporal (b) attention for a given navigation scenario. Here, two dynamic obstacles approach the robot from opposite directions, the robot has just entered the room. The spatial attention highlights the forward lidar sectors in the desired direction of navigation. The robot navigates along a wall that locates on its left-hand side, and we can observe an increased attention on the corresponding lidar sectors. Intuitive to the human eye, the temporal attention focuses the TAGDs of the oncoming dynamic obstacles. Similar to the spatial attention, a slightly

increased temporal attention can be observed in forward direction of the robot. In direct comparison to the temporal stream, the spatial stream exhibits a less sharp attention distribution in this scenario.

4.5.5 Robustness

Verifying the robustness of our approach with respect to ICP accuracy against its dependence on static obstacles for correct alignment, an open space evaluation of the same evaluation environments but without walls reveals an absolute performance decrease of 3.6%. When disabling ICP alignment entirely and feeding non-aligned lidar scans into the TAGD pipeline, the absolute performance drops by 4.8%. In both cases, the performance is still superior to the NO-TEMPORAL ablation, demonstrating decent robustness of the TAGD-based approach against ICP failure in these edge cases. Note that the obstacle parameters of Table 4.1 were used.

4.5.6 Generalization Performance

To investigate the generalization ability of our approach, we evaluated the PyBullet-trained agents in the iGibson simulator [163] in a sim-to-sim transfer, see Figure 4.1. The sensor settings and overall navigation objective remain similar, but two major differences strike: 1) The indoor scenarios are of high fidelity with diverse furniture objects and a more complex room architecture. 2) The pedestrians are represented with real 3D meshes instead of cuboids and have a more refined motion simulation. Precisely, we adapt the navigation task from the 2021 iGibson Social Navigation Challenge [164] that features eight scenes and Optimal Reciprocal Collision Avoidance (ORCA) among pedestrians. The key settings to mention as taken over from the original challenge are the maximum pedestrian speed of $0.5\mathrm{m/s}$, an inverse scene area-related population of 8 m^2 per pedestrian, and a goal sample distance between 1.0 and $10.0\mathrm{m}$.

As seen in Figure 4.7a), OUR controller exhibits the best generalization performance among all approaches. The slightly lower success rates in Figure 4.7a and Table 4.1c point towards a simulator gap and increased difficulty within the scenes. Also, the individual scenes seem to be of varying difficulty to the robot, compare Figure 4.7b). To further differentiate the challenges the robot faces in the iGibson scenes, the top ten collided-with object categories have been recorded, see Figure 4.7c). As the majority of collision events involve walls, the possibly higher degree of confined spaces within the iGibson scene could play a role. Furthermore, tables and chairs are among the most frequent collision causes. These objects are usually thin-legged, providing a challenge for lidar detection at low angular resolutions. In summary, the attention-based architecture surpasses the tested CNN feature extractors in unseen environments.

4.5.7 Real-World Experiment

Using the Robot Operating System (ROS) [108], we transferred the trained controller to a real Kobuki TurtleBot 2, as described in Section 4.4.7. In our experiment, the Gmapping

package [165], a Simultaneous Localization and Mapping algorithm, was used to build an occupancy grid map of real scenarios upfront for path planning. During navigation, Adaptive Monte Carlo Localization [35] estimated the robot's pose in the pre-mapped environment based on the lidar reading and robot odometry.

We tested our learning-based spatiotemporal approach qualitatively in various real-world scenarios, including corridors, intersections, and offices.[3] In a corridor, the two participants overtake the robot from behind or approach it rigorously from the front, see Figure 4.1. The robot smoothly gives room to the pedestrians and avoids collision. At an intersection, pedestrians appear from the blind spots behind a corner. In another test, the pedestrian blocks the doorway to see whether the robot would stop upon facing the impassable situation. All navigation situations are successfully handled by our spatiotemporal controller.

4.6 Conclusions

We proposed a novel and lightweight approach for robot navigation in dynamic indoor environments. Our learning-based approach featuring spatiotemporal attention demonstrates the capacity to highlight collision-relevant features from the sensor data, making the most of the sparse 2D lidar readings. Meanwhile, the introduced temporal accumulation group descriptors (TAGD) help to counteract the robot's self-movement over subsequent lidar readings and therefore support the differentiation between static and dynamic obstacles without explicit object tracking. Our policy directly outputs linear and angular velocity, leading to smooth robot navigation, and outperforms several state-of-the-art approaches in terms of collision rate for different pedestrian speeds and numbers of obstacles. We validate the sim-to-sim generalization capabilities in the iGibson simulator, finding excellent and better than state-of-the-art performance to unseen, more complex indoor environments with different pedestrian dynamics. Lastly, we achieve an effortless sim-to-real transfer into dynamic real-world indoor environments.

In conclusion, the findings of this chapter directly contribute to RQ3 on sensor representations in dynamic environments, outlined in Section 1.2.3. In the big picture of this thesis, this chapter also introduced the 2D lidar sensor for DRL-based robot controllers. In the subsequent chapters, we will consistently follow up on the 2D lidar sensor for obstacle sensing, while transitioning back to the challenge of personalizing human-aware navigation policies. With regard to the policy's behavior, the approaches of this and the previous chapters converge to a specific behavior profile during training, originally informed by reward design, reward weighting, or demonstration data. The trained policy may show satisfying performance or decent preference reflection, however, it cannot adapt to evolving user preferences post-training. Hence, the next chapter places a par-

[3] A video of the real-world experiment is linked in the supplemental material section of the thesis appendix.

ticular focus on policy adaptability to changing user preferences at a post-deployment stage.

5 Demonstration-Enhanced Adaptable Multi-Objective Robot Navigation

Abstract

Preference-aligned robot navigation in human environments is typically achieved through learning-based approaches, utilizing user feedback or demonstrations for personalization. However, personal preferences are subject to change and might even be context-dependent. Yet traditional reinforcement learning (RL) approaches with static reward functions often fall short in adapting to evolving user preferences, inevitably reflecting demonstrations once training is completed. This constraint also applies to the approaches presented in Chapter 2 and 4. This chapter introduces a structured framework that combines demonstration-based learning with multi-objective reinforcement learning (MORL). To ensure real-world applicability, our approach allows for dynamic adaptation of the robot navigation policy to changing user preferences without retraining. It fluently modulates the amount of demonstration data reflection and other preference-related objectives. Through rigorous evaluations, including a baseline comparison and sim-to-real transfer on two robots, we demonstrate our framework's capability to adapt to user preferences accurately while achieving high navigational performance in terms of collision avoidance and goal pursuit.

5.1 Introduction

The previous chapter is an example of how mobile robot navigation has significantly advanced with deep reinforcement learning (RL), enabling end-to-end policies that traverse complex environments with foresighted and nuanced behaviors. In scenarios involving human-robot interaction, however, it becomes crucial to align these policies with user preferences [166], e.g., on approaching behavior, proxemics, and navigational efficiency, to achieve acceptance [39].

However, traditional RL-based navigation methods typically optimize for static and pre-configured objectives in their reward scheme such as path efficiency or obstacle avoidance [135], neglecting user preferences and their variability over time. This has also been the case for the DRL-based approaches of Chapter 2 to 4. As a result, these methods lack mechanisms to adapt to shifting user preferences dynamically and require retraining to accommodate behavior changes, highlighting a significant gap in the current methodology.

A common strategy for addressing user preferences is learning from demonstrations. To preference-align RL-based navigation around the human, Chapter 2 to 3 have employed an additional behavior cloning loss driven by demonstration data. However,

This chapter is a revised and updated version of the peer-reviewed publication [77]. Refer to Section 1.4 for details.

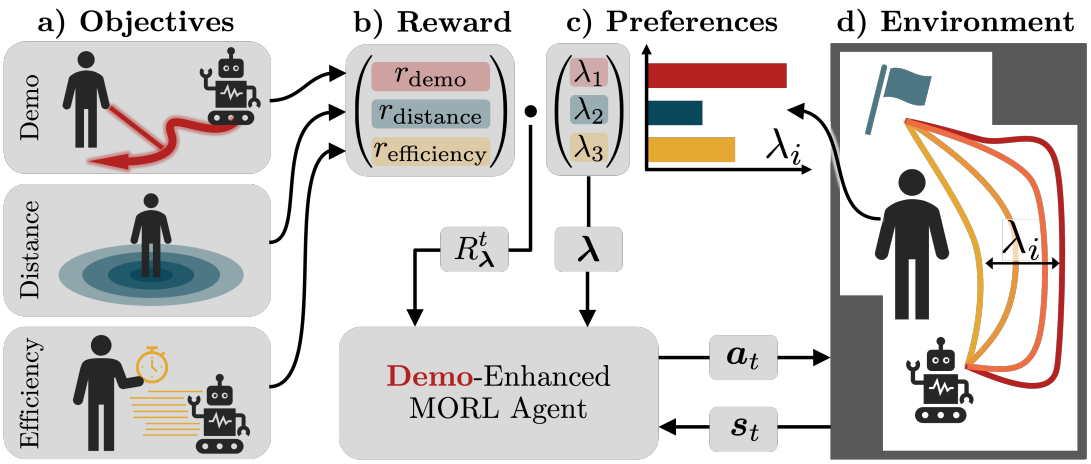


Figure 5.1: Our framework integrates demonstration-based learning into multi-objective reinforcement learning, enabling robots to adapt navigation policies to users' changing preferences even after training. **a)** The navigation style can fluently shift between demonstration-induced, distance keeping, and efficiency objectives. **b)** We modulate the MORL reward vector $\boldsymbol{r}_t$ with a **c)** varying preference $\boldsymbol{\lambda}$, while providing $\boldsymbol{\lambda}$ as input to the agent. **d)** The resulting human-centered policy can generate a spectrum of trajectories, here sketched for the objectives of demonstration-reflection (red, here: wall-following) and path efficiency (yellow).

these approaches do not provide principled ways to dynamically trade off demonstrated behaviors against core navigation objectives such as efficiency and collision avoidance. This can lead to overly conservative or inconsistent behavior, reducing usability in real-world applications. It becomes essential to devise mechanisms that can modulate the influence of demonstrations by user preferences, even after training.

To overcome these challenges, we propose a novel framework that integrates demonstration-based learning (LfD) via inverse reinforcement learning (IRL) into multi-objective reinforcement learning (MORL) to achieve flexible and preference-aware robot navigation (see Figure 5.1). This combination extends MORL's on-the-fly policy adaptation capabilities [167] by modulating the influence of demonstrations and other objectives without retraining.

Specifically, our combined approach of LfD, IRL, and MORL provides a structured way to incorporate user demonstrations as one of multiple competing objectives, enabling situationally adaptable trade-offs between demonstration adherence and navigational core objectives. Focusing on the robotic application, our experimental results demonstrate robust performance and accurate preference reflection for both a static and moving user. Finally, a comprehensive sim-to-real transfer on two different robotic platforms further validates the feasibility and robustness of our method in human-centered navigation tasks.

In summary, the main contributions of our work are:

- A multi-objective reinforcement learning human-aware robot navigation framework that enables policy adaptation to preferences post-training.

- The structured incorporation of demonstration data as a tuneable objective.
- Comparative navigation experiments in simulation validating demonstration modulation, behavior adaptation, robustness and generalization, concluded by a real-world transfer and evaluation on two different robots.

5.2 Related Work

The concept of user-aware personalized navigation is gaining momentum, emphasizing robots that adapt their strategies based on individual user preferences. Users can express preferences through ranking trajectory queries [168], [169] or providing demonstrations [30], [170], as demonstrated in Chapter 2 and 3. Both feedback types can distill a preference-aligned navigation policy. While trajectory ranking can be used to extract user preferences [69], this work establishes a demonstration-infused policy that aligns on-the-fly without retraining through multi-objective reinforcement learning (MORL).

The concept of optimizing for multiple objectives has already been applied in traditional non-RL navigation approaches [171], [172], [173]. Traditional methods, however, are limited by their inability to integrate preference-conveying demonstration data. In the context of RL, MORL extends standard RL by enabling the simultaneous optimization of multiple objectives. MORL frameworks exist for discrete [174] and continuous action spaces [175], [176], while the latter are particularly interesting for robotic tasks. So far, MORL has been applied to autonomous driving [177] and robotic tasks such as manipulation [178], navigation [69], [179], [180], [181], and path planning [182].

Ballou *et al.* [183] used meta reinforcement learning to adjust robot navigation among humans, efficiently fine-tuning policies for changes in the reward function, such as goal pursuit or distance keeping. However, their adaptation to shifting objectives is not instantaneous but rather requires an adaptation training phase. In contrast, our MORL policy adapts to preference weight changes in the preference space immediately.

Cheng *et al.* [179] proposed a MORL-based navigation policy that adapts to dynamic preferences over multiple navigation objectives in human environments, utilizing deep Q-networks for preference-weighted action selection. Similar to our approach, their method processes 2D lidar data as input. However, unlike our approach, they employ a discrete action space with acceleration commands, whereas we utilize MORL-enabled TD3 actor-critic architecture with a continuous action space of linear and angular velocity control for smooth motions.

Cheng *et al.* [180] presented an approach to learn navigation in human-populated environments with a multi-objective reward vector formulation. Compared to our study, they are not accounting for different preferences, as their approach optimizes a fixed set of objectives without mechanisms to adjust trade-offs dynamically. Choi *et al.* [168] proposed to use multi-agent training with parameterized rewards and action commands for adaptable robot navigation. Parameterized rewards can be used with standard RL policies, potentially at the cost of weaker multi-objective optimization. In contrast, our

agent estimates Q-values for different objectives separately while incorporating tunable demonstrations alongside other navigation objectives.

Hwang *et al.* [30] proposed a vision-based MORL framework for adapting robot navigation with discrete actions to human preferences through demonstrations, trajectory comparisons, and language instructions. However, their use of demonstrations is limited to estimating corresponding best-representing preference weights based on given objectives, possibly losing nuanced behavior traits in the demonstration data, whereas our approach directly integrates demonstration data to shape navigation behavior.

5.3 Our Approach

5.3.1 Problem Statement

We consider a wheeled robot navigating in the vicinity of a human and unknown obstacles, pursuing a local goal while avoiding collisions. The robot is controlled via continuous velocity commands. The human has certain preferences about the navigation style of the robot that may change depending on navigational context, such as task or time constraints, and which should be considered by the robot while navigating to the goal. These navigation preferences can be expressed both in the form of a preference vector and demonstrations. We assume the robot is provided a robot-centric goal location and can reliably estimate the human position, obstacles are perceived by the robot through 2D lidar. The navigation policy processes sensor data and goal information along with a preference vector containing user preferences, allowing for on-the-fly behavior adaptation within a single policy. Our approach explicitly focuses on single-human interaction, personalizing robot behavior based on individual user preferences rather than group dynamics.

5.3.2 Multi-Objective Reinforcement Learning

Multi-objective reinforcement learning (MORL) enhances traditional RL by integrating multiple, often conflicting, objectives [167]. In MORL, the agent is trained to learn policies that strike a balance among these diverse objectives, as opposed to a one-dimensional reward function. The MORL problem is formulated within the framework of a Markov Decision Process (MDP), defined by the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$. Here, $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $\mathcal{P} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0, 1]$ is the state transition probability, and γ is the discount factor [105]. A distinctive feature of MORL is the multi-dimensional reward function $\mathcal{R} : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}^n$, which outputs a vector of rewards $\boldsymbol{r}_t$ for n different objectives.

A single policy optimally adheres to a given combination of preferences, represented by the convex preference weight vector $\boldsymbol{\lambda} \in \mathbb{R}^n$. The policy $\pi_{\boldsymbol{\lambda}}(s)$ optimizes a scalarized reward function $R_{\boldsymbol{\lambda}}(s, a) = \boldsymbol{\lambda}^\top \boldsymbol{r}(s, a)$, itemizing the different objectives.

We employ the preference-driven (PD-)MORL TD3 implementation of Basaklar *et al.* [175], precisely MO-TD3-HER, which can learn a single-network policy that covers

the entire preference space.[4] PD-MORL achieves this by four major modifications to TD3's standard actor-critic-structure with respect to the policy loss and preference-space exploration: i) A preference interpolator $I(\boldsymbol{\lambda}) = \boldsymbol{\lambda}_p$ projects the original preference vectors $\boldsymbol{\lambda}$ into a normalized solution space, thereby improving the aligning of preferences with multi-objective value solutions $\boldsymbol{Q}$. ii) The framework is complemented by an angle loss $g(\boldsymbol{\lambda}_p, \boldsymbol{Q})$, designed to minimize the directional angle between the interpolated preference vectors $\boldsymbol{\lambda}_p$ and the multi-objective vector $\boldsymbol{Q}$, thus improving preference reflection. The actor network is updated by maximizing the term $\boldsymbol{\lambda}^T \boldsymbol{Q}$, where $\boldsymbol{\lambda}$ is the original convex preference vector and $\boldsymbol{Q}$ is the critic network's output, while simultaneously minimizing the directional angle term. iii) To efficiently learn across the entire preference space in PD-MORL-TD3, a hindsight experience replay mechanism [184] enhances the preference vector diversity during training. iv) The training process involves running a number of C_p environments in parallel for N time steps, each tailored to explore a distinct segment of the preference-vector space.

While Basaklar *et al.* originally evaluated PD MO-TD3-HER on gym benchmarks [176], we extend it to a three-objective robotic navigation task. The focus of our study is on task-related behavior adaptability, robustness, generalization, and real-world deployment performance. To the best of our knowledge, our study represents the first application of the PD-MORL framework to real-world robot tasks, where sensor-induced noise and partial observability introduce additional challenges.

5.3.2.1 State and Action Space

The state space includes the local goal, human position, and obstacles detected by a lidar sensor. The agent receives the relative 2D goal location $\boldsymbol{p}_g$ and human position $\boldsymbol{p}_h$ in polar coordinates. The $360°$ lidar scan, with a range of $4\ \mathrm{m}$, is min-pooled from 720 to $N_{\text{lidar}} = 30$ rays. These are combined in the state vector as $s_t = (\boldsymbol{p}_g, \boldsymbol{p}_h, \mathcal{L}_t)$, where $\mathcal{L}_t = d_i^t | 0 \leq i < N_{\text{lidar}}$.

The robot is controlled with linear and angular velocity commands $a_t = (v, \omega)$, where $v \in [0, 0.5]\ \mathrm{m/s}$ and $\omega \in [-\pi, \pi]\ \mathrm{rad/s}$. The perception-action loop runs at $5\ \mathrm{Hz}$.

5.3.2.2 Networks

The networks of actor, critic, behavior cloning policy, and reward model (see below) are fully connected multi-layer perceptron (MLP) networks with an identical architecture consisting of 4 layers with 256 neurons each. The uniform architecture is a heuristic choice, validated in preliminary experiments.

5.3.3 Incorporating Demonstrations

As one of our main contributions, we distill nuanced navigation from demonstration trajectories τ into a reward model that natively integrates into MORL as one of the objec-

[4] The code of our approach is linked in the supplemental material section of the thesis appendix.

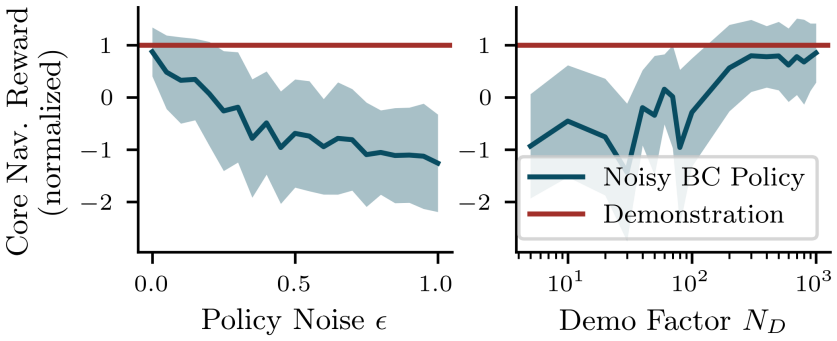


Figure 5.2: Exploration of D-REX-related demonstration parameters averaged over 20 trajectory rollouts, measured against the optimal demonstration behavior reward. **a)** The execution of the ϵ-greedy noise-injected behavior cloning (BC) policy trained with a demonstration augmentation factor of N_D = 1,000 reveals a degradation of navigation performance measured by the normalized core reward r_{core} with growing strength of the injected noise. **b)** The demonstration augmentation factor N_D indicates how many times the optimal human-centric demonstration trajectory (see Section 5.3.4.3) was rolled out with randomized obstacle placement to form the training dataset, showing increased performance with higher N_D.

tives and guides the learning agent to demonstration-like behavior. Through this novel design choice, the influence of demonstrations can be modulated by $\boldsymbol{\lambda}$ post-training.

A reward model is typically derived from pairwise A≻B preference queries in a human feedback process via a ranking loss [69]. However, demonstrations are typically considered equally important, rendering them unsuitable for a ranking-based reward model. Addressing the problem of non-existent ranking from demonstration data, we use a workaround involving artificial rankings. We employ the disturbance-based reward extrapolation (D-REX) approach by Brown *et al.* [185], which imitates pairwise A≻B preference queries by ranking over noise-injected demonstration trajectories. First, a behavior cloning (BC) policy π_{BC} is trained from N_D demonstration trajectories. Subsequently, the BC policy $\pi_{BC}(\cdot|\epsilon)$ is executed with increasing level of ϵ-greedy policy noise $\epsilon \in \mathcal{E} = (\epsilon_1, \epsilon_2, \ldots, \epsilon_d)$ with $\epsilon_1 < \epsilon_2 < \ldots < \epsilon_d$. In short, low-noise trajectories almost perfectly resemble the demonstration trajectory, while they slowly lose their shape with growing levels of noise. Trajectory rollouts generated with lower noise are automatically ranked superior compared to their higher-noise counterparts. Finally, a rich preference-ranking dataset

$$D_{\text{rank}} = \{\tau_i \prec \tau_j | \tau_i \sim \pi_{BC}(\cdot|\epsilon_i), \tau_j \sim \pi_{BC}(\cdot|\epsilon_j), \epsilon_i > \epsilon_j\}$$

is obtained. From D_{rank}, we train a reward model $\hat{R}(s,a) \in [0,1]$ using the Bradley-Terry model [186] with its typical implementation as a binary cross entropy loss such that $\sum_{s\in\tau_i} \hat{R}_\theta(s,a) < \sum_{s\in\tau_j} \hat{R}_\theta(s,a)$ when $\tau_i \prec \tau_j$.

For our ranking dataset D_{rank}, we choose a noise range $\mathcal{E} = (0, \ldots, 0.2)$ and obtain N_D = 1,000 demonstration augmentations with obstacle randomization from a single demonstration pattern.

5.3.4 Reward Vector

The reward vector covers traditional navigational objectives, subsequently referred to as core objectives, and three tuneable distinct style objectives based on quantifiable metrics and preference demonstrations. In our MORL setup, the core objectives are summed and occupy the first entry in the reward vector $\boldsymbol{r}_t$ which is assigned a static preference weight of one. Note that this is neglected in further notations of the convex vector $\boldsymbol{\lambda}$ to focus on the tuneable objectives. For the other objectives occupying entries in the reward vector, the preference weights are dynamic. The reward vector for our MORL framework consists of four components as explained below:

$$\boldsymbol{r}_t = (\underbrace{r^t_{\text{core}}}_{\text{static}}, \underbrace{r^t_{\text{demo}}, r^t_{\text{distance}}, r^t_{\text{efficiency}}}_{\text{dynamic objectives}}) \tag{5.1}$$

5.3.4.1 Navigational Core Objectives

Independent of preferences, the agent must exhibit goal pursuit and collision avoidance. Goal-oriented navigation is achieved by a continuous reward $r^t_{\text{goal}} = 125 \cdot (d^t_g - d^{t-1}_g)$, based on the change in distance $d_g = |\boldsymbol{p}_g|$ from the goal. The total cumulative goal reward $R = \sum_{t=0}^{T} r^t_{\text{goal}}$ is non-discounted to remain independent of the number of steps to the goal, avoiding a bias towards shortest paths and thus the efficiency preference objective. Collision avoidance uses a sparse penalty $r^t_{\text{collision}} = -1{,}000$ for contact between the robot and any obstacle. The core reward function is $r^t_{\text{core}} = r^t_{\text{goal}} + r^t_{\text{collision}}$.

5.3.4.2 Tuneable Preference Objectives

Our three user-centric style objectives cover demonstration-reflection, efficiency, and proxemics: To include proxemics, an important comfort factor in human-aware navigation, we define a quadratic distance penalty for positional closeness $d_h = |\boldsymbol{p}_h|$ to the human within a range $d_{\text{thresh}} = 2\ \text{m}$ as

$$r_{\text{distance}} = -10\frac{(d_h - d_{\text{thresh}})^2}{(d_{\text{thresh}} - d_{\text{min}})^2} \text{ if } d_h \leq d_{\text{thresh}}, \tag{5.2}$$

else zero, with $d_{\text{min}} = 0.3\ \text{m}$.

The second style objective is navigational efficiency, or shortest path navigation, implemented with a constant time penalty $r^t_{\text{efficiency}} = -10$.

The third and last objective is demonstration-like behavior r^t_{demo}, as elaborated below. Note that all rewards of the tuneable objectives are defined as penalties with a uniform range of $[-10, 0]$.

5.3.4.3 Demonstration Acquisition and Reward

Demonstrations can capture nuanced navigation styles that are difficult to express using analytical reward functions, such as characteristically shaped trajectories when ap-

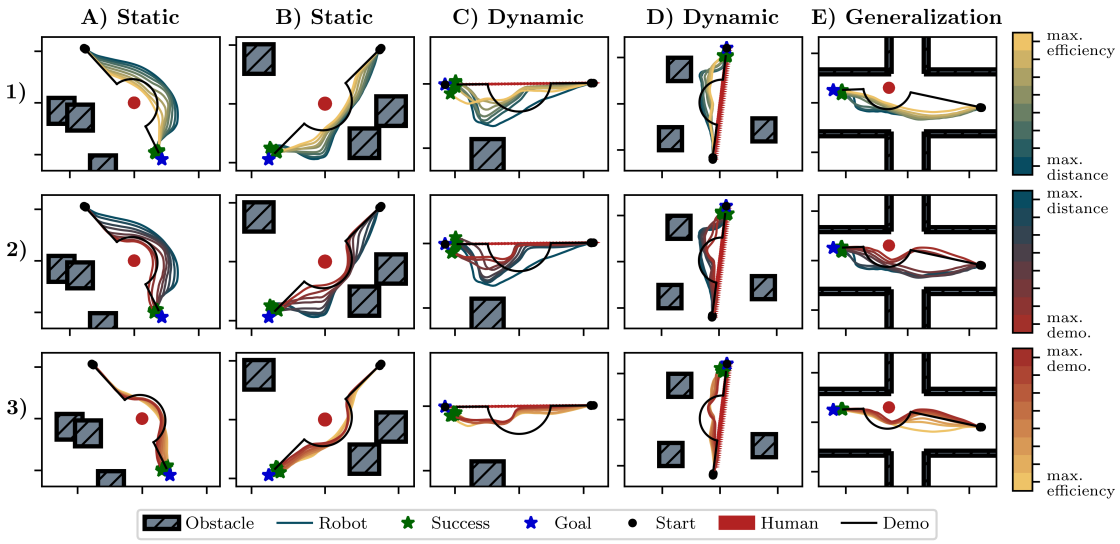


Figure 5.3: Trajectory rollouts in simulation for different preference vectors **(rows)** and different scenes with a static and a dynamic approaching human **(columns)**. As can be seen, the navigation policy shifts its behavior according to the set preference. The colorbars on the right indicate the interpolated preference space Λ_i for each plot row. Static scenarios such as **(A+B)** were covered during training, while a moving human **(C+D)** and the corridor environment **(E)** test for generalization. While shifting **Row 1)** from shortest driving behavior under the maximum efficiency preference (yellow) to distance-keeping (blue), the minimum distance from the human increases. At the same time, the agent has developed a tendency to navigate alongside obstacles when they are located near the path. Shifting towards the maximum demonstration preference **(Row 2)**, the trajectory shapes increasingly resemble the demonstration pattern (black). On the shift back to maximum efficiency **(Row 3)**, the demonstration pattern disappears in favor of shortest trajectories. Comparing the static **(A+B)** vs. moving human **(C+D)**, the demonstration preference reflection becomes less distinct as the agent struggles to follow the static pattern that moves with the now dynamic human, yet efficiency and distance preferences keep up with a moving human. In the corridor intersection scene **(E)**, not included during training of the policy, the agent successfully accounts for the wall, reducing the possible distance-keeping to the human. The varied angle between human and goal from the robot's perspective does not prevent the policy from first approaching the human under the maximum demonstration preference, before continuing towards the goal.

proaching the user. In this work, we rely on a predefined optimal demonstration pattern, see Figure 5.3.A1 (black line), where the robot circumnavigates the human in a distinct circular manner. After directly approaching the human, at $d_h = 1\text{m}$, it executes a $90°$ left-hand turn and orbits the human clockwise at a radius d_h. Once between human and goal, it turns left and proceeds directly towards the target. While not being user demonstrations, the distinct pattern enables a clear performance analysis, as its behavior is by design contradictory to the other two objectives, efficiency and distance-keeping. Specifically, the trajectories are only partially goal-directed, conflicting $r^t_{\text{efficiency}}$, and traverse close to the human at $d_h = 1\,\text{m}$, contradicting r^t_{distance} with an impact radius of $2\,\text{m}$. Anchored solely around the human and the goal position, we can easily augment the single demonstration trajectory by rolling it out N_D times in

randomized obstacle configurations, recording only collision-free rollouts. The resulting dataset is handed to the D-REX pipeline, as elaborated in Section 5.3.3. The final reward term is $r^t_{\text{demo}} = -10 \cdot (\hat{R}_\theta(s_t, a_t) - 1)$.

5.4 Experimental Evaluation

Our experimental evaluation is conducted to validate the following claims:

- C1: The D-REX-based reward model successfully captures and teaches the demonstration patterns to the agent.
- C2: We learn a preference-adaptable, demonstration-modulating, yet reliable navigation policy.
- C3: PD-MORL is crucial to successfully learn our robot navigation task.
- C4: Our policy generalizes from simulation to the real world, even on a robot not used for training.

Our evaluation concludes with a sim-to-real transfer and evaluation on two robots.

5.4.1 Training and Environment

We train using the iGibson simulator [63] with a simulated Kobuki TurtleBot 2. Robot start and goal positions are randomly sampled, 6 to $12\ \mathrm{m}$ apart in open space. A static human is placed between them, aligning with a static-human demonstration pattern. Three static rectangular obstacles are randomly placed, avoiding occupied positions. The robot must navigate to the goal while avoiding both the human and obstacles, which may conflict with the human distance-keeping objective. An episode terminates upon successfully reaching the goal, robot collision, or a timeout after 300 steps. Training is conducted for 600k steps across $C_p = 3$ environments, using $\gamma = 1.0$, and the final model is used for evaluation. For the evaluation of generalization to dynamic environments only, not training, we simulate a moving human approaching the robot with an opposite start goal configuration.

5.4.2 Qualitative Navigation Analysis

Figure 5.3 shows navigation strategies of our MORL agent in static (A+B+E) and dynamic human (C+D) scenarios in simulation, under varying preference weights and obstacle configurations. Three subplot rows interpolate convex preferences between pairwise combinations of two objectives, with the third objective fixed at zero. In Row 1, preferences interpolate between distance and efficiency, parameterized by $\mu \in [0, 1]$, with the vector $\boldsymbol{\lambda}_1(\mu) = (0, \mu, 1 - \mu)$. The other rows follow similar pairwise combinations. The resulting set of $\boldsymbol{\lambda}_i(\mu)$ is $\Lambda_i = \left\{ \left(\frac{i}{N}, 1 - \frac{i}{N}, 0 \right) \mid \mu = \frac{i}{N}, i = 0, \ldots, N \right\}$ with $N = 10$, forming the test set $\Lambda = \Lambda_1 \cup \Lambda_2 \cup \Lambda_3$ with a total of 33 preference vectors, see Section 5.4.3.

The plots depict the robot's trajectories from an initial point (black dot) to a goal (blue star), considering static obstacles and a human (red circle & arrow), with the optimal demonstration trajectory (black line) included.

Starting with the static human in Figure 5.3A+B, the shift from efficiency to distance-keeping (Figure 5.3.1) shows increasing human distance along the path, with the robot eventually passing closely without collision, reducing path length due to the efficiency penalty $r^t_{\text{efficiency}}$. Under maximum human distance preference, the robot occasionally stays close to obstacles before turning towards the goal after passing them.

For the shift from distance-keeping to demonstration-like behavior (Figure 5.3.2), the minimum distance from the human decreases. Supporting C1, trajectories shape into the characteristic demonstration pattern of a straight approach, circular circumnavigation, and a goal-directed turn, yet sharp corners near the human are less pronounced than in the demonstration.

Finally, shifting preferences from demonstration back to efficiency (Figure 5.3.3), demonstration-driven trajectories bend around the human, while efficiency-driven ones head directly to the goal after passing. When obstacles are near the human, collisions are avoided, though at reduced distance. Under maximum distance preference, human distance is maintained before and after obstacles, and all trajectories pass the human on the right, following the demonstration pattern.

To further evaluate the generalization and robustness of our policy, we test it in a moving human environment and a previously unseen scene. In this dynamic setting, which was not covered during training, a human approaches at $0.5\,\mathrm{m/s}$ (Figure 5.3.C+D), the efficiency and distance-keeping objectives are maintained without collisions. The avoidance maneuvers occur more abruptly than in the static case, bending sharply away from the human. As expected, the demonstration pattern is less followed, with the orbiting part shrinking or not completed due to the moving human.

Similarly, we assess generalization and robustness in an unseen corridor intersection scenario (Figure 5.3.E). The agent successfully accounts for the presence of the wall, which limits the possible distance it can maintain from the human. Despite the varied angle between the human and the goal from the robot's perspective, the policy prioritizes initial approach behavior, aligning with the maximum demonstration preference, before continuing toward the goal. This indicates that the learned policy generalizes to unseen spatial configurations while adhering to key objectives.

These results provide evidence for C1 and C2, showing the robot's ability to adjust its behavior from human-distant to demonstration-driven and efficiency-focused navigation.

5.4.3 Quantitative Analysis

5.4.3.1 Preference Reflection

We conducted a quantitative evaluation of the preference-reflecting agent using multiple performance and navigation metrics (Figure 5.4). The agent was tested across

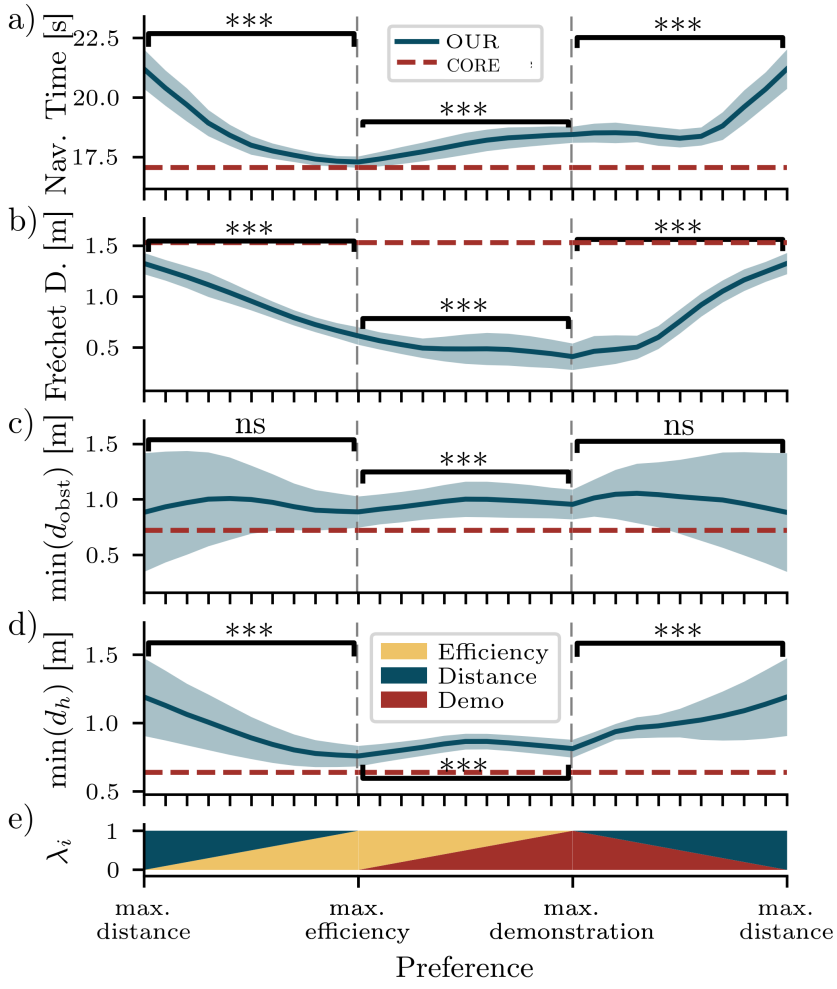


Figure 5.4: Quantitative metrics of OUR agent for different preference configurations **(e)**, tested for statistical significance for dissimilar means between the maximum preferences, with *** for $p < .001$, and ns for not significant. **a)** The navigation time is smallest for maximized efficiency preference, as expected. **b)** The Fréchet distance to the demonstration trajectory decreases as the demonstration preference increases. **c)** The minimum distance to any obstacle is measured using the lidar. **d)** The minimum distance from the human grows with its preference weight. The preference-independent non-MORL policy CORE (red dotted line) that only obeys the navigational core reward term r_{core} of collision avoidance and goal pursuit is included in each plot.

100 episodes in random environments, using different interpolated preference weights $\boldsymbol{\lambda} \in \Lambda$ (colored fractions in Figure 5.4e; see Section 5.4.2). Statistical significance between mean values for the maximum preferences was assessed using a Student's t-test with Bonferroni-correction.

The agent (OUR) achieved a success rate of 100 % with no timeouts or collisions (Table 5.1, first column). As the distance preference increases, both minimum human distance and navigation time rise (Figure 5.4a+d), indicating longer trajectories to maintain greater human distance.

To assess how well the demonstration trajectory is reflected (claim C1), we computed the Fréchet distance [123] between the demonstration and executed trajectories (Fig-

ure 5.4b). The minimum mean Fréchet distance of $0.41\ \mathrm{m}$ occurs when demonstration preference is maximized. Efficiency and distance-keeping preferences also reduce the Fréchet distance, as the demonstration path passes close to the human.

Comparing the trends of minimum obstacle distance ($\min(d_{\text{obst}})$, Figure 5.4c) and minimum human distance ($\min(d_h)$, Figure 5.4d), the agent clearly distinguishes between humans and static obstacles. As the human distance preference increases, the robot maintains a larger distance from the human, while staying close to obstacles, accepting higher collision risk to prioritize proxemic preferences.

Our quantitative analysis supports the findings from the qualitative evaluation, providing measurable evidence for research claims C1 and C2.

5.4.3.2 Ablation Study

We ablated the architecture with respect to the state space and demonstration reward model, compare Table 5.1. The state space changes apply to all involved models: D-REX BC policy, D-REX reward model, actor, and critic. The ablations cover exclusion of human position (OUR-NH), removal of the action a_t as input to the reward model leaving $r^t_{\text{demo}} = \hat{R}_\theta(s_t)$ (OUR-RM), and the combination of both (OUR-RM-NH). Note that the maximum preference vectors in Table 5.1 are $\boldsymbol{\lambda}_{\text{demo}} = (1, 0, 0)$, $\boldsymbol{\lambda}_{\text{dist}} = (0, 1, 0)$, $\boldsymbol{\lambda}_{\text{eff}} = (0, 0, 1)$, respectively.

Compared to OUR, removing the human position from the state space in OUR-NH and OUR-RM-NH reduces distance-reflection capabilities. This is expected due to the correlation between human position and distance preferences in demonstrations. While OUR-RM performs with a similar collision rate, its preference reflection is slightly weaker than OUR.

Metric	$\boldsymbol{\lambda}$	OUR	-NH	-RM	-RM-NH	SAC-PR	-PR-γ
SR↑ [%]	Λ	**100**	96.8	100	79.6	45.4	54.5
CR↓ [%]	Λ	**0**	2.7	0	11.4	53.2	44.4
TR↓ [%]	Λ	**0**	0.5	0	9.0	1.2	1.1
$\min(d_h)$↑ [m]	$\boldsymbol{\lambda}_{\text{dist}}$	1.18	0.52	1.16	0.48	1.06	0.91
Fréchet↓ [m]	$\boldsymbol{\lambda}_{\text{demo}}$	**0.41**	0.57	0.46	0.49	-	1.06
Nav. time↓ [s]	$\boldsymbol{\lambda}_{\text{eff}}$	17.3	**16.9**	17.4	19.2	-	20.8

Table 5.1: Quantitative analysis, ablation, and baseline study with respect to the state space and reward model, bold number highlighting the highest performance. For the ablation identifiers and preference vectors $\{\boldsymbol{\lambda}_{\text{dist}}, \boldsymbol{\lambda}_{\text{demo}}, \boldsymbol{\lambda}_{\text{eff}}\}$, please refer to Section 5.4.3.2. For brevity, the identifiers are shortened after OUR, so that, e.g., -NH corresponds to OUR-NH with the human pose state excluded. The baselines with parameterized rewards are denoted with SAC-RP and SAC-PR-γ, short -PR-γ. The results were averaged over 100 trajectories for single $\boldsymbol{\lambda}$, and for the success rate (SR), collision rate (CR), and timeout rate (TR) additionally over all $\boldsymbol{\lambda}_i \in \Lambda$, precisely $33 \times 100 = 3,300$ trajectories. The baseline SAC-PR had no successful trajectories under $\boldsymbol{\lambda}_{\text{demo}}$ and $\boldsymbol{\lambda}_{\text{eff}}$.

5.4.4 MORL Baseline

As single-policy MORL approaches with continuous action spaces are scarce due to the novelty of PD MO-TD3-HER, we implement an equivalent actor-critic-based MORL baseline with parameterized reward (-PR), analogous to the baselines in [177]. The learning task characteristics and reward vector remain unchanged, while the four performance-boosting modifications of PD-MORL drop out, compare Section 5.3.2.

During training, convex preference weights are sampled at the beginning of each episode. Among actor-critic implementations, TD3 failed to converge on the task, whereas SAC [187] achieved better results. Performance further improved when adjusting the discount factor from $\gamma = 1.0$ to γ = 0.98 in SAC-PR-γ (see Table 5.1). Nevertheless, both SAC-PR and SAC-PR-γ average in success below $55\,\%$. Note that SAC-PR and SAC-PR-γ show weaker preference reflection as compared to OUR, while SAC-PR failed entirely on the edge-case preferences $\boldsymbol{\lambda}_{\text{demo}} = (1, 0, 0)$ and $\boldsymbol{\lambda}_{\text{eff}} = (0, 0, 1)$. The results highlight the superiority of PD-MORL for learning the robot navigation task, supporting C4.

5.4.4.1 Non-MORL Core Navigation Agent

To contextualize the core navigation objectives, we train and quantitatively evaluate a preference-independent, non-MORL policy CORE that optimizes only the navigational core rewards r_{core} (goal and collision), compare the red-dotted line in Figure 5.4. Two metrics stand out: The MORL agent prioritizes obstacles over humans, while the non-MORL baseline, lacking a human-distance reward, treats both similarly. This results in comparable minimum values ($d_h = 0.64\,\text{m}, d_{\text{obst}} = 0.72\,\text{m}$), contrasting with our MORL agent. Its higher demonstration Fréchet distance further confirms the absence of demonstration knowledge.

5.4.5 Real-World Transfer

We evaluated our tuneable policy on a Kobuki TurtleBot 2 using ROS [108] and transferred the TurtleBot-trained policy to a Toyota Human Support Robot (HSR).[5] The agent received ground truth human and goal positions, with the dynamic human localized via a Vive VR tracker. The HSR's lidar, mounted in the front of its rotation center, may cause state space discrepancies for the policy. Due to its $270°$ coverage, compared to the TurtleBot's $360°$ lidar, the rear distance readings were filled with the maximum range of $4\,\text{m}$. The procedure ensures state consistency under the conservative assumption that rear obstacles are unlikely to impact navigation, as the robot can only move forward. Another discrepancy arises in velocity command execution, both in sim-to-real transfer and between robots, due to differences in actuator dynamics and drive mechanisms, potentially affecting navigation performance. We ran navigation tests on both robots for the preference vectors $\boldsymbol{\lambda} \in \Lambda$ with $N = 5$ (see Section 5.4.2).

[5] A video of the real-world experiments is linked in the supplemental material section of the thesis appendix.

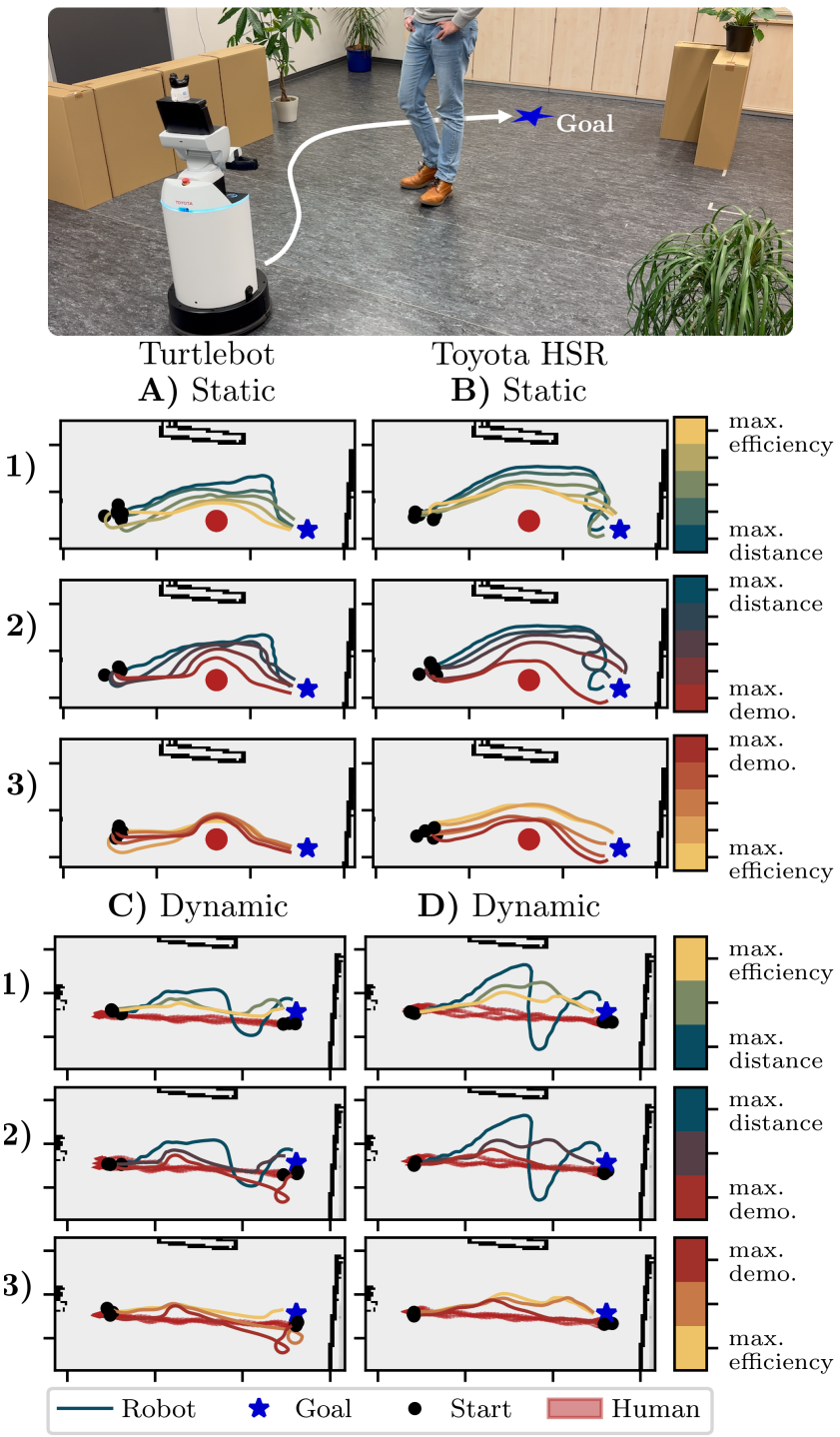


Figure 5.5: Real-world experiment setup **(top)** and results **(bottom)** with the policy OUR in a sim-to-real transfer with the Kobuki TurtleBot 2 **(left)** and the Toyota HSR **(right)**. With a static human as during training **(A+B)**, the navigation behavior in the real world successfully reflects varying preferences on both robots. While the TurtleBot exhibits better demonstration reflection, the HSR keeps more distance from the human under the maximum distance preference. With a dynamic approaching human **(C+D)** that was not accounted for during training, the preference reflection decreases.

The recorded TurtleBot trajectories are shown in Figure 5.5.A and the HSR trajectories in Figure 5.5.B. Both robots adapt their behavior according to preferences. For the maximum distance preference (Figure 5.5.A1), the TurtleBot shows oscillations, presumable due to slight over-steering, while the HSR drives closer to obstacles and exhibits a wider oscillatory motion near the goal (Figure 5.5.B1). These differences may result from lidar state mismatches (e.g., positional offset) or slower action execution due to inertia. For maximum demonstration reflection, the TurtleBot's trajectory aligns better with the demonstration than the HSR (Figure 5.5.2).

Both robots avoid collisions with dynamically approaching humans (Figure 5.5.C+D). As in the dynamic simulation experiments (Figure 5.3), avoidance sharpens for the demonstration objective but fades as the human and robot pass each other. Under the distance preference, sharper inward steering and subsequent overshooting behind the human in simulation become more pronounced in the real world, compare (Figure 5.3.C1+D1). We attribute the sharper inward steering to the static training environment, which prevented the agent from learning in the presence of a moving human. Under static conditions, the agent typically maintains a fixed distance on the human's side, forming a distance-angle mapping for avoidance. This mapping is disrupted by the dynamic human, causing the agent to turn inward as the human passes more quickly. Efficiency-focused behavior transfers flawlessly. Despite minor sim-to-real differences, all real-world trajectories remained collision-free, demonstrating robust sim-to-real generalization. In conclusion, the policy transfers smoothly to real robots, supporting research claim C4.

5.5 Conclusion

In summary, we introduced an innovative framework fusing multi-objective reinforcement learning (MORL) with demonstration-based learning for adaptable, personalized robot navigation around a user with changing preferences. Our approach successfully modulates the conflicting objectives of demonstration data reflection, distance keeping, and navigational efficiency without retraining, a direct contribution to the aspect of adaptability of RQ4, compare Section 1.2.4. To achieve this, we distill demonstration data into a reward model that shapes the agent's trajectories during navigation with variable strength. In various qualitative and quantitative experiments, we demonstrated the adaptability to varying preferences and scenarios. Finally, we successfully deployed the learned agent on two real robots.

A constraint of our approach is the inability to alter the demonstration data itself without retraining. So while the amount of demonstration-reflection can be modulated, instantaneous preference modulation remains constrained within the preference space defined by the demonstrations and the other chosen objectives. Generally, this presents an interesting avenue for future research.

The approach presented in this chapter provides a principled way to adapt to chang-

ing user preferences by accepting a preference vector as input for external control, in effect, establishing a well-defined data protocol for representing preferences. To provide an outlook on how these preference vectors might be leveraged by users or other agents, Hwang *et al.* [30] proposed an approach that uses large language models (LLMs) to translate human feedback into preference vectors. Additionally, they employ optimization-based routines for translating preferences from comparative feedback and demonstrations. Lee *et al.* [181] presented a hybrid approach in which a high-level skill agent adjusts the behavior of a low-level tunable navigation agent based on scene and task context. These examples illustrate promising future research directions for linking user preferences to context, potentially through an agent dedicated to predicting context-based preference vectors.

This chapter is the last chapter integrating the preference expression modality of demonstrations, as the next two chapters focus on the paradigm of learning from human feedback. Specifically, the next chapter places a focus on efficient user querying in RLHF settings for improved information gain (cf. RQ1, Chapter 1.2.1). However, the here-presented configuration of policy state space roughly remains over the next chapters, i.e., a combination of min-pooled 2D lidar paired with human and goal position.

6 EnQuery: Ensemble Policies for Diverse Query-Generation in Preference Alignment of Robot Navigation

Abstract

To align mobile robot navigation policies with user preferences through reinforcement learning from human feedback (RLHF), reliable and behaviorally diverse user queries are required. However, deterministic policies fail to generate a variety of navigation trajectory suggestions for a given navigation task. In this chapter, we introduce EnQuery, a query generation approach using an ensemble of policies that achieve behavioral diversity through a regularization term. For a given navigation task, EnQuery produces multiple navigation trajectory suggestions, thereby optimizing the efficiency of preference data collection with fewer queries. Our methodology demonstrates superior performance in aligning navigation policies with user preferences in low-query regimes, offering enhanced policy convergence from sparse preference queries. The evaluation is complemented with a novel explainability representation, capturing full scene navigation behavior of the mobile robot in a single plot.

6.1 Introduction

For optimal human-robot interactions, robots should customize to user needs. While policies demonstrating superior capabilities are developed through learning-based systems, there arises a need for methods to align these policies with user preferences [57], [60]. Reinforcement learning from human feedback (RLHF) is a state-of-the-art method, where user preferences are transferred into a reward model that aligns policies that interact with the human in the field of large language models [188], or robotics [62], [68], [189], [190], [191]. In the context of mobile robot navigation, humans exhibit diverse preferences about comfortable robot approaching behavior and proxemics [14], calling for preference-aligned navigation policies [69], [169], like the approaches in Chapters 2 and 3.

A core optimization goal in RLHF is to maximize information gain achieved by querying the user [60], [192], in accordance with this thesis' RQ1 for efficient preference collection. Not only does this minimize the effort and fatigue associated with repetitive queries of the user, but it also enhances the quality of collected preference data. Besides query diversity [193], a reliable query test result is essential [192]. However, inconsistent query results due to low reliability undermine their corresponding preference information gain. The reliability can, however, be increased when all test variables are kept con-

This chapter is a revised and updated version of the peer-reviewed publication [78]. Refer to Section 1.4 for details.

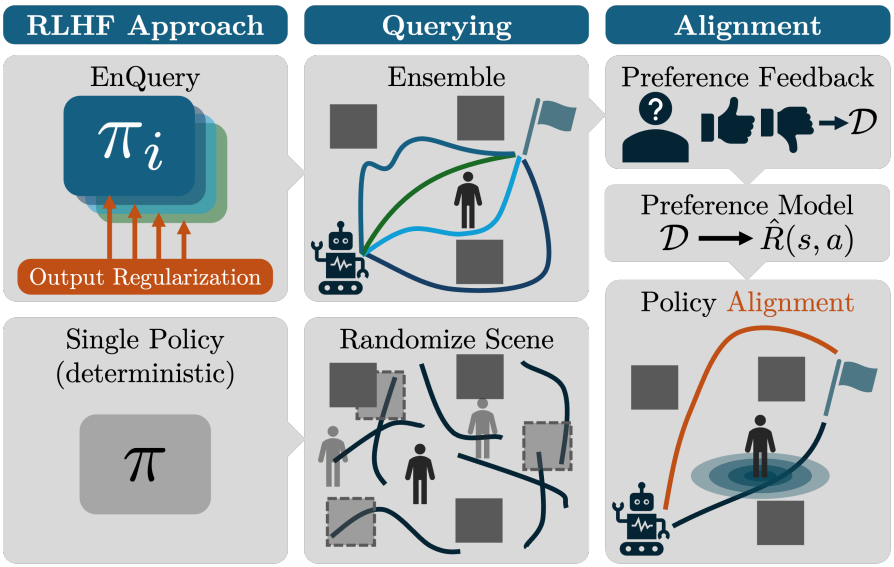


Figure 6.1: Our ensemble of RL policies generates a variety of trajectories for a given navigation task as queries for RL from human feedback. In contrast, deterministic policies are limited to just one trajectory, and the queries' variety depends on trajectory segments from randomized scene configurations. As a result, EnQuery facilitates a higher preference information gain for low-query numbers.

stant across measurements. So to extract consistent user preferences, it is advisable to minimize changes in variables associated with a single query. One approach is to keep the task environment constant while only altering the agent's behavior. This improves the reliability and quality of information extracted from the user feedback.

In a typical robotics RLHF setup employing a deterministic policy, a pool of query trajectories is generated using policy rollouts of different environment configurations [57]. Subsequently, the trajectories are subsampled into segments and presented to the user as pairwise preference queries. However, the diversity of environment task configurations in the query pool and the lack of common reference points in the segments conflict with the concept of re-test reliability through minimal change in variables, as elaborated above.

In contrast for the alignment process of large language models (LLM), it has become a best practice to ask for user ranking between two different outputs for a single input prompt [194]. This is possible due to the generative and non-deterministic nature of these LLMs. With deterministic policies however, this approach would result in two identical outputs, leaving no room for a preference choice of the user.

In this chapter, we therefore present ensemble policies as a possible solution for query diversity under identical policy input on deterministic policies, see Figure 6.1. On a given navigation task, we train the ensemble with a regularization term that encourages dissimilar outputs of the individual ensemble members.

We apply the proposed method to a human-centric robot navigation task employing ensemble queries to capture distance-related preferences. We comprehensively test this approach starting with an analysis of the queries generated by the ensemble. We then

assess a reward model trained on these queries and finally align a baseline-objective navigation policy to reflect the collected preferences.

The **main contributions** of our work are the following:

- EnQuery, an ensemble approach for query generation enabled by a regularization term that ensures behavioral ensemble diversity.
- An extensive quantitative and qualitative analysis of the full query-to-alignment pipeline demonstrating superiority of our method in preference reflection for low-query numbers, compared to a state-of-the-art baseline approach, ultimately enabling more query-efficient feedback processes.
- A novel visualization scheme for enhanced behavior explainability of mobile robot navigation policies of mobile robots that captures full scene behavior in a single plot.

6.2 Related Work

In the domain of robotics, reinforcement learning from human feedback (RLHF) presents a promising solution to the challenges of defining and optimizing reward signals that maximize user preferences. By incorporating human judgments, RLHF enables robots to adapt their learning objectives, ensuring behaviors align with human preferences. This method has shown significant potential in enhancing interaction tasks, robot manipulation [62], and robot navigation [68], [169]. Here, open challenges are the optimization of the human feedback process with respect to efficiency, information gain, and reduction of psychological biases [60].

When it comes to the choice of queries in RLHF, typically a pool of randomly generated trajectories is used. With the goal of optimal information gain in mind, query selection algorithms have emerged to surpass the naive approach of random sampling, thereby reducing the required number of interactions for preference acquisition. Christiano *et al.* [57] choose queries either via uniform sampling or to maximize the variance in an ensemble of reward models. Marta *et al.* [68] maximize the distance in a latent space representation of the trajectories generated through a variational autoencoder. In contrast, we do not perform a selection of queries from a random pool based on certain criteria, but directly generate trajectory queries using the behavior-diverse policy ensemble for a given environment configuration.

Furthermore, psychological factors influence the preference and preference-consistency of the human. As such, serial position effects can cause the start and end of a trajectory to over-proportionally influence the user [60]. We counteract this effect with the design of our navigation ensemble query approach with the alignment of start and end, where the trajectories originate from a common starting point and converge again at the goal location. While our work does not directly investigate the user experience of the proposed querying approach, it is motivated by the need to tailor query

generation to specific tasks for more efficient and intuitive human feedback. In line with this motivation, Dennler *et al.* [12] present a query selection algorithm designed to improve user experience during preference feedback in assistive robot tasks. Through a user study, they show that their method prioritizing user intuitiveness results in significantly higher satisfaction among users compared to baseline approaches. These findings underline the importance of aligning query generation strategies with the specific RLHF task setting in a user-aware manner.

Typically, ensemble strategies are used for measuring uncertainty in learning-based models [195]. In the scope of reinforcement learning, applications of ensembles aim to improve the learning process, more specifically by stabilizing Q-learning and balancing exploration and exploitation [196]. Lee *et al.* [196] randomize the model weight initialization and bootstrap the data presented to each policy during an update. Sheikh *et al.* [197] encourage representation diversity through regularization terms with a similar goal. The lack of such regularization terms was found to cause alignment of the ensemble members over the course of training, even if the networks are differently initialized. In our work, we apply a regularization term on the policy outputs in the ensemble of TD3 agents for maximum behavioral diversity on a given input.

Since the publication of our original study, foundation models have demonstrated increasing potential in preference-learning tasks. A particularly promising direction involves replacing or augmenting human feedback with foundation models, leveraging them as synthetic teachers to reduce the reliance on manual annotations. For instance, Wang *et al.* [198] synthesize preference labels from vision-language models (VLMs) by querying them to evaluate image observations in alignment with textual task descriptions. The system learns rewards without human labeling, and shows promising performance on complex robotic tasks such as the manipulation of deformable objects. Another study combines synthetic feedback from various large language models (LLMs) for more consistent and reliable preference labels [199], thereby improving task learning. Beyond replacing human feedback, LLMs have also been used to increase the information gain of individual queries. Holk *et al.* [191] enrich preference feedback with optional natural language prompts and highlight state-action pairs of high informational value, thereby refining the learning signal. While these approaches primarily focus on the feedback rather than generating queries, they are complementary to our work and illustrate the broader trend of incorporating pretrained models to improve sample efficiency in preference-based reinforcement learning.

6.3 Preliminaries

6.3.1 Problem Definition

We consider a social robot navigation scenario, where the robot pursues a goal in a human environment among static unknown obstacles. The robot is aware of the location of a single human in its vicinity and uses 2D lidar data to sense obstacles. A user has

specific preferences regarding the robot's navigational behavior, such as proxemics and path selection, and expresses them through pairwise comparisons of trajectories. We learn a behavior-diverse policy ensemble, in which each policy's linear and angular velocity commands take a different trajectory for a given navigation task with start and goal positions, while avoiding collisions with obstacles and the human. The resulting trajectory options represent trajectories for A$\succ$B preference comparisons. The navigation policy is obtained using reinforcement learning, as elaborated below.

6.3.2 Reinforcement Learning of Point Navigation

In reinforcement learning, the objective is to optimize state transitions $s_t \rightarrow s_{t+1}$ of a Markov Decision Process, leading to a reward r_t for executing action $a_t = \pi(s_t)$ at time step t, based on the policy π [105]. These sequences (s_t, a_t, r_t, s_{t+1}) are identified as state-action pairs. The optimization goal is to maximize the total return $R = \sum_{i=t}^{T} \gamma^{(i-t)} r_t$, which represents the sum of γ-discounted rewards from time t onward. We use the Twin Delayed Deep Deterministic Policy Gradient (TD3) framework for continuous action spaces. We employ the TD3 implementation of Stable-Baselines3 [200]. All important parameters and notations of our work are listed in Table 6.1.

6.3.2.1 State Space

The state space includes the goal, human position, and obstacles detected by a 2D lidar sensor. Explicitly providing the human position allows the policy to differentiate it from obstacles, fostering human-centric navigation. The environment is observed through a downsampled $360°$, $6\,\mathrm{m}$ lidar, reduced to $N_L = 30$ rays as $\mathcal{L}_t = d_i^t | 0 \leq i < N_L$. The local goal $\boldsymbol{p}_g$ and human position $\boldsymbol{p}_h$ are given in robot-centric polar coordinates. The state vector $s_t = (\boldsymbol{p}_g, \boldsymbol{p}_h, \mathcal{L}_t)$ is processed by separate 2-layer MLP feature extractors (64 units), then concatenated and passed to the TD3 actor-critic networks (128 × 400 × 300 × $1, 2$).

6.3.2.2 Action Space

The policy outputs velocity control commands of linear and angular velocity as $a_t = (v, \omega)$ that directly drive the robot within a range of $v \in [0, 0.5]\,\mathrm{m/s}$ and $\omega \in [-\pi, +\pi]\,\mathrm{rad/s}$.

6.3.2.3 Reward

For the navigation task, we employ the reward function of basal navigation objectives such as goal pursuit and collision avoidance as

$$r^t = r^t_{\text{goal}} + r^t_{\text{collision}} + r^t_{\text{timeout}} + r^t_{\text{loop}}. \tag{6.1}$$

The goal pursuit is encoded in a sparse reward $r^t_{\text{goal}} = c_{\text{goal}}$ for arrival at the goal location such that $d_{\text{g}} = |\boldsymbol{p}_g| \leq 0.4\,\mathrm{m}$. To encourage collision-free navigation, a sparse penalty

$r^t_{\text{collision}} = c_{\text{collision}}$ is provided upon collision. The sparse penalty $r^t_{\text{timeout}} = c_{\text{timeout}}$ penalizes non-goal-oriented behavior after 500 time steps. The last term r^t_{loop} will be explained in Section 6.4.1.

6.3.2.4 Training Environment

For training, we use the iGibson simulator [63] that itself relies on the PyBullet physics engine [107]. In an open space, we first randomly sample a start and goal location for the robot in a range of $2\,\text{m}$ to $10\,\text{m}$. A single human is placed between the start and goal locations. Subsequently, we sample a total of four cubic obstacles at random locations in the vicinity of the human and robot, avoiding already occupied poses. One episode is represented by the rollout of one trajectory in a given environment configuration until one of the following three termination criteria is satisfied: Timeout at more than 500 time steps or $100\,\text{s}$, or successfully reaching the goal position within a threshold of $d_{\text{g}} \leq 0.4\,\text{m}$.

6.4 Our Approach

This section introduces EnQuery with respect to the policy ensemble, the querying methodology, reward model training, and subsequently policy alignment.[6] Ultimately, we present a novel behavior explainability visualization.

6.4.1 Ensemble Generation

We extend the standard single-policy reinforcement learning architecture by introducing a set of N_E policies $\mathcal{E} = \{\pi_i(s_t, a_t) | i \in [N_E]\}$, called the policy ensemble. During training, each ensemble policy is interacting with its own environment instance, and storing the collected experiences into its own replay buffer of size N_B. To achieve behavioral diversity across the ensemble, we use a regularization term to penalize similar outputs. So as a core modification to achieve a diversity of behaviors for a given state s_t, we introduce the novel goal-modulated diversity regularization (GMDR) term

$$\mathcal{L}^i_{\text{GMDR}} = -\tilde{\kappa} \cdot \alpha_{\text{dist}}(d_g) \cdot \sum_{j=0, j\neq i}^{N_E} (a_i - a_j)^2, \tag{6.2}$$

which captures the difference between all pairwise combinations of action outputs $\pi_i(s_t) = a_i$ of the ensemble members i. Here, the scaling factor $\tilde{\kappa} = \kappa / |A|^2$ is normalized by the dimension of the two-dimensional action space A.

A task-specific feature is the goal distance weighting term $\alpha_{\text{dist}}(d_g) = m_{\text{dist}} \cdot d_g + b_{\text{dist}}$ that linearly decays the diversity loss with decreasing distance to the goal d_g. The variables m_{dist} and b_{dist} normalize the term for the expectable distance range to the goal. As a practical motivation, the closer the robot navigates to the goal, the fewer deviations from goal-directed navigation behavior are desired. This helps the convergence of

[6] The code for EnQuery is linked in the supplemental material section in the appendix.

policy training from the goal-reaching perspective, while allowing for greater trajectory diversity when the goal is still far enough away.

In the TD3 architecture, the GMDR is simply added to the loss of the actors as

$$\mathcal{L}^i_{\text{actor}} = \mathcal{L}_{\text{actor}} + \mathcal{L}^i_{\text{GMDR}}. \tag{6.3}$$

We furthermore introduce the reward term r^t_{loop} as a countermeasure for undesired looping behavior that some policies of the ensemble would adapt as a result of the diversity regularization term, see Equation 6.1. Essentially, the looping penalty checks for the self-intersection of the current trajectory, which is sparsely penalized with $r^t_{\text{loop}} = c_{\text{loop}}$. Note that the criterion for self-intersection is applicable solely to trajectory segments that are more than four time steps old. In all other cases, the sparse rewards are zero. The reward function is identical for all agents in the ensemble, and explicitly not the source of ensemble diversity.

Before the ensemble policy is trained, a single policy π_{raw} is trained without GMDR, but on the same reward (Equation 6.1) and task. Subsequently, the replay buffer of the ensemble is initialized with the experiences of the RAW policy, which supports ensemble convergence and training success. Also, all policy members are initialized with the weights of π_{raw}. Finally, the ensemble policies train for $T = 25\text{k}$ time steps, using the learning parameters denoted in Table 6.1. Subsequently, the ensemble is ready to generate trajectory suggestions as queries for a single navigation task.

6.4.2 Querying

Our approach adopts pairwise A$\succ$B preference comparisons as a feedback modality. Preferences $\succ$ are expressed over robot navigation trajectories $\tau = \{(s_0, a_0), (s_1, a_1), ..., (s_T, a_T)\}$, represented by state-action pairs. We indi-

Notation	Value	Description
γ	0.98	RL discount value
T	25e3	Training time steps
N_B	1e6	Replay buffer size
N_L	30	Pooled lidar ray number
c_{goal}	20	Sparse goal reward
$c_{\text{collision}}$	-1.2	Sparse collision reward
c_{timeout}	-20.0	Sparse timeout reward
c_{loop}	-2	Sparse looping reward
N_E	4	Ensemble member number
κ	0.0625	GMDR scaling factor
m_{dist}	1/8	GMDR distance scaling slope
b_{dist}	1/4	GMDR distance scaling intercept
k	20	BL query segment length
λ	0.06	Reward weighting factor

Table 6.1: Notations and parameter settings.

cate $\tau_1 \succ \tau_2$ to indicate preference of trajectory τ_1 over τ_2.

Our ensemble of policies $\mathcal{E}$ generates N_E trajectories for a given environment configuration. From this, we randomly sample a trajectory pair. All self-intersecting and collision-flawed trajectories are filtered. Since we do not conduct a user study but focus on the methodology, we simulate human preferences and query an oracle that will always prefer the trajectory of higher minimum human distance $d_h = |\boldsymbol{p}_h|$. The resulting preference dataset is denoted as $\mathcal{D}_{\text{ens}} = \{\tau_1^i \succ \tau_2^i | i \in [N_Q]\}$ after N_Q oracle queries.

6.4.3 Baseline Querying Approach

We adopt the segment-based uniform querying approach of Christiano *et al.* [57] as a baseline. They achieve diversity not by an ensemble but via randomization of the environment, which translates to randomly generated start, goal, human, and obstacle positions for our environment. A pool of trajectories is generated using the non-ensemble deterministic policy π_{raw}, from which we uniformly sample trajectory segments σ with a length of $k = 20 \simeq 4\,\text{s}$. The preference is subsequently expressed over trajectory segments as $\sigma_1 \succ \sigma_2$, where $\sigma = \{(s_0, a_0), (s_1, a_1), ..., (s_{k-1}, a_{k-1})\}$ denotes a segment sampled from a trajectory τ. The resulting preference dataset is denoted with $\mathcal{D}_{\text{seg}} = \{\sigma_1^i \succ \sigma_2^i | i \in [N_Q]\}$.

6.4.4 Reward Model

To align the navigation policy, we first train a reward model $\hat{R}(s, a)$ from the pairwise preference dataset $\mathcal{D}$ based on the Bradley-Terry model [186], where

$$\hat{P}[\tau_1 \succ \tau_2] = \frac{1}{1 + \exp\left(R(\tau_2) - R(\tau_1)\right)} \tag{6.4}$$

denotes the probability of a human preferring segment $\tau_1 \succ \tau_2$ with the cumulative return $R(\tau_i) = \sum_{(s,a)\in\tau_i} \hat{R}(s, a)$. On that basis, a neural network is trained using a cross-entropy loss such that $\sum_{(s,a)\in\tau_i} \hat{R}(s, a) < \sum_{(s,a)\in\tau_j} \hat{R}(s, a)$ when $\tau_i \prec \tau_j$. The reward model shares the network architecture with the critic and is trained for 10 epochs using a learning rate of 1×10^{-4}, after which the best-performing epoch model is chosen. The output of the reward model is normalized to a distribution mean of zero with standard deviation one.

6.4.5 Policy Alignment

Preference alignment of the navigation policy starts with a converged policy π_{raw}, as introduced above. We take inspiration from the work of Cabi *et al.* [62], who recycle already collected data by updating the existing data buffer with the current reward model. So with data efficiency in mind, we solely rely on the existing replay buffer data for alignment. In other words, it is not necessary during alignment to further explore the environment. A subsequent batch-based policy update on the previous but reward-updated experiences aligns the policy.

To ensure that the aligned policy still obeys the basal navigation objectives defined by Equation 6.1, we balance between the preference reward model and the basal task reward for the updated reward

$$r_t^* = \lambda \hat{R}(s_t, a_t) + (1 - \lambda) r_t \tag{6.5}$$

using the weighting factor $\lambda = 0.06$ determined in preliminary experiments using a grid search. During alignment of π_{raw}, we sample batches from the reward-updated replay buffer as usual and perform 10k policy updates each for one alignment epoch. The models are tested for their navigation success rate after each epoch, where the best-performing epoch is chosen.

6.4.6 Explainability Navigation Plot

In autonomous robot navigation, the interpretability of reinforcement learning (RL) policies is essential. Understanding and foreseeing the robot's actions is key to trust and acceptance, necessitating the development of tools that explain decision-making and ensure it meets human standards.

Recent explainability efforts for RL navigation policies target the reasoning pipeline and resulting behavior [201], [202]. Other works project the learned Q-values into the scene [203]. Yet, we found no visualization to give a complete picture of the behavior across the entire navigation scenario, an important tool to study the quality of navigational preference alignment.

We introduce a novel behavior explanation and visualization method for the navigation policy in a static environment, see Figure 6.5. Based on the concept of a flow fields, it extracts and condenses the preferred navigation direction at all locations at once into a comprehensive bird's-eye-view plot. Firstly, we discretize the environment into a 2D grid of $0.25\,\mathrm{m}$ resolution and place the robot at the center of each traversable cell oriented towards the goal. While keeping the forward velocity at zero, we solely execute the angular velocity command. The robot turns and settles like a compass needle into a certain direction. Whenever the settling results in an oscillation around one direction, we take the mean of the oscillation range. Subsequently, the obtained directional driving preference is recorded together with the magnitude of the corresponding forward velocity. We obtain a matrix of 2D velocity vectors as in a flow field that are visualized using a stream plot. The stream plot sketches the driving behavior across the entire scene at once.

In a second step, we reactivate the forward velocity and roll the trajectories out from each grid cell. Subsequently, the number of traversals through each grid cell is counted to create a heat map, which is visualized behind the stream plot. This completes the picture especially at locations of ending streamlines, whenever the streamline density is too high. For plotting, we use the python library matplotlib [204]. Note that the plotting scheme assumes that the robot is the only entity moving within the scene.

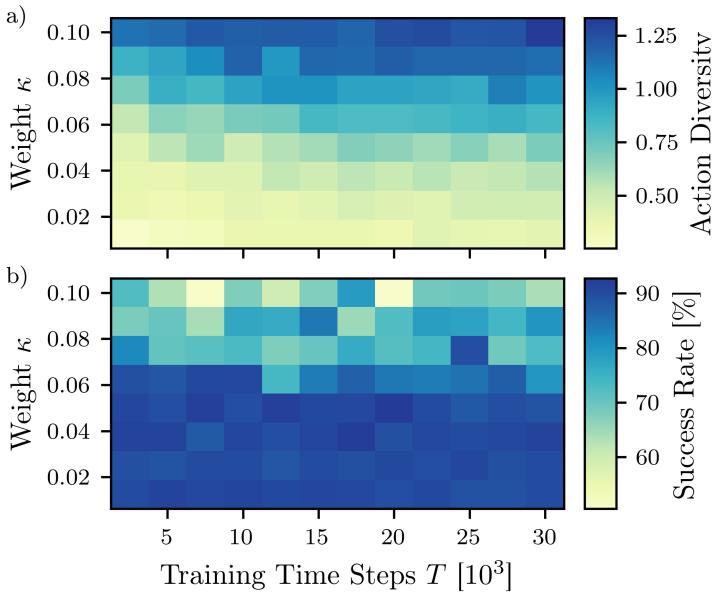


Figure 6.2: **a)** Diversity of actions over the ensemble and **b)** success rate on the navigation task as a function of the total training time steps T and the weighting factor κ of the regularization term. The action diversity grows with the weight κ of the regularization term, while the success rate decreases rapidly for $\kappa > 0.07$.

6.5 Experimental Evaluation

Our experiments investigate the ensemble diversity and success rate with respect to the regularization term Equation 6.2 in qualitative and quantitative measures, the query and reward learning process in comparison to a well-established baseline approach, and finally the preference alignment of the resulting navigation policy.

6.5.1 Ensemble

6.5.1.1 Quantitative

The query ensemble is based on a set of $N_E = 4$ policies that obey the GMDR. First, we evaluate the influence of the κ-scaled GMDR on the learning behavior and diversity of the ensemble. In dependence of the total training time steps and the GMDR's scaling factor κ, the raw action diversity $\sum_{j=0, j\neq i}^{N_E}(a_i - a_j)^2$ is computed for 1,000 randomly sampled states from the replay buffer, see Figure 6.2a. Also, the success rate averaged over 100 trajectories and all ensemble policies is visualized, see Figure 6.2b. Generally, an increasing action diversity can be observed with a growing scale of the regularization term, while the success rate decreases rapidly for $\kappa > 0.07$. Furthermore, the diversity grows with increasing training time steps, without an obvious decrease of the success rate.

Based on the grid search, we settled for an optimal configuration of $\kappa = 0.0625$ and $T = 25$k training time steps. Here, the ensemble achieves a success rate of $91\,\%$ at an action diversity average of $\kappa = 0.7$. To put this in contrast, the non-ensemble agent π_{raw}

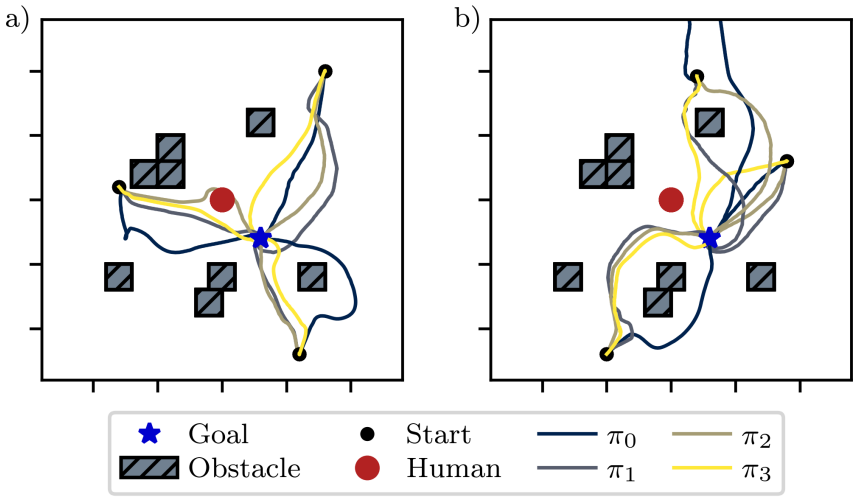


Figure 6.3: Trajectories of the ensemble policies π_i for a given obstacle configuration and randomized start position. Each plot **a)** and **b)** shows three individual start positions. Distinct diversity of the trajectory pathways can be observed.

achieves a success rate of $94\,\%$.

In preliminary experiments we have also experimented with a regularization penalty that was not goal distance-scaled and $\alpha_{\text{dist}} = 1$, resulting in lower success rates for similar action diversity. For optimizing our task setup, we found the goal modulation beneficial.

6.5.1.2 Qualitative

We visualize the trajectories of all ensemble policies π_i in Figure 6.3 for a given environment configuration of eight obstacles. The trajectory shapes vary due to the enforced output diversity, as expected. The diversity spans from avoiding obstacles on the other side (compare π_0), to keeping different distances from the human in the vicinity of the robot. If straight-line navigation to the goal is possible, one policy usually takes the shortest route while the others meander in curvier and longer trajectories.

6.5.2 Reward Model

We analyze the information gain for our ensemble query method (EnQ) against the baseline (BL) of segment-based uniform sampling by Christiano *et al.* [57]. Our measure for the information gain is the prediction accuracy of the reward model on a test dataset in dependence of the number of queries N_Q. Specifically, we query the oracle that prefers trajectories with higher human distance for a total of N_Q times to generate a preference dataset using both our ensemble queries ($\mathcal{D}_{\text{ens}}$) and the segment baseline ($\mathcal{D}_{\text{seg}}$). The accuracies are tested on test similar-sized splits of both the ensemble and segment datasets, respectively, providing both a normal and a cross validation. As can be seen in Figure 6.4, we outperform the segment-based baseline approach for lower query numbers. For higher query numbers $N_Q \geq 18$, the baseline achieves a higher test

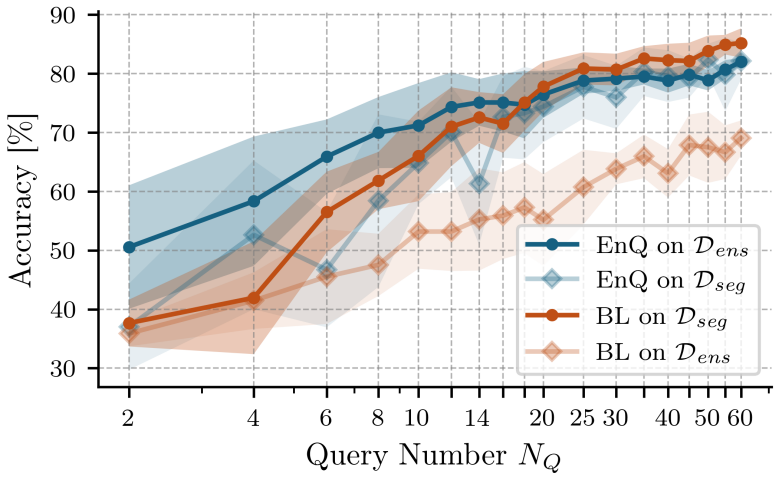


Figure 6.4: Reward model test accuracy for our EnQ approach and the baseline of segment-based uniform sampling [57] for different numbers of queries on their native dataset (e.g., EnQ on $\mathcal{D}_{\text{ens}}$) and in cross validation (e.g., EnQ on $\mathcal{D}_{\text{seg}}$) The process of querying, reward model training, and testing has been repeated ten times, for which mean and standard deviation are shown. We outperform the baseline with a higher test accuracy and thus information gain for low-query numbers, enabling a faster learning curve in time-critical learning scenarios.

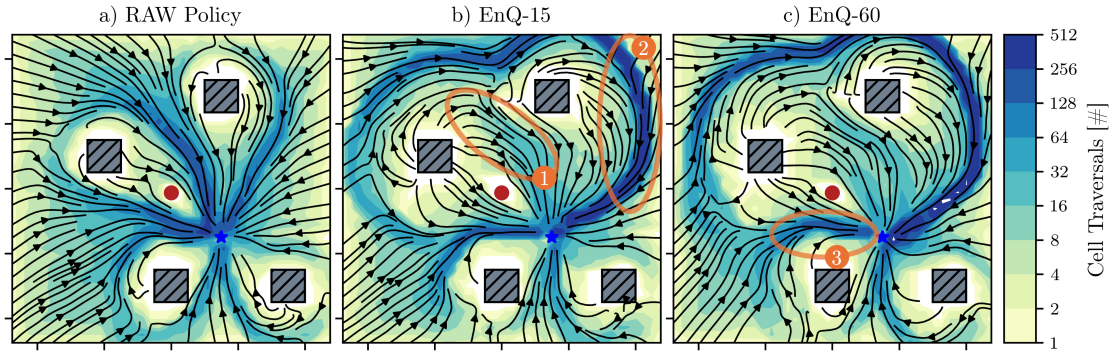


Figure 6.5: Driving behavior for a given scene visualized by our novel explainability navigation plot, compare Section 6.4.6 for **a)** the raw policy π_{raw}, **b)** the preference-aligned policy EnQ for $N_Q = 15$ queries, and **c)** for $N_Q = 60$ queries. The trajectory flow can be derived from any start position in the given scene to the goal (blue star), while circumnavigating the human (red dot). Regions of interest (ROI) are indicated in orange. Under the raw policy, mostly goal-directed and collision-avoiding navigation behavior can be observed. For the aligned policies, a pronounced shift away from the human at the cost of longer trajectories appears, e.g., on the far side of the top right obstacle (ROI 2). At the same time, in terms of traversal the area around the human is thinned out (ROI 1), as indicated by the underlying traversal map. EnQ-60 traverses closer to the human in the direct vicinity (ROI 3).

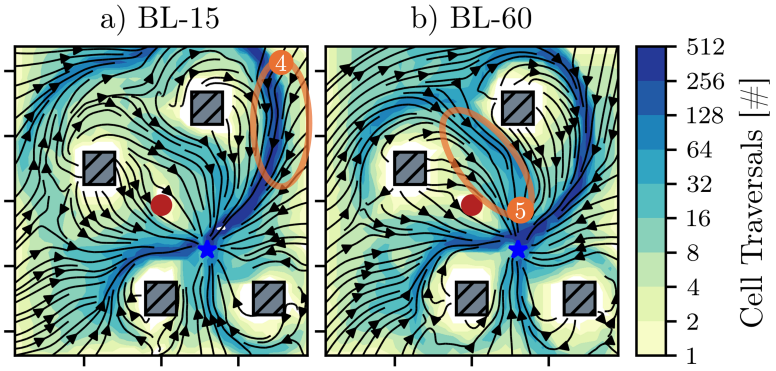


Figure 6.6: Driving behavior for a given scene visualized by our novel explainability navigation plot, compare Section 6.4.6 for the baseline segmentation-based approach with **a)** $N_Q = 15$ and **b)** $N_Q = 60$ queries. Compared to EnQ (see Figure 6.5), the developed outer traversal corridor on the right falls closer to the human (ROI 4). Furthermore on BL-60, the corridor directly above the human (ROI 5) is traversed more often as compared to EnQ-60, indicating less distance-keeping from the human.

accuracy. Notably, the segment-based reward model does not generalize well to the ensemble data, while the ensemble-based reward model generalized well to the segment dataset $\mathcal{D}_{\text{seg}}$. We can conclude that EnQ provides an advantage in information gain feedback processes that need to be query efficient.

6.5.3 Policy Alignment

The following experiments target the final navigation performance of the preference-aligned policy π_{aligned} that should keep a high distance from the human. The aligned policies will be denoted as EnQ for our ensemble-based approach EnQuery and BL for the baseline. Those model names are complemented by the number of queries N_Q used to align the model. We chose $N_Q = 15$ as a low-query and $N_Q = 60$ as a high query number, where the EnQ reward model outperforms the BL in the low-query, and vice

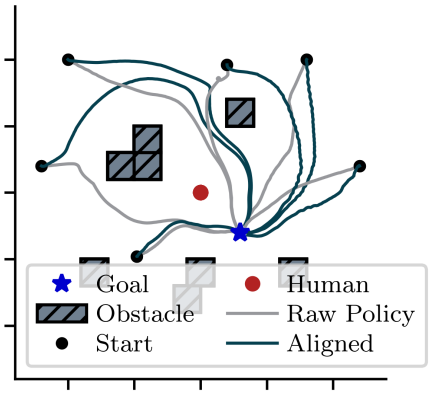


Figure 6.7: Trajectories of the raw and aligned policies π_{raw} and π_{aligned} for a given obstacle configuration and randomized start position. While the raw policy exhibits mostly goal-directed navigation behavior, π_{aligned} reflects the preference of maximum human distance keeping on a majority of the trajectories.

versa for the high query regime, compare Section 6.5.2.

6.5.3.1 Quantitative

The ablation study presented in Table 6.2 quantitatively evaluates the performance of different configurations of a navigation system across several metrics: success rate (SR), collision rate (CR), timeout rate (TR), and minimum human distance ($\min(d_h)$) as our metrics. The results were averaged over 10 aligned policies with different query samples and 100 trajectory rollouts with identical scene setups. For the preference metric $\min(d_h)$, we test for statistical significance of EnQ-15 exhibiting a higher $\min(d_h)$ as compared to the other configurations using a student's t-test, as denoted in the last column of Table 6.2. We compare the aligned models against the RAW policy π_{raw}, specifically in terms of distance from the human in alignment. The ensemble-query aligned model with only 15 queries denoted as EnQ-15 significantly outperforms all other configurations in terms of the highest minimum human distance. EnQ-60 exhibits a higher success rate compared to EnQ-15, but at the cost of weaker distance keeping. The baseline policy BL improves with respect to $\min(d_h)$ as the number of queries increases from 15 to 60. In this regime, our approach EnQ-15 achieves the lowest collision rate at the cost of more timeouts as compared to BL-15. Logically, timeouts become more likely as the robot drives longer, more human-distant trajectories. BL-15 exhibits the highest success rate, but at a lower preference metric compared to EnQ-15. To conclude, our approach outperforms the baseline with respect to the preference metric for the low-query regimes, reflecting the results of the improved reward model below $N_Q = 18$ queries, see Figure 6.4.

	$\min(d_h)$ [m]	SR↑	CR↓	TR↓ [%]	p
EnQ-15	**3.2**	89.8	**3.4**	6.8	-
EnQ-60	2.1	92.4	3.9	3.7	**
BL-15	2.5	**93.9**	3.7	**2.4**	***
BL-60	2.7	86.8	3.6	9.6	***
RAW	1.2	94	6	0	***

Table 6.2: Quantitative analysis of the performance and minimum human distance $\min(d_h)$ averaged over 10 alignment runs and 100 trajectory rollouts with identical scene setups. EnQ-15 denotes the ensemble-query aligned model with $N_Q = 15$, EnQ-60 with $N_Q = 60$ queries. Analogously, BL-15 and BL-60 denote the segment-based baseline querying approach with uniform sampling [57]. RAW denotes the non-aligned initial policy, averaged over 100 trajectory rollouts. SR, CR, and TR denote the success, collision, and timeout rate, respectively. Statistical significance of $\min(d_h)$ for dissimilar distribution means against EnQ-15 is denoted in the last column p, where * for $p \leq 0.05$, ** for $p \leq 0.01$, *** for $p \leq 0.001$. EnQ-15 as our flagship approach exhibits the best alignment in terms of the preference metric, while running into more timeouts due to possibly longer trajectories.

6.5.3.2 Qualitative

We compare the aligned policy EnQ-15 against π_{raw} (RAW) and EnQ-60 qualitatively for a given navigation scenarios, see Figure 6.5 with indicated regions of interest (ROI). With regard to the baseline reward, both policies exhibit obstacle-avoidance and goal pursuit behavior. For the aligned policy, however, the driving patterns bend away from the human, as compared to goal-directed driving directions with π_{raw}. With respect to the human this manifests, e.g., in an accumulation of trajectories on the far side around the top-right obstacle (ROI 2), allowing for more distance from the human. Directly around the human, a thinning of robot's passages can be observed (ROI 1), with less trajectories approaching from the top-left of the plot alongside the human. Furthermore, a noticeable outward bend of trajectories around the human (ROI 1) arises for the aligned agent. Subtle but noticeable, EnQ-60 traverses closer to the human in the direct vicinity (ROI 3).

Comparing against the baseline approach BL-15 and BL-60 in Figure 6.6, two findings strike. The developed outer traversal corridor around the top-right obstacle falls closer to the human (ROI 4) as compared to EnQ. Furthermore on BL-60, the corridor directly above the human (ROI 5) is traversed more often as compared to EnQ-60 and BL-15, indicating less distance-keeping from the human.

A direct comparison of trajectories between π_{aligned} and π_{raw} is visualized in Figure 6.7. Here, a similar picture manifests with the aligned trajectories traversing at a higher human distance, as compared to the goal-directed trajectories of the raw policy.

6.6 Conclusion

This chapter introduces EnQuery, a novel ensemble-based query method for diverse behavior suggestions in reinforcement learning from human feedback (RLHF) with deterministic policies. We apply EnQuery to the field of robot navigation where the robot operates in the vicinity of humans who have specific preferences regarding the robot's navigation style. Using our output diversity regularization during training, the ensemble generates diverse trajectories that can be used to query preferences for any given navigation task. Importantly, generated queries maintain consistent reference points, such as the start and goal positions and use the same environment setup which improves the retest reliability and thus the information extracted from the preference pairs. The experiments show a superior information gain for low-query numbers compared to a widely used baseline querying approach. We then successfully demonstrate the data-efficient preference alignment of a navigation policy by recycling the collected experience data. Finally, our novel method for visualizing navigation policy behavior comprehensively illustrates the alignment result. As a core contribution to RQ1, EnQuery aligns with our broader vision of developing more intuitive, efficient, and human-centric approaches to customize robotic navigation behaviors. Beyond its algorithmic contribution, EnQuery also offers a practical tool to support scalable user studies by generating

meaningful preference queries with common spatial reference points in a behaviorally diverse manner.

While this chapter presented the development and evaluation of EnQuery from a technical perspective, the next chapter adopts a more user-centered focus. Here, EnQuery serves as the query generation backend in a user study designed to investigate an underexplored factor of the impact of interface modality on user preference expression in RHLF setups. Specifically, we examine how immersive VR compared to conventional 2D video interfaces influences the consistency and quality of collected preferences, and in turn, the effectiveness of the resulting preference-aligned navigation policies.

7 The Impact of VR and 2D Interfaces on Human Feedback in Preference-Based Robot Learning

Abstract

Aligning robot navigation with human preferences is essential for ensuring comfortable and predictable robot movement in shared spaces, facilitating seamless human-robot coexistence. While preference-based learning methods, such as reinforcement learning from human feedback (RLHF), enable this alignment, the choice of the preference collection interface may influence the process. Traditional 2D interfaces provide structured views but lack spatial depth, whereas immersive VR offers richer perception, potentially affecting preference articulation. This chapter systematically examines how the interface modality impacts human preference collection and navigation policy alignment. We introduce a novel dataset of 2,325 human preference queries collected through both VR and 2D interfaces, revealing significant differences in user experience, preference consistency, and policy outcomes. Our findings highlight the trade-offs between immersion, perception, and preference reliability, emphasizing the importance of interface selection in preference-based robot learning.

7.1 Introduction

In the previous chapter, we introduced a method for efficient query generation in human-centric robot navigation, designed to enhance the efficiency of preference collection in reinforcement learning from human feedback (RLHF) settings. This directly supports our overarching objective of aligning robot behavior with human preferences in shared environments (cf. Chapter 1.2), thereby improving user comfort, personalization, and the overall quality of human-robot interaction. Recent advances in preference-based learning, including reinforcement learning from human feedback (RLHF) [205], demonstrate the potential of human-in-the-loop methods to shape robot behavior in alignment with user expectations. In fact, preferences have been leveraged in robot learning across various settings, including multi-task learning [206], collaborative tasks [207], language-based tasks [191], [208], and social navigation [102].

A key challenge in preference-based robot learning is the method of preference elicitation. Various interfaces have been explored to facilitate this process, with conventional 2D visualizations such as first-person and bird's-eye-view videos being widely employed. While these 2D interfaces provide accessible and structured representations of navigation scenarios, they lack the depth and spatial context necessary for nuanced human judgment, particularly in complex 3D environments. More immersive alterna-

This chapter is a revised and updated version of the peer-reviewed publication [79]. Refer to Section 1.4 for details.

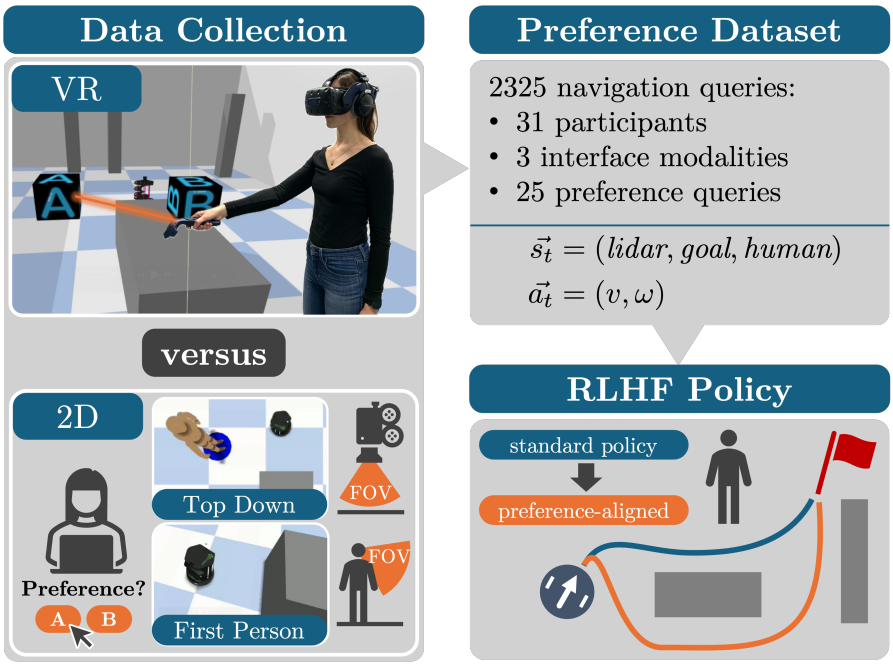


Figure 7.1: **Left:** Our study collects preferences via Virtual Reality (VR) and 2D interfaces (top-down and first-person views), enabling a systematic comparison of interface modalities for eliciting human preferences on robot navigation. **Top-right:** The preference dataset consists of 2,325 navigation queries from 31 participants. **Bottom-right:** Using Reinforcement Learning from Human Feedback (RLHF) with our dataset, we refine a standard navigation policy (blue) into preference-aligned policies (orange), enhancing human-centric navigation.

tives, such as virtual reality (VR), offer a richer perceptual experience, potentially improving preference collection by providing a more realistic sense of robot motion and environmental context [209], [210].

Despite the potential advantages of VR, systematic comparisons between immersive and traditional 2D video-based interfaces for preference collection remain sparse. Previous studies have primarily focused on individual interface performance or user engagement without thoroughly investigating how the chosen interface modality affects preference data quality and the subsequent alignment of robot navigation policies. In line with this thesis' RQ1 a critical question arises: Do user preferences for robot behavior differ between immersive VR and traditional 2D interfaces, and if so, how do these differences influence preference-aligned navigation policies?

In this chapter, we introduce a novel user preference dataset for robot navigation, collected through both VR and 2D interfaces. Our study systematically evaluates the impact of interface modality on the user experience and preference collection process. We compare preferences elicited using a VR interface against those obtained via first-person and bird's-eye/top-down view video interfaces. Finally, we derive preference-aligned navigation policies from the collected preference data, taking into account the interface modality used for collection. Our primary contributions are:

- The collection of a dataset capturing user preferences on robot navigation across

VR and 2D interfaces.

- A quantitative and qualitative evaluation of how the different interface modalities influence the collection of user preferences.
- The derivation and analysis of three preference-aligned navigation policies from modality-specific user preferences.

7.2 Related Work

7.2.1 Preference Learning in Robotics

Integrating human preferences into robotic systems has gained significant attention in recent years, particularly in human-robot interaction and autonomous navigation. Preference-based reinforcement learning (PbRL) [24], [57], [58], [188] is a key approach that enables robots to align their behaviors with human expectations by iteratively refining a policy [211] through preference feedback rather than explicit reward functions. Wang *et al.* [212] proposed a preference-based action representation learning (PbARL) approach that efficiently fine-tunes pre-trained policies to human preferences, allowing for effective personalization in robot behavior without requiring extensive retraining. Similarly, Palan *et al.* [25] introduced a hybrid learning framework that combines expert demonstrations with preference queries to improve the efficiency of reward function learning, mitigating the limitations of inverse reinforcement learning and standard PbRL methods. The effectiveness of learning is also contingent on the quality of queries presented to humans. To enhance query informativeness, various PbRL active learning techniques have been developed, leveraging policy ensembles (cf. Chapter 6) or unsupervised learning [68], [189].

Beyond reinforcement learning, Bacchin *et al.* [19] introduced a people-aware navigation system for telepresence robots that fuses remote operator commands with a probabilistic model of human-robot interaction. Their system dynamically adjusts the robot's behavior based on inferred social signals, demonstrating how preference-aware navigation can enhance both user satisfaction and social compatibility. In another study, Zhou *et al.* [16] explored how human preferences can guide the improvement of inappropriate robot behaviors. Their findings highlight the importance of capturing nuanced human feedback to refine robot motion strategies for social navigation.

Recent works have also examined the role of virtual environments in preference learning. In Chapter 3 we presented a VR-based demonstration interface for learning personalized robot navigation policies, emphasizing the benefits of VR in capturing human motion preferences for dynamic environments. This approach highlights the utility of immersive settings in enabling intuitive and expressive demonstrations by users.

7.2.2 Interfaces in HRI

The design of effective query interfaces for human-in-the-loop preference learning plays a crucial role in shaping the quality and reliability of collected preference data. Traditional preference elicitation methods often rely on 2D graphical user interfaces (GUIs), such as first-person and bird's-eye view perspectives, which have been widely used for interactive reinforcement learning [213] and human-robot collaboration [69], [214]. However, recent advancements in immersive technologies, including VR and mixed reality, have introduced novel interaction paradigms that promise more natural and context-rich preference acquisition [209].

Wonsick and Padır [65] classify VR interfaces for robot control into five areas: visualization, control, interaction, usability, and infrastructure. Their comparison of VR and traditional keyboard-mouse-monitor setups (KBM) for humanoid teleoperation shows that VR enhances engagement, intuitive control, and spatial awareness while reducing cognitive load, whereas KBM benefits from widely available hardware.

LeMasurier *et al.* [215] further investigated the trade-offs between 2D and VR interfaces for human-in-the-loop robot planning in navigation and manipulation tasks. Their study finds that while KBM interfaces yield higher task performance, VR interfaces lead to fewer collisions, making them preferable for high-risk scenarios where safety is paramount.

Wozniak *et al.* [66] explored the effectiveness of VR interfaces for correcting robot perception errors, comparing them with traditional screen-based interfaces. Their study found the VR interface to be more immersive and enjoyable for the users, who preferred it over the screen-based alternative.

These studies underscore the growing relevance of immersive interfaces for robotics. Our study therefore investigates the differences between VR and 2D KBM interfaces for user preference acquisition.

7.3 Method

We subsequently provide an overview of the navigation task of the robot, the interface, and the user study setup for data collection.

7.3.1 Problem Statement

This work investigates how different user interfaces influence the user preference collection for learning-based robot behavior adaptation. We focus on a query interface where users provide pairwise comparisons of pre-recorded robot navigation trajectories, a key component of preference-based reinforcement learning (PbRL). We analyze the impact of interface modalities (VR vs. 2D GUI) and scene perspective on user experience and preference expression. As an application scenario, we consider a human-aware robot navigation task in which a robot navigates to a goal in an environment with static obstacles and a nearby human. The human may have specific preferences regarding the

robot's navigation behavior.

7.3.2 Learning Robot Navigation

In line with our target methodology, PbRL, the navigation task is solved via reinforcement learning, where an agent learns to navigate the robot to the goal using velocity commands. As a simulation environment, iGibson [63] with its PyBullet physics engine is used to simulate a Kobuki TurtleBot 2i. The navigation scene resembles an open space with position-randomized small and large box obstacles and a static human. Start, goal and human position are sampled in close proximity to each other. The core navigation reward

$$r^t = r^t_{\text{goal}} + r^t_{\text{time}} + r^t_{\text{collision}} + r^t_{\text{timeout}} + r^t_{\text{loop}} + r^t_{\text{jerk}} \tag{7.1}$$

contains a sparse goal reward of $r^t_{\text{goal}} = +20$, a continuous time penalty of $r^t_{\text{time}} = -0.001$, sparse collision ($r^t_{\text{collision}} = -20$) and timeout penalties ($r^t_{\text{timeout}} = -1.0$), a sparse penalty upon self-intersection of the trajectory $r^t_{\text{loop}} = -2$ and a jerk penalty $r^t_{\text{jerk}} = -w_{\text{jerk}}\|j^t\|^2 / \max_{\tau \leq t} \|j^t\|^2$, with $j^t = (a^t - 2a^{t-1} + a^{t-2})f^2$, the weight $w_{\text{jerk}} = 0.0005$, the action a^t at time t and f the control frequency of $5\,\text{Hz}$. Here, the term sparse indicates that the reward term only take the stated value when their condition is fulfilled and are zero otherwise, respectively. Episodes end upon reaching the goal, a collision, or a timeout.

7.3.3 Query Generation

To generate queries of a navigating robot for user evaluation, we use EnQuery (cf. Chapter 6), an ensemble-based query generation method designed to improve the efficiency and reliability of user preference collection in PbRL. EnQuery is particularly suited for applications where behavior diversity is required under consistent environmental conditions, such as in a given navigation scenario.

Following Chapter 6, we employ an ensemble $\mathcal{E} = \pi_i(s_t, a_t) \mid i \in [N_E]$ of $N_E = 4$ policies, referred to as the policy ensemble of TD3 [103] reinforcement learning (RL) policies. These policies are trained with a regularization term that promotes behavioral diversity among ensemble members. Thus, we obtain two distinct 2D trajectories connecting the same start and goal. This approach ensures that all generated trajectory options are grounded in a common reference frame, which aims to improve retest reliability by reducing variations in extraneous environmental factors.

Once trained, the ensemble $\mathcal{E}$ is used to generate diverse trajectory options for the randomized scene configuration, by sampling two individual ensemble policies and rolling them out in the sampled navigation scenario. To ensure meaningful queries, trajectories that result in collisions or self-intersections are filtered out. We generate a dataset D_Q of $N = 500$ queries, from which we sample subsets for the participants to rate in the user study, either in VR or as 2D video playback.

7.3.4 Query Interfaces

To collect user preferences, we employ three types of query interfaces: an immersive virtual reality (VR) setup and two 2D video-based KBM interfaces on a desktop computer. Both interfaces show the same navigation environment but differ in perspective and immersion. The VR interface provides an interactive and immersive experience, whereas the 2D video KBM interface offers a more conventional, screen-based alternative. Within the 2D interface, we present two distinct perspectives: Top-down (2D-TD), as in [69], also known as a bird's-eye view, and first-person view (2D-FPV), which more closely resembles a VR perspective. For a given query, the human position is defined, serving additionally as the observer position in VR and 2D-FPV.

7.3.4.1 Virtual Reality

The VR interface is based on PyBullet VR [107], ensuring native compatibility with the EnQuery training environments. We connect an HTC Vive Pro Eye setup as VR hardware. A transparent blue cylinder indicates the static human observer position on the floor. Additionally, a floating dialogue in front of the user conveys instructions and announces the upcoming trajectory with labels (A or B) for $2\,\mathrm{s}$. For preference selection, participants interact with a floating selection menu by pointing and clicking on one of two labeled boxes, A or B.

7.3.4.2 2D Video

In contrast, the 2D video KBM interface is implemented using the open-source library Pygame [216] on a desktop computer. The full-screen interface contains a video frame with clickable buttons positioned to the side for starting queries and selecting preferences. To maintain consistency in visualization, all query videos are recorded from two perspectives (2D-TD and 2D-FPV) at a resolution of 720×404 pixels.

Each video begins with a $2\,\mathrm{s}$ trajectory label (A or B), ensuring clear differentiation between the two options. While the 2D-FPV video is recorded with a $60°$ vertical field of view that tracks the robot from the perspective of the human, the 2D-TD perspective is set to capture the entire navigation path. Additionally, in 2D-TD, the human is represented by a neutral wooden mannequin 3D model.

7.3.4.3 Trial

The robot is initialized at the query-specific start position for both trajectory options. Once the participant initiates the trial by clicking a button, the pre-recorded trajectory or video plays, and the robot navigates through the scene. The start and goal positions are not visualized. Each query consists of two trajectories presented sequentially. To prevent bias or premature selection, the preference selection menu is disabled until both trajectories have been displayed. Then, the user can select their preferred trajectory.

Queries cannot be repeated, and the user must make a selection before proceeding to the next query.

7.3.5 User Study

We conducted a user study to compile a dataset of participants' navigation preferences and assess user experience across three preference interfaces (VR, 2D-TD, 2D-FPV).[7] Data collection was divided into three distinct stages: S1: Collecting preferences through navigation queries for each interface modality, S2: Post-interface interaction questionnaires assessing user experience, and S3: A final ranking survey comparing the three interfaces. Before testing, all participants received detailed study information, provided written consent, and completed a demographic questionnaire. The study was structured into three blocks in randomized order, each corresponding to one of the interfaces as an experimental condition. Each block presented the same 25 preference queries in random order (S1), initially sampled for each participant from the query dataset D_Q. By presenting identical queries across different interfaces, we could later investigate the impact of the interface modality on participants' navigation preferences. After each interface block, participants completed a questionnaire (S2) regarding their experience with the interface and the queries, as shown in 7.1, questions Q1–Q10. Upon completion of all three blocks, the final ranking survey (S3), based on the Technology Acceptance Model (TAM) [217], asked participants to rank the interfaces by perceived usefulness, intention to use, and ease of use, as shown in 7.1, Questions R1–R3. All collected data was anonymized using a coding table for participant IDs. Each session lasted approximately one hour.

7.3.6 Participants

A total of 32 individuals (10 women, 22 men) participated in the study in exchange for a EUR 15 monetary compensation. All participants reported having corrected-to-normal vision. One participant was removed due to technical issues during data collection, leaving $N = 31$ participants (10 women, 21 men). The mean age of the sample was 24.6 years (SD = 3.7). Participants rated their experience with AR/VR on a 7-point Likert scale, with a mean rating of 3.1 (SD = 1.5). Participants also rated their experience with robotics on the same scale, yielding a mean score of 3.6 (SD = 2.0). The study adhered to the principles outlined in the Helsinki Declaration.

7.4 Experimental Evaluation

We propose the following hypotheses that we aim to evaluate with our study: (H1) The user experience differs between the interface modalities. (H2) The user preferences for robot navigation differ between the interface modalities. (H3) A preference discrepancy

[7] The dataset is publicly available and linked in the supplemental material in the appendix.

between interfaces reflects in the navigation behavior of interface-specific preference-aligned policies.

7.4.1 Interface Questionnaire

Targeting the users' interface experience and the expression of preferences, we analyze the 10-item questionnaire (Likert scale, score 1-7) of S2, see Figure 7.2. We used a Friedman test to statistically evaluate whether the interface modality (VR, 2D-TD, 2D-FPV) had a significant impact on the ratings (H1), in each of the 10 questions. Note that we chose a non-parametric alternative to the repeated-measures ANOVA to account for the ordinal scale level of the responses (7-point Likert scale). We followed up with three pairwise Wilcoxon signed-rank comparisons with Bonferroni correction when the Friedman test revealed a significant impact of the interface modality. Supporting H1, statistically significant differences in favor of the VR interface were found for the ease of expressing preferences compared to 2D-FPV (Q1), participants' confidence in evaluation compared to both 2D interfaces (Q2), the naturalness of providing preferences (Q5), a clearer spatial understanding compared to 2D-FPV (Q6). We included Q8 from a validated presence scale [218], confirming higher immersion levels in VR compared to both 2D interfaces. Participants reported that the VR interface was significantly more fun to use and less boring (Q9, Q10, System Usability Scale [219]) compared to the 2D interfaces, which further supports H1. No significant effects were observed for the remaining questions.

7.4.2 Interface Ranking

After the three preference query blocks, participants ranked the interfaces (S3) based on the TAM (see Figure 7.3 and Table 7.1). The forced-choice ranking did not allow ties. A chi-square test assessed deviations from equal choice distributions across VR, 2D-TD, and 2D-FPV. Significant effects were further examined using pairwise z-tests with Bonferroni correction.

For **Usefulness**, rankings were not evenly distributed ($p < .001$), with VR being more often preferred over both alternatives, and 2D-TD preferred more often over 2D-FPV. VR was ranked first by $81.2\,\%$, while 2D-FPV was consistently last.

For **Ease of Use**, no significant differences were found, with rankings evenly distributed across interfaces.

For **Intention to Use**, VR was preferred more often ($p < .001$), significantly outranking both alternatives, while 2D-TD was preferred more often over 2D-FPV. Notably, $90.6\,\%$ ranked VR first, and 2D-FPV received no first-choice votes.

Overall, VR was significantly more often preferred for usefulness and intention to use, aligning with [66], likely due to enhanced perception of robot behavior. Increased fun (Q9, Figure 7.2) may explain why the majority of participants expressed their intention to use VR. No interface stood out for ease of use, possibly because the preference task itself was similar to operate between interfaces. Across all criteria, 2D-FPV was the

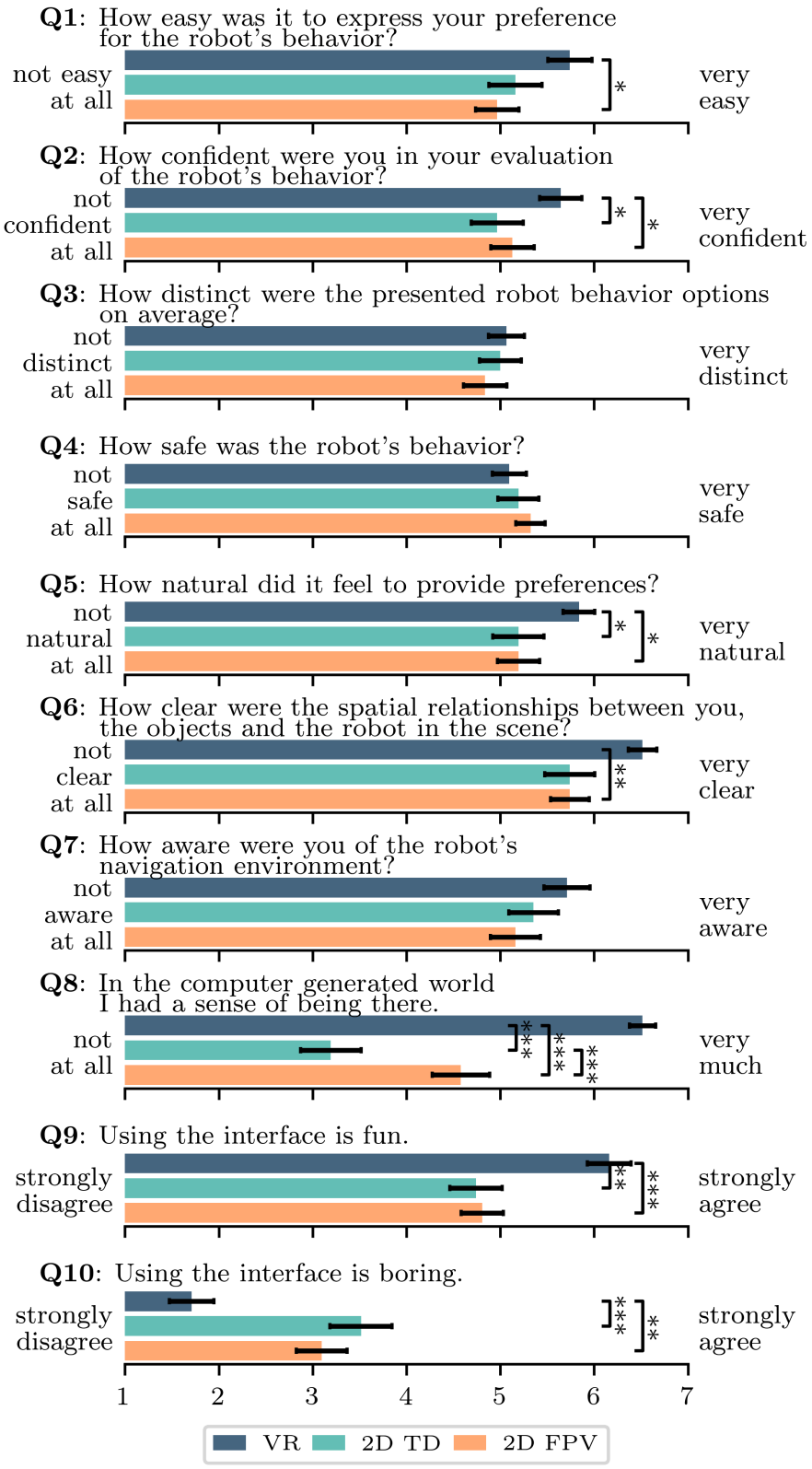


Figure 7.2: Survey results (S2) comparing user experiences across three interface conditions: virtual reality (VR), 2D top-down (2D-TD), and 2D first-person view (2D-FPV). Participants rated their experience across multiple aspects after each block. Ratings were provided on a Likert scale (1-7), bars indicate score means, standard errors are indicated. Asterisks denote significance levels (* $p < .05$, ** $p < .01$, *** $p < .001$).

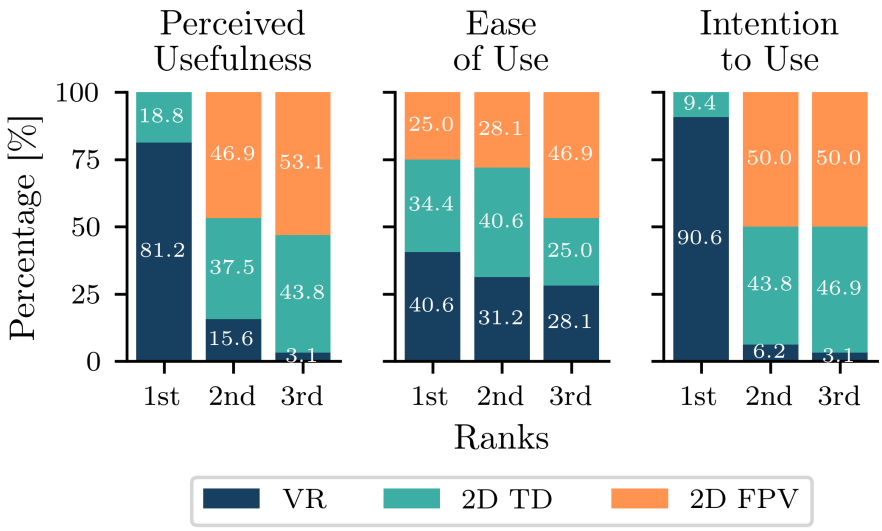


Figure 7.3: User rankings (S3) of three modalities, namely Virtual Reality (VR), 2D Top-Down (2D-TD), and 2D First-Person View (2D-FPV), based on perceived usefulness, ease of use, and intention to use. Each bar represents the percentage of participants who assigned first, second, and third ranks to each modality. VR is predominantly ranked highest in usefulness and intention to use.

least often preferred.

7.4.3 Dataset Overview

The dataset contains 2,325 user preference queries for robot navigation, collected across three interface modalities: VR, 2D-TD, and 2D-FPV. The dataset's structure follows the data collection design: Participant → interface modality → query → preference label.

In more detail, each query stores both trajectories and the scene configuration (e.g., for replay in VR), the 2D videos, and RL episodes (state, action, next state, reward) for both trajectories, where states $\mathbf{s}_t = (\text{lidar}, \text{goal}, \text{human})$ capture lidar data, robot-centric goal and human position, while actions $\mathbf{a}_t = (v, \omega)$ represent velocity commands. Preferences are stored as A/B labels. In addition to the state-action pairs, we provide the robot trajectory as a 2D path, the static human pose, and the obstacle poses in world coordinates.

As we later show, our dataset enables the distillation of preference models, e.g., for the preference alignment of RL robot navigation policies.

	Ranking Instruction (Options: 1st, 2nd, 3rd)
R1	Order the three interfaces for their usefulness for rating the robot behavior.
R2	Order the three interfaces for their ease of use.
R3	Order the three interfaces based on your intention to use.

Table 7.1: Ranking instructions (S3) for the interfaces upon completion of all three interface modality blocks.

Modality Pair	Agreement [%]	Standard Deviation [%]
VR - 2D-TD	69.2	9.6
VR - 2D-FPV	68.6	11.8
2D-TD - 2D-FPV	67.0	14.2

Table 7.2: Mean agreement and standard deviation between different interface modalities, aggregated on a per-participant basis. Note that the block order of interfaces has been randomized among participants. Preference changes between interfaces occur but not significantly more often for specific interface combinations.

7.4.4 User Preferences

This section deals with a quantitative analysis of the preference dataset collected in S1 with respect to the effect of the interface modality.

7.4.4.1 Modality Agreement

Querying the same 25 trajectory pairs in all three interface modalities to each participant allows us to examine whether participants exhibit different preferences between the interfaces. We compute the interface modality agreement by matching the block-randomized queries between modalities and checking for preference agreement, see Table 7.2. The agreement is aggregated on a per-participant basis and subsequently averaged over all participants. With the values of all three combinations averaging around $70\,\%$, we can conclude that preferences do change between interfaces, but not noticeably more often between specific interface combinations. We conclude that consistency in the interface is key during dataset collection, as preferences can be inconsistent with an interface change. As reflected by the standard deviation, we observe considerable variation in interface agreement among participants. This finding underscores the necessity for interface consistency when preference data is collected or merged.

7.4.4.2 Disagreement Analysis

We now transition from inter-modality agreement to the cases of disagreement. When participants preferred trajectory A in one modality but trajectory B for the same AB-query in a different modality, we term this inter-modality disagreement. For each interface combination of these inter-modality disagreements, we explore the differences between the two preferred trajectories. Note that we report these differences but refrain from conducting inferential statistical tests because the characteristics of the two trajectories in the same query were experimentally controlled. Figure 7.4 shows the differences in preference for selected trajectory metrics, while the x-axis indicates the interface transition. Because different participants had their own sampled subset of 25 queries with a different trajectory profile distribution, we first apply z-score normalization based on all trajectories (preferred and rejected) shown to a participant. Subsequently, the average differences were aggregated on a per-participant basis to account for the varying number of disagreements per participant. We measure changes in the trajectory length (a),

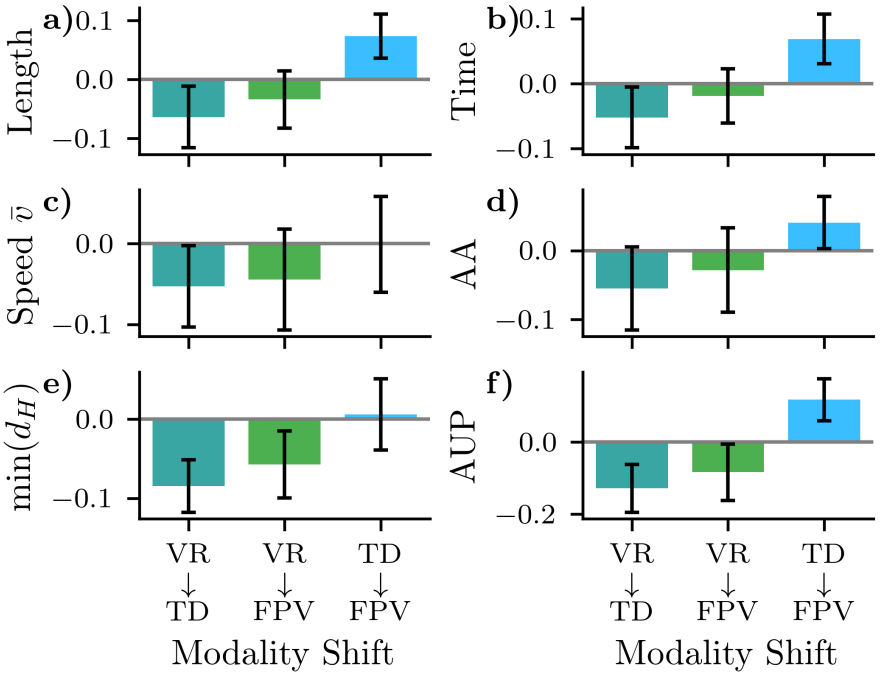


Figure 7.4: Change in the preferred trajectory with a modality shift in cases of preference disagreement between interfaces for a given participant. Metrics are z-standardized for all queried trajectories per participant. Bars show mean change, and error bars indicate the standard error of the participant means averaged over their disagreements. Participants preferred shorter and more straightforward driving trajectories in 2D interfaces compared to VR, with the robot occasionally traversing closer to the human.

time (b), average speed (c), curvature/accumulated angle (AA) (d), minimum distance from the human (min(d_H)) (e), and area under the path (AUP) (f). The sign of the metric change corresponds to the interface transition $A \rightarrow B$. For readability, only one direction is shown. The reverse transition has the same magnitude with an inverted sign.

In cases where participants exhibited different preferences for the same queries in VR and on 2D interfaces, participants preferred shorter and more straightforward driving trajectories (measured by the area under the path) in 2D compared to VR. Additionally, the robot may traverse closer to the human, compare (e). Similarly, query disagreements point to a preference for more straightforward driving styles when transitioning from the 2D-FPV to the 2D-TD interface, see (f).

7.4.5 Policy Alignment

To examine how preferences collected from different interfaces impact policy alignment and navigation behavior, we employ a preference-based reinforcement learning (PbRL) approach. This method trains reward functions from the collected preference dataset, enabling the subsequent alignment of human-aware RL navigation policies for each query interface. Following [57], [58], we learn a parametric reward function $\hat{r}_\psi$ from human preferences. For policy optimization, we implement a PbRL algorithm

following [57], using TD3 [103] as the base RL algorithm. Splitting up our preference dataset by interface, we train three individual reward functions modeled as an MLP with $[256, 256, 256]$ hidden units based on the queries collected from the participants. The policies for each condition were then trained for 500k time steps by weighting

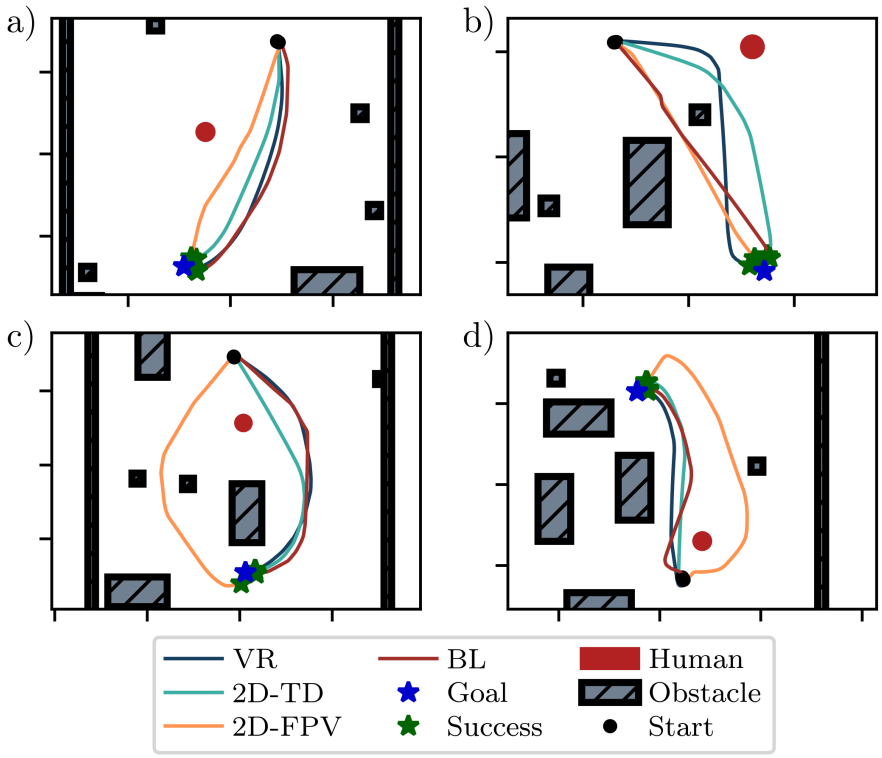


Figure 7.5: Navigation behavior comparison between aligned policies π_{VR}, $\pi_{\text{2D-TD}}$, and $\pi_{\text{2D-FPV}}$ and a non-aligned baseline counterpart π_{BL} in four navigation scenarios. The aligned policies exhibit smoother and more obstacle-aware trajectories than the non-aligned policy π_{BL}, with π_{VR} and $\pi_{\text{2D-TD}}$ demonstrating the best balance between efficiency and safety.

Metric	VR	2D-TD	2D-FPV	BL
CR [%]	5.8	5.8	4.9	5.2
SR [%]	94.2	94.1	**94.9**	94.8
TR [%]	0.0	0.1	0.2	0.0
Steps	49.1	49.5	55.9	50.0
Path Length [m]	4.9	4.9	**5.4**	4.9
Time [s]	**9.6**	9.7	11.0	9.8
AA [rad]	3.2	3.1	**4.1**	2.8
AUP [m²]	2.2	1.8	**3.4**	2.1
$\min(d_h)$ [m]	**1.33**	1.20	1.28	1.25
Speed $\bar{v}$ [m/s]	0.51	0.50	0.49	0.51

Table 7.3: Quantitative metrics for our preference-aligned π_{VR}, $\pi_{\text{2D-TD}}$, and $\pi_{\text{2D-FPV}}$ and a non-aligned baseline π_{BL}, averaged over 1,000 trajectories in randomly sampled scene configurations.

the original navigation task reward and their respective preference reward model as $r = \lambda \hat{r}_\psi + (1-\lambda) r_{\text{core}}$, with $\lambda = 0.2$. The value of λ was empirically determined through iterative experimentation to ensure training stability. Note that the preference reward model ranges from -1 to +1, while the original task reward ranges from a minimum of -10 to a maximum of +20, with these extrema driven by the sparse terms in the function. In addition to the trained policies π_{VR}, $\pi_{\text{2D-TD}}$, and $\pi_{\text{2D-FPV}}$, we also include a non-aligned baseline policy π_{BL}, trained on the same task without preference-based rewards, for comparison.

Figure 7.5 illustrates navigation trajectories of the policies in four distinct scenarios. The plots illustrate the paths taken by each policy from the start to the goal, navigating around static obstacles and avoiding the human. All policies successfully navigate the scenes, while the aligned policies exhibit smoother and more obstacle-aware trajectories compared to their non-aligned counterpart π_{BL}. In Scenario d), both π_{BL} and $\pi_{\text{2D-FPV}}$ are prone to inefficient routes. $\pi_{\text{2D-FPV}}$ shows the same conservative behavior in Scenario c) as well. Overall, the results indicate that the aligned policies π_{VR} and $\pi_{\text{2D-TD}}$ achieve a superior trade-off between efficiency and safety.

The quantitative results in Table 7.3 confirm that preference-aligned policies improve human-aware navigation, with π_{VR} achieving the best balance between efficiency and safety by maintaining the highest human clearance (1.33 m) while ensuring low travel times. $\pi_{\text{2D-FPV}}$ prioritizes safety, exhibiting the lowest collision rate (4.9%) but at the cost of longer paths (5.4 m) and higher angular accumulation (4.1 rad). $\pi_{\text{2D-TD}}$ exhibits the most straightforward navigation, reflected in the lowest area under path (AUP). These findings, supported by the qualitative analysis, highlight that π_{VR} offers the most balanced navigation performance, while $\pi_{\text{2D-FPV}}$ navigates overly conservatively, and π_{BL} is less cautious. Consequently, we find empirical support for H3.

7.5 Conclusion

Our study systematically examined how interface modality affects human preference collection for robot navigation and the resulting policy alignment. The results confirm that the choice of interface modality significantly impacts how users express preferences, perceive the interaction, and ultimately shape robot navigation behavior. The VR interface provided a more immersive and intuitive experience, leading to greater confidence and ease in preference expression. This aligns with prior findings on the benefits of immersive environments for user engagement [220]. However, the study also revealed that preferences were not entirely consistent between interfaces, with participants favoring shorter, more direct paths in 2D interfaces while exhibiting greater tolerance for curved trajectories with increased human clearance in VR. This suggests that the visualization and spatial representation of the robot's movement influence user preferences, highlighting the importance of maintaining interface consistency during preference collection. The navigation policies trained on interface-specific preferences

demonstrated noticeable differences in robot behavior. Directly contributing to the overarching thesis RQ1 (Section 1.2.1), these findings highlight the necessity of considering interface effects in preference-based reinforcement learning, as user preference shifts due to modality changes can directly impact policy training outcomes. To support further research on these effects and the collected user preferences, the preference dataset is publicly available and linked in the supplemental materials in the appendix.

This chapter is the last to present methodology for preference-reflecting robot navigation. Until this point, we have considered personalization frameworks leveraging VR-based demonstrations, multi-objective reinforcement learning, and RLHF. Without exception, all presented approaches in the preceding chapters are human-aware and based on deep RL-based learning pipelines. However, as robots employing these policies enter real-world scenarios involving frequent user interactions, the black-box nature of their neural network reasoning may negatively affect user acceptance if behavior turns out to be counterintuitive. This issue is reflected in the overarching RQ4 (cf. Section 1.2.4), and the following chapter investigates techniques for effectively communicating robot decision-making processes to non-expert users.

8 Immersive Explainability: Visualizing Robot Navigation Decisions through XAI Semantic Scene Projections in Virtual Reality

Abstract

End-to-end robot policies like those from the previous chapters achieve high performance through neural networks trained via reinforcement learning (RL). Yet, their black-box nature and abstract reasoning pose challenges for human-robot interaction (HRI), because humans may experience difficulty in understanding and predicting the robot's navigation decisions, hindering trust development. In this chapter, we present a virtual reality (VR) interface that visualizes explainable AI (XAI) outputs and the robot's lidar perception to support intuitive interpretation of RL-based navigation behavior. By visually highlighting objects based on their attribution scores, the interface grounds abstract policy explanations in the scene context. This XAI visualization bridges the gap between obscure numerical XAI attribution scores and a human-centric semantic level of explanation. A within-subjects study with 24 participants evaluated the effectiveness of our interface for four visualization conditions combining XAI and lidar. Participants ranked scene objects across navigation scenarios based on their importance to the robot, followed by a questionnaire assessing subjective understanding and predictability. Results show that semantic projection of attributions significantly enhances non-expert users' objective understanding and subjective awareness of robot behavior. In addition, lidar visualization further improves perceived predictability, underscoring the value of integrating XAI and sensor for transparent, trustworthy HRI.

8.1 Introduction

The approaches presented in the preceding chapters rely on high-performance robot policies driven by deep reinforcement learning (RL), enabling robots to navigate complex and human environments with remarkable autonomy. Increasingly, such approaches are designed for human-robot interaction (HRI) settings, yet the decision-making processes behind their policies often remain intransparent to end-users because they depend on neural networks that are effectively black boxes [221], creating barriers to user comprehension and trust. This is further compounded by the "perceptual belief problem" [222] that arises from people's difficulty in understanding what robots know about the shared environment, e.g., due to limited familiarity with robotic sensing capabilities. The lack of understanding impedes robot predictability by the user which can impact user trust, as illustrated in Figure 8.1.

This chapter is a revised and updated version of the peer-reviewed publication [80]. Refer to Section 1.4 for details.

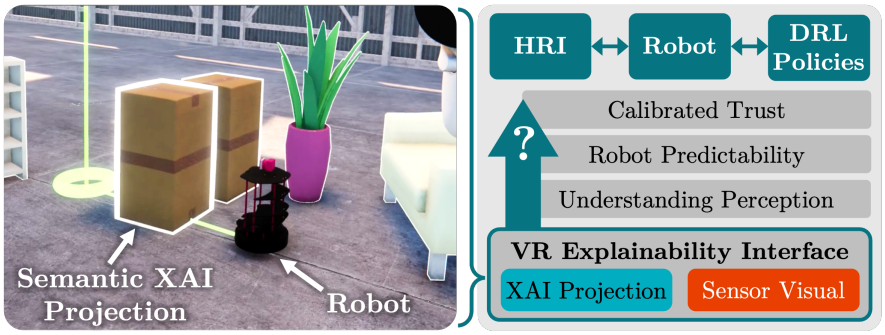


Figure 8.1: Our immersive VR explainability interface communicates XAI attributions and sensor perception of an RL robot navigation policy to non-expert users, by grounding them in the object semantics of the scene. Objects that are important to the policy are highlighted using a glowing outline. A better perception understanding in combination with the user's perceived ability to predict the robot can lead to calibrated trust towards the robot.

Users were found to have a clear interest in robots capable of explaining their navigational decisions [72]. A recent study showed that the effectiveness of XAI methods across different applications varies significantly [223], underlining the need for more user-focused evaluations of how XAI explanations are conveyed. This is especially challenging for the continuous, dynamic decision-making process of a navigating robot. In line with the HRI-transparency facet of overarching RQ4 (cf. Section 1.2.4), the question arises as to how non-expert users can effectively and intuitively understand both the robot's perceptual capabilities and the explanations generated by XAI methods [224], [225].

Therefore, we propose an immersive virtual reality (VR) interface that integrates two key elements for novice users: a clear visualization of a) the robot's sensor data and b) the contextual XAI outputs. We visualize the attribution scores of an RL-based policy by continuously projecting them onto the objects that influence the robot's decision process, visually making them glow based on their inferred importance. Through various navigation task and obstacle configurations, we allow users to gain insights into how the robot perceives its environment and is influenced by different obstacles on its way to the goal. We hypothesize that this approach not only enhances the user comprehension and predictability of robot behavior, but also improves trust in the robot's actions.

The primary contributions of our work are threefold:

- A VR interface that communicates robot perception and navigation policy explanations grounded in scene semantics.
- Extensive assessment of this novel visualization to explain robot navigation decisions in a $N = 24$ user study.
- Empirical demonstration of significantly enhanced user understanding and predictability of the robot, with a potential for enhanced trust calibration.

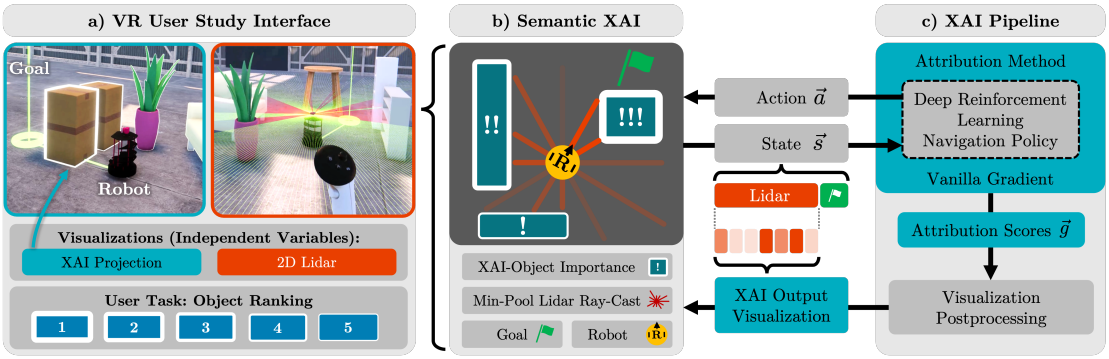


Figure 8.2: Architecture of our XAI-VR interface. **a)** The VR interface visualizes the robot in a navigation scenario, the object-projected XAI attribution scores, and the 2D lidar sensor to the user. In our user study, the visualizations (XAI and lidar) represent the independent variables (IVs), while we measure the users' performance in ranking the robot-surrounding objects according to their importance to the robot, defined by the visualized attribution scores. **b)** Objects are highlighted according to their importance by a white outline of variable thickness, here depicted in a top-down schematic. Their importance is assigned by the ray-casts of the 2D lidar sensor, which project the post-processed attribution scores of the lidar-containing state space into the scene. Specifically, the state space contains a min-pooled set of lidar readings and the robot-centric goal position. **c)** The XAI technique *Vanilla Gradient* generates gradient-based attributions $\vec{g}$ for the RL-trained navigation policy. The lidar-related part of $\vec{g}$ is post-processed for visualization in VR using Equation 8.2.

8.2 Related Work

8.2.1 General and RL-based XAI

Explainable Artificial Intelligence (XAI) has been a well-established research area for several years, leading to the development and deployment of numerous methods [226] across diverse application domains [227]. Core goals of XAI include enhancing transparency, trust, and human understanding of black box behavior [228], with key questions often centered around what information should be conveyed, to whom, and in what form [227], [229]. Much of the existing work in XAI has focused on standard supervised learning tasks such as classification and regression [230]. Clearly distinct from these standard scenarios, Explainable Reinforcement Learning (XRL) has emerged as a specialized subfield aiming to bring interpretability to sequential decision-making systems trained as reinforcement-learning agents [221], [231]. While the overarching goals and evaluation metrics of XRL align closely with those in traditional XAI, e.g., criteria such as fidelity, comprehensibility, and usefulness [231], [232], the temporal nature of the RL setting and potentially complex environments motivate a clear conceptual distinction between XAI and XRL [231]. Milani *et al.* [231] classify XRL methods into three categories based on the targeted RL agent component: 1) *Feature Importance*: explaining influences on the agent in a given state, 2) *Learning Process and Markov Decision Process*: identifying influential training samples, and 3) *Policy-Level*: describing overall policy behavior. We adopt a feature importance approach, using an attribution

method to obtain heatmaps over sensor readings, which are subsequently processed to map sensor-level saliency to semantic scene elements for non-expert users. Importantly, the goal of this work is not to develop a new XAI method, but rather to evaluate the effectiveness of existing XAI techniques when embedded within an immersive VR environment. We assess our approach through a dual evaluation strategy combining a proxy task and a questionnaire, addressing both objective performance-based and subjective user-centered criteria.

8.2.2 Explainability in Robotics and HRI

Although XAI methods are largely developed for technical settings, applying them in human-centered robotics requires grounding abstract model outputs in ways that support user comprehension. This is particularly important in HRI, where robotic behavior should be intuitive for users to understand.

Halilovic *et al.* argue for tailoring robot explanations to the users' cognitive capabilities and task context [224], recognizing that overly abstract explanations may hinder comprehension among users. In another study, they present a real-time, multimodal explanation system that incorporates robot personality and spatial context to modulate the explanation strategy [233]. Our interface similarly operates in real time, but focuses on visually and spatially grounding attributions of RL-based decisions through semantic object highlighting. It therefore addresses the dynamic environmental context.

Das and Chernova introduce a framework that generates semantically grounded explanations for robot failures using scene graphs and pairwise ranking to highlight relevant spatial relations and object attributes [234]. Their method improves the user understanding of the robot by linking failures to specific semantic elements in the scene. We adopt this notion of semantic grounding and extend it to navigation, projecting attribution scores onto meaningful objects within an immersive virtual environment.

Wang *et al.* explore the use of augmented reality to display robot intentions to users [235]. Their augmented reality interface aids spatial awareness and interpretability by projecting the robot's internal states into the user's visual field. We adopt a similar spatial visualization paradigm but in a VR setting, enabling tighter integration of policy explanations with environmental semantics of the scene.

He *et al.* combine SHAP-CAM with depth-based RL to highlight influential input regions in drone navigation policies [201]. By overlaying saliency maps on depth images, they contribute a technically grounded approach to interpreting deep RL policies. While their visualizations remain on a technical level, our work embeds attribution-based explanations into a user-centric, spatial, and interactive VR interface, thereby enhancing the interpretability of RL policies through situated and dynamic visualization.

Hald *et al.* examine the role of robot explanations following task failures, concluding that while such explanations can guide users toward appropriate trust calibration, they are insufficient alone to repair trust [225]. Rather than post hoc trust repair, our system supports continuous, real-time visual saliency explanations, aiming to proactively

support the formation of calibrated trust during task execution.

Finally, Edmonds *et al.* investigate how different explanation modalities affect human trust in robots, comparing real-time visualizations of internal decision-making to summary text explanations [236]. They show that comprehensive, real-time visual feedback is more effective in fostering trust, even when not aligned with task-optimal model components. We adopt this insight by using dynamic visualizations of attribution scores during navigation, embedding them in a VR interface to enhance user understanding and trust.

Against this background, we hypothesize that the visualization of the XAI outputs improves

H1 users' objective understanding of the robotic decision-making process,

H2 users' subjective ability of perceiving, understanding and predicting the robotic information, and

H3 calibrated trust towards the robot.

We additionally explore whether this potential benefit is more pronounced when the visualization of XAI output is complemented with the visualization of the robot's sensor.

8.3 Methodology

This section introduces core concepts such as VR interface, robot navigation policy, explainability method and post-processing, and the user study setup.

8.3.1 Virtual Reality Interface

To visualize a robot navigation task for the user, we develop a VR interface based on the Unity game engine, optimized for Meta Quest 3 hardware, see Figure 8.2. The VR scene shows the robot navigating from a start to an end position while avoiding 3D obstacles, e.g., furniture and other objects. The goal location of the robot is visualized as a green circle on the floor. The user observes the robot navigation task from a fixed position nearby. Unity handles the simulation of the robot's top-mounted 2D lidar sensor through ray-casts. For the perceptual explainability, we visualize the otherwise invisible lidar rays in VR by rendering their 3D raycasts in real-time. The simulated 360 rays are displayed within the policy's detection range of $6\,\mathrm{m}$. When a ray intersects with an object, its color changes from green to red, providing an immediate visual cue of potential obstacles. Furthermore, 3D objects are highlighted with an outline of dynamic width to display their importance reflecting the XAI outputs, as further elaborated in Section 8.3.4.2. The Unity interface exchanges states, actions, and attribution scores with the RL policy and the attached explainability pipeline on a Python server via a socket connection. This data is sent to the server at the inference frequency of the policy, which also triggers updates of the XAI visualizations.

8.3.2 Navigation Policy

We employ an RL-based robot navigation policy π driven by a neural network learned using the TD3 algorithm [103]. The policy is trained for obstacle avoidance on its way to a local goal using a 360° 2D lidar sensor in environments with randomized obstacle and goal positions. The state $s = [L, G]$ consists of 15 entries of min-pooled lidar sensor data L, down-sampled in sectors from 360 rays, and 2 entries of the robot-centric goal in polar coordinates G. The policy produces a two-dimensional output dictating linear (v) and angular (ω) velocity commands for the robot as action $a_t = (v, \omega)$. The learning task is described to the RL agent with a sparse goal reward ($+20$), sparse collision (-20) and timeout penalties (-1), jerk ($-1\mathrm{e}{-7} \cdot \left\|(a_t - 2a_{t-1} + a_{t-2}) f^2\right\|^2 / J_{\max}$) and time penalties ($-0.001$), and an obstacle distance-keeping penalty (-0.001 if $d_{\text{min}} < 0.4\,\mathrm{m}$) based on the distance to the nearest obstacle d_{min}. The multi-layer perceptron policy contains three layers with [256, 128, 64] neurons respectively and is trained for 500k time steps using Stable-Baselines3 [200].

8.3.3 Attribution Scores of the Navigation Policy

Attribution methods quantify the influence of each input dimension with respect to the model decision for a single input sample. Within this category, several methods have been proposed [237], [238], [239], [240], which differ not only in their conceptual underpinnings, but some also require non-trivial choices of hyperparameters that can influence the outcome significantly [241]. For its conceptual and algorithmic simplicity, we use the gradient of the policy with respect to an input state s at timestep t as the attribution method [240], a method also known as Vanilla Gradient. We selected Vanilla Gradient for two practical reasons: it is computationally efficient enough to support real-time inference, and it requires no hyperparameter tuning, which simplifies implementation and ensures reproducibility. We emphasize that attribution methods explain a scalar output, i.e., in the case of our policy network π, the output of a single neuron. Although explanations for both linear and angular velocity of the robot could be combined, the complexity of their interaction and the necessary communication to users exceed the scope of this work, which focuses on the VR projection of these explanations. Therefore, we restrict our analysis to attribution scores for the robot's linear velocity v. Further, we solely focus on explanations of the perception-part of the state space, hence we select the components from the gradient that correspond to the lidar components L of the input. The goal location G, while essential for task execution, serves as contextual information rather than direct sensory input and is visualized separately in the VR environment, without additional dynamic highlighting. To summarize, the attribution scores $\vec{g}$ are given by:

$$\vec{g} = \left(\frac{\partial \pi(s_t)_v}{\partial s_t} \right)_L \tag{8.1}$$

Because the scores are derivatives, their interpretation is as follows: Values close to zero indicate that a feature has no or little influence on the policy output. A high posi-

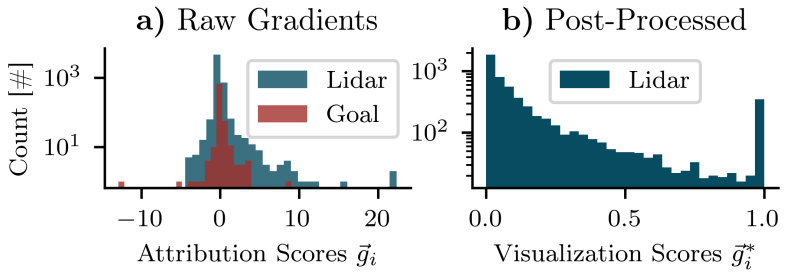


Figure 8.3: **a)** The distribution of raw lidar attribution scores $\vec{g}$ provided by Vanilla Gradient for all navigation state-action pairs presented during the user study. **b)** After postprocessing for visualization (Equation 8.2), the distribution of $\vec{g}^*$ shifts into the range $[0, 1]$.

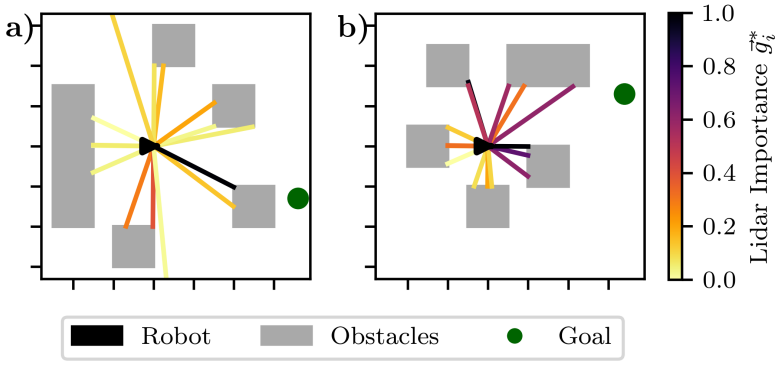


Figure 8.4: Example scenes with post-processed XAI attribution scores $\vec{g}^*$ of the linear velocity output, indicated as color-coding for their respective min-pooled lidar ray. The robot (black triangle) is facing to the right, while different obstacles (grey boxes) influence the navigation policy that should pursue the goal (green dot). Depending on the scene setup, the obstacle's influence on the policy varies. Axis ticks denote $1\,\mathrm{m}$ distances.

tive attribution value for a lidar input indicates that increasing the corresponding depth reading (i.e., perceiving more open space) results in a higher velocity command. Conversely, a high negative attribution implies that a decrease in the depth reading causes the policy to increase velocity. Figure 8.3a (blue) presents a histogram over all values in all $\vec{g}$ provided by Vanilla Gradient for all navigation state-action pairs presented during the user study. Note the logarithmic scaling of the y-axis. The vast majority of lidar attribution scores are around zero or positive, indicating a learned tendency to reduce forward speed upon nearing obstacles. The red histogram shows the attribution scores corresponding to the goal G. We see that scores for the goal are closer to and centered around zero, indicating that the policy primarily focuses on the lidar input.

8.3.4 Visualizing Attribution Scores

In order to transform the abstract numerical attribution scores into an intuitive visual representation, we apply two post-processing steps: a) We further simplify the attribution scores and b) connect them with the scene by associating each with an object to achieve a semantic mapping.

8.3.4.1 Simplification of Attribution Scores

While the sign of the attribution scores does have a semantic meaning, as discussed above, the pure magnitude is of far greater importance. Hence, we work with the absolute value, discarding the sign. Further, the raw values of the attribution scores are less important for the ranking task than their relative relationship. Hence, we apply a rescaling operation to obtain a mapping to range $[0, 1]$ for each $\vec{g}$. The full transformation of $\vec{g}$ to post-processed $\vec{g}^*$ is given by

$$\vec{g}^* = \frac{|\vec{g}| - \min(|\vec{g}|)}{\max(|\vec{g}|) - \min(|\vec{g}|)} \tag{8.2}$$

The effect of this post-processing is shown in Figure 8.3b. The scores now more uniformly span the full spectrum, yielding a visually uniform grading. The abrupt peak to the right implies that in many cases the majority of the attribution mass concentrates on few lidar rays.

8.3.4.2 Score-to-Object Mapping

In Figure 8.4a and b we can see an abstract visualization of the robot in two different scenes. The 15 lidar rays are colored according to their corresponding value in $\vec{g}^*$. In Figure 8.4a the policy has a strong focus on the goal-occluding obstacle. In Figure 8.4b where the obstacles are generally closer to the robot, the policy's focus in the direction of the goal is less sharp. Overall, the backward-facing lidar rays receive less attribution. To perform the semantic mapping of our post-processed attribution scores into the scene, we associate each lidar-hit object in the robot's vicinity with the score of the hitting lidar ray in $\vec{g}^*$. If an object is intersected by multiple rays, as in the illustration, the maximum value from all candidate rays is used. Finally, object importance is visualized in the VR environment by outlining affected objects in white, using a world-space line thickness proportional to their importance.

8.3.5 User Study

The user study is designed to evaluate the impact of different configurations of the VR interface on human objective and subjective understanding of the robot's navigation decisions, as well as their trust in the robot.[8]

8.3.5.1 Design

We assess the objective understanding of the robot's navigation decisions in a ranking task, in which participants are tasked to rank the importance of objects for the robot's navigation policy in four blocks. After each ranking block, subjective measures are taken using a questionnaire. The questionnaire is conducted directly in VR and includes 8

[8] A video of the interface and user study setup is linked in the supplemental material section of the thesis appendix.

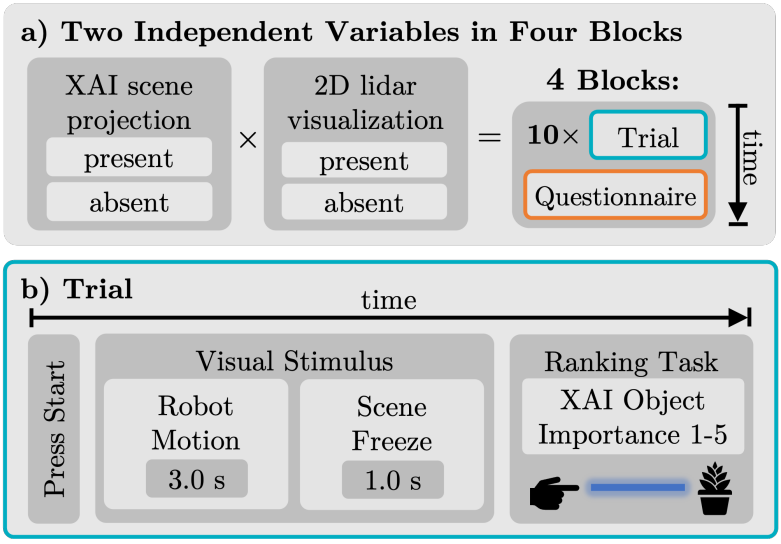


Figure 8.5: **a)** Fully-crossed combinations of two independent variables (IVs): XAI scene projection and 2D lidar sensor visualization. They sum up to four experimental conditions, represented by four blocks. Each block was followed by a questionnaire. **b)** Participants start each trial pressing the A button on the controller. The robot navigated for 3 s, halted, and the XAI and/or lidar visualization remained after another second. Afterwards, participants ranked the importance of five scene objects for the robot policy.

questions (7-point Likert scale, labels: "Totally Agree" and "Totally Disagree"), see Section 8.4.1.2 and Figure 8.7.

We employ a two-factorial within-subjects design to isolate the effects of two visualization features: XAI (present, absent) and lidar (present, absent). Their fully-crossed combinations result in four interface configurations, as illustrated in Figure 8.5a. Each block presents one of these configurations and consists of 12 trials with a unique navigation scenario. Scenarios are configured by varying five obstacle placements, robot start and goal locations, and the participant's observer position. The robot is initialized facing the goal direction. In total, 48 unique scenarios are randomized across all four blocks. To mitigate training and ordering effects, the sequence of the blocks is fully counterbalanced, resulting in $4! = 24$ unique orderings. The study involves 24 participants, each assigned a different block sequence.

8.3.5.2 Ranking Task

Each trial begins with the presentation of a new robot navigation scenario including five obstacles, which the participant views from a distinct perspective, see Figure 8.5b. The number of obstacles is kept constant across trials to ensure a similar difficulty of the ranking task. Depending on the experimental block, either the XAI or lidar visualization was shown. The robot's goal position is indicated by a green torus, and a real-time updated line connects the goal and current robot position for a clear navigation context. The robot starts to move when the participant presses a button of the handheld

controller. After $3\,\mathrm{s}$, the movement is paused and marked by a stop sign on the robot. The final state of the visualizations remains visible for an additional $1\,\mathrm{s}$ to allow the participant to process the current navigation step, which they are instructed to base their ranking on.

Participants then rank the importance of each object with respect to the robot's policy by pointing and selecting the objects. Rank labels are displayed on top of the objects, ordered from most (1) to least (5) important. Participants can revise their ranking decision by pressing another button.

The collected rankings are later compared to ground-truth object importance derived from scene-projected attribution scores. To measure agreement between the participant's object ranking and the ground-truth importance order, we employ Kendall's τ [242], a non-parametric correlation metric. Kendall's τ quantifies the similarity between rankings by evaluating the proportion of concordant and discordant pairs.

8.3.5.3 Procedure

Before the experiment, participants received detailed instructions about the experiment, provided written consent, and completed a demographic questionnaire. They were informed about the robot's navigation task, the XAI output visualization and how its lidar sensor perceives the environment. The experimenter instructed them for the ranking tasks (S1). Each participant completed two training trials with explanations to become familiar with the visualizations and ranking task and proceeded with the first experimental block. After they had completed the first ranking block, they answered the questionnaire measuring the subjective experience of the previously presented interface configuration (S2). Upon completion of all ranking blocks and questionnaires, participants answered a freeform questionnaire targeting their object ranking strategy (S3).

8.3.5.4 Participants

A total of $N = 24$ individuals (9 women, 15 men) participated in the study in exchange for a EUR 10 monetary compensation. All participants reported having (corrected-to-) normal vision. Their mean age was 24.6 years (SD = 3.6 years). Participants rated their experience with AR/VR on a 7-point Likert scale (1 = No experience at all, 7 = A lot of experience), with a mean rating of 2.4 (SD = 1.3). Participants also rated their experience with robotics (M =2.8, SD = 1.9), and their experience with artificial intelligence (M = 4.1, SD = 1.8). The study adhered to the principles outlined in the Declaration of Helsinki.

8.4 Experimental Evaluation

This section presents the results of the user study, which evaluates the established hypotheses (H1 - H3).

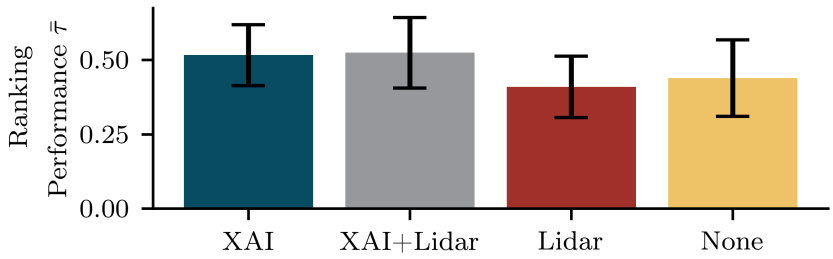


Figure 8.6: Object ranking performance (S1) of participants for each presented visualization combination of XAI and lidar conditions as measured by Kendall's τ. Means and standard deviations are shown. As can be seen, the XAI visualizations increase the users' object ranking performance with respect to their attribution score-derived ground truth importance.

Predictor	df_d	df_n	F	p	η_p^2
XAI	**1**	**23**	**15.20**	**< .001**	**0.40**
Lidar	1	23	0.20	.657	0.01
XAI × Lidar	1	23	0.92	.346	0.04

Table 8.1: The results of rmANOVA of the ranking task performance (S1) demonstrate a significant effect of the semantic XAI visualization.

8.4.1 User Study

The collected data covers the objective visualization-dependent object ranking performance (S1) and subjective evaluations of the post-block questionnaire (S2).

8.4.1.1 Ranking

The ranking task (S1) quantitatively assessed users' understanding of the XAI visualizations. We compute Kendall's τ between the participants' ranking and the ground truth order of objects for every trial and aggregate the results for each of the four experimental conditions and each participant, see Figure 8.6.

A repeated-measures (rm)ANOVA confirms a significant effect of the XAI visualization on the participants' ranking performance of the five scene objects, see Table 8.1. Participants performed better with ($M = 0.52$, $SD = 0.11$) than without XAI ($M = 0.42$, $SD = 0.12$). This benefit is expected as the XAI visualization conveys attribution scores. Although neither the main effect of the lidar visualization nor its interaction with the XAI visualization was significant, participants achieved descriptively the best ranking performance when both XAI and lidar were visible.

Interestingly, even without visualizations, participants achieved a certain ranking accuracy, possibly by using heuristics, e.g., prioritizing objects that are closer to or in front of the robot, or those that appeared to influence its navigation.

While edge cases such as ties in the ground truth order due to occluded objects receiving a zero importance score or subtle importance differences indistinguishable from outline thickness can impact Kendall's τ as the absolute ranking performance, the relevant conclusions are to be drawn from the relative performance differences.

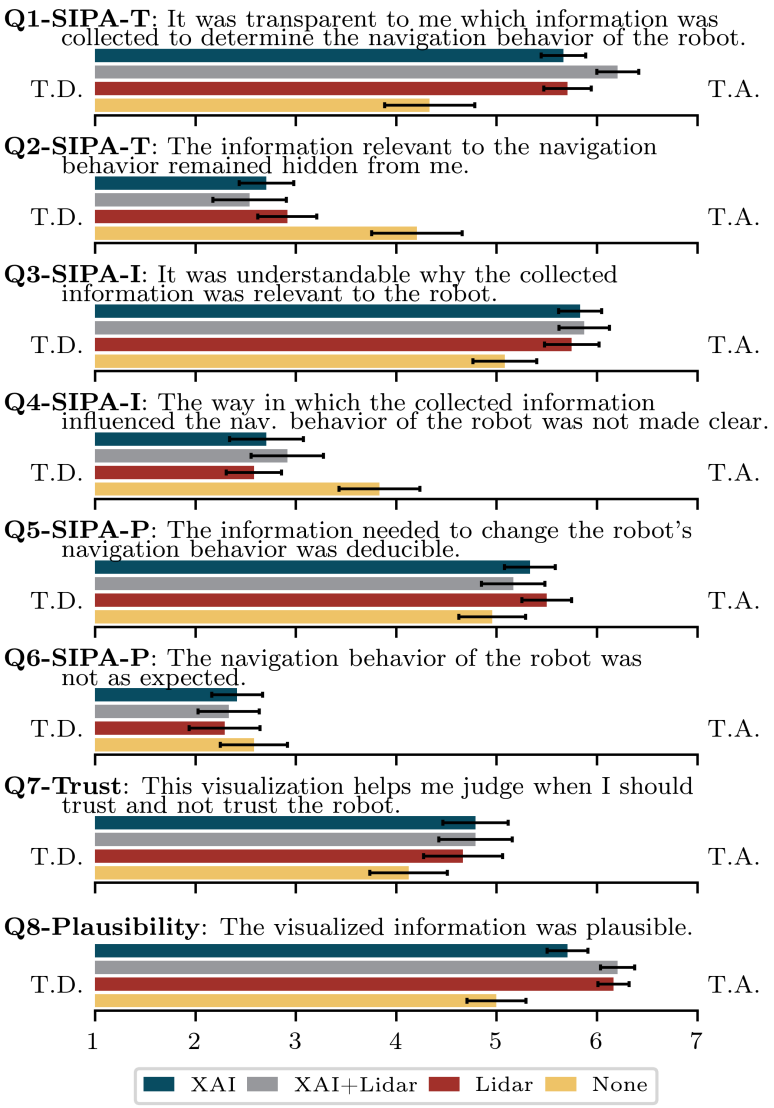


Figure 8.7: Participants rated their experience with respect to the SIPA scale for explanations (Q1-Q6), trust (Q7) and plausibility (Q8) with regards to the visualizations XAI and lidar (IVs). All questions shared the same labels: Totally disagree (T.D.) and Totally Agree (T.A.), abbreviated here for visual clarity. Ratings were provided on a Likert scale (1-7), bars indicate score means, error bars show standard errors.

In conclusion, the participants showed an improved understanding of the object importance to the RL policy with the XAI visualization, finding support for H1.

8.4.1.2 Questionnaire

Targeting the users' understanding of our explanation visualizations in terms of transparency (T), intelligibility (I), robot predictability (P), and trust (T) towards the robot, we analyze the 8-item questionnaire (Likert scale, score 1-7) of S2, see Figure 8.7. Reverse-coded items (Q2, Q4, Q6) served as attention checks.

Items Q1-Q6 represent a short version of the Subjective Information Processing Awareness (SIPA) scale [243] for user-centered assessment of XAI: Perception (Q1+2),

Item/Scale	*Predictor*	df_d	df_n	F	p	η_p^2
Q1-6: SIPA	**XAI**	**1**	**23**	**5.39**	**.030**	**0.19**
	Lidar	**1**	**23**	**4.43**	**.046**	**0.16**
	XAI × Lidar	**1**	**23**	**6.90**	**.015**	**0.23**
Q7: Trust	XAI	1	23	1.62	.216	0.07
	Lidar	1	23	1.72	.202	0.07
	XAI × Lidar	1	23	0.80	.380	0.03
Q8: Plaus.	XAI	1	23	3.86	.062	0.14
	Lidar	**1**	**23**	**22.77**	**< .001**	**0.50**
	XAI × Lidar	1	23	3.26	.084	0.12

Table 8.2: Results of the rmANOVAs for post-block questionnaire (S2), specifically for the short SIPA scale (Q1-Q6), Q7-Trust and Q8-Plausibility.

intelligibility (Q3+4) and prediction (Q5+6). We invert the reverse-coded items, aggregate the scores of the single SIPA items to a mean score for each experimental condition, and perform a two-factorial rmANOVA (Table 8.2) to infer the contributions of the XAI and lidar visualization conditions.

Both XAI and the interaction of XAI and lidar show a statistically significant effect on the mean SIPA score, with the conditions XAI ($M = 5.5$, $SD = 1.0$), XAI+Lidar ($M = 5.6$, $SD = 1.0$), and Lidar only ($M = 5.5$, $SD = 1.0$) achieving higher SIPA scores compared to the condition without XAI and lidar ("None") ($M = 4.6$, $SD = 1.4$). This underlines that any additional visualization of the robot's information processing (XAI, lidar or both) improves the participants' impression of being able to perceive, understand and predict the robot's navigation behavior, supporting H2.

Item Q7 assesses participants' trust calibration towards the robot and was derived from the Explanations Satisfaction Scale (ESS) [244]. While the visualizations (XAI, XAI+lidar, lidar) showed a descriptive improvement over "None," no significant effects were found (not supporting H3). The absence of significance may be related to specific aspects of the study design: First, the use of a single-item metric may not have been sensitive enough to capture changes of trust calibration in this study. Second, the visualizations may not have been relevant to trust calibration. Also, from a participant's perspective, it may not have been clear what to trust the robot for, as the participants observed the error-free robot navigation from a distance and were not personally involved in the scenario.

The final item Q8 targets the plausibility of the visualized information and refers to the explanation concept of coherence [232]. Here, the lidar has a significant effect on the measured plausibility of visualizations. Both XAI and its interaction with lidar are not significant. This indicates that the lidar visualization rather than the XAI visualization appears more plausible for users, presumably because the lidar rays are directly linked to the robot's perception.

We conclude that the semantic XAI projection helped the users to objectively perceive the information leveraged by the navigation policy, and also created the subjective

impression for users of being able to perceive, understand and predict the robot's information processing, i.e., decision-making in navigation behavior (supporting H2). While the objective understanding in the ranking task was not significantly affected through the visualization of the lidar rays, the subjective information processing awareness (perception, understanding, predicting) of the users as well as the perceived plausibility of the interface improves. Finally, neither the semantic XAI projection nor the lidar visualization changes the user's impression of enhancing the trust calibration process.

8.4.1.3 Freeform Feedback

Upon completion of all blocks, we asked participants two freeform questions (S3) to learn about their mental model (RQ3) and object ranking strategy: FQ1 - What rules do you think the robot followed when choosing a path? FQ2 - Which strategy did you use for ranking objects' influence? Participants identified several recurring patterns when asked about their mental models (FQ1) and object ranking strategies (FQ2). Regarding robot path selection rules (FQ1), participants frequently stated that the robot prioritized collision avoidance, selected shorter and direct routes for efficiency, showed differential treatment to objects based on their distance, and favored smooth trajectories. Almost all responses built upon the constellation of objects, i.e., the scene context, rather than the perception and action capabilities of the robot. This underlines the relevance of scene context for the participants' mental model of explainability and the need to promote their awareness of the robot's perception. For the ranking task (FQ2), common strategies included considering the object's outline thickness, object size or perceived collision hazard, proximity to the robot's intended path, and frequently a combination of these factors.

8.5 Conclusion

We present a novel VR-based interface that integrates dynamic, scene-grounded XAI outputs and sensor visualizations to support non-expert users in understanding an RL-based robot navigation policy. We thereby align numerically obscure robot policy explanations to the users' cognitive capabilities and task context. Our user study shows that semantically projecting attribution scores significantly improves non-expert users' objective understanding and subjective awareness of the robot's decision-making, thereby increasing the perceived predictability of its behavior. Visualizing the robot's lidar rays also contributes substantially to users' subjective awareness, indicating that combining XAI and sensor visualizations is essential for optimizing user experience in VR. Based on these findings, future research should jointly evaluate objective and subjective metrics to guide the design of effective explanation tools for human-robot interaction. In this study, the visual explanation of the robot's scene understanding was presented independently at each time step. An important avenue for future research is the incorporation of the temporal dimension of robot trajectories into the XAI method—for example, gen-

erating a unified explanation that captures and visualizes the policy's reasoning over the entire trajectory history. Such temporal integration becomes particularly relevant in environments featuring dynamic obstacles. Moreover, investigating how users transfer the understanding they acquire to novel navigation scenarios would be valuable. Specifically, this could be examined by quantitatively assessing their prediction performance regarding obstacle influence and the resulting policy behavior, thereby directly targeting a learning effect. Overall, our results highlight the potential of immersive VR explanation interfaces to facilitate more transparent human-robot interaction in complex environments, with applications in supporting safer collaboration, aiding debugging and validation, and facilitating training and education.

With the presented findings, this chapter represents a direct contribution to overarching RQ4, see Section 1.2.4, and contextualizes the presented DRL navigation controllers of the preceding chapters from an HRI perspective. Furthermore, this chapter is the final one to present empirical results from conducted research. The next chapter introduces the recent advances in foundation models for robotics, followed by an outlook and a conclusion chapter (Chapter 10 and 11, respectively).

9 Foundation Models in Robotics: Toward Personalization

In the previous chapters of this thesis, we developed robust and efficient methods for capturing, learning, and integrating human preferences into personalized robot navigation policies. Recently, robotics started entering a phase of transformative change driven by the emergence of Foundation Models (FMs), which introduce new capabilities in generalization, reasoning, and multimodal interaction. These developments open novel opportunities to further advance personalization, adaptability, and user-centric explainability in robotic systems. Accordingly, this chapter broadens the perspective of the thesis by providing an introduction to FMs, discussing their potential benefits, challenges, state-of-the-art approaches, and promising directions for future research in navigation, HRI, and personalized robotics. Following this chapter, we will contextualize our works against the advances of FMs in Chapter 10.

9.1 FMs for Embodied Intelligence

FMs refer to large-scale pre-trained neural networks designed to leverage extensive and diverse internet-scale datasets to establish robust knowledge bases [245], [246]. These models exhibit strong adaptability and generalization capabilities, enabling their application across various tasks and domains [247].

In robotics, FMs recently received outstanding attention as they are promising architectures to the long-standing goal of achieving general-purpose AI [246]. In that regard, they have the potential to reduce the dependence on task-specific highly engineered models and policies. One property makes them specifically appealing: they enable intuitive control over robots through natural language instructions [248], or simply by showing the robot an image of a task's goal state [249]. This significantly simplifies the interaction paradigms between human and robot.

Foundation models in robotics can be categorized into several types: large language models (LLMs), vision-language models (VLMs), robot foundation models (RFMs), and embodied multimodal models (EMMs) [245]. LLMs excel in communication, task planning, and common-sense reasoning, e.g., Llama [250] or the Generative Pre-training Transformer (GPT) architecture [251]. VLMs enable open-vocabulary visual recognition, semantic scene understanding, and grounding of language inputs into visual observations, with prominent models such as CLIP [252] and BLIP-2 [253]. RFMs typically integrate vision, language, and action modalities (hence also called Vision-Language-Action model, short VLA), enabling direct generation of robot control commands from multimodal inputs. Notable example architectures include RT-1 [254], RT-2 [247], Octo [255], Open-VLA [248], and GR00T N1 [256] by Nvidia. Extending the typical VLA modalities, a recent RFM additionally incorporates proprioceptive and

other sensor data, supporting richer sensorimotor reasoning [257]. Finally, EMMs, such as PaLM-E [258], Gato [259], and RoboCat [249], are general-purpose architectures trained across various robot embodiments and sensorimotor modalities, including vision, language, proprioception, and low-level control. They are specifically designed to support cross-embodiment policy reuse and zero-shot task transfer.

The adoption of FMs in robotics offers substantial benefits. As mentioned earlier, a key advantage is the generalizability and adaptability to diverse instructions and unseen environments [245], [246], [260]. Pre-trained FMs enable rapid fine-tuning to a specific task with comparatively little training data, thereby improving data efficiency as compared to a task-specific policy trained from scratch [245], [261], [262], [263]. For analogous reasons, FMs are characterized by a high task-versatility given minimal fine-tuning, allowing robots to tackle diverse challenges from manipulation and navigation [256], [258] to human-robot interaction [264]. Furthermore, FMs are capable of rich contextual understanding and advanced reasoning, thereby significantly enhancing robotic planning, problem-solving and decision-making capabilities [247]. For example, a robot can infer that the instruction "prepare for guests" entails both cleaning and organizing without explicit task decomposition. Due to their exposure to diverse training data, FMs enhance the error tolerance and robustness in previously unseen situations [256], [258]. Lastly, their training data-induced native linguistic capabilities enable natural communication with users and substantially facilitate natural HRI [264].

9.2 Limitations of FMs

Despite their promising potential, the deployment of FMs in robotics currently faces several challenges. A primary issue is the embodiment gap, as FMs are typically pre-trained in non-embodied contexts such as internet-scale text and images datasets, but are supposed to operate in the physical world [246]. For instance, the description of how to grasp an object in text does not easily translate to a robot physically picking it up. Additionally, while non-embodied training data such as text, images, and videos are broadly available online, high-quality domain-specific robot training data is scarce [265]. This is influenced by the substantial costs and logistical complexities involved in recording and annotating extensive robotic datasets [254], [265], [266]. Another unresolved challenge is how to structure an FM's output such that its predicted actions can be effectively interpreted and executed by different robotic embodiments. For instance, slight incongruencies in robot geometry, locomotion parameters, joint configurations, or end-effector designs should not significantly constrain the transferability of a costly trained FM. Another challenge lies in the computational complexity of FM inference due to large model sizes and their transformer-based autoregressive architecture, which makes it difficult to achieve real-time performance [259], [267]. Yet, real-time inference at high control frequencies is essential in scenarios where robots must instantly react to dynamic environments, such as in a social navigation scenario [268]. Finally, it remains a challenge

to establish reliable benchmarking frameworks to ensure reproducibility, operational safety and uncertainty quantification across different robot platforms and research institutions [246].

Regarding the integration of generic LLMs with robotic architectures, Kim *et al.* [269] and Wang *et al.* [270] point out some notable limitations beyond the aforementioned general characteristics of FMs in their surveys: LLMs tend to generate inaccurate or unexpected responses, also called hallucinations [269], potentially leading to misleading and unexpected user experiences. Furthermore, their emergent capability of in-context learning (ICL) is not yet consistent or reliable in practice.

Also, the effectiveness of LLMs relies on the wording and quality of prompts, with some models requiring carefully crafted, lengthy prompts for reliable outputs [270]. When integrating VLMs in robotics, a key limitation can be the inability to fully grasp spatial relationships between objects and depth information in general [270], as depth and 3D data are typically not included in their training domain. Given that the safety of robotic systems is critical for deployment, LLM-based robotic applications therefore require filtering and correction mechanisms to ensure deployment safety [269].

9.3 FMs for Robot Navigation

This section illustrates how FMs are applied to navigation tasks in the first place, before we gradually transition to personalization of robot navigation through FMs in Section 9.5. Generally, strong visual context-based reasoning makes FMs a natural fit for navigation tasks in unknown environments.

9.3.1 Vision-and-Language Navigation in Static Environments

For the task of vision-and-language navigation (VLN, also referred to as open-vocabulary navigation), recent approaches commonly leverage world knowledge of pre-trained LLMs and VLMs for zero-shot performance on navigation tasks [271], [272], [273], [274], [275]. While these approaches share a common goal, they employ different strategies to achieve it.

One strategy involves the direct use of open-vocabulary image classification models, as exemplified by Gadre *et al.*'s "CLIP on Wheels" (CoW) [271]. Here, the authors leverage CLIP's robust visual-semantic embeddings in combination with a simple exploration policy, enabling agents to recognize goal objects in unknown environments without explicit training. Despite demonstrating promising zero-shot capabilities, CoW's effectiveness remains limited, primarily due to shortcomings in exploration strategies and inconsistent object recognition triggered by CLIP. This suggests that purely reactive approaches may not sufficiently exploit semantic context for navigation.

To address the limitations inherent in reactive, classification-based methods, Huang *et al.* [272] and Long *et al.* [274] adopt a complementary, planning-oriented approach. Both methods first construct explicit intermediate representations,

spatial-semantic maps or landmark-based value maps, which are derived from VLM and LLM outputs, respectively. Specifically, Huang *et al.*'s VLMaps projects pixel-level embeddings from VLM-encoded images onto reconstructed 3D environments enriched with linguistic annotations. An LLM subsequently utilizes this structured representation to generate actionable, zero-shot navigation instructions. Long *et al.* similarly employ an LLM to decompose high-level instructions into explicit landmark-action sequences (InstructNav), which are then converted into spatially referenced value maps that guide navigation. Both approaches highlight a critical paradigm shift from purely reactive decision-making toward structured semantic reasoning, substantially outperforming simpler baseline models like CoW in multi-object navigation tasks. Nevertheless, the dependence of InstructNav on closed-source commercial models to achieve the highest performance underscores significant concerns regarding accessibility, transparency, and reproducibility in practical applications.

In contrast to these mapping-centric approaches, Lin *et al.* [273] and Zhang *et al.* [275] emphasize the direct generation of low-level control commands from visual-language inputs, respectively. Lin *et al.* propose ADAPT, which significantly deviates from on-the-fly planning by retrieving suitable navigation actions from a pre-built prompt database. The strength of ADAPT lies in explicitly aligning actions with language instructions, thereby equipping agents with robust cross-modal reasoning abilities. Rather than dynamically synthesizing behaviors, the agent retrieves known, verified actions from a database. On the other hand, Zhang *et al.* present NaVid, a notable approach for continuous-environment navigation that directly generates low-level motor commands from video-based inputs, leveraging a general-purpose video-based VLM architecture. Unlike VLMaps and InstructNav, NaVid does not require explicit depth, odometry, or intermediate map information for a successful real-world deployment. Nonetheless, NaVid's reliance on computationally heavy video models results in significant inference latency, limiting the approach's applicability in dynamic environments.

Taken together, these recent approaches to VLN in static environments illustrate two principal trends. On one hand, structured semantic reasoning via intermediate representations (VLMaps, InstructNav) provides robust interpretability and strong task performance but introduces complexity and possible external dependencies. On the other, direct, cross-modally aligned action generation (ADAPT, NaVid, CoW) simplifies decision-making pipelines but faces challenges related to generalization, computational costs, and consistency of zero-shot predictions. As emphasized by Wu *et al.* [276], challenges remain across all methodologies, including limited generalization to unseen environments, inefficiencies in real-time computation, and sim-to-real transfer hurdles. New evaluation metrics are being refined to better quantify instruction following, but further research is needed to overcome dataset biases and enable interactive, lifelong learning in real-world settings.

9.3.2 Social and Dynamic Navigation

While the aforementioned VLN approaches focused on static environments, foundation models have also been applied to social navigation tasks in dynamic human environments [101], [268], [277], [278]. These methods emphasize continuous contextual understanding, adaptation to human behaviors, and lifelong learning to facilitate robust interactions.

A prominent paradigm in recent literature is the utilization of VLMs for robust semantic understanding in dynamic human environments. For instance, Narasimhan *et al.* propose the OLiVia-Nav framework [277], where a VLM continually assesses and encodes social and environmental contexts to maintain a dynamic database of previous encounters. Through periodic internal updates of its visual embedding representations, OLiVia-Nav incrementally improves its own scene understanding. This lifelong learning approach explicitly targets improved generalization to novel social scenarios, reflecting the necessity of adaptability in real-world dynamic settings.

A similar emphasis on contextual reasoning is found in the CoNVOI approach by Sathyamoorthy *et al.* [268]. Here, a VLM directly interprets context-based instructions to generate trajectories around obstacles and humans without additional training or fine-tuning. These semantically appropriate, zero-shot trajectories are then executed by a dedicated traditional motion planner based on the Dynamic Window Approach.

Other works merge FMs with more traditional established controllers based on optimization or DRL [101], [278]. Song *et al.* [278] propose the VLM-Social-Nav framework that integrates real-time scene interpretation with optimization-based planning. Upon detecting human presence, VLM-Social-Nav assesses the context through RGB input, generating explicit social navigation commands consistent with a provided contextual rule set. These commands are subsequently converted into a cost-based representation. Based on the cost representation, the optimization-based planner simultaneously considers obstacle avoidance and goal-directed navigation. This hybrid formulation effectively combines semantic reasoning with classical optimization methods, though the necessity of rapid inference still poses challenges to real-time applicability in highly dynamic scenarios.

A similar hybrid approach is represented by the SRLM framework of Wang *et al.* [101]. It integrates LLMs and DRL to enable human-in-the-loop social robot navigation. Their approach converts natural language feedback into structured reward signals, subsequently used to fine-tune a DRL policy. Importantly, SRLM dynamically combines the DRL-derived navigation policy with a separate VLM-based action policy, achieving both immediate contextual generalization from foundation models and the adaptability provided by incremental policy updates.

The aforementioned approaches explicitly recognize the limitations of the pure zero-shot paradigm for navigation in dynamic human environments and pair their VLMs with reactive, more established motion controllers.

In summary, these approaches highlight a clear shift toward the integration of the

zero-shot semantic reasoning of FMs into navigation approaches for dynamic human-centered environments. However, common limitations include a high inference latency and limited robustness in dynamic or complex settings.

9.4 Enhancing Human-Robot Interaction with FMs

Through their integration of natural language and multimodal reasoning, FMs offer significant potential to enhance HRI. A recent survey by Shi *et al.* [279] points out how LLMs in robotic systems enable more coherent, personalized, and emotionally aware conversations, especially for vulnerable user groups. Additionally, VLMs promise understanding of affective and contextual human cues from multimodal data, addressing one of the central challenges in HRI. This section provides an overview of recent notable approaches that both integrate LLMs to interact with users [264], [280].

Bärmann *et al.* [280] propose a framework that integrates LLMs into the high-level control of a humanoid robot for dialog-based task execution and self-correcting behavior. Human feedback is processed via an auxiliary LLM that rewrites interaction transcripts, with these corrected examples stored in memory to improve future interactions with the user. The system enables robots to incrementally learn from suboptimal human interactions, thereby improving HRI by aligning robotic responses and actions with user expectations over time. Limitations the authors point out are the LLMs sensitivity to user prompt phrasing and the resulting possibility of misleadingly summarized interactions examples stored in memory.

User expectations also play a central role for Kim *et al.*, who investigate the design requirements for integrating LLMs with robots through a comprehensive user study [264]. It was found that the LLMs' enhancement of the HRI depends on task context, with higher user acceptance in learning and negotiation tasks, but reported communication hurdles and increased social pressure in efficiency-focused tasks. Their main finding is that advanced robot conversation skills raise user expectations for advanced non-verbal communication skills, underscoring the importance of aligning verbal and embodied behaviors. Currently, the conversation skills facilitated by LLMs significantly outperform the physical capabilities of robots, a significant pitfall in HRI.

9.5 Personalizing Robot Behavior via Language Interfaces

This section reviews approaches that utilize LLMs to personalize robot behavior and interactions based on inferred user preferences [28], [30], [99], [100], [199], [281].

9.5.1 Personalizing Robot Navigation and Manipulation

A major theme shared across several recent works is the integration of iterative and interactive user feedback loops into robot behavior personalization. Dai *et al.*'s ORION [99] exemplifies this strategy by embedding an LLM into a continuous think-act-ask loop to

iteratively refine robot navigation based on user inputs. A notable feature is the LLM's autonomous decision-making on when and how to query the user for task clarification, which allows the system to balance initiative and responsiveness. However, the work also highlights challenges in balancing task completion with navigation and interaction, as well as long-term memory retention. Similarly emphasizing iterative refinement, Han *et al.*'s LLM-Personalize [28] leverages repeated human preference signals to incrementally fine-tune long-horizon household task plans. Both approaches demonstrate improvements in task alignment through interaction-driven refinement.

Complementing iterative personalization, other approaches bridge FM-based semantic reasoning with classical control to enable real-time personalized behavior, conceptually similar to the hybrid non-personalized approaches discussed in one of the previous sections [101], [278]. Martinez-Baselga *et al.* [100] employ VLM-generated cost functions that reflect user instructions to directly modulate a classical Model Predictive Controller (MPC), effectively decoupling high-latency VLM inference from the low-latency motion controller. In this framework, high-level natural language queries such as "drive carefully" or "navigate as if you were in a hospital" are automatically translated into specific MPC navigation parameters for personalization.

Also hybrid by design, Hwang *et al.*'s modular MORL framework [30] translates natural language instructions into explicit numerical preference vectors, rapidly guiding behavior adaptations of vision-based object-goal navigation without necessitating policy re-training. Their approach is related to our human-aware navigation MORL approach based on 2D lidar presented in Chapter 5. These vectors are generated from demonstrations, comparative feedback, and verbal instructions. For demonstrations and comparative feedback, an optimization routine identifies the most suitable preference vector, whereas verbal instructions are translated directly into reward weights via an LLM. While the use of LLMs for generating numerical preference vectors is conceptually attractive, the limited scope of their user study and absence of a systematic analysis of the LLM preference translation process into preference vectors reduce the interpretability and generalizability of their results.

Interpretable and user-centric personalization is another critical dimension prominently featured in recent research. Wu *et al.*'s TidyBot [281] stands out by explicitly utilizing LLM-generated interpretable rules derived from minimal user interactions to manage household object rearrangement tasks. These reusable rules generalize user preferences that originate from a handful of preference examples, outperforming baseline methods with respect to preference in a $N = 40$-user study.

9.5.2 Personalizing Language Interaction and Dialogue

In parallel with behavioral personalization, recent works also explicitly address the personalization of robot dialogue and social interaction [282], [283]. Tang *et al.*'s [282] propose an LLM framework that integrates LLMs that enables robots to exhibit dynamic personalities and adapt to human users on an emotional level. However, their findings

seem preliminary due to the solely simulation-based validation; thus a study on a real robot with real users is the logical next step. Concurrently, Li *et al.*'s LD-Agent [283] highlights the benefits of maintaining long-term memory and dynamic persona modeling, enabling more coherent interactions across repeated encounters. Both approaches highlight the importance of sustained emotional and contextual personalization, but also stress the need for validation with real-world robots with a greater focus on user studies.

9.5.3 Conclusion

Synthesizing findings on personalization, Zhang *et al.*'s recent survey [284] provides a valuable overarching perspective, categorizing approaches by granularity (user-, persona-, global-level) and pointing to methods such as retrieval-augmented generation (RAG), prompt engineering, fine-tuning, and RLHF. Their survey also identifies major challenges, including the cold-start problem of user interactions, benchmarking limitations, and the lack of unified datasets across personalization levels, highlighting directions where further research is required.

To summarize, the presented approaches underscore the possibilities of FM-driven robot personalization via user instructions or feedback for navigation, manipulation, and dialogue interaction. They furthermore demonstrate zero-shot agent personalization through world knowledge of FMs while reducing the necessity for extensive preference data collection to a handful of natural language interactions. While the highlighted results for robot personalization are promising, they are still in early stages. Future research is required that involves users, subjective user-centric metrics, and real robotic systems.

9.6 Explainable Robotics through FMs

Recent efforts at the intersection of HRI and LLMs have explored how natural language interfaces can enhance the explainability of autonomous systems.

One commonality of all following approaches is to structure multimodal data into coherent explanations using LLMs or VLMs [285], [286], [287].

Sotomi *et al.* [285] propose a multimodal explainability module that integrates LLMs, VLMs, and Grad-CAM heatmaps to enable real-time, human-digestible verbalized and visual explanations during autonomous robot navigation. In a $N = 30$-user study, the authors find significantly improved social acceptance of the robot behavior among users using their approach. A key bottleneck of their approach is the high latency of explanation generation ($\sim 20\,\mathrm{s}$), also with regard to deployment in dynamic environments.

Wang *et al.* [286] propose RONAR, an LLM-based system that narrates the perceptual experiences of a robot to users. The approach conveys robot intent and failure analysis to users in narrative form, allowing users to better support robots to recover from failures. Their approach outperforms other state-of-the-art systems in a $N = 24$-user study,

effectively improving robot transparency in human interactions.

Liu *et al.* propose REFLECT [287], a framework that leverages LLMs for failure explanation and self-correction in the robot's task execution. By distilling multisensory robot data into the explanation, the approach corrects both planning and execution failures using progressive LLM prompting. The study does not involve human users, but demonstrates how formally user-motivated explanations can also improve the performance of robotic systems themselves. Limitations of the study lie in the static, simplistic environments, falling short of operation in complex dynamic scenes.

Generally, behavior explanation increases the user-perceived transparency [73]. When explanations are furthermore personalized, as demonstrated by Verhagen *et al.* [288], user trust towards the agent can be increased.

Based on the highlighted works, FMs are highly promising to convey a robot's complex internal state and perception to users in an intuitive manner. Not only does this have positive effects on the user, but can also benefit the robot's performance when FM-based control is at work, a win-win situation.

9.7 Future Directions and Open Research Questions

To summarize the open challenges of FMs as highlighted by the research above, recurring limitations include high inference latency, sensitivity to prompt phrasing, and the risk of hallucinations. These issues raise concerns around robustness, reproducibility, and ultimately safety when deploying FMs in real-world robotic systems. As a result, there is a clear need for standardized benchmarks to evaluate and ensure the reliable integration of FMs into robotic applications. Kawaharazuka *et al.* [289] emphasize in their survey that as tasks executed through language instructions become more prevalent, quantitatively evaluating performance becomes increasingly difficult. They further note that FMs still struggle with the fine-grained motion skills required for robotic behavior in dynamic environments, motivating hybrid frameworks with established controllers. Moreover, several sensing modalities, such as depth data, force feedback, and inertial measurements, remain underexplored in FM-based approaches. On the one hand, collecting high-quality data for these modalities poses a significant challenge, as they are not as readily available online as the vision and language data typically used to train FMs. On the other hand, this type of data has a large potential for robotics, particularly in light of the currently observed performance on other training data domains. As a result, FM-driven robots are still rarely deployed in real household or outdoor environments, and current evaluations often rely on simplified or toy problem settings. However, this is likely to change in the coming years as ongoing research continues to address these challenges. So with regard to FMs for HRI and robot personalization, new frameworks should be validated in user studies to complement objective performance metrics with subjective evaluations from the users' perspective.

10 Outlook

The outlook section begins by building on the preceding discussion on foundation models and situates the approaches developed in this thesis within that broader context in Section 10.1. Subsequently, Section 10.2 outlines potential research directions, motivated both by the limitations of current methods and the opportunities arising from FM-based systems.

10.1 Comparison with Foundation Model-Based Approaches

So how do the approaches of this thesis compare against the novel FM architectures?

Unlike natural language feedback for FMs, the virtual reality demonstration and feedback interfaces presented in Chapters 2, 3, and 7 offer direct, spatially grounded preference expression. While verbalized feedback may be faster and more intuitive for users to express, it is inherently descriptive and indirect, requiring interpretation by the agent before behavioral adaptation can occur. For example, instructing a robot to navigate around a lounge area via language involves specifying trajectory shapes and obstacle distances, which can introduce ambiguity. In contrast, a demonstration in VR provides precise spatially grounded motion guidance with minimal room for misinterpretation.

When we compare the DRL-based (personalized) navigation controllers (Chapter 2, 3, 4, and 5) to the vision-language navigation (VLN) approaches above, a few differences stand out: Most prominently, we focus on local human-aware obstacle avoidance, while most VLN approaches pursue goal-object navigation tasks in static environments. Our approaches rely on 2D lidar, enabling accurate obstacle perception for the DRL policy, which is running with low latency. As the name suggests, VLN approaches mostly navigate via RGB image perception at high latencies. This gives the RL-driven controllers an advantage in dynamic human environments, e.g., for more reliable collision avoidance.

The challenge of zero-shot behavior adaptation to changing user preferences is addressed by our MORL approach in Chapter 5. Specifically, we achieve continuous and precise behavior control over the policy via a preference vector. In contrast, when relying on linguistic interfaces to convey preferences to a navigation policy, the resulting behavior alignment may be highly dependent on the FM's interpretation of the feedback, giving the user less control to fine-tune the behavior.

Comparing our immersive XAI interface presented in Chapter 8 against the presented FM approaches for XAI, we are not limited by the latencies of FMs for explanation generation that other studies report. While FMs may be capable of responding to XAI-related queries by users more generally, a specifically designed XAI visualization like ours may be of higher value to users in certain context, also from an educational

perspective.

Now, the limitations of our approaches in context of FM architectures will be outlined. While personalized FMs can generalize user preferences to novel environments and offer interpretable reasoning for such generalization, our learning frameworks, specifically, the RL+BL pipelines from Chapters 2 and 3, and the RLHF-based alignment approach in Chapter 7, distill preferences in a more abstract way. This abstraction during learning complicates the interpretability of the learned behaviors and limits the transparency of preference alignment. The advantage of generalization of FMs also benefits the scalability of preferences, as these two properties are tightly connected. For instance, a small number of interactions with the user are sufficient to internalize their preference and generalize it to similar scenarios, effectively enabling user modeling from sparse data. While the behavior adaptation in Chapter 5 is limited to the pre-defined preference space via the objectives, FMs likely do not face similar limitations.

10.2 Future Directions

Based on the findings of this thesis, future research should address several directions to improve personalization and, consequently, the user's interaction experience with robots. First, since high-quality user interactions, whether through demonstration, comparison, or language, are limited, effective user modeling from sparse verbal and nonverbal interactions should be further investigated. Second, robots operating in long-term human interaction settings should efficiently adapt to evolving user preferences, potentially involving active user querying. Third, future work should explore how the benefits of our approaches can be integrated with those of FMs for improved personalization. For instance, given the latency limitations of FMs remain, hybrid systems that combine the reasoning abilities of FMs with the low-latency control of conventional policies represent a promising direction, similar to [100]. Ultimately, from the end user's perspective, a unified system architecture capable of preference learning from different interaction modalities, handling task instruction, and providing behavior explanation is desirable. In this context, FMs hold promise as social mediators for intuitive, efficient, and human-centered robot interaction through personalization.

VR interfaces have proven effective for collecting spatially grounded user preferences in a safe environment, likely due to their spatial immersion and high visual fidelity. The next logical step is the advancement to augmented reality (AR) interfaces, as AR technology becomes available to a growing user base (e.g., Apple Vision Pro [290], Meta Orion Glasses concept [291]). AR technology can visually overlay robot internal states such as map representation, explanations, and potentially also preference-related features directly onto the user's household environment, where interaction with the robot takes place [292]. For instance, demonstration trajectories drawn at the user's fingertip into the real robot environment have an advantage, they are spatially grounded within the correct environment. This further reduces the domain gap between demonstration

and policy execution. Some works have already leveraged the AR capabilities of non-wearable devices, such as smartphones and tablets, to collect robot navigation preferences [98], and to enable data collection in the absence of a physical robot [293].

Generally, AR technology also offers great potential for the efficient creation of better training datasets for robot policies [294], [295], [296]. For example, a user can wear AR glasses while naturally performing the task of interest. Simultaneously, the AR glasses record first-person RGB-D video of the task execution and track the user's hand movements. For robots equipped with a head-mounted camera and two articulated manipulators capable of mimicking human hand motions, this perspective helps to reduce the disparity between human and robot viewpoints in training data.

This thesis advances user-driven robot personalization through controlled, spatially grounded data collection, and real-time deployment. This introduces a level of control precision not yet achieved by current FM approaches. These contributions provide a strong basis for future work that integrates foundation models and AR technology, enabling adaptable yet precise preference-aligned robot control in real-world human environments.

11 Conclusion

11.1 Summary

This thesis comprises several approaches that share a common goal: Enabling preference-reflecting, learning-based human-aware robot navigation. This common goal is approached from different directions, reflecting different methodological challenges associated with achieving this goal. Our focus lies on how human preferences about a navigating robot can be collected, how to distill these preferences into a preference-reflecting robot policy, how this policy best observes its environment, and finally how the navigation policy can remain adaptable to evolving user preferences as well as how transparent decision-making can be ensured to enhance human-robot interaction. To tackle these challenges, our approaches leverage, combine, and refine various core methodologies such as virtual reality (VR), reinforcement learning (RL), behavioral cloning (BC), inverse RL, RL from human feedback (RLHF), and explainable AI (XAI). Throughout the projects, real-robot deployments demonstrate the robustness of our work.

11.2 Key Findings by Research Question

11.2.1 RQ1: Efficient Preference Collection

How can human navigation preferences be efficiently collected and encoded into robot learning systems? This question was addressed through the design and evaluation of intuitive demonstration and feedback interfaces, with a focus on maximizing user-friendliness and data efficiency. Chapter 2 introduced an immersive virtual reality (VR) interface enabling non-expert users to demonstrate personalized navigation trajectories, paired with a learning framework. Chapter 3 extended this setup to dynamic household environments, leveraging depth vision for preference anchoring. Chapter 6 proposed the EnQuery method for query generation in RLHF settings to improve the information gain with limited interactions. Finally, Chapter 7 systematically compared preference expression and user experience via VR versus 2D interfaces.

In summary, we found that drawing trajectories in VR in a spatially grounded manner is an intuitive and comfortable way for non-expert users to express their preferences. When using our hybrid learning framework (RL+BC), few demonstrations suffice to learn a personalized navigation controller that outperforms traditional methods in perceived comfort, closeness, and preference reflection. The hybrid learning framework also extends into dynamic environments, as quantified by a novel personalization metric. When aligning preferences with an RLHF setup, our ensemble approach EnQuery achieves higher preference reflection in low-query regimes. Regarding the interface used for preference collection, users consistently favored the VR perspective over

a 2D video interface. Also, there is disagreement between preferences expressed with different interface modalities, making personalization results sensitive to the interface choice.

11.2.2 RQ2: Preference vs. Task Balancing

How can navigation policies balance user-specific preferences with task completion objectives? This question was addressed through the implementation of different learning frameworks that balance preference reflection and task completion in a quantifiable manner. Chapter 2 and 3 apply a demonstration-infused hybrid learning framework (RL+BC) to the problem. Chapter 5 systematically extends a MORL framework to trade off between preferences and other objectives in an adaptable manner.

In summary, we found that the RL+BC framework successfully internalizes user preferences, does not lose track of the navigation task, and successfully trades off between personalization and efficiency. It reflects user preferences wherever applicable and generalizes to other navigation scenarios in a goal-oriented manner. Our novel preference reflection trajectory metric quantifies personalization, and when the robot deviates from demonstrated patterns. Finally, the MORL approach to personalization allows for a fine-grained trade-off between preference reflection and other navigation objectives, parameterized through a structured preference vector.

11.2.3 RQ3: Sensor Representations for Navigation

What sensor representations enable robust RL-based navigation in dynamic indoor environments populated by humans, also for preference reflection? To tackle this problem within the scope of our RL-based navigation frameworks, Chapter 3 systematically employs a depth vision-based state representation in multiple variants. Chapter 4 introduces a 2D lidar state representation for dynamic indoor environments with multiple moving humans.

In summary, compression of the depth vision using a VAE enables preference anchoring for navigation in dynamic indoor environments. Using a lidar sensor, our novel TAGD state representation with spatiotemporal attention outperforms state-of-the-art baselines and improves the generalization performance in human-populated dynamic indoor environments.

11.2.4 RQ4: Adaptability and Transparency

How can personalized navigation policies remain adaptable to changing user preferences and provide transparent decision-making to users? This question targets the post-training stage of policy deployment and has been tackled in a two-faceted way: The MORL approach presented in Chapter 5 utilizes a dynamic preference vector to provide control over the policy behavior to adapt to changing user preferences. We found that by systematically integrating demonstrations as one of the MORL objectives, the approach enables fine-grained tuning of demonstration-reflection itself, an advantage

over the aforementioned RL+BC framework. Users can effectively fine-tune the navigation controller after deployment. Finally, Chapter 8 communicates policy reasoning and robot perception to non-expert users through a VR interface. By grounding the XAI insights visually in the semantics of the scenes, the users' objective understanding and subjective awareness of the robot's decision-making significantly enhances.

11.3 Impact and Broader Implications

We believe that our findings contribute to the evolving field of user-centric robotics and provide methodological and conceptual impulses for designing socially appropriate, personalized robot. Our work demonstrates that it is feasible to move beyond static, rule-based models of human-aware navigation toward data-driven, user-aligned policies that are responsive to individual preferences. By leveraging user-intuitive demonstrations, preference-infused learning frameworks, and efficient sensor representations, this thesis contributes novel strategies to address the challenge of robot personalization. Our findings show that human-in-the-loop learning can be enjoyable and intuitive, even for non-expert users. However, the user experience, the expressed preferences, and the resulting policy alignments were found to vary with the interface modality, indicating that interface choice and design are critical components of the personalization pipeline. Finally, for learning-based policies, one of the primary challenges is the lack of transparency in their decision-making processes. We bridge the gap between black-box policies and user interpretability with an immersive VR environment, enhancing non-expert users' understanding of the robot with the goal of meaningful long-term human–robot interaction. Taken together, this thesis proposes and validates structured frameworks for learning and deploying personalized robot navigation, moving us closer to seamless human–robot coexistence in daily life.

References

[1] P. Asgharian, A. M. Panchea, and F. Ferland, "A Review on the Use of Mobile Service Robots in Elderly Care," *Robotics*, vol. 11, no. 6, p. 127, Dec. 2022. doi: 10.3390/robotics11060127.

[2] E. Schneiders, A. M. Kanstrup, J. Kjeldskov, and M. B. Skov, "Domestic Robots and the Dream of Automation: Understanding Human Interaction and Intervention," in *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, ACM, May 2021, pp. 1–13. doi: 10.1145/3411764.3445629.

[3] C. Esterwood, K. Essenmacher, H. Yang, F. Zeng, and L. P. Robert, "A Personable Robot: Meta-Analysis of Robot Personality and Human Acceptance," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 6918–6925, Jul. 2022. doi: 10.1109/LRA.2022.3178795.

[4] N. Gasteiger, M. Hellou, and H. S. Ahn, "Factors for Personalization and Localization to Optimize Human–Robot Interaction: A Literature Review," *International Journal of Social Robotics*, Aug. 2021. doi: 10.1007/s12369-021-00811-8.

[5] J. Yang, C. Vindolet, J. R. G. Olvera, and G. Cheng, "On the impact of robot personalization on human-robot interaction: A review," *arXiv:2401.11776* [*cs*], Jan. 2024. doi: 10.48550/arXiv.2401.11776.

[6] J. Qin, S. Ban, W. Zhu, Y. Wang, and D. Samaras, "Learning Human-Aware Robot Policies for Adaptive Assistance," *arXiv:2412.11913* [*cs*], Dec. 2024. doi: 10.48550/arXiv.2412.11913.

[7] N. Abdulazeem and Y. Hu, "Human Factors Considerations for Quantifiable Human States in Physical Human–Robot Interaction: A Literature Review," *Sensors*, vol. 23, no. 17, p. 7381, Jan. 2023. doi: 10.3390/s23177381.

[8] J. K. Burgoon, L. A. Stern, and L. Dillman, *Interpersonal Adaptation: Dyadic Interaction Patterns*. Cambridge University Press, 1995.

[9] H. Bekkering, E. R. A. De Bruijn, R. H. Cuijpers, R. Newman-Norlund, H. T. Van Schie, and R. Meulenbroek, "Joint Action: Neurocognitive Mechanisms Supporting Human Interaction," *Topics in Cognitive Science*, vol. 1, no. 2, pp. 340–352, Apr. 2009. doi: 10.1111/j.1756-8765.2009.01023.x.

[10] M. B. Mirza, M. Cullen, T. Parr, S. Shergill, and R. J. Moran, "Contextual perception under active inference," *Scientific Reports*, vol. 11, p. 16 223, Aug. 2021. doi: 10.1038/s41598-021-95510-9.

[11] S. Rossi, F. Ferland, and A. Tapus, "User profiling and behavioral adaptation for HRI: A survey," *Pattern Recognition Letters*, User Profiling and Behavior Adaptation for Human-Robot Interaction, vol. 99, pp. 3–12, Nov. 2017. doi: 10.1016/j.patrec.2017.06.002.

[12] N. Dennler, Z. Shi, S. Nikolaidis, and M. Matarić, "Improving User Experience in Preference-Based Optimization of Reward Functions for Assistive Robots," *arXiv:2411.11182*, Nov. 2024. doi: 10.48550/arXiv.2411.11182.

[13] K. Dautenhahn, B. Ogden, and T. Quick, "From embodied to socially embedded agents–implications for interaction-aware robots," *Cognitive Systems Research*, vol. 3, no. 3, pp. 397–428, 2002. doi: 10.1016/S1389-0417(02)00050-5.

[14] D. S. Syrdal, K. Lee Koay, M. L. Walters, and K. Dautenhahn, "A personalized robot companion? - The role of individual differences on spatial preferences in HRI scenarios," in *RO-MAN 2007 - The 16th IEEE International Symposium on Robot and Human Interactive Communication*, Aug. 2007, pp. 1143–1148. doi: 10.1109/ROMAN.2007.4415252.

[15] M. Kollmitz, T. Koller, J. Boedecker, and W. Burgard, "Learning Human-Aware Robot Navigation from Physical Interaction via Inverse Reinforcement Learning," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, Oct. 2020, pp. 11 025–11 031. doi: 10.1109/IROS45743.2020.9340865.

[16] Y. Zhou, J. Vroon, and G. Kortuem, "Exploring Human Preferences for Adapting Inappropriate Robot Navigation Behaviors: A Mixed-Methods Study," *IEEE Robotics and Automation Letters*, pp. 1–8, 2024. doi: 10.1109/LRA.2024.3498432.

[17] B. Irfan, A. Ramachandran, S. Spaulding, D. F. Glas, I. Leite, and K. L. Koay, "Personalization in Long-Term Human-Robot Interaction," in *2019 14th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, IEEE, Mar. 2019, pp. 685–686. doi: 10.1109/HRI.2019.8673076.

[18] M. E. Ligthart, M. A. Neerincx, and K. V. Hindriks, "Memory-based personalization for fostering a long-term child-robot relationship," in *2022 17th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, IEEE, 2022, pp. 80–89. doi: 10.1109/HRI53351.2022.9889446.

[19] A. Bacchin, G. Beraldo, J. Miura, and E. Menegatti, "Preference-Based People-Aware Navigation for Telepresence Robots," *International Journal of Social Robotics*, Apr. 2024.
doi: 10.1007/s12369-024-01131-3.

[20] K. S. Sikand, S. Rabiee, A. Uccello, X. Xiao, G. Warnell, and J. Biswas, "Visual representation learning for preference-aware path planning," in *2022 International Conference on Robotics and Automation (ICRA)*, IEEE, 2022, pp. 11 303–11 309.
doi: 10.1109/ICRA46639.2022.9811828.

[21] J. S. Holtrop, L. D. Scherer, D. D. Matlock, R. E. Glasgow, and L. A. Green, "The Importance of Mental Models in Implementation Science," *Frontiers in Public Health*, vol. 9, p. 680 316, Jul. 2021.
doi: 10.3389/fpubh.2021.680316.

[22] G. Hoffman, T. Bhattacharjee, and S. Nikolaidis, "Inferring Human Intent and Predicting Human Action in Human–Robot Collaboration," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 7, no. 1, pp. 73–95, Jul. 2024.
doi: 10.1146/annurev-control-071223-105834.

[23] B. Holman, A. Anwar, A. Singh, M. Tec, J. Hart, and P. Stone, "Watch where you're going! gaze and head orientation as predictors for social robot navigation," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2021, pp. 3553–3559.
doi: 10.1109/ICRA48506.2021.9561286.

[24] B. Ibarz, J. Leike, T. Pohlen, G. Irving, S. Legg, and D. Amodei, "Reward learning from human preferences and demonstrations in Atari," in *Advances in Neural Information Processing Systems*, vol. 31, Curran Associates, Inc., 2018.

[25] M. Palan, N. C. Landolfi, G. Shevchuk, and D. Sadigh, "Learning Reward Functions by Integrating Human Demonstrations and Preferences," *arXiv:1906.08928 [cs]*, Jun. 2019.
doi: 10.48550/arXiv.1906.08928.

[26] E. Biyik, N. Huynh, M. Kochenderfer, and D. Sadigh, "Active Preference-Based Gaussian Process Regression for Reward Learning," in *Robotics: Science and Systems XVI*, Robotics: Science and Systems Foundation, Jul. 2020.
doi: 10.15607/RSS.2020.XVI.041.

[27] S. A. Mehta and D. P. Losey, "Unified Learning from Demonstrations, Corrections, and Preferences during Physical Human-Robot Interaction," *arXiv:2207.03395 [cs]*, Jul. 2022.
doi: 10.48550/arXiv.2207.03395.

[28] D. Han, T. McInroe, A. Jelley, S. V. Albrecht, P. Bell, and A. Storkey, "LLM-Personalize: Aligning LLM Planners with Human Preferences via Reinforced Self-Training for Housekeeping Robots," in *Proceedings of the 31st International Conference on Computational Linguistics*, O. Rambow, L. Wanner, M. Apidianaki, H. Al-Khalifa, B. D. Eugenio, and S. Schockaert, Eds., Association for Computational Linguistics, Jan. 2025, pp. 1465–1474.

[29] P. Sharma, B. Sundaralingam, V. Blukis, C. Paxton, T. Hermans, A. Torralba, J. Andreas, and D. Fox, "Correcting Robot Plans with Natural Language Feedback," *arXiv:2204.05186 [cs]*, Apr. 2022.
doi: 10.48550/arXiv.2204.05186.

[30] M. Hwang, L. Weihs, C. Park, K. Lee, A. Kembhavi, and K. Ehsani, "Promptable Behaviors: Personalizing Multi-Objective Rewards from Human Preferences," in *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Jun. 2024, pp. 16 216–16 226.
doi: 10.1109/CVPR52733.2024.01535.

[31] A. Elfes, "Occupancy Grids: A Stochastic Spatial Representation for Active Robot Perception," *arXiv:1304.1098 [cs]*, Mar. 2013.
doi: 10.48550/arXiv.1304.1098.

[32] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968.
doi: 10.1109/TSSC.1968.300136.

[33] S. LaValle, "Rapidly-exploring random trees: A new tool for path planning," *Research Report 9811*, 1998.

[34] S. M. LaValle and J. J. Kuffner, "Randomized Kinodynamic Planning," *The International Journal of Robotics Research*, vol. 20, no. 5, pp. 378–400, May 2001.
doi: 10.1177/02783640122067453.

[35] F. Dellaert, D. Fox, W. Burgard, and S. Thrun, "Monte Carlo localization for mobile robots," in *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No.99CH36288C)*, vol. 2, May 1999, 1322–1328 vol.2.
doi: 10.1109/ROBOT.1999.772544.

[36] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Robotics Automation Magazine*, vol. 4, no. 1, pp. 23–33, Mar. 1997.
doi: 10.1109/100.580977.

[37] O. Khatib, "Real-Time Obstacle Avoidance for Manipulators and Mobile Robots," *The International Journal of Robotics Research*, vol. 5, no. 1, pp. 90–98, Mar. 1986.
doi: 10.1177/027836498600500106.

[38] S. Ge and Y. Cui, "Dynamic Motion Planning for Mobile Robots Using Potential Field Method," *Autonomous Robots*, vol. 13, no. 3, pp. 207–222, Nov. 2002. doi: 10.1023/A:1020564024509.

[39] C. Mavrogiannis, F. Baldini, A. Wang, D. Zhao, P. Trautman, A. Steinfeld, and J. Oh, "Core Challenges of Social Robot Navigation: A Survey," *ACM Transactions on Human-Robot Interaction*, vol. 12, Feb. 2023. doi: 10.1145/3583741.

[40] R. Möller, A. Furnari, S. Battiato, A. Härmä, and G. M. Farinella, "A survey on human-aware robot navigation," *Robotics and Autonomous Systems*, vol. 145, p. 103 837, Nov. 2021. doi: 10.1016/j.robot.2021.103837.

[41] D. Helbing and P. Molnár, "Social force model for pedestrian dynamics," *Physical Review E*, vol. 51, no. 5, pp. 4282–4286, May 1995. doi: 10.1103/PhysRevE.51.4282.

[42] M. Kollmitz, K. Hsiao, J. Gaa, and W. Burgard, "Time dependent planning on a layered social cost map for human-aware robot navigation," in *2015 European Conference on Mobile Robots (ECMR)*, IEEE, Sep. 2015, pp. 1–6. doi: 10.1109/ECMR.2015.7324184.

[43] P. Fiorini and Z. Shiller, "Motion Planning in Dynamic Environments Using Velocity Obstacles," *The International Journal of Robotics Research*, vol. 17, no. 7, pp. 760–772, Jul. 1998. doi: 10.1177/027836499801700706.

[44] F. Vesentini, R. Muradore, and P. Fiorini, "A survey on Velocity Obstacle paradigm," *Robotics and Autonomous Systems*, vol. 174, p. 104 645, Apr. 2024. doi: 10.1016/j.robot.2024.104645.

[45] J. van den Berg, M. Lin, and D. Manocha, "Reciprocal Velocity Obstacles for real-time multi-agent navigation," in *2008 IEEE International Conference on Robotics and Automation*, May 2008, pp. 1928–1935. doi: 10.1109/ROBOT.2008.4543489.

[46] S. Quinlan and O. Khatib, "Elastic bands: Connecting path planning and control," in [*1993*] *Proceedings IEEE International Conference on Robotics and Automation*, IEEE, 1993, pp. 802–807.

[47] G. Pérez, N. Zapata-Cornejo, P. Bustos, and P. Núñez, "Social Elastic Band with Prediction and Anticipation: Enhancing Real-Time Path Trajectory Optimization for Socially Aware Robot Navigation," *International Journal of Social Robotics*, Apr. 2024. doi: 10.1007/s12369-024-01135-z.

[48] N. Hirose, D. Shah, A. Sridhar, and S. Levine, "SACSoN: Scalable Autonomous Control for Social Navigation," *IEEE Robotics and Automation Letters*, pp. 1–8, 2023.
doi: 10.1109/LRA.2023.3329626.

[49] X. Xiao, B. Liu, G. Warnell, and P. Stone, "Motion planning and control for mobile robot navigation using machine learning: A survey," *Autonomous Robots*, vol. 46, no. 5, pp. 569–597, Jun. 2022.
doi: 10.1007/s10514-022-10039-8.

[50] Z. Xie and P. Dames, "DRL-VO: Learning to Navigate Through Crowded Dynamic Scenes Using Velocity Obstacles," *IEEE Transactions on Robotics*, vol. 39, no. 4, pp. 2700–2719, Aug. 2023.
doi: 10.1109/TRO.2023.3257549.

[51] B. Singh, R. Kumar, and V. P. Singh, "Reinforcement learning in robotic applications: A comprehensive survey," *Artificial Intelligence Review*, vol. 55, no. 2, pp. 945–990, Feb. 2022.
doi: 10.1007/s10462-021-09997-9.

[52] H. Ravichandar, A. S. Polydoros, S. Chernova, and A. Billard, "Recent Advances in Robot Learning from Demonstration," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 3, no. 1, pp. 297–330, May 2020.
doi: 10.1146/annurev-control-100819-063206.

[53] M. Vecerik, T. Hester, J. Scholz, F. Wang, O. Pietquin, B. Piot, N. Heess, T. Rothörl, T. Lampe, and M. Riedmiller, "Leveraging Demonstrations for Deep Reinforcement Learning on Robotics Problems with Sparse Rewards," *arXiv:1707.08817 [cs]*, Oct. 2018.
doi: 10.48550/arXiv.1707.08817.

[54] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, "Deep Reinforcement Learning: A Brief Survey," *IEEE Signal Processing Magazine*, vol. 34, no. 6, pp. 26–38, Nov. 2017.
doi: 10.1109/MSP.2017.2743240.

[55] Y. Duan, X. Chen, R. Houthooft, J. Schulman, and P. Abbeel, "Benchmarking Deep Reinforcement Learning for Continuous Control," in *Proceedings of The 33rd International Conference on Machine Learning*, PMLR, Jun. 2016, pp. 1329–1338.

[56] L. Tai, G. Paolo, and M. Liu, "Virtual-to-real deep reinforcement learning: Continuous control of mobile robots for mapless navigation," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, Sep. 2017, pp. 31–36.
doi: 10.1109/IROS.2017.8202134.

[57] P. F. Christiano, J. Leike, T. B. Brown, M. Martic, S. Legg, and D. Amodei, "Deep reinforcement learning from human preferences," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, ser. NIPS'17, Curran Associates Inc., Dec. 2017, pp. 4302–4310.

[58] C. Wirth, R. Akrour, G. Neumann, and J. Fürnkranz, "A Survey of Preference-Based Reinforcement Learning Methods," *Journal of Machine Learning Research*, vol. 18, no. 136, pp. 1–46, 2017.
doi: 10.5445/IR/1000118270.

[59] S. Arora and P. Doshi, "A survey of inverse reinforcement learning: Challenges, methods and progress," *Artificial Intelligence*, vol. 297, p. 103 500, Aug. 2021.
doi: 10.1016/j.artint.2021.103500.

[60] T. Kaufmann, P. Weng, V. Bengs, and E. Hüllermeier, "A Survey of Reinforcement Learning from Human Feedback," *arXiv:2312.14925* [*cs*], Dec. 2023.
doi: 10.48550/arXiv.2312.14925.

[61] V. G. Goecks, G. M. Gremillion, V. J. Lawhern, J. Valasek, and N. R. Waytowich, "Efficiently combining human demonstrations and interventions for safe training of autonomous systems in real-time," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 2462–2470.
doi: 10.1609/aaai.v33i01.33012462.

[62] S. Cabi, S. G. Colmenarejo, A. Novikov, K. Konyushkova, S. Reed, R. Jeong, K. Zolna, Y. Aytar, D. Budden, M. Vecerik, O. Sushkov, D. Barker, J. Scholz, M. Denil, N. de Freitas, and Z. Wang, "Scaling data-driven robotics with reward sketching and batch reinforcement learning," *arXiv:1909.12200* [*cs*], Jun. 2020.
doi: 10.48550/arXiv.1909.12200.

[63] C. Li, F. Xia, R. Martín-Martín, M. Lingelbach, S. Srivastava, B. Shen, K. E. Vainio, C. Gokmen, G. Dharan, T. Jain, A. Kurenkov, K. Liu, H. Gweon, J. Wu, L. Fei-Fei, and S. Savarese, "iGibson 2.0: Object-Centric Simulation for Robot Learning of Everyday Household Tasks," in *Proceedings of the 5th Conference on Robot Learning*, PMLR, Jan. 2022, pp. 455–465.

[64] J. Liang, V. Makoviychuk, A. Handa, N. Chentanez, M. Macklin, and D. Fox, "Gpu-accelerated robotic simulation for distributed reinforcement learning," in *Conference on Robot Learning*, PMLR, 2018, pp. 270–282.

[65] M. Wonsick and T. Padir, "A Systematic Review of Virtual Reality Interfaces for Controlling and Interacting with Robots," *Applied Sciences*, vol. 10, no. 24, p. 9051, Jan. 2020.
doi: 10.3390/app10249051.

[66] M. K. Wozniak, R. Stower, P. Jensfelt, and A. Pereira, "Happily Error After: Framework Development and User Study for Correcting Robot Perception Errors in Virtual Reality," in *2023 32nd IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, Aug. 2023, pp. 1573–1580.
doi: 10.1109/RO-MAN57019.2023.10309446.

[67] E. Ellis, G. R. Ghosal, S. J. Russell, A. Dragan, and E. Bıyık, "A generalized acquisition function for preference-based reward learning," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2024, pp. 2814–2821.
doi: 10.1109/ICRA57147.2024.10611472.

[68] D. Marta, S. Holk, C. Pek, J. Tumova, and I. Leite, "VARIQuery: VAE Segment-Based Active Learning for Query Selection in Preference-Based Reinforcement Learning," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2023, pp. 7878–7885.
doi: 10.1109/IROS55552.2023.10341795.

[69] D. Marta, S. Holk, C. Pek, J. Tumova, and I. Leite, "Aligning Human Preferences with Baseline Objectives in Reinforcement Learning," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, May 2023, pp. 7562–7568.
doi: 10.1109/ICRA48891.2023.10161261.

[70] J. Liang, U. Patel, A. J. Sathyamoorthy, and D. Manocha, "Crowd-Steer: Real-time Smooth and Collision-Free Robot Navigation in Densely Crowded Scenarios Trained using High-Fidelity Simulation," in *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, International Joint Conferences on Artificial Intelligence Organization, Jan. 2021, pp. 4221–4228.
doi: 10.24963/ijcai.2020/583.

[71] K. Baraka and M. Veloso, "Adaptive Interaction of Persistent Robots to User Temporal Preferences," in *Social Robotics*, A. Tapus, E. André, J.-C. Martin, F. Ferland, and M. Ammi, Eds., vol. 9388, Springer International Publishing, 2015, pp. 61–71.
doi: 10.1007/978-3-319-25554-5_7.

[72] A. Suresh, A. Taylor, L. D. Riek, and S. Martínez, "Robot Navigation in Risky, Crowded Environments: Understanding Human Preferences," *IEEE Robotics and Automation Letters*, vol. 8, no. 9, pp. 5632–5639, 2023.
doi: 10.1109/LRA.2023.3290533.

[73] G. Angelopoulos, L. Mangiacapra, A. Rossi, C. Di Napoli, and S. Rossi, "What is behind the curtain? Increasing transparency in reinforcement learning with human preferences and explanations," *Engineering Applications of Artificial Intelligence*, vol. 149, p. 110 520, Jun. 2025.
doi: 10.1016/j.engappai.2025.110520.

[74] J. de Heuvel, N. Corral, L. Bruckschen, and M. Bennewitz, "Learning Personalized Human-Aware Robot Navigation Using Virtual Reality Demonstrations from a User Study," in *2022 31th IEEE International Conference on Robot Human Interactive Communication (RO-MAN)*, 2022, pp. 898–905.
doi: 10.1109/RO-MAN53752.2022.9900554.

[75] J. de Heuvel, N. Corral, B. Kreis, J. Conradi, A. Driemel, and M. Bennewitz, "Learning Depth Vision-Based Personalized Robot Navigation From Dynamic Demonstrations in Virtual Reality," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2023, pp. 6757–6764.
doi: 10.1109/IROS55552.2023.10341370.

[76] J. de Heuvel, X. Zeng, W. Shi, T. Sethuraman, and M. Bennewitz, "Spatiotemporal Attention Enhances Lidar-Based Robot Navigation in Dynamic Environments," *IEEE Robotics and Automation Letters*, vol. 9, no. 5, pp. 4202–4209, May 2024.
doi: 10.1109/LRA.2024.3373988.

[77] J. de Heuvel, T. Sethuraman, and M. Bennewitz, "Demonstration-Enhanced Adaptable Multi-Objective Robot Navigation," in *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2025, pp. 16 877–16 884.
doi: 10.1109/IROS60139.2025.11246392.

[78] J. de Heuvel, F. Seiler, and M. Bennewitz, "EnQuery: Ensemble Policies for Diverse Query-Generation in Preference Alignment of Robot Navigation," in *2024 33rd IEEE International Conference on Robot and Human Interactive Communication (ROMAN)*, Aug. 2024, pp. 303–310.
doi: 10.1109/RO-MAN60168.2024.10731470.

[79] J. de Heuvel, D. Marta, S. Holk, I. Leite, and M. Bennewitz, "The Impact of VR and 2D Interfaces on Human Feedback in Preference-Based Robot Learning," in *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2025, pp. 19 024–19 030.
doi: 10.1109/IROS60139.2025.11246802.

[80] J. de Heuvel, S. Müller, M. Wessels, A. Akhtar, C. Bauckhage, and M. Bennewitz, "Immersive Explainability: Visualizing Robot Navigation Decisions through XAI Semantic Scene Projections in Virtual Reality," in *2025 34th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, Aug. 2025, pp. 1633–1639.
doi: 10.1109/RO-MAN63969.2025.11217609.

[81] T. Kruse, A. K. Pandey, R. Alami, and A. Kirsch, "Human-aware robot navigation: A survey," *Robotics and Autonomous Systems*, vol. 61, no. 12, pp. 1726–1743, Dec. 2013.
doi: 10.1016/j.robot.2013.05.007.

[82] A. Y. Ng, D. Harada, and S. J. Russell, "Policy Invariance Under Reward Transformations: Theory and Application to Reward Shaping," in *Proceedings of the Sixteenth International Conference on Machine Learning*, ser. ICML '99, Morgan Kaufmann Publishers Inc., Jun. 1999, pp. 278–287.

[83] M. Pfeiffer, S. Shukla, M. Turchetta, C. Cadena, A. Krause, R. Siegwart, and J. Nieto, "Reinforced Imitation: Sample Efficient Deep Reinforcement Learning for Mapless Navigation by Leveraging Prior Demonstrations," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 4423–4430, Oct. 2018.
doi: 10.1109/LRA.2018.2869644.

[84] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," *Robotics and Autonomous Systems*, vol. 57, no. 5, pp. 469–483, May 2009.
doi: 10.1016/j.robot.2008.10.024.

[85] A. L. Thomaz and C. Breazeal, "Reinforcement learning with human teachers: Evidence of feedback and guidance with implications for learning performance," in *Proceedings of the 21st National Conference on Artificial Intelligence - Volume 1*, ser. AAAI'06, AAAI Press, Jul. 2006, pp. 1000–1005.

[86] M. Hellou, N. Gasteiger, J. Y. Lim, M. Jang, and H. S. Ahn, "Personalization and Localization in Human-Robot Interaction: A Review of Technical Methods," *Robotics*, vol. 10, no. 4, p. 120, Dec. 2021.
doi: 10.3390/robotics10040120.

[87] K. Bungert, L. Bruckschen, S. Krumpen, W. Rau, M. Weinmann, and M. Bennewitz, "Human-Aware Robot Navigation Based on Learned Cost Values from User Studies," in *2021 30th IEEE International Conference on Robot Human Interactive Communication (RO-MAN)*, Aug. 2021, pp. 337–342.
doi: 10.1109/RO-MAN50785.2021.9515481.

[88] N. Pérez-Higueras, F. Caballero, and L. Merino, "Teaching Robot Navigation Behaviors to Optimal RRT Planners," *International Journal of Social Robotics*, vol. 10, no. 2, pp. 235–249, Apr. 2018.
doi: 10.1007/s12369-017-0448-1.

[89] L. Bruckschen, K. Bungert, N. Dengler, and M. Bennewitz, "Human-Aware Robot Navigation by Long-Term Movement Prediction," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, Oct. 2020, pp. 11 032–11 037.
doi: 10.1109/IROS45743.2020.9340776.

[90] X. Xiao, B. Liu, G. Warnell, J. Fink, and P. Stone, "APPLD: Adaptive Planner Parameter Learning From Demonstration," *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 4541–4547, Jul. 2020.
doi: 10.1109/LRA.2020.3002217.

[91] H. Zender, P. Jensfelt, and G.-J. M. Kruijff, "Human-and situation-aware people following," in *RO-MAN 2007-The 16th IEEE International Symposium on Robot and Human Interactive Communication*, IEEE, 2007, pp. 1131–1136.
doi: 10.1109/ROMAN.2007.4415250.

[92] M. Luber, L. Spinello, J. Silva, and K. O. Arras, "Socially-aware robot navigation: A learning approach," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, Oct. 2012, pp. 902–907.
doi: 10.1109/IROS.2012.6385716.

[93] V. Narayanan, B. M. Manoghar, V. Sashank Dorbala, D. Manocha, and A. Bera, "ProxEmo: Gait-based Emotion Learning and Multi-view Proxemic Fusion for Socially-Aware Robot Navigation," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, Oct. 2020, pp. 8200–8207.
doi: 10.1109/IROS45743.2020.9340710.

[94] O. Liu, D. Rakita, B. Mutlu, and M. Gleicher, "Understanding human-robot interaction in virtual reality," in *2017 26th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, Aug. 2017, pp. 751–757.
doi: 10.1109/ROMAN.2017.8172387.

[95] M. Moletta, M. K. Wozniak, M. C. Welle, and D. Kragic, "A virtual reality framework for human-robot collaboration in cloth folding," in *2023 IEEE-RAS 22nd International Conference on Humanoid Robots (Humanoids)*, IEEE, 2023, pp. 1–7.
doi: 10.1109/Humanoids57100.2023.10375184.

[96] Q. Zhang, N. Tsoi, and M. Vázquez, "SEAN-VR: An Immersive Virtual Reality Experience for Evaluating Social Robot Navigation," in *Companion of the 2023 ACM/IEEE International Conference on Human-Robot Interaction*, ser. HRI '23, Association for Computing Machinery, Mar. 2023, pp. 902–904.
doi: 10.1145/3568294.3580039.

[97] S. Nakaoka, Y. Kawasaki, and M. Takahashi, "Learning User-Preferred Robot Navigation Based on Social Force Model from Human Feedback in Virtual Reality Environments," in *2023 32nd IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, Aug. 2023, pp. 1182–1187.
doi: 10.1109/RO-MAN57019.2023.10309609.

[98] M. Nigro, A. O'Connell, T. Groechel, A.-M. Velentza, and M. Matarić, "An Interactive Augmented Reality Interface for Personalized Proxemics Modeling: Comfort and Human–Robot Interactions," *IEEE Robotics & Automation Magazine*, pp. 2–11, 2024.
doi: 10.1109/MRA.2024.3415108.

[99] Y. Dai, R. Peng, S. Li, and J. Chai, "Think, Act, and Ask: Open-World Interactive Personalized Robot Navigation," in *2024 IEEE International Conference on Robotics*

and Automation (ICRA), May 2024, pp. 3296–3303.
doi: 10.1109/ICRA57147.2024.10610178.

[100] D. Martinez-Baselga, O. de Groot, L. Knoedler, J. Alonso-Mora, L. Riazuelo, and L. Montano, "Hey Robot! Personalizing Robot Navigation through Model Predictive Control with a Large Language Model," *arXiv:2409.13393 [cs]*, Sep. 2024.
doi: 10.48550/arXiv.2409.13393.

[101] W. Wang, L. Mao, R. Wang, and B.-C. Min, "SRLM: Human-in-Loop Interactive Social Robot Navigation with Large Language Model and Deep Reinforcement Learning," *arXiv:2403.15648 [cs]*, Mar. 2024.
doi: 10.48550/arXiv.2403.15648.

[102] R. Wang, W. Wang, and B.-C. Min, "Feedback-efficient Active Preference Learning for Socially Aware Robot Navigation," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2022, pp. 11 336–11 343.
doi: 10.1109/IROS47612.2022.9981616.

[103] S. Fujimoto, H. Hoof, and D. Meger, "Addressing Function Approximation Error in Actor-Critic Methods," in *Proceedings of the 35th International Conference on Machine Learning*, PMLR, Jul. 2018, pp. 1587–1596.

[104] A. Nair, B. McGrew, M. Andrychowicz, W. Zaremba, and P. Abbeel, "Overcoming Exploration in Reinforcement Learning with Demonstrations," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, May 2018, pp. 6292–6299.
doi: 10.1109/ICRA.2018.8463162.

[105] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. The MIT Press, Nov. 2018, vol. 1.

[106] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, "Deterministic Policy Gradient Algorithms," in *Proceedings of the 31st International Conference on Machine Learning*, PMLR, Jan. 2014, pp. 387–395.

[107] E. Coumans and Y. Bai, *Pybullet: Physics simulation for games visual effects robotics and reinforcement learning*, 2016 - 2021. [Online]. Available: `http://pybullet.org` Accessed: Jul. 8, 2021.

[108] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "ROS: An open-source Robot Operating System," in *ICRA Workshop on Open Source Software*, vol. 3, Kobe, Japan, 2009, p. 5.

[109] A. Francis, C. Pérez-D'Arpino, C. Li, F. Xia, A. Alahi, R. Alami, A. Bera, A. Biswas, J. Biswas, R. Chandra, H.-T. L. Chiang, M. Everett, S. Ha, J. Hart, J. P. How, H. Karnan, T.-W. E. Lee, L. J. Manso, R. Mirsky, S. Pirk, P. T. Singamaneni, P. Stone, A. V. Taylor, P. Trautman, N. Tsoi, M. Vázquez, X. Xiao, P. Xu, N. Yokoyama, A. Toshev, and R. Martín-Martín, "Principles and Guidelines for Evaluating Social Robot Navigation Algorithms," *ACM Transactions on Human-Robot Interaction*, vol. 14,

no. 2, 34:1–34:65, Feb. 2025.
doi: 10.1145/3700599.

[110] C. Theodoridou, D. Antonopoulos, A. Kargakos, I. Kostavelis, D. Giakoumis, and D. Tzovaras, "Robot Navigation in Human Populated Unknown Environments based on Visual-Laser Sensor Fusion," in *The15th International Conference on PErvasive Technologies Related to Assistive Environments*, ACM, Jun. 2022, pp. 336–342.
doi: 10.1145/3529190.3534740.

[111] M. Laskin, A. Srinivas, and P. Abbeel, "CURL: Contrastive Unsupervised Representations for Reinforcement Learning," in *Proceedings of the 37th International Conference on Machine Learning*, PMLR, Nov. 2020, pp. 5639–5650.

[112] X. Gao, X. Zhao, and M. Tan, "Modeling Socially Normative Navigation Behaviors from Demonstrations with Inverse Reinforcement Learning," in *2019 IEEE 15th International Conference on Automation Science and Engineering (CASE)*, IEEE, Aug. 2019, pp. 1333–1340.
doi: 10.1109/COASE.2019.8843123.

[113] H. Karnan, A. Nair, X. Xiao, G. Warnell, S. Pirk, A. Toshev, J. Hart, J. Biswas, and P. Stone, "Socially CompliAnt Navigation Dataset (SCAND): A Large-Scale Dataset of Demonstrations for Social Navigation," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 11 807–11 814, Oct. 2022.
doi: 10.1109/LRA.2022.3184025.

[114] C. Chen, Y. Liu, S. Kreiss, and A. Alahi, "Crowd-Robot Interaction: Crowd-Aware Robot Navigation With Attention-Based Deep Reinforcement Learning," in *2019 International Conference on Robotics and Automation (ICRA)*, May 2019, pp. 6015–6022.
doi: 10.1109/ICRA.2019.8794134.

[115] L. Tai, J. Zhang, M. Liu, and W. Burgard, "Socially Compliant Navigation Through Raw Depth Inputs with Generative Adversarial Imitation Learning," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, May 2018, pp. 1111–1117.
doi: 10.1109/ICRA.2018.8460968.

[116] V. Tolani, S. Bansal, A. Faust, and C. Tomlin, "Visual Navigation Among Humans With Optimal Control as a Supervisor," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 2288–2295, Apr. 2021.
doi: 10.1109/LRA.2021.3060638.

[117] D. Hoeller, L. Wellhausen, F. Farshidian, and M. Hutter, "Learning a State Representation and Navigation in Cluttered and Dynamic Environments," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 5081–5088, Jul. 2021.
doi: 10.1109/LRA.2021.3068639.

[118] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, "Social LSTM: Human Trajectory Prediction in Crowded Spaces," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Jun. 2016, pp. 961–971.
doi: 10.1109/CVPR.2016.110.

[119] T. Fernando, S. Denman, S. Sridharan, and C. Fookes, "Soft + Hardwired attention: An LSTM framework for human trajectory prediction and abnormal event detection," *Neural Networks*, vol. 108, pp. 466–478, Dec. 2018.
doi: 10.1016/j.neunet.2018.09.002.

[120] C. Xu, W. Liu, J. Wang, L. Ma, F. Yin, and Z. Deng, "DUNE: Sim2Real Transfer for Depth-based Navigation in Unstructured Dynamic Indoor Environments," in *ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Apr. 2025, pp. 1–5.
doi: 10.1109/ICASSP49660.2025.10890168.

[121] Z. Xu, X. Zhan, Y. Xiu, C. Suzuki, and K. Shimada, "Onboard dynamic-object detection and tracking for autonomous robot navigation with rgb-d camera," *IEEE Robotics and Automation Letters*, vol. 9, no. 1, pp. 651–658, 2023.
doi: 10.1109/LRA.2023.3334683.

[122] B. Shen, F. Xia, C. Li, R. Martín-Martín, L. Fan, G. Wang, C. Pérez-D'Arpino, S. Buch, S. Srivastava, L. Tchapmi, M. Tchapmi, K. Vainio, J. Wong, L. Fei-Fei, and S. Savarese, "iGibson 1.0: A Simulation Environment for Interactive Tasks in Large Realistic Scenes," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sep. 2021, pp. 7520–7527.
doi: 10.1109/IROS51168.2021.9636667.

[123] H. Alt and M. Godau, "Computing the fréchet distance between two polygonal curves," *International Journal of Computational Geometry & Applications*, vol. 05, no. 01n02, pp. 75–91, Mar. 1995.
doi: 10.1142/S0218195995000064.

[124] K. D. Katyal, G. D. Hager, and C.-M. Huang, "Intent-aware pedestrian prediction for adaptive crowd navigation," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2020, pp. 3277–3283.
doi: 10.1109/ICRA40945.2020.9197434.

[125] A. Wang, C. Mavrogiannis, and A. Steinfeld, "Group-based Motion Prediction for Navigation in Crowded Environments," in *Proceedings of the 5th Conference on Robot Learning*, PMLR, Jan. 2022, pp. 871–882.

[126] Y. Wang, Q. Mao, H. Zhu, J. Deng, Y. Zhang, J. Ji, H. Li, and Y. Zhang, "Multi-modal 3d object detection in autonomous driving: A survey," *International Journal of Computer Vision*, vol. 131, no. 8, pp. 2122–2152, 2023.
doi: 10.1007/s11263-023-01784-z.

[127] C. Wang, C. Ma, M. Zhu, and X. Yang, "PointAugmenting: Cross-Modal Augmentation for 3D Object Detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 11 794–11 803.
doi: 10.1109/CVPR46437.2021.01162.

[128] J. Li, H. Qin, J. Wang, and J. Li, "Openstreetmap-based autonomous navigation for the four wheel-legged robot via 3d-lidar and ccd camera," *IEEE Transactions on Industrial Electronics*, vol. 69, no. 3, pp. 2708–2717, 2021.
doi: 10.1109/TIE.2021.3070508.

[129] M. Himmelsbach, F. v. Hundelshausen, and H.-J. Wuensche, "Fast segmentation of 3D point clouds for ground vehicles," in *2010 IEEE Intelligent Vehicles Symposium*, Jun. 2010, pp. 560–565.
doi: 10.1109/IVS.2010.5548059.

[130] C. Chen, S. Hu, P. Nikdel, G. Mori, and M. Savva, "Relational Graph Learning for Crowd Navigation," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, Oct. 2020, pp. 10 007–10 013.
doi: 10.1109/IROS45743.2020.9340705.

[131] E. Salvato, G. Fenu, E. Medvet, and F. A. Pellegrino, "Crossing the Reality Gap: A Survey on Sim-to-Real Transferability of Robot Controllers in Reinforcement Learning," *IEEE Access*, vol. 9, pp. 153 171–153 187, 2021.
doi: 10.1109/ACCESS.2021.3126658.

[132] X. Chen, J. Hu, C. Jin, L. Li, and L. Wang, "Understanding Domain Randomization For Sim-To-Real Transfer," in *10th International Conference on Learning Representations, ICLR 2022*, 2022.

[133] B. Qin, Z. J. Chong, S. H. Soh, T. Bandyopadhyay, M. H. Ang, E. Frazzoli, and D. Rus, "A Spatial-Temporal Approach for Moving Object Recognition with 2D LIDAR," in *Experimental Robotics*, M. A. Hsieh, O. Khatib, and V. Kumar, Eds., vol. 109, Springer International Publishing, 2016, pp. 807–820.
doi: 10.1007/978-3-319-23778-7_53.

[134] Y. Song, Y. Tian, G. Wang, and M. Li, "2D LiDAR Map Prediction via Estimating Motion Flow with GRU," in *2019 International Conference on Robotics and Automation (ICRA)*, May 2019, pp. 6617–6623.
doi: 10.1109/ICRA.2019.8793490.

[135] C. Pérez-D'Arpino, C. Liu, P. Goebel, R. Martín-Martín, and S. Savarese, "Robot Navigation in Constrained Pedestrian Environments using Reinforcement Learning," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, May 2021, pp. 1140–1146.
doi: 10.1109/ICRA48506.2021.9560893.

[136] J. de Heuvel, W. Shi, X. Zeng, and M. Bennewitz, "Subgoal-Driven Navigation in Dynamic Environments Using Attention-Based Deep Reinforcement Learning," in *2023 21st International Conference on Advanced Robotics (ICAR)*, Dec. 2023, pp. 79–85.
doi: 10.1109/ICAR58858.2023.10406349.

[137] O. Brock and O. Khatib, "High-speed navigation using the global dynamic window approach," in *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No.99CH36288C)*, vol. 1, May 1999, 341–346 vol.1.
doi: 10.1109/ROBOT.1999.770002.

[138] M. Missura and M. Bennewitz, "Predictive Collision Avoidance for the Dynamic Window Approach," in *2019 International Conference on Robotics and Automation (ICRA)*, May 2019, pp. 8620–8626.
doi: 10.1109/ICRA.2019.8794386.

[139] J. Shabbir and T. Anwer, "A Survey of Deep Learning Techniques for Mobile Robot Applications," *arXiv:1803.07608 [cs]*, Mar. 2018.
doi: 10.48550/arXiv.1803.07608.

[140] H.-T. L. Chiang, A. Faust, M. Fiser, and A. Francis, "Learning navigation behaviors end-to-end with autorl," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 2007–2014, 2019.
doi: 10.1109/LRA.2019.2899918.

[141] A. Faust, K. Oslund, O. Ramirez, A. Francis, L. Tapia, M. Fiser, and J. Davidson, "Prm-rl: Long-range robotic navigation tasks by combining reinforcement learning and sampling-based planning," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2018, pp. 5113–5120.
doi: 10.1109/ICRA.2018.8461096.

[142] Y. Han, I. H. Zhan, W. Zhao, J. Pan, Z. Zhang, Y. Wang, and Y.-J. Liu, "Deep Reinforcement Learning for Robot Collision Avoidance With Self-State-Attention and Sensor Fusion," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 6886–6893, Jul. 2022.
doi: 10.1109/LRA.2022.3178791.

[143] A. Bayoumi and M. Bennewitz, "Learning optimal navigation actions for foresighted robot behavior during assistance tasks," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2016, pp. 207–212.
doi: 10.1109/ICRA.2016.7487135.

[144] W. Zhi, T. Lai, L. Ott, and F. Ramos, "Anticipatory navigation in crowds by probabilistic prediction of pedestrian future movements," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2021, pp. 8459–8464.
doi: 10.1109/ICRA48506.2021.9561022.

[145] A. Pfrunder, P. V. K. Borges, A. R. Romero, G. Catt, and A. Elfes, "Real-time autonomous ground vehicle navigation in heterogeneous environments using a 3D LiDAR," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sep. 2017, pp. 2601–2608.
doi: 10.1109/IROS.2017.8206083.

[146] T. Fan, X. Cheng, J. Pan, P. Long, W. Liu, R. Yang, and D. Manocha, "Getting Robots Unfrozen and Unlost in Dense Pedestrian Crowds," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1178–1185, Apr. 2019.
doi: 10.1109/LRA.2019.2891491.

[147] A. J. Sathyamoorthy, J. Liang, U. Patel, T. Guan, R. Chandra, and D. Manocha, "DenseCAvoid: Real-time Navigation in Dense Crowds using Anticipatory Behaviors," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, May 2020, pp. 11 345–11 352.
doi: 10.1109/ICRA40945.2020.9197379.

[148] J. Jin, N. M. Nguyen, N. Sakib, D. Graves, H. Yao, and M. Jagersand, "Mapless Navigation among Dynamics with Social-safety-awareness: A reinforcement learning approach from 2D laser scans," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, May 2020, pp. 6979–6985.
doi: 10.1109/ICRA40945.2020.9197148.

[149] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, "PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Jul. 2017, pp. 77–85.
doi: 10.1109/CVPR.2017.16.

[150] Z. Ling, Y. Yao, X. Li, and H. Su, "On the Efficacy of 3D Point Cloud Reinforcement Learning," *arXiv:2306.06799 [cs]*, Jun. 2023.
doi: 10.48550/arXiv.2306.06799.

[151] S. Kraemer, C. Stiller, and M. E. Bouzouraa, "LiDAR-Based Object Tracking and Shape Estimation Using Polylines and Free-Space Information," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2018, pp. 4515–4522.
doi: 10.1109/IROS.2018.8593385.

[152] H. Chen and P. Lu, "Real-time identification and avoidance of simultaneous static and dynamic obstacles on point cloud for UAVs navigation," *Robotics and Autonomous Systems*, vol. 154, p. 104 124, Aug. 2022.
doi: 10.1016/j.robot.2022.104124.

[153] H. Zhao, L. Jiang, J. Jia, P. H. S. Torr, and V. Koltun, "Point Transformer," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 16 259–16 268.
doi: 10.1109/ICCV48922.2021.01595.

[154] E. Kazemi and I. Soltani, "MarineFormer: A Spatio-Temporal Attention Model for USV Navigation in Dynamic Marine Environments," *arXiv:2410.13973* [*cs*], Dec. 2024.
doi: 10.48550/arXiv.2410.13973.

[155] Z. Zhang, H. Fu, J. Yang, and Y. Lin, "Deep reinforcement learning for path planning of autonomous mobile robots in complicated environments," *Complex & Intelligent Systems*, vol. 11, no. 6, p. 277, May 2025.
doi: 10.1007/s40747-025-01906-9.

[156] N. Dengler, J. D. A. Ferrandis, J. Moura, S. Vijayakumar, and M. Bennewitz, "Learning Goal-Directed Object Pushing in Cluttered Scenes with Location-Based Attention," *arXiv:2403.17667* [*cs*], Mar. 2025.
doi: 10.48550/arXiv.2403.17667.

[157] K. S. Arun, T. S. Huang, and S. D. Blostein, "Least-Squares Fitting of Two 3-D Point Sets," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-9, no. 5, pp. 698–700, Sep. 1987.
doi: 10.1109/TPAMI.1987.4767965.

[158] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *arXiv:1509.02971* [*cs, stat*], Jul. 2019.
doi: 10.48550/arXiv.1509.02971.

[159] Z. Niu, G. Zhong, and H. Yu, "A review on the attention mechanism of deep learning," *Neurocomputing*, vol. 452, pp. 48–62, Sep. 2021.
doi: 10.1016/j.neucom.2021.03.091.

[160] T. Luong, H. Pham, and C. D. Manning, "Effective Approaches to Attention-based Neural Machine Translation," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, L. Màrquez, C. Callison-Burch, and J. Su, Eds., Association for Computational Linguistics, Sep. 2015, pp. 1412–1421.
doi: 10.18653/v1/D15-1166.

[161] S. Samavi, J. R. Han, F. Shkurti, and A. P. Schoellig, "SICNav: Safe and Interactive Crowd Navigation using Model Predictive Control and Bilevel Optimization," *IEEE Transactions on Robotics*, vol. 41, pp. 801–818, 2025.
doi: 10.1109/TRO.2024.3484634.

[162] H. Kolivand, M. S. Rahim, M. S. Sunar, A. Z. A. Fata, and C. Wren, "An integration of enhanced social force and crowd control models for high-density crowd simulation," *Neural Computing and Applications*, vol. 33, no. 11, pp. 6095–6117, Jun. 2021.
doi: 10.1007/s00521-020-05385-6.

[163] B. Shen, F. Xia, C. Li, R. Martín-Martín, L. Fan, G. Wang, C. Pérez-D'Arpino, S. Buch, S. Srivastava, L. Tchapmi, M. Tchapmi, K. Vainio, J. Wong, L. Fei-Fei, and S. Savarese, "iGibson 1.0: A Simulation Environment for Interactive Tasks in Large Realistic Scenes," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sep. 2021, pp. 7520–7527.
doi: 10.1109/IROS51168.2021.9636667.

[164] *iGibson Challenge 2021*. [Online]. Available: `https://svl.stanford.edu/igibson/challenge2021.html` Accessed: Aug. 4, 2023.

[165] G. Grisetti, C. Stachniss, and W. Burgard, "Improving Grid-based SLAM with Rao-Blackwellized Particle Filters by Adaptive Proposals and Selective Resampling," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, IEEE, 2005, pp. 2432–2437.
doi: 10.1109/ROBOT.2005.1570477.

[166] P. Vamplew, R. Dazeley, C. Foale, S. Firmin, and J. Mummery, "Human-aligned artificial intelligence is a multiobjective problem," *Ethics and Information Technology*, vol. 20, no. 1, pp. 27–40, Mar. 2018.
doi: 10.1007/s10676-017-9440-6.

[167] C. F. Hayes, R. Rădulescu, E. Bargiacchi, J. Källström, M. Macfarlane, M. Reymond, T. Verstraeten, L. M. Zintgraf, R. Dazeley, F. Heintz, E. Howley, A. A. Irissappane, P. Mannion, A. Nowé, G. Ramos, M. Restelli, P. Vamplew, and D. M. Roijers, "A practical guide to multi-objective reinforcement learning and planning," *Autonomous Agents and Multi-Agent Systems*, vol. 36, no. 1, p. 26, Apr. 2022.
doi: 10.1007/s10458-022-09552-y.

[168] J. Choi, C. Dance, J.-e. Kim, K.-s. Park, J. Han, J. Seo, and M. Kim, "Fast Adaptation of Deep Reinforcement Learning-Based Navigation Skills to Human Preference," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, May 2020, pp. 3363–3370.
doi: 10.1109/ICRA40945.2020.9197159.

[169] L. Keselman, K. Shih, M. Hebert, and A. Steinfeld, "Optimizing Algorithms from Pairwise User Preferences," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2023, pp. 4161–4167.
doi: 10.1109/IROS55552.2023.10342081.

[170] E. Bıyık, D. P. Losey, M. Palan, N. C. Landolfi, G. Shevchuk, and D. Sadigh, "Learning reward functions from diverse sources of human feedback: Optimally integrating demonstrations and preferences," *The International Journal of Robotics Research*, vol. 41, no. 1, pp. 45–67, Jan. 2022.
doi: 10.1177/02783649211041652.

[171] G. Ferrer and A. Sanfeliu, "Anticipative kinodynamic planning: Multi-objective robot navigation in urban and dynamic environments," *Autonomous Robots*, vol. 43, no. 6, pp. 1473–1488, Aug. 2019.
doi: 10.1007/s10514-018-9806-6.

[172] S. Kumar and A. Sikander, "A modified probabilistic roadmap algorithm for efficient mobile robot path planning," *Engineering Optimization*, vol. 55, no. 9, pp. 1616–1634, Sep. 2023.
doi: 10.1080/0305215X.2022.2104840.

[173] S. B. Banisetty, S. Forer, L. Yliniemi, M. Nicolescu, and D. Feil-Seifer, "Socially Aware Navigation: A Non-linear Multi-objective Optimization Approach," *ACM Transactions on Interactive Intelligent Systems (TiiS)*, vol. 11, no. 2, 15:1–15:26, Jul. 2021.
doi: 10.1145/3453445.

[174] R. Yang, X. Sun, and K. Narasimhan, "A Generalized Algorithm for Multi-Objective Reinforcement Learning and Policy Adaptation," in *Advances in Neural Information Processing Systems*, vol. 32, Curran Associates, Inc., 2019.

[175] T. Basaklar, S. Gumussoy, and U. Y. Ogras, "PD-MORL: Preference-Driven Multi-Objective Reinforcement Learning Algorithm," *Proceedings of the 11th International Conference on Learning Representations*, 2023.

[176] J. Xu, Y. Tian, P. Ma, D. Rus, S. Sueda, and W. Matusik, "Prediction-Guided Multi-Objective Reinforcement Learning for Continuous Robot Control," in *Proceedings of the 37th International Conference on Machine Learning*, PMLR, Nov. 2020, pp. 10 607–10 616.

[177] X. He and C. Lv, "Toward personalized decision making for autonomous vehicles: A constrained multi-objective reinforcement learning technique," *Transportation Research Part C: Emerging Technologies*, vol. 156, p. 104 352, Nov. 2023.
doi: 10.1016/j.trc.2023.104352.

[178] S. Huang, A. Abdolmaleki, G. Vezzani, P. Brakel, D. J. Mankowitz, M. Neunert, S. Bohez, Y. Tassa, N. Heess, M. Riedmiller, and R. Hadsell, "A Constrained Multi-Objective Reinforcement Learning Framework," in *Proceedings of the 5th Conference on Robot Learning*, PMLR, Jan. 2022, pp. 883–893.

[179] G. Cheng, Y. Wang, L. Dong, W. Cai, and C. Sun, "Multi-objective deep reinforcement learning for crowd-aware robot navigation with dynamic human preference," *Neural Computing and Applications*, Jun. 2023.
doi: 10.1007/s00521-023-08385-4.

[180] C.-L. Cheng, C.-C. Hsu, S. Saeedvand, and J.-H. Jo, "Multi-objective crowd-aware robot navigation system using deep reinforcement learning," *Applied Soft Computing*, vol. 151, p. 111 154, Jan. 2024.
doi: 10.1016/j.asoc.2023.111154.

[181] K. Lee, S. Kim, and J. Choi, "Adaptive and explainable deployment of navigation skills via hierarchical deep reinforcement learning," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2023, pp. 1673–1679. doi: 10.1109/ICRA48891.2023.10160371.

[182] N. Wilde, S. L. Smith, and J. Alonso-Mora, "Scalarizing Multi-Objective Robot Planning Problems Using Weighted Maximization," *IEEE Robotics and Automation Letters*, vol. 9, no. 3, pp. 2503–2510, Mar. 2024. doi: 10.1109/LRA.2024.3357313.

[183] A. Ballou, X. Alameda-Pineda, and C. Reinke, "Variational meta reinforcement learning for social robotics," *Applied Intelligence*, vol. 53, no. 22, pp. 27 249–27 268, Nov. 2023. doi: 10.1007/s10489-023-04691-5.

[184] M. Andrychowicz, F. Wolski, A. Ray, J. Schneider, R. Fong, P. Welinder, B. McGrew, J. Tobin, P. Abbeel, and W. Zaremba, "Hindsight experience replay," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, ser. NIPS'17, Curran Associates Inc., Dec. 2017, pp. 5055–5065.

[185] D. S. Brown, W. Goo, and S. Niekum, "Better-than-Demonstrator Imitation Learning via Automatically-Ranked Demonstrations," in *Proceedings of the Conference on Robot Learning*, PMLR, May 2020, pp. 330–359.

[186] R. A. Bradley and M. E. Terry, "Rank analysis of incomplete block designs: I. The method of paired comparisons," *Biometrika*, vol. 39, no. 3/4, pp. 324–345, 1952. doi: 10.2307/2334029.

[187] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor," in *Proceedings of the 35th International Conference on Machine Learning*, PMLR, Jul. 2018, pp. 1861–1870.

[188] D. M. Ziegler, N. Stiennon, J. Wu, T. B. Brown, A. Radford, D. Amodei, P. Christiano, and G. Irving, "Fine-Tuning Language Models from Human Preferences," *arXiv:1909.08593 [cs, stat]*, Jan. 2020. doi: 10.48550/arXiv.1909.08593.

[189] D. Marta, S. Holk, C. Pek, and I. Leite, "SEQUEL: Semi-Supervised Preference-Based RL with Query Synthesis Via Latent Interpolation," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2024, pp. 9585–9592. doi: 10.1109/ICRA57147.2024.10610534.

[190] S. Holk, D. Marta, and I. Leite, "POLITE: Preferences Combined with Highlights in Reinforcement Learning," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024. doi: 10.1109/ICRA57147.2024.10610505.

[191] S. Holk, D. Marta, and I. Leite, "PREDILECT: Preferences Delineated with Zero-Shot Language-based Reasoning in Reinforcement Learning," in *Proceedings of the 2024 ACM/IEEE International Conference on Human-Robot Interaction*, ser. HRI '24, Association for Computing Machinery, Mar. 2024, pp. 259–268.
doi: 10.1145/3610977.3634970.

[192] E. Bıyık, M. Palan, N. C. Landolfi, D. P. Losey, and D. Sadigh, "Asking Easy Questions: A User-Friendly Approach to Active Reward Learning," in *Proceedings of the Conference on Robot Learning*, PMLR, May 2020, pp. 1177–1190.

[193] E. Biyik and D. Sadigh, "Batch Active Preference-Based Learning of Reward Functions," in *Proceedings of The 2nd Conference on Robot Learning*, PMLR, Oct. 2018, pp. 519–528.

[194] Y. Wang, W. Zhong, L. Li, F. Mi, X. Zeng, W. Huang, L. Shang, X. Jiang, and Q. Liu, "Aligning Large Language Models with Human: A Survey," *arXiv:2307.12966 [cs]*, Jul. 2023.
doi: 10.48550/arXiv.2307.12966.

[195] J. Gawlikowski, C. R. N. Tassi, M. Ali, J. Lee, M. Humt, J. Feng, A. Kruspe, R. Triebel, P. Jung, R. Roscher, M. Shahzad, W. Yang, R. Bamler, and X. X. Zhu, "A survey of uncertainty in deep neural networks," *Artificial Intelligence Review*, vol. 56, no. S1, pp. 1513–1589, Oct. 2023.
doi: 10.1007/s10462-023-10562-9.

[196] K. Lee, M. Laskin, A. Srinivas, and P. Abbeel, "SUNRISE: A Simple Unified Framework for Ensemble Learning in Deep Reinforcement Learning," in *Proceedings of the 38th International Conference on Machine Learning*, PMLR, Jul. 2021, pp. 6131–6141.

[197] H. Sheikh, M. Phielipp, and L. Boloni, "Maximizing ensemble diversity in deep reinforcement learning," in *International Conference on Learning Representations*, 2022.

[198] Y. Wang, Z. Sun, J. Zhang, Z. Xian, E. Biyik, D. Held, and Z. Erickson, *RL-VLM-F: Reinforcement Learning from Vision Language Foundation Model Feedback*, Jun. 2024.
doi: 10.48550/arXiv.2402.03681. arXiv: 2402.03681 [cs].

[199] R. Wang, D. Zhao, Z. Yuan, I. Obi, and B.-C. Min, "PrefCLM: Enhancing Preference-Based Reinforcement Learning With Crowdsourced Large Language Models," *IEEE Robotics and Automation Letters*, vol. 10, no. 3, pp. 2486–2493, Mar. 2025.
doi: 10.1109/LRA.2025.3528663.

[200] A. Raffin, A. Hill, A. Gleave, A. Kanervisto, M. Ernestus, and N. Dormann, "Stable-Baselines3: Reliable Reinforcement Learning Implementations," *Journal of Machine Learning Research*, vol. 22, no. 268, pp. 1–8, 2021.

[201] L. He, N. Aouf, and B. Song, "Explainable Deep Reinforcement Learning for UAV autonomous path planning," *Aerospace Science and Technology*, vol. 118, p. 107 052, Nov. 2021.
doi: 10.1016/j.ast.2021.107052.

[202] F. Cruz, R. Dazeley, P. Vamplew, and I. Moreira, "Explainable robotic systems: Understanding goal-driven actions in a reinforcement learning scenario," *Neural Computing and Applications*, vol. 35, no. 25, pp. 18 113–18 130, Sep. 2023.
doi: 10.1007/s00521-021-06425-5.

[203] A. Heuillet, F. Couthouis, and N. Díaz-Rodríguez, "Explainability in deep reinforcement learning," *Knowledge-Based Systems*, vol. 214, p. 106 685, Feb. 2021.
doi: 10.1016/j.knosys.2020.106685.

[204] J. D. Hunter, "Matplotlib: A 2D Graphics Environment," *Computing in Science & Engineering*, vol. 9, no. 3, pp. 90–95, May 2007.
doi: 10.1109/MCSE.2007.55.

[205] S. Casper, X. Davies, C. Shi, T. Krendl Gilbert, J. Scheurer, J. Rando Ramirez, R. Freedman, T. Korbak, D. Lindner, P. Freire, T. Wang, S. Marks, C.-R. Segerie, M. Carroll, A. Peng, P. Christoffersen, M. Damani, S. Slocum, U. Anwar, A. Siththaranjan, M. Nadeau, E. J. Michaud, J. Pfau, D. Krasheninnikov, X. Chen, L. Langosco, P. Hase, E. Bıyık, A. Dragan, D. Krueger, D. Sadigh, and D. Hadfield-Menell, "Open Problems and Fundamental Limitations of Reinforcement Learning from Human Feedback," *Transactions on Machine Learning Research*, Dec. 2023.
doi: 10.3929/ethz-b-000651806.

[206] D. J. H. Iii and D. Sadigh, "Few-Shot Preference Learning for Human-in-the-Loop RL," in *Proceedings of The 6th Conference on Robot Learning*, PMLR, Mar. 2023, pp. 2014–2025.

[207] M. D. Zhao, R. Simmons, and H. Admoni, "Learning Human Contribution Preferences in Collaborative Human-Robot Tasks," in *Proceedings of The 7th Conference on Robot Learning*, PMLR, Dec. 2023, pp. 3597–3618.

[208] S. Gillet, D. Marta, M. Akif, and I. Leite, "Shielding for Socially Appropriate Robot Listening Behaviors," in *2024 33rd IEEE International Conference on Robot and Human Interactive Communication (ROMAN)*, Aug. 2024, pp. 2279–2286.
doi: 10.1109/RO-MAN60168.2024.10731356.

[209] N. Tsoi, A. Xiang, P. Yu, S. S. Sohn, G. Schwartz, S. Ramesh, M. Hussein, A. W. Gupta, M. Kapadia, and M. Vázquez, "SEAN 2.0: Formalizing and Generating Social Situations for Robot Navigation," *IEEE Robotics and Automation Letters*, pp. 1–8, 2022.
doi: 10.1109/LRA.2022.3196783.

[210] G. Baker, T. Bridgwater, P. Bremner, and M. Giuliani, "Towards an immersive user interface for waypoint navigation of a mobile robot," *arXiv:2003.12772*, Mar. 2020.
doi: 10.48550/arXiv.2003.12772.

[211] K. Amin, N. Jiang, and S. Singh, "Repeated Inverse Reinforcement Learning," in *Advances in Neural Information Processing Systems*, vol. 30, Curran Associates, Inc., 2017.

[212] R. Wang, D. Zhao, D. Suh, Z. Yuan, G. Chen, and B.-C. Min, "Personalization in Human-Robot Interaction through Preference-based Action Representation Learning," *arXiv:2409.13822 [cs]*, Sep. 2024.
doi: 10.48550/arXiv.2409.13822.

[213] D. Marta, C. Pek, G. I. Melsión, J. Tumova, and I. Leite, "Human-feedback shield synthesis for perceived safety in deep reinforcement learning," *IEEE Robotics and Automation Letters*, vol. 7, no. 1, pp. 406–413, 2021.
doi: 10.1109/LRA.2021.3128237.

[214] S. Gillet, M. T. Parreira, M. Vázquez, and I. Leite, "Learning Gaze Behaviors for Balancing Participation in Group Human-Robot Interactions," in *2022 17th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, Mar. 2022, pp. 265–274.
doi: 10.1109/HRI53351.2022.9889416.

[215] G. LeMasurier, J. Tukpah, M. Wonsick, J. Allspaw, B. Hertel, J. Epstein, R. Azadeh, T. Padir, H. A. Yanco, and E. Phillips, "Comparing a 2D Keyboard and Mouse Interface to Virtual Reality for Human-in-the-Loop Robot Planning for Mobile Manipulation," in *2024 33rd IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, IEEE, Aug. 2024, pp. 2197–2203.
doi: 10.1109/RO-MAN60168.2024.10731138.

[216] *Pygame/pygame*, Mar. 2025. [Online]. Available: `https://github.com/pygame/pygame` Accessed: Mar. 2, 2025.

[217] F. D. Davis, "Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology," *MIS Quarterly*, vol. 13, no. 3, pp. 319–340, 1989.
doi: 10.2307/249008.

[218] M. Usoh, E. Catena, S. Arman, and M. Slater, "Using Presence Questionnaires in Reality," *Presence: Teleoperators and Virtual Environments*, vol. 9, no. 5, pp. 497–503, Oct. 2000.
doi: 10.1162/105474600566989.

[219] J. Brooke, "SUS: A 'Quick and Dirty' Usability Scale," in *Usability Evaluation In Industry*, CRC Press, 1996.

[220] B. Schöne, M. Wessels, and T. Gruber, "Experiences in Virtual Reality: A Window to Autobiographical Memory," *Current Psychology*, vol. 38, no. 3, pp. 715–719, Jun. 2019.
doi: 10.1007/s12144-017-9648-y.

[221] E. Puiutta and E. M. S. P. Veith, "Explainable Reinforcement Learning: A Survey," in *Machine Learning and Knowledge Extraction*, A. Holzinger, P. Kieseberg, A. M. Tjoa, and E. Weippl, Eds., Springer International Publishing, 2020, pp. 77–95.
doi: 10.1007/978-3-030-57321-8_5.

[222] S. Thellman and T. Ziemke, "The Perceptual Belief Problem: Why Explainability Is a Tough Challenge in Social Robotics," *ACM Transactions on Human-Robot Interaction*, vol. 10, no. 3, pp. 1–15, Sep. 2021.
doi: 10.1145/3461781.

[223] Y. Rong, T. Leemann, T.-T. Nguyen, L. Fiedler, P. Qian, V. Unhelkar, T. Seidel, G. Kasneci, and E. Kasneci, "Towards Human-Centered Explainable AI: A Survey of User Studies for Model Explanations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46, no. 4, pp. 2104–2122, Apr. 2024.
doi: 10.1109/TPAMI.2023.3331846.

[224] A. Halilovic and S. Krivic, "Robot Explanation Identity," *arXiv:2405.13841*, May 2024.
doi: 10.48550/arXiv.2405.13841.

[225] K. Hald, K. Weitz, E. André, and M. Rehm, ""An Error Occurred!" - Trust Repair With Virtual Robot Using Levels of Mistake Explanation," in *Proceedings of the 9th International Conference on Human-Agent Interaction*, ACM, Nov. 2021, pp. 218–226.
doi: 10.1145/3472307.3484170.

[226] T. Speith, "A Review of Taxonomies of Explainable Artificial Intelligence (XAI) Methods," in *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, ser. FAccT '22, Association for Computing Machinery, Jun. 2022, pp. 2239–2250.
doi: 10.1145/3531146.3534639.

[227] R. Tomsett, D. Braines, D. Harborne, A. Preece, and S. Chakraborty, "Interpretable to Whom? A Role-based Model for Analyzing Interpretable Machine Learning Systems," *arXiv:1806.07552* [*cs*], Jun. 2018.
doi: 10.48550/arXiv.1806.07552.

[228] C. Molnar, *Interpretable Machine Learning*. Lulu.com, 2020.

[229] Q. V. Liao, D. Gruen, and S. Miller, "Questioning the AI: Informing Design Practices for Explainable AI User Experiences," in *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, ser. CHI '20, Association for Com-

puting Machinery, Apr. 2020, pp. 1–15.
doi: 10.1145/3313831.3376590.

[230] N. Burkart and M. F. Huber, "A Survey on the Explainability of Supervised Machine Learning," *Journal of Artificial Intelligence Research*, vol. 70, pp. 245–317, Jan. 2021.
doi: 10.1613/jair.1.12228.

[231] S. Milani, N. Topin, M. Veloso, and F. Fang, "Explainable Reinforcement Learning: A Survey and Comparative Review," *ACM Comput. Surv.*, vol. 56, no. 7, 168:1–168:36, Apr. 2024.
doi: 10.1145/3616864.

[232] M. Nauta, J. Trienes, S. Pathak, E. Nguyen, M. Peters, Y. Schmitt, J. Schlötterer, M. Van Keulen, and C. Seifert, "From Anecdotal Evidence to Quantitative Evaluation Methods: A Systematic Review on Evaluating Explainable AI," *ACM Computing Surveys*, vol. 55, no. 13s, pp. 1–42, Dec. 2023.
doi: 10.1145/3583558.

[233] A. Halilovic and S. Krivic, "The Influence of a Robot's Personality on Real-Time Explanations of Its Navigation," in *Social Robotics*, A. A. Ali, J.-J. Cabibihan, N. Meskin, S. Rossi, W. Jiang, H. He, and S. S. Ge, Eds., Springer Nature, 2024, pp. 133–147.
doi: 10.1007/978-981-99-8718-4_12.

[234] D. Das and S. Chernova, "Semantic-Based Explainable AI: Leveraging Semantic Scene Graphs and Pairwise Ranking to Explain Robot Failures," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sep. 2021, pp. 3034–3041.
doi: 10.1109/IROS51168.2021.9635890.

[235] C. Wang, A. Belardinelli, S. Hasler, T. Stouraitis, D. Tanneberg, and M. Gienger, "Explainable Human-Robot Training and Cooperation with Augmented Reality," in *Extended Abstracts of the 2023 CHI Conference on Human Factors in Computing Systems*, ACM, Apr. 2023, pp. 1–5.
doi: 10.1145/3544549.3583889.

[236] M. Edmonds, F. Gao, H. Liu, X. Xie, S. Qi, B. Rothrock, Y. Zhu, Y. N. Wu, H. Lu, and S.-C. Zhu, "A tale of two explanations: Enhancing human trust by explaining robot behavior," *Science Robotics*, vol. 4, no. 37, eaay4663, Dec. 2019.
doi: 10.1126/scirobotics.aay4663.

[237] M. T. Ribeiro, S. Singh, and C. Guestrin, ""Why Should I Trust You?": Explaining the Predictions of Any Classifier," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '16, Association for Computing Machinery, Aug. 2016, pp. 1135–1144.
doi: 10.1145/2939672.2939778.

[238] S. M. Lundberg and S.-I. Lee, "A Unified Approach to Interpreting Model Predictions," in *Advances in Neural Information Processing Systems*, vol. 30, Curran Associates, Inc., 2017.

[239] M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic Attribution for Deep Networks," in *Proceedings of the 34th International Conference on Machine Learning*, PMLR, Jul. 2017, pp. 3319–3328.

[240] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps," *arXiv:1312.6034* [*cs*], Apr. 2014.
doi: 10.48550/arXiv.1312.6034.

[241] I. Hameed, S. Sharpe, D. Barcklow, J. Au-Yeung, S. Verma, J. Huang, B. Barr, and C. B. Bruss, "BASED-XAI: Breaking Ablation Studies Down for Explainable Artificial Intelligence," *arXiv:2207.05566* [*cs*], Sep. 2022.
doi: 10.48550/arXiv.2207.05566.

[242] M. G. Kendall, "A new measure of rank correlation," *Biometrika*, vol. 30, no. 1-2, pp. 81–93, Jun. 1938.
doi: 10.1093/biomet/30.1-2.81.

[243] T. Schrills, M. Zoubir, M. Bickel, S. Kargl, and T. Franke, "Are users in the loop? Development of the subjective information processing awareness scale to assess XAI," in *Proceedings of the ACM CHI Workshop on Operationalizing Human-Centered Perspectives in Explainable AI* (*HCXAI 2021*), 2021.

[244] R. R. Hoffman, S. T. Mueller, G. Klein, and J. Litman, "Measures for explainable AI: Explanation goodness, user satisfaction, mental models, curiosity, trust, and human-AI performance," *Frontiers in Computer Science*, vol. 5, Feb. 2023.
doi: 10.3389/fcomp.2023.1096257.

[245] R. Firoozi, J. Tucker, S. Tian, A. Majumdar, J. Sun, W. Liu, Y. Zhu, S. Song, A. Kapoor, K. Hausman, B. Ichter, D. Driess, J. Wu, C. Lu, and M. Schwager, "Foundation models in robotics: Applications, challenges, and the future," *The International Journal of Robotics Research*, vol. 44, no. 5, pp. 701–739, Apr. 2025.
doi: 10.1177/02783649241281508.

[246] Y. Hu, Q. Xie, V. Jain, J. Francis, J. Patrikar, N. Keetha, S. Kim, Y. Xie, T. Zhang, H.-S. Fang, S. Zhao, S. Omidshafiei, D.-K. Kim, A.-a. Agha-mohammadi, K. Sycara, M. Johnson-Roberson, D. Batra, X. Wang, S. Scherer, C. Wang, Z. Kira, F. Xia, and Y. Bisk, "Toward General-Purpose Robots via Foundation Models: A Survey and Meta-Analysis," *arXiv:2312.08782* [*cs*], Oct. 2024.
doi: 10.48550/arXiv.2312.08782.

[247] B. Zitkovich, T. Yu, S. Xu, P. Xu, T. Xiao, F. Xia, J. Wu, P. Wohlhart, S. Welker, A. Wahid, Q. Vuong, V. Vanhoucke, H. Tran, R. Soricut, A. Singh, J. Singh, P. Sermanet, P. R. Sanketi, G. Salazar, M. S. Ryoo, K. Reymann, K. Rao, K. Pertsch, I. Mordatch, H. Michalewski, Y. Lu, S. Levine, L. Lee, T.-W. E. Lee, I. Leal, Y. Kuang, D. Kalashnikov, R. Julian, N. J. Joshi, A. Irpan, B. Ichter, J. Hsu, A. Herzog, K. Hausman, K. Gopalakrishnan, C. Fu, P. Florence, C. Finn, K. A. Dubey, D. Driess, T. Ding, K. M. Choromanski, X. Chen, Y. Chebotar, J. Carbajal, N. Brown, A. Brohan, M. G. Arenas, and K. Han, "RT-2: Vision-Language-Action Models Transfer Web Knowledge to Robotic Control," in *Proceedings of The 7th Conference on Robot Learning*, PMLR, Dec. 2023, pp. 2165–2183.

[248] M. J. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. Foster, G. Lam, P. Sanketi, Q. Vuong, T. Kollar, B. Burchfiel, R. Tedrake, D. Sadigh, S. Levine, P. Liang, and C. Finn, "OpenVLA: An Open-Source Vision-Language-Action Model," *arXiv:2406.09246* [*cs*], Sep. 2024.
doi: 10.48550/arXiv.2406.09246.

[249] K. Bousmalis, G. Vezzani, D. Rao, C. Devin, A. X. Lee, M. Bauza, T. Davchev, Y. Zhou, A. Gupta, A. Raju, A. Laurens, C. Fantacci, V. Dalibard, M. Zambelli, M. Martins, R. Pevceviciute, M. Blokzijl, M. Denil, N. Batchelor, T. Lampe, E. Parisotto, K. Żołna, S. Reed, S. G. Colmenarejo, J. Scholz, A. Abdolmaleki, O. Groth, J.-B. Regli, O. Sushkov, T. Rothörl, J. E. Chen, Y. Aytar, D. Barker, J. Ortiz, M. Riedmiller, J. T. Springenberg, R. Hadsell, F. Nori, and N. Heess, "RoboCat: A Self-Improving Foundation Agent for Robotic Manipulation," *arXiv:2306.11706* [*cs*], no. arXiv:2306.11706, Jun. 2023.
doi: 10.48550/arXiv.2306.11706.

[250] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, D. Bikel, L. Blecher, C. C. Ferrer, M. Chen, G. Cucurull, D. Esiobu, J. Fernandes, J. Fu, W. Fu, B. Fuller, C. Gao, V. Goswami, N. Goyal, A. Hartshorn, S. Hosseini, R. Hou, H. Inan, M. Kardas, V. Kerkez, M. Khabsa, I. Kloumann, A. Korenev, P. S. Koura, M.-A. Lachaux, T. Lavril, J. Lee, D. Liskovich, Y. Lu, Y. Mao, X. Martinet, T. Mihaylov, P. Mishra, I. Molybog, Y. Nie, A. Poulton, J. Reizenstein, R. Rungta, K. Saladi, A. Schelten, R. Silva, E. M. Smith, R. Subramanian, X. E. Tan, B. Tang, R. Taylor, A. Williams, J. X. Kuan, P. Xu, Z. Yan, I. Zarov, Y. Zhang, A. Fan, M. Kambadur, S. Narang, A. Rodriguez, R. Stojnic, S. Edunov, and T. Scialom, "Llama 2: Open Foundation and Fine-Tuned Chat Models," *arXiv:2307.09288* [*cs*], Jul. 2023.
doi: 10.48550/arXiv.2307.09288.

[251] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving Language Understanding by Generative Pre-Training," *Preprint*, pp. 1–12, 2018.

[252] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning Transferable

Visual Models From Natural Language Supervision," in *Proceedings of the 38th International Conference on Machine Learning*, PMLR, Jul. 2021, pp. 8748–8763.

[253] J. Li, D. Li, S. Savarese, and S. Hoi, "BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models," in *Proceedings of the 40th International Conference on Machine Learning*, PMLR, Jul. 2023, pp. 19 730–19 742.

[254] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, J. Hsu, J. Ibarz, B. Ichter, A. Irpan, T. Jackson, S. Jesmonth, N. J. Joshi, R. Julian, D. Kalashnikov, Y. Kuang, I. Leal, K.-H. Lee, S. Levine, Y. Lu, U. Malla, D. Manjunath, I. Mordatch, O. Nachum, C. Parada, J. Peralta, E. Perez, K. Pertsch, J. Quiambao, K. Rao, M. Ryoo, G. Salazar, P. Sanketi, K. Sayed, J. Singh, S. Sontakke, A. Stone, C. Tan, H. Tran, V. Vanhoucke, S. Vega, Q. Vuong, F. Xia, T. Xiao, P. Xu, S. Xu, T. Yu, and B. Zitkovich, "RT-1: Robotics Transformer for Real-World Control at Scale," *arXiv:2212.06817* [*cs*], Aug. 2023. doi: 10.48550/arXiv.2212.06817.

[255] O. M. Team, D. Ghosh, H. Walke, K. Pertsch, K. Black, O. Mees, S. Dasari, J. Hejna, T. Kreiman, C. Xu, J. Luo, Y. L. Tan, L. Y. Chen, P. Sanketi, Q. Vuong, T. Xiao, D. Sadigh, C. Finn, and S. Levine, "Octo: An Open-Source Generalist Robot Policy," *arXiv:2405.12213* [*cs*], May 2024.
doi: 10.48550/arXiv.2405.12213.

[256] NVIDIA, J. Bjorck, F. Castañeda, N. Cherniadev, X. Da, R. Ding, L. " Fan, Y. Fang, D. Fox, F. Hu, S. Huang, J. Jang, Z. Jiang, J. Kautz, K. Kundalia, L. Lao, Z. Li, Z. Lin, K. Lin, G. Liu, E. Llontop, L. Magne, A. Mandlekar, A. Narayan, S. Nasiriany, S. Reed, Y. L. Tan, G. Wang, Z. Wang, J. Wang, Q. Wang, J. Xiang, Y. Xie, Y. Xu, Z. Xu, S. Ye, Z. Yu, A. Zhang, H. Zhang, Y. Zhao, R. Zheng, and Y. Zhu, "GR00T N1: An Open Foundation Model for Generalist Humanoid Robots," *arXiv:2503.14734* [*cs*], Mar. 2025.
doi: 10.48550/arXiv.2503.14734.

[257] Covariant, *Introducing RFM-1: Giving robots human-like reasoning capabilities*, Mar. 2024. [Online]. Available: `https://covariant.ai/insights/introducing-rfm-1-giving-robots-human-like-reasoning-capabilities/` Accessed: Jul. 4, 2025.

[258] D. Driess, F. Xia, M. S. M. Sajjadi, C. Lynch, A. Chowdhery, B. Ichter, A. Wahid, J. Tompson, Q. Vuong, T. Yu, W. Huang, Y. Chebotar, P. Sermanet, D. Duckworth, S. Levine, V. Vanhoucke, K. Hausman, M. Toussaint, K. Greff, A. Zeng, I. Mordatch, and P. Florence, "PaLM-E: An Embodied Multimodal Language Model," *Proceedings of the 40th International Conference on Machine Learning (ICML)*, Jun. 2023.

[259] S. Reed, K. Zolna, E. Parisotto, S. G. Colmenarejo, A. Novikov, G. Barth-Maron, M. Gimenez, Y. Sulsky, J. Kay, J. T. Springenberg, T. Eccles, J. Bruce, A. Razavi, A. Edwards, N. Heess, Y. Chen, R. Hadsell, O. Vinyals, M. Bordbar, and N. de Freitas, "A Generalist Agent," *arXiv:2205.06175 [cs]*, Nov. 2022.
doi: 10.48550/arXiv.2205.06175.

[260] S. Sartor and N. Thompson, "Neural Scaling Laws in Robotics," *arXiv:2405.14005 [cs]*, Jan. 2025.
doi: 10.48550/arXiv.2405.14005.

[261] R. Bommasani et al., "On the Opportunities and Risks of Foundation Models," *arXiv:2108.07258 [cs]*, Jul. 2022.
doi: 10.48550/arXiv.2108.07258.

[262] B. Min, H. Ross, E. Sulem, A. P. B. Veyseh, T. H. Nguyen, O. Sainz, E. Agirre, I. Heintz, and D. Roth, "Recent Advances in Natural Language Processing via Large Pre-trained Language Models: A Survey," *ACM Comput. Surv.*, vol. 56, no. 2, 30:1–30:40, Sep. 2023.
doi: 10.1145/3605943.

[263] W. Ye, Y. Zhang, M. Wang, S. Wang, X. Gu, P. Abbeel, and Y. Gao, "Foundation Reinforcement Learning: Towards Embodied Generalist Agents with Foundation Prior Assistance," *arXiv:2310.02635 [cs]*, Oct. 2023.

[264] C. Y. Kim, C. P. Lee, and B. Mutlu, "Understanding Large-Language Model (LLM)-powered Human-Robot Interaction," in *Proceedings of the 2024 ACM/IEEE International Conference on Human-Robot Interaction*, ser. HRI '24, Association for Computing Machinery, Mar. 2024, pp. 371–380.
doi: 10.1145/3610977.3634966.

[265] A. O'Neill et al., "Open X-Embodiment: Robotic Learning Datasets and RT-X Models : Open X-Embodiment Collaboration," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, May 2024, pp. 6892–6903.
doi: 10.1109/ICRA57147.2024.10611477.

[266] A. Khazatsky, K. Pertsch, S. Nair, A. Balakrishna, S. Dasari, S. Karamcheti, S. Nasiriany, M. Srirama, L. Chen, K. Ellis, P. Fagan, J. Hejna, M. Itkina, M. Lepert, Y. Ma, P. Miller, J. Wu, S. Belkhale, S. Dass, H. Ha, A. Jain, A. Lee, Y. Lee, M. Memmel, S. Park, I. Radosavovic, K. Wang, A. Zhan, K. Black, C. Chi, K. Hatch, S. Lin, J. Lu, J. Mercat, A. Rehman, P. Sanketi, A. Sharma, C. Simpson, Q. Vuong, H. Walke, B. Wulfe, T. Xiao, J. Yang, A. Yavary, T. Zhao, C. Agia, R. Baijal, M. Castro, D. Chen, Q. Chen, T. Chung, J. Drake, E. Foster, J. Gao, D. Herrera, M. Heo, K. Hsu, J. Hu, D. Jackson, C. Le, Y. Li, R. Lin, Z. Ma, A. Maddukuri, S. Mirchandani, D. Morton, T. Nguyen, A. O'Neill, R. Scalise, D. Seale, V. Son, S. Tian, E. Tran, A. Wang, Y. Wu, A. Xie, J. Yang, P. Yin, Y. Zhang, O. Bastani, G. Berseth, J. Bohg, K. Goldberg, A. Gupta, A. Gupta, D. Jayaraman, J. Lim, J. Malik, R. Martín-Martín, S. Ramamoorthy, D. Sadigh, S. Song, J. Wu, M. Yip, Y. Zhu, T. Kollar, S. Levine,

and C. Finn, "DROID: A Large-Scale In-The-Wild Robot Manipulation Dataset," in *Robotics: Science and Systems XX*, Robotics: Science and Systems Foundation, Jul. 2024.
doi: 10.15607/RSS.2024.XX.120.

[267] K. Pertsch, K. Stachowicz, B. Ichter, D. Driess, S. Nair, Q. Vuong, O. Mees, C. Finn, and S. Levine, "FAST: Efficient Action Tokenization for Vision-Language-Action Models," *arXiv:2501.09747 [cs]*, Jan. 2025.
doi: 10.48550/arXiv.2501.09747.

[268] A. J. Sathyamoorthy, K. Weerakoon, M. Elnoor, A. Zore, B. Ichter, F. Xia, J. Tan, W. Yu, and D. Manocha, "CoNVOI: Context-aware Navigation using Vision Language Models in Outdoor and Indoor Environments," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2024, pp. 13 837–13 844.
doi: 10.1109/IROS58592.2024.10802716.

[269] Y. Kim, D. Kim, J. Choi, J. Park, N. Oh, and D. Park, "A survey on integration of large language models with intelligent robots," *Intelligent Service Robotics*, vol. 17, no. 5, pp. 1091–1107, Sep. 2024.
doi: 10.1007/s11370-024-00550-5.

[270] J. Wang, E. Shi, H. Hu, C. Ma, Y. Liu, X. Wang, Y. Yao, X. Liu, B. Ge, and S. Zhang, "Large language models for robotics: Opportunities, challenges, and perspectives," *Journal of Automation and Intelligence*, vol. 4, no. 1, pp. 52–64, Mar. 2025.
doi: 10.1016/j.jai.2024.12.003.

[271] S. Y. Gadre, M. Wortsman, G. Ilharco, L. Schmidt, and S. Song, "CoWs on Pasture: Baselines and Benchmarks for Language-Driven Zero-Shot Object Navigation," in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Jun. 2023, pp. 23 171–23 181.
doi: 10.1109/CVPR52729.2023.02219.

[272] C. Huang, O. Mees, A. Zeng, and W. Burgard, "Visual Language Maps for Robot Navigation," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, May 2023, pp. 10 608–10 615.
doi: 10.1109/ICRA48891.2023.10160969.

[273] B. Lin, Y. Zhu, Z. Chen, X. Liang, J. Liu, and X. Liang, "ADAPT: Vision-Language Navigation with Modality-Aligned Action Prompts," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Jun. 2022, pp. 15 375–15 385.
doi: 10.1109/CVPR52688.2022.01496.

[274] Y. Long, W. Cai, H. Wang, G. Zhan, and H. Dong, "InstructNav: Zero-shot System for Generic Instruction Navigation in Unexplored Environment,"

arXiv:2406.04882 [*cs*], Jun. 2024.
doi: 10.48550/arXiv.2406.04882.

[275] J. Zhang, K. Wang, R. Xu, G. Zhou, Y. Hong, X. Fang, Q. Wu, Z. Zhang, and H. Wang, "NaVid: Video-based VLM Plans the Next Step for Vision-and-Language Navigation," *arXiv:2402.15852* [*cs*], Jun. 2024.
doi: 10.48550/arXiv.2402.15852.

[276] W. Wu, T. Chang, X. Li, Q. Yin, and Y. Hu, "Vision-language navigation: A survey and taxonomy," *Neural Computing and Applications*, vol. 36, no. 7, pp. 3291–3316, Mar. 2024.
doi: 10.1007/s00521-023-09217-1.

[277] S. Narasimhan, A. H. Tan, D. Choi, and G. Nejat, "OLiVia-Nav: An Online Lifelong Vision Language Approach for Mobile Robot Social Navigation," *arXiv:2409.13675* [*cs*], Mar. 2025.
doi: 10.48550/arXiv.2409.13675.

[278] D. Song, J. Liang, A. Payandeh, A. H. Raj, X. Xiao, and D. Manocha, "VLM-Social-Nav: Socially Aware Robot Navigation Through Scoring Using Vision-Language Models," *IEEE Robotics and Automation Letters*, vol. 10, no. 1, pp. 508–515, Jan. 2025.
doi: 10.1109/LRA.2024.3511409.

[279] Z. Shi, E. Landrum, A. O'Connell, M. Kian, L. Pinto-Alva, K. Shrestha, X. Zhu, and M. J. Matarić, "How Can Large Language Models Enable Better Socially Assistive Human-Robot Interaction: A Brief Survey," *Proceedings of the AAAI Symposium Series*, vol. 3, no. 1, pp. 401–404, May 2024.
doi: 10.1609/aaaiss.v3i1.31245.

[280] L. Bärmann, R. Kartmann, F. Peller-Konrad, J. Niehues, A. Waibel, and T. Asfour, "Incremental learning of humanoid robot behavior from natural interaction and large language models," *Frontiers in Robotics and AI*, vol. 11, p. 1 455 375, Oct. 2024.
doi: 10.3389/frobt.2024.1455375.

[281] 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," *Autonomous Robots*, vol. 47, no. 8, pp. 1087–1102, Dec. 2023.
doi: 10.1007/s10514-023-10139-z.

[282] C. Tang, C. Tang, S. Gong, T. M. Kwok, and Y. Hu, "Robot Character Generation and Adaptive Human-Robot Interaction with Personality Shaping," *arXiv:2503.15518* [*cs*], Mar. 2025.
doi: 10.48550/arXiv.2503.15518.

[283] H. Li, C. Yang, A. Zhang, Y. Deng, X. Wang, and T.-S. Chua, "Hello Again! LLM-powered Personalized Agent for Long-term Dialogue," *arXiv:2406.05925* [*cs*], Feb. 2025.
doi: 10.48550/arXiv.2406.05925.

[284] Z. Zhang, R. A. Rossi, B. Kveton, Y. Shao, D. Yang, H. Zamani, F. Dernoncourt, J. Barrow, T. Yu, S. Kim, R. Zhang, J. Gu, T. Derr, H. Chen, J. Wu, X. Chen, Z. Wang, S. Mitra, N. Lipka, N. Ahmed, and Y. Wang, "Personalization of Large Language Models: A Survey," *arXiv:2411.00027* [*cs*], Oct. 2024.
doi: 10.48550/arXiv.2411.00027.

[285] O. Sotomi, D. Kodi, and A. Arab, "Trust Through Transparency: Explainable Social Navigation for Autonomous Mobile Robots via Vision-Language Models," *arXiv:2504.05477* [*cs*], Apr. 2025.
doi: 10.48550/arXiv.2504.05477.

[286] Z. Wang, B. Liang, V. Dhat, Z. Brumbaugh, N. Walker, R. Krishna, and M. Cakmak, "I Can Tell What I am Doing: Toward Real-World Natural Language Grounding of Robot Experiences," *arXiv:2411.12960* [*cs*], Nov. 2024.
doi: 10.48550/arXiv.2411.12960.

[287] Z. Liu, A. Bahety, and S. Song, "REFLECT: Summarizing Robot Experiences for Failure Explanation and Correction," in *Proceedings of The 7th Conference on Robot Learning*, PMLR, Dec. 2023, pp. 3468–3484.
doi: 10.48550/arXiv.2310.02635.

[288] R. S. Verhagen, M. A. Neerincx, C. Parlar, M. Vogel, and M. L. Tielman, "Personalized Agent Explanations for Human-Agent Teamwork: Adapting Explanations to User Trust, Workload, and Performance," in *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems*, ser. AAMAS '23, International Foundation for Autonomous Agents and Multiagent Systems, May 2023, pp. 2316–2318.

[289] K. Kawaharazuka, Tatsuya Matsushima, Andrew Gambardella, Jiaxian Guo, Chris Paxton, and A. and Zeng, "Real-world robot applications of foundation models: A review," *Advanced Robotics*, vol. 38, no. 18, pp. 1232–1254, Sep. 2024.
doi: 10.1080/01691864.2024.2408593.

[290] Apple Inc., *Introducing Apple Vision Pro: Apple's first spatial computer*, Jun. 2023. [Online]. Available: `https://www.apple.com/newsroom/2023/06/introducing-apple-vision-pro/` Accessed: Jul. 1, 2025.

[291] Meta Platforms, Inc., *Introducing Orion, Our First True Augmented Reality Glasses*, Sep. 2024. [Online]. Available: `https://about.fb.com/news/2024/09/introducing-orion-our-first-true-augmented-reality-glasses/` Accessed: Jul. 1, 2025.

[292] M. Walker, T. Phung, T. Chakraborti, T. Williams, and D. Szafir, "Virtual, Augmented, and Mixed Reality for Human-robot Interaction: A Survey and Virtual Design Element Taxonomy," *ACM Transactions on Human-Robot Interaction*, vol. 12, no. 4, 43:1–43:39, Jul. 2023.
doi: 10.1145/3597623.

[293] J. Wang, C.-C. Chang, J. Duan, D. Fox, and R. Krishna, "EVE: Enabling Anyone to Train Robots using Augmented Reality," in *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology*, ser. UIST '24, Association for Computing Machinery, Oct. 2024, pp. 1–13.
doi: 10.1145/3654777.3676413.

[294] Y. Yang, B. Ikeda, G. Bertasius, and D. Szafir, "Arcade: Scalable demonstration collection and generation via augmented reality for imitation learning," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2024, pp. 2855–2861.
doi: 10.1109/IROS58592.2024.10801810.

[295] N. Nechyporenko, R. Hoque, C. Webb, M. Sivapurapu, and J. Zhang, "ARMADA: Augmented Reality for Robot Manipulation and Robot-Free Data Acquisition," *arXiv:2412.10631 [cs]*, Dec. 2024.
doi: 10.48550/arXiv.2412.10631.

[296] S. Chen, C. Wang, K. Nguyen, L. Fei-Fei, and C. K. Liu, "ARCap: Collecting High-quality Human Demonstrations for Robot Learning with Augmented Reality Feedback," *arXiv:2410.08464 [cs]*, Oct. 2024.
doi: 10.48550/arXiv.2410.08464.

[297] J. de Heuvel, *Supplemental Videos for the Dissertation "Learning Personalized and Human-Aware Robot Navigation"*, Jul. 2025.
doi: 10.5281/zenodo.15825155.

Supplemental Material

Supplementary resources related to this dissertation are available online and are organized by chapter. These materials include demonstration videos hosted on YouTube and corresponding source code or data repositories on GitHub. Please note that the external links provided (e.g., YouTube and GitHub) point to publicly hosted supplemental content. While every effort has been made to ensure their long-term availability, persistent access cannot be guaranteed.

Videos

- Chapter 3: `https://youtu.be/cd442N1MaWc`
- Chapter 4: `https://youtu.be/cYNUFD_rGNE`
- Chapter 5: `https://youtu.be/vS22B3HRdL4`
- Chapter 8: `https://youtu.be/pcO6XPirmUY`

For archival purposes, the videos above have also been deposited in a Zenodo repository [297]: `https://doi.org/10.5281/zenodo.15825155`.

Repositories

- Chapter 6: `https://github.com/hrl-bonn/EnQuery`
- Chapter 5: `https://github.com/HumanoidsBonn/demo_enhanced_morl_nav`
- Chapter 7: `https://github.com/HumanoidsBonn/rlhf_prefnav_interface_study`

List of Figures

List of Tables

List of Algorithms

List of Acronyms

A*	A-Star Path Planning Algorithm
AA	Accumulated Angle
AI	Artificial Intelligence
AMCL	Adaptive Monte Carlo Localization
(rm)ANOVA	(Repeated Measures) Analyis of Variance
AR	Augmented Reality
AUP	Area Under Path
BC	Behavior/Behavioral Cloning
BL	Baseline
CLIP	Contrastive Language-Image Pretraining
CNN	Convolutional Neural Network
CR	Collision Rate
DDPG	Deep Deterministic Policy Gradient
DRL	Deep Reinforcement Learning
D-REX	Disturbance-Based Reward Extrapolation
DWA	Dynamic Window Approach
ESS	Explanation Satisfaction Scale
FMs	Foundation Models
FOV	Field of View
FPV	First-Person View
FQ	Freeform Question
GMDR	Goal-Modulated Diversity Regularization
GUI	Graphical User Interface
GPT	Generative Pre-training Transformer
HER	Hindsight Experience Replay
HSR	Human Support Robot
HRI	Human-Robot Interaction
ICP	Iterative Closest Point
ICL	In-Context Learning (of LLMs)
IRL	Inverse Reinforcement Learning
IV	Independent Variable
KMB	Keyboard-Mouse-Monitor
KL	Kullback-Leibler (Divergence)
LfD	Learning from Demonstration
LLMs	Large Language Models
LSTM	Long Short-Term Memory
M	Mean (used in statistical results)
MCL	Monte Carlo Localization
MDP	Markov Decision Process
MLP	Multi-Layer Perceptron
MO-TD3-HER	Multi-Objective TD3 with HER
MORL	Multi-Objective Reinforcement Learning

PbRL	Preference-based Reinforcement Learning
PD-MORL	Preference-Driven Multi-Objective Reinforcement Learning
ORCA	Optimal Reciprocal Collision Avoidance
PPO	Proximal Policy Optimization
RL	Reinforcement Learning
RLHF	Reinforcement Learning from Human Feedback
RAG	Retrieval-Augmented Generation
RFMs	Robot Foundation Models
RGB-D	Color image (RGB) with associated Depth data
ROS	Robot Operating System
ROI	Region of Interest
RQ	Research Question
RRT	Rapidly-exploring Random Tree
SR	Success Rate
SAC	Soft Actor-Critic
SC	Social Cost (Model)
SD	Standard Deviation (used in statistical results)
SIPA	Subjective Information Processing Awareness (Scale)
TAGD	Temporal Accumulation Group Descriptor
TAM	Technology Acceptance Model
TD	Top-Down
TD3	Twin-Delayed Deep Deterministic Policy Gradient
TR	Timeout Rate
VAE	Variational Autoencoder
VLA	Vision-Language-Action
VLM	Vision-Language Model
VLN	Vision-and-Language Navigation
VR	Virtual Reality
XAI	Explainable Artificial Intelligence
XRL	Explainable Reinforcement Learning

Erklärung zu verwendeten Hilfsmitteln

Bei der Anfertigung dieser Arbeit wurden generative KI-Systeme zur sprachlichen Verfeinerung verwendet.