THÈSE

Pour obtenir le grade de

**DOCTEUR DE L'UNIVERSITÉ GRENOBLE ALPES**

École doctorale : MSTII - Mathématiques, Sciences et technologies de l'information, Informatique
Spécialité : Informatique
Unité de recherche : Laboratoire d'Informatique de Grenoble

# Association de Données Multi-Résidents dans les Maisons Intelligentes : Modélisation et Apprentissage

# Multi-Resident Data Association in Smart Homes: Modeling and Learning

Présentée par :


## Xi CHEN


Direction de thèse :

| | |
|---|---|
| **Dominique VAUFREYDAZ**<br>PROFESSEUR DES UNIVERSITES, UNIVERSITE GRENOBLE ALPES | Directeur de thèse |
| **JULIEN CUMIN**<br>DOCTEUR EN SCIENCES, ORANGE INNOVATION | Co-encadrant de thèse |
| **Fano RAMPARANY**<br>DOCTEUR EN SCIENCES, ORANGE INNOVATION | Co-encadrant de thèse |

Rapporteurs :

**CHRISTOPHE LOHR**
MAITRE DE CONFERENCES HDR, IMT ATLANTIQUE BREST
**DAN ISTRATE**
MAITRE DE CONFERENCES HDR, UNIVERSITE DE TECHNOLOGIE DE COMPIEGNE

Thèse soutenue publiquement le **2 avril 2026**, devant le jury composé de :

| | |
|---|---|
| **SOPHIE DUPUY-CHESSA,**<br>PROFESSEURE DES UNIVERSITES, UNIVERSITE GRENOBLE ALPES | Présidente |
| **DOMINIQUE VAUFREYDAZ,**<br>PROFESSEUR DES UNIVERSITES, UNIVERSITE GRENOBLE ALPES | Directeur de thèse |
| **CHRISTOPHE LOHR,**<br>MAITRE DE CONFERENCES HDR, IMT ATLANTIQUE BREST | Rapporteur |
| **DAN ISTRATE,**<br>MAITRE DE CONFERENCES HDR, UNIVERSITE DE TECHNOLOGIE DE COMPIEGNE | Rapporteur |
| **OLIVIER ROMAIN,**<br>PROFESSEUR DES UNIVERSITES, CY CERGY PARIS UNIVERSITE | Examinateur |
| **SYLVAIN GIROUX,**<br>FULL PROFESSOR, UNIVERSITE DE SHERBROOKE | Examinateur |

Invités :

**JULIEN CUMIN**
DOCTEUR EN SCIENCES, ORANGE INNOVATION
**FANO RAMPARANY**
DOCTEUR EN SCIENCES, ORANGE INNOVATION

# ABSTRACT

Accurate activity tracking in multi-resident smart homes is pivotal for delivering personalized assistance and elderly healthcare. However, relying on non-intrusive ambient sensors (e.g., motion detectors, door contacts) introduces a critical challenge: the Data Association problem, where anonymous binary sensor events must be correctly attributed to specific residents. Existing approaches predominantly rely on filtering techniques that map discrete events into continuous latent spaces, often resulting in semantic misalignment and limited capability to capture complex, non-Markovian behavioral dependencies.

This thesis addresses these limitations by establishing a bottom-up, data-driven framework for multi-resident data association. First, we formalize the problem mathematically and propose a novel Markov Decision Process (MDP) modeling paradigm. Unlike traditional filtering, this framework fundamentally reformulates the task as a sequential decision-making problem. This approach constructs a richer and more complete state representation by explicitly integrating discrete event sequences and historical context, thereby resolving the information bottlenecks inherent in latent-space approximations.

Building upon this foundation, we investigate two distinct policy paradigms. In the realm of supervised data-driven policies, we introduce NEP+Search, a reward-guided strategy based on next-event probability estimation. Subsequently, we propose a holistic Behavioral Cloning (BC) approach utilizing a Transformer encoder. Unlike the search-based method, this approach functions as an end-to-end policy that directly optimizes the alignment between predicted assignment actions and ground-truth labels. Empirical evaluations on the CASAS, MARBLE, and MuRAL datasets demonstrate that these MDP-based methods significantly outperform traditional filtering baselines (e.g., sMRT), particularly in resolving complex simultaneous activities.

To overcome the generalization bottleneck of supervised methods, we advance towards generative reasoning agents. We propose LADA, which reformulates the policy as a text-conditioned generative task. Experiments reveal that LADA achieves zero-shot accuracy superior to supervised experts on the MARBLE dataset, though its performance relies heavily on large-scale models. Finally, to reconcile reasoning power with computational efficiency, we present LLM+BC, which fine-tunes lightweight LLMs (e.g., Qwen-4B) using Low-Rank Adaptation (LoRA). This method achieves state-of-the-art performance on the highly noisy MuRAL dataset, surpassing both zero-shot and traditional supervised baselines.

Collectively, this research provides a comprehensive progression from rigorous mathematical formalization to advanced generative AI solutions, offering scalable, accurate, and privacy-preserving methodologies for future smart environments.

**Keywords** Multi-resident Data Association, Smart Home, Markov Decision Process, Large Language Models, Ambient Intelligence, Deep Learning, Human Activity Recognition.

# RÉSUMÉ

Le suivi précis des activités dans les maisons intelligentes multi-résidents est primordial pour fournir une assistance personnalisée et des soins de santé adaptés aux personnes âgées. Cependant, l'utilisation de capteurs ambiants non intrusifs (tels que les détecteurs de mouvement ou les contacts de porte) introduit un défi critique : le problème d'association de données, où des événements de capteurs binaires et anonymes doivent être correctement attribués à des résidents spécifiques. Les approches existantes reposent principalement sur des techniques de filtrage qui projettent les événements discrets dans des espaces latents continus, ce qui entraîne souvent un désalignement sémantique et limite la capacité à capturer des dépendances comportementales complexes et non markoviennes.

Cette thèse aborde ces limitations en établissant un cadre ascendant et guidé par les données pour l'association de données multi-résidents. Premièrement, nous formalisons mathématiquement le problème et proposons un nouveau paradigme de modélisation basé sur les Processus Décisionnels de Markov (Markov Decision Process (MDP) en anglais). Contrairement au filtrage traditionnel, ce cadre reformule fondamentalement la tâche comme un problème de prise de décision séquentielle. Cette approche construit une représentation d'états plus riche et plus complète en intégrant explicitement les séquences d'événements discrets et le contexte historique, résolvant ainsi les goulots d'étranglement informationnels inhérents aux approximations par espaces latents.

Sur cette base, nous étudions deux paradigmes de politiques distincts. Dans le domaine des politiques supervisées guidées par les données, nous introduisons NEP+Search, une stratégie guidée par la récompense et basée sur l'estimation de la probabilité du prochain événement. Par la suite, nous proposons une approche holistique de Clonage Comportemental (Behavior Cloning (BC) en anglais) utilisant

un encodeur Transformer. Contrairement à la méthode basée sur la recherche, cette approche fonctionne comme une politique de bout en bout qui optimise directement l'alignement entre les actions d'assignation prédites et les étiquettes de vérité terrain. Les évaluations empiriques sur les jeux de données CASAS, MARBLE et MuRAL démontrent que ces méthodes basées sur les MDP surpassent significativement les références de filtrage traditionnelles (telles que sMRT), en particulier pour la résolution d'activités simultanées complexes.

Afin de surmonter les limites de généralisation des méthodes supervisées, nous nous intéressons aux techniques d'agents de raisonnement génératifs. Nous proposons LADA, qui reformule la politique comme une tâche générative conditionnée par le texte. Les expériences révèlent que LADA atteint une précision zero-shot supérieure aux experts supervisés sur le jeu de données MARBLE, bien que sa performance dépende fortement de modèles à grande échelle. Enfin, pour concilier puissance de raisonnement et efficacité computationnelle, nous présentons LLM+BC, qui affine des LLM légers (par exemple, Qwen-4B) en utilisant l'adaptation de bas rang (LoRA). Cette méthode atteint des performances de pointe sur le jeu de données très bruité MuRAL, surpassant à la fois les références zero-shot et les références supervisées traditionnelles.

De manière globale, ces travaux de recherche présentent une progression complète allant d'une formalisation mathématique rigoureuse à des solutions d'IA générative avancées, proposant des méthodologies évolutives, précises et respectueuses de la vie privée pour les futurs environnements intelligents.

**Mots-clés** Association de données multi-résidents, Maison intelligente, Processus décisionnel de Markov, Grands modèles de langage, Intelligence ambiante, Apprentissage profond, Reconnaissance d'Activités Humaines.

# ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my advisors, Prof. Dominique, Dr. Julien, and Dr. Fano. Without their dedicated mentorship, encouragement, and expertise, this thesis would not have been possible. I feel truly blessed to have advisors who cared deeply about the scientific rigor of my work, as well as my personal well-being as I navigated the realities of Ph.D. life. In particular, I am deeply indebted to Fano for extending his mentorship to the ski slopes. Thanks to his patient coaching, I can now skillfully tackle the moguls with confidence. I thank them for all their help and dedication.

I would also like to express my sincere thanks to Christophe Lohr and Dan Istrate for generously lending their time to review this manuscript, and to the entire jury for their insightful evaluation of my thesis.

I also wish to extend my thanks to the current and former members of the M-PSI team at the LIG laboratory, with a special mention to Yangtao, Anderson, and Lykon for their support. I would like to express my sincere gratitude to Mélissa Courla and Sybille Caffiau for their invaluable support and assistance in setting up and managing the DOMUS testbed for the MuRAL dataset. I am also very thankful to all the volunteers who generously dedicated their time to participate in the data collection sessions.

Beyond the university, I am equally grateful to my Orange colleagues. Special thanks to my team and project managers, Julien Riera and Sebastien Deleplace, who were consistently attentive to my needs and ensured I had everything required to succeed in my work. I am immensely grateful to Pauline and Catherine for their warmth and for the many insightful discussions about life in France; their guidance was invaluable in helping me navigate and integrate into this new environment. My

sincere thanks also go to Juan and Yoann for leading the way; their successful defenses provided an excellent model for my own. To Miriam, Leonardo, Xavier, Simon, Théo, and Marc—thank you for the shared journey; you will be defending your own theses sooner than you think! Finally, I want to express my appreciation to all my other colleagues, past and present, who have made Orange such a welcoming and pleasant place to work.

Words cannot fully express my gratitude to my girlfriend, Yongyu, who has stood by me with extraordinary patience and unwavering love throughout the past decade. Her presence has been the absolute foundation of my well-being and the ultimate completion of this thesis.

To my parents: I am profoundly grateful for your selfless devotion and for providing everything within your power to support my aspirations, most notably in making my journey to France possible. It is my greatest privilege to be your child. A special thanks goes to my brother, who has always cheered me all the way from China.

A special thanks goes to my friends Qilong, Xinhao and Yongxin for their ongoing support and for helping me put together a wonderful defense reception.

To everyone who has been part of this journey, simply: *Xie Xie*!

Xi

# CONTENTS

# NOMENCLATURE

## Abbreviations

| | |
|---|---|
| **Acc** | Resident Assignment Accuracy. 52, 57, 58, 62, 78, 80–84, 112, 127 |
| **ADL** | Activity of Daily Living. 1, 4, 5, 21–23 |
| **AI** | Artificial Intelligence. 1 |
| **AP** | Access Point. 20 |
| **BC** | Behavior Cloning. 8, 9, 69–72, 78–83, 87, 88, 92, 110–112, 116–119, 126–129, 133–138, 140 |
| **BCE** | Binary Cross-Entropy. 8, 71, 133 |
| **BLEU** | Bilingual Evaluation Understudy. 53, 54, 56, 57, 61, 62, 78, 80, 82–84, 110, 115, 127 |
| **BPE** | Byte Pair Encoding. 92 |
| **CNN** | Convolutional Neural Network. 4, 22 |
| **CoT** | Chain-of-Thought. xi, xix, 9, 91, 95, 97, 102, 114–118, 120, 122, 127, 129, 134, 136, 138 |
| **CRF** | Conditional Random Field. 4, 22, 26 |
| **CRL** | Correct Run Length. 53, 56–58, 62, 78, 80–84, 110, 112, 115, 116, 127, 128 |
| **CSBN** | Collaborative Sensor Body Networks. 19 |
| **CSI** | Channel State Information. 20 |
| **DAgger** | Dataset Aggregation. 135 |

**DDP** Distributed Data Parallel. 124
**DP** Differential Privacy. 140
**DT** Decision Tree. 4, 22

**FFN** Feed-Forward Network. 92, 93
**FL** Federated Learning. 140
**FMCW** Frequency Modulated Continuous Wave. 20

**GM-PHD** Gaussian Mixture Probability Hypothesis Density. 27, 28, 38
**GRPO** Group Relative Policy Optimization. 138

**HAR** Human Activity Recognition. xvii, 1–5, 8, 11–13, 16–21, 23, 28–30, 35–37, 90, 91, 139
**HAVC** Heating, Ventilating and Air Conditioning. 2
**HMM** Hidden Markov Model. 4, 22, 24–26, 28, 38

**IoT** Internet of Things. 1

**LLM** Large Language Model. xix, 5, 7–9, 12, 23, 32, 36, 90–100, 102, 105, 109–114, 116–119, 121, 124, 126–129, 131, 134, 136, 137, 139, 140
**LoRA** Low-Rank Adaptation. 9, 118, 121, 124, 129, 134
**LOS** Line-Of-Sight. 20
**LSTM** Long Short-Term Memory. 4, 22

**MDP** Markov Decision Process. 7, 8, 38–42, 44, 45, 48, 52, 55–57, 59, 63–65, 67, 70, 71, 88, 89, 91, 95–98, 116, 119, 131–134, 138, 139

**NB** Naive Bayes. 4, 22, 24
**NEP** Next-Event Prediction. 8, 39, 46, 54, 60–63, 67, 68, 70, 77–82, 87, 88, 91, 110–112, 116, 132–134, 138
**NLL** Negative Log-Likelihood. 120
**NLP** Natural Language Processing. 5, 53
**NTP** Next-Token Prediction. 94, 119, 120

**PPO** Proximal Policy Optimization. 94, 137

**RBPF** Rao-Blackwellised Particle Filter. 25, 27, 28, 38
**RCE** Resident Count Error. 54, 57, 61, 62, 64, 78, 80, 82, 84, 110, 114, 115, 127, 129
**RFT** Rejection Sampling Fine-Tuning. 137

| | |
|---|---|
| **RL** | Reinforcement Learning. 137 |
| **RLHF** | Reinforcement Learning from Human Feedback. 90, 94, 95 |
| **RMSNorm** | Root Mean Square Layer Normalization. 74, 75, 93 |
| **RNN** | Recurrent Neural Network. 4, 22 |
| **RoPE** | Rotary Positional Encoding. 74, 93 |
| **SFT** | Supervised Fine-Tuning. 120 |
| **SLM** | Small Language Model. 140 |
| **SwiGLU** | Swish-Gated Linear Unit. 74, 93 |

# LIST OF FIGURES

# List of tables

# CHAPTER 1

# INTRODUCTION

## 1.1 Smart Home Applications

Smart homes refer to residential environments equipped with interconnected sensors, devices, and automated systems designed to enhance comfort, safety, and efficiency. The concept of smart homes can be traced back to the 1970s, when it was initially defined as residential spaces featuring interactive automation systems for lighting, temperature control, household appliances, and security monitoring [141]. With the rapid advancement of Internet of Things (IoT) and Artificial Intelligence (AI) technologies over the past two decades, smart homes have evolved from convenience-oriented automation systems into user-centered intelligent environments capable of supporting a wide range of advanced services. These services primarily focus on the following key domains:

— **Healthcare and Assisted Living:** Driven by global population aging and the rising costs associated with institutional care [155], healthcare has become one of the most critical application domains. By leveraging Human Activity Recognition (HAR) to model Activities of Daily Living (ADLs) such as cooking, bathing, and medication intake, sensor-equipped smart homes can assess residents' functional health status and detect potentially hazardous situations,

including falls, prolonged inactivity, or missed routines, enabling timely interventions or automatic emergency responses [224, 184, 27, 121, 10, 1, 75]. In addition, long-term analysis of activity patterns allows the identification of subtle behavioral changes that may indicate early cognitive or other health decline [85, 21, 158, 224, 194, 39, 211, 156]. Prior studies further show that such systems can support individuals with limited mobility in a non-intrusive manner, reducing social isolation while improving comfort and overall quality of life [40].

— **Home Security and Safety:** Smart homes are increasingly utilized for security monitoring and hazard detection, where HAR strengthens traditional systems by distinguishing normal resident behavior from anomalous activities [150, 4, 82, 151, 69, 74]. For example, HAR can detect suspicious movements or the presence of unfamiliar individuals and issue alerts to residents or relevant authorities [74]. Compared with conventional motion-triggered alarms, HAR-enabled systems interpret action sequences (e.g., breaking a window followed by entering a room) as meaningful intrusion events and can also recognize safety-critical hazards such as fires or kitchen accidents to initiate appropriate responses [74]. Overall, contextual analysis of sensor streams helps differentiate benign events from threats, enabling more proactive and reliable home security.

— **Energy Efficiency:** Energy efficiency represents a major motivation for deploying smart home technologies, as household consumption is intrinsically linked to resident activity. Based on real-time presence and behavior patterns, smart home systems leverage HAR to dynamically adjust environmental controls—such as Heating, Ventilating and Air Conditioning (HAVC), lighting, and appliances [7, 114, 190, 157, 123, 208]. By identifying contexts like unoccupied rooms or idle appliances, these systems significantly reduce energy waste while maintaining user comfort [208]. Moreover, this occupant-centered approach supports smart grid demand-response strategies by optimizing energy-intensive operations based on activity levels, enabling fine-grained energy optimization.

— **Automation:** Smart homes aim to improve everyday convenience through intelligent automation. Even in non-critical scenarios, HAR can enhance user experience by allowing environments to respond proactively to residents' activities. By understanding ongoing behaviors, smart home systems can automate routine tasks, such as starting a coffee machine upon recognizing morning routines or adjusting lighting settings when residents are relaxing. This capability transcends early home automation by minimizing manual interaction and enabling environments to adapt seamlessly to residents' needs through advanced, context-aware behavior modeling [74].

Collectively, the diverse applications described above underscore that the realization of the smart home vision fundamentally hinges on two critical pillars: **Sensing Technologies** and **Human Activity Recognition**. While sensing infrastructures provide the essential data foundation by capturing physical signals and environmental

states, HAR serves as the computational core that translates these raw data streams into meaningful semantic insights. Therefore, the synergy between unobtrusive sensing modalities and accurate recognition algorithms is indispensable for enabling the adaptive, efficient, and user-centered services central to modern smart living.

## 1.2 Sensing Technologies in Smart Homes

Smart home systems rely on a variety of sensing technologies to observe occupants and their interactions with the living environment. These sensing modalities differ in terms of the information they capture, deployment requirements, and their suitability for long-term residential use.

**Wearable** sensors, such as smartwatches, are typically in direct contact with the human body and collect high-precision, user-specific low-level activity signals through integrated sensors, including inertial sensors and heart rate monitors [113, 220]. These devices enable detailed monitoring of individual physiological and physical activities, such as breathing, heart rate, walking, and other forms of movement. However, this sensing paradigm is inherently user-centric and provides limited awareness of the surrounding environment and other occupants in multi-resident settings. As a result, it lacks contextual information about shared spaces and interactions, which makes it less effective for recognizing higher-level, coarse-grained activities [220]. Moreover, the requirement for on-body wear introduces compromises in user comfort and unobtrusiveness, which can hinder long-term adoption in everyday living environments.

**Vision- and audio-based** sensing systems provide rich perceptual information for observing human behavior and spatial context, enabling accurate subject identification and activity understanding [139, 106, 117, 26]. Vision-based approaches typically rely on RGB cameras, depth cameras [102], or infrared cameras [11] to capture a wide range of activities and interactions between residents and their surrounding environment. Audio-based systems commonly employ microphones to recognize activities characterized by distinctive acoustic patterns and, in some cases, to infer speaker identity [66, 129]. Despite their effectiveness, both modalities raise significant privacy and intrusiveness concerns, as they involve capturing identifiable visual or acoustic signals, which pose major barriers to their adoption in private residential environments unless dedicated privacy-preserving mechanisms are employed.

**Radio- and light-based** sensing technologies, including WiFi signal analysis [78, 180, 199, 83, 105, 122], millimeter-wave radar [207, 137], LiDAR point clouds [13, 12, 122], and visible light reflections [131], enable contact-free monitoring by analyzing signal variations induced by human presence and movement. These approaches can capture fine-grained motion patterns and support multi-subject perception, but their performance is often sensitive to environmental changes and interference, and they typically require environment-specific calibration or subject-dependent training.

**Ambient** sensing constitutes a foundational sensing paradigm for smart homes and has been widely adopted in real-world deployments. It relies on environmental sensors such as motion detectors, contact switches, and object interaction sensors to capture interactions between occupants and their surroundings [62, 215, 197]. Ambient sensors are unobtrusive, privacy-preserving, cost-effective, and easy to deploy at scale, making them particularly suitable for long-term residential monitoring. Although ambient sensing provides only indirect observations of human behavior and does not directly encode subject identity, its robustness and practicality have made it a dominant sensing infrastructure in smart home research.

Owing to these advantages, this thesis focuses exclusively on smart home environments equipped with ambient sensing technologies. By restricting our study to ambient sensor data, we aim to address the fundamental challenges arising from indirect and anonymous observations while maintaining strong relevance to realistic and deployable smart home systems.

## 1.3 Human Activity Recognition

HAR functions as the computational core of smart homes, translating raw sensor data into context-aware insights that enable proactive, personalized services beyond static routines. Research in HAR [36] fundamentally relies on defining "activity" at varying levels of abstraction, typically distinguishing between **low-level atomic actions**—simple, context-independent movements like "opening a door"—and **high-level complex activities**, which are goal-oriented sequences such as "cooking" or other ADLs. Consequently, the vast majority of methodologies utilize sensor events representing these low-level atomic actions as fundamental building blocks, with the ultimate objective of inferring high-level ADLs to support intelligent decision-making.

Early approaches to ADL recognition were based on knowledge and ontology-driven methods [45, 133, 215] often requiring significant domain knowledge and manual feature engineering. With the advancement of machine learning, statistical models such as Naive Bayes (NB) [182], Decision Tree (DT) [29], Hidden Markov Models (HMMs) [80] and Conditional Random Fields (CRFs) [192] were widely adopted to capture temporal dependencies and probabilistic transitions between activity states. The introduction of deep learning further advanced the field. Convolutional Neural Networks (CNNs) have been utilized to automatically extract local patterns from sensor activation [171, 86], while Recurrent Neural Networks (RNNs), particularly Long Short-Term Memory (LSTM) networks [172], have proven effective in modeling long-range temporal dependencies in sensor event sequences. Inspired by the success of language models, Bouchabou et al. [35, 34] abstracted sensor events as tokens in a language sequence to explore the correlations between sensor events. They adopted the ELMo [163] model, which is based on bidirectional language modeling, to simultaneously predict preceding and subsequent events in the context, thereby learning

the latent associations and dependency structures among sensor events. Following the success of Large Language Models (LLMs) in Natural Language Processing (NLP), researchers have begun to apply LLMs for HAR [84, 176, 56, 170, 191, 186, 54], exploiting their powerful contextual reasoning and few-shot learning capabilities to model complex event dependencies.

## 1.4 The Multi-Resident Data Association Challenge

Despite extensive research on HAR over the past two decades, studies based on ambient sensors have predominantly focused on single-resident scenarios. In multi-resident environments, ambient sensor–based activity recognition systems face a fundamental challenge: correctly associating each sensor event with the individual who triggered it [197]. This challenge, known as the **Data Association** problem [204, 197, 25, 31, 98, 46], is particularly severe for ambient sensors, which lack explicit identity cues and rely solely on indirect contextual evidence. In shared living spaces, sensor events generated by multiple residents are often temporally interleaved, disrupting the temporal and semantic coherence of individual activity streams. For instance, concurrent movements by two residents across different rooms may produce a merged event sequence that falsely suggests implausible trajectories for each individual. Prior empirical studies consistently show that such misassociations substantially degrade ADLs recognition performance [51, 98, 198, 47, 46], and the problem exacerbates as the number of residents increases, since events triggered by others effectively act as noise for any given individual.

Early work on multi-resident data association in smart home environments was fundamentally shaped by the seminal study of Wilson and Atkeson in 2005 [204]. They were the first to clearly formalize data association as a system-level inference problem, arguing that sensor events must be interpreted in the context of a global state that jointly represents all residents and their activities. By modeling this state explicitly and performing probabilistic filtering over competing hypotheses, their work established a principled framework for reasoning about interleaved behaviors under uncertainty.

In the following decade, as machine learning and later deep learning techniques rapidly advanced, much of the subsequent research shifted its focus toward improving data association through increasingly powerful classifiers and representation learning. Many methods [63, 64, 65, 173, 98, 58] attempted to leverage larger datasets and richer feature representations to directly predict resident identities or activities from sensor events. However, in doing so, they often departed from Wilson and Atkeson's original insight that data association is inherently a global, state-dependent problem. Rather than explicitly modeling the joint system state, these approaches typically performed event-level or locally conditioned inference, relying on learned features to compensate for the lack of structured state modeling. As a result, despite improve-

ments in representation capacity, their reasoning remained short-sighted, with limited ability to enforce long-term temporal coherence or global consistency across residents.

More recent works, notably sMRT [197] and GAMUT [198], mark a conceptual return to Wilson's original modeling philosophy. These methods explicitly reconstruct a global system state and perform filtering over multiple resident hypotheses, thereby restoring the ability to reason about concurrent trajectories and dynamic resident populations. At the same time, they depart from earlier symbolic formulations by representing resident states in continuous latent spaces derived from sensor embeddings. While this latent-state formulation improves robustness and scalability across environments, the underlying dynamics are still modeled using relatively simple linear transformations and Gaussian assumptions. Consequently, despite their stronger state modeling, these approaches make limited use of modern deep learning to capture richer, non-linear temporal dependencies or long-range behavioral patterns.

## 1.5 Research Questions and Contributions

In this thesis, we investigate the problem of multi-resident data association in smart home environments. Specifically, we are interested in how to leverage ambient sensor data to assign each sensor event to the resident who triggered it, thereby enabling independent tracking of multiple residents living in the same environment. This data association process serves as a fundamental prerequisite for downstream tasks such as multi-resident activity recognition, behavior modeling, and long-term monitoring.

The overarching research question addressed in this work is:

***How can ambient sensor data in multi-resident environments be systematically associated with the residents who triggered them, so as to support accurate and scalable per-resident tracking?***

To answer this question, we decompose it into three more focused research questions (RQs) and provide corresponding contributions for each.

**RQ1: How can the data association problem be formally defined and mathematically modeled?** A major challenge in multi-resident data association lies in the lack of a unified mathematical formulation that clearly specifies the objects being modeled and the structure of the decision process. Existing approaches often rely on heuristics or implicit assumptions, making it difficult to analyze or extend them in a principled manner.

**Contribution 1.** We propose a formal mathematical definition of the data association problem grounded in ambient sensor environments. We explicitly distinguish between two levels of mapping: assigning sensor events to residents and assigning sensor events to activity categories. In this formulation, data association is defined

as mapping each event to a subset of residents, while activity recognition is defined as mapping each event to an activity label. By treating these mappings as decision actions, we model the data association process as a Markov Decision Process (MDP), specifying the state, action space, transition dynamics, and reward structure. To the best of our knowledge, this work provides the first complete and principled MDP-based formulation of multi-resident data association.

**RQ2: How can the proposed MDP formulation be combined with modern deep learning techniques to enable scalable, data-driven solutions?** While the MDP formulation provides a theoretical foundation, practical deployment requires methods that can effectively exploit large-scale data and learn complex patterns from sensor streams.

**Contribution 2.** We introduce two complementary learning and inference paradigms built upon the proposed MDP framework. The first paradigm adopts a next-event prediction training strategy, where a deep neural network learns local transition probabilities and inference is performed via heuristic search to recover a globally consistent assignment trajectory. The second paradigm formulates data association as a multi-label classification problem and trains a policy network via behavior cloning to directly predict the resident subset for each event during inference. Both paradigms are instantiated using state-of-the-art Transformer-based architectures, enabling the models to learn rich temporal and contextual representations from data. Together, these approaches demonstrate how the MDP formulation can be scaled up into fully data-driven solutions.

**RQ3: How can general knowledge and reasoning capabilities be incorporated to enable unsupervised generalization across environments and datasets?** Despite strong performance on known datasets, learned models often struggle to generalize to unseen environments with different sensor layouts or resident behaviors.

**Contribution 3.** To address this limitation, we extend the policy learning framework by replacing the trained policy network with an LLM. By leveraging in-context learning and chain-of-thought reasoning, the LLM can perform data association without task-specific training, relying instead on its general knowledge and reasoning abilities. This design enables unsupervised generalization across environments and datasets, opening a new direction for applying foundation models to multi-resident sensing problems.

In summary, this work advances multi-resident data association by providing a unified theoretical formulation, scalable learning-based solutions, and a pathway toward generalizable reasoning-driven approaches.

## 1.6 Thesis Overview

This thesis addresses the challenge of ambient sensor data association in multi-resident environments by establishing a rigorous mathematical framework and progressing from data-driven optimization algorithms to advanced LLM agents. The structure of the thesis is organized as follows:

**Chapter 2** establishes the theoretical groundwork for this research. It begins with a systematic review of the HAR landscape, analyzing various sensor modalities and datasets. Through this comprehensive survey, we identify Data Association as the critical bottleneck impeding accurate activity tracking in multi-resident environments. Consequently, this chapter provides the formal mathematical definitions for both the HAR framework and the Data Association task, establishing a rigorous mapping function that accommodates complex scenarios. Based on this formalization, the chapter critiques the prevalence of filtering-based data association methods (e.g., sMRT [197]) in existing literature, highlighting the fundamental misalignment between their continuous latent space assumptions and the discrete nature of sensor events.

**Chapter 3** proposes the core methodological innovation of this thesis: a novel MDP modeling paradigm. Designed to resolve the limitations identified in the previous chapter, this framework treats sensor events as discrete data points and enables infinite-length context stacking. Building upon this new modeling paradigm, we introduce our first data-driven policy implementation: **NEP+Search**. We formulate the solution as a reward-guided search strategy that employs a Next-Event Prediction (NEP) model to calculate likelihoods based on individual resident histories. Extensive experiments validate that this approach significantly outperforms traditional filtering baselines, confirming the effectiveness of both the MDP modeling paradigm and the search-based policy.

**Chapter 4** addresses the limitations of NEP+Search—specifically its independent treatment of residents and indirect probability modeling—by proposing a holistic Deep Learning policy, referred to as **Behavior Cloning (BC)**. We model the policy directly using a deep neural network with a Transformer encoder backbone, trained via BC. This method processes the joint states of all residents simultaneously and utilizes a multi-label classification framework with Binary Cross-Entropy (BCE) loss. We also introduce a permutation-based data augmentation technique to ensure permutation invariance. Experimental results demonstrate that BC outperforms NEP+Search in complex scenarios involving multiple residents, setting a new benchmark for supervised Data Association.

**Chapter 5** marks a paradigm shift from specific supervised policies to generalized reasoning agents. We introduce **LADA** (LLM-based Autonomous Data Association), which reformulates the decision-making process into a text-conditioned generative policy. By deploying pre-trained LLMs (e.g., GPT-4o) as the core agent, this chapter

explores the use of zero-shot learning and Chain-of-Thought (CoT) reasoning. We demonstrate that LADA achieves robust, unsupervised event allocation in unseen settings without environment-specific training. However, the chapter also identifies a performance gap when applying this zero-shot approach to smaller-scale LLMs.

**Chapter 6** synthesizes the strengths of the previous approaches to address the limitations of smaller LLMs and the challenges of complex data distributions. We introduce the **LLM+BC** method, which applies the BC training paradigm to fine-tune smaller open-source models (specifically Qwen3-4B) using Low-Rank Adaptation (LoRA). This chapter demonstrates that fine-tuning enables lightweight models to transition from pure commonsense reasoning to data-driven pattern recognition, achieving state-of-the-art performance on complex datasets like MuRAL and significantly enhancing the capabilities of 4B-parameter models.

Finally, **Chapter 7** summarizes the contributions of this thesis, discusses the implications of the proposed methods, and outlines potential directions for future research in multi-resident activity tracking.

# CHAPTER 2

# LITERATURE REVIEW ON MULTI-RESIDENT ACTIVITY RECOGNITION AND DATA ASSOCIATION

## 2.1 Introduction

Human Activity Recognition (HAR) has emerged as a cornerstone technology for enabling proactive services in smart environments, ranging from healthcare monitoring to automated energy management. While single-resident HAR has achieved significant maturity, real-world deployments often involve *multi-subject environments*—such as shared households or assisted living facilities—where the presence of multiple individuals introduces complex social interactions and interleaved event streams. Recognizing "who is doing what" in these settings remains a non-trivial challenge that sits at the intersection of signal processing, pattern recognition, and temporal reasoning.

The objective of this chapter is to provide a comprehensive review of the current landscape of multi-subject HAR and to identify the critical gaps that this thesis aims to bridge. The discussion is organized as follows:

— **Formalization and Taxonomy:** We first address the lack of a unified mathematical framework for multi-subject HAR. By analyzing how different studies

categorize activities (e.g., concurrent, sequential, or collaborative), we establish a rigorous formalization that serves as the foundation for the subsequent chapters.

— **Sensing Modalities:** We evaluate the strengths and weaknesses of various sensing paradigms, including wearable, vision, and radio-based systems. We specifically justify our focus on *ambient sensing* as a privacy-preserving and cost-effective solution for long-term smart home monitoring.

— **Algorithmic Paradigms:** We delve into the two dominant computational strategies in ambient-based HAR: *Data Association-based methods*, which attempt to disentangle the mixed event stream before classification; and *Data Association-free methods*, which treat the problem as a multi-label or multi-task learning challenge.

By synthesizing these perspectives, this chapter highlights the inherent limitations of current Markovian state representations and the scalability issues of traditional classifiers. These insights directly motivate our research into higher-order Markov models and Large Language Models (LLMs) as robust alternatives for modeling the intricate temporal and spatial dependencies of multi-resident activities.

## 2.2 Definition and Formalization of Human Activity in Sensory Environments

Defining and formalizing human activities in a sensory environment is crucial for HAR, as it largely determines the scope of analysis, ensures consistency in methodology, and provides a rigorous foundation for designing and evaluating algorithms. However, existing research on definition and formalization [31, 215, 117, 91] suffers from three fundamental limitations in this regard:

**(1) Lack of unified formalization beyond terminology classification.** Most existing works focus on classifying activities based on terminology, without providing a comprehensive mathematical formalization that generalizes across activity types. For instance, Benmansour et al. [31] categorizes activities as *Sequential*, *Interleaving*, *Concurrent*, *Parallel*, and *Collaborative*, in an attempt to cover a wide spectrum of activity structures in both single- and multi-resident environments. Similar typologies are used in [215, 117, 91]. While such taxonomies help capture the complexity of real-world activities, they typically avoid defining shared underlying structures across categories. As a result, methods built upon these formalisms often struggle to generalize across different activity types or sensor settings. For example, methods designed under the sequential formation often assume that activities can be represented as ordered sensor event chains (e.g., "open fridge $\rightarrow$ use stove $\rightarrow$ wash dishes"), which enables modeling through simple temporal transitions or Markov chains [132]. However, when the same framework is applied to concurrent activities (e.g., "using the stove while fetching

ingredients from the fridge"), the representation no longer holds: concurrent streams break the single linear ordering assumption, requiring either multiple parallel models or an ad-hoc redesign of the state representation. This incompatibility illustrates how taxonomy-based categorizations fail to provide a unified structure across activity types.

**(2) Inconsistencies in terminology across studies.** Even when activity types are defined, their interpretations vary significantly across studies, making it difficult to compare or unify approaches. For example, *concurrent* activities are defined in [215] as activities performed simultaneously by different subjects, while in [91] they refer to multiple activities performed by the same subject at the same time. Conversely, the term *parallel* is used in [31, 117] to describe the former case. Furthermore, when parallel activities involve interdependence among multiple subjects, they are sometimes reclassified under terms like *cooperative*, *collaborative*, or *group* activities, depending on the level of interaction. These inconsistencies hinder the development of standardized datasets, models, and evaluation protocols.

**(3) Lack of bottom-up, compositional modeling.** Existing definitions [31, 215, 91] of activities are often presented as top-level abstractions, with little attention paid to their decomposition into lower-level primitives such as *actions*, or to their grounding in observable physical signals. Specifically, there is a notable gap in connecting abstract concepts like activities to concrete sensory phenomena such as sensor readings and sensor events. This disconnect leads to inconsistencies in how data is represented (e.g., whether a sample corresponds to a raw reading or a reading change event), the level of granularity at which activity recognition is performed (e.g., per reading, per event, or per segment), and the segmentation strategies adopted (e.g., assigning labels to individual events versus entire activity segments). These modeling choices are rarely justified or discussed in a principled manner, further hindering methodological comparability across studies.

To address these limitations, we propose a concise and generalizable mathematical formalization of human activities in sensory environments. Building upon insights from existing research while aiming for consistency across sensor modalities and subject configurations, our formulation is designed to be both expressive and efficient. It captures the hierarchical structure of activities, from raw sensor readings to events, actions, and composite activities, and serves as a unifying foundation for various HAR tasks. Based on this formalization, we also define the core problems in HAR, including event-subject data association and activity classification, in a principled and coherent manner.

### 2.2.1 Sensory Environment

***Sensors*** are devices designed to capture physical properties of their surrounding environment and convert them into measurable signals, such as electrical outputs. In the context of HAR, existing research typically categorizes sensors based on their

underlying sensing technologies into five main types: **wearable-based, ambient-based, radio-based, vision-based, and audio-based.** Each type of sensor provides different levels of granularity and types of information, contributing uniquely to the recognition process. A detailed comparison of these sensor categories will be presented in Section 2.3.

A ***sensory environment*** refers to a physical setting instrumented with a set of sensors. Formally, it is defined as a set $\mathrm{E} = \{o_i\}_{1 \le i \le I}$, where each $o_i$ denotes a sensor along with its associated attributes (e.g., type, location, and capabilities), and $I$ is the total number of sensors deployed in the environment.

Within a sensory environment, a set of ***subjects*** $\mathrm{U} = \{u_k\}_{1 \le k \le K}$ can enter, exit, and perform various activities. Here, $K$ denotes the total number of distinct subjects that have appeared up to time $t$. As time evolves, $K$ is a non-decreasing function of $t$, since new subjects may appear but previously observed subjects remain counted.

### 2.2.2 Action as Sensor Event

A ***sensor reading*** refers to the physical state captured by a sensor at a specific point in time. For an ambient sensor $o_i \in \mathrm{E}$, let $r_t^{(i)}$ denote its reading at time $t$, which may represent a binary state (e.g., ON/OFF), a numerical value (e.g., temperature or light intensity), or a modality-specific measurement (e.g., an image frame captured by a camera or an audio segment recorded by a microphone).

A ***sensor event*** is defined as the change between two readings. Formally, we define a sensor event as a tuple:

$$e_t = (o_i, c_t), \tag{2.1}$$

where $o_i \in \mathrm{E}$ is a sensor in the environment, and $c_t$ is the change in its state, computed as:

$$c_{t_2} = \Delta r^{(i)} = r_{t_2}^{(i)} \ominus r_{t_1}^{(i)}, \tag{2.2}$$

with $t_1 < t_2$ denoting two time points at which the sensor readings are recorded, and $\ominus$ meaning computing the difference between two readings. The form of $c_t$ (e.g., binary transition, numeric difference, or categorical shift) depends on the sensor type.

An ***action*** refers to a subject's behavior that induces a physical change in the environment, resulting in a measurable difference in sensor readings. In other words, an action is defined as any human behavior in the sensory environment that leads to a sensor event. For example, in the case of a camera, an action manifests as a change in the light captured between two frames because of subjects' movement. For wearable inertial sensors, actions correspond to changes in acceleration, orientation, or position. For ambient sensors, actions occur when a subject interacts with the environment in a way that alters a sensor's state. For instance, a door sensor switching from "open" to "closed" reflects the subject's action of closing the door.

Table 2.1 – Categories of sensing modalities used in multi-subject HAR

| Sensor category | Examples | Detectable Infomation |
|---|---|---|
| Wearable | Inertial sensors (e.g., accelerometers, gyroscopes) | Subjects' low-level physical activities |
| | Physiological sensors (e.g., blood pressure sensor) | Changes in the subjects' physiological states |
| Ambient | Binary (e.g. motion sensors, magnetic sensors, pressure mats) | Interactions of the subjects with their surrounding environment |
| | Non-binary (e.g., temperature sensors) | Changes in the subjects' surrounding environment |
| Radio | WiFi, RFID, mmWave, Visible Light Sensing, LIDAR | Subjects' fine-grained physical movements |
| | BLE, UWB | Subject's position in the environment |
| Vision | RGB cameras, depth cameras, infrared cameras | Any type of activity and interaction between the subjects and their surrounding environment |
| Audio | Microphones installed on virtual assistants | Activities with specific audio patterns |
| | | Subjects' presence |

It is important to recognize that not all real-world behaviors of residents can be captured by the available sensors. The observability of an action is determined by the placement, type, and coverage of the deployed sensing infrastructure. As summarized in Table 2.1, different sensor modalities vary in their ability to detect physical changes. In our formalization, we restrict the notion of actions to those that are observable. For example, if a camera is installed only in the living room, then any actions performed in the kitchen will not generate sensor events and thus will not be represented in the action sequence.

### 2.2.3 Association between Events and Subjects

When the sensory environment contains more than one subject (i.e., $K > 1$), associating sensor events with responsible subject(s) is non-trivial. Unlike in single-subject settings where each event can be unambiguously attributed, multi-subject environments introduce uncertainty and ambiguity in identifying which subject triggered a given event. For example, vision-based sensors may require face recognition [3] to identify individuals, radio-based sensors often rely on gait analysis [116, 71], and ambient-based sensors typically require reasoning over the sequence and semantics of triggered events [197, 198, 48].

To address this challenge, the problem of ***Data Association*** [204, 197, 25, 31, 98, 46] is defined as the task of assigning each observed sensor event to a corresponding subset of subjects who may have caused it. Formally, for any given sensor event $e_t$ at the time step $t$, the goal is to determine the set of subjects responsible for triggering this event by constructing a mapping function:

$$M_U : e_t \mapsto \mathcal{U}_t, \tag{2.3}$$

where $\mathcal{U}_t \subseteq \mathrm{U}$ is a subset of the total subject set $\mathrm{U} = \{u_k\}_{1 \le k \le K}$.

Depending on the sensor modality and the nature of the event, $\mathcal{U}_t$ can be an empty set, a single subject, or multiple subjects. The empty-set case corresponds to events not attributable to any resident, which we uniformly treat as **noise**. Such noise may arise from sensor malfunction, measurement errors, or natural environmental changes (e.g., daily variations in temperature or illumination that trigger thermometer or light sensors). A single-subject case corresponds to actions such as one person opening a door, whereas multi-subject cases occur when several people interact within a shared space.

### 2.2.4 Activity as Event Sequence

An ***activity*** is composed of an intentional, conscious, and subjectively meaningful sequence of actions performed by one or more subjects [117]. Since each action leads to a sensor-detectable event, we represent an activity as a time-ordered sequence of sensor events:

$$A = [e_t]_{1 \le t \le T}, \tag{2.4}$$

where $e_t$ is a sensor event occurring at time step $t$, and $T$ denotes the time of the last event in the sequence.

For example, a cooking activity may involve a sequence of events such as opening the refrigerator, retrieving ingredients, and using the stove, potentially performed by multiple subjects. These events, captured by various sensors, reflect the physical actions that constitute the activity.

As illustrated in Figure 2.1, an activity is considered a ***multi-subject activity*** if it involves more than one subject. Formally, for activity $A$, let $\mathcal{U}_t = M_U(e_t)$ be the set of subjects associated with event $e_t$. Then $A$ is a multi-subject activity if

$$\left| \bigcup_{t=1}^{T} \mathcal{U}_t \right| > 1. \tag{2.5}$$

### 2.2.5 Association between Events and Activities

In practical applications, what one can directly observe from the environment are sensor events rather than complete activity sequences. As a result, recognizing which activity is being performed based on observed sensor events becomes a key task, commonly referred to as HAR.

Let $C = \{a_j\}_{1 \le j \le J}$ denote a predefined set of possible activity classes, where each $a_j$ corresponds to a semantic activity label (e.g., *cooking*, *bathing*, *sleeping*). Given a sequence of sensor events $\{e_t\}_{1 \le t \le T}$, the goal is to determine which activity each event contributes to. We define an activity classification mapping:

$$M_A : e_t \mapsto a_j, \quad a_j \in C, \tag{2.6}$$

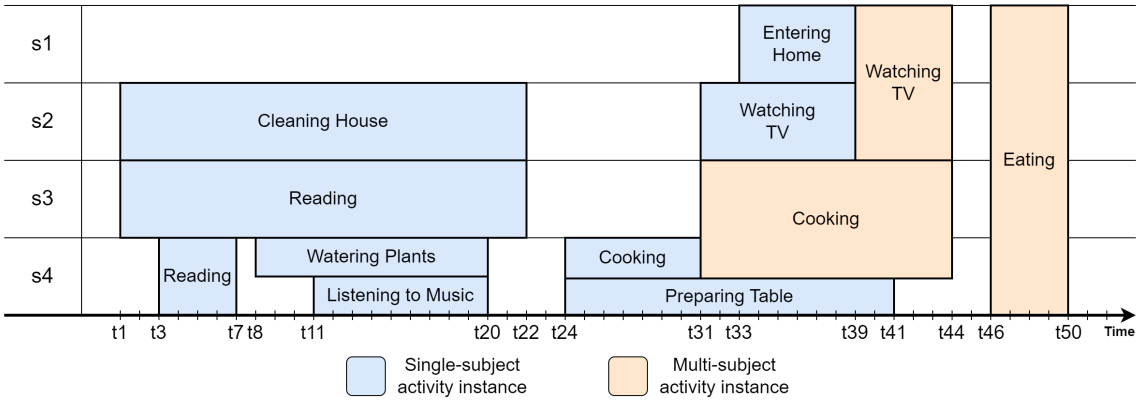



Figure 2.1 – Single- and multi-subject activity instances in a smart-home environment with four subjects. Intervals without labels emulate a missing ground truth, as commonly found in HAR datasets.

which classifies each event $e_t$ into a candidate activity class $a_j$.

This mapping enables the grouping of temporally and semantically related events into coherent activity segments. Formally, an activity segment of class $a_j$ can be reconstructed from the set of events associated with it:

$$A_{a_j} = [e_t \mid M_A(e_t) = a_j]_{1 \leq t \leq T}. \tag{2.7}$$

This formulation provides a bridge between low-level sensor observations and high-level semantic activity understanding, which is essential for downstream reasoning tasks in smart environments.

## 2.3 Sensing Modalities

The choice of sensing modalities in a smart environment critically affects the granularity, semantics, and usability of sensor events for HAR. Different sensor types offer trade-offs in terms of intrusiveness, coverage, and the type of information captured. In this section, we compare various sensing modalities used in HAR, highlighting their strengths and limitations. Based on this comparison, we identify the most relevant modalities to our study and briefly review representative HAR approaches for those outside our scope.

### 2.3.1 Comparison of Sensing Modalities

Table 2.1 provides an overview of commonly used sensing modalities in multi-subject HAR, along with representative sensor types and the types of information they can capture. These modalities differ significantly in terms of sensing mechanisms and

Table 2.2 – Pros and cons of different methods for multi-subject HAR

| Method | Pros | Cons |
|---|---|---|
| Wearable | Easy user identification | Can mainly recognize multi-subject low-level physical activities<br><br>Require subjects to carry devices |
| Ambient | Can exploit interaction events between subjects and environment for HAR | Require data association techniques in order to identify subjects |
| Radio | Can exploit fine-grained subjects' movements for HAR | Prone to environmental interferences and changes<br><br>Require training on specific subjects in order to identify them |
| Vision/Audio | High accuracy in activity recognition, subjects identification, and time intervals detection | Privacy issues (perceived intrusiveness and legal issues unless specific privacy-preserving hardware/software techniques are provided) |

the nature of the observable data, ranging from low-level physical signals to high-level semantic interactions.

While Table 2.1 outlines what each modality is capable of detecting, a deeper understanding of their practical suitability for multi-subject HAR requires examining their methodological strengths and limitations. Inspired by a previous survey [117], Table 2.2 compares the pros and cons of different sensing-based approaches, highlighting their effectiveness, usability, and deployment challenges in real-world multi-subject settings. Hybrid combinations of sensing modalities are not included in the table, as each combination has specific pros and cons.

As shown in Table 2.2, wearable sensors offer a straightforward means for identifying individual subjects, but are limited to recognizing low-level physical activities and require subjects to carry or wear dedicated devices—an assumption that does not hold in many real-world scenarios, such as smart home environments. Radio-based methods enable fine-grained motion tracking without requiring subjects to carry devices, but they are highly sensitive to environmental changes and often require per-subject training to achieve reliable identification, limiting their generalization ability. Vision- and audio-based approaches provide highly informative data and rich semantic understanding but raise significant privacy concerns, especially in home environments, and often face legal and ethical deployment barriers. In contrast, ambient sensors offer a non-intrusive and widely deployable alternative that captures user interactions with the environment through naturally occurring events (e.g., door opening, appliance usage, motion detection). However, their primary challenge lies in the lack of inherent subject identity information—they typically cannot distinguish which subject triggered a given sensor event. This necessitates the development of robust data association methods to link sensor events to the correct individuals, especially in multi-resident scenarios. Based on these considerations, this research focuses on ambient-based sensing due to its practicality and pervasiveness in smart environments, and we address the core challenge it presents: associating sensor events with individual subjects in multi-subject settings.

Table 2.3 – A categorization of methods in existing literature based on the sensing modality and key technical strategy.

| Modality | Key technical strategy | References |
|---|---|---|
| Ambient | Multi-label classification | [130, 111, 17, 103, 104, 46, 127, 128, 41, 115, 181] |
| | Probabilistic Graphical Models | [32, 188] |
| | Subject separation | [42] |
| | Data Association | [159, 215, 93, 197, 198] |
| Radio / Light | WiFi CSI analysis | [78, 180, 199, 83, 105, 122] |
| | mmWave point clouds | [207, 137] |
| | LIDAR point clouds | [13, 12, 122] |
| | Skeleton reconstruction | [119] |
| | Visible light reflections | [131] |
| Wearable | Distributed Detection | [205, 87, 118] |
| Multimodal | Probabilistic Graphical Models | [162, 14, 196, 90, 195, 76] |
| | Micro-localization | [44, 77, 112, 23, 24, 161] |

Table 2.3 categorizes representative HAR methods in the literature based on two dimensions: the sensing modality used, and the core technical strategy adopted. As shown, existing work spans a wide range of modalities. In the following, we briefly review the main approaches adopted for each sensing modality to highlight their technical characteristics and common modeling [25]. Given our focus on ambient sensing in this work, methods based on ambient sensors—particularly those addressing multi-subject scenarios through data association—are discussed in deeper detail in the Section 2.4.

### 2.3.2 Wearable-based methods

Thanks to mobile and wearable devices worn by users (e.g., smartphones, smartwatches), it is possible to collect sensor data (e.g., from inertial sensors) that are automatically associated with the corresponding user. While this setting makes it possible to easily recognize parallel single-subject activities (different users are separately performing different activities), it is more challenging to detect multi-subject activities (multiple users jointly performing an activity).

As shown in Table 2.3, a few works proposed distributed approaches to recognize multi-subject identified-group activities. Indeed, it is possible to rely on opportunistic networks to dynamically group wearable devices that are close in space, in order to combine their data to capture the occurrence of group activities. Peer-to-peer strategies have been proposed for outdoor crowded settings, where a decentralized approach is necessary for the sake of scalability (e.g., in emergency situations). Such systems rely on direct communication between devices of different users to exchange sensor data, compute the pairwise similarity of user data patterns, and finally propagate information about crowd behavior [205, 87]. More recently, Collaborative Sensor Body Networks (CSBN) have been proposed to combine data from wearable devices of multiple users [118] for multi-subject activity recognition.

### 2.3.3 Radio- and light-based methods

The analysis of the signal from radio- and light-based devices is becoming popular in the HAR domain due to their contactless nature. The signals generated by such devices are susceptible to changes due to the environment, and they have the potential to capture human activities since human bodies absorb, reflect, and scatter signals [70]. As described in Table 2.3, multiple methods based on these sensing modalities have been used to recognize parallel-single subject and multi-subject unidentified-group activities.

Among the many types of radio- and light-based devices used for multi-subject HAR (e.g., mmWave radars [110, 206, 209], LIDAR [122, 13, 12], Visible Light Sensing [131], ), WiFi Access Points (APs) are the most commonly available in many smart environments, including smart homes [225]. Due to its robustness to multi-path effects, the majority of approaches propose methods based on the analysis of the Channel State Information (CSI) signal [199].

These approaches typically rely on WiFi APs acting as transmitters and on receiver devices (e.g., laptops). Since humans move and interact in the environment while performing activities, the CSI signal reflected by the human body will interfere with the signals in the Line-Of-Sight (LOS) path. The receiver is in charge of analyzing such small changes in the signal for multi-subject HAR. In particular, by combining signal processing techniques and deep learning models, it is possible to estimate the number of users in the environment and infer the activity performed by each user [78, 180, 83]. Existing approaches for multi-subject HAR are capable of isolating signals for each user so that it is possible to recognize activities for each subject in the environment. In particular, some approaches specifically focus on gestures (e.g., pushing, kicking, pulling) performed by a target subject in a multi-subject environment [145, 100].

A major drawback of WiFi-based HAR is that the resolution of captured data is physically limited by the large wavelength (5 GHz) and narrow bandwidth (20 MHz) of WiFi signals. Therefore, there has been increasing interest in radio technologies with higher frequencies and larger bandwidths. These technologies widely use Frequency Modulated Continuous Wave (FMCW) radars which transmit an electromagnetic wave with a linearly increasing frequency over time and capture signals reflected by objects. Using an FMCW radar operating within the frequency range of 5.4 to 7.2 GHz, it is possible to generate 3D skeletons from raw radio signals for multiple subjects [119, 222], which can then be used to recognize physical activities. Millimeter-wave FMCW radars are one of the common types of high-resolution radio signals used for HAR. These radars typically operate within the frequency range of 76 GHz to 81 GHz or 60 GHz to 64 GHz, providing a bandwidth of 4-5 GHz. Multiple works have generated silhouettes and 3D skeletons of multiple subjects using raw signals from cascaded millimeter-wave radars [110, 206, 209]. Point cloud mmWave data have also been used to track multiple subjects and identify them based on gait [144, 101, 125, 223, 213, 107, 105]. Some studies focus on detecting the breathing and heart rates of

multiple subjects with mmWave signals [6], or identifying atomic gestures of multiple subjects [9].

While some works focus on HAR with mmWave signals [207, 137], most radio-based works surveyed in this paper actually tackle other related problems: tracking, pose estimation, gait, subject identification, etc. Nevertheless, these information can be used as preliminary context analysis for HAR systems: tracking and subject identification can be used for data association approaches; postures are directly related to human activites; etc. In general, activities that involve physical movement of multiple subjects in close proximity will benefit from radio-based works, while activities that involve interactions with specific devices for example will be better captured by environmental approaches. Moreover, trade-offs between resolution, robustness, and cost, are central problems, which lead in particular to a limitation in the number of available datasets. As such, the number of works on multi-subject HAR with radio and light-based approaches is growing quickly but still quite limited.

### 2.3.4 Vision-based methods

Thanks to vision-based approaches, it is possible to detect with high accuracy human activities performed both by single subjects and groups of subjects, as well as to identify the involved subjects and the time intervals in which such activities occur. Since this thesis is not focused on video-based methods, relevant methods can be referred to in existing surveys [117, 26].

## 2.4 Ambient-based Activity Recognition

Ambient sensors, due to their non-intrusive nature and low energy consumption, have become a crucial modality for perceiving residents' daily activities in smart home environments. These sensors are often strategically deployed throughout the smart environment. As residents interact with the environment, their actions trigger sensors, resulting in the generation of ambient sensor data. The collected data can be leveraged for a wide range of downstream applications, including health monitoring [224, 194, 39, 211, 156, 184, 27, 121, 10, 1, 158, 21], home security [150, 4, 82, 151, 69], building automation and energy saving [7, 114, 190, 157, 123, 208], and even inferring residents' communication availability [68, 67]. These applications heavily rely on algorithms that can extract meaningful contextual information from ambient sensor data. Among various types of context, residents' Activities of Daily Living (ADLs) represent one of the most fundamental and informative aspects.

In the following, we present a detailed discussion of ambient sensor-based HAR methods, organized according to the number of subjects involved and the specific task being addressed. We divide the discussion into three categories. We begin with activity classification in single-subject settings, where the focus is on how prior work

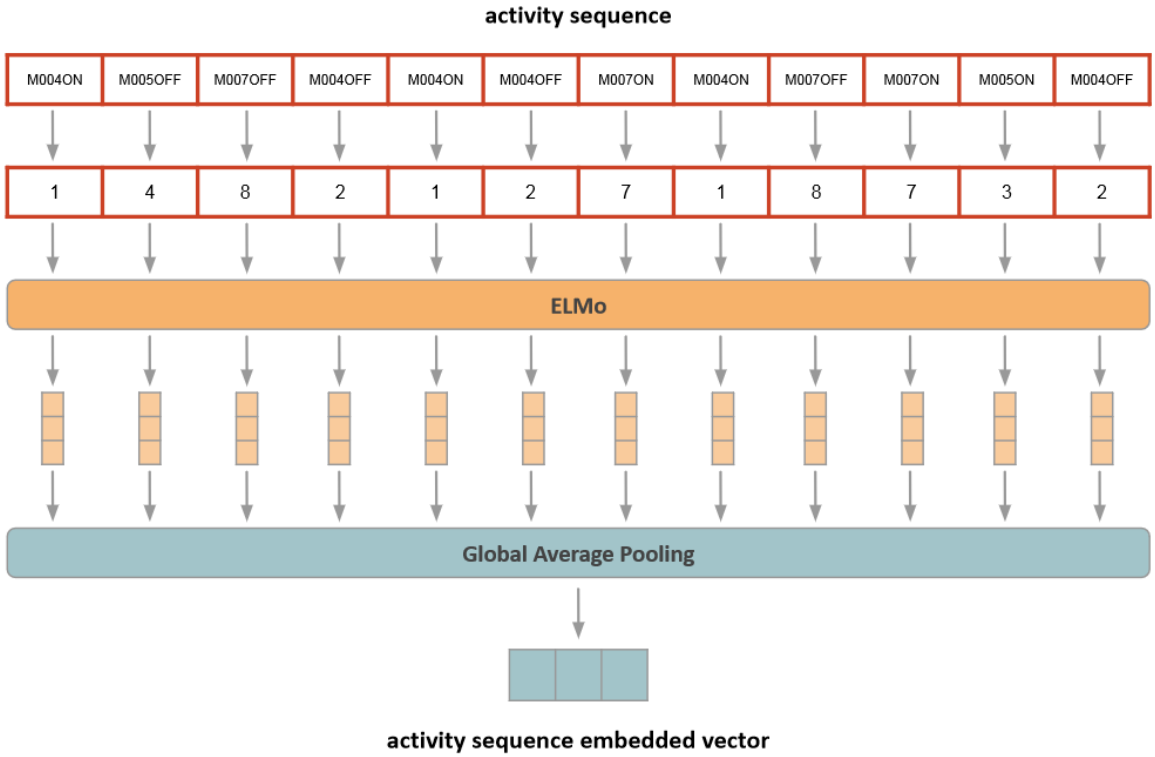



Figure 2.2 – ELMo Activity Sequence Embedding Model. Reproduced from Bouchabou et al. [34].

processes sensor data and learns sequential patterns for activity recognition. We then extend to multi-subject environments, examining how these single-subject models have been adapted or restructured to address the challenge of Data Association—i.e., determining which subject is responsible for each observed sensor event. Finally, we consider activity classification in multi-subject settings, where the goal is to recognize complex, and possibly collaborative, activities involving multiple individuals.

### 2.4.1 Single-Subject Activity Classification

Early approaches to ADL recognition were based on knowledge and ontology-driven methods [45, 133, 215] often requiring significant domain knowledge and manual feature engineering. With the advancement of machine learning, statistical models such as Naive Bayes (NB) [182], Decision Tree (DT) [29], Hidden Markov Model (HMM) [80] and Conditional Random Field (CRF) [192] were widely adopted to capture temporal dependencies and probabilistic transitions between activity states. The introduction of deep learning further advanced the field. Convolutional Neural Networks (CNNs) have been utilized to automatically extract local patterns from sensor activation [171, 86], while Recurrent Neural Networks (RNNs), particularly Long Short-Term Memory (LSTM) networks [172], have proven effective in modeling long-range temporal dependencies in sensor event sequences. Inspired by the success

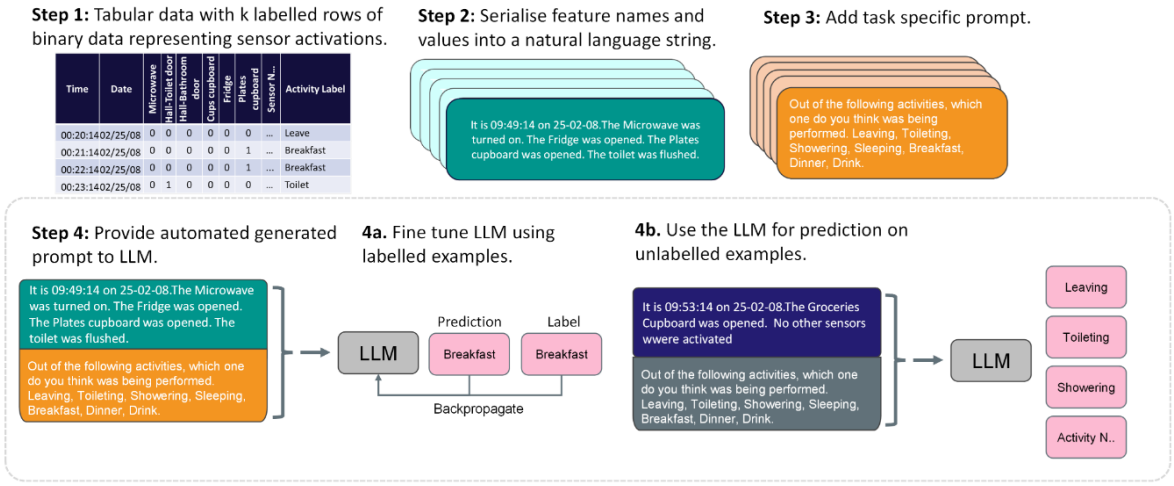



Figure 2.3 – Pipeline from Binary to Text Data and Using LLMs for the Final Classification [56].

of language models, Bouchabou et al. [35, 34] abstracted sensor events as tokens in a language sequence to explore the correlations between sensor events. As illustrated in Figure 2.2, they adopted the ELMo [163] model, which is based on bidirectional language modeling, to simultaneously predict preceding and subsequent events in the context, thereby learning the latent associations and dependency structures among sensor events. Following the success of LLMs in natural language processing, researchers have begun to apply LLMs for activity recognition [84, 176, 56, 170, 191, 186, 54], exploiting their powerful contextual reasoning and few-shot learning capabilities to model complex event dependencies. Figure 2.3 illustrates an example of finetuning an LLM to perform activity classification [56].

### 2.4.2 Multi-Subject Data Association

Despite the methodological advances in single-subject environments, sensor data in multi-subject settings from different residents can overlap or interfere with each other, making activity recognition significantly more challenging. This introduces the problem of Data Association [204, 197, 25, 31, 98, 46], which refers to the task of correctly assigning sensor events to their corresponding residents. Earlier empirical studies have shown that good data association can greatly improve the accuracy of multi-resident ADL [51, 98, 198, 47, 46].

Following a detailed review of Data Association methods developed over the past two decades, a clear line of methodological progression emerges: the modeling of system state, which refers to a representation that records all relevant information needed to characterize the system at a given time, such as subjects' locations and activities in an HAR system. In the following, we organize our discussion around how different approaches formalize and utilize system state in order to perform data association, highlighting key design choices and their implications.

#### 2.4.2.1 Stateless Methods

Earlier approaches to multi-resident data association primarily treated the task as a direct classification problem. Crandall and Cook [63, 64] applied a NB classifier to each individual sensor event, assigning it to the most likely subject identity. However, this naive approach has been shown to be limited in practice, as it tends to assign events to the subject who most frequently triggered the corresponding sensor in the training data. In real-world environments, sensors are typically non-user-specific. For example, a living room sofa can be used by any resident. If the system always assigns sofa-related events to Alice merely because she historically used the sofa most frequently, it will fail to robustly handle cases where Bob uses the same sofa.

The deeper issue lies in the stateless nature of this method: it classifies each event in isolation, without leveraging temporal or contextual dependencies. In contrast, one can easily imagine a scenario where, if the system has just assigned an event in Alice's bedroom to Alice, then an immediately following sofa event in the living room is less likely to be Alice's and more likely to belong to Bob. This type of contextual reasoning requires maintaining a notion of system state: information that captures the dynamic configuration of the environment, including the number of active residents, their current activities, and recent event history. The richer the state representation, the more reasonable the inferences the system can make.

At the same time, the system state is not directly observable. In another word, it must be inferred. This creates a natural feedback loop: Data Association depends on the current state, and the state itself depends on past data associations. This mutual dependency motivates the use of temporal models such as Markov chains [132], which leverage prior state information to inform the inference of the current or next state.

#### 2.4.2.2 First-Order Markov State Methods

Early in 2005, Wilson and Atkeson [204] systematically defined the *Data Association* problem in multi-resident environments and emphasized the necessity of explicit state modeling. They represented the system state at time $t$ as the collection of all individual resident states:

$$s_t = \{s_t^1, s_t^2, \ldots, s_t^K\},$$

where $K$ is the number of residents, and each resident's state $s_t^i$ includes the resident's location (inferred from the room associated with a triggered sensor) and activity type (e.g., *moving* or *not moving*).

Under the *first-order Markov assumption*, each resident's current state depends only on their own previous state. The transition dynamics of each individual were formalized using an HMM, where the hidden state is the location-activity pair of a resident, and the observation is the corresponding binary sensor event. State transition probabilities are determined by motion patterns, and the observation model is defined by the likelihood of a sensor being triggered given the resident's current state.

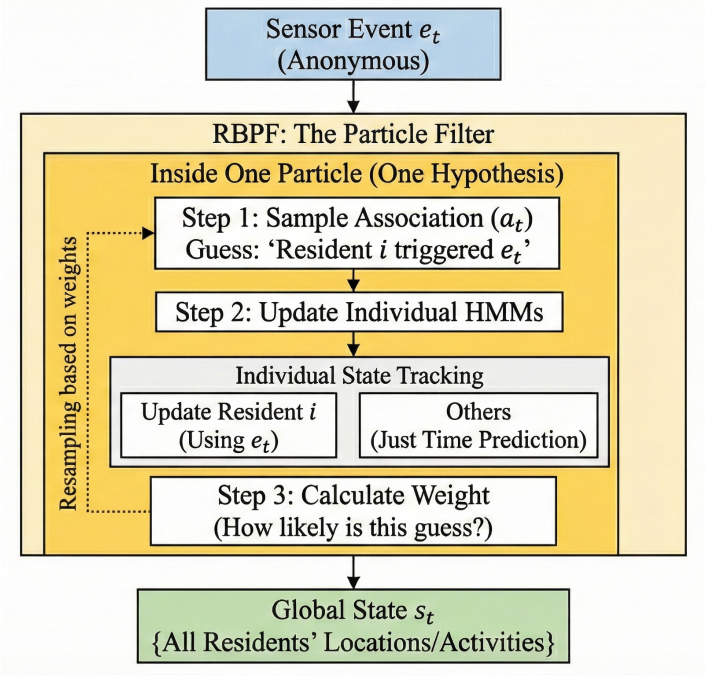



Figure 2.4 – Flowchart of the multi-resident RBPF data association method proposed by Wilson and Atkeson [204].

However, in multi-resident scenarios, directly modeling the joint state of all residents leads to an exponential increase in the state space and introduces a severe *Data Association* challenge: the system does not know which resident triggered which sensor event. To address this, Wilson and Atkeson proposed a Rao-Blackwellised Particle Filter (RBPF) framework, which decomposes the problem into association sampling and per-resident filtering as illustrated in Figure 2.4: (1) the particle filter samples the data association variable $\theta_t$, representing a hypothesis about which resident is responsible for each sensor event; (2) within each particle, an independent single-resident HMM is maintained and updated for each resident, conditioned on the current association hypothesis; (3) particle weights are computed based on how well the hypothesized states explain the observations, allowing the filter to retain high-confidence hypotheses while discarding unlikely ones.

This foundational approach effectively decomposed the multi-subject Data Association task into three conceptual sub-tasks: ***global state representation***, ***individual state transition***, and ***joint state hypothesis filtering***. This formulation enables a modular and scalable approach to reasoning under uncertainty in multi-resident environments. In particular, by modeling state transitions at the level of individual subjects and filtering hypotheses based on observed sensor events, the complex multi-resident problem is reduced to a set of simpler sub-problems. In addition, a key advantage of this decomposition is that individual state transitions can be learned from single-resident data and transferred to multi-resident settings, allowing existing

models—such as those based on language modeling [35]—to be reused and generalized. Many subsequent methods that claim to be "unsupervised", such as those by Riboni et al. [159] and Wang and Cook [197, 198], in fact follow this same paradigm of single-subject modeling combined with observation-driven filtering.

We believe that this paradigm remains a promising foundation for future research. In particular, future progress may come from (i) enriching the expressiveness of the global state representation to better capture inter-resident interactions, (ii) extending individual state transition models beyond the first-order Markov assumption to account for higher-order temporal dependencies, and (iii) advancing filtering strategies to more effectively retain plausible joint state hypotheses consistent with real-world observations.

#### 2.4.2.3 Partial State Methods

Despite Wilson and Atkeson's early emphasis on the importance of system-level state modeling, subsequent works did not substantially advance the completeness of state representation or its role in future inference. For instance, Crandall et al. [65] treated user identity as the hidden state and applied an HMM to infer the subject responsible for each sensor event. Singla et al. [173], on the other hand, modeled only the activity as the hidden state, predicting the next activity directly from the current event without explicitly modeling subject identities. Other efforts, such as those by Hsu et al. [98] and Cook et al. [58], adopted CRFs and HMMs, respectively, and jointly modeled user identity and activity as part of the system state. While this allows for mutual inference between the two variables, it still falls short of the holistic state modeling proposed by Wilson and Atkeson. Specifically, these methods maintain only partial and individual state representations at each time step $t$, typically limited to the inferred identity and activity of the subject associated with the current observation $O_t$. They do not preserve a global view of the environment—such as the total number of active residents or the past locations and activities of each individual—which is essential for reasoning about temporal and spatial continuity in multi-resident settings. As a result, their state representations remain shallow and fragmented, limiting their ability to perform coherent multi-step reasoning. In effect, many of these approaches behave similarly to stateless classification models, performing per-event inference without maintaining a structured understanding of the evolving environment.

#### 2.4.2.4 Latent State Methods

While Wilson and Atkeson [204] originally advocated for a complete *global state* formulation, explicitly maintaining the joint state of all residents, most subsequent works deviated from this principle and relied instead on *partial state* representations, thereby losing the ability to reason over the full environment. More recently, methods such as sMRT [197] and GAMUT [198] can be viewed as a return to Wilson's original vision of global state modeling. As shown in Figure 2.5, they have introduced a

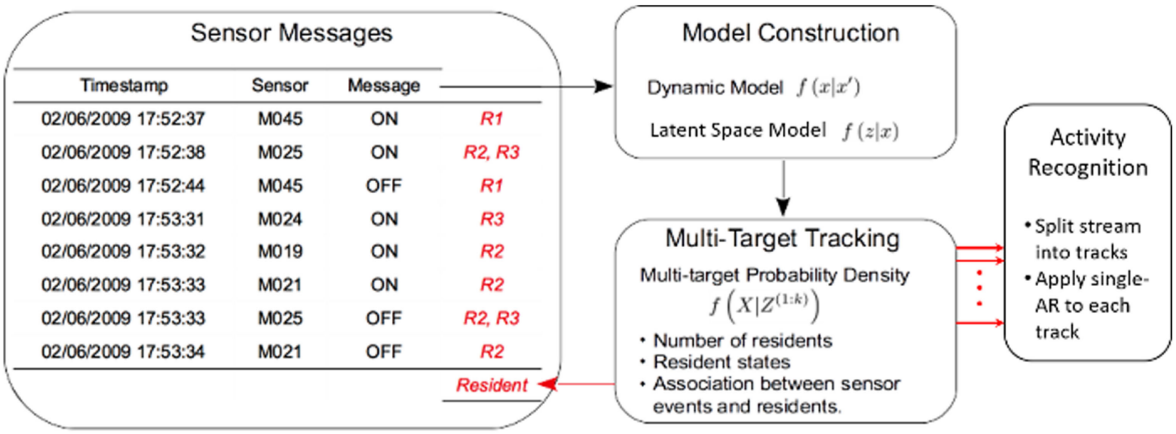



Figure 2.5 – Flowchart of the gaussian multi-resident tracking method (GAMUT) proposed by Wang and Cook. [198].

new paradigm in which the system state is represented not by explicit symbolic variables (e.g., resident location or activity), but by *latent vectors* learned from sensor embeddings. In these methods, each resident $i$ at time $t$ is associated with a continuous latent state vector $z_t^i \in \mathbb{R}^d$, and the global state of the system is defined as the collection of all active residents' latent states:

$$x_t = \{z_t^1, z_t^2, \ldots, z_t^K\}.$$

This representation substantially expands the expressive power of the state space. Rather than explicitly enumerating symbolic state variables (location, activity, identity), the latent vectors encode sensor semantics, spatial correlations, and behavioral regularities in a continuous embedding space. As a result, state transitions can be modeled as smooth dynamics in this latent space, allowing the system to generalize across different sensor deployments and household layouts. The transition model typically assumes a *linear Gaussian dynamic*, where each resident's latent state evolves with constant velocity, while the observation model is defined as a Gaussian likelihood over the latent vectors of the triggered sensors.

For filtering, both sMRT and GAMUT adopt the Gaussian Mixture Probability Hypothesis Density (GM-PHD) filter. This choice is significant: unlike the RBPF of Wilson and Atkeson [204], which relies on discrete sampling of global state hypotheses, the GM-PHD filter directly propagates the intensity function of latent states, enabling simultaneous tracking of an unknown and time-varying number of residents. Through recursive prediction and update steps, the GM-PHD filter maintains a multi-modal distribution over possible resident states, while naturally supporting *birth* and *death* processes to handle residents entering and leaving the environment. Data association is performed by maximizing the likelihood of assigning each observed sensor event to one of the maintained resident hypotheses, effectively decoupling identity management from explicit symbolic tracking.

Compared to Wilson and Atkeson's explicit symbolic state formulation, this latent-state framework achieves two main improvements: (1) the use of continuous hidden vectors allows richer semantic representation of resident states—for example, when a person is transitioning between two sensors, their state can be expressed as a linear combination of the corresponding sensor embeddings; (2) the GM-PHD filter provides a principled Bayesian mechanism to maintain multiple resident trajectories and dynamically estimate the number of active residents, which earlier HMM/RBPF-based approaches lacked.

Nevertheless, several limitations remain. The models still rely on a first-order linear Gaussian assumption, which cannot capture more complex or higher-order transition patterns. Latent embeddings, while expressive, reduce interpretability and make it difficult to incorporate prior knowledge. Finally, maintaining large Gaussian mixtures can become computationally costly as the number of residents and sensors increases.

### 2.4.3 Mutli-Subject Activity Classification

In recent years, some studies have opted to bypass the explicit Data Association step by directly classifying sequences of sensor events that contain activity traces from multiple users. Due to its simplicity and modeling convenience, this strategy has been widely adopted over the past decade. The central idea is to implicitly encode user identity into the final activity label, thereby integrating Data Association and Activity Classification into a single classification task. Two main modeling strategies have emerged under this paradigm. The first incorporates user identity into the classifier itself, by training a separate classifier for each user—a technique often referred to as *multi-task learning*. The second encodes user identity into the label space, by defining each activity label as user-specific, resulting in a *multi-label classification* problem.

#### 2.4.3.1 Multi-Task Learning

The first category of methods considers multi-subject activity recognition as a multi-task learning problem where each task is to recognize a subject's activities, usually with subject-specific classifiers. In the case of neural network models, separate output layers are assigned to each subject [130]. Similarly, for graphical probabilistic models like HMM, each subject is assigned to a specific chain of hidden states [51, 15, 32, 188, 189]. Other works proposed similar approaches based on Formal Concept Analysis [94]. The advantage of multi-task learning is the ability of the model to learn feature representations that are applicable to different tasks from the shared layers. In the context of HAR, this implies modeling the commonalities among different subjects, while the differences between subjects are separately captured by different classifiers. However, this approach presupposes a fixed number of subjects and cannot deal with situations where a subject is involved in parallel activities and unidentified-group activities, as only one singular activity is predicted for each given subject.

#### 2.4.3.2 Multi-label Classification

Another approach is multi-label classification, which considers the recognition of different subjects' activities as a single unified task of inferring any non-zero number of labels at a time. These labels can be subject-specific as well as anonymous. For example, in a scenario where Subject A is having snacks while watching TV, Subject B is reading and an anonymous subject is using the bathroom, a multi-label model can infer four labels which are "A is watching TV", "A is having snacks", "B is reading" and "using the bathroom". In the same scenario, a multi-task classification model would struggle to handle such a complex situation. There are mainly four types of multi-label methods typically used in multi-subject HAR, which we summarise in Table 2.4. Binary relevance [111, 17, 103], consists in predicting each activity label with a binary classifier. A major drawback of this approach is that dependencies between different activity labels are not taken into account during classification. To overcome this drawback, the output of a binary classifier can be inputted as features into the next binary classifier, thus forming a chain of classifiers. This extension of binary relevance is typically named classifier chain [126, 104]. However, the performance of the classifier chain depends on the order of binary classifiers in the chain, and there is no method in general to find the best possible order. Another method treats each combination of labels in the training dataset as a new label, leading to a classical classification problem. Most works in HAR refer to it as label combination [46, 32, 127, 126, 128, 135, 181, 41, 42, 120, 43]. In this approach, any label dependency can be exploited by the model, since all combinations of labels can be predicted. However, it also makes the model unable to infer combinations that do not exist in the training set, reducing the generality of the model. More importantly, the number of combined labels increases exponentially with the number of initial labels. This makes this approach difficult to apply to complex situations with many subjects and activity classes. To avoid the exponentially growing complexity of label combination, the Random $k$-labelsets method [111, 115] randomly partitions the labelset into subsets of size $k$ and then applies the label combination method for each subset to result in an ensemble of classifiers. This effectively reduces complexity but loses some combinations which further reduces the generalization potential of the mode. The performance will be very sensitive to the choice of hyperparameter $k$.

Recently, multi-task and multi-label methods have been combined to take advantage of the strengths of both approaches. For example, the work in [153], proposes a multi-task deep learning architecture with a multi-label classification head for each subject. Hence, for each subject, this approach is capable to detect activities performed in parallel by the same subject.

Implicit data association methods assume that different users have different behavioural habits when performing a given activity, thus generating features that are different enough to distinguish between subjects. However, this assumption does not always hold, especially when there are not enough sensors or if the activity involves minimal interaction. Such approaches will also struggle to scale with the number

Table 2.4 – Multi-label classification methods used in multi-subject HAR.

| Method [Ref.] | Description | Pros | Cons |
|---|---|---|---|
| Binary Relevance [111, 17, 103] | Building a separate binary classifier for each label | Low complexity, linear to the number of subjects | Ignores label correlations |
| Classifier Chain [126, 104] | Chaining each binary classifier, with the upstream classification results added to the input of the downstream classifiers | Considers label correlations, linear to the number of subjects | Performance depends on the order of classifiers |
| Label combination [46, 32, 127, 126] [128, 135, 181, 41] [42, 43] | Creating a new label to classify for each unique combination of labels | Captures all label relationships and combinations of the training dataset | Poor scalability, complexity increases exponentially with the number of subjects |
| Random k-labelsets [111, 115] | Randomly sampling subsets of size $k$ from labelset and applying label combination for each subsets | Reduced complexity, considers label correlations | Loss of combinations, sensibility to sample size |

of subjects in the environment. If many different subjects perform the same activity classes, it will most likely be difficult to learn sufficiently discriminating features from data to distinguish each subject accurately. As a result, models can be prone to overfitting in implicit data association. For example, a model may always assign an activity to the subject who performs it most frequently in the training set.

## 2.5 Datasets

In this section, we summarize the main public datasets that have been proposed for ambient-based multi-subject HAR. These datasets, along with their main characteristics, are summarized in Table 2.5. Some of these datasets were released before a complete formalization of multi-subject HAR was established. Hence, activity labels and subject identification labels may not be provided for each event in these datasets. This may impose limitations on how they can be used for training or evaluating Data Association and Activity Classification tasks. According to our formalization, both Data Association and Activity Classification operate at the level of individual sensor events. Therefore, datasets lacking event-level annotations may not be directly applicable within our framework or may require significant preprocessing or reinterpretation. To clarify the scope of datasets used in our evaluation, we summarize the annotation granularity of commonly used multi-subject HAR datasets in Table 2.6.

In the following, we describe the main multi-subjects datasets.

### 2.5.1 CASAS

CASAS [59] is a comprehensive collection of datasets that includes both single-person and multi-person subsets, totaling over 89 datasets. These datasets were collected from more than 50 distinct environments, with each subset recorded using

Table 2.5 – Multi-subject datasets.

| Dataset | Sensors | # subjects at the same time | # distinct subjects | Considered activity classes |
|---|---|---|---|---|
| CASAS (8-Kyoto) [61] | Motion sensors, Temperature sensors, Door sensors, Item Sensors, Burner sensor, Cold/Hot Water sensors, Electricity usage | 2 | 2 | Bathing, Eating, Enter Home, Housekeeping, Leave Home, Cooking, Personal Hygiene, Sleeping, Wandering, Watch TV, Work |
| CASAS (10-Tulum) [57] | Motion sensors, Temperature sensors, Door sensors | 2 | 2 | Sleeping, Cooking, Taking Medicines, Entering Home, Leaving home |
| CASAS (4-Kyoto) [173] | Motion sensors, Temperature sensors, Door sensors, Item Sensors | 2 | 40 | Sleeping, Cooking, Taking Medicines, Entering Home, Leaving home |
| ARAS [16] | force sensitive resistors, pressure mats, contact sensors, proximity sensors, sonar distance sensors, photocells, temperature sensors, infrared (IR) receivers | 2 | 4 | Idle, Sleeping, Brushing teeth, Watching TV, Toileting, Eating Preparing a meal, Hanging out laundry, Having a guest, Doing cleaning, Having a nap |
| MARBLE [22] | Door sensors, Mat sensors, Plug sensors, Micro-localization, Smartwatch, Phone usage sensor | 1-4 | 12 | Answering Phone, Clearing Table, Cooking, Eating, Entering/Leaving Home, Making Phone Call, Setting Up Table, Taking Medicines, Using PC, Washing Dishes, Watching TV |
| MuRAL [49] | Door sensors, Plug sensors, Motion sensors | 1-4 | 18 | Entering and Leaving, Personal washing, Watching TV, Toileting, Eating Preparing a meal, Entertaining, Having a guest, Doing cleaning, Bedroom Activities |
| SDHAR-HOME [154] | Vibration sensors, Temperature sensor, Humidity sensor, Contact sensors, PIR sensors, Illuminance sensors, Power sensors, Micro-localization, Wristband data | 2 | 2 | Bathroom Activity, Chores, Cook, Dishwashing, Dress, Eat, Laundry, Make Simple Food, Out Home,Pet,Read, Relax, Shower, Sleep, Take Meds, Watch TV, Work,Other, |

between 20 and 86 sensors of various types. Most CASAS subsets focus on single-person scenarios, and only some of them involve multiple persons, and the event-level annotations are often lacking in these multi-person datasets. For example, although *Paris* and *Laval* are multi-subject, they are not annotated. *Tulum*, *Kyoto-8* and *Cario* include two persons, but only provide the start and end times of each resident's

Table 2.6 – Multi-subject datasets: annotations and types of activities.

| Dataset | Data Association Granularity | Activity Classification Granularity |
|---|---|---|
| MARBLE [22] | Event-level | Event-level |
| CASAS (8-Kyoto) [61] | Activity-level | Interval-level |
| CASAS (10-Tulum) [57] | Activity-level | Interval-level |
| CASAS (4-Kyoto) [173] | Event-level | Event-level |
| ARAS [16] | Activity-level | Event-level |
| MuRAL [49] | Event-level | Event-level |

activities, without labeling which events correspond to which resident or activity. Only *Kyoto-4* (also called *ADLMR*) labels all events with the corresponding resident ID and activity, making it highly suitable for Data Association and Activity Classification, and thus more widely used.

One notable feature of CASAS is its use of ceiling-mounted motion sensors to track individual users' local positions. Additionally, door sensors, item contact sensors, and other complementary sensors provide more comprehensive activity information. However, the ceiling-mounted nature of motion sensors makes it difficult to describe their positions in natural language, as they lack distinguishing spatial references. This limitation presents challenges for LLM-based reasoning [84, 48], as spatial semantics play a crucial role in activity recognition and user localization.

### 2.5.2 ARAS

The ARAS dataset [16] is a multi-person dataset collected over one month of real-life activities by two pairs of users in two different real homes. It utilizes a variety of environmental sensors, including photocell, temperature, distance, force, and contact sensors. One of its key advantages is its continuous data collection in real-world environments over an entire month, which is exceptionally rare among publicly available datasets. Moreover, unlike CASAS, ARAS sensors have well-defined contextual information, including their location, function, and type, all of which can be described in natural language. As a result, ARAS was one of the earliest datasets explored for applying large language models to sensor-based activity recognition [84, 48]. However, ARAS provides subject-specific annotations for 27 activity categories at the event level but does not specify which subject triggered each event. This limitation reduces its suitability for evaluating models that track individual users. Furthermore, ARAS only supports two simultaneous subjects, limiting its applicability to more complex multi-person household scenarios.

### 2.5.3 MARBLE

Compared to CASAS and ARAS, MARBLE [22] is a more recently developed dataset explicitly designed to capture richer and more diverse multi-user interactions. It contains sensor data collected in environments with 1 to 4 subjects present simultaneously. Unlike CASAS and ARAS, MARBLE integrates both environmental sensors and wearable devices, including smartwatches, smartphones, and micro-localization systems, allowing for more precise user tracking. Moreover, MARBLE's environmental sensors, similar to those in ARAS, have well-defined contextual information, including location, function, and type, all of which can be described in natural language. MARBLE provides fully event-level annotated environmental sensor data, specifying both the identity of the triggering subject and their corresponding activity, addressing a key limitation in CASAS and ARAS. The diversity of subject numbers, the rich sensor context, and the completeness of MARBLE's annotations make it highly suitable for training and evaluating models in multi-user activity recognition, sensor event separation, and user tracking. Unfortunately, the activities collected in the MARBLE dataset are scripted and do not align with real-world activity timelines. Typically, each activity, such as cooking or watching TV, lasts only a few minutes, introducing a bias between the dataset and real-world activity distributions. The scripted nature and short activity durations (often a few minutes) significantly deviate from real-world behavior patterns, limiting the dataset's generalizability for long-form reasoning tasks.

### 2.5.4 MuRAL

As part of this thesis research contribution [1], MuRAL [49] (**Mu**lti-**R**esident **A**mbient sensor dataset with natural **L**anguage) is a multi-subject smart home dataset specifically designed to support language-driven human activity recognition. The dataset was collected in the *DOMUS* [2] intelligent apartment, a 60 $m^2$ living lab replicating a real residential environment. As illulstrated in Figure 2.6, the environment includes a kitchen, dining area, living room, bathroom, and two bedrooms. A total of 23 ambient sensors were deployed, comprising 6 infrared motion sensors, 10 magnetic contact sensors, and 7 smart plugs, each assigned to a distinct activity zone to maximize interpretability. Bedrooms were excluded from direct sensing for privacy, with only door sensors monitoring entry and exit. All sensor data were integrated via the OpenHAB smart home platform [3].

MuRAL involves 18 participants across 21 sessions, each lasting approximately one hour and comprising 2–4 residents (19 sessions with 3 participants, one with 2, and one with 4). Participants were assigned social roles (e.g., roommates, family members) and contextual settings (e.g., weekday morning, weekend afternoon) but acted freely, producing natural and diverse interaction patterns. To ensure ecological validity,

1. https://mural.imag.fr
2. https://www.liglab.fr/fr/recherche/plateformes/domus (last seen in September 2025)
3. https://www.openhab.org/ (last seen in September 2025)

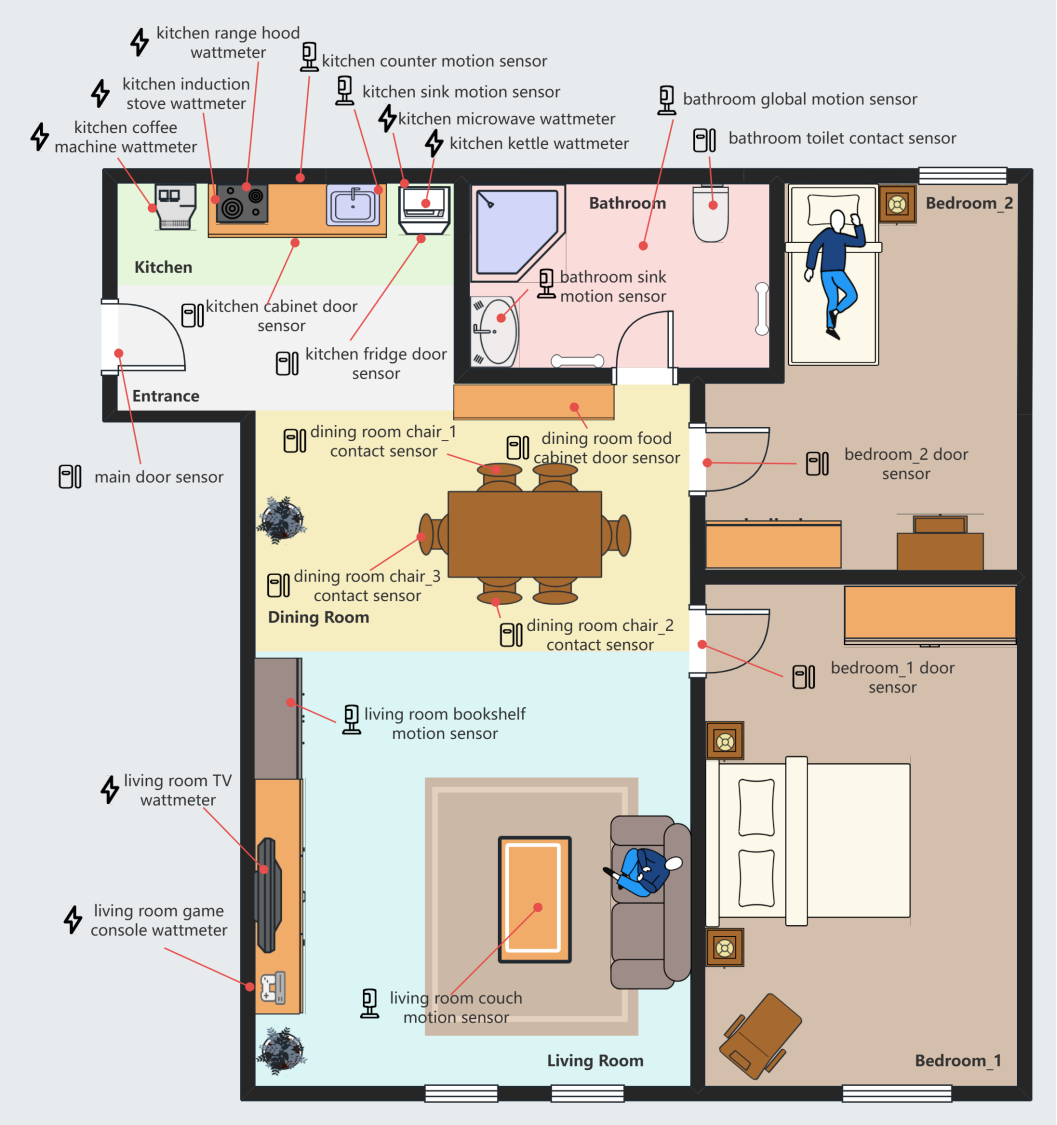



Figure 2.6 – Floor plan of the MuRAL DOMUS intelligent apartment with the names and locations of all ambient sensors [49].

the apartment was stocked with food, books, and entertainment, enabling realistic activities such as cooking, dining, reading, and gaming. Personal care activities (e.g., showering, toilet use) were simulated for ethical reasons.

Unlike prior datasets, MuRAL provides *fine-grained natural language annotations*. Each event is labeled with the triggering resident, an optional textual description, and a higher-level activity category. Residents were anonymized with alphabetical identifiers (A, B, C, . . . ). Natural language descriptions capture detailed contextual information, while activity labels span 27 categories (e.g., *preparing dinner*, *watching TV*, *chatting*, *using PC*).

Table 2.7 summarizes key statistics. On average, each session contains 407 sensor events, 200 natural language annotations (11.7 words per description), and 42 activity instances spanning 16 distinct activity types. Activities last between 2–7 minutes, and sensor deployment averages 19 sensors per session. These characteristics make MuRAL a challenging benchmark for reasoning over ambiguous, overlapping, and socially situated multi-resident activities.

Table 2.7 – Statistical summary of the MuRAL dataset.

| Statistic | Mean | Min | Max |
|---|---|---|---|
| #Residents per session | 3.00 | 2 | 4 |
| #Sensor events | 406.8 | 266 | 558 |
| #Natural language descriptions | 200.3 | 128 | 284 |
| Description length (#words) | 11.7 | 9.8 | 13.2 |
| #Activity instances | 41.7 | 20 | 63 |
| #Activity types | 15.7 | 10 | 19 |
| Activity duration (min) | 3.75 | 1.98 | 7.01 |
| #Sensors used per session | 18.8 | 15 | 22 |

Overall, MuRAL introduces a rich, naturalistic, and fine-grained resource that supports tasks ranging from resident identification and action description to high-level activity recognition in complex multi-user environments.

## 2.6 Conclusion

In this chapter, we analyzed the limitations of existing research on multi-subject HAR. First, we pointed out that prior works lack a unified and rigorous formalization of multi-subject HAR. Many approaches only discuss activity types at the terminology level (e.g., sequential, concurrent, collaborative), which leads to inconsistencies across studies and prevents methods from being generalized across different settings. Second, while methods based on wearable, radio, vision, and audio modalities have shown effectiveness, they often suffer from practical limitations such as intrusiveness, high deployment cost, or privacy concerns. Third, even within the ambient sensing paradigm—which is more privacy-friendly and widely adopted in smart homes—existing methods fall into two groups with critical drawbacks:

— *Data Association-based methods* improve activity classification by separating events by resident, but current models are limited to simplified state representations (e.g., first-order Markov), restricting their ability to capture complex temporal and spatial dependencies.

— *Data Association-free methods* (e.g., multi-task or multi-label classification) integrate subject identity into the classifier or label space, but suffer from poor scalability as the number of residents and activities grows, and cannot achieve fine-grained event-level tracking.

To address these issues, in the earlier sections we introduced a simple and unified formalization of multi-subject HAR, and justified our choice of focusing on *ambient sensing* as the primary modality of interest due to its privacy-friendliness and feasibility in real-world smart home deployment. In light of the aforementioned limitations, the subsequent chapters of this thesis will focus on the following aspects:

1. **Extending Data Association models.** We enhance system state representations and generalize state transition models from first-order to higher-order Markov processes, enabling richer temporal and spatial reasoning in multi-resident environments.
2. **Language modeling of mixed events.** We investigate how event sequence modeling techniques, particularly LLMs, can be adapted from single-resident HAR to disentangle and interpret mixed-event streams in multi-resident settings.
3. **Scaling LLMs for multi-resident HAR.** We explore how LLMs can be scaled to handle multi-resident HAR tasks, aiming to improve system generalization and multi-task processing capabilities for real-world smart home deployment.

# CHAPTER 3

# DATA ASSOCIATION VIA MDP MODELING AND SEARCH-BASED POLICY OPTIMIZATION

## 3.1 Introduction

Human Activity Recognition (HAR) in multi-resident environments has become an important topic in smart home and pervasive computing research. Compared with single-resident settings, sensor readings from multiple concurrent residents often overlap or interfere in both time and space, making it difficult to determine *who triggered which event*. This constitutes a problem known as *Data Association* problem [204, 31, 25, 98, 46, 197]. Numerous empirical studies have demonstrated that accurate data association substantially improves the overall performance of multi-resident activity recognition systems [51, 98, 198, 47, 46]. Therefore, achieving robust data association and trajectory maintenance under real-world conditions—open-world settings with an unknown and time-varying number of residents, and noisy binary sensor observations—remains a fundamental challenge toward deployable multi-resident HAR systems.

As discussed in the literature review of data association (Section 2.4.2), existing methods can be broadly categorized by how they model and utilize **system state**. Early **stateless** approaches treated data association as an event-wise classification

task, assigning each sensor trigger to the most likely resident [63, 64]. Such models ignore temporal and contextual dependencies, often exhibiting a "frequency bias" that attributes frequently used sensors (e.g., a living room couch pressure mat) to the same resident, and thus fail in multi-resident shared spaces. Later **partial-state** methods introduced limited temporal dependencies through a small set of hidden variables—such as resident identity [65], activity [173], or their joint combination [98, 58]. Although this enables short-term consistency of data attribution, these models still lack a persistent global view of the environment: they do not maintain knowledge of how many residents are active, where each is located, or how their activities evolve. Consequently, events are interpreted in isolation rather than as part of a coherent, continuously updated system state, limiting the model's ability to reason about spatial continuity, temporal consistency, and inter-resident interactions that are crucial in multi-resident scenarios. A major contribution by Wilson and Atkeson [204] proposed to explicitly maintain a **global system state** that represents the joint configuration of all residents, each described by a location–activity pair. They modeled each individual's state transition as a first-order Hidden Markov Model (HMM) and used a Rao-Blackwellised Particle Filter (RBPF) to perform hypothesis sampling and filtering in the joint state space, thereby coupling data association and state evolution in a unified inference loop. This state-centric method became a conceptual foundation for subsequent work.

More recently, methods such as sMRT and GAMUT [197, 198] revisited global state modeling by replacing explicit symbolic variables with **continuous latent vectors** learned from sensor embeddings and employing Gaussian Mixture Probability Hypothesis Density (GM-PHD) filters to track an unknown, time-varying number of residents. While this latent-state formulation enhances representational flexibility, it also introduces fundamental limitations. The GM-PHD filter maintains only the first-order moment (intensity function) of the multi-target posterior rather than explicit identity trajectories, effectively assuming independence among residents. As a result, identity continuity is not explicitly enforced: when observations are sparse, noisy, or semantically ambiguous, the filter tends to instantiate new density components instead of updating existing ones, leading to fragmented and short-lived trajectories. Moreover, the skip-gram–based embedding mechanisms used in sMRT and GAMUT capture only local co-occurrence relationships between sensors and activities, without modeling long-range temporal dependencies, inter-resident interactions, or high-level behavioral regularities.

These structural limitations motivate the need for a more principled framework that combines compact, recursively updatable, higher-order Markov state representations with data-driven yet structured modeling of behavioral dynamics. That is, a formulation that unifies state evolution, decision-making, and consistency enforcement within a single optimizable process.

To this end, we propose a Markov Decision Process (MDP) formulation of the data association problem, where each state encodes the complete assignment history up

to the current event and integrates predictive models of individual behavior. In this MDP framework, each action corresponds to assigning the current event to a specific resident or creating a new resident identity. We formally prove that solving the data association problem is equivalent to finding an optimal policy in this formulation. To optimize the policy, we adopt a heuristic tree search strategy guided by a likelihood-based reward function. The reward model is composed of two components: (i) a single-resident event sequence model that estimates the likelihood of observing an event given a resident's past history, and (ii) a new resident creation model that evaluates the plausibility of initiating a new sequence for a new resident. These components enable the search algorithm to evaluate assignment consistency and dynamically balance reuse versus creation of resident trajectories. Together, this framework provides a scalable and theoretically grounded solution for multi-resident data association.

We evaluate our method on 3 publicly available datasets: CASAS [173], MARBLE [22], and MuRAL [49], which cover diverse multi-resident scenarios and sensing configurations. Using complementary evaluation metrics, our approach consistently outperforms state-of-the-art baselines across all datasets, demonstrating strong performance and generalization capabilities.

Our contributions are the following:

— We propose an original mathematical formulation of the multi-resident Data Association problem as an MDP, and formally prove that the optimal policy of this MDP is equivalent to solving the original data association task.

— We propose a practical framework for policy optimization tailored to data association: constructing a reward model with single-resident Next-Event Prediction (NEP) and new resident creation and optimizing the assignment policy via heuristic search algorithms.

— We conduct extensive experiments on three representative datasets. Using three complementary metrics, we demonstrate that our method consistently outperforms state-of-the-art baselines.

## 3.2 Method Overview

Figure 3.1 provides an overview of our framework. Given a stream of mixed sensor events of multiple residents (left), we formulate multi-resident data association as an MDP (Section 3.3). At each time step, the current state encodes the historical assignments of all residents and the new event that needs to be assigned. The policy optimization process is then performed via heuristic tree search (Section 3.4): for each candidate action (assigning the event to an existing resident or creating a new one), the reward model (Section 3.5) estimates the likelihood of this assignment based on prior context. The search algorithm explores these candidate actions and selects the

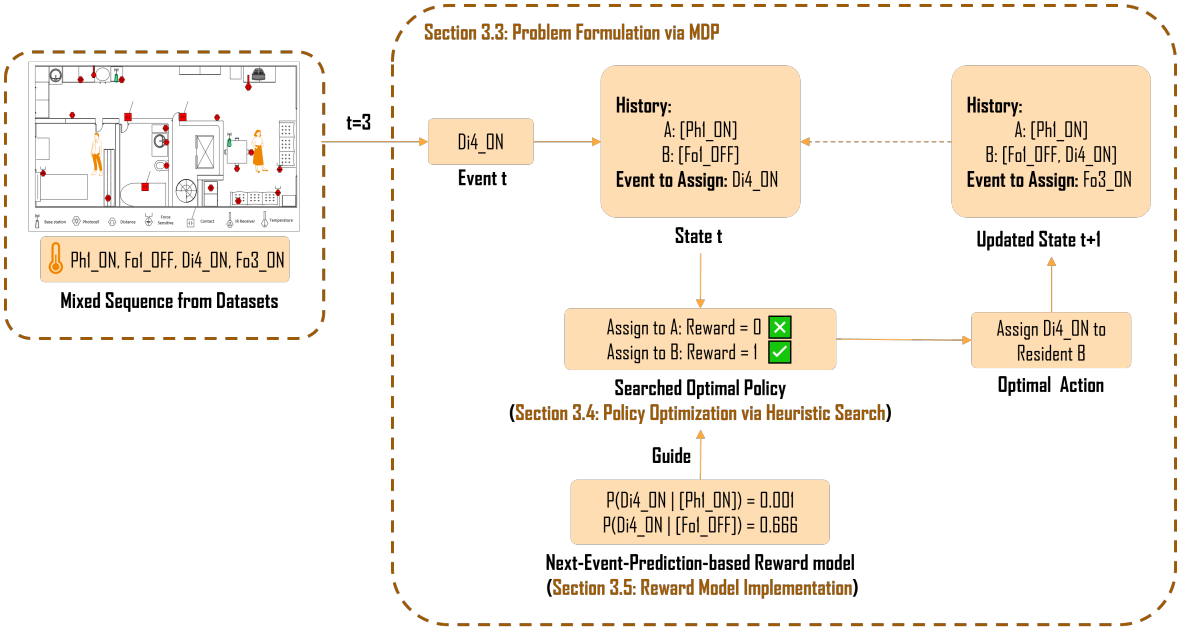



Figure 3.1 – Overview of multi-resident data association modeled as a Markov Decision Process. Taking the assignment of the third event in an illustrative multi-resident sequence as an example.

path that maximizes the cumulative reward, resulting in a complete assignment of the mixed event stream into separate sequences for each resident.

## 3.3 Problem Formulation via MDP

In this section, we present how the Data Association problem can be formally modeled as an MDP. In Section 3.3.1, we provide the basic definition of an MDP. Then, in Section 3.3.2, we formally describe how to model Data Association as an MDP, laying the foundation for solving the problem through MDP-based approaches in the subsequent sections.

### 3.3.1 Definition of MDP

An MDP is a mathematical framework that models the interaction between an agent and its environment in sequential decision-making settings. An agent is formally defined as an entity that makes decisions and takes actions within an environment, aiming to maximize long-term cumulative reward.

A typical MDP can be defined by a tuple $(S, A, P, R)$, where

— $S$ is the state space, representing all possible states of the agent,

— $A$ is the action space, representing all actions that the agent is allowed to execute,

— $P(s'|s, a)$ is the state transition model, the probability of reaching state $s'$ from state $s$ by taking action $a$,
— $R(s, a, s')$ is the reward model, the immediate reward for transitioning from $s$ to $s'$ via action $a$.

The agent's objective is to optimize a decision policy $\pi : S \to A$, which maps states to actions, such that the expected cumulative reward over a sequence of actions is maximized:

$$\pi^* = \arg\max_{\pi} \mathbb{E}\left[\sum_{t=0}^{T} R(s_t, \pi(s_t), s_{t+1})\right], \tag{3.1}$$

where $t$ is the index of discrete time step and $T$ is the maximum of time steps. In this research, we only consider a finite number of decision steps.

### 3.3.2 Adapting MDP to Data Association

Under the MDP framework, the Data Association problem is modeled by the following components:

**State** At any event step $t$, the state $s_t$ encodes the **assignment history** prior to $t$, denoted as $h_{<t}$, along with the **current event** $e_t$ to be assigned. Formally, we define the state as:

$$s_t = (h_{<t}, e_t). \tag{3.2}$$

Here, $h_{<t}$ serves as a dynamic record, implemented as a mapping from each resident to their assigned event sequence (cf. state example in Figure 3.1). This state representation captures both past context and the current decision point, and thus satisfies the Markov property.

**Action** As shown in the example of Figure 3.1, at each step $t$, the agent takes an action $a_t$, assigning the current event $e_t$ to a subset of residents:

$$a_t : e_t \mapsto \widehat{\mathcal{U}}_t \subseteq \mathbb{U}_t, \tag{3.3}$$

where $\mathbb{U}_t = \{u_1, \ldots, u_n, u_{\text{new}}\}$ includes all known residents and a placeholder $u_{\text{new}}$ for dynamically introducing new ones.

This formulation supports both single- and multi-resident assignments, enabling the model to handle overlapping activities and adapt to scenarios with an increasing number of residents.

**Transition Model** Given a state $s_t = (h_{<t}, e_t)$ and an action $a_t$, the environment deterministically transitions to the next state by updating the assignment history and moving to the next event. Specifically, the current event $e_t$ is appended to the event

list of each assigned resident in $\widehat{\mathcal{U}}_t = a_t(e_t)$, yielding a new history $h_{<t+1}$ (cf. the updated state in Figure 3.1). If $\widehat{\mathcal{U}}_t$ includes a resident not previously seen in $h_{<t}$, a new resident is created accordingly and labeled using the next available alphabetical ID. The next unprocessed event $e_{t+1}$ becomes the new focus of assignment. Formally:

$$s_{t+1} = (h_{<t+1}, e_{t+1}). \tag{3.4}$$

This transition is deterministic and satisfies the Markov property: all necessary information for future decisions is encapsulated in the updated state.

**Reward Model** The reward model evaluates how likely a proposed resident assignment is to have caused the current event, guiding the agent toward consistent and accurate decisions. At each step $t$, after assigning event $e_t$ to resident set $\widehat{\mathcal{U}}_t = a_t(e_t)$, the agent receives a scalar reward:

$$R(s_t, a_t) = \log P(e_t \mid \widehat{\mathcal{U}}_t, h_{<t}), \tag{3.5}$$

where $s_t = (h_{<t}, e_t)$ is the current state, and $h_{<t}$ denotes the assignment history. This likelihood-based score reflects how well the selected residents explain the observed event in context. Taking the logarithm improves numerical stability and enables additive decomposition across residents, which facilitates optimization.

Evaluating the exact joint likelihood for every possible subset is computationally intractable. Therefore, we approximate the set-level reward by decomposing it into individual relevance scores. Specifically, we treat the contribution of each resident to the event as an additive factor. This allows us to factorize the log-likelihood of the set $\widehat{\mathcal{U}}_t$ into the sum of log-likelihoods of its members, yielding a computation-able reward formulation:

$$R(s_t, a_t) = \sum_{u_k \in \widehat{\mathcal{U}}_t} \log P(e_t \mid u_k, h_{<t}) + \sum_{u_k \in \mathbb{U}_t \setminus \widehat{\mathcal{U}}_t} \log(1 - P(e_t \mid u_k, h_{<t})). \tag{3.6}$$

This formulation supports multi-resident assignments, rewards consistency with historical context, and penalizes implausible associations. By maximizing the total reward over the sequence, the agent searches for the assignment trajectory that best explains the event stream under a probabilistic, likelihood-driven framework.

**Policy** In our MDP formulation, a *policy* defines a strategy for assigning each event $e_t$ to a resident set $\widehat{\mathcal{U}}_t$, based on the current state $s_t$. That is, $\pi(s_t) = a_t$, where $a_t$ is an assignment action.

Executing a policy $\pi$ over a sequence of sensor events $[e_t]_{1 \le t \le T}$ induces a complete assignment mapping $M_\pi$, such that:

$$\forall t \in \{1, \ldots, T\}, \quad \pi(s_t) = \widehat{\mathcal{U}}_t = M_\pi(e_t). \tag{3.7}$$

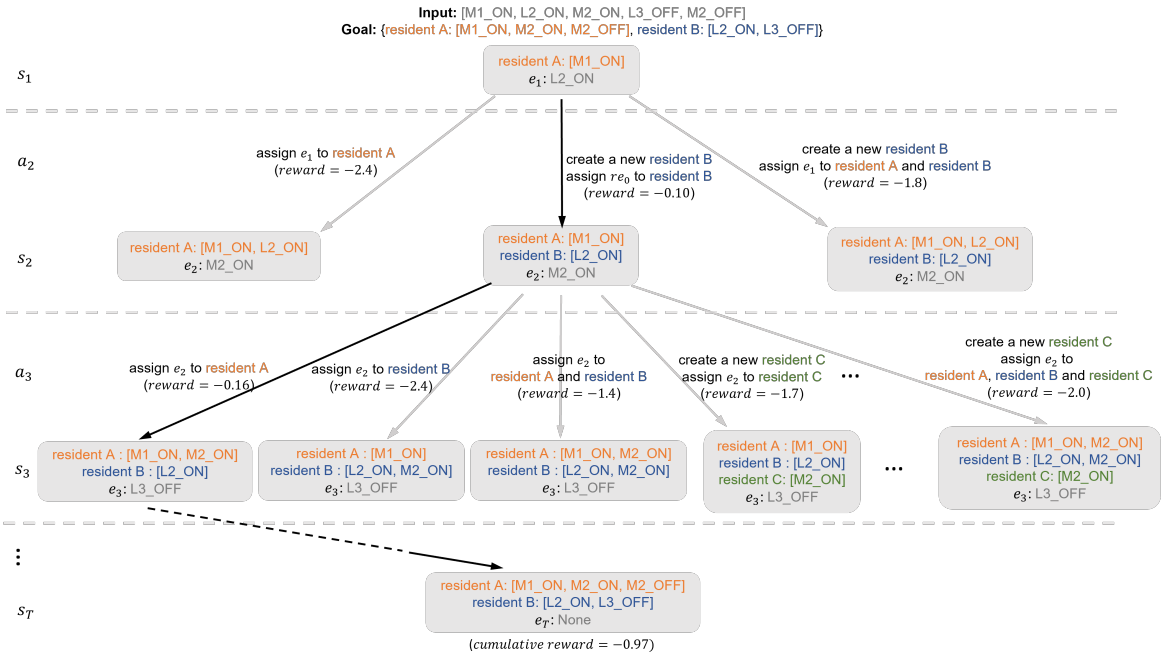



Figure 3.2 – An illustrative example of tree-search-based policy optimization.

The goal of Data Association is to recover the true mapping $M$, which corresponds to a ground-truth policy $\pi^*$ that assigns the correct resident set at every step:

$$\forall t \in \{1, \ldots, T\}, \quad M(e_t) = \mathcal{U}_t = \pi^*(s_t). \tag{3.8}$$

Therefore, the Data Association problem can be formally modeled as a policy optimization task, where the objective is to find an optimal policy $\pi^{\text{opt}}$ that approximates $\pi^*$ by maximizing the cumulative reward:

$$\pi^{\text{opt}} = \arg\max_{\pi} \sum_{t=1}^{T} R(s_t, \pi(s_t)). \tag{3.9}$$

This reward-guided optimization provides an indirect path toward recovering $\pi^*$, by favoring assignment sequences that best explain the observed event stream.

## 3.4 Policy Optimization via Heuristic Search

Finding an optimal policy requires selecting, at each step, the best resident set from a combinatorial action space. To structure this process, we model policy optimization as a tree search problem.

### 3.4.1 Policy Optimization as Tree Search

The policy optimization in our MDP formulation can be naturally viewed as a tree search process (Figure 3.2). Each node in the search tree corresponds to an MDP state $s_t$, which encodes the history of assignments $h_{<t}$ and the current event $e_t$ to be assigned. From a node, each outgoing edge represents an action $a_t$: assigning $e_t$ to one of the existing residents or creating a new resident, leading to a new state where the assignment history is updated and the next event $e_{t+1}$ is ready to be processed. The reward associated with an edge (shown in parentheses in the figure) is computed by the reward model and provides feedback on the quality of that action.

A complete path from the root (where the very first event is assigned to Resident A by default) to a leaf (all events processed) represents one possible full assignment of the entire sequence. The cumulative reward of a path is obtained by summing the stepwise rewards along it, and the search algorithm aims to find the path with the highest cumulative reward. In Figure 3.2, black arrows indicate the actions selected by the search algorithm, whereas gray arrows denote alternative actions that were explored but not chosen.

### 3.4.2 Heuristic Search Algorithms

The search space of policy grows exponentially with the number of events and potential residents, making exhaustive tree search intractable. To address this challenge, we employ heuristic search algorithms that approximate optimal policies by selectively exploring the most promising parts of the search space.

These algorithms evaluate partial assignment sequences using the cumulative reward defined by the reward model, and expand only the top-ranked candidates according to predefined heuristics. This allows the agent to efficiently navigate the combinatorial space without committing too early to suboptimal decisions.

In this research, two main-stream heuristic search [143] are discussed:

**Greedy Search** Greedy search represents the simplest form of heuristic search algorithms. At each step $t$, the algorithm selects the action $a_t$ that yields the highest immediate reward $R(s_t, a_t)$, without considering future consequences. Formally, the greedy action is computed as:

$$a_t^{\text{greedy}} = \arg\max_{a \in A_t} R(s_t, a), \tag{3.10}$$

where $A_t$ is the action space at step $t$.

In the context of our MDP for Data Association, this corresponds to assigning the current event $e_t$ to the resident set $\widehat{\mathcal{U}}_t$ that appears most likely to have triggered it, based on the current history $h_{<t}$. While this approach is computationally efficient

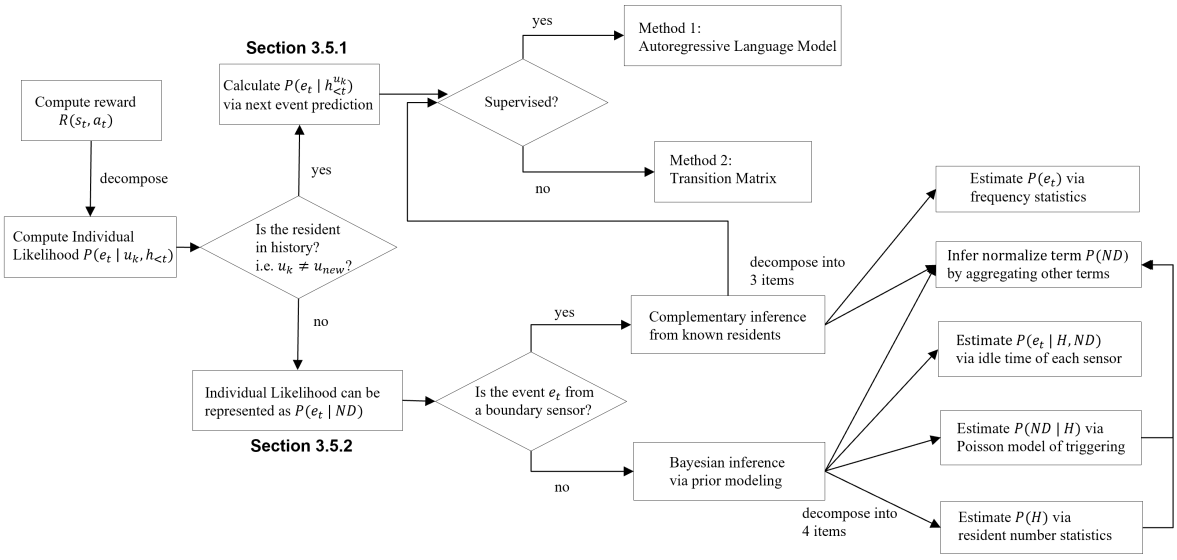



Figure 3.3 – Computational Workflow for the Reward Model Implementation.

and performs reasonably well in scenarios where local decisions dominate global behavior, it may suffer from error accumulation. Incorrect early assignments can bias the resident histories, leading to compounding mistakes in later steps.

**Beam Search** Beam Search [143] improves upon greedy search by maintaining multiple candidate trajectories at each step. It keeps a beam of the top-$b$ partial assignment paths based on their cumulative rewards, allowing exploration of multiple high-probability sequences in parallel. At each time step, for every partial path in the beam, all possible actions are considered, and the resulting new paths are scored and pruned to retain only the top $b$ candidates.

Formally, at step $t$, given a beam $\mathcal{B}_{t-1}$ of partial trajectories, Beam Search generates:

$$\mathcal{B}_t = \text{Top-}b\,\{(s_t, R_{\leq t})\}\,, \tag{3.11}$$

where $R_{\leq t}$ is the cumulative reward up to step $t$. The search continues until all events are assigned.

In our Data Association MDP, Beam Search allows the model to delay hard commitments and consider alternative resident assignment paths that may lead to better global consistency. It is particularly effective in capturing non-local dependencies and correcting early-stage assignment ambiguities.

## 3.5 Reward Model Implementation

To compute the reward (c.f. Equation 3.6) defined in the MDP formalization, the fundamental component is the **individual likelihood** term $P(e_t \mid u_k, h_{<t})$. It

estimates the probability that a resident $u_k \in \mathbb{U}_t$ triggers the current event $e_t$ given the assignment history $h_{<t}$. This section details how we compute this term to build up the reward model. We present an overview of the workflow to compute this term in Figure 3.3.

Recall that $\mathbb{U}_t = \{u_1, u_2, \ldots, u_n, u_{\text{new}}\}$, denoting the set of all residents available at time $t$, where $u_{\text{new}}$ represents a newly introduced resident. We distinguish between two cases:

— **Existing residents** ($u_k \in \mathbb{U}_t \setminus \{u_{\text{new}}\}$): For residents with prior assignments in $h_{<t}$, we compute $P(e_t \mid u_k, h_{<t})$ using an NEP model based on the resident's historical sequence. Section 3.5.1 presents two implementations: an autoregressive language model and a statistics-based transition model.
— **New resident** ($u_{\text{new}}$): For unobserved residents with no history, we define $P(e_t \mid u_{\text{new}}, h_{<t})$ using a prior distribution that reflects the likelihood of a previously unseen resident triggering $e_t$. This prior can be uniform, data-driven, or derived from population-level statistics. Details are given in Section 3.5.2.

Once the individual likelihoods are computed, they can be substituted into Equation 3.6 to evaluate the reward of a candidate assignment.

### 3.5.1 Next Event Prediction for an Existing Resident

Directly modeling the probability $P(e_t \mid u_k, h_{<t})$ requires reasoning over every existing resident's behavior in the context of the full assignment history. To make this estimation tractable, we approximate this conditional probability by modeling the behavioral trajectory of each resident independently, based on their own previously assigned events. Formally, we assume:

$$P(e_t \mid u_k, h_{<t}) = P(e_t \mid h_{<t}^{u_k}), \tag{3.12}$$

where $h_{<t}^{u_k} = [e_1^{(k)}, e_2^{(k)}, \ldots, e_m^{(k)}]$ and $m < t$. For all $1 \leq j \leq m$, $e_j^{(k)}$ is an event previously assigned to resident $u_k$ before step $t$. This approximation assumes that the resident's future behavior depends primarily on their own event history, and that the influence of other residents is negligible given the current context. In Chapter 4, an improved approach will jointly consider all observed residents to avoid this assumptions.

In the following paragraphs, we provide both a supervised method and an unsupervised method to estimate this approximated individual likelihood $P(e_t \mid h_{<t}^{u_k})$.

**Method 1: Supervised Autoregressive Language Model** To predict the likelihood $P(e_t \mid h_{<t}^{u_k})$ for a given resident $u_k$, we use a supervised autoregressive modeling approach inspired by language modeling [147, 148]. The core idea is to model the

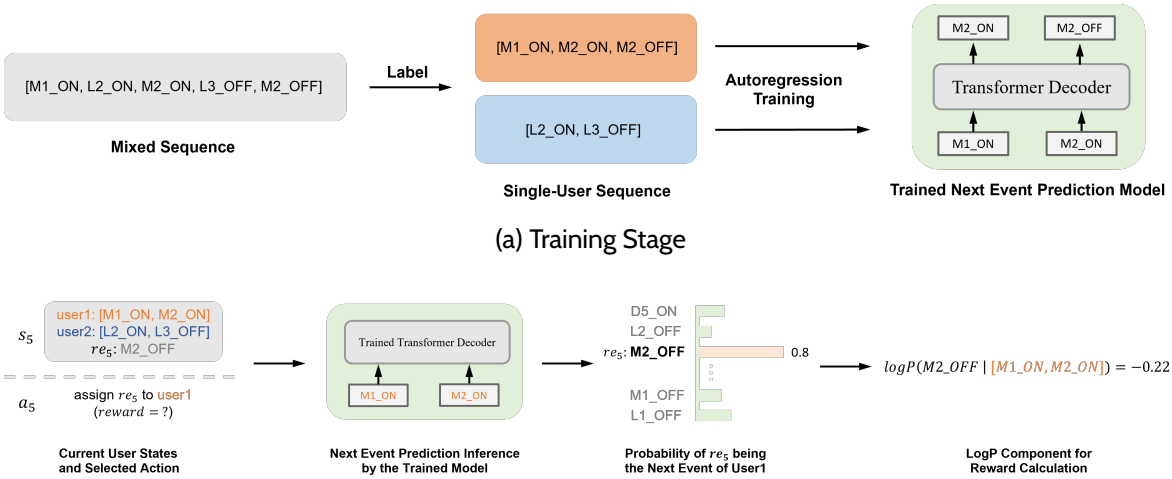



Figure 3.4 – Next event prediction with a supervised autoregressive language model.

resident's behavior as a sequential process and train a neural network model to predict the next event based on the previous ones. A similar approach has also been explored in the work by Takeda et al. [179]. Figure 3.4 illustrates the training and inference stages of this method.

We implement this as follows:

1. **Data preparation:** From the ground-truth annotations, we extract per-resident event sequences $h^{u_k} = [e_1^{(k)}, \ldots, e_L^{(k)}]$. Each event $e_i^{(k)}$ is treated as a discrete token and one-hot encoded for input to the model.
2. **Model training:** Due to its superior ability to model long-term dependencies and its success in autoregressive prediction tasks [147, 148], we adopt a Transformer decoder to model the conditional probability of the next event. The training objective maximizes the likelihood of observing the correct next event in each resident's trajectory:

$$\max_{\theta} \sum_{k=1}^{K} \sum_{i=1}^{L-1} \log P_\theta(e_{i+1}^{(k)} \mid e_1^{(k)}, \ldots, e_i^{(k)}) \tag{3.13}$$

3. **Inference:** Given a historical sequence $h_{<t}^{u_k}$ of a resident $u_k$, the model outputs the probability distribution over the next possible events. The individual likelihood $P(e_t \mid h_{<t}^{u_k})$ is obtained by reading off the probability of the actual observed event $e_t$ from the output distribution.

**Method 2: Unsupervised Statistics-based Transition Matrix** Obtaining fully labeled resident-event sequences is costly or infeasible in real deployments. However,

we observe that short-range transitions in mixed-resident sequences are primarily driven by individual behavior, and thus remain coherent. Similar assumptions have been adopted in prior unsupervised association methods [197, 198], where global transition statistics approximate resident-level dynamics. Based on this, we use a first-order Markov model to construct a transition matrix that supports local assignment decisions in our framework.

Our unsupervised estimation of $P(e_t \mid h_{u_k}^{<t})$ proceeds in two steps:

1. **Transition Matrix Construction.** From the mixed-resident sequences of the entire dataset, we count the number of times each event $e_j$ follows $e_i$, yielding raw transition counts $\text{Count}(e_i \rightarrow e_j)$. We normalize the counts to obtain a transition matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$, where each entry reflects the empirical first-order transition probability:

$$A_{i,j} = \frac{\text{Count}(e_i \rightarrow e_j)}{\sum_{k=1}^{N} \text{Count}(e_i \rightarrow e_k)} \tag{3.14}$$

2. **Estimate Next-Event Likelihood.** Given the most recent event $e_m^{(k)}$ assigned to resident $u_k$, we estimate the likelihood of the current event $e_t$ as:

$$P(e_t \mid h_{u_k}^{<t}) \approx A_{e_m^{(k)}, e_t} \tag{3.15}$$

### 3.5.2 Prior Modeling for New Residents

In real-world smart homes, the number of residents is neither fixed nor known in advance. A practical data-association system must decide online whether a new event $e_t$ should be assigned to a previously observed resident or to a new one. In our MDP formulation (Section 3.3.2), this is achieved by extending the action space to include a new resident placeholder $u_{\text{new}}$, with the associated individual likelihood $P(e_t \mid u_{\text{new}}, h_{<t})$. This section describes how we estimate this term.

We denote by $ND$ (*Never Detected*) the hypothesis in which "the resident has never been detected by any sensor until step $t$". This can be further categorized into two mutually exclusive and exhaustive hypotheses:

— $H$ (Home): the resident has been at home since the system started, but has not triggered any sensor since before time $t$. This case typically occurs during the initial phase of the system (i.e., when time $t$ is small);

— $NH$ (Not Home): the resident is not (and has never been) in the environment until step $t$.

Given the current event $e_t$, categorizing the $ND$ hypothesis into cases $H$ and $NH$ offers a simplified modeling approach for estimating $P(e_t \mid ND)$. A key observation is that if $e_t$ corresponds to a sensor that could not possibly be triggered under the

Table 3.1 – Interpretation and estimation of each term in Equation 3.16 for non-boundary sensor triggers.

| Term | Meaning and Intuition | Estimation Method |
|---|---|---|
| $P(e_t \mid H)$ | Likelihood that a previously undetected resident, silently present in the environment, triggers event $e_t$. Certain events (e.g., getting up) are more likely after long inactivity. | Approximated using an *idle time heuristic*: time since last known event of the same resident type, normalized over all event types. |
| $P(ND \mid H)$ | Probability that a present resident has triggered no sensor events so far. Decreases over time. | Modeled as a Poisson process: $P(ND \mid H) = e^{-\lambda \mathcal{T}}$, with $\lambda$ as event rate and $\mathcal{T}$ as elapsed time. |
| $P(H)$ | Prior probability that there are more residents than currently detected. | Estimated as: $P(H) = \frac{P(N>K)}{P(N \geq K)}$, based on total resident count distribution. |
| $P(ND)$ | Overall probability that a resident has not been detected up to time $t$, under both $H$ and $NH$. | Total probability: $P(ND) = P(ND \mid H) \cdot P(H) + (1 - P(H))$. Used for normalization. |

$NH$ condition, then the $ND$ hypothesis effectively reduces to $H$. Intuitively, when a resident is not at home, the only sensors they could possibly trigger are those located at the physical boundary of the environment, such as a front door sensor or an entryway motion detector. Based on this insight, we define such sensors as *boundary sensors*, and assume that they can be identified based on user-provided configuration.

Accordingly, we distinguish between two categories of events in our computation: those triggered by **boundary sensors** and those triggered by **non-boundary sensors**.

#### 3.5.2.1 Case 1: Non-Boundary Sensor Triggers

When $e_t$ originates from a non-boundary sensor, the $NH$ hypothesis becomes invalid, meaning that the likelihood $P(e_t \mid NH) = 0$. By applying the law of total probability and Bayes' theorem, we obtain a simpler form:

$$P(e_t \mid ND) = P(e_t \mid H) \frac{P(ND \mid H) P(H)}{P(ND)}. \tag{3.16}$$

To compute this, we need to estimate each of its components: $P(e_t \mid H)$, $P(ND \mid H)$, $P(H)$, and $P(ND)$. In the following, we describe the modeling assumptions and estimation strategies for each of these terms. We provide a summary for each term in Table 3.1.

— $P(e_t \mid H)$: To approximate the likelihood that a silently present resident triggers the current event $e_t$, we introduce an **idle time heuristic**.

The key idea is that a longer period of inactivity increases the plausibility that a resident has remained in the environment without triggering any sensors. More

specifically, certain event types tend to occur after long idle periods, and the first observable action after a resident has been inactive (e.g., sitting, sleeping, or staying still) is often marked by a characteristic sensor event. For instance, a resident might sit quietly on a chair: the pressure-on event occurs at the beginning, while the pressure-off event—after a long idle period—marks their next observable activity. If the resident was already seated before the system started, the delayed off-event may appear as their first detectable action.

Therefore, if an event type $e$ is frequently observed in the training data to follow a long silence, it is likely to indicate the reappearance of a previously undetected but present resident. Guided by this intuition, we approximate $P(e_t \mid H)$ by measuring the time interval between the current event and the most recent event previously triggered by the same resident (from training data). This interval reflects the resident's inactivity duration before $e_t$. The resulting value is normalized across all event types to produce a valid probability distribution over $P(e \mid H)$.

— $P(ND \mid H)$: This term represents the probability that a resident has been present in the home but has not triggered any sensor events since the system started. Intuitively, sensor activations from a resident can be viewed as random and memoryless events over time—especially in ambient sensing environments where low-frequency activities are common. Based on this assumption, we model resident-triggered sensor events as a Poisson process, which captures the expected number of events within a given time window.

Under this model, the probability that no event has been triggered within time interval $\mathcal{T}$ follows an exponential decay:

$$P(ND \mid H) = e^{-\lambda \mathcal{T}}, \tag{3.17}$$

where $\lambda$ denotes the average event rate per resident, and $\mathcal{T}$ is the time elapsed since system initialization. This reflects that the longer a resident remains silent, the less likely they are still present but undetected.

— $P(H)$: This term represents the prior probability that at least one additional, previously undetected resident is present in the environment. We estimate it based on a learned distribution $P(N)$, which models the prior belief over the total number of residents in the environment.

Given that $K$ residents have already been detected up to time $t$, we condition on this knowledge to compute the probability that more residents remain undetected:

$$P(H) = \frac{P(N > K)}{P(N \geq K)}. \tag{3.18}$$

This conditional formulation ensures that the estimate adapts to the number of currently known residents and reflects the residual uncertainty about unobserved ones.

— $P(ND)$: Using the law of total probability with $P(ND \mid NH) = 1$, we get:

$$P(ND) = P(ND \mid H)P(H) + (1 - P(H)). \tag{3.19}$$

This serves as the normalization constant in Equation 3.16.

#### 3.5.2.2 Case 2: Boundary Sensor Triggers

When $e_t$ originates from a boundary sensor, we cannot distinguish whether the triggering resident corresponds to the $NH$ (Not Home) or $H$ (Home) hypotheses, as both cases can plausibly result in a boundary sensor activation—e.g., a resident not home entering from outside or a resident at home leaving through the door.

Intuitively, boundary sensors—such as door contacts or entryway motion detectors—are precisely the types of sensors that blur the line between internal and external presence. A boundary event does not reveal whether the person was previously inside or just arriving from outside. Unlike interior sensors, which only make sense for someone already in the home, boundary sensors can be triggered bi-directionally. This semantic ambiguity makes direct probabilistic modeling of $P(e_t \mid NH)$ infeasible, as we cannot observe or characterize the behavior of truly absent residents (under $NH$).

To address this, we sidestep the modeling of $P(e_t \mid NH)$ entirely by reformulating the likelihood of a never-detected resident triggering the event, $P(e_t \mid ND)$, using a complement-based approach. The core intuition is that if none of the currently known residents are likely to have triggered the event $e_t$, then it becomes more plausible that a new resident is responsible.

Following Bayes' theorem, we express the new-resident likelihood as:

$$P(e_t \mid ND) = \frac{P(e_t)P(ND \mid e_t)}{P(ND)} \tag{3.20}$$

$$= \frac{P(e_t)}{P(ND)}\big(1 - P(D \mid e_t)\big), \tag{3.21}$$

where $D$ (*Detected*) denotes the complement of $ND$, i.e., the resident who triggered $e_t$ has already been observed before time $t$.

This formulation captures the idea that the probability of a new resident depends not on modeling absent behavior, but on how unlikely it is that existing residents caused the event. If all known residents have low likelihoods under their personal behavior models, the residual probability is attributed to a potential new arrival.

We estimate $P(D \mid e_t)$ by summing over all $K$ known residents:

$$P(D \mid e_t) = \sum_{k=1}^{K} P(u_k \mid e_t, h_{<t}) \tag{3.22}$$

$$= \frac{1}{P(e_t)} \sum_{k=1}^{K} P(e_t \mid u_k, h_{<t}) P(u_k), \tag{3.23}$$

where we assume a uniform prior over the $K + 1$ candidate residents, giving $P(u_k) = \frac{1}{K+1}$. The likelihood terms $P(e_t \mid u_k, h_{<t})$ are computed using the methods introduced in Section 3.5.1, and the normalization term $P(ND)$ is shared with the non-boundary case and has already been computed in Equation 3.16. The prior $P(e_t)$ is estimated from empirical event frequencies in the training set.

## 3.6 Experiments

We performed comprehensive experiments to verify the effectiveness of the proposed MDP framework. Sections 3.6.1 and 3.6.2 detail the evaluation metrics for the Data Association task and the experimental setup. Subsequently, Sections 3.6.3 and 3.6.4 provide qualitative and quantitative comparisons with state-of-the-art methods, followed by extensive ablation studies in Sections 3.6.5–3.6.8.

### 3.6.1 Evaluation Metrics

#### 3.6.1.1 Resident Assignment Accuracy (Acc)

Prior evaluations of data association often suffer from global bias and cascading error effects. Some studies [159, 48] assess performance only indirectly via downstream activity recognition, while others [63, 65, 98] evaluate identity assignments based on a global one-to-one mapping between predicted and ground truth labels in a small chunk. However, since our method can process the entire sequence of a session, our evaluation scope is significantly larger. Once this mapping is misaligned—especially in the early part of a session—all subsequent predictions may be incorrectly penalized, even if identity tracking later becomes accurate.

To mitigate this issue, we propose a *window-based evaluation protocol* that enables localized assessment of identity consistency. The key idea is to decouple the prediction phase from the evaluation phase: we first perform data association over the entire session to produce a complete sequence of predicted identity labels. We then slide a fixed-length window (by default, the window size $w$ is 16, the stride is 1) across the session and assess identity tracking consistency within each window.

In each window, we extract the resident labels from both the ground truth and predicted sequences, and assign them symbolic codes (A, B, C, ...) based on their

order of first appearance. This yields a one-to-one mapping between raw labels and symbolic identifiers *within* the window. We then relabel both the ground truth and prediction sequences using this mapping, and compute a per-event consistency score. The score for the window is the average consistency across all steps, and the final tracking score is the mean over all windows.

This evaluation strategy preserves temporal context from the full-session prediction, while preventing early errors from dominating the accuracy measurement. It provides a robust means to analyze *cascading error*, as performance degradation under increasing window sizes reflects how well a method maintains identity continuity over time.

#### 3.6.1.2 Correct Run Length (CRL)

To complement the accuracy metric, we introduce CRL as an additional evaluation measure. CRL directly reflects the quality of resident tracking by quantifying **how long a model can continuously maintain correct identity assignments before the first error occurs**. Specifically, for each sliding window and for each ground-truth resident, we count the length of the initial consecutive segment of correct predictions until the first mistake. These lengths are then averaged across all residents and all windows to produce the CRL score. A larger CRL indicates that the model sustains more stable and coherent tracking of residents.

To contextualize CRL, we also report, for reference, the *average number of events per resident*. This auxiliary value allows readers to interpret CRL in terms of the typical sequence length over which the tracking is evaluated, but it is not used as a primary metric.

#### 3.6.1.3 Bilingual Evaluation Understudy (BLEU)

To address the inadequacy of Acc and CRL in measuring the local coherence of separated sequences, we adapt the Natural Language Processing (NLP) metric BLEU [138] to evaluate Data Association. For any mixed sequence, we connect the separated sequence generated by the model using a delimiter (an <EOS> token) to form a candidate sequence. Similarly, we construct a reference sequence using the ground truth in the same way. Given a candidate sequence and a reference sequence, the BLEU metric considers the precision of $N$-grams, which is the proportion of $N$-gram phrases in the candidate sequence that appear in the reference sequence. This evaluation method focuses more on assessing whether the results of the model's separation are coherent and consistent. In our subsequent experiments, we divide the mixed sequence into segments of length 30 and evaluate the BLEU score for each segment. Finally, we compute the average BLEU score across all segments.

#### 3.6.1.4 Resident Count Error (RCE)

Since BLEU and Resident Assignment Accuracy cannot directly reflect the model's real-time assessment of the number of residents, we designed a dedicated error metric for the number of residents in order to systematically evaluated the estimation of the number of individuals in the results. To compute this metric, we divide the mixed sequence into segments of 30 events. We chose 30 as the segment length, which is sufficient to involve multiple residents while still keeping the local representation. For each segment, we count how many different residents the model assigns the $L$ events to, obtaining a predicted value $\hat{K}$. This predicted value is then compared with the ground truth $K$, and the absolute error for each segment is calculated. Finally, the average of these absolute errors across all segments is computed.

The error metric can be expressed as:

$$\text{RCE} = \frac{1}{S} \sum_{i=1}^{S} \left| K_i - \hat{K}_i \right|$$

where $S$ is the total number of segments, $\hat{K}_i$ is the predicted number of residents for the $i$-th segment, and $K_i$ is the ground truth number of residents for the $i$-th segment.

### 3.6.2 Experimental Setup

We conduct experiments on three benchmark datasets (CASAS [59], MARBLE [22], and MuRAL [49]) to compare our proposed method with sMRT [197] and the naive "All-to-One" baseline, which assumes that all events belong to a single resident. In the subsequent ablation study, we investigate the impact of the type and hyperparameters of the search algorithm, the hyperparameters and effectiveness of NEP, and the influence of prior knowledge about the number of residents on the overall performance. To conduct these experiments, we randomly split the sessions of each datasets into training, validation, and test sets with a ratio of 7:1:2.

Since our method is capable of performing data association over event sequences of arbitrary length, we did not apply any data segmentation or preprocessing for the experiments. Instead, the entire event sequence of each data session was fed into the model in one pass. These sequences range in length from 5 to 558 events. Our method ultimately returns the corresponding resident assignment for each event in the sequence.

Using the training set data, we learned the idle time, lambda, and P(N) for each sensor to support the computation of new resident creation introduced in Section 3.5.2. This statistics-based learning process does not require any additional hyperparameters. For the NEP module, we first conducted preliminary experiments on each dataset to find the Transformer-related training hyperparameters and achieve satisfactory prediction accuracy. As the optimal parameters were found to be consistent across

Table 3.2 – Qualitative comparison of different methods. $K$ means the number of residents in the input sequence, $B$ means the Beam Size and $L$ means the context length.

| **Aspect** | Riboni et al.[159] | sMRT/GAMUT[197, 198] | **Ours** |
|---|---|---|---|
| Seq. Length | No continuity | Short (<10) | Unlimited |
| #Residents | No | Yes | Yes |
| Single → Multi | Yes | Yes | Yes |
| #Params. | $< 1K$ | $< 10K$ | 0.3M |
| #Input | 1 | 1 | $K \times B \times L$ |
| #Output | 1 | 1 | $K \times L$ |
| Data Utilization | Inefficient | Inefficient | Efficient |

all three datasets, we utilized a fixed configuration to ensure fair comparisons and minimize confounding variables. The settings are as follows: the context sequence length is set to 4, the number of layers in the transformer decoder is 3, the embedding dimension is 64, the feedforward dimension is 512, the dropout rate is 0.3, the softmax temperature is 1, the learning rate is 1e-4, the number of training epochs is 200, and the batch size is 512. Based on the above default setting, we study in the subsequent experiments the impact of two key hyperparameters: epoch (Section 3.6.6) and context length (Section 3.6.7).

For the MDP search-related hyperparameters, such as the type of search algorithm and the beam width, we conducted comparative experiments to evaluate their impact. Based on the results, we selected beam search with a width of 6 as the default setting for MuRAL [49] and Greedy for CASAS [59] and MARBLE [22]. Detailed experimental results and explanations can be found in Section 3.6.5.

### 3.6.3 Qualitative Comparison with Prior Works

In this section, we qualitatively compare our approach with previous state-of-the-art methods [159, 197, 198] to highlight its advantages. The comparison is summarized in Table 3.2.

**Sequence Length Scalability.** Complete and explicit state modeling is crucial for maintaining the continuity of the separated event sequences. Riboni et al. [159] do not model state at all and only judge whether two consecutive events are triggered by the same resident, resulting in no continuity in the output sequences. In contrast, sMRT [197] and GAMUT [198] employ a first-order Markov modeling of resident states. However, the simplicity of this modeling leads to a strong reliance on observations: in our reproduction experiments, a resident track rarely survives beyond 10 events before being automatically terminated, making it difficult to form long, continuous trajectories. Our approach, by explicitly and incrementally modeling each resident's state, is able to continuously update and maintain the long-term trajectories of residents. This leads to significantly improved scalability of the separated

sequences in terms of sequence length. It is worth noting that the scalability discussed here focuses purely on whether the output sequence remains continuous over time, without considering accuracy; the quantitative evaluation of separation accuracy will be presented in later sections.

**Resident Number Scalability.** All state-of-the-art methods can generalize from single-resident data to multi-resident scenarios. Trained solely on single-resident data (36 out of 76 sessions) of MARBLE [22], our model achieves an accuracy ($w = 16$) of 0.615 and BLEU of 0.616 on 2-resident scenarios. On 4-resident scenarios, our method achieves 0.483 accuracy and 0.523 BLEU with an average of 4.28 events per resident, and a CRL of 2.12. These results indicate that our approach can leverage single-resident data to generalize to multi-resident separation tasks. This property is particularly beneficial in practice, as it enables the collection of data in simple single-resident settings for training, while avoiding the high cost of collecting and annotating multi-resident data. In addition, we evaluated the precision of our method when creating new resident identities. Specifically, for the first event of each newly appearing resident within a session (52, 136, and 63 events in CASAS, MARBLE, and MuRAL datasets respectively), our method achieves a precision of 0.750, 0.662, and 0.683 for correctly creating a new resident at those points.

**Data and Computation Efficiency.** The efficiency of a data association method depends on both how effectively it leverages available data and the computational cost per event. Riboni et al.[159] rely on a transition matrix of adjacent event types to decide whether two events are triggered by the same resident. This requires only $N \times N$ parameters (where $N$ is the number of event types) and a single matrix lookup per decision, making it extremely lightweight computationally. However, this formulation cannot exploit more complex temporal patterns, resulting in a method that is computationally efficient but data inefficient. Our approach is compatible with this style of modeling and can fall back to such lightweight patterns when annotated multi-resident data is scarce. sMRT [197] and GAMUT [198] build on skip-gram embeddings of sensor types and use simple linear operators for state transitions. While slightly more computationally expensive than Riboni et al., these models are still fundamentally linear and therefore unable to exploit the deeper sequential structure in the data. Our method, using an autoregressive language model, requires on average $B \times L \times K$ events as input and $B \times K$ as output per event assignment, where $B$ is the beam size (equal to 1 in greedy decoding) and $L$ is the context length (set to 4 in our experiments). In our implementation, our model has 0.3M parameters, making their computational costs comparable.

### 3.6.4 Quantitative Performance Comparison with Baselines

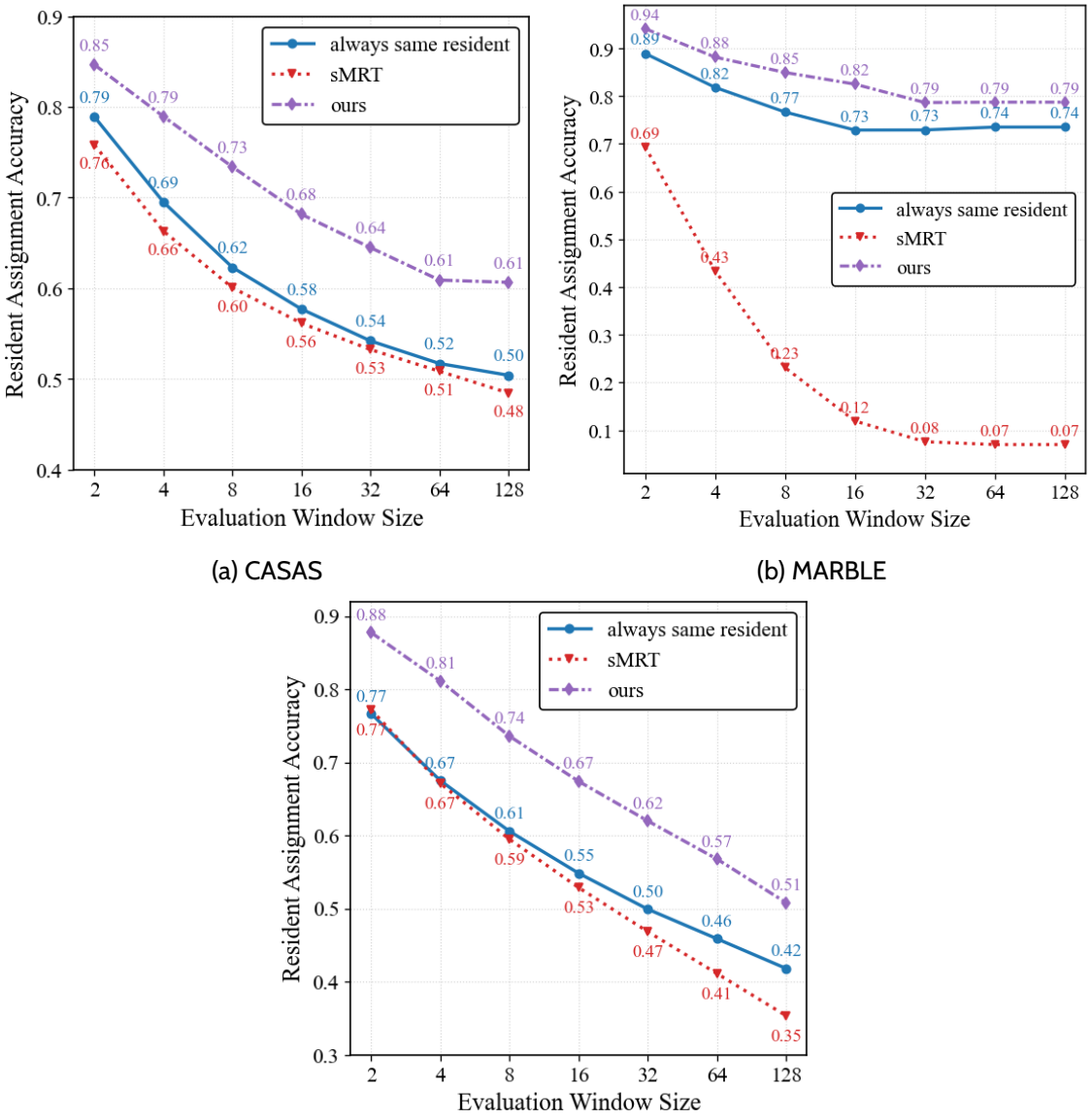
We compare our proposed MDP-based method with baseline sMRT [197] and baseline "All-to-One" baseline which assumes that all events belong to a single resident.

Table 3.3 – Comparison of Data Association performances of baseline approaches and our approach on 3 datasets. Acc and CRL are calculated using a window size $w = 16$.

| Dataset | CASAS [173] | | | | MARBLE [22] | | | | MuRAL [49] | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Metric | Acc | CRL | BLEU | RCE | Acc | CRL | BLEU | RCE | Acc | CRL | BLEU | RCE |
| All-to-One | 0.580 | **5.372** | 0.565 | 0.995 | 0.729 | 8.606 | 0.707 | 0.789 | 0.551 | 4.489 | 0.510 | 1.593 |
| sMRT | 0.553 | 3.060 | 0.532 | 1.227 | 0.119 | 0.641 | 0.079 | 15.10 | 0.529 | 3.321 | 0.497 | 1.560 |
| **Ours** | **0.682** | 4.688 | **0.652** | **0.019** | **0.825** | **8.847** | **0.812** | **0.500** | **0.674** | **4.982** | **0.755** | **0.375** |

Table 3.3 presents the performance comparison on the CASAS, MARBLE, and MuRAL datasets in terms of Acc, CRL, BLEU, and RCE. For Acc and CRL, we adopt a fixed window size of $w = 16$, which balances temporal context with the relatively short session lengths in MARBLE. Overall, sMRT performs significantly worse than our method across all datasets and completely fails on the MARBLE dataset, which contains fewer events per session. On CASAS, our method achieves an accuracy of 0.682, a CRL of 4.688, and the best BLEU score (0.652) with a very small RCE of 0.019, clearly outperforming sMRT. We observe that on the CASAS dataset, our method achieves a slightly lower CRL than the strategy of always predicting the same resident. This is because, under the latter strategy, the correct run length for the first resident equals the full sequence length. When the number of residents is small — for example, only two in CASAS — dividing this length by two still results in a relatively long value. This reflects an inherent advantage of always predicting the same resident under this particular metric. On the MARBLE dataset, sMRT fails completely because the trajectories within each session are too short, providing insufficient observations to form coherent tracks. As a result, it assigns almost every event to a new resident ID, leading to a complete breakdown of the results. This is reflected by the RCE reaching 15.1 within a window of only 16 events. However, our method remains consistently robust, achieving performance comparable to that on other datasets across all metrics, with a CRL as high as 8.8. On MuRAL, our method again achieves the best performance with 0.674 accuracy, 4.982 CRL and 0.755 BLEU, and lowers the RCE to 0.375. Overall, these results consistently demonstrate that our MDP-based approach produces more accurate event-to-resident assignments, generates more coherent per-resident sequences, and delivers more precise resident count estimation compared to the existing methods.

To further examine the impact of different evaluation window sizes and the cumulative effect of errors as sequence length increases, we compare the Acc and CRL of sMRT and our approach in Figure 3.5 and Figure 3.6, respectively. For better visualization, the horizontal axis of both figures is plotted on a $\log_2$ scale. As shown in Figure 3.5, the accuracy of all methods decreases rapidly as the window size increases. This trend arises because each data association decision depends on previous assignments, causing earlier mistakes to propagate and reduce subsequent accuracy. In other words, the longer the sequence, the more difficult it becomes to maintain correct resident assignments. A similar phenomenon can be observed in Figure 3.6 for CRL, which measures the average sequence length of correct predictions



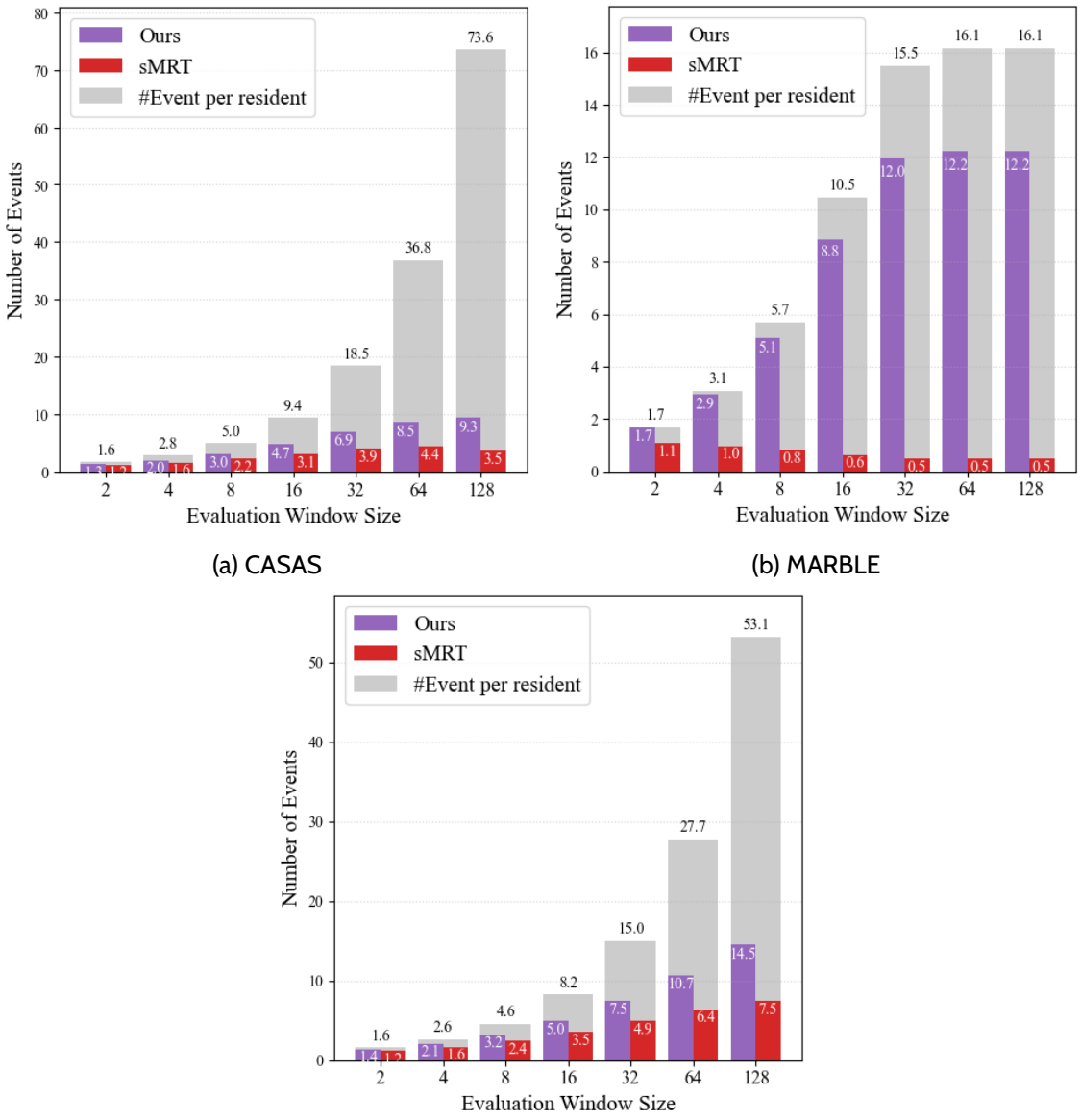

(a) CASAS

(b) MARBLE

(c) MuRAL

Figure 3.5 – Resident assignment accuracy as a function of the evaluation window size ($\log_2$ scale) on the CASAS, MARBLE, and MuRAL datasets.

before the first error occurs for each resident. As the window size increases, the average number of events per resident also grows proportionally, but neither sMRT nor our method can sustain correctness at the same rate, resulting in a smaller proportion of correctly predicted events in each resident's sequence. This highlights the inherent cascading error effect. Nevertheless, our approach demonstrates a noticeably slower degradation of both Acc and CRL compared to sMRT as the window length increases, showing stronger tracking ability. It is worth noting that when the window size exceeds 16, the Acc and CRL results on MARBLE become less meaningful because the




(a) CASAS

(b) MARBLE

(c) MuRAL

Figure 3.6 – Evaluation of Correct Run Length (CRL) and the average number of events per resident (gray bars) in different evaluation window sizes ($\log_2$ scale) for CASAS, MARBLE, and MuRAL. CRL quantifies how long a method can continuously assign the correct resident before making an error.

average length of MARBLE sessions is 26.21, which is shorter than 32 events, leading to statistical saturation. This can be seen from the growth of the average number of events per resident beyond $w = 16$: if the growth were proportional, the average would increase to about 14 events per resident at $w = 32$, whereas in reality it only reaches about 12. This confirms the saturation effect of MARBLE at $w = 32$.

In summary, our MDP-based approach delivers **higher assignment accuracy**,

Table 3.4 – Impact of search algorithms on the Data Association performance of our method.

| Dataset | Metric | Greedy | BS 3 | BS 6 | BS 9 | BS 12 |
|---|---|---|---|---|---|---|
| CASAS [173] | Acc | **0.682** | 0.665 | 0.666 | 0.666 | 0.665 |
| | CRL | **4.688** | 4.028 | 3.975 | 3.963 | 3.944 |
| | BLEU | **0.652** | 0.614 | 0.618 | 0.608 | 0.613 |
| | RCE | **0.019** | 0.019 | 0.019 | 0.019 | 0.019 |
| MARBLE [22] | Acc | **0.825** | 0.695 | 0.690 | 0.687 | 0.690 |
| | CRL | **8.847** | 5.705 | 5.654 | 5.686 | 5.533 |
| | BLEU | **0.812** | 0.728 | 0.731 | 0.721 | 0.678 |
| | RCE | **0.500** | 0.588 | 0.591 | 0.593 | 0.598 |
| MuRAL [49] | Acc | 0.671 | 0.654 | **0.674** | 0.676 | 0.658 |
| | CRL | 4.963 | 4.854 | **4.982** | 4.954 | 4.819 |
| | BLEU | 0.748 | 0.741 | **0.755** | 0.749 | 0.746 |
| | RCE | **0.410** | 0.554 | 0.429 | 0.446 | 0.464 |

**stronger sequential coherence**, and **greater long-term stability** than the baselines, as evidenced both by overall metrics (Table 3.3) and detailed window-based analyses (Figures 3.5 and 3.6).

### 3.6.5 Impact of the Search Algorithm

In this section, we experimentally study the influence of different search strategies: Greedy, Beam Search with sizes 3, 6, 9, and 12 (noted respectively as BS 3, BS 6, BS 9, and BS 12). The comparison results are reported in Table 3.4, which shows a clear difference across datasets: greedy search works best on CASAS and MARBLE, whereas a moderate beam size (BS 6) gives the best results on MuRAL.

For assignments that already have very low plausibility (i.e., are highly likely to be incorrect), greedy search directly filters them out, whereas beam search temporarily retains such assignments. However, these likely incorrect assignments introduce events that probably do not belong to a given resident into that resident's historical state. Such events act as noise and subsequently affect reward estimation and the search process. As a result, although search-based methods can prevent the policy from prematurely converging to a local optimum, their negative impact on NEP can also propagate and influence the final performance.

As shown in the results in Table 3.4, this negative effect outweighs the benefit of avoiding local optima on both the CASAS and MARBLE datasets, and is particularly pronounced on MARBLE. According to our observations, the MARBLE event sequences are relatively clean and simple, with strong correlations between consecutive events and little inherent noise in the training data. Consequently, NEP on MARBLE is especially sensitive to the noise introduced by beam search, leading to performance degradation. In contrast, CASAS features a dense and uniformly distributed sensor

Table 3.5 – Comparison of two different Next Event Prediction Methods.

| Dataset | CASAS [173] | | | | MARBLE [22] | | | | MuRAL [49] | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Metric | Acc | CRL | BLEU | RCE | Acc | CRL | BLEU | RCE | Acc | CRL | BLEU | RCE |
| TM | 0.642 | 4.231 | 0.617 | 0.185 | 0.769 | 7.562 | 0.798 | **0.364** | 0.635 | 3.738 | 0.696 | 0.411 |
| **ALM** | **0.682** | **4.688** | **0.652** | **0.019** | **0.825** | **8.847** | **0.812** | 0.500 | **0.674** | **4.982** | **0.755** | **0.375** |

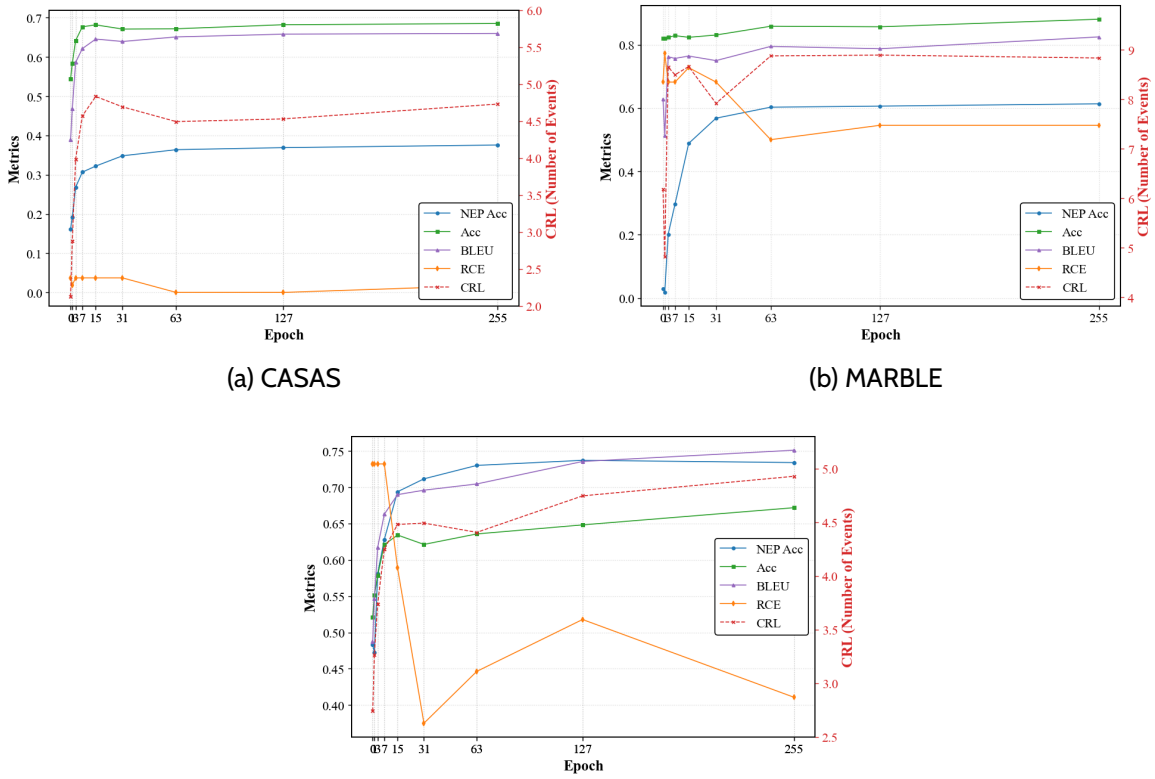
layout, where nearby sensors are often strongly correlated. This property makes locally optimal assignment solutions typically coincide with globally optimal ones, and additional search therefore tends to introduce unnecessary noise rather than yield benefits. The MuRAL dataset, on the other hand, exhibits much higher complexity: a given sensor event can be followed by many plausible subsequent events, and the average number of residents reaches three, which is higher than in the other two datasets. This increased complexity makes the positive effects of broader search more meaningful, allowing beam search to slightly outperform greedy search.

### 3.6.6 Impact of Next Event Prediction

In our proposed method, NEP is used to construct the reward function, providing feedback to guide the agent's assignment decisions. To investigate the impact of NEP on the overall performance, we first compare the results of two different NEP implementations: Unsupervised Transition Matrix (TM), and Supervised Autoregressive Language Model (ALM). The results are presented in Table 3.5.

By comparing two different implementations of the NEP module, we observe that the ALM consistently outperforms the TM across all metrics, except for the RCE on the MARBLE dataset. This aligns with intuition, as ALM is trained in a supervised manner using ground-truth resident assignments, whereas TM is an unsupervised method relying solely on event transition statistics in multi-resident scenarios. These results indicate that accurate NEP enhances the effectiveness of our framework. Notably, even without access to labeled data, the TM-based approach achieves competitive performance. Overall, these findings demonstrate that our framework remains effective even without assignment supervision, and has strong potential for deployment in settings where single-resident scenarios are more prevalent.

To further investigate the impact of NEP quality on our method, we trained the ALM for varying numbers of epochs (from 1 to 256), thereby obtaining models with different levels of predictive accuracy, as reflected by the "NEP Acc" curve in Figure 3.7. The results show that as the NEP accuracy improves, the overall performance of data association also increases accordingly. Notably, BLEU shows the most significant improvement among the evaluation metrics, which suggests that NEP plays a crucial role in ensuring the semantic coherence and consistency of the separated per-resident event sequences. This finding is consistent with the design intuition behind NEP: by modeling the likelihood of each resident's individual event sequence, the model captures sequence continuity, which is then used as a reward signal to guide the




(a) CASAS

(b) MARBLE

(c) MuRAL

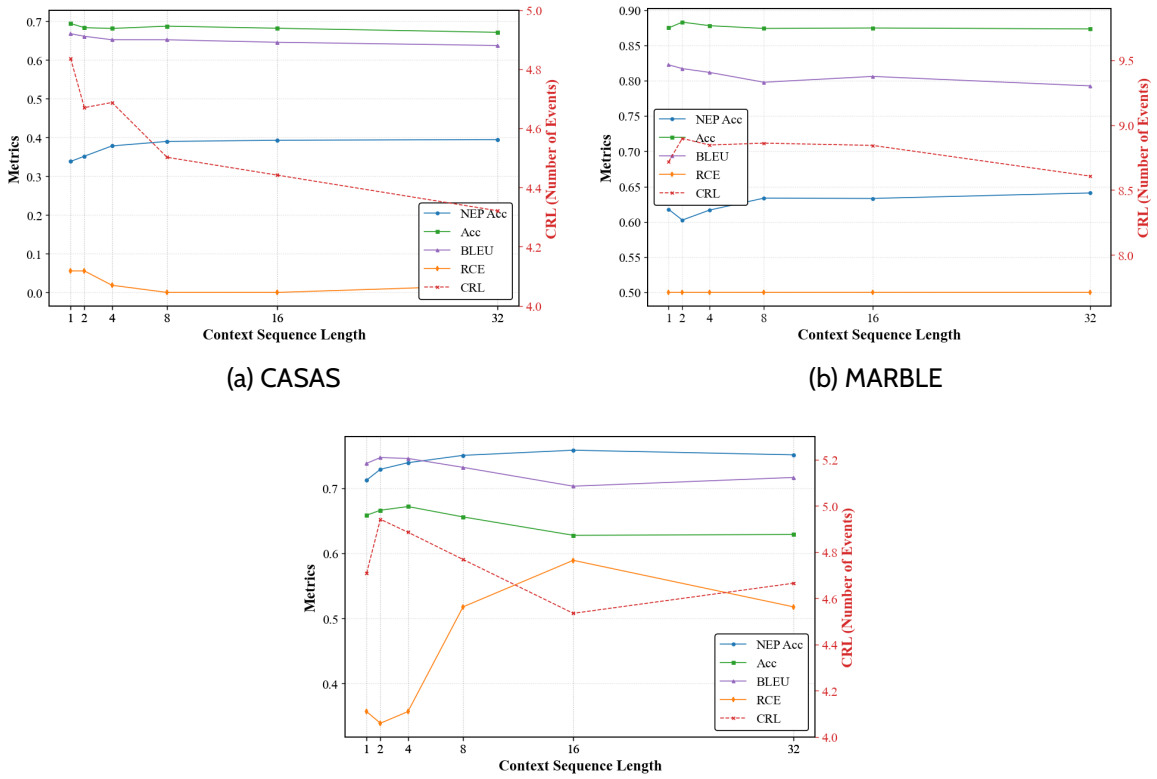
Figure 3.7 – Evaluation of the impact of the number of NEP training epochs on the data association performance of the proposed MDP-based framework

agent's assignment policy. Accurate NEP thus enhance the informativeness of the reward function and enable more effective search for plausible separation trajectories.

### 3.6.7 Impact of Context Length

One interesting hyperparameter in our framework is the context length $L$ used in the NEP module. Intuitively, increasing $L$ provides the NEP model with more historical context, which may improve prediction accuracy and thereby influence downstream data association performance. To investigate this effect, we varied $L$ from 1 to 32 and evaluated both NEP accuracy and final assignment metrics across the three datasets. The results are presented in Figure 3.8.

As shown, NEP accuracy improves with longer context across all datasets, peaking around $L = 16$. Beyond this point, further gains are minimal. However, we observe a surprising and counterintuitive trend in the downstream Data Association performance. For example, while NEP accuracy on the MuRAL dataset increases from approximately 0.71 at $L = 1$ to 0.78 at $L = 16$, the Acc drops from 0.67 to 0.63, the CRL drops from 4.7 to 4.5, RCE simultaneously degrades from 0.35 to 0.59, and BLEU score drops




(a) CASAS

(b) MARBLE

(c) MuRAL

Figure 3.8 – Evaluation of the impact of the context length on the data association performance of the proposed MDP-based framework

from 0.74 to 0.70. This divergence suggests that, although longer contexts help the NEP module make more confident predictions at train time, they may inadvertently harm the global sequence assignment at inference time. We believe that an excessively long context length increases the likelihood of errors in residents' assignment histories during inference, causing these errors to further affect subsequent assignments and thereby exacerbating error accumulation. A quantitative validation of this hypothesis will be presented in the next chapter.

### 3.6.8 Sensitivity to the Resident Number Prior Distribution

As said in Section 3.5.2, the prior distribution over the number of residents $P(N)$ influences the probability $P(e_t \mid u_{\text{new}}, h_{<t})$, which determines whether a sensor event should be attributed to an existing resident or a newly introduced resident. This prior directly affects the MDP's reward computation and thereby steers the decision trajectory explored by the search algorithm. Misestimating the resident number prior can lead to over-segmentation (creating unnecessary residents) or under-segmentation (failing to detect new ones), especially in dynamic multi-resident environments.

Table 3.6 – Impact of the resident number prior $P(N)$ on Data Association performance of our approach.

| Dataset | CASAS [173] | | | | MARBLE [22] | | | | MuRAL [49] | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Metric | Acc | CRL | BLEU | RCE | Acc | CRL | BLEU | RCE | Acc | CRL | BLEU | RCE |
| Single-resident | 0.599 | 5.561 | 0.600 | 0.963 | 0.764 | 8.660 | 0.764 | 0.727 | 0.667 | 4.092 | 0.736 | 0.714 |
| Uniform | 0.612 | 3.722 | 0.594 | 0.593 | 0.827 | 8.861 | 0.798 | 0.500 | 0.637 | 3.957 | 0.732 | 0.625 |
| Data-driven | 0.682 | 4.688 | 0.652 | 0.019 | 0.825 | 8.847 | 0.812 | 0.500 | 0.674 | 4.982 | 0.755 | 0.375 |

We evaluate the impact of this prior under three configurations:

— **Single-resident prior**: assumes exactly one resident is present with probability 1.

— **Uniform prior**: assumes the number of residents is uniformly distributed from 1 to 5.

— **Data-driven prior**: estimated from the training data, reflecting the empirical distribution of resident counts.

Table 3.6 summarizes the performance across the three datasets. On the MuRAL dataset, where the number of residents per session varies between 1 and 4 (with 2 and 3 being the most frequent), the data-driven prior achieves the best RCE, confirming its alignment with actual test distribution. In contrast, the CASAS dataset always contains two residents, and the data-driven priors perform best, while the single-resident prior results in a larger RCE nearly up to 1. This is reasonable: in the CASAS dataset there are always two residents, so when the prior is set to 1, the average resident count error is close to 1. Interestingly, in the MARBLE dataset—where resident count ranges from 1 to 4, but single-resident sessions are most frequent—the single-resident prior slightly outperforms data-driven, though both remain competitive.

These results confirm that the resident number prior $P(N)$ has a clear influence on the system's behavior. However, they also demonstrate the robustness of our method to prior mis-specification. Even under extreme settings—such as applying a uniform 5-resident prior to the strictly 2-resident CASAS dataset—the Data Association performance does not collapse. Similarly, the single-resident prior, though overly restrictive, performs comparably to the data-driven setting in CASAS, and remains reasonably effective in MARBLE. This robustness can be attributed to the MDP formulation and the search strategy: even when the prior nudges the model toward creating new residents, inconsistent sequences are penalized by the reward model and filtered by the search. Conversely, plausible new-resident hypotheses can still be recovered when the prior is under-expressive, if their event sequences are semantically coherent.

In practical deployments, resident count distributions may depend on contextual variables such as time of day or day of the week, or reflect rare events such as visitors. While a data-driven prior yields good results in most cases, future work could explore context-conditioned or online adaptive priors, as well as learning priors jointly with the reward model. Overall, our findings suggest that while the resident number prior meaningfully affects resident count estimation, the system maintains

performance across a wide range of prior settings, highlighting the practical viability of the approach.

## 3.7 Conclusion

In this chapter, we have addressed the critical challenge of Data Association in multi-resident smart home environments—a task essential for transforming raw, interleaved sensor streams into coherent, person-specific activity trajectories. By tracing the evolution of data association through the lens of system state modeling, we identified a fundamental limitation in existing state-of-the-art latent-state filters. Specifically, we highlighted the structural mismatch between filtering-based approaches (e.g., sMRT [197] and GAMUT [198])—originally designed for continuous tracking—and the sporadic, discrete nature of ambient sensor events. In such frameworks, the lack of frequent corrective observations forces an over-reliance on open-loop prediction. This causes the estimated resident state to become excessively diffuse during inactive periods, leading to high uncertainty that inevitably breaks trajectory continuity.

To overcome these structural weaknesses, we introduced a novel framework that reformulates data association as an MDP. Unlike recursive filters that struggle with observation gaps, our MDP formulation explicitly encodes the complete assignment history and treats data association as a global optimization problem. We formally proved that the optimal policy of this MDP is equivalent to the solution of the data association task.

Practically, we implemented this framework using a heuristic tree search guided by a likelihood-based reward model. This mechanism effectively replaces the diverging uncertainty of filters with a robust decision-making process that dynamically balances the likelihood of extending existing resident trajectories against the plausibility of creating new ones. Furthermore, by integrating single-resident event sequence modeling, our approach captures long-range temporal dependencies and behavioral regularities that purely local embeddings fail to represent.

Our extensive evaluation across three diverse datasets—CASAS, MARBLE, and MuRAL—consistently demonstrated that the proposed method outperforms state-of-the-art baselines. These results confirm that moving from continuous filtering to a discrete, decision-theoretic MDP framework significantly enhances the robustness of multi-resident Data Association systems, particularly in open-world settings characterized by sensor noise and an unknown number of residents.

Ultimately, this work provides a scalable and theoretically grounded solution for identity tracking in pervasive computing. Future research could explore extending this framework to real-time, online settings where computational efficiency is paramount, or integrating more complex inter-resident interaction models into the reward function to further refine multi-agent behavioral analysis.

# CHAPTER 4

# LEARNING ROBUST POLICIES FOR MULTI-RESIDENT DATA ASSOCIATION: FROM HEURISTIC SEARCHING TO BEHAVIOR CLONING

## 4.1 Introduction

In the previous chapter, we formulated the multi-resident sensor-event Data Association problem as a Markov Decision Process (MDP) and introduced a heuristic search approach guided by a Next-Event Prediction (NEP) model to approximate the reward function and plan for optimal policies. This approach is hereafter referred to as **NEP+Search**. It provides a clear, rigorous, and theoretically grounded decision-making framework. However, when applied to real-world scenarios, the NEP+Search approach exhibits several key limitations:

**1. The policy relies on heuristic signals that are not directly aligned with resident assignment decisions.** In the NEP+Search framework, the NEP model provides heuristic scores by estimating the likelihood of the next sensor event based on existing resident-specific event histories, rather than directly modeling the decision of assigning an event to residents or creating a new resident. This indirect alignment becomes especially problematic when new residents must be introduced: since NEP

extrapolates from known resident sequences, the search procedure must rely on ad hoc case analyses—such as whether the triggered sensor is a boundary sensor or whether any current resident could plausibly trigger it, none of which are explicitly learned from data. As a result, the policy is guided by proxy heuristic signals instead of decisions grounded in the true resident–event assignment distribution, potentially biasing the search away from the underlying data-generating process.

**2. The next-event prediction model lacks a global multi-resident perspective.** The NEP model conditions its predictions on the event sequence of a single resident at a time, without access to the joint event histories of all residents in the environment. As a result, it cannot directly compare how well the current event matches different residents from a global system-level perspective, nor can it, in principle, model interactions among residents such as competition, coordination, or mutual exclusion. While the subsequent search procedure implicitly considers the global system state by evaluating multiple residents in parallel, this comparison is performed externally by the search algorithm rather than being learned in a data-driven manner by the model itself. From a statistical learning perspective, conditioning only on single-resident sequences forces NEP to learn from a more ambiguous and marginalised data distribution, leading to probability estimates that are less discriminative and harder to refine with respect to subtle differences among competing resident hypotheses.

**3. The NEP+Search framework incurs substantial redundant computation.** For each incoming event, the NEP model must be evaluated separately for every candidate resident in order to estimate the likelihood of that resident triggering the event. With $K$ active residents, this requires $K$ forward passes of the prediction model per event. When combined with a beam search of width $B$, the total number of forward evaluations scales to $K \times B$ per decision step. These repeated evaluations largely differ only in the resident-specific event history while sharing the same global context, leading to significant computational redundancy. Such a computation pattern scales poorly as the number of residents or the search width increases, making the NEP+Search framework inefficient for long sequences or densely populated multi-resident environments.

The above limitations stem from a fundamental mismatch between the NEP+Search paradigm and the underlying data association problem. By relying on per-resident next-event heuristics, NEP+Search neither aligns its learning objective with the resident assignment decision, nor conditions on the full multi-resident system state in a unified manner. Moreover, its search-based inference introduces substantial redundant computation that scales poorly with the number of residents and the search width.

To overcome these limitations of heuristic-guided planning, we propose a behavior cloning framework that learns a direct multi-resident assignment policy from expert annotations. By modeling resident assignment as a supervised sequence prediction problem, the proposed policy is explicitly aligned with the data association decision and operates on the full multi-resident state in a single forward pass. This design

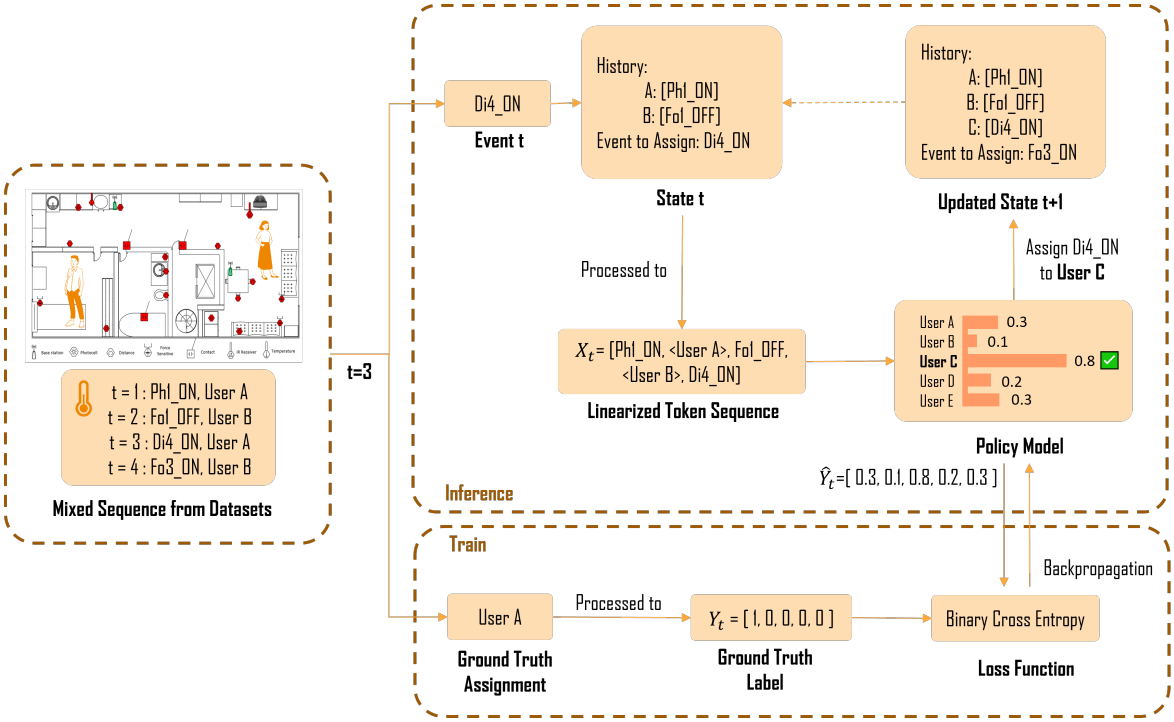



Figure 4.1 – Overview of the proposed behavior cloning policy-learning data association framework.

enables the model to jointly reason over all residents, naturally captures inter-resident interactions, and avoids the computational redundancy inherent to per-resident heuristic evaluation. Empirically, the learned policy consistently achieves strong performance across multiple real-world smart home datasets, with particularly clear advantages in environments featuring frequent multi-resident interactions and sufficient training data.

This work makes three main contributions. First, we formulate multi-resident sensor-event data association as a direct policy learning problem and introduce a Behavior Cloning (BC) [134] approach that eliminates heuristic reward approximation and online search. Second, we design a Transformer-based policy architecture that encodes the complete multi-resident event context and produces joint resident assignment decisions in a single inference step, including explicit support for new resident emergence. Third, through extensive experimental evaluation, we demonstrate that supervised policy learning provides a scalable and effective alternative to heuristic search, achieving improved alignment with the underlying data distribution and robust performance on diverse multi-resident smart home datasets.

## 4.2 Method

In this section, we introduce a direct policy-learning approach for multi-resident data association. Motivated by the limitations of heuristic NEP+Search planning, we formulate resident assignment as a supervised sequence modeling problem and learn a policy that directly maps the global multi-resident state to assignment actions.

### 4.2.1 Overview

Figure 4.1 shows an overview of the proposed framework. It processes a stream of sensor events from a smart home and assigns each incoming event to a resident set based on previously inferred event histories. When a new event arrives, the system combines it with the existing resident-specific histories to form the current linearized token sequence input to the policy model. The policy model outputs an assignment probability for each resident, and the event is assigned to the residents with high confidence. The selected assignment is then used to update resident histories for subsequent events. During training, the model learns from ground-truth resident assignments using a supervised loss, enabling a unified pipeline for both training and inference.

### 4.2.2 Learning Objective

Our goal is to learn a resident-assignment policy whose decision distribution is directly aligned with the empirical distribution of ground-truth assignments observed in data. To this end, we adopt BC [134], which frames policy learning as supervised imitation of an expert policy and explicitly minimizes the divergence between the learned policy and the expert's action distribution. Unlike heuristic planning or reward-based optimization, BC optimizes the policy to reproduce expert decisions at the action level, thereby achieving maximal alignment between model predictions and ground-truth resident assignments.

In our MDP formulation introduced in Chapter 3, a policy $\pi(s_t) = a_t$ specifies the assignment action taken at event step $t$, where $a_t$ assigns the current event $e_t$ to a resident set $\widehat{\mathcal{U}}_t$. An expert policy $\pi^*$ deterministically assigns each state $s_t$ to the ground-truth resident set $\mathcal{U}_t$. The objective of BC is to learn a parametric policy $\pi_\theta$ whose action distribution aligns with that of the expert policy $\pi^*$.

Formally, the learned policy $\pi_\theta$ induces a conditional distribution over assignment actions,

$$P_\theta(a_t \mid s_t),$$

while the expert policy $\pi^*$ defines the target action

$$a_t^* = \pi^*(s_t) = \mathcal{U}_t.$$

BC therefore optimizes the parameters $\theta$ of policy $\pi_\theta$ by maximizing the likelihood of the expert actions:

$$\theta^{\mathrm{BC}} = \arg\max_\theta \sum_{t=1}^{T} \log P_\theta(a_t^* \mid s_t) = \arg\max_\theta \sum_{t=1}^{T} \log P_\theta(\mathcal{U}_t \mid s_t). \tag{4.1}$$

Let $\mathbb{U} = \{u_1, \ldots, u_K\}$ denote the universal set of candidate residents, where $K$ is the total size of the candidate pool. Therefore, the assigned resident set $\mathcal{U}_t$ is a subset of $\mathbb{U}$. While resident assignments can theoretically exhibit inter-dependencies, we assume the assignment probability of each resident is conditionally independent given the state $s_t$. This is justified because our state representation $s_t$ implicitly captures these dependencies by encoding the comprehensive history context of all residents. The log-likelihood of the expert action can then be factorized over residents:

$$\log P_\theta(\mathcal{U}_t \mid s_t) = \sum_{u_k \in \mathcal{U}_t} \log P_\theta(u_k \mid s_t) + \sum_{u_k \notin \mathcal{U}_t} \log(1 - P_\theta(u_k \mid s_t)) \,. \tag{4.2}$$

To express this in standard supervised-learning form, we define binary labels

$$y_{t,k} = \begin{cases} 1, & u_k \in \mathcal{U}_t, \\ 0, & u_k \notin \mathcal{U}_t, \end{cases}$$

indicating whether resident $u_k$ is part of the expert action at step $t$. Substituting into Equation 4.2 yields the equivalent **Binary Cross-Entropy (BCE) loss**:

$$\theta^{\mathrm{BC}} = \arg\min_\theta \sum_{t=1}^{T} \sum_{u_k \in \mathbb{U}} \Big[ - y_{t,k} \log P_\theta(u_k \mid s_t) - (1 - y_{t,k}) \log \left(1 - P_\theta(u_k \mid s_t)\right) \Big]. \tag{4.3}$$

This BCE objective is mathematically equivalent to the objective in Eq. 4.1, but it adopts the conventional multi-label classification paradigm widely used in neural-network training. Importantly, BCE loss naturally supports multi-resident assignment: each resident $u_k$ is treated as an independent label associated with a binary classifier, implemented via a sigmoid output $\hat{y}_{t,k} = P_\theta(u_k \mid s_t)$. During inference, the learned policy produces one probability for each resident, and residents with scores above a threshold $\tau$ (e.g., $\tau = 0.5$ in this study) are selected as the predicted assignment set. This threshold-based decoding aligns exactly with the action semantics of our MDP, enabling the policy to output arbitrary subsets of residents.

Optimizing this BCE objective trains the policy to imitate the expert's action behavior directly, without requiring environment interaction or explicit reward optimization. Consequently, Behavior Cloning offers a simple, effective, and data-efficient approach for recovering an approximation of the ground-truth policy $\pi^*$ in multi-resident data association.

### 4.2.3 Input State Representation

Given the learning objective in our BC setting, the policy model $\pi_\theta$ takes as input the state $s_t$ and outputs the per-resident assignment probabilities $P_\theta(u_k \mid s_t)$. As formalized in Section 3.3.2, the state at step $t$ is defined as $s_t = (h_{<t}, e_t)$, where $h_{<t}$ summarizes all previously assigned events for every resident, and $e_t$ denotes the current event awaiting assignment. Designing an effective representation for this state is non-trivial: it contains multiple event sequences across potentially several residents, each with its own temporal dynamics, while the assignment policy model requires a unified, machine-readable input.

Since our policy model is based on sequence modeling, we convert this structured, multi-resident state into a flat token sequence. Doing so requires a careful linearization strategy that preserves the hierarchical relationships within the state. We denote the maximum allowable resident set by $\mathbb{U} = \{u_1, u_2, \ldots, u_K\}$, where $K$ represents an upper bound on the number of potential residents in the environment. However, at any step $t$, the system may have identified only a subset of these residents through its past assignment decisions. Let $n \leq K$ denote the number of distinct residents that have been identified so far and that possess at least one previously assigned event. These identified residents form the relevant population for state encoding. Residents who have not yet appeared in the system, and thus have no event history, are excluded from the representation, which keeps the input sequence compact and avoids unnecessary padding or noise.

The linearized token sequence is constructed in three stages:

**(1) History extraction.** For each active resident $u_k$, we extract the most recent $L$ events from their historical sequence to form a resident-specific local context:

$$[e_1^{u_k}, e_2^{u_k}, \ldots, e_L^{u_k}].$$

Here, the context length $L$ is a configurable window size that limits computational complexity while ensuring that the model has access to the most informative portion of each resident's history. Importantly, when a resident has fewer than $L$ past events, we include all available events without applying any padding, thus maintaining a faithful representation of the underlying data.

**(2) Resident delimiters.** To clearly demarcate the boundaries between residents, we insert a special delimiter token <$u_k$> after the event sequence belonging to resident $u_k$. These explicit delimiters help the sequence model learn resident-specific behavioral patterns and ensure that signals from different residents' histories remain distinguishable, even after linearization.

**(3) Current event.** Finally, the current event $e_t$, which is the focal point for the decision at step $t$, is appended as the last token in the sequence. This design allows attention-based models to treat the current event as a query token that can selectively attend to relevant segments of the aggregated resident histories.

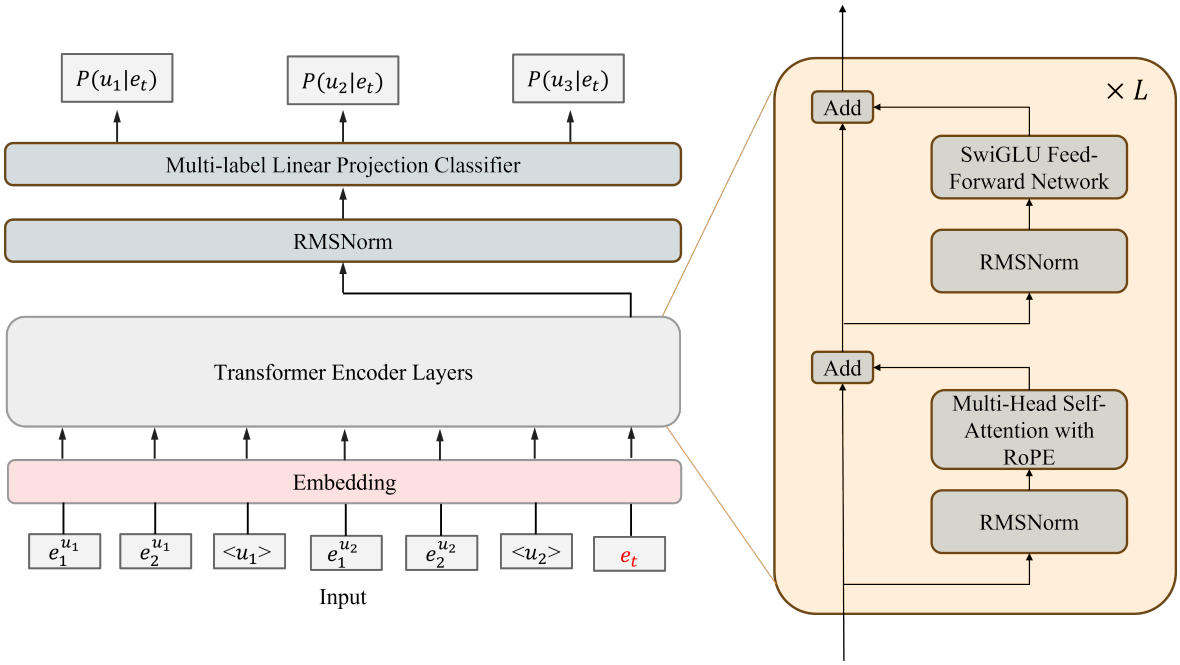



Figure 4.2 – Architecture of the proposed Transformer-based policy network.

Formally, the resulting linearized sequence is:

$$X_t = \left[e_1^{u_1}, \ldots, e_L^{u_1}, <u_1>,\ e_1^{u_2}, \ldots, e_L^{u_2}, <u_2>,\ \ldots,\ e_1^{u_n}, \ldots, e_L^{u_n}, <u_n>,\ e_t\right],$$

where $n$ denotes the number of known residents at time $t$.

This flattened state representation enables the policy model to process the entire multi-resident context in a single forward pass. More importantly, it allows the current event token $e_t$ to directly attend to and semantically compare itself with the historical activity traces of each identified resident. Through this mechanism, the model can evaluate how well the current event aligns with the behavioral patterns, temporal dynamics, and contextual cues associated with each resident $u_k$. In doing so, the model not only captures fine-grained temporal structure within individual histories but, critically, learns to match $e_t$ to the most plausible resident based on these learned semantic relationships.

### 4.2.4 Network Architecture

The policy model receives the linearized state representation introduced in the previous subsection and maps it to a set of assignment probabilities $P_\theta(u_k \mid e_t)$ for the residents identified at step $t$. The architecture consists of an embedding module, a stack of transformer encoder layers, a normalization layer, and a multi label linear projection classifier. Figure 4.2 presents an overview of the entire model architecture.

**Input Token Embeddings.** Each token in the linearized input sequence $X_t$ is first converted into a one-hot vector that represents its discrete identity in the token vocabulary. This sparse representation is then projected into a continuous vector space through a learned linear transformation, which serves as the embedding matrix of the model. The embedding module therefore produces dense vector representations for all types of tokens in the sequence, including resident specific event tokens, resident delimiter tokens such as $<u_k>$, and the current event token $e_t$. These embeddings form the input to the transformer encoder and provide the model with a continuous representation on which attention and subsequent layers can operate effectively.

**Transformer Encoder Layers.** The embedded sequence is processed by a series of transformer encoder layers. Each layer contains a Multi-Head Self-Attention module [193] and a Swish-Gated Linear Unit (SwiGLU) Feed-Forward module [168]. The SwiGLU Feed-Forward module can be expressed as:

$$\text{SwiGLU}(x) = \text{Linear}_3(\text{Linear}_1(x) \odot \text{Swish}(\text{Linear}_2(x))),$$

where $\odot$ denotes element-wise multiplication. By replacing the standard activation of the conventional Transformer Feed Forward Network [193] with a gated linear unit using a Swish function [152], the resulting SwiGLU module improves the expressive power of the network by introducing multiplicative interactions between feature dimensions, allowing the model to capture more complex dependencies within the latent representations.

Both the Multi-Head Self-Attention and SwiGLU Feed-Forward modules utilize a Pre-Normalization architecture: they are preceded by Root Mean Square Layer Normalization (RMSNorm) [219] and followed by residual connections. This design significantly improves training stability and facilitates the training of deep encoder stacks.

Within the self-attention module, we incorporate Rotary Positional Encoding (RoPE) [175] by applying it to the query and key vectors. Unlike standard additive embeddings, RoPE introduces relative positional information multiplicatively via rotation matrices. This allows the attention mechanism to naturally capture the temporal structure of the sequence during pairwise computations.

Through repeated applications of self attention enhanced by rotary positional encoding, the current event token $e_t$ learns to attend to all historical events belonging to all identified residents. This enables the model to identify relevant semantic relations between the current event and the activity patterns expressed in each resident's history. Consequently, the model can infer whether the current event is temporally consistent with a resident's recent activities, whether it continues an ongoing behavioral pattern, and whether it aligns with contextual cues that have appeared earlier in the sequence.

**Output Representation.** The decision for resident assignment concerns only the relationship between the current event and the historical activity traces encoded earlier in the sequence. All relevant contextual information flows into the representation of the final token through the self attention mechanism. The current event token attends to all preceding tokens in the sequence, including the events associated with each identified resident and their corresponding delimiter tokens. Through this process, its final hidden state incorporates both semantic and temporal evidence drawn from the entire resident history. For this reason, the hidden state associated with the current event token is considered to be the output representation that contains all relevant information for the decision.

Specifically, at the end of the final transformer encoder layer, the model applies RMSNorm only to the hidden state located at the last position of the sequence. This position corresponds to the current event token $e_t$, which is the target of the assignment decision at step $t$. The normalized hidden state at this position is extracted as the summary representation $h_t$. The vector $h_t$ is subsequently used as the sole input to the classification module.

**Multi Label Linear Projection Classifier.** The final component of the model is a linear projection classifier that produces assignment probabilities for a predefined maximum of $K$ residents, where $K$ specifies an upper bound on the number of possible residents in the environment. The classifier receives the summary representation $h_t$ and computes

$$P_\theta(u_k \mid e_t) = \sigma\big(W_k^\top h_t + b_k\big),$$

for each $k$ in the range from 1 to $K$. Here $W_k$ and $b_k$ are learnable parameters and $\sigma$ denotes the sigmoid function. All $K$ probabilities are produced at every step, but only the first $n$ correspond to residents that have been identified so far through previous assignment decisions. The probability at position $n + 1$ is reserved for a potential new resident. A non-negligible value at this position indicates that the model predicts the current event to originate from an unseen resident. The outputs at positions greater than $n + 1$ do not correspond to any meaningful resident identity and are expected to be driven toward zero during training.

This formulation allows the classifier to represent both known residents and the possibility of discovering new ones, while maintaining a fixed output dimensionality across all steps. When combined with the contextual representation produced by the transformer encoder, the architecture supports flexible and accurate multi resident event assignment in dynamic smart home environments.

### 4.2.5 Training Data Preparation

To train the policy model, we construct a supervised dataset in which each sample contains a structured representation of the historical event context and the corresponding resident assignment label for the current event. The data preparation pipeline

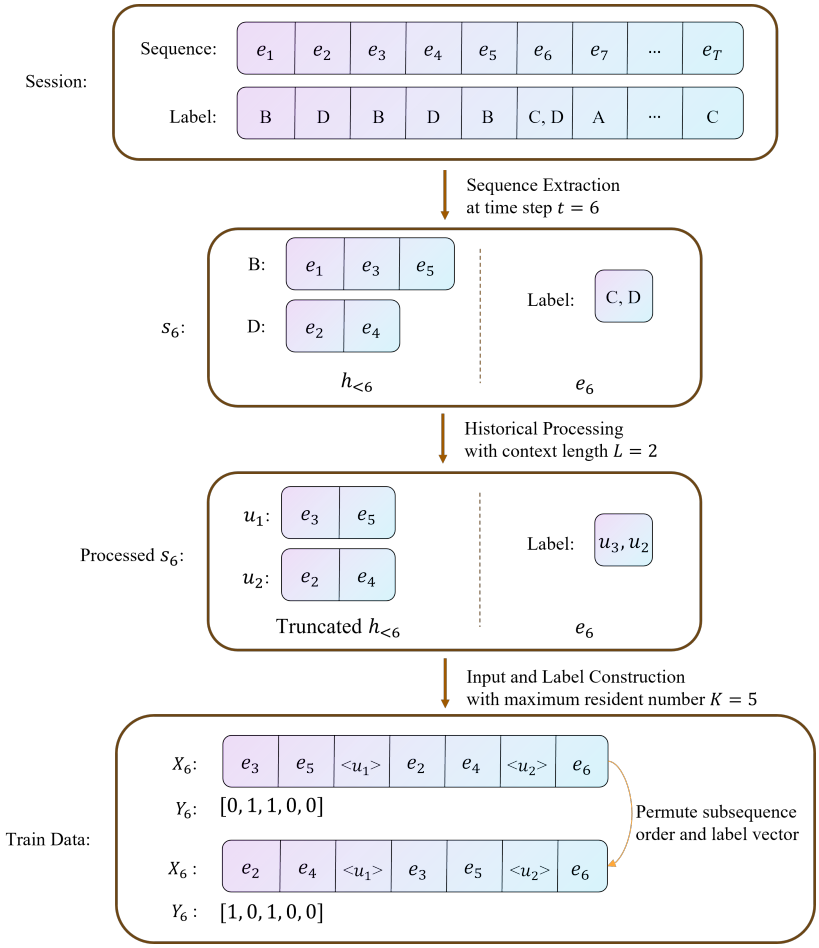



Figure 4.3 – The training data preparation process of the proposed behavior cloning method

transforms raw session logs into model-ready input sequences and multi-label targets as illustrated in Figure 4.3.

**Sequence Extraction.** We iterate through all sessions in the training set, treating each as a chronological event sequence. A sliding window procedure converts a session with events $[e_1, e_2, \ldots, e_t]$ into multiple training instances. For every time step $t$, events 1 through $t-1$ serve as historical context, and event $t$ becomes the prediction target. Each valid instance is then processed to separate resident-specific histories, truncate them, and structure the context, after which it is added to the raw training set.

**Historical Processing and Input Construction.** For each instance, historical events are grouped by resident, yielding one subsequence per resident appearing in the context. For example:

$$\mathrm{B} : [e_1, e_3, e_5], \qquad \mathrm{D} : [e_2, e_4].$$

These residents are considered existing, while the resident of the current event may be existing or new depending on prior occurrences.

All historical residents are then mapped to consecutive internal indices $\{u_1, \ldots, u_n\}$, providing a normalized identifier space used consistently across the model. This remapping also applies to the label vector so that its $i$-th position corresponds to resident $u_i$. Each resident-specific subsequence is truncated to its most recent $L$ events. Theoretically, any instances exceeding the resident limit $K$ would be discarded. In practice, however, this case never occurs as $K$ is set to exceed the maximum number of residents found in the training set.

After truncation and indexing, we construct the model input sequence by concatenating all resident subsequences in a predefined order, inserting each resident's delimiter token, and appending the current event token. Since the ordering of residents in the input carries no semantic meaning, we enforce invariance through permutation-based data augmentation. For each instance, all permutations of the historical residents are enumerated. The input sequence is reconstructed according to each permutation, and the same permutation is applied to the label vector. This avoids reliance on positional heuristics and drives the model to associate the current event with the behavioral patterns encoded in each resident's history instead of with the position of the label, resulting in permutation-equivariant behavior in the output.

**Label Construction.** The label of each instance is encoded as a length-$K$ binary vector using a multi-hot representation. Each active entry corresponds to a resident associated with the event: the $i$-th entry is set to one if the event belongs to the remapped resident $i$, and the next unused index is activated when a new resident is involved. This representation allows multiple residents to be predicted for the same event, while all unrelated positions remain zero.

**Token Indexing.** Event and resident-delimiter tokens are finally mapped to integer indices using a fixed vocabulary. This vocabulary is pre-constructed to include every distinct event category and the requisite resident delimiters, assigning a unique integer identifier to each. This process produces the indexed sequences used as model input.

## 4.3 Experimental Evaluation

### 4.3.1 Experimental Setup

To strictly validate the performance improvements of Behavior Cloning compared to the heuristic search strategy (NEP+Search), we maintain the exact experimental protocol established in Chapter 3. We utilize the same three multi-resident datasets—CASAS [62], MARBLE [22], and MuRAL [49]—and adhere to the identical training, validation, and testing splits. This consistency ensures that any observed

Table 4.1 – Comparison of Data Association performances of baseline approaches and our approach on 3 datasets. Acc and CRL are calculated using a window size $w = 16$.

| Dataset | CASAS [173] | | | | MARBLE [22] | | | | MuRAL [49] | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Metric | Acc | CRL | BLEU | RCE | Acc | CRL | BLEU | RCE | Acc | CRL | BLEU | RCE |
| All-to-One | 0.580 | 5.372 | 0.565 | 0.995 | 0.729 | 8.606 | 0.707 | 0.789 | 0.551 | 4.489 | 0.510 | 1.593 |
| sMRT [197] | 0.553 | 3.060 | 0.532 | 1.227 | 0.119 | 0.641 | 0.079 | 15.10 | 0.529 | 3.321 | 0.497 | 1.560 |
| NEP+Search | 0.682 | 4.688 | 0.652 | **0.019** | **0.825** | **8.847** | **0.812** | 0.500 | 0.674 | 4.982 | 0.755 | **0.375** |
| BC | **0.757** | **6.234** | **0.727** | 0.021 | 0.792 | 8.710 | 0.797 | **0.136** | **0.728** | **5.476** | **0.774** | 0.437 |

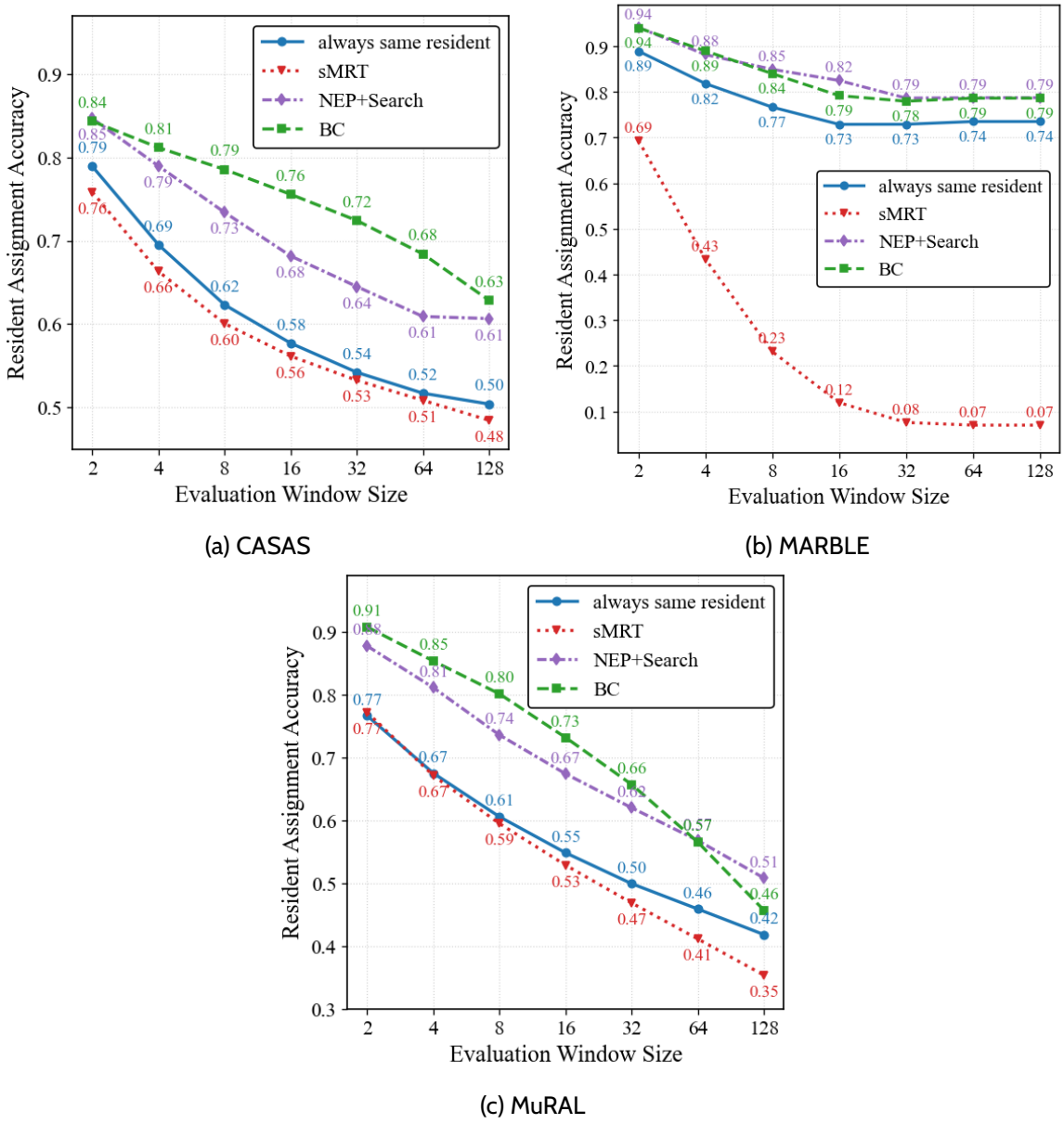
performance gains are attributable solely to the proposed learning-based policy formulation rather than variations in data distribution.

The policy network architecture utilizes a consistent Transformer encoder structure across all experiments, consisting of 3 layers with a model dimension ($d_{\text{model}}$) of 128, 8 attention heads, and a feed-forward dimension of 512. Training is performed with a batch size of 512, a learning rate of $10^{-4}$, and a dropout rate of 0.2. These model architectures and training hyperparameters were chosen based on preliminary trials on the validation set, without extensive hyperparameter tuning, to demonstrate the robustness and generalizability of the proposed method. However, to accommodate the varying complexity and density of the datasets, we adjusted the training duration and the resident history window size ($L$) for each dataset. Specifically, the MARBLE dataset was trained for 200 epochs with a history window of $L = 11$. In contrast, both the CASAS and MuRAL datasets were trained for 100 epochs, with history windows set to $L = 9$ and $L = 1$, respectively. These hyperparameters were obtained via grid search on the validation set, and the search procedure and its impact will be presented in the following sections. As for inference, we apply a fixed multi-label decision threshold of $\tau = 0.5$ across all datasets. This value is used as a standard default without further tuning, in order to avoid dataset-specific optimization and to ensure fair and reproducible evaluation.

### 4.3.2 Result Comparison

The comparison in Table 4.1 highlights the different strengths of our BC-learned policy and the previous NEP+Search approach across the three datasets. Overall, BC yields strong performance on datasets where multi-resident interactions are frequent and where sufficient training data is available, while NEP+Search benefits from environments whose structure closely matches its next-event-prediction formulation.

On CASAS, BC clearly surpasses NEP+Search in Resident Assignment Accuracy (Acc), Correct Run Length (CRL), and Bilingual Evaluation Understudy (BLEU), indicating that the learned policy captures resident-specific behavioral patterns more effectively. Although NEP+Search attains a slightly lower Resident Count Error (RCE), this advantage is small compared with BC's substantial improvements in token-level and sequence-level prediction accuracy.




(a) CASAS

(b) MARBLE

(c) MuRAL

Figure 4.4 – Resident assignment accuracy as a function of the evaluation window size ($\log_2$ scale) on the CASAS, MARBLE, and MuRAL datasets.

The results on MARBLE exhibit a different trend. MARBLE consists primarily of single-resident sessions, which naturally align with the event-by-event prediction mechanism used in NEP+Search. In this setting, the core advantage of BC—its ability to jointly reason over multiple residents—cannot be meaningfully exploited. Moreover, MARBLE is considerably smaller than the CASAS and MuRAL datasets. Since BC relies on a higher-capacity transformer network and a more expressive modeling paradigm, it benefits from larger amounts of training data; the limited scale of MARBLE therefore constrains its performance. As a result, BC performs comparably to NEP+Search on

MARBLE rather than offering notable improvements.

On MuRAL, which features richer multi-resident activity structure, BC again performs strongly, achieving the best Acc, CRL, and BLEU scores. NEP+Search attains the lowest RCE, reflecting its emphasis on global consistency, but its event-level accuracy lags behind BC. These trends indicate that BC is particularly effective when resident interactions are prominent and sufficient training data is available.

Taken together, the results show that while BC matches NEP+Search on MARBLE due to structural and data constraints, it provides clear advantages on datasets with richer multi-resident dynamics, demonstrating its effectiveness as a general-purpose policy for multi-resident data association.

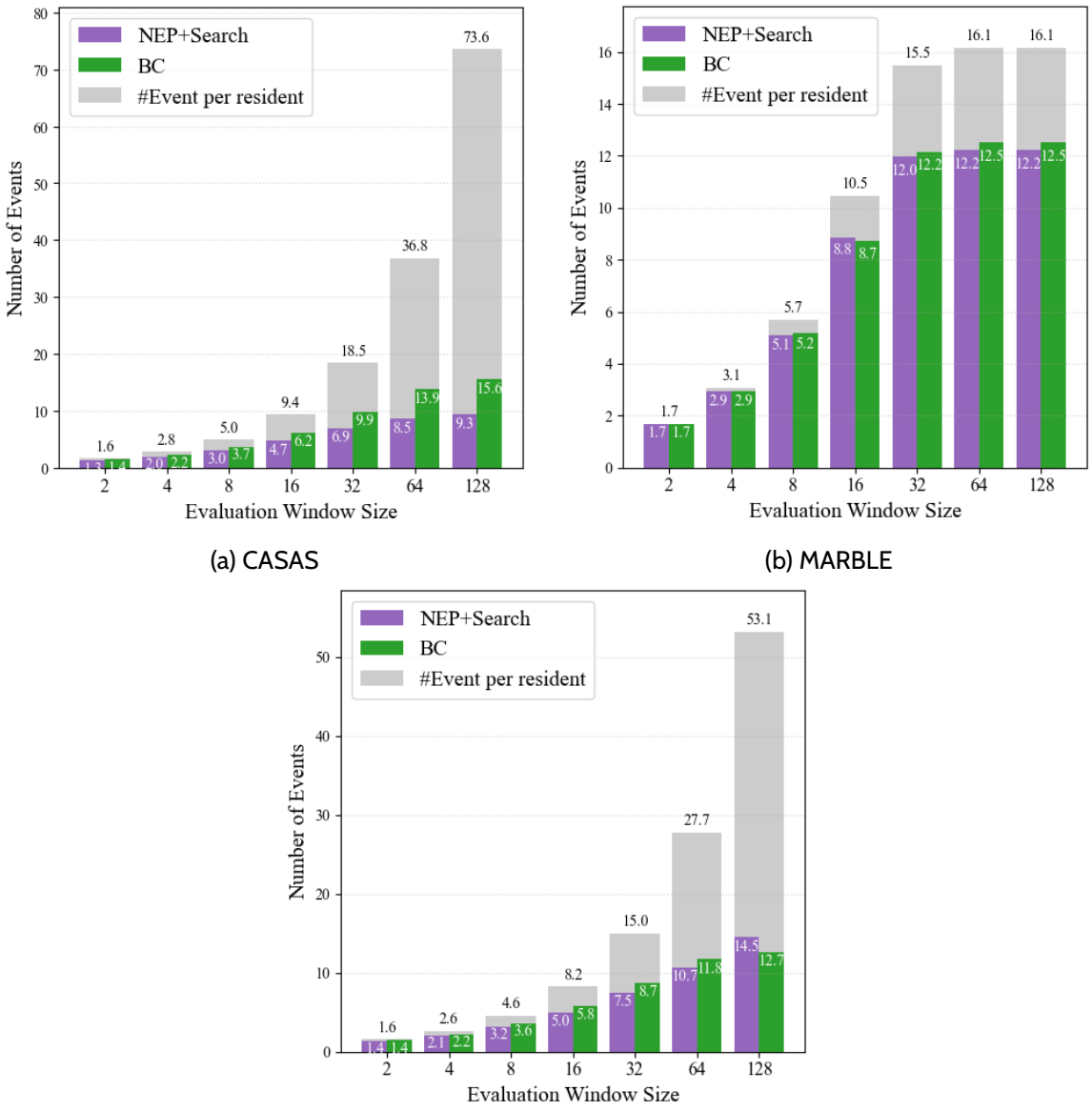
### 4.3.3 Impact of Evaluation Window Size

To further examine how error accumulation affects long-horizon assignment performance, Figure 4.4 and Figure 4.5 compare respectively the Acc and the CRL of BC and NEP+Search under different evaluation window sizes on the three datasets. The horizontal axis is plotted on a $\log_2$ scale for clearer visualization. As expected, the Acc of all methods decreases as the window size increases, since each prediction depends on the previously assigned labels and early mistakes propagate through the sequence. From the CRL perspective, although the CRL increases as the evaluation window size grows, its proportion relative to the average number of events per resident gradually decreases. This phenomenon can also be attributed to the accumulation of errors.

Across all datasets, BC exhibits higher Acc and CRL than NEP+Search, demonstrating that it provides stronger local decision-making capability. On CASAS, BC not only starts with the highest Acc but also maintains a slower decline than NEP+Search up to a window size of 64. This indicates that BC is not only more accurate in short horizons but is also relatively robust to early prediction errors. Even at the longest window size of 128, where cumulative errors inevitably appear, BC still outperforms all other methods by a clear margin in both Acc and CRL.

On MARBLE, which consists predominantly of single-resident sequences and therefore closely matches the next-event prediction assumption of NEP+Search, the two methods behave very similarly. This similarity is evident in both figures: Acc curves almost overlap, and CRL values are nearly identical across window sizes, despite the gradual increase in events per resident. Because MARBLE provides limited multi-resident ambiguity, BC cannot leverage its richer reasoning mechanism, and both methods exhibit comparable error accumulation behavior.

The MuRAL dataset presents a more nuanced pattern. As shown in Figure 4.4, BC maintains higher Acc and a slower decline than NEP+Search up to a window size of 16, demonstrating superior short-range assignment precision. This advantage is also reflected in Figure 4.5, where BC achieves longer CRL at small and medium window sizes. However, as the window size further increases and the average number




(a) CASAS

(b) MARBLE

(c) MuRAL

Figure 4.5 – Evaluation of Correct Run Length (CRL) and the average number of events per resident (gray bars) in different evaluation window sizes for CASAS, MARBLE, and MuRAL.

of events per resident grows substantially, BC's CRL advantage diminishes and its Acc begins to drop faster than that of NEP+Search, eventually falling slightly below it.

This behavior reveals a limitation of BC's state representation on MuRAL. In our configuration, each resident's effective history length is short ($L = 1$), resulting in highly focused but fragile internal representations. Once an early misassignment occurs, the resident history becomes misaligned, and the transformer encoder continues to propagate this incorrect state, leading to faster degradation in both Acc and

CRL as the sequence length increases. In contrast, NEP+Search maintains weaker temporal coupling between assignments, which results in shorter CRL overall but slower long-horizon error accumulation. The quantitative impact of history length on Acc will be further analyzed in Section 4.3.5.

Overall, the combined evidence from both figures shows that BC delivers the strongest performance in terms of local assignment accuracy and sustained correct runs, particularly in short- to medium-range evaluation horizons and in datasets with substantial multi-resident interaction. However, NEP+Search, while weaker in local prediction quality and CRL, offers more stable long-horizon behavior under extreme error accumulation scenarios.

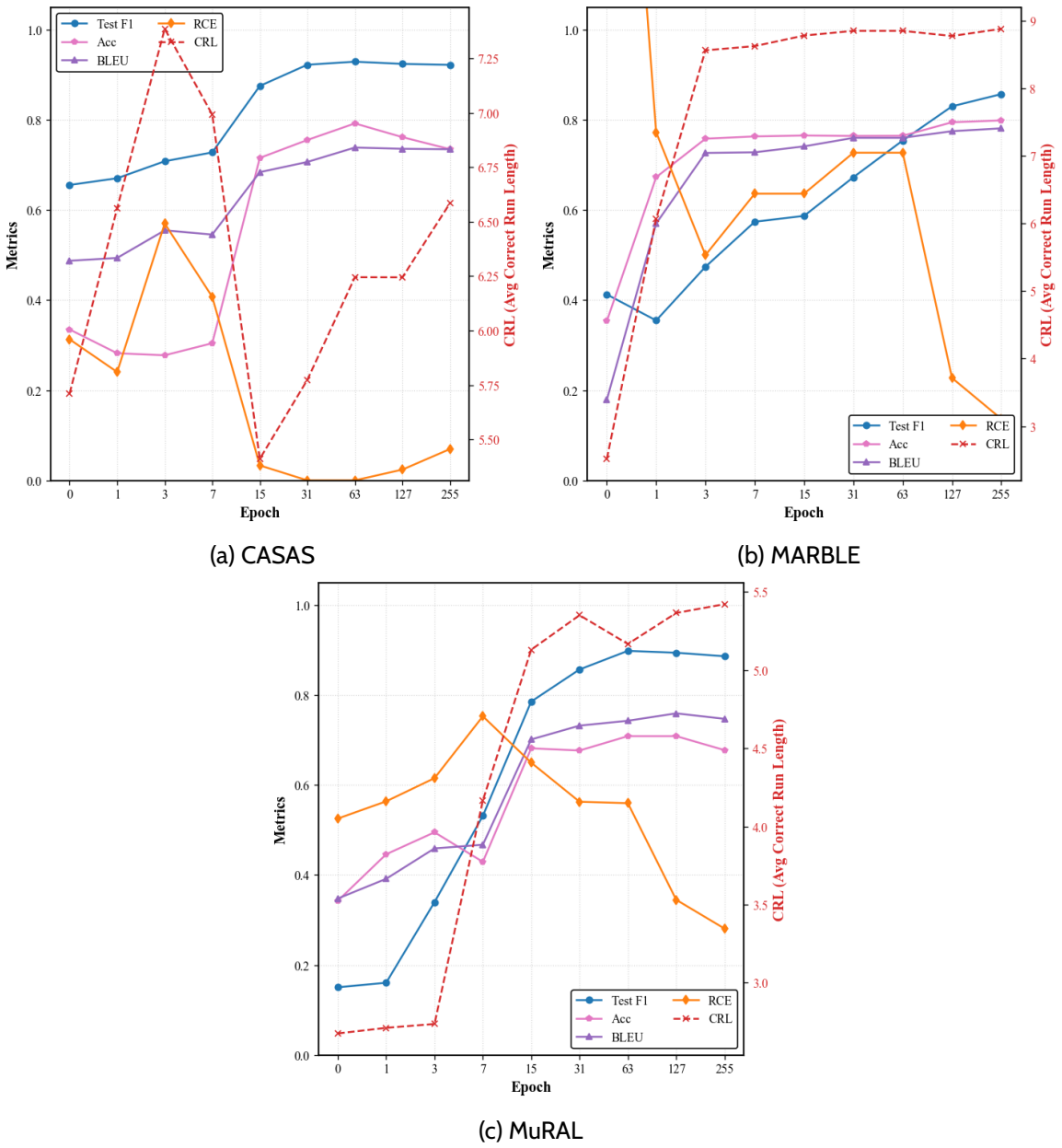
### 4.3.4 Impact of Policy Network

To understand how the learned policy evolves during training and how its performance propagates into the downstream Data Association task, we track the dynamics of the policy model across training epochs. Figure 4.6 plots the test-set F1 score of the policy model together with the Data Association evaluation metrics (Acc, BLEU, RCE, and CRL) obtained during inference for different training epochs on the three datasets.

Across all datasets, the policy network exhibits a clear and consistent improvement in its predictive ability as training progresses. The F1 score surpasses 0.8 on all datasets and approaches 0.9 on both CASAS and MuRAL. This demonstrates that the mapping from the multi-resident contextual event representation to the resident assignment decision is learnable under the current dataset scale and model capacity. Moreover, as the test F1 score increases throughout training, the corresponding Data Association metrics also show aligned improvements, indicating a tight coupling between the accuracy of the policy classifier and the quality of downstream sequence-level Data Association. This alignment validates both the modeling formulation and the supervised training paradigm.

Beyond this overall trend, each dataset reveals a characteristic progression pattern that reflects its structure. On CASAS, policy accuracy and BLEU steadily increase with training. However, both RCE and CRL exhibited a certain degree of fluctuation prior to the 15th epoch. These fluctuations stem from extreme policies generated by the policy network before convergence—such as assigning all events to a single resident—which results in simultaneously high values for both RCE and CRL.

MARBLE displays faster early convergence in terms of Acc and BLEU. This is because it primarily contains single-resident sessions, the policy faces a simpler classification scenario, allowing metrics such as Acc and BLEU to saturate early. However, CRL on MARBLE improves gradually as long with the improvement of the test F1. This implies that accurately identifying the number of resident in local necessitates a more precise policy network, which, in turn, requires more comprehensive training.




(a) CASAS

(b) MARBLE

(c) MuRAL

Figure 4.6 – Performance evolution of the policy network across training epochs on all datasets.

Overall, these results show that the performance of the policy network directly shapes the downstream Data Association behavior. The tight correspondence between improvements in F1 score and gains in Acc, BLEU, and CRL indicates that enhancing the policy's classification quality naturally translates into stronger and more coherent multi-resident assignment performance. This reinforces the central role of the policy network in our framework and confirms the effectiveness of the BC-based training strategy.

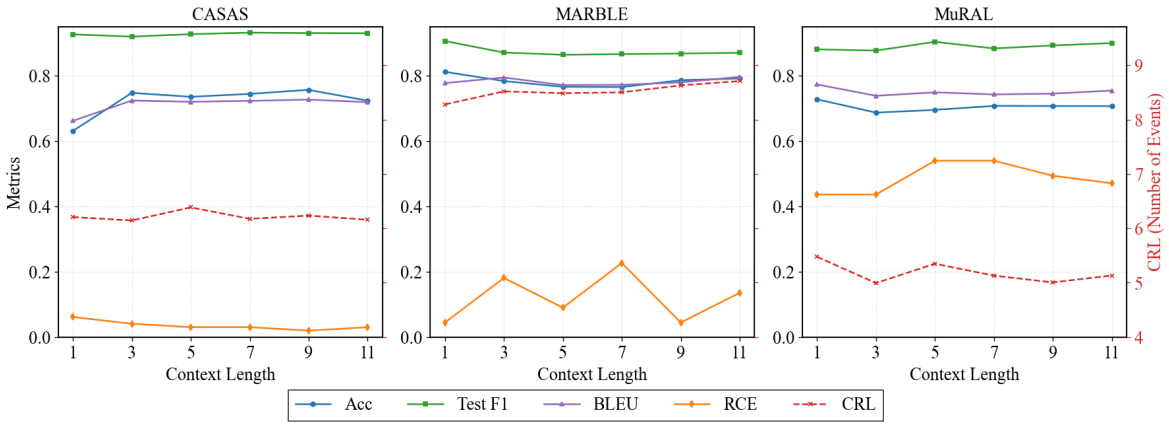


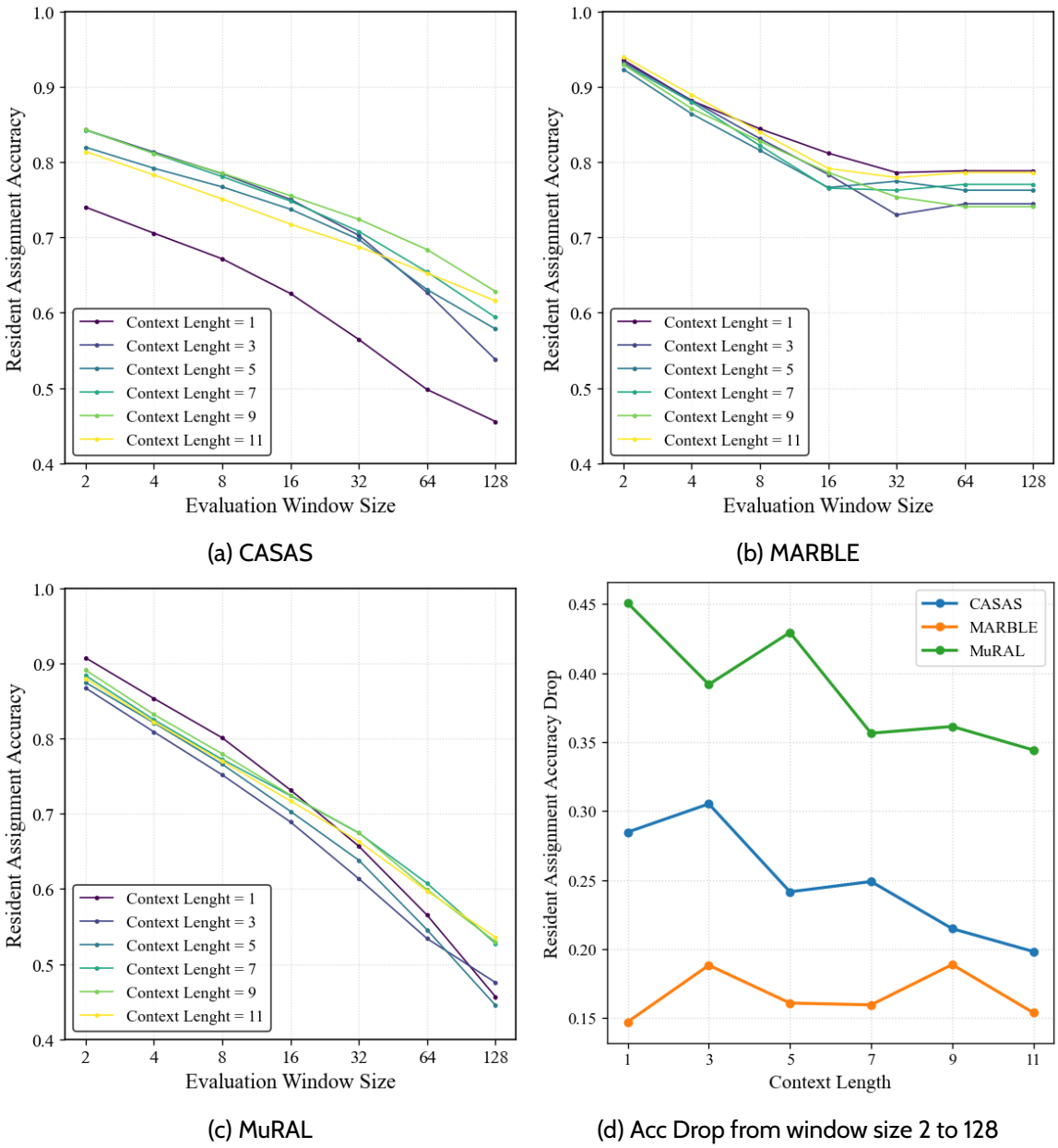

Figure 4.7 – Performance evolution of the policy network across context lengths on all datasets.

### 4.3.5 Impact of Context Length

To investigate how the amount of historical information provided to the policy network influences both learning and inference, we evaluate the model under different context length configurations. Figure 4.7 summarizes the Test F1 scores of the trained policy and the corresponding Data Association metrics (Acc, BLEU, RCE, and CRL) across context lengths ranging from 1 to 11. Overall, the Test F1 curves indicate that the policy network is relatively insensitive to context length during supervised training. CASAS shows nearly identical Test F1 across all settings, MARBLE exhibits a mild decrease with longer contexts, and MuRAL displays a slight increase. The magnitude of variation, however, remains small, suggesting that the classifier can learn the mapping from historical context to resident assignment even when only limited past information is available.

In contrast, the effect of context length becomes more pronounced during inference. The Data Association metrics reveal that the optimal context length for training the classifier does not always coincide with the context length that yields the strongest autoregressive performance. For example, MuRAL achieves the best Data Association results at context length 1 despite having its lowest Test F1 under this configuration, whereas CASAS performs substantially worse at context length 1 despite having a similar Test F1 to longer contexts. These discrepancies highlight that context length affects not only the classifier's supervised accuracy but also the stability of the resident histories that accumulate during autoregressive inference.

To further understand how context length affects robustness to long-horizon error accumulation, Figure 4.8 presents accuracy results across evaluation window sizes and summarizes the total degradation from window size 2 to 128. Across all datasets, shorter contexts generally experience steeper accuracy drops as the evaluation horizon grows. On CASAS, context length 1 suffers the largest decline, reflecting its vulnerability to compounding errors when only minimal historical information is available.




(a) CASAS

(b) MARBLE

(c) MuRAL

(d) Acc Drop from window size 2 to 128

Figure 4.8 – Data Association performance under different context lengths, with panels (a)–(c) showing accuracy across window sizes and panel (d) summarizing long-horizon accuracy degradation.

Longer contexts (7–11) provide more stable long-horizon behavior and produce the smallest degradation. MARBLE shows much smaller variation overall, consistent with its predominantly single-resident structure, which inherently limits the potential for cascading identity-switch errors. Even so, medium context lengths (5–9) still exhibit slightly greater robustness.

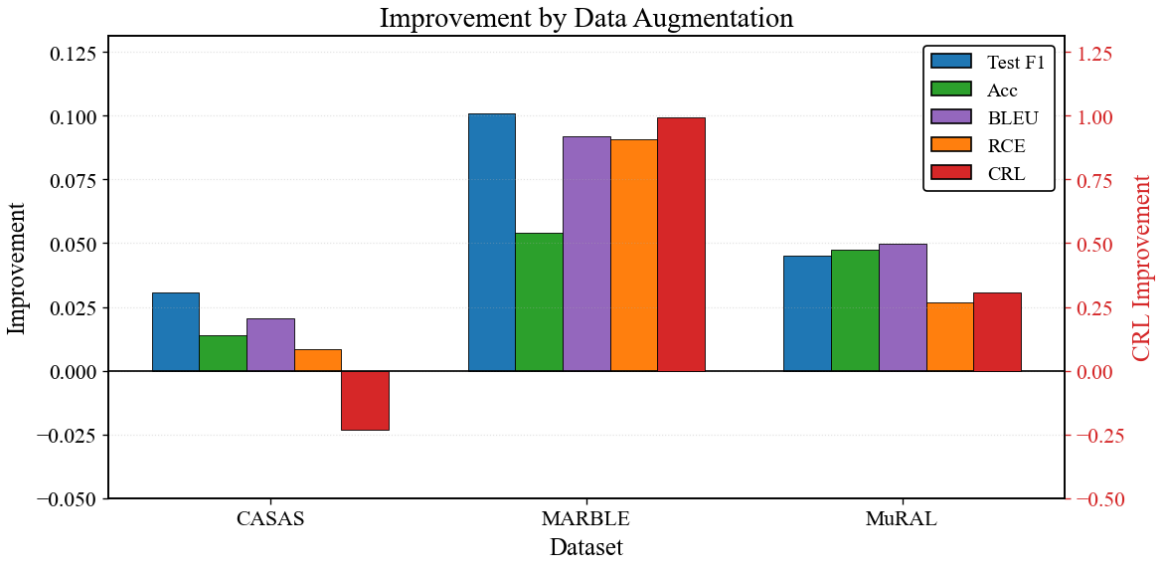



Figure 4.9 – Effect of permutation-based data augmentation on Test F1 and Data Association metrics across the three datasets.

MuRAL presents a more nuanced pattern. Although context length 1 yields the best short-window accuracy, it also experiences one of the steepest declines as the window size increases. This suggests that minimal contexts enhance local precision—likely due to reduced interference from complex multi-resident interactions—but weaken global consistency by making the model more sensitive to early mistakes. Longer contexts in MuRAL mitigate this effect by providing richer temporal structure that stabilizes the resident histories over longer sequences.

Taken together, these results demonstrate that context length plays a dual role: it has limited influence on training-time classification accuracy but significantly shapes the model's autoregressive behavior during inference. Short contexts may improve short-range precision on datasets with high resident complexity, whereas longer contexts offer greater resilience to error accumulation in long sequences. Selecting an appropriate context length therefore requires balancing dataset characteristics with the desired trade-off between local accuracy and global consistency.

### 4.3.6 Impact of Data Augmentation

To verify the effectiveness of the permutation-based data augmentation strategy, we conduct a comparative experiment evaluating the Behavior Cloning policy with and without augmentation. The results are summarized in Figure 4.9, which reports the performance improvement across all datasets under an identical number of optimization steps, with early stopping applied on the validation set to avoid overfitting. This setup ensures that the observed differences in performance originate from data augmentation rather than disparities in training duration.

Table 4.2 – Performance comparison of BC and NEP+Search in assigning events to the correct number of residents on CASAS and MuRAL.

| #Residents | Dataset | CASAS | | MuRAL | |
|---|---|---|---|---|---|
| | Method | NEP+Search | BC | NEP+Search | BC |
| Multi | Correct | 0.007 | 0.354 | 0.022 | 0.418 |
| | Over-Assigned | 0.000 | 0.000 | 0.000 | 0.010 |
| | Under-Assigned | 0.993 | 0.646 | 0.978 | 0.572 |
| Single | Over-Assigned | 0.024 | 0.081 | 0.212 | 0.028 |

The extent of improvement varies across datasets and is closely tied to their resident-distribution characteristics. On CASAS, where every session contains exactly two residents with relatively balanced activity patterns, data augmentation yields only modest gains. Because both residents appear frequently and symmetrically, permuting their identities introduces limited new supervisory signal, resulting in small but consistent improvements.

In contrast, MARBLE exhibits substantial improvements across nearly all metrics. Its sessions contain between one and four residents with highly uneven frequencies, which makes the model more susceptible to positional bias and overfitting to common resident orderings. Data augmentation effectively alleviates these issues by exposing the model to diverse permutations of resident identities, leading to notable gains in both Test F1 and Data Association performance. MuRAL, which always involves between 2 and 4 residents (predominantly 3) with moderate imbalance, shows improvements that fall between the two extremes, consistent with its intermediate level of resident heterogeneity.

These findings indicate that data augmentation is particularly beneficial when the resident identity distribution is uneven or highly variable. By promoting permutation invariance and mitigating positional biases, augmentation strengthens the generalization ability of the Behavior Cloning policy and contributes to more reliable assignment predictions during inference.

### 4.3.7 Assignments of Multi-Resident Events

In multi-resident scenarios, certain ambient sensors may be triggered by multiple residents simultaneously. For example, two residents located at the same place can jointly activate a nearby motion sensor. Among the three datasets used in this work, both CASAS and MuRAL contain events that belong to more than one resident. Specifically, 14.4% of the events in CASAS and 8.5% of the events in MuRAL are multi-resident events. We evaluate and compare whether NEP+Search and BC assign these multi-resident events to the correct number of residents, as well as whether single-resident events are incorrectly over-assigned to multiple residents. The results are summarized in Table 4.2.

From Table 4.2, we observe that BC consistently outperforms NEP+Search in assigning multi-resident events to the correct number of residents on both CASAS and MuRAL. This difference can be attributed to the fundamental modeling assumptions of the two approaches. BC jointly considers the historical events and dependencies of multiple residents when making an assignment decision, which enables it to explicitly model co-occurring activities and shared sensor triggers. As a result, BC is better equipped to recognize when a single event should be associated with more than one resident. In contrast, NEP+Search performs data association by sequentially reasoning over individual resident event sequences. At each step, it primarily evaluates the compatibility between the current event and a single resident's history, making it less sensitive to cross-resident dependencies. This design leads NEP+Search to strongly favor single-resident explanations, causing a large fraction of multi-resident events to be under-assigned in both datasets.

Importantly, BC does not exhibit a generally aggressive over-assignment behavior. For single-resident events, BC maintains a relatively low over-assignment rate, particularly on the MuRAL dataset, where it is significantly lower than that of NEP+Search. This indicates that BC is able to improve correctness on multi-resident events without substantially increasing false multi-resident assignments for single-resident events. Overall, these results suggest that explicitly modeling joint resident histories and their dependencies is crucial for accurately handling multi-resident sensor events in realistic smart home environments.

## 4.4 Conclusion

In this chapter, building upon the previously proposed MDP framework, we introduce a Behavior Cloning approach that directly leverages expert assignment annotations in the dataset to train a policy network for event assignment prediction, thereby achieving a direct alignment between decision making and data. Through theoretical analysis, we demonstrate that the binary cross-entropy loss is an appropriate training objective for our framework. We design a structured and serialized representation of state inputs and adopt a Transformer encoder based on modern architectural modules as the backbone of the policy network. To further facilitate effective training, we propose a full-permutation-based data augmentation strategy to increase the diversity of the input data, enabling the policy model to be invariant to the ordering of residents. Experimental results show that, compared with the previous NEP+Search approach, our BC method achieves significant improvements across multiple data association metrics on the CASAS and MuRAL datasets. Moreover, by jointly considering the states of all residents, BC is better than NEP+Search at identifying multi-resident events and assigning them to the correct number of individuals. In summary, the BC method proposed in this chapter further enhances the data-driven nature of the MDP modeling framework and, to a certain extent, alleviates the limitations of NEP+Search.

# CHAPTER 5

# TOWARDS LLM-POWERED POLICY FOR MULTI-RESIDENT DATA ASSOCIATION

## 5.1 Introduction

In the previous chapters, we modeled the *Data Association* problem as a Markov Decision Process (MDP), where each state represents the historical event assignments and the next sensor event to be processed. Based on this formulation, we proposed two complementary approaches: one integrates a language-model-based reward model to guide the search and decision process (Chatper 3), and the other directly employs the language model as a policy model to make event-to-resident assignment decisions (Chapter 4). Although these language-model frameworks achieve noticeable improvements over previous state-of-the-art (SOTA) methods [197, 198], several challenges remain:

— **Data Collection**: Due to the high cost of setting up experimental environments and the sensitivity of personal daily living data, collecting ambient sensor datasets is often challenging.
— **Model Generalization**: Due to varying sensor setups and activity routines, models trained on specific datasets often struggle to transfer their capabilities to different environments or configurations.

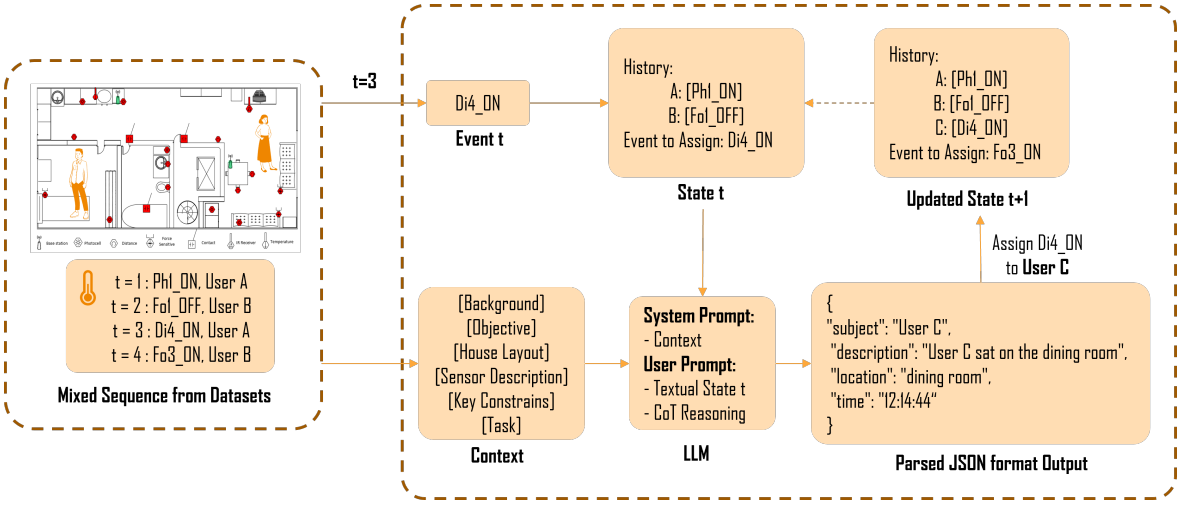



Figure 5.1 – Illustration of the proposed LLM-based multi-resident data association method

— **Context Integration**: Contextual information like sensor locations, functions, time, environment, and resident habits is crucial due to the simplicity of ambient sensor data. However, traditional deep learning methods often fail to efficiently encode this information, making Data Association and downstream Human Activity Recognition (HAR) less precise and flexible.

In recent years, significant advancements have been made in **Large Language Models (LLMs)**, with series of models such as ChatGPT [2], Gemini [183], Llama [187], Qwen [212], DeepSeek [92], etc. Compared with traditional language models, LLMs scale up the number of parameters from millions to over one hundred billion, with some models even exceeding this range [38, 52, 187]. They are trained on vast, diverse, and carefully curated text corpora, covering multiple domains, languages, and discourse styles [149, 221]. Furthermore, beyond standard pretraining objectives, LLMs are often refined through instruction tuning and Reinforcement Learning from Human Feedback (RLHF) [136, 28], which align their behavior more closely with human intentions and preferences. These advancements collectively endow LLMs with remarkable capabilities, including sophisticated natural language understanding and generation, robust commonsense reasoning, contextual and temporal awareness, logical and causal inference, as well as strong generalization across a wide range of linguistic and reasoning tasks [2, 201]. As a result, LLMs have evolved from simple next-word predictors into general-purpose reasoning agents capable of interpreting, planning, and interacting in complex real-world contexts. These advanced abilities endow LLMs with the potential to address the aforementioned challenges of traditional language models in ambient-based multi-resident Data Association: 1) LLMs' in-context learning capability [38] reduces the need for training datasets; 2) by adapting relevant prompts, LLMs can swiftly adapt to novel environments or adjust to new sensor configurations; 3) leveraging the expressiveness of natural language, LLMs can integrate the different types of contextual information; 4) LLMs can connect

related events using attention mechanisms, integrating common sense and reasoning to identify meaningful sensor event combinations. Furthermore, LLMs' generation capabilities allow them to generate grouped coherent sequences. Therefore, LLMs have the potential to separate mixed event sequences in multi-person scenarios; 5) LLMs possess the ability to explain their reasoning process, thereby enhancing the explainability of the inference.

Following the advancement of LLMs, Takeda et al. [178] first used the GPT2 model [148] for generative prediction of sensor event sequences, predicting future sensor events based on the labels of the ongoing activity and the sensor events that have already occurred. This work strengthened the association of smart-home-based HAR with language models and brought the large GPT model [147] into the scope of HAR. Leveraging LLMs' powerful in-context learning [38] and reasoning [202] abilities, Gao et al. [84] first used an LLM to perform unsupervised annotation on single-person activity samples in the ARAS dataset [16], demonstrating the potential of LLMs for unsupervised HAR. In this work, Gao et al. used sensor reading data within a 5-minute sliding window as input data. They employed a Chain-of-Thought approach [202] to instruct the LLM to analyze the functions of the activated sensors. By integrating context information on room layout, time, and the duration of sensor activation, LLM was finally instructed to choose an activity as the recognition results from nine activities selected by the authors. Although the experimental results show comparable accuracy to supervised trained models, this work is limited to nine easily distinguishable activity categories in single-person scenarios, overlooking the recognition of other more challenging categories and failing to provide prompts for reproducibility. Furthermore, using sensor readings in fixed time windows rather than sensor events as input data limits the model's ability to perceive the subject's behavior at a finer granularity. Civitarese et al. [55] used instead sensor event streams as input data, transforming windowed events into textual representations and, together with carefully designed prompts, enabling LLMs to achieve zero-shot activity classification accuracy comparable to that of traditional supervised deep learning approaches. However, these prior works are all limited to HAR tasks in single-resident scenarios, and research on applying LLMs to multi-resident data association remains largely unexplored.

The objective of this study is to leverage the advanced capabilities of LLMs to tackle the identified challenges in Multi-Resident Data Association. To achieve this, we propose **LADA** (**L**LM-powered **A**mbient-based **D**ata **A**ssociation), a framework for multi-resident data association that leverages an LLM as a zero-shot decision maker. This framework builds upon the MDP decision-making formulation proposed in Chapter 3. Instead of the supervised neural network–based policy used in Chapter 4 to assign events to residents, we adopt an off-the-shelf LLM. Through zero-shot Chain-of-Thought (CoT) [202] reasoning, the framework enables unsupervised data association. Experiments on MARBLE and MuRAL demonstrate that the unsupervised LADA outperforms the supervised Next-Event Prediction (NEP)+Search and achieves performance

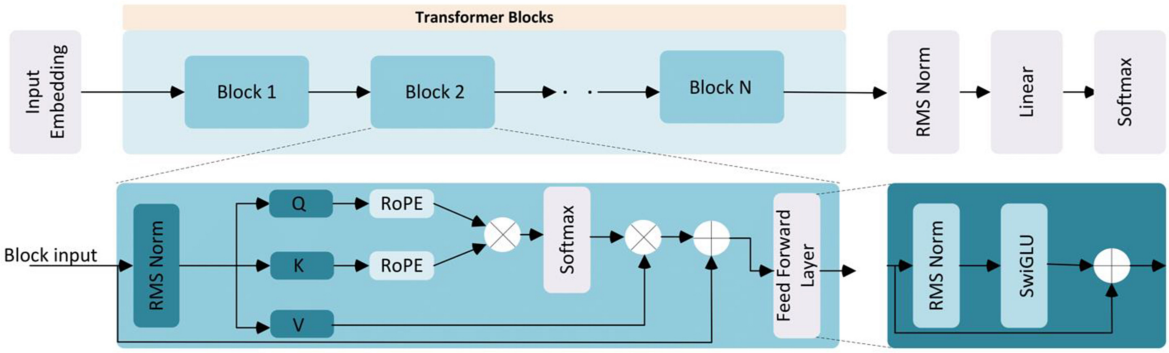



Figure 5.2 – Illustration of the overall architecture of Llama 3 [79]. Reproduced from Zafarmomen and Samadi [217].

competitive with Behavior Cloning (BC), a more advanced supervised method. These results validate the Data Association capabilities of the LADA framework and highlight the superior unsupervised generalization power enabled by LLMs.

## 5.2 Principle of Large Language Models

### 5.2.1 Mainstream Architecture

Figure 5.2 illustrates an overview of the Llama 3 architecture [79], which is representative of the standard architecture adopted by many contemporary LLMs. Specifically, after the input text is tokenized into subword units using a Byte Pair Encoding (BPE) tokenizer [166], the resulting token sequence is fed into a neural network. Although alternative architectures, such as state space models exemplified by S4 [89] and Mamba [88], have attracted substantial research attention for efficient long-sequence modeling, the dominant neural architecture underlying current LLMs is the *Transformer* [193]. The original Transformer architecture consists of an encoder–decoder structure, but in practice, many successful models employ only one of these components: encoder-based models such as BERT [73] and decoder-based models such as GPT [38].

Most modern LLMs [187, 212, 92] adopt the *decoder-only* architecture, which extends the original Transformer decoder with larger hidden dimensions, deeper network layers, and various architectural refinements to improve efficiency and stability. Specifically, the input token sequence is first transformed into embeddings through an *Input Embedding Module*, and then processed sequentially through a stack of Transformer decoder layers. Each *decoder layer* consists of two main submodules: a *Causal Multi-Head Self-Attention Module* and a *Feed-Forward Network (FFN)* module, connected via residual pathways.

Unlike the original Transformer that applies *Post-Norm* normalization (layer nor-

malization after residual addition), most recent LLMs [187, 79, 212, 92] adopt the *Pre-Norm* configuration, placing normalization before each submodule to improve training stability for very deep networks [210]. Furthermore, many LLMs [187, 79, 212, 92] replace standard Layer Normalization with *Root Mean Square Layer Normalization (RMSNorm)* [219] to reduce computational overhead while maintaining comparable performance.

In the multi-head self-attention module, most modern LLMs incorporate *Rotary Positional Encoding (RoPE)* [175] to encode relative positional information directly into the representations of each token's key and value vectors. This approach replaces the fixed or learned absolute position embeddings used in earlier Transformer variants and enables better extrapolation to longer sequences by embedding continuous rotational relationships in the embedding space.

To further optimize memory and computational efficiency—particularly during autoregressive inference where key-value (KV) caching is critical—several architectural variants have been developed. *Multi-Query Attention (MQA)* [167] reduces the size of the KV cache by sharing the key and value projections across all attention heads, while retaining independent query projections for each head. Building upon this idea, *Grouped-Query Attention (GQA)* [8] introduces a middle ground between standard multi-head attention and MQA, where a small group of heads share a single set of keys and values, achieving a favorable balance between efficiency and expressivity. More recently, *Multi Latent Attention (MLA)* was introduced in DeepSeek-V2 [72], which employs a latent-space compression mechanism to project multiple attention heads into a shared low-dimensional latent space before computing attention weights. This design significantly reduces both computation and memory usage, while maintaining high modeling fidelity, making it particularly suitable for large-scale mixture-of-experts architectures.

Following the Self-Attention Module, the FFN plays a crucial role in performing nonlinear transformations and expanding the model's representational capacity. Traditionally, the FFN consists of two linear projections with a nonlinear activation function in between, typically formulated as:

$$\mathrm{FFN}(x) = \mathrm{Linear}_2(\sigma(\mathrm{Linear}_1(x))),$$

where $\sigma(\cdot)$ is usually a ReLU or GELU activation [96].

Modern large language models have introduced several improvements to this module to enhance both expressivity and computational efficiency. A widely adopted variant is the *Swish-Gated Linear Unit (SwiGLU)* activation function [168], which replaces the standard activation with a gated linear unit using a Swish function [152]. The resulting architecture improves gradient flow and training stability while reducing overparameterization. The SwiGLU form can be expressed as:

$$\mathrm{SwiGLU}(x) = \mathrm{Linear}_3(\mathrm{Linear}_1(x) \odot \mathrm{Swish}(\mathrm{Linear}_2(x))),$$

where $\odot$ denotes element-wise multiplication. This design has been adopted in many state-of-the-art LLMs, including PaLM [52], LLaMA [187] and Qwen3 [212].

### 5.2.2 Training Paradigm

The training of LLMs typically follows a three-stage paradigm, each progressively aligning the model's behavior with human-like reasoning and instruction-following capabilities.

**(1) Pretraining.** The first stage involves large-scale unsupervised learning over diverse textual corpora using the *Next-Token Prediction (NTP)* objective, also known as *autoregressive language modeling*. Formally, the model is trained to maximize the likelihood of the next token $x_t$ conditioned on all previous tokens $x_{<t}$:

$$\mathcal{L}_{\text{NTP}} = -\sum_{t=1}^{T} \log P(x_t \mid x_{<t}; \theta).$$

This objective encourages the model to capture syntactic, semantic, and contextual relationships within natural language. The data used for pretraining are typically collected from web-scale corpora, books, code repositories, and high-quality curated text sources [149, 38, 52, 187]. During this stage, the model learns a broad world knowledge and linguistic representation but remains task-agnostic.

**(2) Instruction tuning.** In the second stage, the pretrained model is fine-tuned on datasets that consist of human-written instruction–response pairs [200, 136]. This process—often referred to as *instruction tuning*—teaches the model to better follow explicit user intents expressed in natural language. It reshapes the model's latent representations to prefer aligned, concise, and instruction-compliant outputs. The resulting model exhibits improved zero-shot and few-shot generalization across a wide variety of natural language tasks, even without further gradient updates.

**(3) Reinforcement Learning from Human Feedback (RLHF).** The third stage, RLHF, further aligns the model's responses with human preferences and value judgments [53, 174, 136, 28]. This process consists of three components: (i) generating multiple candidate responses for the same prompt using the instruction-tuned model, (ii) training a *reward model* that predicts human preference scores for these responses, and (iii) optimizing the model via reinforcement learning, typically using the Proximal Policy Optimization (PPO) algorithm [164]. RLHF helps reduce toxic, irrelevant, or untruthful outputs, enabling models to produce more helpful, honest, and harmless responses.

Overall, this three-stage training pipeline—comprising large-scale pretraining, supervised instruction tuning, and reinforcement learning with human feedback—forms the core foundation of today's state-of-the-art LLMs such as GPT-4 [2], PaLM 2 [20], and LLaMA 3 [187]. It effectively bridges the gap between raw language modeling and human-aligned, context-aware reasoning behavior.

### 5.2.3 Chain of Thought Prompting

Chain-of-Thought (CoT) prompting is a technique that significantly enhances the reasoning capabilities of LLMs by encouraging them to generate intermediate reasoning steps before producing the final answer. Unlike standard prompting, which asks the model to map the input directly to the output (i.e., $x \rightarrow y$), CoT introduces a sequence of rationale steps $z_1, z_2, \ldots, z_n$ to bridge the gap between the input and the final prediction (i.e., $x \rightarrow z \rightarrow y$).

This paradigm was first formalized by Wei et al. [202], who demonstrated that providing a few exemples of reasoning chains (Few-Shot CoT) enables models to solve complex arithmetic, commonsense, and symbolic reasoning tasks that were previously unsolvable by standard prompting methods.

Furthermore, Kojima et al. [108] discovered that LLMs are "Zero-Shot Reasoners." By simply appending the instruction “Let’s think step by step” to the input prompt, the model can autonomously generate reasoning chains without the need for task-specific demonstrations.

Formally, given an input $x$, standard prompting maximizes the probability $P(y|x)$, whereas CoT aims to maximize the joint probability of the reasoning chain $z$ and the answer $y$:

$$P(y|x) = \sum_{z} P(y, z|x) = \sum_{z} P(y|z, x)P(z|x) \tag{5.1}$$

This decomposition allows the model to allocate more computation to the logical progression of the problem, thereby improving interpretability and accuracy.

### 5.2.4 LLM as an Agent

An LLM can be conceptually viewed as an *agent* that interacts with its environment through sequential decision-making. During text generation, at each decoding step $t$, the model observes a state $s_t$—which corresponds to the context of all previously generated tokens—and performs an action $a_t$ by selecting the next token according to its learned policy $\pi_\theta(a_t \mid s_t)$. The environment then deterministically transitions to the next state $s_{t+1} = s_t \oplus a_t$, where $\oplus$ denotes token concatenation. This process mirrors an MDP, where the model’s objective is to maximize an implicit reward, such as linguistic coherence, factual correctness, or alignment with human preferences (as in RLHF).

Recent research has increasingly adopted this *LLM-as-an-agent* perspective [214, 140, 169, 50]. Within this view, the LLM acts as a policy that generates actions—such as reasoning steps, tool invocations, or dialogue responses—conditioned on the current context and inferred goals. For example, the ReAct framework [214] integrates reasoning and acting by allowing the model to interleave thought and action tokens,

while the Generative Agents framework [140] demonstrates how LLMs can simulate autonomous human-like behaviors in interactive environments by maintaining memory and reflection. Similarly, Reflexion [169] introduces verbal reinforcement learning, enabling an LLM to refine its own policy based on feedback, and the recent survey by Cheng et al. [50] provides a unified overview of this emerging paradigm where LLMs operate as reasoning, planning, and decision-making agents.

Building on this interpretation, our work extends the notion of "LLM as an agent" from linguistic environments to sensor-driven physical environments. In our MDP-based formulation of multi-resident sensor data association introduced in Chapter 3, the observed environment consists of sequences of sensor events rather than words. Each incoming event represents a new observation $o_t$, and the agent (the language model) must decide which resident triggered it—analogous to predicting the next token in a text sequence. The historical assignments of residents to past events constitute the state $s_t$, while the assignment action $a_t$ corresponds to selecting an existing resident or creating a new one.

Under this analogy, the LLM serves as a policy model that maps contextual histories of sensor observations to assignment decisions. Its intrinsic capability for contextual reasoning and sequential prediction allows it to naturally fit within the MDP framework, enabling it to act as a semantic, reasoning-driven agent for data association in complex multi-resident smart home environments.

## 5.3 Methodology Formalisation

Building upon the MDP formulation in previous chapters, we redefine the policy $\pi$ from a numerical mapping to a generative process. We denote this policy as $\pi_{\text{LLM}}$, which leverages an LLM to map the current state $s_t$ and a fixed contextual prior $c$ to an action $a_t$.

**Context Definition** We define a time-invariant contextual variable $c$ that encapsulates the static semantic knowledge of the smart-home environment. It is formalized as a tuple:

$$c = (c_{\text{env}}, c_{\text{res}}, c_{\text{task}}), \tag{5.2}$$

where $c_{\text{env}}$ encodes the spatial layout and sensor semantics, $c_{\text{res}}$ summarizes the residents' profiles and behavioral habits, and $c_{\text{task}}$ specifies the system objective (e.g., "Identify the resident triggering the sensor"). This context $c$ remains fixed throughout the decision sequence $\{e_t\}_{t=1}^{T}$ and serves as a *system prompt* that conditions the generative distribution of the LLM.

Importantly, this design leverages the context-conditioned reasoning capability of language models. By embedding structured environmental and behavioral priors into the prompt, the LLM can align its decision logic with the specific descriptions

provided in $c$, without additional gradient updates. Thus, the context $c$ acts as a static, human-readable prior that grounds the policy behavior dynamically through semantic inference.

**LLM-based Policy Definition** At each time step $t$, the system observes a state $s_t = (h_{<t}, e_t)$, where $h_{<t}$ is the history of past assignments and $e_t$ is the current sensor event. Unlike traditional policies that output a probability distribution over class labels, $\pi_{\text{LLM}}$ generates a natural language sequence $\mathbf{y}_t$. Formally, we define a serialization function $\mathcal{T}(\cdot)$ that converts the structured state and context into a textual prompt. The policy execution is defined as:

$$\mathbf{y}_t \sim \text{LLM}(\mathbf{y}_t \mid \mathcal{T}(s_t, c)). \tag{5.3}$$

The generated sequence $\mathbf{y}_t$ contains three components: (1) an CoT reasoning path $r_t$, (2) a descriptive summary of the behavior, and (3) the final resident assignment. The final MDP action $a_t$ (i.e., the resident set $\widehat{\mathcal{U}}_t$) is then obtained via a deterministic parsing function $f_{\text{parse}}$:

$$a_t = f_{\text{parse}}(\mathbf{y}_t), \quad \text{where } \widehat{\mathcal{U}}_t = a_t(e_t). \tag{5.4}$$

**State Update** Upon executing action $a_t$, the system deterministically transitions to the next state $s_{t+1}$:

$$s_{t+1} = (h_{<t+1}, e_{t+1}), \tag{5.5}$$

where the history $h_{<t+1}$ is updated by appending the current event $e_t$ and the assignment $\widehat{\mathcal{U}}_t$. Notably, LLM's open-vocabulary capability allows for dynamic label creation: if the model predicts a new resident label $u_{\text{new}}$, a new profile entry is automatically initialized in $h_{<t+1}$.

**Implicit Reward as Likelihood Maximization** Unlike the explicit reward maximization in Chapter 3, the LLM-based policy is driven by its pre-trained objective of next-token prediction. The model selects the action sequence that maximizes the conditional likelihood given the context and state. We interpret the log-probability of the generated answer as an *implicit reward* signal, $R_{\text{implicit}}$. For a generated sequence $\mathbf{y}_t$ consisting of $L$ tokens $(w_1, \ldots, w_L)$, this is formalized as:

$$R_{\text{implicit}}(s_t, a_t \mid c) \propto \log P_{\text{LLM}}(\mathbf{y}_t \mid s_t, c) = \sum_{i=1}^{L} \log P(w_i \mid w_{<i}, \mathcal{T}(s_t, c)). \tag{5.6}$$

Under this formulation, the policy optimizes its behavior by generating decisions that are semantically coherent and contextually plausible with respect to the prompt $c$. The conventional maximization of expected return is thus replaced by the maximization of semantic consistency within the language model's probability landscape.

**Iterative Decision Process** The execution of $\pi_{\text{LLM}}$ over the event stream yields a trajectory of assignments:

$$M_{\pi_{\text{LLM}}} : \{e_1, \ldots, e_T\} \xrightarrow{c} \{\widehat{\mathcal{U}}_1, \ldots, \widehat{\mathcal{U}}_T\}. \tag{5.7}$$

This formulation unifies the decision process: the evolving state $s_t$ maintains the temporal continuity of residents' locations, while the static context $c$ ensures global consistency through grounded reasoning.

**Discussion** This formulation treats the language model as a context-conditioned agent that directly performs the assignment decision at each step. The MDP structure of states, actions, and deterministic transitions remains intact, while the static context $c$ functions as a *system-level prior* that activates the LLM's context reasoning capability. Through this mechanism, the model dynamically learns task-relevant associations within a single inference session, without additional training, enabling semantic, environment-aware, and interpretable decisions across the event stream.

## 5.4 System Implementation

To operationalize the proposed MDP formulation and the LLM-based policy framework, we implement a modular system that simulates the interaction between the environment, the agent, and the evolving state of multiple residents. The system consists of three main components: `Environment`, `StateManager`, and `LADAAgent`, each responsible for a distinct but interdependent part of the overall reasoning and decision process.

### 5.4.1 Environment Module

The `Environment` module serves as the interface through which the agent receives real-time observations from the smart home setting. In our implementation, it sequentially feeds the agent with events from datasets, thereby simulating a continuous perception process analogous to how a real-world system would observe sensor activations over time. Each event is represented as a tuple $[t, e, u, a]$, denoting the event time, sensor observation, ground-truth user, and activity label.

At each time step, the `Environment` module provides a structured textual observation that includes the event timestamp, sensor identity, action type, and corresponding spatial location, for example:

```
Current sensor event observation:
{
"time": "10:54:03",
"event": "P1 OFF",
"sensor": "P1",
```

```
"action": "OFF",
"location": "dining room"
}
```

The module provides a stepwise interface similar to that of a reinforcement learning environment, where each call to the `step()` function returns the next observation in chronological order. Each observation is then passed to the language-model-based agent for reasoning and resident assignment. This sequential design ensures that the data association process unfolds in temporal order, closely mirroring the natural evolution of sensor events and activities within the smart home environment.

### 5.4.2 State Manager

The `StateManager` module maintains the dynamic state of all residents who have been observed so far, serving as the system's internal memory between consecutive decisions. Instead of initializing a fixed number of residents at the beginning, new residents are created on demand as the system encounters previously unseen identities. For each resident, the module stores their most recent behavior description, current location, and a bounded history of past sensor-triggered events, forming a compact but expressive state representation.

At any given time step $t$, the state $s_t$ can be serialized into human-readable text or structured JSON for direct inclusion in the LLM prompt. The textual summary presents each resident's recent behavior in chronological order, for example:

```
User_1 (latest 5 behaviors):
[10:51:00] R7 ON: User opened the fridge door
[10:51:05] R7 OFF: User closed the fridge door
[10:51:24] R2 ON: User opened the cutlery drawer
[10:51:32] R2 OFF: User closed the cutlery drawer
[10:51:44] P1 ON: User sat down on the dining room chair (P1)

User_2 (latest 1 behavior):
[10:52:35] P4 ON: User sat down on the dining room chair (P4)
```

This structured yet interpretable form enables the language-model agent to reason over both temporal and spatial continuity in user behaviors. By providing the LLM with explicit, context-rich state descriptions, the system allows in-context reasoning about where each resident currently is, what they are doing, and whether a newly observed event could plausibly belong to them. The bounded history length also ensures computational efficiency and maintains the prompt within the token limits of the underlying model.

### 5.4.3 Language Model Agent

**LLM-based Policy Overview** The `LADAAgent` module implements the decision-making policy by leveraging an LLM as a context-conditioned reasoning agent. Fun-

damentally, an LLM predicts the next token conditioned on its prior context, and the quality of this conditioning directly determines the accuracy and coherence of subsequent predictions. In our system, this autoregressive mechanism is harnessed to perform event-to-resident assignments, where the LLM's conditional prediction acts as an implicit reward maximization process. To fully exploit its context-reasoning ability, we design a structured prompting scheme that combines static system-level knowledge with dynamic user- and event-specific reasoning instructions.

**System Prompt Design** The first layer of conditioning is the *system prompt*, which provides static background knowledge that remains fixed across all inference steps. It defines the environment in which reasoning takes place, including: (1) the general background and objective of the task; (2) the spatial layout of the smart home and inter-room connectivity; (3) the mapping between sensor identifiers and physical objects; and (4) key behavioral and physical constraints that govern user mobility and spatial exclusivity. For instance, the system prompt encodes facts such as: a user cannot be present on two pressure mats simultaneously, only one person can occupy the bathroom at a time, and transitions between distant rooms must respect temporal feasibility. This structured global context serves as a static prior that grounds the model's reasoning in the spatial, semantic, and physical realities of the environment. Taking the MARBLE dataset as an example, the final system prompt use in this study is shown as following:

**System Prompt for MARBLE dataset used in our proposed LADA method**

**[Background]**
You are in a smart home where some sensors are installed. There are at most four people living in the house. They engage in daily activities at home. Their actions in the house can trigger the sensors and be recorded as sensor events. Patterns in sensor event sequences help you infer their movement and actions.

**[Objective]**
Given the layout of the house and the description of the sensors, you need to analyse step by step the input sensor event sequences to determine their actions and tracks with the time.

**[House Layout]**
Start at the Hall (bottom-left). The Hall has two openings only: north into the Medicine Area, and east into the Kitchen. There is no direct Hall access to the Living Room, Dining Room, or Office.
From the Hall → Medicine Area → Living Room (east from Medicine Area) → Office (open passage to the east).
From the Hall → Kitchen → Dining Room (fully open to the east).
Overall arrangement by bands:
top row: Medicine Area (left), Living Room (center), Office (right);
bottom row: Hall (left), Kitchen (center), Dining Room (right). No other direct

connections exist.
Each room contains specific facilities and sensors. The Hall serves as the entrance and passage area, with no sensors. The Medicine Area includes a medicine cabinet equipped with a medicine magnetic sensor (R6) to detect door openings. The Living Room contains 4 couches, each with a pressure mat (P3, P7, P8, P9). Face to the couches, there is a television connected to a smart plug (E2). Next to the living room, there is an office with a dest and an office chair (P2). The Kitchen, located in the center, includes several sensors: a cooker smart plug (E1) for the stove, a magnetic sensor (R5) for the pot cabinet, a magnetic sensor (R7) for the refrigerator door, a magnetic sensor (R1) for the pantry, and a magnetic sensor (R2) for the cutlery drawer. The Dining Room features a dining table with four dining chairs (P1, P4, P5, P6) and is designed as an open dining space connected to the Kitchen.

**[Sensor Description]**
SENSOR EXPLANATIONS (In column order)

| **Sensor ID** | **Sensor Type** | **Description** |
|---|---|---|
| R1 | Magnetic | pantry |
| R2 | Magnetic | cutlery drawer |
| R5 | Magnetic | pots drawer |
| R6 | Magnetic | medicines cabinet |
| R7 | Magnetic | fridge |
| E1 | Smart Plug | stove |
| E2 | Smart Plug | television |
| P1 | Pressure Mat | dining room chair |
| P2 | Pressure Mat | office chair |
| P3 | Pressure Mat | living room couch |
| P4 | Pressure Mat | dining room chair |
| P5 | Pressure Mat | dining room chair |
| P6 | Pressure Mat | dining room chair |
| P7 | Pressure Mat | living room couch |
| P8 | Pressure Mat | living room couch |
| P9 | Pressure Mat | living room couch |

**[Key Constraints]**
When processing each sensor event step by step, you must carefully consider location constraints:
- A user cannot be on two different pressure mats (P1, P2, P3, ..., P9) simultaneously
- The bathroom typically accommodates only one person at a time
- Always check each user's recent activities to determine their current location status
- Consider time sequence and movement feasibility when assigning events
- If a user is already at an exclusive location (pressure mat, bathroom, etc.), they cannot be at a different exclusive location simultaneously
- The pet can also trigger sensors, but less frequently (consider this only when both users are occupied)

**[Task]**

For each sensor event, you will receive:
- The current event observation (time, event, sensor, action, location)
- The list of observed users with their recent activity history

Your task is to:
1. Analyze each observed user's current state based on their activity history
2. Determine if the current event can be assigned to an existing user or requires a new user
3. Provide a natural language description of the activity
4. Output the result in JSON format with fields: subject, description, location, time

**User Prompt and CoT Reasoning Process** At each time step, a *user prompt* is dynamically constructed by combining: (1) the current sensor event observation from the `Environment`; and (2) the latest multi-resident state summary from the `StateManager`. The prompt then embeds an explicit block of CoT reasoning instructions that guide the LLM to conduct structured, interpretable analysis before making a decision.

This reasoning process unfolds in two stages. **Step 1: Analysis and Reasoning.** The model is required to first determine the current state of each user by chronologically reviewing their recent activities, identifying any devices that require continuous physical presence (e.g., pressure mats, chairs, or bathroom sensors), and checking whether corresponding deactivation events (OFF, closing) have occurred. Based on this, the model infers whether each user is *occupied*, *free*, or *recently moved*. It then performs a constraint verification step, checking if the current event involves a presence-required device, and whether each user can realistically trigger it given their current position and the elapsed time. This instruction design explicitly activates the model's CoT reasoning capability, encouraging it to articulate intermediate logic such as "User_1 is seated and cannot reach P4," before concluding the assignment.

**Step 2: Decision and Output.** After the analysis, the model summarizes its reasoning and decides whether the event should be assigned to an existing user or a newly created one. The final decision is expressed in a structured JSON format that includes the fields `subject`, `description`, `location`, and `time`. This standardized schema enforces output consistency and interpretability, making it straightforward for the system to parse and update internal states automatically. Taking the MARBLE dataset as an example, the final user prompt used in this study is shown as following:

**User Prompt for MARBLE dataset used in our proposed LADA method**

Observed users:
— User_1 (latest 1 activities):
[12:14:44] P4 ON: User_1 sat on the dining room chair associated with pressure mat P4.
You can assign this event to:

1. An existing user (from the list above)
2. A new user (User_2) if this event belongs to someone not yet observed

Current sensor event observation:

```
{
  "time": "12:14:44",
  "event": "P1 ON",
  "sensor": "P1",
  "action": "ON",
  "location": "dining room"
}
```

Please follow these steps to analyze this event:

**Step 1: Analysis and Reasoning**
**First, determine each user's current state:**
For EACH observed user, you MUST:

1. Go through their recent activities chronologically (from oldest to newest)
2. Identify any device/location that requires continuous physical presence (seating devices, bathroom, etc.)
3. For each such device, check if there's a corresponding deactivation/leaving event (OFF, closing, etc.) AFTER the activation event
4. List the user's CURRENT state:
   - If they have an activation event (ON, opening, etc.) for a device that requires presence, AND no deactivation event after it → User is CURRENTLY at that device
   - If they have a deactivation event after an activation → User has LEFT that device
   - If no device requiring presence → User is mobile/free

**Example of state determination:**

- User has: [10:51:00] Device_A ON, [10:51:30] Device_B ON, [10:52:00] Device_A OFF
- Current state: User is at Device_B (because Device_A was left, but Device_B ON has no corresponding OFF)

**Then, check constraints for the current event:**

1. **Identify if the current event involves a device that requires physical presence**
   - Seating devices (chairs, couches, beds, pressure mats, etc.) - require sitting/being on them
   - Bathroom/toilet - requires being inside
   - Any device that requires direct physical contact or continuous presence
2. **For each existing user, check if they can be at the current event location:**
   - If the current event is from a device that requires physical presence:

    * Check if the user is CURRENTLY at a DIFFERENT device that also requires physical presence
    * If YES → User CANNOT trigger this event (they are still at the other device)
    * If NO → User can potentially trigger this event
  — If the current event does NOT require continuous presence (door, drawer, appliance, etc.):
    * User can trigger it if they could have moved to that location (consider time and distance)

3. **Assignment logic:**
  — If an existing user is NOT currently at a conflicting device (based on their current state you determined above), and they could reasonably be at the current event location → assign to them
  — If an existing user IS currently at a conflicting device → assign to a DIFFERENT user or create a new user (User_2)
  — Always verify by checking the complete activity sequence, not just the last event

**Key principles:**

— Always consider whether a device/location requires continuous physical presence
— Check the complete sequence of events for each user, not just isolated events
— Physical constraints apply to locations that require presence, not to momentary actions or activities that allow mobility
— Time and movement feasibility matter: consider if a user could realistically move from their last known location to the current event location

**Step 2: Output Format**
After your analysis, provide the output in JSON format with the following fields:

— subject: must be one of the users mentioned above (e.g., User_1, User_2, etc.)
— description: natural language description of the user's activity
— location: the location of this event
— time: the time of this event

Please structure your response as follows:

1. First provide your analysis and reasoning
2. Then provide the JSON output

Example format:
Analysis:
**Current state of each user:**

— User_1: [Explicitly state their current state based on activity history - are they at a device requiring presence? Which device?]
— User_2: [Explicitly state their current state]

**Current event analysis:**

— Event type: [Does this event require continuous physical presence?]
— Constraint check: [For each user, can they be at this location given their current state?]
— Assignment reasoning: [Why this user can or cannot trigger this event]

[Your final reasoning about which user this event belongs to and why]
JSON Output:

```
{
  "subject": "User_X",
  "description": "...",
  "location": "...",
  "time": "..."
}
```

**Output Control and Robustness** Using an JSON parser, the system parses the LLM's JSON output and performs validity checks before integrating the result into the state representation. If a parsing failure occurs, a retry mechanism is triggered to regenerate the output up to a fixed number of attempts. This approach enhances robustness against ill-formed responses and ensures reliable operation in long sequential reasoning sessions. Overall, the combination of a globally informative system prompt, a dynamically updated user prompt, explicit reasoning instructions, and structured output control transforms the LLM from a generic text generator into a grounded, interpretable reasoning policy. Through this layered design, the `LADAAgent` produces temporally coherent, physically feasible, and semantically consistent event-to-resident assignments across sequential smart home observations.

### 5.4.4 Module Integration and Evolution

These three modules together form a closed-loop decision pipeline. The `Environment` supplies the temporal sequence of sensor observations, the `LADAAgent` interprets each observation and assigns it to a resident using contextual reasoning, and the `StateManager` updates and summarizes the evolving multi-resident state for subsequent steps. Over time, this iterative process allows the system to maintain coherent per-resident activity trajectories and adapt dynamically to the emergence of new residents or changes in behavior patterns. Furthermore, the modular design facilitates extensions such as alternative reasoning prompts, probabilistic post-processing, or hybrid reward-guided search strategies, enabling continuous improvement and flexible deployment of the LLM-based data association framework.

## 5.5 Experiments

### 5.5.1 Experimental Setup

We evaluate our system using multiple large language models under a unified experimental configuration. The default model is `GPT-4o`, which serves as the base policy for all primary experiments. The model operates in greedy decoding mode with a sampling temperature of 0.0 to ensure maximum reproducibility. Each event is processed sequentially, and a maximum of three automatic retries is allowed in cases of JSON parsing failure, as described in Section 5.4.

For closed-source GPT models, inference is performed via the LiteLLM API due to the lack of local deployment support. For open-source comparison, we additionally deploy several public language models (*e.g.*, `Qwen3-32B`) using the `vLLM` inference engine in parallel mode on two NVIDIA A100 (80 GB) GPUs. All models follow the same prompting structure and reasoning instructions to ensure a fair comparison.

Experiments are conducted on the MARBLE [22] and MuRAL [49] datasets, both of which provide richly annotated multi-resident sensor streams with semantically interpretable device and location descriptions. We exclude the CASAS [60] dataset because it contains a large number of ceiling-mounted motion sensors that are evenly distributed throughout the house, lacking clear spatial localization and semantic grounding. Such sensor signals cannot be described linguistically in a way that allows the language model to reason about physical presence or movement constraints. Each evaluated session contains temporally ordered event sequences, where the system receives events one by one and predicts both the responsible resident and a natural-language description of the corresponding activity.

### 5.5.2 Qualitative Results

To better illustrate the reasoning process and behavior of our model, we present two qualitative result examples of GPT-4o from the MARBLE dataset in Tables 6.1 and 5.2. These examples show how the proposed method performs sensor-event association in both two-resident and four-resident scenarios, highlighting its strengths and typical failure cases.

**Two-resident scenario.** Table 6.1 displays a sequence of 33 sensor events from the *A2a_instance2* session, where two residents (A and B) conduct overlapping activities in a shared kitchen–dining area. In the first 19 events, the model attributes all activations to resident A, even though the ground truth includes both A and B interacting with appliances such as the fridge (R7), pantry (R1), and stove (E1). This is due to the absence of clear mutual exclusivity among these events—their spatial proximity and temporal continuity make them appear as if they were performed by a single person.

Table 5.1 – Qualitative example of 33 sequential sensor events from the session A2a_instance2 of MARBLE dataset. Each row color indicates the prediction outcome: red represents incorrect predictions, green indicates correctly predicted events for resident A, and blue denotes correctly predicted events for resident B.

| Step | Time | Event | Facility | Predicted ID | GT ID | Correct |
|---|---|---|---|---|---|---|
| 1 | 16:10:04 | R7 ON | Fridge Door | A | A | ✓ |
| 2 | 16:10:11 | R7 OFF | Fridge Door | A | A | ✓ |
| 3 | 16:10:13 | R1 ON | Pantry Door | A | B | ✗ |
| 4 | 16:10:22 | R1 OFF | Pantry Door | A | B | ✗ |
| 5 | 16:10:26 | E1 ON | Stove | A | B | ✗ |
| 6 | 16:10:31 | R7 ON | Fridge Door | A | A | ✓ |
| 7 | 16:10:32 | R1 ON | Pantry Door | A | B | ✗ |
| 8 | 16:10:35 | R7 OFF | Fridge Door | A | A | ✓ |
| 9 | 16:10:47 | R1 OFF | Pantry Door | A | A | ✓ |
| 10 | 16:11:04 | R2 ON | Cutlery Drawer | A | B | ✗ |
| 11 | 16:11:06 | R2 OFF | Cutlery Drawer | A | B | ✗ |
| 12 | 16:11:12 | R2 ON | Cutlery Drawer | A | A | ✓ |
| 13 | 16:11:15 | R2 OFF | Cutlery Drawer | A | A | ✓ |
| 14 | 16:11:32 | E1 OFF | Stove | A | B | ✗ |
| 15 | 16:11:48 | R6 ON | Medicines Cabinet | A | A | ✓ |
| 16 | 16:11:52 | R6 OFF | Medicines Cabinet | A | A | ✓ |
| 17 | 16:12:09 | R6 ON | Medicines Cabinet | A | A | ✓ |
| 18 | 16:12:11 | R6 OFF | Medicines Cabinet | A | A | ✓ |
| 19 | 16:12:28 | P1 ON | Dining Chair 1 | A | A | ✓ |
| 20 | 16:13:18 | R7 ON | Fridge Door | B | B | ✓ |
| 21 | 16:13:25 | R7 OFF | Fridge Door | B | B | ✓ |
| 22 | 16:13:31 | P4 ON | Dining Chair 2 | B | B | ✓ |
| 23 | 16:13:35 | P4 OFF | Dining Chair 2 | B | B | ✓ |
| 24 | 16:13:46 | R7 ON | Fridge Door | B | B | ✓ |
| 25 | 16:13:50 | R7 OFF | Fridge Door | B | B | ✓ |
| 26 | 16:14:00 | R2 ON | Cutlery Drawer | B | B | ✓ |
| 27 | 16:14:04 | R2 OFF | Cutlery Drawer | B | B | ✓ |
| 28 | 16:14:24 | P4 ON | Dining Chair 2 | B | B | ✓ |
| 29 | 16:16:45 | P1 OFF | Dining Chair 1 | A | A | ✓ |
| 30 | 16:16:49 | P4 OFF | Dining Chair 2 | B | B | ✓ |
| 31 | 16:17:01 | P2 ON | Office Chair | A | A | ✓ |
| 32 | 16:17:28 | R7 ON | Fridge Door | B | B | ✓ |
| 33 | 16:17:34 | R7 OFF | Fridge Door | B | B | ✓ |

Table 5.2 – Qualitative example of 29 sequential sensor events the session D4mae_instance1 of MARBLE dataset. Each row color corresponds to the ground truth resident: green (A), blue (B), orange (C), purple (D). Red rows represent incorrect predictions.

| Step | Time | Event | Facility | Predicted ID | GT ID | Correct |
|---|---|---|---|---|---|---|
| 1 | 11:27:40 | R7 ON | Fridge | A | A | ✓ |
| 2 | 11:27:44 | R7 OFF | Fridge | A | A | ✓ |
| 3 | 11:27:58 | P1 ON | Dining Chair 1 | A | A | ✓ |
| 4 | 11:28:02 | P4 ON | Dining Chair 2 | B | B | ✓ |
| 5 | 11:28:03 | P5 ON | Dining Chair 3 | C | C | ✓ |
| 6 | 11:29:48 | P6 ON | Dining Chair 4 | D | D | ✓ |
| 7 | 11:31:54 | P4 OFF | Dining Chair 2 | B | B | ✓ |
| 8 | 11:31:56 | P1 OFF | Dining Chair 1 | A | A | ✓ |
| 9 | 11:31:57 | P5 OFF | Dining Chair 3 | C | C | ✓ |
| 10 | 11:31:59 | P6 OFF | Dining Chair 4 | D | D | ✓ |
| 11 | 11:32:15 | E2 ON | Television | A | B | ✗ |
| 12 | 11:32:17 | P8 ON | Living Room Couch 2 | A | A | ✓ |
| 13 | 11:32:18 | P9 ON | Living Room Couch 3 | E | C | ✗ |
| 14 | 11:32:22 | P3 ON | Living Room Couch 1 | B | B | ✓ |
| 15 | 11:33:46 | P9 OFF | Living Room Couch 3 | C | C | ✓ |
| 16 | 11:33:55 | R5 ON | Pots Drawer | C | C | ✓ |
| 17 | 11:33:57 | R5 OFF | Pots Drawer | C | C | ✓ |
| 18 | 11:34:06 | E1 ON | Stove | C | C | ✓ |
| 19 | 11:34:08 | R1 ON | Pantry Door | C | C | ✓ |
| 20 | 11:34:23 | R1 OFF | Pantry Door | C | C | ✓ |
| 21 | 11:35:17 | R1 ON | Pantry Door | C | C | ✓ |
| 22 | 11:35:23 | R1 OFF | Pantry Door | C | C | ✓ |
| 23 | 11:35:25 | P8 OFF | Living Room Couch 2 | A | A | ✓ |
| 24 | 11:36:17 | R5 ON | Pots Drawer | C | C | ✓ |
| 25 | 11:36:22 | R5 OFF | Pots Drawer | C | C | ✓ |
| 26 | 11:36:43 | E1 OFF | Stove | C | C | ✓ |
| 27 | 11:37:10 | R7 ON | Fridge | C | C | ✓ |
| 28 | 11:37:15 | R7 OFF | Fridge | C | C | ✓ |
| 29 | 11:37:27 | R7 ON | Fridge | C | C | ✓ |

Indeed, even a human observer might reasonably interpret this early sequence as one resident preparing food in the kitchen.

A key transition occurs at step 19 (P1 ON, *Dining Chair 1*), when resident A sits down, implying that they are now stationary. The subsequent event at step 20 (R7 ON, *Fridge Door*) cannot plausibly be performed by the same individual, since it requires movement away from the dining area. The model, guided by its contextual reasoning, thus infers the presence of a second active user (resident B) at this point. From step 20 onward, it consistently distinguishes between the two users: A remains seated or later moves to the office chair, while B continues to manipulate the fridge and dining chairs. This pattern demonstrates the model's ability to dynamically update its user hypotheses when logically incompatible actions emerge in the sequence.

**Four-resident scenario.** Table 5.2 presents a more complex example with four residents (A–D) active across kitchen, dining, and living areas. The model correctly identifies the initial seating configuration (steps 1–6) and the near-simultaneous stand-ups (steps 7–10), indicating that spatially distinct pressure-mat events are separable and short bursts of parallel transitions are handled well.

Two errors merit closer inspection. First, at step 11 (E2 ON, *Television*), all four residents have just left their chairs, so multiple residents are equally plausible television users. In such a tie, a reasonable heuristic is to favor the earliest resident who regained mobility in the living area—here, resident B. The model instead assigns this event to A, revealing a limitation in leveraging *temporal precedence among newly mobile candidates*. Second, at step 13 (P9 ON, *Living Room Couch 3*), the model assigns the event to an incorrect new identity (shown as "E" in the table), whereas ground truth is resident C. This misassignment is not driven by exclusivity or timing but is a pure identification error. Notably, by step 15 (P9 OFF at the same location), the model re-aligns the identity to C, suggesting an implicit short-horizon *self-correction* mechanism that restores local temporal–spatial consistency once additional evidence arrives.

Beyond these isolated mistakes, the model accurately tracks resident C's extended cooking routine (steps 16–29) involving coherent manipulations of the stove (E1), pantry (R1), and fridge (R7), demonstrating stable identity maintenance over long action chains.

**Summary.** Across both examples, our qualitative observations reveal a consistent pattern: the proposed LLM-based model excels when contextual cues (e.g., spatial exclusivity, stationary state, or temporal continuity) are present, but can be momentarily confused in densely overlapping or spatially ambiguous situations. Importantly, the model's inference mechanism exhibits human-like reasoning. It initially assumes minimal resident count and only introduces new residents when incompatible evidence arises. These qualitative results complement the quantitative findings by showing that

Table 5.3 – Comparison of Data Association performances of baseline approaches and our approach on 2 datasets. Acc and CRL are calculated using a window size $w = 16$.

| Dataset | MARBLE [22] | | | | MuRAL [49] | | | |
|---|---|---|---|---|---|---|---|---|
| Metric | Acc | CRL | BLEU | RCE | Acc | CRL | BLEU | RCE |
| All-to-One | 0.729 | 8.606 | 0.707 | 0.789 | 0.551 | 4.489 | 0.510 | 1.593 |
| sMRT | 0.119 | 0.641 | 0.079 | 15.10 | 0.529 | 3.321 | 0.497 | 1.560 |
| NEP+Search | 0.825 | 8.847 | 0.812 | 0.500 | 0.674 | 4.982 | 0.755 | **0.375** |
| BC | 0.792 | 8.710 | 0.797 | 0.136 | **0.728** | **5.476** | **0.774** | 0.437 |
| LADA | **0.849** | **8.944** | **0.883** | **0.045** | 0.707 | 5.337 | 0.750 | 0.702 |

logical consistency and contextual reasoning are central to robust multi-resident data association in smart home environments.

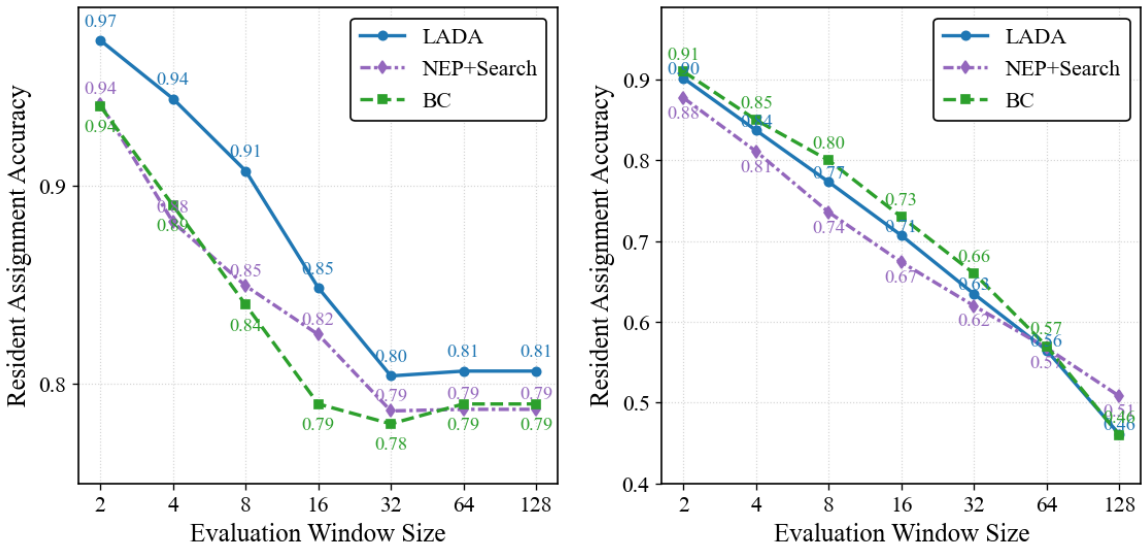
### 5.5.3 Quantitative Results

In this subsection, we quantitatively compare the data association performance of LADA with the methods proposed in the previous sections on the MARBLE and MuRAL datasets.

**Overall performance.** As shown in Table 5.3, **LADA** demonstrates consistently strong data association performance across both datasets, outperforming or matching competitive baselines while requiring no task-specific training.

On MARBLE [22], LADA achieves the highest accuracy (0.849), surpassing all baseline approaches, including NEP+Search (0.825) and BC (0.792). It also attains the best long-range consistency with the highest Correct Run Length (CRL) score (8.944) and the highest Bilingual Evaluation Understudy (BLEU) score (0.883), indicating superior sequence-level identity assignment. Moreover, LADA yields the lowest Resident Count Error (RCE) (0.045) by a clear margin, reflecting highly accurate resident-count reasoning on this dataset. We attribute LADA's strong performance on MARBLE to the characteristics of the dataset itself: sensor events in MARBLE are semantically well-defined, exhibit minimal noise, and correspond to relatively simple and regular resident activity patterns. In addition, mutual exclusivity between sensors is more pronounced, which reduces ambiguity in resident attribution. These properties align well with the strengths of LLM-based reasoning, enabling LADA to effectively leverage semantic cues and logical consistency when performing data association.

On MuRAL [49], LADA continues to show strong generalization performance. While BC achieves the highest accuracy (0.728) and CRL (5.476), LADA remains highly competitive, reaching an accuracy of 0.707 and a CRL of 5.337. Its BLEU score (0.750) is comparable to other strong baselines, and its RCE (0.702) indicates stable resident-count estimation, although NEP+Search attains the lowest RCE on this dataset.



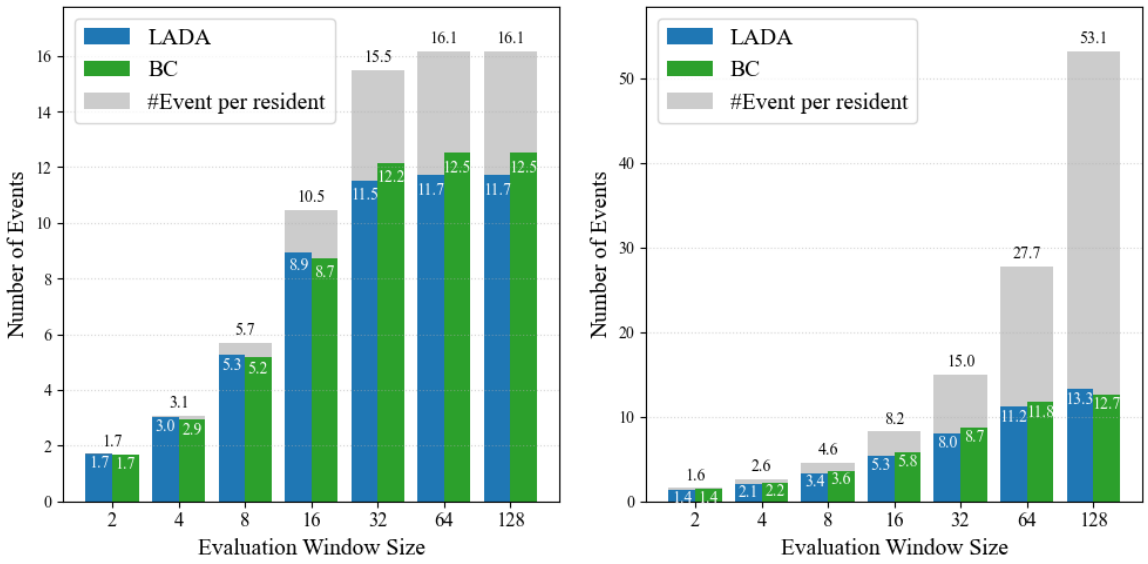

(a) MARBLE (b) MuRAL

Figure 5.3 – Resident assignment accuracy as a function of the evaluation window size ($\log_2$ scale) on the MARBLE, and MuRAL datasets.

Overall, these results demonstrate that LADA achieves robust and competitive data association performance across diverse environments. Without relying on supervised training or dataset-specific adaptation, LADA matches or exceeds strong baseline methods on key metrics, highlighting the effectiveness of LLM-driven reasoning for multi-resident data association tasks.

**Error Accumulation.** As discussed in Chapter 3, within a fixed evaluation window, misallocations accumulated during the reasoning process may propagate to subsequent decisions, leading to a gradual degradation in assignment accuracy. Consequently, increasing the evaluation window size generally amplifies error accumulation and reduces overall data association performance.

Figure 5.3 plots resident-assignment accuracy as a function of the evaluation window size. On MARBLE, all methods exhibit declining accuracy as the window length increases; however, LADA shows a noticeably slower degradation than NEP+Search and BC for small to medium window sizes. In particular, for $w \leq 8$, LADA's accuracy decreases more gradually than the baselines, indicating strong robustness to local error accumulation. Although LADA's performance also drops more rapidly when the window size exceeds 8, it consistently maintains higher accuracy than the other methods across all evaluated window lengths. This suggests that LADA is more resilient to short-range reasoning errors while still preserving an advantage under longer temporal horizons. On MuRAL, the three methods exhibit highly similar degradation trends as the evaluation window increases, reflecting the greater temporal complexity and stronger long-range dependencies in this dataset. Accuracy decreases at comparable

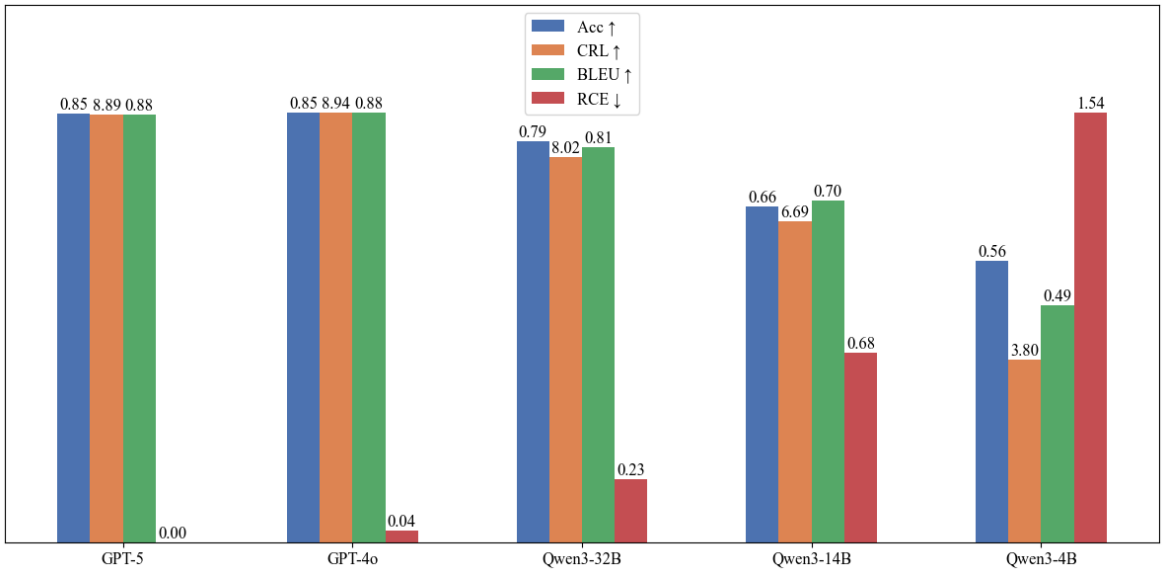



(a) MARBLE (b) MuRAL

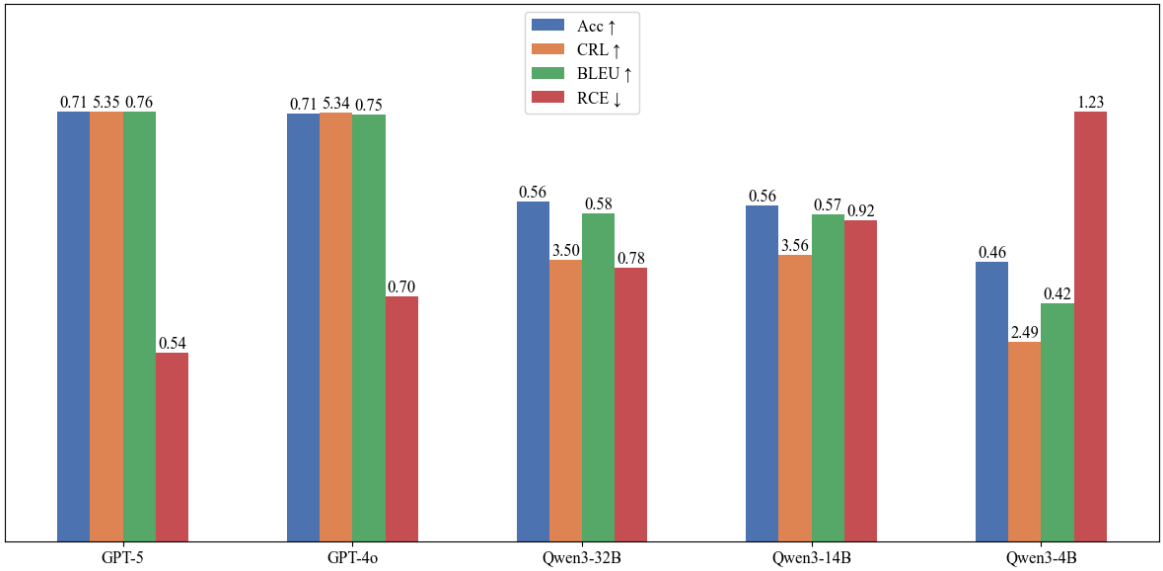
Figure 5.4 – Evaluation of Correct Run Length (CRL) and the average number of events per resident (gray bars) in different evaluation window sizes ($\log_2$ scale) for MARBLE, and MuRAL. CRL quantifies how long a method can continuously assign the correct resident before making an error.

rates for LADA, NEP+Search, and BC, indicating that error propagation affects all approaches in a similar manner. Overall, BC achieves slightly higher accuracy than LADA across most window sizes, while LADA remains competitive but does not show a clear robustness advantage in this more challenging setting. Taken together, these results indicate that LADA is particularly effective at mitigating error accumulation in locally structured scenarios, such as MARBLE, whereas in datasets with long and highly interleaved activity sequences, like MuRAL, the benefit of LLM-based reasoning is less pronounced and performance is largely governed by dataset complexity.

Figure 5.4 further evaluates the CRL, defined as the average number of consecutive correctly assigned events for each resident before the first error occurs, together with the average number of events per resident (gray bars). On the MARBLE dataset, LADA maintains a slight advantage over or performs comparably to BC in short-to-medium evaluation windows ($w$ = 2–16). However, in the longer horizons ($w \geq 32$), BC surpasses LADA, stabilizing at a higher CRL (e.g., 12.5 vs. 11.7 at $w = 128$). However, as shown in Figure 5.3, LADA consistently achieves higher Resident Assignment Accuracy (Acc) than BC across these intervals. This implies that LADA tends to commit its first error earlier in long sequences, leading to a lower CRL compared to BC. Nevertheless, it is significantly more robust to such errors, allowing it to recover and maintain higher overall Acc. On the MuRAL dataset, the trend is distinct. BC consistently outperforms LADA across small and medium window sizes ($w$ = 2–64), showing stronger resistance to early assignment errors within typical activity spans.




(a) MARBLE




(b) MuRAL

Figure 5.5 – Comparison of LADA's performance when using different underlying LLMs on MARBLE (top) and MuRAL (bottom). We evaluate closed-source models (GPT-5 and GPT-4o) and open-source Qwen3 models of varying sizes (32B, 14B, 4B). Larger and more capable LLMs consistently yield better accuracy, identity consistency, and sequencing quality, while smaller models lead to substantial performance degradation.

### 5.5.4 Impact of Models

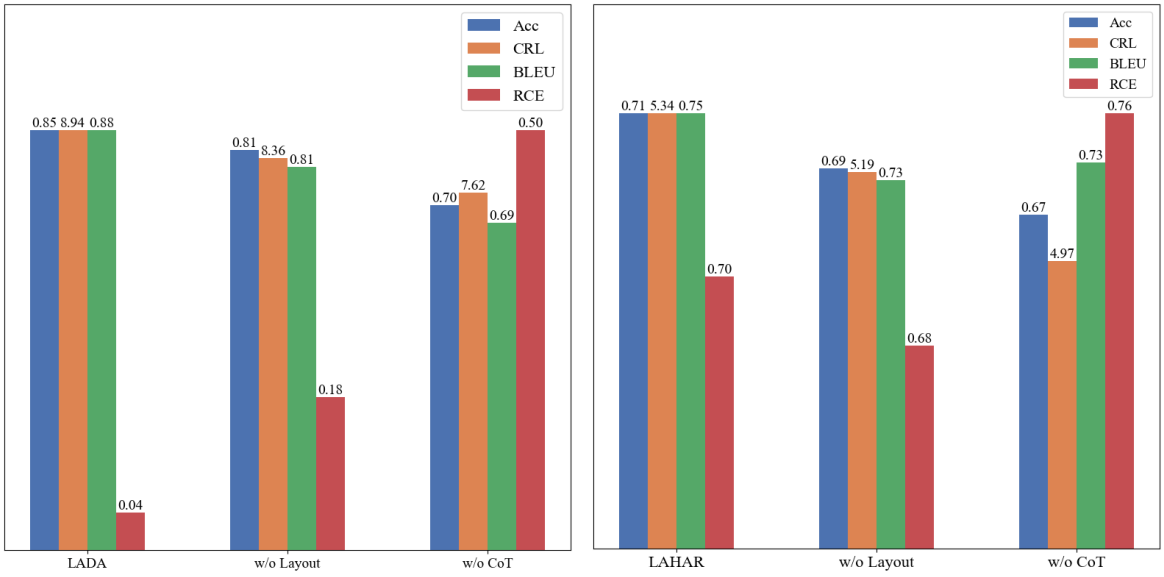
The policy model in LADA is implemented using a large language model, meaning that the overall performance of LADA is closely tied to the reasoning capability and

capacity of the underlying LLM. To examine how different model sizes and families affect LADA, we evaluate a range of both closed-source and open-source LLMs. For closed-source models, in addition to OpenAI's GPT-4o model used in our baseline, we further include GPT-5, one of the most advanced commercially available LLMs at the time of experimentation. Because both models are proprietary, their exact parameter counts are undisclosed; however, it is widely believed that GPT-5 is larger than GPT-4o, and that both models fall within the trillion-parameter scale or beyond. Their inclusion allows us to assess how state-of-the-art reasoning capabilities influence LADA's performance. For open-source models, we adopt three variants from the widely used Qwen3 family: 32B, 14B, and 4B. These models support an optional "thinking mode" that enables extended exploratory CoT reasoning. However, since our prompts already prescribe explicit CoT reasoning steps, and to ensure a controlled and fair comparison, we disable this feature during evaluation. This set of experiments allows us to systematically analyze how model capacity and architectural differences influence LADA's ability to perform multi-resident data association under a unified zero-shot prompting framework.

Figure 5.5 compares the performance of LADA when equipped with different LLM backbones on the MARBLE and MuRAL datasets. Overall, the results show a clear trend: the performance of LADA degrades as the capability of the underlying LLM decreases. Among all metrics, RCE is particularly sensitive to model capacity and consistently exposes performance gaps between models. This suggests that accurate estimation of local resident counts is highly correlated with the quality of data association, and that RCE serves as the most robust indicator for evaluating Data Association performance.

Across both datasets, GPT-5 achieves the best performance. On MARBLE, its RCE drops to as low as 0.002, and all other metrics surpass those of the supervised baselines. However, the improvement from GPT-4o to GPT-5 is relatively modest, especially on MuRAL where only the RCE metric shows a noticeable reduction (0.17), while the remaining metrics remain nearly identical. This indicates that LADA's performance is, to some extent, no longer bottlenecked by the LLM's raw capability and further increases in LLM power may lead to diminishing returns.

In contrast, the Qwen3 family exhibits substantial performance drops on both datasets due to their smaller parameter scales. The degradation is more pronounced on MuRAL than on MARBLE, suggesting that MuRAL contains more complex or harder-to-interpret activity patterns for LLMs. Notably, Qwen3-4B performs worse than the trivial "All-to-One" baseline on both datasets (cf. Table 5.3), indicating that small LLMs fail under LADA's zero-shot setting. Nevertheless, the performance gap between Qwen3-32B and Qwen3-14B is relatively small, implying that LADA's framework may partially mitigate the influence of model size by leveraging structured prompts and CoT reasoning.




(a) MARBLE (b) MuRAL

Figure 5.6 – Ablation study on MARBLE (left) and MuRAL (right). "w/o CoT" removes LADA's chain-of-thought reasoning, and "w/o Layout" removes layout-grounding information. Both ablations lead to noticeable performance drops across all metrics, demonstrating the importance of these components.

### 5.5.5 Impact of CoT Reasoning and Layout

To assess the contribution of the key components in LADA, we perform an ablation study by removing chain-of-thought reasoning ("w/o CoT") and layout-grounding information ("w/o Layout") on both the MARBLE and MuRAL datasets. The results are summarized in Figure 5.6.

Overall, both ablations lead to clear and consistent performance degradation across all four evaluation metrics. Removing CoT reasoning substantially weakens LADA's ability to maintain temporally coherent identity assignments. Without step-by-step reasoning, the model is forced to output identity labels directly from the prompt, resulting in more frequent early-stage assignment errors and shorter continuous runs of correct predictions. This effect is especially visible in the CRL metric, where "w/o CoT" shows the largest declines on both datasets. The BLEU and RCE metrics similarly reveal that removing CoT disrupts the model's ability to produce logically consistent association sequences, suggesting that explicit reasoning steps are critical for stabilizing decision-making of LADA over long event chains.

Removing layout grounding ("w/o Layout") also leads to a noticeable drop in performance, though its impact differs from that of CoT. The absence of spatial information reduces LADA's ability to leverage sensor placement and environmental structure when resolving ambiguous or overlapping interactions. As a result, the

model exhibits more uncertainty in distinguishing residents whose activities occur near each other, reflected by lower accuracy and shorter CRL. The effects are consistent across both datasets, with MuRAL showing slightly larger sensitivity due to its richer spatial diversity and more complex multi-resident activities.

Taken together, these results demonstrate that both CoT reasoning and layout grounding are essential components of LADA. CoT primarily enhances temporal consistency and reasoning stability, while layout cues provide spatial disambiguation that complements the model's zero-shot reasoning process. Their combined effect is necessary for achieving the strong performance observed in the full model.

## 5.6 Conclusion

In this chapter, we propose an LLM powered ambient based multi resident Data Association method named LADA. Building upon our original MDP framework, this method replaces the policy component with an off the shelf LLM, enabling it to perform contextual reasoning based on textualized system states together with broader contextual information, thereby achieving resident assignment. Extensive experiments on the MARBLE and MuRAL datasets demonstrate the effectiveness of LADA. Without any training, LADA achieves performance comparable to that of trained NEP+Search and BC on both datasets, and shows particularly strong performance on the MARBLE dataset, surpassing all existing methods. Ablation studies further confirm the importance of contextual inputs and chain of thought reasoning. Performance comparisons across models reveal a strong correlation between model size and the data association performance of LADA, highlighting the importance of future research on model size reduction for LADA.

# CHAPTER 6

# FINE-TUNING LLMS FOR DATA-DRIVEN DATA ASSOCIATION POLICIES

## 6.1 Introduction

In the preceding chapter, we introduced LADA (LLM-based Autonomous Data Association), a paradigm that deploys Large Language Models (LLMs) as a general-purpose policy. By leveraging zero-shot Chain-of-Thought (CoT) prompting, LADA successfully unlocked the context reasoning capabilities of LLMs, enabling unsupervised Data Association without the need for domain-specific training.

However, despite its theoretical appeal, our empirical evaluations revealed two critical limitations that hinder LADA's practical deployment. First, regarding environmental adaptability, LADA exhibits a significant performance dichotomy. While it excels on the semantically clear MARBLE dataset, it struggles to generalize to the MuRAL dataset—an environment characterized by high sensor noise and intricate, interleaving resident behaviors. In this complex setting, the general reasoning of zero-shot models failed to surpass the supervised Behavior Cloning (BC) baseline, suggesting that generic semantic knowledge is insufficient to capture domain-specific statistical distributions. Second, regarding model scalability and deployment, LADA's success is heavily contingent on the emergent abilities of massive proprietary models

(e.g., GPT-4o). Our experiments indicate that smaller-scale open-source models (such as the Qwen3-4B) lack the intrinsic reasoning capacity to follow complex zero-shot instructions, leading to tracking failure. Relying on hyperscale cloud models introduces prohibitive inference costs, high latency, and privacy concerns, rendering them unsuitable for local edge deployment in real-world smart homes.

To bridge these gaps, this chapter presents a synthesis of our previous methodologies, effectively combining the generative paradigm of LADA with the supervised framework of BC. We retain the formulation established in the previous chapter, which treats Data Association as a conditional text generation task, but move away from zero-shot inference. Instead, we adopt the supervised training strategy introduced in Chapter 4, employing Low-Rank Adaptation (LoRA) [99] to fine-tune smaller, open-source LLMs on ground-truth demonstrations. This method is referred to as LLM+BC. It enables lightweight models to learn complex, noise-resilient policies directly from data, thereby achieving the high performance of supervised methods while preserving the flexible, text-based interface of LLMs for efficient local execution.

Our empirical evaluations on the MARBLE and MuRAL datasets yield several key insights. First, we demonstrated that generic semantic knowledge alone is insufficient for handling high-entropy environments. While the zero-shot LADA framework struggled with the sensor noise and interleaving behaviors of MuRAL, the LLM+BC approach significantly outperformed both the zero-shot baseline and traditional supervised methods. This confirms that open-source models with modest parameter sizes (e.g., 4B parameters), when fine-tuned via LoRA, can internalize domain-specific statistical distributions more effectively than generic hyperscale models. Second, the structural transition from explicit CoT reasoning to direct state-to-action mapping proved advantageous for robustness. The LLM+BC method exhibited a slower rate of performance degradation as the evaluation window increased, suggesting that the model learned to mitigate error accumulation through supervised exposure to expert trajectories. However, qualitative analysis highlighted persistent challenges, particularly regarding rigid temporal interpretation and the resolution of simultaneous events in shared spaces, indicating areas where physical constraints need to be more explicitly integrated into the learning process. Finally, regarding practical deployment, this chapter validates the feasibility of Edge AI for smart homes. By successfully replacing proprietary cloud-based models with efficient, locally executable LLMs, we have established a pathway for data association systems that are not only high-performing but also privacy-preserving and cost-effective. The cross-environment evaluation further suggests that training on datasets with clear semantic boundaries (like MARBLE) yields more transferable policies than training on highly complex, noisy data, providing valuable guidelines for future data collection and curriculum training strategies.

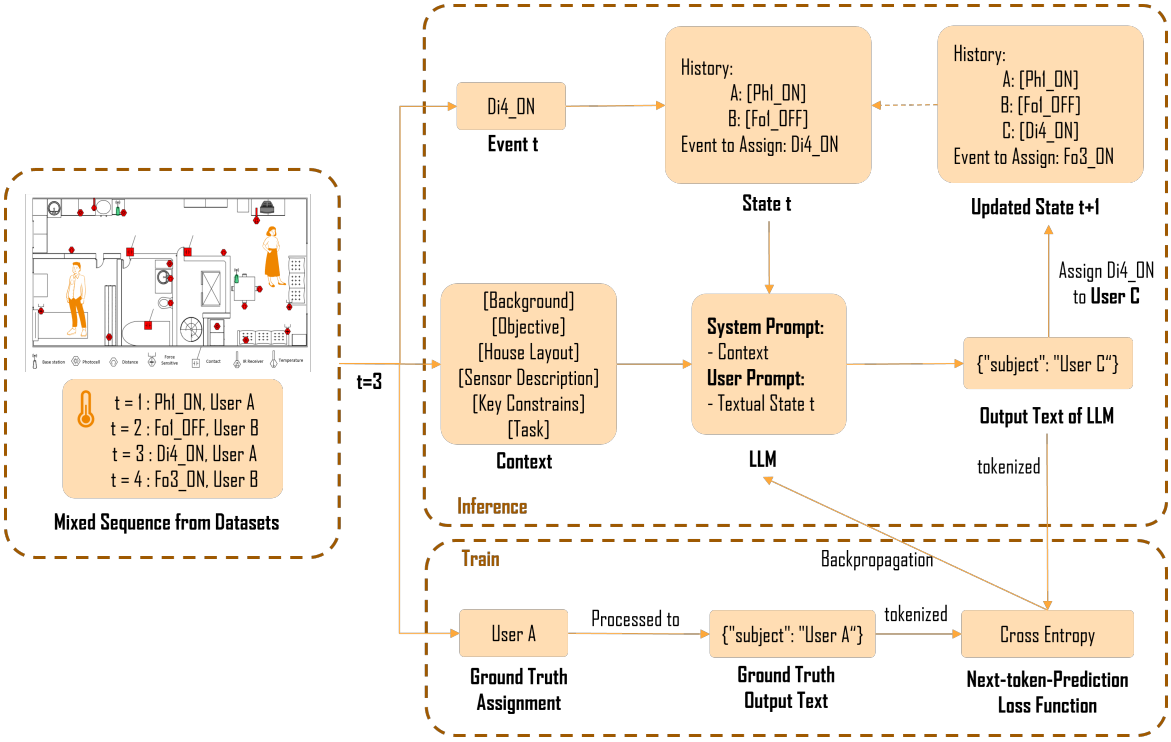



Figure 6.1 – Illustration of the proposed LLM+BC multi-resident data association method

## 6.2 Methodology

Figure 6.1 illustrates our proposed LLM+BC framework, which reformulates the multi-resident data association problem as a supervised conditional text generation task. The workflow begins by serializing the static environmental priors and the dynamic interaction history into a structured textual prompt consisting of a System Prompt and a User Prompt. This context conditions the Large Language Model to directly generate the resident assignment in a concise, machine-parsable JSON format, effectively mapping the current state to an action without intermediate reasoning steps. During the training phase, we employ a BC strategy where the ground-truth assignments serve as target sequences and the model is fine-tuned by minimizing the Cross-Entropy Loss on the Next-Token Prediction (NTP) objective, thereby aligning its internal policy with the expert distribution.

### 6.2.1 Problem Formulation

Adhering to the generative paradigm established in Chapter 5, we continue to formulate the multi-resident data association problem as a conditional text generation task. The Markov Decision Process (MDP) state $s_t$ and context $c$ are serialized into a textual prompt $\mathcal{X}_t$, and the policy $\pi_\theta$ is parameterized by an LLM.

However, distinct from the zero-shot approach in the previous chapter, we adapt this formulation for Supervised Fine-Tuning (SFT). In this supervised setting, we streamline the generation process by omitting the intermediate CoT reasoning steps. Instead of generating a rationale followed by a conclusion, the model is trained to map the input prompt $\mathcal{X}_t$ *directly* to the target semantic token $\mathcal{Y}_t$ corresponding to the resident identity (e.g., `{"subject": "User A, User B"}`).

Formally, the optimization objective shifts from triggering emergent reasoning to maximizing the likelihood of the ground-truth action label given the context:

$$a_t \sim P_\theta(\mathcal{Y}_t \mid \mathcal{X}_t), \tag{6.1}$$

where the model implicitly learns the complex statistical dependencies of the environment through gradient updates, rather than explicit in-context deduction.

### 6.2.2 Training Objective and Optimization

To fine-tune the model parameters $\theta$, we adopt the standard Next-Token Prediction (NTP) [30, 147] objective. The fundamental principle of NTP is the autoregressive decomposition of the sequence probability: the joint probability of a target sequence is modeled as the product of the conditional probabilities of each token given its predecessors.

Consider a specific decision step $t$ from the interaction trajectory. The input consists of a prompt sequence $\mathcal{X}^t$ (serializing the context $c$ and state $s_t$) and a target label sequence $\mathcal{Y}^t = \{y_1^t, y_2^t, \ldots, y_{L_t}^t\}$, where $L_t$ denotes the length of the target response for event $e_t$. The training objective is to maximize the likelihood of the target tokens $\mathcal{Y}^t$ conditioned on the prompt $\mathcal{X}^t$. Formally, the probability of the target sequence at step $t$ is factorized using the chain rule:

$$P_\theta(\mathcal{Y}^t \mid \mathcal{X}^t) = \prod_{i=1}^{L_t} P_\theta(y_i^t \mid \mathcal{X}^t, y_{<i}^t), \tag{6.2}$$

where $y_{<i}^t = \{y_1^t, \ldots, y_{i-1}^t\}$ denotes the history of generated tokens within the current response prior to index $i$.

We optimize the model by minimizing the **Cross-Entropy Loss** over the generated tokens. For a single decision step $t$, the loss function is defined as the Negative Log-Likelihood (NLL):

$$\mathcal{L}_t(\theta) = -\sum_{i=1}^{L_t} \log P_\theta(y_i^t \mid \mathcal{X}^t, y_{<i}^t). \tag{6.3}$$

During training, we employ *teacher forcing* [203], where the model receives the ground-truth history $y_{<i}^t$ as input when predicting $y_i^t$. Crucially, a **loss mask** is applied

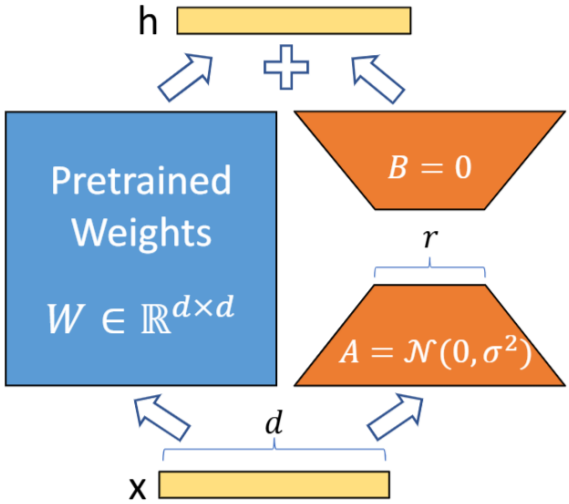



Figure 6.2 – Illustration of the Low-Rank Adaptation (LoRA). Reproduced from Hu et al. [99].

to the prompt tokens $\mathcal{X}^t$. This ensures that gradients are back-propagated exclusively from the prediction of the target resident assignment $\mathcal{Y}^t$, allowing the model to adapt its decision head while maintaining stable representations of the input context.

### 6.2.3 Parameter-Efficient Fine-Tuning via Low-Rank Adaptation

Given the immense parameter size of modern LLMs, performing full fine-tuning is computationally prohibitive and memory-intensive, making it impractical for local deployment in smart home environments. Furthermore, full parameter updates risk *catastrophic forgetting*, where the model loses its general linguistic reasoning capabilities while fitting the specific task data.

To address these challenges, we employ **Low-Rank Adaptation (LoRA)** [99]. As illustrated in Figure 6.2, LoRA freezes the pre-trained model weights and injects trainable rank-decomposition matrices into each layer of the Transformer architecture, thereby significantly reducing the number of trainable parameters.

The theoretical foundation of LoRA is based on the hypothesis that the change in weights during model adaptation has a low "intrinsic rank" [5]. Consider a pre-trained weight matrix $W_0 \in \mathbb{R}^{d \times k}$. During full fine-tuning, the weight is updated to $W_0 + \Delta W$. LoRA constrains this update $\Delta W$ by representing it as the product of two low-rank matrices $B \in \mathbb{R}^{d \times r}$ and $A \in \mathbb{R}^{r \times k}$, where the rank $r \ll \min(d, k)$. The forward pass for an input $x$ is thus modified as:

$$h = W_0 x + \frac{\alpha}{r} BAx, \tag{6.4}$$

where $W_0$ remains frozen, while $A$ and $B$ contain the trainable parameters. Here, $\alpha$ is a constant scalar hyperparameter used for scaling. The scaling factor $\frac{\alpha}{r}$ is critical for

stabilizing training. It balances the magnitude of the gradient updates, ensuring that the need for hyperparameter retuning is minimized when varying the rank $r$.

Regarding initialization, we follow the standard practice: $A$ is initialized with random Gaussian noise, and $B$ is initialized to zero. This ensures that $\Delta W = 0$ at the beginning of training, preserving the original behavior of the pre-trained model essentially. A key advantage of this approach for our application is the **zero inference latency**. After training, the learned low-rank matrices can be explicitly merged back into the original weights ($W' = W_0 + \frac{\alpha}{r}BA$). This yields a standard model architecture that requires no additional computation during the inference phase, perfectly aligning with our requirement for efficient edge deployment.

### 6.2.4 Prompt Template and Output Format

To enable the language model to process sensor data, we adopt the serialization framework established in Chapter 5, mapping the numerical state $s_t$ and context $c$ into a structured textual input $\mathcal{X}_t$. While the overall architecture of the prompt—comprising a static **System Prompt** and a dynamic **User Prompt**—remains consistent with LADA, we introduce specific modifications to adapt the input for direct supervised fine-tuning rather than zero-shot reasoning.

The System Prompt encodes the static context $c$, including the `[Background]`, `[Objective]`, `[House Layout]`, `[Sensor Description]`, and `[Key Constraints]`. These sections are identical to those defined in Chapter 5, ensuring the model retains a grounded understanding of the physical environment.

However, we significantly modify the `[Task]` section. Unlike the zero-shot setting where the model was instructed to "think step-by-step" and provide a rationale, the refined task description explicitly constrains the model to function as a pattern recognition engine. It directs the model to output the prediction directly based on historical patterns, removing any requests for intermediate reasoning or CoT generation.

**System Prompt for MARBLE dataset used in our proposed LLM+BC method**

**[Background]**
Same as LADA, See Chapter 5.

**[Objective]**
Same as LADA, See Chapter 5.

**[House Layout]**
Same as LADA, See Chapter 5.

**[Sensor Description]**
Same as LADA, See Chapter 5.

**[Key Constraints]**
Same as LADA, See Chapter 5.

**[Task]**
Based on the historical activity patterns of each user and the current sensor event, predict which user(s) triggered the current event. Output user indices (0-indexed) as a multi-label prediction.

The User Prompt represents the dynamic state $s_t$. As shown in our prompt design, it serializes the current sensor reading $e_t$ (including timestamp, sensor ID, and location) and the recent history of residents $h_{<t}$ into a human-readable format. In contrast to LADA, we streamline this prompt by removing the reasoning instructions. Instead, the prompt ends directly with the query and the required JSON output format, minimizing token consumption and focusing the model's attention solely on the mapping from state to label. Consequently, the target output $\mathcal{Y}_t$ is simplified. Instead of a verbose paragraph containing reasoning steps, the ground-truth target for fine-tuning is formatted as a concise JSON object containing only the resident's identity (e.g., `{"subject": "User A"}`). This strict formatting reduces the sequence length significantly, accelerating both the training and inference processes while enforcing a consistent, machine-parsable interface for the downstream system.

**User Prompt for MARBLE dataset used in our proposed LLM+BC method**

Observed users:
— User_1 (latest 1 events):
[12:14:44] P4 ON
You can assign this event to:
1. An existing user (from the list above)
2. A new user (User_2) if this event belongs to someone not yet observed
Current sensor event observation:

```
{
  "time": "12:14:44",
  "event": "P1 ON",
  "sensor": "P1",
  "action": "ON",
  "location": "dining room"
}
```

Based on the historical patterns and location constraints, which user(s) triggered this event?
Output your prediction in JSON format: {{"subject": "User_X"}} or {{"subject": "User_X, User_Y"}} for multiple users.

## 6.3 Experimental Evaluation

### 6.3.1 Experimental Setup

Consistent with previous chapters, we adhere to the experimental protocol established in Chapter 3. Given the utilization of LLMs, we exclude the CASAS dataset from our validation, following the precedent set in Chapter 5.

We employ the Qwen3-4B-Instruct-2507 as the foundation model for all experiments. The LoRA configuration consists of a rank $r = 16$, a scaling factor $\alpha = 32$, and a dropout rate of 0.05. This adaptation is applied to all linear projections within both the attention mechanisms and feed-forward networks. Training is performed using Bfloat16 (bf16) precision to ensure computational efficiency. All experiments were conducted using the Hugging Face `Transformers` Trainer framework and the Hugging Face `PEFT` library for the LoRA implementation. The computational setup consisting of two NVIDIA A100 (80GB) GPUs. To leverage hardware parallelism, we employed the `Accelerate` library for Distributed Data Parallel (DDP) training. The model was fine-tuned for 1 epoch with a maximum sequence length of 1024 tokens given the length of the prompt is about 900 tokens. We set the per-device batch size to 6 with gradient accumulation steps of 4, resulting in an effective global batch size of 48 ($6 \times 2 \times 4$). Optimization was performed using a cosine learning rate scheduler with a peak learning rate of $2 \times 10^{-4}$ and a warmup ratio of 0.1, regularized by a weight decay of 0.01. The Permutation-based Data augmentation strategy introduced in Section 4.2.5 were also incorporated to enhance model robustness.

For inference, we configure the generation process to produce a maximum of 50 new tokens. We align our history context lengths $L$ with those in Chapter 4, using a window of 1 for MuRAL and 11 for MARBLE.

### 6.3.2 Qualitative Results

Table 6.1 presents a sequential excerpt of 33 sensor events from Session 12 of the MuRAL dataset.

While the system correctly initializes tracks for Residents A and B upon their simultaneous entry (Steps 0–3), a critical identity switch occurs at Step 5. We hypothesize that this early failure stems from the model's rigid interpretation of temporal constraints. As shown in the log, the Living Room Couch turns `OFF` at 19:04:41 (Step 4), and the Bathroom turns `ON` merely one second later at 19:04:42 (Step 5). The LLM likely inferred that it was physically infeasible for Resident A to traverse from the living room to the bathroom within such a negligible time window. Consequently, to satisfy this apparent physical constraint, the model assigned the bathroom activity to Resident B, leading to a persistent identity swap for the subsequent events. In reality, the sensor timestamps did not perfectly align with physical movements; Resident A had placed a bag on the couch at Step 2 and immediately departed. The `OFF` event at

Table 6.1 – Qualitative example of 33 sequential sensor events from the session 12 of MuRAL dataset. Each row color corresponds to the ground truth resident: green (A), blue (B), orange (C). Red rows represent incorrect predictions.

| Step | Time | Event | Facility | Predicted ID | GT ID | Correct |
|---|---|---|---|---|---|---|
| 0 | 19:04:24 | OPEN | Main Door | A, B | A, B | ✓ |
| 1 | 19:04:30 | CLOSE | Main Door | A, B | A, B | ✓ |
| 2 | 19:04:35 | ON | Living Room Couch | A | A | ✓ |
| 3 | 19:04:36 | OPEN | Bedroom 1 Door | B | B | ✓ |
| 4 | 19:04:41 | OFF | Living Room Couch | A | A | ✓ |
| 5 | 19:04:42 | ON | Bathroom Global | B | A | ✗ |
| 6 | 19:04:42 | CLOSE | Bedroom 1 Door | A | B | - |
| 7 | 19:04:46 | ON | Living Room Couch | A | B | - |
| 8 | 19:04:49 | CLOSE | Toilet Contact | B | A | - |
| 9 | 19:04:52 | OFF | Living Room Couch | A | B | - |
| 10 | 19:05:00 | OFF | Bathroom Global | B | A | - |
| 11 | 19:05:26 | ON | Bathroom Global | B | A | - |
| 12 | 19:05:37 | OPEN | Toilet Contact | B | A | - |
| 13 | 19:05:44 | ON | Bathroom Sink | B | A | - |
| 14 | 19:05:52 | OFF | Bathroom Sink | B | A | - |
| 15 | 19:06:04 | OFF | Bathroom Global | B | A | - |
| 16 | 19:06:13 | OPEN | Fridge Door | B | A | - |
| 17 | 19:06:14 | CLOSE | Fridge Door | B | A | - |
| 18 | 19:07:58 | OPEN | Dining Chair 2 | B | A | - |
| 19 | 19:08:06 | ON | Living Room Couch | A | B | - |
| 20 | 19:08:10 | OFF | Living Room Couch | A | B | - |
| 21 | 19:08:16 | OPEN | Dining Chair 3 | A | B | - |
| 22 | 19:11:49 | OPEN | Main Door | C | C | ✓ |
| 23 | 19:11:53 | CLOSE | Main Door | C | C | ✓ |
| 24 | 19:12:06 | OPEN | Bedroom 2 Door | C | C | ✓ |
| 25 | 19:12:09 | CLOSE | Dining Chair 2 | B | A | - |
| 26 | 19:12:10 | CLOSE | Dining Chair 3 | A | B | - |
| 27 | 19:12:12 | CLOSE | Bedroom 2 Door | C | C | ✓ |
| 28 | 19:12:21 | ON | Kitchen Sink | B | A | - |
| 29 | 19:12:24 | ON | Kitchen Counter | B | B | ✓ |
| 30 | 19:12:43 | OFF | Kitchen Sink | B | A | ✗ |
| 31 | 19:12:51 | ON | Stove | B | A | - |
| 32 | 19:13:07 | OPEN | Bedroom 2 Door | C | C | ✓ |

Table 6.2 – Comparison of Data Association performances of baseline approaches and the proposed LLM+BC approach on 2 datasets. Acc and CRL are calculated using a window size $w = 16$.

| Dataset | MARBLE [22] | | | | MuRAL [49] | | | |
|---|---|---|---|---|---|---|---|---|
| Metric | Acc | CRL | BLEU | RCE | Acc | CRL | BLEU | RCE |
| All-to-One | 0.729 | 8.606 | 0.707 | 0.789 | 0.551 | 4.489 | 0.510 | 1.593 |
| sMRT | 0.119 | 0.641 | 0.079 | 15.10 | 0.529 | 3.321 | 0.497 | 1.560 |
| NEP+Search | 0.825 | 8.847 | 0.812 | 0.500 | 0.674 | 4.982 | 0.755 | 0.375 |
| BC | 0.792 | 8.710 | 0.797 | 0.136 | 0.728 | 5.476 | 0.774 | 0.437 |
| LADA | **0.849** | **8.944** | **0.883** | **0.045** | 0.707 | 5.337 | 0.750 | 0.702 |
| LLM+BC | 0.824 | 8.783 | 0.841 | 0.136 | **0.773** | **6.034** | **0.799** | **0.261** |

Step 4 was merely a delayed deactivation caused by the sensor's timeout mechanism, which the model failed to account for.

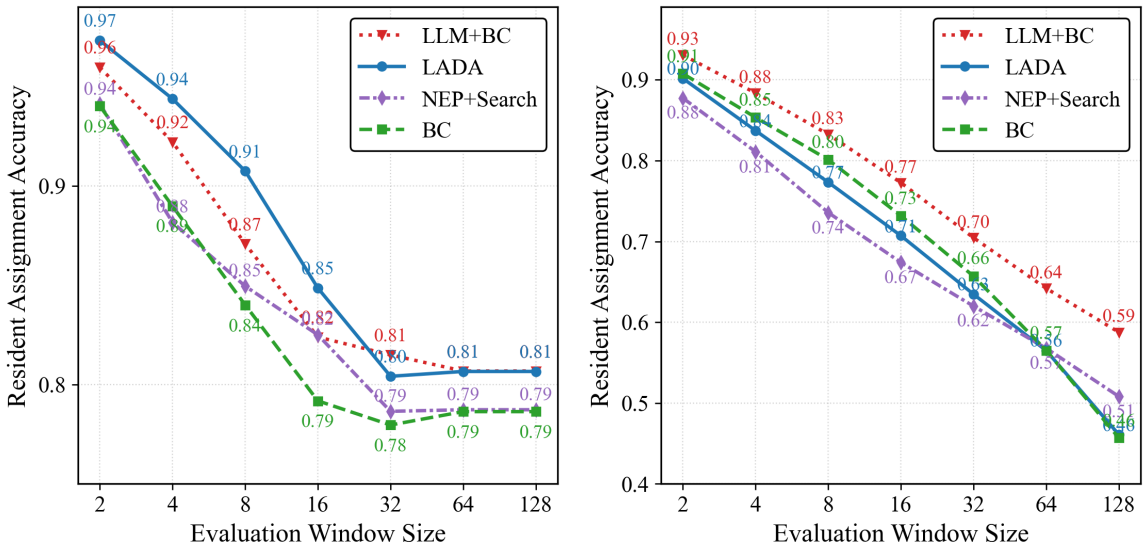
A second challenge arises during the interaction in the kitchen area (Steps 25–31), illustrating the difficulty of tracking concurrent activities in shared zones. At Steps 25 and 26, distinct sensors on Dining Chairs 2 and 3 trigger `CLOSE` events almost simultaneously, indicating that two individuals stood up to move elsewhere. However, the model fails to register this as a dual transition. When activities subsequently occur in the kitchen (Sink, Counter, and Stove at Steps 28–31), the model attributes them solely to a single resident (Predicted ID: B). This suggests that the model overlooked the cue provided by the simultaneous chair events, failing to realize that both residents had entered the kitchen. Consequently, it collapses the multi-resident scenario into a single-resident track, highlighting the complexity of distinguishing identities when residents occupy the same confined functional area.

Despite these challenges with Residents A and B, the system demonstrates robustness in handling cardinality changes. When Resident C enters the facility at Step 22, the model successfully distinguishes this new individual from the existing entangled tracks. The events associated with Resident C (Steps 22–32) are tracked with perfect accuracy, indicating that the reasoning engine remains effective for distinct, unambiguous trajectories.

### 6.3.3 Quantitative Results

Table 6.2 presents a comparative analysis of our LLM+BC approach against other methods. Notably, LLM+BC outperforms the standard BC method proposed in Chapter 4—which relies on a smaller, non-pretrained Transformer Encoder—across both the MARBLE and MuRAL datasets. This suggests that even with a parameter size of 4B, the inherent capabilities of the LLM in contextual and commonsense reasoning provide a distinct advantage for data association.

Specifically, the LLM+BC method achieves substantial success on the MuRAL



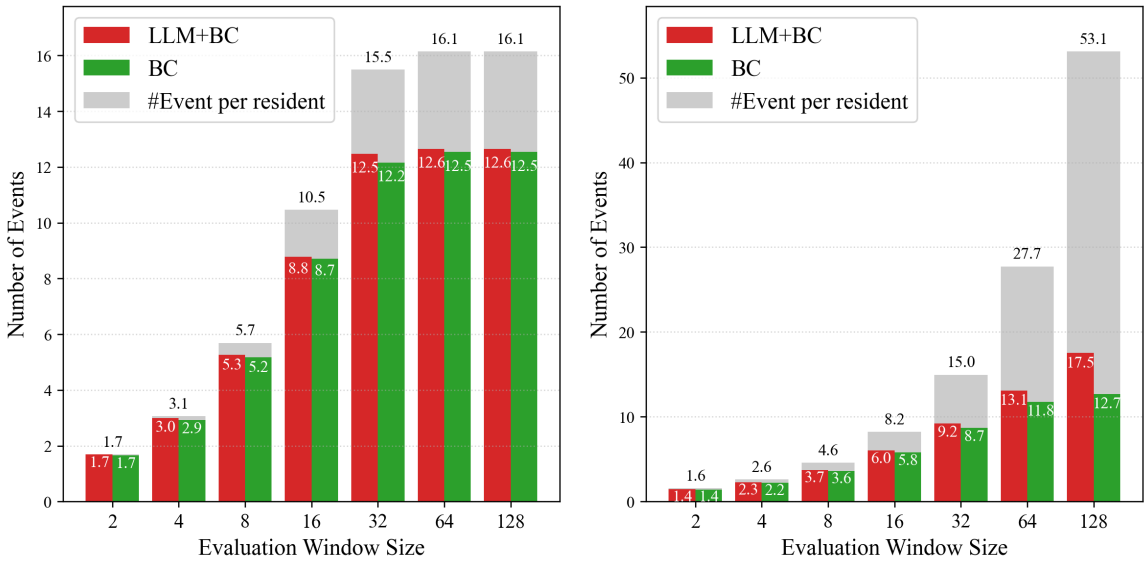

(a) MARBLE (b) MuRAL

Figure 6.3 – Resident assignment accuracy of LLM+BC approach as a function of the evaluation window size ($\log_2$ scale) on the MARBLE, and MuRAL datasets.

dataset, which is characterized by more complex data patterns. It significantly surpasses all previously proposed methods across all four evaluation metrics. Compared to LADA, which employs CoT reasoning without task-specific training, LLM+BC improves Accuracy by nearly 7%, boosts Bilingual Evaluation Understudy (BLEU) by 5%, and reduces Resident Count Error (RCE) to a mere 0.261. This empirically validates the effectiveness of the Behavior Cloning paradigm when applied to Large Language Models.

However, on the MARBLE dataset—which features clearer single-sensor semantics, simpler overall sequence patterns, and a smaller data volume—the performance dynamics differ. While LLM+BC still outperforms the standard BC baseline, it does not reach the performance level of LADA. We attribute this to two primary factors. First, LADA utilizes GPT-4o, a model with significantly larger capacity and stronger foundational capabilities than our 4B-parameter backbone. Second, the resident behaviors in MARBLE are simpler and exhibit strong mutual exclusivity between sensor events. These characteristics are easily captured by the explicit reasoning steps in CoT, thereby allowing LADA to achieve superior performance in such structurally simpler scenarios.

Figures 6.3 and 6.4 illustrate the trends of Resident Assignment Accuracy (Acc) and Correct Run Length (CRL) for the LLM+BC method with respect to the evaluation window size. As observed in Figure 6.3, although the overall performance of LLM+BC on the MARBLE dataset initially lags behind LADA, it exhibits a notably slower rate of accuracy decay, eventually surpassing LADA when $w > 32$. On the MuRAL dataset,




(a) MARBLE (b) MuRAL

Figure 6.4 – Evaluation of Correct Run Length (CRL) of LLM+BC approach and the average number of events per resident (gray bars) in different evaluation window sizes ($\log_2$ scale) for MARBLE, and MuRAL. CRL quantifies how long a method can continuously assign the correct resident before making an error.

Table 6.3 – Comparison of Data Association performances of baseline approaches and our approach on 2 datasets. Acc and CRL are calculated using a window size $w = 16$.

| Dataset | MARBLE [22] | | | | MuRAL [49] | | | |
|---|---|---|---|---|---|---|---|---|
| Metric | Acc | CRL | BLEU | RCE | Acc | CRL | BLEU | RCE |
| Trained on MuRAL | 0.765 | 8.853 | 0.760 | 0.727 | 0.773 | 6.034 | 0.799 | 0.261 |
| Trained on MARBLE | 0.824 | 8.783 | 0.841 | 0.136 | 0.695 | 5.184 | 0.753 | 0.613 |

LLM+BC demonstrates superior performance across all window sizes, accompanied by an even more gradual decline. This slower performance degradation compared to both standard BC and LADA suggests that the combination of task-specific fine-tuning and the LLM's inherent reasoning capabilities synergistically enhances the method's robustness against error accumulation. Furthermore, the comparison of CRL between LLM+BC and BC in Figure 6.4 serves to further corroborate the distinct advantages that the Large Language Model brings to the Data Association task.

### 6.3.4 Cross-Environment Evaluation

To evaluate the generalization capability of our proposed framework across different environments, we conducted a cross-evaluation experiment as detailed in Table 6.3. It is important to note that for these experiments, the system prompts were dynamically adjusted to match the target dataset's specific resident nomenclature and

sensor topology, while the model weights remained fixed.

The results reveal a significant asymmetry in transferability. The model trained on the simpler MARBLE dataset demonstrates robust generalization when applied to the more complex MuRAL dataset. Its performance is comparable to that of the LADA method, which similarly operates without MuRAL-specific training. This suggests that the fundamental association logic learned from the high-contrast patterns in MARBLE is effectively transferable to more chaotic environments. Conversely, the model trained on the complex MuRAL dataset fails to generalize to the simpler MARBLE environment. Both the quantitative metrics and qualitative observations indicate that the model's behavior degenerated into a naive "All-to-one" baseline. Specifically, despite an Accuracy of 0.765, the exceptionally high RCE (0.727) reveals that the model indiscriminately assigned the vast majority of events to a single resident identity. This suggests that the model, having overfitted to the intricate distributions of MuRAL, lost the ability to correctly process the straightforward, mutually exclusive patterns present in MARBLE.

## 6.4 Conclusion

In this chapter, we introduced the LLM+BC framework, a hybrid methodology that synthesizes the semantic understanding of LLMs with the statistical rigor of supervised BC. By transitioning from the zero-shot reasoning paradigm of LADA to a fine-tuning approach, we effectively reformulated multi-resident data association as a supervised conditional text generation task. Methodologically, this involved stripping away intermediate CoT reasoning steps to enforce a direct state-to-action mapping, optimized via a standard causal language modeling objective. Crucially, by employing LoRA, we demonstrated that smaller, open-source models (e.g., Qwen3-4B) can be efficiently adapted to learn domain-specific policies without the prohibitive computational costs of full-parameter training.

Our empirical evaluations validate the efficacy of this approach. The LLM+BC framework not only consistently outperformed traditional encoder-based baselines, confirming the value of pre-trained linguistic knowledge, but also surpassed the zero-shot LADA method in high-entropy environments characterized by sensor noise and complex interleaving behaviors. While zero-shot reasoning remains superior in semantically unambiguous scenarios, our results indicate that supervised fine-tuning offers significantly greater robustness against error accumulation and superior adaptability to intricate sensor distributions. Furthermore, the successful deployment of a 4B-parameter model highlights the practical feasibility of privacy-preserving, low-latency Edge AI for smart homes, bridging the gap between hyperscale generative capabilities and local real-time execution.

# CHAPTER 7

# CONCLUSIONS AND PERSPECTIVES

In this concluding chapter, we summarize the primary contributions and key findings presented throughout this manuscript. Furthermore, we critically examine the limitations inherent to our work and outline potential avenues for future research to address these challenges and extend the current scope.

## 7.1 Conclusions

Our research focused on addressing the challenge of ambient sensor data association in multi-resident environments, with a commitment to data-driven, scalable, and accurate solutions. We achieved this through three key innovations: (1) a bottom-up modeling paradigm for data association; (2) optimizations tailored for data-driven performance; and (3) the incorporation of Pre-trained Large Language Models (LLMs) to enable zero-shot unsupervised capabilities. These contributions are detailed below.

### 7.1.1 Data Association Problem Definition and Modeling

Our first contribution is the formal mathematical definition of the Data Association task, upon which we established a comprehensive Markov Decision Process (MDP)

modeling framework. This lays a solid theoretical foundation for the subsequent contributions of this thesis.

Literature review shows that previous definitions of Data Association were largely confined to phenomenological descriptions, lacking a rigorous and unified mathematical formulation. To address this, in Chapter 2, we formalized the Data Association task as a mapping function that projects events from a sensor stream to a subset of the resident identity pool. Because the cardinality of this subset can exceed one, this definition inherently accommodates complex scenarios where a single event is triggered simultaneously by multiple residents. Furthermore, by including a "new" identity within the pool, our definition provides native support for the dynamic addition of known residents in multi-user environments.

Moreover, previous research [204, 197, 198] indicates that effective Data Association methods are predominantly modeled based on Filtering. However, filtering relies on continuous data, whereas ambient sensor events are fundamentally discrete and heterogeneous. This discrepancy forces suboptimal adaptations in traditional approaches. For instance, sMRT [197] maps discrete events into a continuous latent space via word embeddings to apply filtering. However, this approach constrains the system state to a latent vector representation, sacrificing contextual interpretability and limiting the model to the expressive bottleneck of first-order Markov assumptions. Additionally, filtering algorithms typically assume linear motion in a continuous latent space, which presents an inherent misalignment with discrete event transitions driven by logic and behavioral habits. These issues restrict the scalability of data-driven alignment.

In response, drawing inspiration from both our formal definition and the limitations of prior filtering algorithms, we proposed an MDP-based modeling paradigm. Unlike filtering, our MDP approach treats sensor events as discrete data points, allowing for the stacking of infinite-length context within the system state. This breaks the limitations of traditional first-order Markov properties and significantly enriches the modeling of the system state. Experimental results from multiple methods derived from this MDP framework have consistently outperformed the traditional filtering-based sMRT baseline. This validates the fundamental soundness and robustness of our proposed problem definition and MDP modeling paradigm.

### 7.1.2 Data-driven Policy Design and Training

In our first contribution regarding MDP modeling, we formulated the Data Association solution process as a policy execution of assignment actions. Building upon this conceptualization, our second contribution involves designing and training policy models that are fully data-driven.

In Chapter 3, we proposed a policy based on reward-guided search, named NEP+Search. We employ a Next-Event Prediction (NEP) model to calculate the

likelihood of the current event belonging to each resident based on their respective assignment history. For potential new residents, we derive the likelihood using parameterized priors and the likelihoods of known residents. Subsequently, our method constructs a reward function for assigning events to a specific subset of residents based on these likelihoods. Guided by these rewards, the agent in the MDP framework searches for the assignment that maximizes the reward, thereby serving as the optimal policy. Extensive experiments across three datasets demonstrate that NEP+Search significantly outperforms sMRT on all four designed metrics. This not only validates the effectiveness of our first contribution—the MDP modeling—but also confirms that NEP+Search is an effective policy modeling approach.

However, the NEP+Search strategy considers each resident's history independently, failing to holistically consider the joint states of all residents. Furthermore, it predicts the probability of the next event rather than directly predicting resident assignments or the creation of new residents. These factors limit the model's ability to fully capture the underlying data distribution. To address this, in Chapter 4, we proposed modeling the entire policy directly with a deep neural network, moving beyond the indirect next-event generation of NEP+Search. We trained this network using Behavior Cloning (BC) from imitation learning; hence, we refer to this method as BC. The BC model accepts the historical context of all residents simultaneously and directly outputs the corresponding resident indices. Specifically, we employed a Transformer encoder as the backbone language model architecture and utilized Binary Cross-Entropy (BCE) loss in a multi-label classification framework, treating each possible resident index as a binary classification label. To ensure permutation invariance regarding resident indices, we introduced a permutation-based data augmentation method to construct the training dataset. Final experiments indicate that the BC method outperforms NEP+Search on the CASAS and MuRAL datasets and performs comparably on MARBLE, demonstrating its effectiveness. Notably, we observed that BC handles events involving multiple residents more effectively than NEP+Search. Additionally, our ablation studies verified the efficacy of the permutation-based data augmentation.

The evolution from the probability-based NEP+Search to the holistic BC approach underscores the versatility and efficacy of our data-driven policy framework. By effectively learning from sensor patterns, these methods not only resolve the structural limitations of traditional filtering but also provide accurate, scalable solutions for complex multi-resident tracking tasks, setting a new benchmark for supervised Data Association

### 7.1.3 Large Language Models as a General Policy

Despite the high accuracy achieved by the data-driven policies in our second contribution, they are inherently constrained by the distributions of their training data. This dependency severely restricts their generalizability to novel or unfamiliar

environments, as the learned policies often fail to transfer to new spatial layouts or resident behaviors without retraining. To address this limitation, our third contribution shifts the paradigm from supervised specific policies to a generalized reasoning agent by deploying LLMs as a General Policy. By leveraging the inherent semantic knowledge and reasoning powers of LLMs, we demonstrate that this general policy can achieve robust event allocation in unseen settings without requiring environment-specific training.

In Chapter 5, we propose the LADA method, which fundamentally reformulates the decision-making policy within the MDP framework. Diverging from the discriminative classification strategies used in previous NEP+Search and BC approaches, LADA defines a text-conditioned generative policy. By directly deploying pre-trained LLMs as the core agent, we leverage their inherent capabilities for contextual reasoning and Chain-of-Thought (CoT) derivation. We condition the LLM using customized prompts that encapsulate environmental context, enabling it to perform commonsense reasoning and directly generate decisions in a structured JSON format. Empirical results demonstrate that LADA achieves zero-shot, unsupervised performance comparable to supervised baselines on both the MuRAL and MARBLE datasets, with the GPT-4o model achieving state-of-the-art results on MARBLE. However, a comparative analysis reveals that the effectiveness of LADA diminishes significantly on smaller-scale LLMs (fewer than 32B parameters), which lack the emergent reasoning capabilities of larger models.

To unlock high performance in smaller LLMs and to transition from pure common sense reasoning to data-driven pattern recognition in complex environments, Chapter 6 introduces the LLM+BC method. Building upon the LADA framework, we apply the BC training paradigm to fine-tune the Qwen3-4B model using Low-Rank Adaptation (LoRA). We observe that the fine-tuned LLM achieves substantial improvements on the MuRAL dataset—which is characterized by complex, data-dependent patterns—outperforming all other methods. Furthermore, LLM+BC significantly enhances the performance of 4B-parameter models across both datasets compared to the zero-shot LADA baseline. These findings validate the efficacy of fine-tuning for adapting LLMs to the Data Association task and demonstrate the robustness of the BC paradigm when applied to generative model architectures.

## 7.2 Limitations and Perspectives

Despite the significant contributions of our research, they are primarily foundational, establishing a modeling perspective alongside paradigms for training and inference. To achieve robust deployment in real-world scenarios, certain limitations remain that necessitate further exploration.

### 7.2.1 Distribution Shift and Compounding Errors

In our second contribution, we employed BC to learn the policy using expert trajectories derived from ground-truth labels. While effective, this supervised approach relies on the assumption that the training and testing data are independent and identically distributed (i.i.d.). However, in sequential decision-making tasks like ours, this assumption is fundamentally violated, showing a phenomenon known as covariate shift. Because the training dataset consists exclusively of "perfect" expert trajectories, the policy learns to act only based on correct historical contexts. During inference, however, the agent will inevitably make prediction errors. In the context of Data Association, assigning an event to the wrong resident corrupts the system state (i.e., the residents' history). Since the model was never exposed to such "corrupted" states during training, it lacks the capability to recover or correct itself. This leads to compounding errors, where a single misallocation propagates through the sequence, causing the agent to drift further from the correct trajectory and potentially resulting in a collapse of tracking performance.

To overcome this limitation, a promising perspective is to implement Dataset Aggregation (DAgger) [160]. Unlike static BC, DAgger allows the learner policy to execute assignment actions during training, inevitably generating incorrect assignments that lead to deviant system states (e.g., a resident's history incorrectly indicating they are in the kitchen). Instead of correcting the state immediately, DAgger queries the ground-truth expert to provide the correct assignment action given this corrupted history. By aggregating these experiences, the policy learns specifically how to recover from tracking errors, effectively training the agent to re-associate events correctly even after previous misallocations, thereby ensuring long-term robustness.

We conducted preliminary experiments implementing a DAgger-based approach. We constructed training datasets via a dynamic sampling strategy that permitted a gradually increasing proportion of historical errors to exist within the input sequences. Our initial results indicated that while DAgger demonstrated a marginal advantage over the BC baseline under specific parameter configurations, the overall optimal performance remained statistically comparable to that of BC. This suggests that simply exposing the model to random historical errors may be insufficient to fully resolve the distribution shift. Future work should therefore investigate more sophisticated error-sampling strategies or refined curriculum learning schedules to effectively maximize the potential of DAgger in complex Data Association tasks.

### 7.2.2 Life-cycle Management and State Re-initialization

While DAgger aims to enhance the policy's intrinsic capability to recover from errors, a complementary perspective focuses on the system-level management of the tracking life-cycle. In classical Multi-Target Tracking (MTT) theory [33], a track is not static but undergoes distinct phases: initiation, maintenance, and termination. Our current framework maintains resident states indefinitely, creating a vulnerability

where a single accumulation of a “catastrophic error” can permanently diverge the system state from reality—a phenomenon analogous to the “kidnapped robot problem” in localization [185].

To mitigate this, future work could incorporate an active “Error-Triggered Re-initialization” mechanism. This approach relies on monitoring the agent’s uncertainty during inference. By leveraging metrics such as Predictive Entropy or Maximum Softmax Probability [95], the system can detect Out-of-Distribution (OOD) states where the current sensor observations fundamentally contradict the maintained history (e.g., physically impossible transitions). Upon detecting such divergence, the system would enforce a “track termination”, clearing the corrupted context buffers and re-initializing the resident’s state as a blank slate.

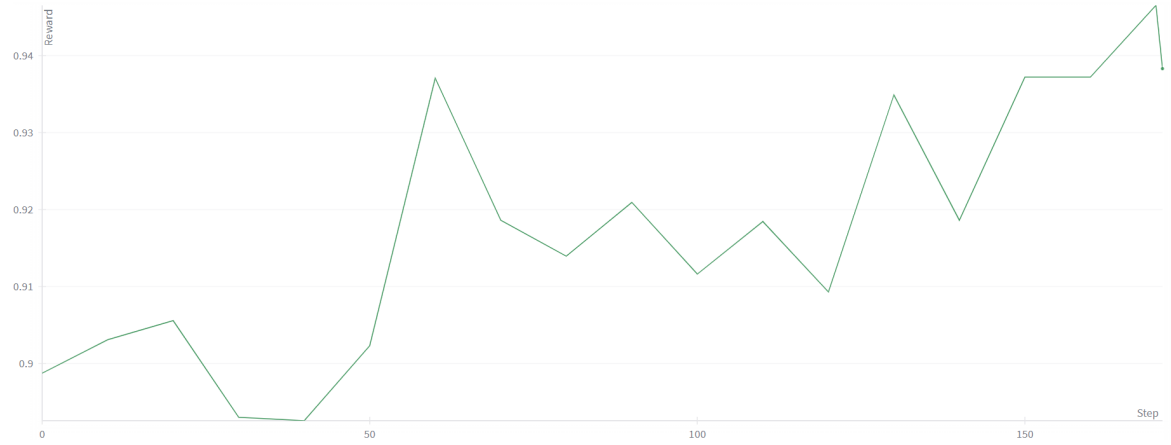
Although this "hard reset" effectively truncates the propagation of errors, it introduces the challenge of “Track Fragmentation” (or Identity Switches). The system must find a trade-off between maintaining long-term context and preventing error accumulation. Therefore, future research should explore “Soft Reset” strategies, i.e. selectively pruning low-confidence historical segments while retaining high-confidence static attributes, to minimize the "cold-start" period required to rebuild the semantic context.

A concrete and low-risk implementation of this strategy is “Opportunistic State Alignment”. For instance, when the system infers with high confidence that the environment is vacant (e.g., after observing “Leave Home” events for all residents), a complete state reset can be executed without penalty. In this specific scenario, the re-initialized “blank” internal state achieves a zero-error alignment with the physical ground truth (i.e., an empty house). This effectively bypasses the cold-start risk, as the erasure of historical context is semantically consistent with the absence of residents, serving as a natural synchronization point to eliminate accumulated drift.

### 7.2.3 Integration of Explicit Reasoning and Supervised Fine-tuning

Our third contribution explored two distinct paradigms for deploying LLMs: LADA, which leverages zero-shot CoT reasoning, and LLM+BC, which applies supervised fine-tuning. Our experiments revealed a performance dichotomy: LADA excels in semantically clear environments like MARBLE by exploiting explicit reasoning, whereas LLM+BC achieves state-of-the-art performance in complex, noisy environments like MuRAL by fitting specific data distributions.

However, a critical limitation of the current LLM+BC approach is the loss of the explicit reasoning capability observed in LADA. We argue that maintaining CoT capability is essential not just for interpretability, but for enabling unsupervised generalization to novel environments. While shallow pattern matching suffices for the training domain, explicit reasoning allows the model to ground decisions in logical deduction, which is robust to domain shifts. To bridge the gap between LADA’s




(a) Validation Reward


(b) Generation Entropy

Figure 7.1 – Preliminary training curves of GRPO on the MARBLE dataset.

generalization potential and LLM+BC's performance, future work should focus on Reasoning-Aware Fine-tuning through two advanced paradigms:

(1) Rejection Sampling Fine-Tuning (RFT): To synthesize reasoning data, we can employ a Rejection Sampling strategy [218, 216]. In this approach, the Zero-shot LADA model acts as a generator to produce multiple candidate reasoning chains for each training sample. We then filter these trajectories, retaining only those that lead to the correct ground-truth assignment. Fine-tuning the model on these "correct-reasoning" samples effectively distills the reasoning capability into the policy, enabling it to handle complex noise while retaining logical coherence.

(2) Outcome-Supervised Reinforcement Learning: Alternatively, we can utilize Reinforcement Learning (RL) to incentivize the model to autonomously discover effective reasoning patterns. Since the intermediate rationale is unsupervised but the final assignment can be verified, the ground-truth label serves as a sparse reward signal. Algorithms such as Proximal Policy Optimization (PPO) [164] or the more

efficient Group Relative Policy Optimization (GRPO) [92] can be employed to optimize the policy. This "outcome-supervised" approach encourages the emergence of long-chain CoT reasoning that is strictly aligned with task accuracy, effectively teaching the model to "think" within the domain constraints. To validate the feasibility of this direction, we conducted a preliminary pilot study applying the GRPO algorithm on the MARBLE dataset, utilizing `Qwen3-4B-Instruct-2507` as the base policy model. In this setup, we configured the group size to 4, allowing the policy to sample four distinct reasoning trajectories per input. We utilized a strict outcome-based sparse reward mechanism, where a reward of 1 is granted solely if the final prediction matches the Ground Truth label. As shown in Figure 7.1, initial training logs of a single epoch are highly encouraging: the validation reward (Figure 7.1a) demonstrated a robust upward trend, peaking at approximately 0.95, while the policy generation entropy (Figure 7.1b) maintained a stable level around 0.5. These results confirm that even with sparse signals and limited sampling, the model can successfully align its reasoning process with the correct outcomes while preserving sufficient exploration.

### 7.2.4 Transition from Online Filtering to Non-Myopic Smoothing

A fundamental limitation shared by our current MDP and BC frameworks is their strictly causal and greedy nature—effectively operating as online filters with greedy decoding. At each time step, the agent makes an irreversible assignment decision based solely on the history up to that point and immediately commits to the single highest-probability option. However, in multi-resident environments, such ambiguity is often resolvable only through future observations. For instance, as shown in Table 6.1, distinguishing whether the rapid sequence of kitchen activations at Steps 28–31 originates from a single resident's workflow or two co-located individuals effectively requires observing their subsequent divergence. Our current myopic approach precludes the use of such future evidence, leading to locally optimal but globally inconsistent trajectories.

To overcome this, future research should shift the paradigm from instantaneous decision-making to non-myopic global optimization, integrating fixed-lag smoothing [19] with inference-time search. First, we propose expanding the decision horizon by incorporating multi-turn Reinforcement Learning strategies that allow for deferred decision making or retrospective correction. Specifically, the action space could include a *Wait* action to buffer ambiguous events, allowing the system to revisit and revise assignments within a fixed temporal window. Second, to support this delayed commitment, we can extend the BC policy from a deterministic actor to a probabilistic evaluator. Drawing inspiration from the beam search success in our first contribution (NEP+Search), we can maintain $k$ most promising assignment trajectories simultaneously using Beam Search. By combining the temporal flexibility of smoothing with the probabilistic exploration of search, the system can effectively leverage future context to resolve present uncertainties, ensuring long-term tracking consistency.

### 7.2.5 Scalable Data Generation via Embodied AI Simulation

A persistent bottleneck in multi-resident research is the scarcity of large-scale, annotated real-world datasets. Collecting high-quality sensor data from physical homes is costly, privacy-intrusive, and difficult to scale. Moreover, existing datasets often lack diversity in spatial layouts and resident behaviors, limiting the generalization capability of our models. To overcome this data scarcity, future work should leverage the recent advancements in generative multi-agent systems and Embodied AI simulators to construct high-fidelity, multi-resident virtual environments.

Historically, the Human Activity Recognition (HAR) community has relied on specialized simulators like OpenSHS [18] and SIMACT [37]. However, these tools are predominantly tailored for single-resident scenarios. Even when adapted for multiple users, they operate primarily via parallel script execution, fundamentally lacking the physical interaction engines required to model authentic multi-resident dynamics. Critical spatial constraints—such as collision avoidance, resource contention (e.g., one resident waiting for the bathroom), and joint activities—are often absent. Consequently, the generated data fails to capture the intricate spatial correlations essential for training robust Data Association models.

To bridge this gap, future work should converge three distinct lines of research: the data generation methodology pioneered in early smart home simulations, the behavioral realism of modern generative agents, and the physical fidelity of Embodied AI. Early works such as Synnott et al. [177] conceptually proposed using multi-agent systems to synthetically populate smart home datasets. However, these agents were typically governed by rigid, rule-based scripts that failed to capture the fluidity of human interaction. A paradigm shift occurred recently with the generative agents framework introduced by Park et al. [140], which demonstrated that LLM-driven agents can exhibit complex, emergent social behaviors—such as organizing parties or sharing daily routines—in a 2D sandbox. A prospective would be lifting these generative agents from 2D sandboxes into high-fidelity 3D Embodied AI simulators like VirtualHome [146] or AI2-THOR [109]. By equipping the physics-aware avatars in these simulators with the cognitive architectures of Generative Agents, we can simulate authentic multi-resident scenarios where agents not only navigate physical constraints (e.g., collision avoidance) but also engage in meaningful spatiotemporal interactions (e.g., queuing for the bathroom, cooking together). This fusion has the potential to generate synthetic datasets that closely resemble real-world scenarios in both physical signal patterns and high-level behavioral logic.

### 7.2.6 Ethical Implications and Safety Risks

Finally, we must critically address the ethical implications and potential risks inherent in deploying our proposed systems. Although our methods rely on non-intrusive binary sensors rather than cameras, the high-precision tracking achieved by our MDP and LLM policies inevitably generates granular profiles of residents' daily

lives. Such detailed digital footprints could reveal sensitive information—ranging from health conditions (e.g., bathroom frequency) to personal habits—raising significant privacy concerns if processed centrally. To mitigate this, future deployment should transition from centralized cloud processing to Privacy-Preserving Edge Computing. Specifically, integrating Federated Learning (FL) [124] would allow local models to update gradients on edge devices (e.g., home gateways) without ever transmitting raw sensor logs to a central server. Furthermore, applying Differential Privacy (DP) [81] techniques could add statistical noise to the updates, mathematically guaranteeing that individual behavioral patterns cannot be reverse-engineered.

A second critical risk stems from the probabilistic nature of LLMs. In our LADA and LLM+BC frameworks, there exists a risk of hallucination or confabulation, where the model might confidently attribute an event to the wrong resident or invent interactions that never occurred. In healthcare or elderly care scenarios, such false attribution could lead to missed emergency alerts or incorrect medical diagnoses. Therefore, developing Trustworthy AI guardrails is imperative. Future work should implement rigorous Uncertainty Quantification mechanisms to suppress model outputs when confidence is low, and incorporate Human-in-the-Loop protocols where high-stakes decisions (e.g., detecting a fall or anomaly) require verification, ensuring the system serves as a supportive tool rather than an autonomous arbiter.

A third ethical dimension concerns environmental sustainability and the carbon footprint of AI. LLMs are notoriously energy-intensive. While much attention is paid to the training costs, in a real-time smart home system that processes continuous sensor streams 24/7, the cumulative inference-time energy consumption becomes the dominant environmental burden [142]. Deploying a massive general-purpose LLM (e.g., 70B parameters) to handle simple daily events is computationally inefficient and environmentally irresponsible. To address this, future work must adhere to the principles of Green AI [165]. We propose shifting focus from seeking pure accuracy improvements via larger models to optimizing efficiency. Techniques such as Model Distillation [97] and Quantization (e.g., 4-bit loading) should be employed to compress the heavy teacher LLM into lightweight student models or Small Language Models (SLMs). These compact models can operate efficiently on low-power edge devices (e.g., Raspberry Pi) rather than energy-hungry GPU clusters, thereby reconciling high-performance intelligence with ecological responsibility.

# BIBLIOGRAPHY


[1] Giovanni Acampora, Diane J Cook, Parisa Rashidi, and Athanasios V Vasilakos. A survey on ambient intelligence in healthcare. *Proceedings of the IEEE*, 101(12):2470–2494, 2013. 2, 21

[2] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 90, 94

[3] Insaf Adjabi, Abdeldjalil Ouahabi, Amir Benzaoui, and Abdelmalik Taleb-Ahmed. Past, present, and future of face recognition: A review. *Electronics*, 9(8):1188, 2020. 15

[4] Md Afroz, Emmanuel Nyakwende, and Birendra Goswami. Intrusion detection in smart home environments: A machine learning approach. *Transportation Research Procedia*, 84:243–250, 2025. 2, 21

[5] Armen Aghajanyan, Luke Zettlemoyer, and Sishir Gupta. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics*, pages 7319–7328, 2021. 121

[6] Adeel Ahmad, June Chul Roh, Dan Wang, and Aish Dubey. Vital signs monitoring of multiple people using a fmcw millimeter-wave sensor. In *2018 IEEE Radar Conference (RadarConf18)*, pages 1450–1455. IEEE, 2018. 21

[7] Simin Ahmadi-Karvigh, Ali Ghahramani, Burcin Becerik-Gerber, and Lucio Soibelman. Real-time activity recognition for energy efficiency in buildings. *Applied energy*, 211:146–160, 2018. 2, 21

[8] Joshua Ainslie et al. Gqa: Training generalized multi-query transformer models from multi-head checkpoints. *arXiv preprint arXiv:2305.13245*, 2023. 93

[9] Ahsan Jamal Akbar, Zhiyao Sheng, Qian Zhang, and Dong Wang. Cross-domain gesture sequence recognition for two-player exergames using cots mmwave radar. *Proceedings of the ACM on Human-Computer Interaction*, 7(ISS):327–356, 2023. 21

[10] Ahmad Akl, Jasper Snoek, and Alex Mihailidis. Unobtrusive detection of mild cognitive impairment in older adults through home monitoring. *IEEE journal of biomedical and health informatics*, 21(2):339–348, 2015. 2, 21

[11] Aparna Akula, Anuj K Shah, and Ripul Ghosh. Deep learning approach for human action recognition in infrared images. *Cognitive Systems Research*, 50:146–154, 2018. 3

[12] Mohammad Arif Ul Alam, Fernando Mazzoni, Md Mahmudur Rahman, and Jared Widberg. Lamar: Lidar based multi-inhabitant activity recognition. In *MobiQuitous 2020-17th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*, pages 1–9, 2020. 3, 19, 20

[13] Mohammad Arif Ul Alam, Md Mahmudur Rahman, and Jared Q Widberg. Palmar: Towards adaptive multi-inhabitant activity recognition in point-cloud technology. In *IEEE INFOCOM 2021-IEEE Conference on Computer Communications*, pages 1–10. IEEE, 2021. 3, 19, 20

[14] Mohammad Arif Ul Alam, Nirmalya Roy, and Archan Misra. Tracking and behavior augmented activity recognition for multiple inhabitants. *IEEE Transactions on Mobile Computing*, 20(1):247–262, 2019. 19

[15] Hande Alemdar and Cem Ersoy. Multi-resident activity tracking and recognition in smart environments. *Journal of Ambient Intelligence and Humanized Computing*, 8(4):513–529, 2017. 28

[16] Hande Alemdar, Halil Ertan, Ozlem Durmaz Incel, and Cem Ersoy. Aras human activity datasets in multiple homes with multiple residents. In *2013 7th International Conference on Pervasive Computing Technologies for Healthcare and Workshops*, pages 232–235. IEEE, 2013. 31, 32, 91

[17] Alaa Alhamoud, Vaidehi Muradi, Doreen Böhnstedt, and Ralf Steinmetz. Activity recognition in multi-user environments using techniques of multi-label classification. In *Proceedings of the 6th International Conference on the Internet of Things*, pages 15–23, 2016. 19, 29, 30

[18] N. Alshammari, T. Alshammari, M. Sedky, J. Champion, and C. Bauer. Openshs: Open smart home simulator for dataset generation. In *2017 IEEE International Conference on Smart Computing (SMARTCOMP)*, pages 1–6, 2017. 139

[19] Brian DO Anderson and John B Moore. *Optimal filtering*. Prentice-hall Englewood Cliffs, 1979. 138

[20] Rohan Anil, Andrew M Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, et al. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*, 2023. 94

[21] Ane Alberdi Aramendi, Alyssa Weakley, Asier Aztiria Goenaga, Maureen Schmitter-Edgecombe, and Diane J Cook. Automatic assessment of functional

health decline in older adults based on smart home data. *Journal of biomedical informatics*, 81:119–130, 2018. 2, 21

[22] Luca Arrotta, Claudio Bettini, and Gabriele Civitarese. The marble dataset: Multi-inhabitant activities of daily living combining wearable and environmental sensors data. In *International Conference on Mobile and Ubiquitous Systems: Computing, Networking, and Services*, pages 451–468. Springer, 2021. 31, 32, 33, 39, 54, 55, 56, 57, 60, 61, 64, 77, 78, 106, 110, 126, 128

[23] Luca Arrotta, Claudio Bettini, and Gabriele Civitarese. Micar: multi-inhabitant context-aware activity recognition in home environments. *Distributed and Parallel Databases*, pages 1–32, 2022. 19

[24] Luca Arrotta, Claudio Bettini, Gabriele Civitarese, and Riccardo Presotto. Context-aware data association for multi-inhabitant sensor-based activity recognition. In *2020 21st IEEE International Conference on Mobile Data Management (MDM)*, pages 125–130. IEEE, 2020. 19

[25] Luca Arrotta, Gabriele Civitarese, Xi Chen, Julien Cumin, and Claudio Bettini. Multi-subject human activities: A survey of recognition and evaluation methods based on a formal framework. *Expert Systems with Applications*, page 126178, 2024. 5, 15, 19, 23, 37

[26] Muhammad Haseeb Arshad, Muhammad Bilal, and Abdullah Gani. Human activity recognition: Review, taxonomy and open challenges. *Sensors*, 22(17):6463, 2022. 3, 21

[27] Johanna Austin, Hiroko H Dodge, Thomas Riley, Peter G Jacobs, Stephen Thielke, and Jeffrey Kaye. A smart-home system to unobtrusively and continuously assess loneliness in older adults. *IEEE journal of translational engineering in health and medicine*, 4:1–11, 2016. 2, 21

[28] Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022. 90, 94

[29] Ling Bao and Stephen S. Intille. Activity recognition from user-annotated acceleration data. In *Pervasive Computing: Second International Conference, PERVASIVE 2004, Linz/Vienna, Austria, April 21-23, 2004. Proceedings*, pages 1–17, Berlin, Heidelberg, 2004. Springer. 4, 22

[30] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155, 2003. 120

[31] Asma Benmansour, Abdelhamid Bouchachia, and Mohammed Feham. Multi-occupant activity recognition in pervasive smart home environments. *ACM Computing Surveys (CSUR)*, 48(3):1–36, 2015. 5, 12, 13, 15, 23, 37

[32] Asma Benmansour, Abdelhamid Bouchachia, and Mohammed Feham. Modeling interaction in multi-resident activities. *Neurocomputing*, 230:133–142, 2017. 19, 28, 29, 30

[33] Samuel S Blackman and Robert Popoli. *Design and analysis of modern tracking systems*. Artech House, 1999. 135

[34] Damien Bouchabou, Christophe Lohr, Ioannis Kanellos, and Sao Mai Nguyen. Human activity recognition (har) in smart homes. *arXiv preprint arXiv:2112.11232*, 2021. xvii, 4, 22, 23

[35] Damien Bouchabou, Sao Mai Nguyen, Christophe Lohr, Benoit Leduc, and Ioannis Kanellos. Fully convolutional network bootstrapped by word encoding and embedding for activity recognition in smart homes. In *Deep Learning for Human Activity Recognition: Second International Workshop, DL-HAR 2020, Held in Conjunction with IJCAI-PRICAI 2020, Kyoto, Japan, January 8, 2021, Proceedings 2*, pages 111–125. Springer, 2021. 4, 23, 26

[36] Damien Bouchabou, Sao Mai Nguyen, Christophe Lohr, Benoit LeDuc, and Ioannis Kanellos. A survey of human activity recognition in smart homes based on iot sensors algorithms: Taxonomies, challenges, and opportunities with deep learning. *Sensors*, 21(18):6037, 2021. 4

[37] Bruno Bouchard, Abdenour Bouzouane, and Sylvain Giroux. Simact: a 3d open source smart home simulator for activity recognition. In *2010 IEEE/ACM International Conference on Green Computing and Communications and Int'l Conference on Cyber, Physical and Social Computing*, pages 722–727, 2010. 139

[38] Tom B. Brown et al. Language models are few-shot learners. *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 90, 91, 92, 94

[39] Andrew Chan, Joanne Cai, Linna Qian, Brendan Coutts, Steven Phan, Geoff Gregson, Michael Lipsett, Adriana M Ríos Rincón, et al. In-home positioning for remote home health monitoring in older adults: Systematic review. *JMIR aging*, 7(1):e57320, 2024. 2, 21

[40] Marie Chan, Daniel Estève, Christophe Escriba, and Eric Campo. A review of smart homes—present state and future challenges. *Computer methods and programs in biomedicine*, 91(1):55–81, 2008. 2

[41] Dong Chen, Sira Yongchareon, Edmund M-K Lai, Quan Z Sheng, and Veronica Liesaputra. Locally weighted ensemble-detection-based adaptive random forest classifier for sensor-based online activity recognition for multiple residents. *IEEE Internet of Things Journal*, 9(15):13077–13085, 2021. 19, 29, 30

[42] Dong Chen, Sira Yongchareon, Edmund M-K Lai, Jian Yu, and Quan Z Sheng. Hybrid fuzzy c-means cpd-based segmentation for improving sensor-based multiresident activity recognition. *IEEE Internet of Things Journal*, 8(14):11193–11207, 2021. 19, 29, 30

[43] Dong Chen, Sira Yongchareon, Edmund M-K Lai, Jian Yu, Quan Z Sheng, and Yafeng Li. Transformer with bidirectional gru for nonintrusive, sensor-based activity recognition in a multiresident environment. *IEEE Internet of Things Journal*, 9(23):23716–23727, 2022. 29, 30

[44] Hao Chen, Seung Hyun Cha, and Tae Wan Kim. A framework for group activity detection and recognition using smartphone sensors and beacons. *Building and Environment*, 158:205–216, 2019. 19

[45] Liming Chen and Chris Nugent. Ontology-based activity recognition in intelligent pervasive environments. *International Journal of Web Information Systems*, 5(4):410–430, 2009. 4, 22

[46] Rong Chen and Yu Tong. A two-stage method for solving multi-resident activity recognition in smart environments. *Entropy*, 16(4):2184–2203, 2014. 5, 15, 19, 23, 29, 30, 37

[47] Xi Chen, Julien Cumin, Fano Ramparany, and Dominique Vaufreydaz. Generative resident separation and multi-label classification for multi-person activity recognition. In *2024 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops)*, pages 1–6. IEEE, 2024. 5, 23, 37

[48] Xi Chen, Julien Cumin, Fano Ramparany, and Dominique Vaufreydaz. Towards LLM-powered ambient sensor based multi-person human activity recognition. In *2024 IEEE 30th International Conference on Parallel and Distributed Systems (ICPADS)*, pages 609–616. IEEE Computer Society, 2024. 15, 32, 52

[49] Xi Chen, Julien Cumin, Fano Ramparany, and Dominique Vaufreydaz. MuRAL: A Multi-Resident Ambient Sensor Dataset Annotated with Natural Language for Activities of Daily Living. *HAL preprint hal-05048859*, 2025. xvii, 31, 32, 33, 34, 39, 54, 55, 57, 60, 61, 64, 77, 78, 106, 110, 126, 128

[50] Yuheng Cheng, Ceyao Zhang, Zhengwen Zhang, Xiangrui Meng, Sirui Hong, Wenhao Li, Zihao Wang, Zekai Wang, Feng Yin, Junhua Zhao, et al. Exploring large language model based intelligent agents: Definitions, methods, and prospects. *arXiv preprint arXiv:2401.03428*, 2024. 95, 96

[51] Yi-Ting Chiang, Kuo-Chung Hsu, Ching-Hu Lu, Li-Chen Fu, and Jane Yung-Jen Hsu. Interaction models for multiple-resident activity recognition in a smart home. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3753–3758. IEEE, 2010. 5, 23, 28, 37

[52] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113, 2023. 90, 94

[53] Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017. 94

[54] Gabriele Civitarese, Michele Fiori, Priyankar Choudhary, and Claudio Bettini. Large language models are zero-shot recognizers for activities of daily living. *arXiv preprint arXiv:2407.01238*, 2024. 5, 23

[55] Gabriele Civitarese, Michele Fiori, Priyankar Choudhary, and Claudio Bettini. Large language models are zero-shot recognizers for activities of daily living. *ACM Transactions on Intelligent Systems and Technology*, 16(4):1–32, 2025. 91

[56] Ian Cleland, Luke Nugent, Federico Cruciani, and Chris Nugent. Leveraging large language models for activity recognition in smart environments. In *2024 International Conference on Activity and Behavior Computing (ABC)*, pages 1–8. IEEE, 2024. xvii, 5, 23

[57] Diane J Cook. Learning setting-generalized activity models for smart spaces. *IEEE intelligent systems*, 27(1):32, 2012. 31, 32

[58] Diane J Cook, Aaron Crandall, Geetika Singla, and Brian Thomas. Detection of social interaction in smart spaces. *Cybernetics and Systems: An International Journal*, 41(2):90–104, 2010. 5, 26, 38

[59] Diane J Cook, Aaron S Crandall, Brian L Thomas, and Narayanan C Krishnan. Casas: A smart home in a box. *Computer*, 46(7):62–69, 2012. 30, 54, 55

[60] Diane J. Cook, Aaron S. Crandall, Brian L. Thomas, and Narayanan C. Krishnan. CASAS: A smart home in a box. *Computer*, 46(7):62–69, 2013. 106

[61] Diane J Cook and Maureen Schmitter-Edgecombe. Assessing the quality of activities in a smart environment. *Methods of information in medicine*, 48(5):480, 2009. 31, 32

[62] Diane J. Cook, Maureen Schmitter-Edgecombe, Aaron Crandall, Chad Sanders, and Brian Thomas. Collecting and disseminating smart home sensor data in the CASAS project. In *Proceedings of the CHI Workshop on Developing Shared Home Behavior Datasets to Advance HCI and Ubiquitous Computing Research*, 2009. 4, 77

[63] Aaron S Crandall and Diane Cook. Attributing events to individuals in multi-inhabitant environments. In *2008 IET 4th International Conference on Intelligent Environments*, pages 1–8. IET, 2008. 5, 24, 38, 52

[64] Aaron S Crandall and Diane J Cook. Resident and caregiver: Handling multiple people in a smart care facility. In *AAAI Fall symposium: AI in eldercare: new solutions to old problems*, pages 39–47, 2008. 5, 24, 38

[65] Aaron S Crandall and Diane J Cook. Using a hidden markov model for resident identification. In *2010 Sixth International Conference on Intelligent Environments*, pages 74–79. IEEE, 2010. 5, 26, 38, 52

[66] Federico Cruciani, Anastasios Vafeiadis, Chris Nugent, Ian Cleland, Paul McCullagh, Konstantinos Votis, Dimitrios Giakoumis, Dimitrios Tzovaras, Liming Chen, and Raouf Hamzaoui. Feature learning for human activity recognition using convolutional neural networks: A case study for inertial measurement unit and audio data. *CCF Transactions on Pervasive Computing and Interaction*, 2(1):18–32, 2020. 3

[67] Julien Cumin, Grégoire Lefebvre, Fano Ramparany, and James L Crowley. Psines: Activity and availability prediction for adaptive ambient intelligence.

*ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, 15(1):1–12, 2020. 21

[68] Julien Cumin, Fano Ramparany, James L Crowley, et al. Inferring availability for communication in smart homes using context. In *2018 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, pages 1–6. IEEE, 2018. 21

[69] Jessamyn Dahmen, Brian L Thomas, Diane J Cook, and Xiaobo Wang. Activity learning as a foundation for security monitoring in smart homes. *Sensors*, 17(4):737, 2017. 2, 21

[70] L Minh Dang, Kyungbok Min, Hanxiang Wang, Md Jalil Piran, Cheol Hee Lee, and Hyeonjoon Moon. Sensor-based and vision-based human activity recognition: A comprehensive survey. *Pattern Recognition*, 108:107561, 2020. 20

[71] Deepraj Das, Soumik Mandal, Ashutosh Golande, Argha Sen, and Sandip Chakraborty. mmwave based user identification using gait signatures. In *2025 17th International Conference on COMmunication Systems and NETworks (COMSNETS)*, pages 1032–1036. IEEE, 2025. 15

[72] DeepSeek-AI. Deepseek-v2: A strong, economical, and efficient mixture-of-experts language model. *arXiv preprint arXiv:2405.04434*, 2024. 93

[73] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *NAACL-HLT*, 2019. 92

[74] Giovanni Diraco, Alessandro Leone, and Pietro Siciliano. Towards abnormal behavior detection of elderly people using big data. In *Advances in Data Science: Methodologies and Applications*, pages 13–33. Springer, 2021. 2

[75] Giovanni Diraco, Gabriele Rescio, Andrea Caroppo, Andrea Manni, and Alessandro Leone. Human action recognition in smart living services and applications: context awareness, data availability, personalization, and privacy. *Sensors*, 23(13):6040, 2023. 2

[76] Afsaneh Doryab and Julian Togelius. Activity recognition in collaborative environments. In *The 2012 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2012. 19

[77] He Du, Zhiwen Yu, Fei Yi, Zhu Wang, Qi Han, and Bin Guo. Recognition of group mobility level and group structure with mobile devices. *IEEE Transactions on Mobile Computing*, 17(4):884–897, 2017. 19

[78] Pengsong Duan, Chen Li, Jie Li, Xianfu Chen, Chao Wang, and Endong Wang. Wisdom: Wi-fi based contactless multi-user activity recognition. *IEEE Internet of Things Journal*, 2022. 3, 19, 20

[79] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. xviii, 92, 93

[80] Thi V Duong, Hung Hai Bui, Dinh Q Phung, and Svetha Venkatesh. Activity recognition and abnormality detection with the switching hidden semi-markov model. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 838–845. IEEE, 2005. 4, 22

[81] Cynthia Dwork. Differential privacy: A survey of results. In *International conference on theory and applications of models of computation*, pages 1–19. Springer, 2008. 140

[82] Daniel Fährmann, Laura Martín, Luis Sánchez, and Naser Damer. Anomaly detection in smart environments: a comprehensive survey. *IEEE access*, 2024. 2, 21

[83] Chunhai Feng, Sheheryar Arshad, Siwang Zhou, Dun Cao, and Yonghe Liu. Wi-multi: A three-phase system for multiple human activity recognition with commercial wifi devices. *IEEE Internet of Things Journal*, 6(4):7293–7304, 2019. 3, 19, 20

[84] Jiayuan Gao, Yingwei Zhang, Yiqiang Chen, Tengxiang Zhang, Boshi Tang, and Xiaoyu Wang. Unsupervised human activity recognition via large language models and iterative evolution. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 91–95. IEEE, 2024. 5, 23, 32, 91

[85] Giorgos Giannios, Lampros Mpaltadoros, Vasilis Alepopoulos, Margarita Grammatikopoulou, Thanos G Stavropoulos, Spiros Nikolopoulos, Ioulietta Lazarou, Magda Tsolaki, and Ioannis Kompatsiaris. A semantic framework to detect problems in activities of daily living monitored through smart home sensors. *Sensors*, 24(4):1107, 2024. 2

[86] Munkhjargal Gochoo, Tan-Hsu Tan, Shing-Hong Liu, Fu-Rong Jean, Fady S Alnajjar, and Shih-Chia Huang. Unobtrusive activity recognition of elderly people living alone using anonymous binary sensors and dcnn. *IEEE journal of biomedical and health informatics*, 23(2):693–702, 2018. 4, 22

[87] Dawud Gordon, Martin Wirz, Daniel Roggen, Gerhard Tröster, and Michael Beigl. Group affiliation detection using model divergence for wearable devices. In *Proceedings of the 2014 ACM International Symposium on Wearable Computers*, pages 19–26, 2014. 19

[88] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. In *First conference on language modeling*, 2024. 92

[89] Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. *arXiv preprint arXiv:2111.00396*, 2021. 92

[90] Tao Gu, Liang Wang, Hanhua Chen, Xianping Tao, and Jian Lu. Recognizing multiuser activities using wireless body sensor networks. *IEEE transactions on mobile computing*, 10(11):1618–1631, 2011. 19

[91] Tao Gu, Zhanqing Wu, Xianping Tao, Hung Keng Pung, and Jian Lu. epsicar: An emerging patterns based approach to sequential, interleaved and concur-

rent activity recognition. In *2009 IEEE International Conference on Pervasive Computing and Communications*, pages 1–9, 2009. 12, 13

[92] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638, 2025. 90, 92, 93, 138

[93] Jinghuan Guo, Yiming Li, Mengnan Hou, Shuo Han, and Jianxun Ren. Recognition of daily activities of two residents in a smart home based on time clustering. *Sensors*, 20(5):1457, 2020. 19

[94] Jianguo Hao, Abdenour Bouzouane, and Sébastien Gaboury. Recognizing multi-resident activities in non-intrusive sensor-based smart homes by formal concept analysis. *Neurocomputing*, 318:75–89, 2018. 28

[95] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136*, 2016. 136

[96] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016. 93

[97] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 140

[98] Kuo-Chung Hsu, Yi-Ting Chiang, Gu-Yang Lin, Ching-Hu Lu, Jane Yung-Jen Hsu, and Li-Chen Fu. Strategies for inference mechanism of conditional random fields for multiple-resident activity recognition in a smart home. In *Trends in Applied Intelligent Systems: 23rd International Conference on Industrial Engineering and Other Applications of Applied Intelligent Systems, IEA/AIE 2010, Cordoba, Spain, June 1-4, 2010, Proceedings, Part I 23*, pages 417–426. Springer, 2010. 5, 15, 23, 26, 37, 38, 52

[99] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3, 2022. xix, 118, 121

[100] Jingzhi Hu, Tianyue Zheng, Zhe Chen, Hongbo Wang, and Jun Luo. Muse-fi: Contactless muti-person sensing exploiting near-field wi-fi channel variation. In *Proceedings of the 29th Annual International Conference on Mobile Computing and Networking*, pages 1–15, 2023. 20

[101] Xu Huang, Nitish Patel, and Kit P Tsoi. Application of mmwave radar sensor for people identification and classification. *Sensors*, 23(8):3873, 2023. 20

[102] Ahmad Jalal, Yeon-Ho Kim, Yong-Joong Kim, Shaharyar Kamal, and Daijin Kim. Robust human activity recognition from depth video using spatiotemporal multi-fused features. *Pattern recognition*, 61:295–308, 2017. 3

[103] Manan Jethanandani, Thinagaran Perumal, Yuh-Ching Liaw, Jieh-Ren Chang, Abhishek Sharma, and Yipeng Bao. Binary relevance model for activity recognition in home environment using ambient sensors. In *2019 IEEE International*

*Conference on Consumer Electronics-Taiwan (ICCE-TW)*, pages 1–2. IEEE, 2019. 19, 29, 30

[104] Manan Jethanandani, Abhishek Sharma, Thinagaran Perumal, and Jieh-Ren Chang. Multi-label classification based ensemble learning for human activity recognition in smart home. *Internet of Things*, 12:100324, 2020. 19, 29, 30

[105] Feng Jin, Renyuan Zhang, Arindam Sengupta, Siyang Cao, Salim Hariri, Nimit K Agarwal, and Sumit K Agarwal. Multiple patients behavior detection in real-time using mmwave radar and deep cnns. In *2019 IEEE Radar Conference (RadarConf)*, pages 1–6. IEEE, 2019. 3, 19, 20

[106] Shian-Ru Ke, Hoang Thuc, Yong-Jin Lee, Jenq-Neng Hwang, Jang-Hee Yoo, and Kyoung-Ho Choi. A review on video-based human activity recognition. *Computers*, 2(2):88–131, Jun 2013. 3

[107] Nicolas Knudde, Baptist Vandersmissen, Karthick Parashar, Ivo Couckuyt, Azarakhsh Jalalvand, André Bourdoux, Wesley De Neve, and Tom Dhaene. Indoor tracking of multiple persons with a 77 ghz mimo fmcw radar. In *2017 European Radar Conference (EURAD)*, pages 61–64. IEEE, 2017. 20

[108] Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213, 2022. 95

[109] Eric Kolve, Roozbeh Mottaghi, Winson Han, Eli VanderBilt, Luca Weihs, Alvaro Herrasti, Daniel Gordon, Yuke Zhu, Abhinav Gupta, and Ali Farhadi. Ai2-thor: An interactive 3d environment for visual ai. *arXiv preprint arXiv:1712.05474*, 2017. 139

[110] Hao Kong, Xiangyu Xu, Jiadi Yu, Qilin Chen, Chenguang Ma, Yingying Chen, Yi-Chao Chen, and Linghe Kong. m3track: mmwave-based multi-user 3d posture tracking. In *Proceedings of the 20th Annual International Conference on Mobile Systems, Applications and Services*, pages 491–503, 2022. 20

[111] Rahul Kumar, Imroj Qamar, Jaskaran Singh Virdi, and Narayanan Chatapuram Krishnan. Multi-label learning for activity recognition. In *2015 International Conference on Intelligent Environments*, pages 152–155. IEEE, 2015. 19, 29, 30

[112] Patrick Lapointe, Kévin Chapron, Kévin Bouchard, et al. A new device to track and identify people in a multi-residents context. *Procedia Computer Science*, 170:403–410, 2020. 19

[113] Oscar D Lara and Miguel A Labrador. A survey on human activity recognition using wearable sensors. *IEEE communications surveys & tutorials*, 15(3):1192–1209, 2012. 3

[114] Behzad Lashkari, Yuxiang Chen, and Petr Musilek. Energy management for smart homes—state of the art. *Applied Sciences*, 9(17):3459, 2019. 2, 21

[115] Athanasios Lentzas, Eleana Dalagdi, and Dimitris Vrakas. Multilabel classification methods for human activity recognition: A comparison of algorithms. *Sensors*, 22(6):2353, 2022. 19, 29, 30

[116] Jincheng Li, Binbin Li, Lin Wang, and Wenyuan Liu. Passive multiuser gait identification through micro-doppler calibration using mmwave radar. *IEEE Internet of Things Journal*, 11(4):6868–6877, 2023. 15

[117] Qimeng Li, Raffaele Gravina, Ye Li, Saeed H Alsamhi, Fangmin Sun, and Giancarlo Fortino. Multi-user activity recognition: Challenges and opportunities. *Information Fusion*, 63:121–135, 2020. 3, 12, 13, 16, 18, 21

[118] Qimeng Li, Raffaele Gravina, Sen Qiu, Zhelong Wang, Weilin Zang, and Ye Li. Group walking recognition based on smartphone sensors. In *Body Area Networks: Smart IoT and Big Data for Intelligent Health Management: 14th EAI International Conference, BODYNETS 2019, Florence, Italy, October 2-3, 2019, Proceedings 14*, pages 91–102. Springer, 2019. 19

[119] Tianhong Li, Lijie Fan, Mingmin Zhao, Yingcheng Liu, and Dina Katabi. Making the invisible visible: Action recognition through walls and occlusions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 872–881, 2019. 19, 20

[120] Jia-Ming Liang, Ping-Lin Chung, Yi-Jyun Ye, and Shashank Mishra. Applying machine learning technologies based on historical activity features for multi-resident activity recognition. *Sensors*, 21(7):2520, 2021. 29

[121] Lili Liu, Eleni Stroulia, Ioanis Nikolaidis, Antonio Miguel-Cruz, and Adriana Rios Rincon. Smart homes and home health monitoring technologies for older adults: A systematic review. *International journal of medical informatics*, 91:44–59, 2016. 2, 21

[122] Fei Luo, Stefan Poslad, and Eliane Bodanese. Temporal convolutional networks for multiperson activity recognition using a 2-d lidar. *IEEE Internet of Things Journal*, 7(8):7432–7442, 2020. 3, 19, 20

[123] Umar Mahmud, Shariq Hussain, and Tehmina Karamat. Towards an intelligent energy conservation approach for context-aware systems in smart environments. *Frontiers in Computer Science*, 6:1525382, 2024. 2, 21

[124] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282, 2017. 140

[125] Zhen Meng, Song Fu, Jie Yan, Hongyuan Liang, Anfu Zhou, Shilin Zhu, Huadong Ma, Jianhua Liu, and Ning Yang. Gait recognition for co-existing multiple people using millimeter wave sensing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 849–856, 2020. 20

[126] Raihani Mohamed, Thinagaran Perumal, Md Nasir Sulaiman, and Norwati Mustapha. Multi resident complex activity recognition in smart home: A literature review. *Int. J. Smart Home*, 11(6):21–32, 2017. 29, 30

[127] Raihani Mohamed, Thinagaran Perumal, Md Nasir Sulaiman, Norwati Mustapha, and MN Zainudin. Modeling activity recognition of multi resident using label combination of multi label classification in smart home. In *AIP*

*conference proceedings*, volume 1891, page 020094. AIP Publishing, 2017. 19, 29, 30

[128] Raihani Mohamed, Muhammad Noorazlan Shah Zainudin, Thinagaran Perumal, and Sufri Muhammad. Adaptive profiling model for multiple residents activity recognition analysis using spatio-temporal information in smart home. In *Proceedings of the 8th International Conference on Computational Science and Technology: ICCST 2021, Labuan, Malaysia, 28–29 August*, pages 789–802. Springer, 2022. 19, 29, 30

[129] Vimal Mollyn, Karan Ahuja, Dhruv Verma, Chris Harrison, and Mayank Goel. Samosa: Sensing activities with motion and subsampled audio. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 6(3):1–19, 2022. 3

[130] Anubhav Natani, Abhishek Sharma, and Thinagaran Perumal. Sequential neural networks for multi-resident activity recognition in ambient sensing smart homes. *Applied Intelligence*, 51:6014–6028, 2021. 19, 28

[131] Viet Nguyen, Mohamed Ibrahim, Siddharth Rupavatharam, Minitha Jawahar, Marco Gruteser, and Richard Howard. Eyelight: Light-and-shadow-based occupancy estimation and room activity recognition. In *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*, pages 351–359. IEEE, 2018. 3, 19, 20

[132] James R Norris. *Markov chains*. Number 2. Cambridge university press, 1998. 12, 24

[133] George Okeyo, Liming Chen, Hui Wang, and Roy Sterritt. Dynamic sensor data segmentation for real-time knowledge-driven activity recognition. *Pervasive and Mobile Computing*, 10:155–172, 2014. 4, 22

[134] Takayuki Osa, Joni Pajarinen, Gerhard Neumann, J Andrew Bagnell, Pieter Abbeel, and Jan Peters. An algorithmic perspective on imitation learning. *Foundations and Trends® in Robotics*, 7(1-2):1–179, 2018. 69, 70

[135] Nadia Oukrich, AM El Bouazaoui Cherraqi, and Driss Elghanami. Multi-resident activity recognition method based in deep belief network. *J. Artif. Intell*, 11:71–78, 2018. 29, 30

[136] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022. 90, 94

[137] Sameera Palipana, Dariush Salami, Luis A Leiva, and Stephan Sigg. Pantomime: Mid-air gesture recognition with sparse millimeter-wave radar point clouds. *Proceedings of the ACM on interactive, mobile, wearable and ubiquitous technologies*, 5(1):1–27, 2021. 3, 19, 21

[138] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th*

*annual meeting of the Association for Computational Linguistics*, pages 311–318, USA, 2002. Association for Computational Linguistics. 53

[139] Preksha Pareek and Ankit Thakkar. A survey on video-based human action recognition: recent updates, datasets, challenges, and applications. *Artificial Intelligence Review*, 54(3):2259–2322, 2021. 3

[140] Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th annual acm symposium on user interface software and technology*, pages 1–22, 2023. 95, 96, 139

[141] Youna Park and Jieun Han. Smart home advancements for health care and beyond: Systematic review of two decades of user-centric innovation. *Journal of Medical Internet Research*, 27:e62793, 2025. 1

[142] David Patterson, Joseph Gonzalez, Quoc Le, Chen Liang, Lluis-Miquel Munguia, Daniel Rothchild, David So, Maud Texier, and Jeff Dean. Carbon emissions and large neural network training. *arXiv preprint arXiv:2104.10350*, 2021. 140

[143] Judea Pearl. *Heuristics: intelligent search strategies for computer problem solving*. Addison-Wesley Longman Publishing Co., Inc., 1984. 44, 45

[144] Jacopo Pegoraro, Francesca Meneghello, and Michele Rossi. Multiperson continuous tracking and identification from mm-wave micro-doppler signatures. *IEEE Transactions on Geoscience and Remote Sensing*, 59(4):2994–3009, 2020. 20

[145] Qifan Pu, Sidhant Gupta, Shyamnath Gollakota, and Shwetak Patel. Whole-home gesture recognition using wireless signals. In *Proceedings of the 19th annual international conference on Mobile computing & networking*, pages 27–38, 2013. 20

[146] Xavier Puig, Kevin Ra, Marko Boben, Jiaman Li, Tingwu Wang, Sanja Fidler, and Antonio Torralba. Virtualhome: Simulating household activities via programs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8494–8502, 2018. 139

[147] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018. 46, 47, 91, 120

[148] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019. 46, 47, 91

[149] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020. 90, 94

[150] Asif Rahim, Yanru Zhong, Tariq Ahmad, Sadique Ahmad, Paweł Pławiak, and Mohamed Hammad. Enhancing smart home security: anomaly detection and

face recognition in smart home iot devices using logit-boosted cnn models. *Sensors*, 23(15):6979, 2023. 2, 21

[151] Md Motiur Rahman, Deepti Gupta, Smriti Bhatt, Shiva Shokouhmand, and Miad Faezipour. A comprehensive review of machine learning approaches for anomaly detection in smart homes: experimental analysis and future directions. *Future Internet*, 16(4):139, 2024. 2, 21

[152] Prajit Ramachandran, Barret Zoph, and Quoc V Le. Searching for activation functions. *arXiv preprint arXiv:1710.05941*, 2017. 74, 93

[153] E Ramanujam and Thinagaran Perumal. Mlmo-hsm: Multi-label multi-output hybrid sequential model for multi-resident smart home activity recognition. *Journal of Ambient Intelligence and Humanized Computing*, 14(3):2313–2325, 2023. 29

[154] Raúl Gómez Ramos, Jaime Duque Domingo, Eduardo Zalama, Jaime Gómez-García-Bermejo, and Joaquín López. Sdhar-home: A sensor dataset for human activity recognition at home. *Sensors*, 22(21):8109, 2022. 31

[155] Parisa Rashidi and Alex Mihailidis. A survey on ambient-assisted living tools for older adults. *IEEE journal of biomedical and health informatics*, 17(3):579–590, 2012. 1

[156] Iris Rawtaer, Rathi Mahendran, Ee Heok Kua, Hwee Pink Tan, Hwee Xian Tan, Tih-Shih Lee, and Tze Pin Ng. Early detection of mild cognitive impairment with in-home sensors to monitor behavior patterns in community-dwelling senior citizens in singapore: Cross-sectional feasibility study. *Journal of medical Internet research*, 22(5):e16854, 2020. 2, 21

[157] Ubaid ur Rehman, Pedro Faria, Luis Gomes, and Zita Vale. Future of energy management models in smart homes: A systematic literature review of research trends, gaps, and future directions. *Process Integration and Optimization for Sustainability*, pages 1–30, 2025. 2, 21

[158] Daniele Riboni, Claudio Bettini, Gabriele Civitarese, Zaffar Haider Janjua, and Rim Helaoui. Smartfaber: Recognizing fine-grained abnormal behaviors for early detection of mild cognitive impairment. *Artificial intelligence in medicine*, 67:57–74, 2016. 2, 21

[159] Daniele Riboni and Flavia Murru. Unsupervised recognition of multi-resident activities in smart-homes. *IEEE Access*, 8:201985–201994, 2020. 19, 26, 52, 55, 56

[160] Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 627–635. JMLR Workshop and Conference Proceedings, 2011. 135

[161] Silvia Rossi, Giovanni Acampora, and Mariacarla Staffa. Working together: a dbn approach for individual and group activity recognition. *Journal of Ambient Intelligence and Humanized Computing*, 11:6007–6019, 2020. 19

[162] Nirmalya Roy, Archan Misra, and Diane Cook. Ambient and smartphone sensor assisted adl recognition in multi-inhabitant smart environments. *Journal of ambient intelligence and humanized computing*, 7(1):1–19, 2016. 19

[163] Justyna Sarzynska-Wawer, Aleksander Wawer, Aleksandra Pawlak, Julia Szymanowska, Izabela Stefaniak, Michal Jarkiewicz, and Lukasz Okruszek. Detecting formal thought disorder by deep contextualized word representations. *Psychiatry Research*, 304:114135, 2021. 4, 23

[164] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. 94, 137

[165] Roy Schwartz, Jesse Dodge, Noah A Smith, and Oren Etzioni. Green ai. *Communications of the ACM*, 63(12):54–63, 2020. 140

[166] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2016. 92

[167] Noam Shazeer. Fast transformer decoding: One write-head is all you need. In *arXiv preprint arXiv:1911.02150*, 2019. 93

[168] Noam Shazeer. Glu variants improve transformer. *arXiv preprint arXiv:2002.05202*, 2020. 74, 93

[169] Noah Shinn et al. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. 95, 96

[170] Milyun Ni'ma Shoumi and Sozo Inoue. Leveraging the large language model for activity recognition: A comprehensive review. *International Journal of Activity and Behavior Computing*, 2024(2):1–27, 2024. 5, 23

[171] Deepika Singh, Erinc Merdivan, Sten Hanke, Johannes Kropf, Matthieu Geist, and Andreas Holzinger. Convolutional and recurrent neural networks for activity recognition in smart environment. In *Towards Integrative Machine Learning and Knowledge Extraction: BIRS Workshop, Banff, AB, Canada, July 24-26, 2015, Revised Selected Papers*, pages 194–205. Springer, 2017. 4, 22

[172] Deepika Singh, Erinc Merdivan, Ismini Psychoula, Johannes Kropf, Sten Hanke, Matthieu Geist, and Andreas Holzinger. Human activity recognition using recurrent neural networks. In *Machine Learning and Knowledge Extraction: First IFIP TC 5, WG 8.4, 8.9, 12.9 International Cross-Domain Conference, CD-MAKE 2017, Reggio, Italy, August 29–September 1, 2017, Proceedings 1*, pages 267–274. Springer, 2017. 4, 22

[173] Geetika Singla, Diane J Cook, and Maureen Schmitter-Edgecombe. Recognizing independent and joint activities among multiple residents in smart environments. *Journal of ambient intelligence and humanized computing*, 1(1):57–63, 2010. 5, 26, 31, 32, 38, 39, 57, 60, 61, 64, 78

[174] Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. Learning to summa-

rize with human feedback. *Advances in neural information processing systems*, 33:3008–3021, 2020. 94

[175] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024. 74, 93

[176] Yuan Sun and Jorge Ortiz. An ai-based system utilizing iot-enabled ambient sensors and llms for complex activity tracking. *arXiv preprint arXiv:2407.02606*, 2024. 5, 23

[177] Jonathan Synnott, Liming Chen, Chris D Nugent, and George Moore. Simulation of smart home activity datasets. *Sensors*, 14(8):14162–14179, 2014. 139

[178] Naoto Takeda, Roberto Legaspi, Yasutaka Nishimura, Kazushi Ikeda, Atsunori Minamikawa, Thomas Plötz, and Sonia Chernova. Sensor event sequence prediction for proactive smart home support using autoregressive language model. In *2023 19th International Conference on Intelligent Environments (IE)*, pages 1–8. IEEE, 2023. 91

[179] Naoto Takeda, Roberto Legaspi, Yasutaka Nishimura, Kazushi Ikeda, Atsunori Minamikawa, Thomas Plötz, and Sonia Chernova. Sensor event sequence prediction for proactive smart home support using autoregressive language model. In *2023 19th International Conference on Intelligent Environments (IE)*, pages 1–8, 2023. 47

[180] Sheng Tan, Linghan Zhang, Zi Wang, and Jie Yang. Multitrack: Multi-user tracking and activity recognition using commodity wifi. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–12, 2019. 3, 19, 20

[181] Tan-Hsu Tan, Munkhjargal Gochoo, Shih-Chia Huang, Yi-Hung Liu, Shing-Hong Liu, and Yun-Fa Huang. Multi-resident activity recognition in a smart home using rgb activity image and dcnn. *IEEE Sensors Journal*, 18(23):9718–9727, 2018. 19, 29, 30

[182] Emmanuel Munguia Tapia, Stephen S Intille, and Kent Larson. Activity recognition in the home using simple and ubiquitous sensors. In *International conference on pervasive computing*, pages 158–175. Springer, 2004. 4, 22

[183] Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023. 90

[184] Thomas Tegou, Ilias Kalamaras, Markos Tsipouras, Nikolaos Giannakeas, Kostantinos Votis, and Dimitrios Tzovaras. A low-cost indoor activity monitoring system for detecting frailty in older adults. *Sensors*, 19(3):452, 2019. 2, 21

[185] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic robotics*. MIT press, 2005. 136

[186] Megha Thukral, Sourish Gunesh Dhekane, Shruthi K Hiremath, Harish Haresamudram, and Thomas Ploetz. Layout-agnostic human activity recognition in smart homes through textual descriptions of sensor triggers (tdost). *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 9(1):1–38, 2025. 5, 23

[187] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023. 90, 92, 93, 94

[188] Son N Tran, Tung-Son Ngo, Qing Zhang, and Mohan Karunanithi. Mixed-dependency models for multi-resident activity recognition in smart homes. *Multimedia Tools and Applications*, 79:23445–23460, 2020. 19, 28

[189] Son N Tran, Dung Nguyen, Tung-Son Ngo, Xuan-Son Vu, Long Hoang, Qing Zhang, and Mohan Karunanithi. On multi-resident activity recognition in ambient smart-homes. *Artificial Intelligence Review*, 53(6):3929–3945, 2020. 28

[190] Dipti Trivedi and Venkataramana Badarla. Occupancy detection systems for indoor environments: A survey of approaches and methods. *Indoor and Built Environment*, 29(8):1053–1069, 2020. 2, 21

[191] Chukwuemeka Ugwu and Oluwafemi Oyeleke. Potential of large language model for activity recognition in activities of daily living: A systematic review. *Authorea Preprints*, 2025. 5, 23

[192] Tim Van Kasteren, Athanasios Noulas, Gwenn Englebienne, and Ben Kröse. Accurate activity recognition in a home setting. In *Proceedings of the 10th international conference on Ubiquitous computing*, pages 1–9, 2008. 4, 22

[193] Ashish Vaswani et al. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. 74, 92

[194] Ju Wang, Nicolai Spicher, Joana M Warnecke, Mostafa Haghi, Jonas Schwartze, and Thomas M Deserno. Unobtrusive health monitoring in private spaces: The smart home. *Sensors*, 21(3):864, 2021. 2, 21

[195] Liang Wang, Tao Gu, Xianping Tao, Hanhua Chen, and Jian Lu. Recognizing multi-user activities using wearable sensors in a smart home. *Pervasive and Mobile Computing*, 7(3):287–298, 2011. 19

[196] Liang Wang, Tao Gu, Xianping Tao, and Jian Lu. Sensor-based human activity recognition in a multi-user scenario. In *Ambient Intelligence: European Conference, AmI 2009, Salzburg, Austria, November 18-21, 2009. Proceedings*, pages 78–87. Springer, 2009. 19

[197] Tinghui Wang and Diane J Cook. smrt: Multi-resident tracking in smart homes with sensor vectorization. *IEEE transactions on pattern analysis and machine intelligence*, 2020. 4, 5, 6, 8, 15, 19, 23, 26, 37, 38, 48, 54, 55, 56, 65, 78, 89, 132

[198] Tinghui Wang and Diane J Cook. Multi-person activity recognition in continuously monitored smart homes. *IEEE transactions on emerging topics in computing*, 10(2):1130–1141, 2021. xvii, 5, 6, 15, 19, 23, 26, 27, 37, 38, 48, 55, 56, 65, 89, 132

[199] Wei Wang, Alex X Liu, Muhammad Shahzad, Kang Ling, and Sanglu Lu. Device-free human activity recognition using commercial wifi devices. *IEEE Journal on Selected Areas in Communications*, 35(5):1118–1131, 2017. 3, 19, 20

[200] Jason Wei, Maarten Bosma, Vincent Y Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. *arXiv preprint arXiv:2109.01652*, 2021. 94

[201] Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, et al. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*, 2022. 90

[202] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022. 91, 95

[203] Ronald J Williams and David Zipser. A learning algorithm for continually running fully recurrent neural networks. *Neural computation*, 1(2):270–280, 1989. 120

[204] Daniel H Wilson and Chris Atkeson. Simultaneous tracking and activity recognition (star) using many anonymous, binary sensors. In *International Conference on Pervasive Computing*, pages 62–79. Springer, 2005. xvii, 5, 15, 23, 24, 25, 26, 27, 37, 38, 132

[205] Martin Wirz, Daniel Roggen, and Gerhard Troster. Decentralized detection of group formations from wearable acceleration sensors. In *2009 International Conference on Computational Science and Engineering*, volume 4, pages 952–959. IEEE, 2009. 19

[206] Zhi Wu, Dongheng Zhang, Chunyang Xie, Cong Yu, Jinbo Chen, Yang Hu, and Yan Chen. Rfmask: A simple baseline for human silhouette segmentation with radio signals. *IEEE Transactions on Multimedia*, 2022. 20

[207] Zhijing Wu, Zhihui Cao, Xuliang Yu, Jiang Zhu, Chunyi Song, and Zhiwei Xu. A novel multi-person activity recognition algorithm based on point clouds measured by millimeter-wave mimo radar. *IEEE Sensors Journal*, 2023. 3, 19, 21

[208] Zhouwen Wu, Xia Chen, Yujun Lin, Jinyu Wen, and Yin Chen. A smart home energy management system based on human activity recognition and deep reinforcement learning. *Energy and Buildings*, 325:114951, 2024. 2, 21

[209] Chunyang Xie, Dongheng Zhang, Zhi Wu, Cong Yu, Yang Hu, and Yan Chen. Rpm: Rf-based pose machines. *IEEE Transactions on Multimedia*, 2023. 20

[210] Ruibin Xiong, Yi Yang, Di He, Kai Zheng, Shuxin Zheng, Eric Xing, and Yuxin Zhang. On layer normalization in the transformer architecture. In *International Conference on Machine Learning (ICML)*, 2020. 93

[211] Takao Yamasaki and Shuzo Kumagai. Nonwearable sensor-based in-home assessment of subtle daily behavioral changes as a candidate biomarker for mild cognitive impairment. *Journal of personalized medicine*, 12(1):11, 2021. 2, 21

[212] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025. 90, 92, 93, 94

[213] Xin Yang, Jian Liu, Yingying Chen, Xiaonan Guo, and Yucheng Xie. Mu-id: Multi-user identification through gaits using millimeter wave radios. In *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*, pages 2589–2598. IEEE, 2020. 20

[214] Shunyu Yao et al. React: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*, 2023. 95

[215] Juan Ye, Graeme Stevenson, and Simon Dobson. Kcar: A knowledge-driven approach for concurrent activity recognition. *Pervasive and Mobile Computing*, 19:47–70, 2015. 4, 12, 13, 19, 22

[216] Zheng Yuan, Hongyi Yuan, Chuanqi Tan, Wei Wang, Songfang Huang, and Fei Huang. Rrhf: Rank responses to align language models with human feedback without tears. In *Advances in Neural Information Processing Systems*, volume 36, pages 42582–42613, 2023. 137

[217] Nima Zafarmomen and Vidya Samadi. Can large language models effectively reason about adverse weather conditions? *Environmental Modelling & Software*, 188:106421, 2025. xviii, 92

[218] Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. Star: Bootstrapping reasoning with reasoning. In *Advances in Neural Information Processing Systems*, volume 35, pages 15476–15488, 2022. 137

[219] Biao Zhang and Rico Sennrich. Root mean square layer normalization. *Advances in Neural Information Processing Systems (NeurIPS) Workshop*, 2019. 74, 93

[220] Shibo Zhang, Yaxuan Li, Shen Zhang, Farzad Shahabi, Stephen Xia, Yu Deng, and Nabil Alshurafa. Deep learning in human activity recognition with wearable sensors: A review on advances. *Sensors*, 22(4):1476, 2022. 3

[221] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022. 90

[222] Mingmin Zhao, Yonglong Tian, Hang Zhao, Mohammad Abu Alsheikh, Tianhong Li, Rumen Hristov, Zachary Kabelac, Dina Katabi, and Antonio Torralba. Rf-based 3d skeletons. In *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication*, pages 267–281, 2018. 20

[223] Peijun Zhao, Chris Xiaoxuan Lu, Jianan Wang, Changhao Chen, Wei Wang, Niki Trigoni, and Andrew Markham. mid: Tracking and identifying people with millimeter wave radar. In *2019 15th International Conference on Distributed Computing in Sensor Systems (DCOSS)*, pages 33–40. IEEE, 2019. 20

[224] Baraa Zieni, Matthew A Ritchie, Anna Maria Mandalari, and Francesca Boem. An interdisciplinary overview on ambient assisted living systems for health monitoring at home: Trade-offs and challenges. *Sensors*, 25(3):853, 2025. 2, 21

[225] Han Zou, Jianfei Yang, Hari Prasanna Das, Huihan Liu, Yuxun Zhou, and Costas J Spanos. Wifi and vision multimodal learning for accurate and robust device-free human activity recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 0–0, 2019. 20