DomAIn: Towards Programless Smart Homes

Yueyuan Sui[1*], Yiting Zhang[1*], Yanchen Liu[2], Minghui Zhao[2], Kaiyuan Hou[2], Jingping Nie[2], Xiaofan Jiang[2], Stephen Xia[1]
Northwestern University[1], Columbia University[2]
{yueyuansui2024,yitingzhang2025}@u.northwestern.edu
{yl4189,mz2866,kh3119,jn2551}@columbia.edu
jiang@ee.columbia.edu,stephen.xia@northwestern.edu

Abstract

We are far from the full adoption of smart home technologies, let alone realizing truly intelligent homes. There are many barriers preventing homeowners from fully adopting smart home technologies, chief among them being their low perceived usefulness (PU) and perceived ease of use (PEoU). Current smart home ecosystems and frameworks only support simple triggers and require homeowners to specify step-by-step logic or purchase new devices to automate new tasks, which severely limits the PU and PEoU of these technologies. We propose DomAIn, a smart home platform that automatically generates, programs, and deploys logic to satisfy a wide range of home-based tasks based on available devices in the home environment, without requiring users to manually "program" any logic. We demonstrate through real deployments and user studies that by incorporating DomAIn, a platform that reduces the need for users to program logic, we can significantly improve a variety of factors that affect PU and PEoU, such as customizability and complexity, by up to 38%, as well as satisfy a diverse range of tasks in home scenarios.

CCS Concepts

• **Human-centered computing** → **Ubiquitous and mobile computing systems and tools**; • **Computing methodologies** → *Artificial intelligence*; • **Computer systems organization** → *Embedded and cyber-physical systems*.

Keywords

smart home, task automation, user study, usability, human-centered computing

ACM Reference Format:
Yueyuan Sui[1*], Yiting Zhang[1*], Yanchen Liu[2], Minghui Zhao[2],, Kaiyuan Hou[2], Jingping Nie[2], Xiaofan Jiang[2], Stephen Xia[1] . 2025. DomAIn: Towards Programless Smart Homes. In *The 3rd International Workshop on Human-Centered Sensing, Modeling, and Intelligent Systems (HumanSys '25), May 6–9, 2025, Irvine, CA, USA*. ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3722570.3726888

[*]Equal contribution.

1 Introduction

Around 70% of U.S. homes now have at least one smart device [11], and the market is projected to reach $182 billion by 2025 [5]. Yet, widespread adoption remains elusive [7].

Two key barriers are low *perceived usefulness* (PU) and *perceived ease of use* (PEoU) [13]. PU depends on how well devices can be customized to user needs, while PEoU is limited by device interoperability and setup complexity. Existing ecosystems such as IFTTT [8], Amazon Alexa [2], and Apple Home [9] often require compatible hardware and manual trigger-action programming, restricting flexibility.

Smart devices are typically marketed for narrow applications, with limited flexibility. A pet camera, for example, is not easily repurposed for infant monitoring. Current platforms require users to *manually configure or program logic* to perform complex tasks, often without support for model-driven or personalized automation. Consider fall detection: cameras offer high accuracy but low privacy; vibration sensors reverse that trade-off. Ideally, a system would combine multiple sensors based on user preferences.

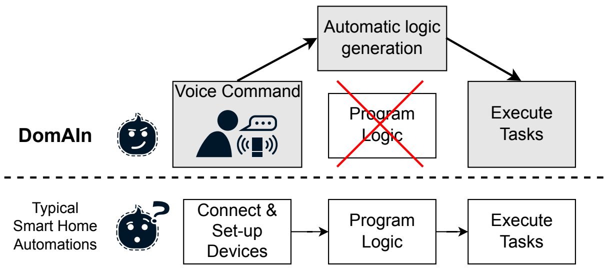
We propose DomAIn, a smart home platform that uses voice commands to automate a wide range of tasks, without requiring users to program logic. DomAInemploys a three-stage **logic generation pipeline** to dynamically build execution pipelines based on available devices and user-defined preferences across four dimensions: accuracy, privacy, coverage, and power efficiency (Figure 1). It adapts to device availability, automatically reconfiguring pipelines as needed.

Our contributions include:

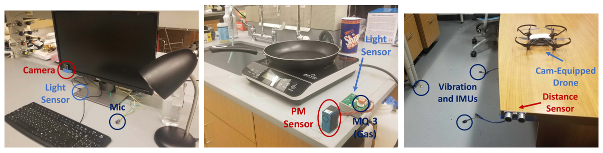
1. We present DomAIn, a configurable smart home system that dynamically leverages in-home resources to provide personalized services without user programming.
2. We propose a three-stage logic generation pipeline that assembles execution plans based on device availability and user preferences, satisfying up to 87% of tasks.
3. We validate DomAInthrough real-world deployments and user studies, showing up to 38% improvement in key PU and PEoU factors.

2 Related Works

Several research prototypes of smart homes have been developed over the years. [4] created an architecture using robots and multimedia devices to track user behavior, [12] focused on optimizing comfort with minimal energy consumption, and [6] introduced a framework for connecting various smart devices. These works typically require specific sensor types, whereas we focus on leveraging whatever sensors are available in the home to reduce adoption barriers.



(a)



(b) Left) Living room study. Middle) Kitchen. Right) Bedroom.

Figure 1: Left) Typical process for automating home services using existing smart home ecosystems. DomAIn eliminates the need for residents to "program" logic and enables more complex actions beyond simple triggers in existing smart home frameworks. Right) Deployment of DomAIn and various sensors/devices during a user study in a home setting.

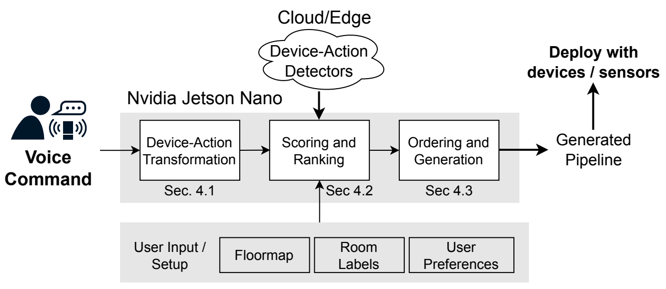


Figure 2: DomAIn's system architecture.

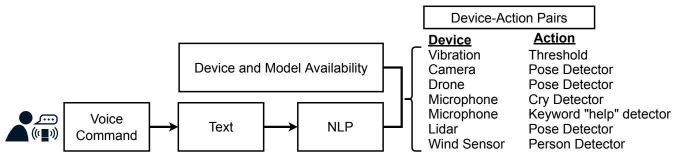


Figure 3: Device-action translation uses a tuned NLP model to map spoken commands into *viable device-action pairs*, based on whether the data collected from the sensor or device can be used to satisfy the spoken command by running an "action", model, or algorithm.

There is extensive research on using sensors to detect human activity [14–16, 18–21], and exploring drones in smart homes [17]. Our work doesn't introduce new detection algorithms but instead creates a platform that dynamically leverages existing sensors and methods to automatically execute tasks with minimal user guidance.

Commercial platforms from Google [10], Amazon [2], and Apple [9] allow device configuration, but users still need to program logic, and complex tasks often require specific products with built-in functionality. Open platforms like Home Assistant [3] provide centralized control but still require programming knowledge. Our work explores how a smart home AI can automatically generate execution pipelines from voice commands without requiring manual programming.

While interoperability remains a major challenge that industry is addressing through initiatives like the Matter protocol [1], we focus on how smart home systems can dynamically utilize available devices to reduce adoption barriers when interoperability is achieved.

3 DomAIn System Design

DomAIn is designed to mimic how a person would approach enabling smart services. When a user speaks a command, DomAIn interprets it through a three-stage logic generation pipeline that automatically creates and deploys an execution pipeline tailored to available devices and user preferences.

3.1 Three-Stage Logic Generation Pipeline

Table 1: a) Example voice command inputs and sensor-action pairs generated by our tuned GPT-3 NLP model. b) Performance of our translation NLP model before and after tuning.

(a)

Voice Command	Output Device-Action Pairs
"Let me know when the dog needs to go outside."	motion - high frequency detector microphone - barking detector vibration - high energy activity detection ...
"Tell me when my package arrives."	camera - object detector proximity - distance thresholding microphone - door bell detector ...
"Don't let anyone into the kitchen until tomorrow."	proximity - distance thresholding light - binary "on" detector camera - person detector ...

(b)

	Recall	Precision
Before Tuning	0.00	0.00
After Tuning	0.96	0.92

3.1.1 *Device-Action Translation.*

When humans implement a task, they first determine which sensors could satisfy it. For example, fall detection could use cameras, vibration sensors, or microphones. DomAIn translates voice commands into potential device-action pairs using a tuned GPT-3 NLP model. We created a dataset of 600 command-solution pairs, achieving 96% recall and 92% precision after tuning. This module also identifies the room referenced in commands with 92% accuracy, filtering out device-action pairs using unavailable sensors.

The tuning process involved three volunteers manually labeling command-device-action pairs, creating a training set that maps commands like "Alert me if someone falls" to a variety of solutions including camera-pose detection, vibration-threshold detection, and

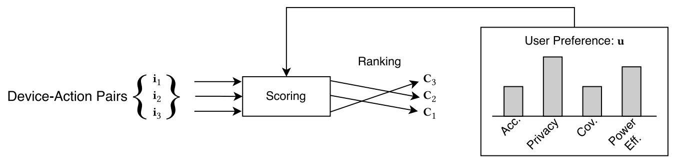


Figure 4: Pipeline for the Scoring Module. This module uses user preferences, u, to generate a score, $\mathbf{c}_i$, for each device-action i to determine the most fitting devices/sensors and actions/models to deploy that satisfies user preferences.

microphone-audio analysis. By using a large language model foundation, our system can handle a wide variety of natural language inputs and interpret user intent accurately without requiring rigid command formats.

3.1.2 Scoring and Ranking. Each sensing approach has different strengths and weaknesses. For example, cameras offer high accuracy but low privacy, while vibration sensors provide better privacy but lower accuracy. DomAIn ranks potential solutions based on user preferences across four dimensions (accuracy, privacy, power efficiency, and coverage) using a characteristic vector for each device-action pair.

The characteristic vector $\mathbf{c}_i$ for device-action pair i represents its performance across the four dimensions with values between 0 and 1. For instance, a camera with pose detection might have $\mathbf{c}_i = 0.85, 0.30, 0.30, 0.75$, indicating high accuracy (0.85) and coverage (0.75) but low privacy (0.30) and power efficiency (0.30). User preferences are captured in a similar vector $\mathbf{u}$ that expresses the importance of each dimension to the user.

We implemented four scoring methods and compared their performance:

(1) *Norm-based Distance*: $s_i = 1 - ||\mathbf{u} - \mathbf{c}_i||$
(2) *reLU Norm*: $s_i = 1 - ||reLU\mathbf{u} - c_i||$
(3) *MLP-DNN*: A deep neural network trained on human-scored pairings
(4) *Random Forest*: A regression model trained on human-scored pairings

Our evaluation found random forest regression performed best, achieving an 87% task satisfaction rate. Devices scoring below a threshold (0.4) are removed from consideration.

3.1.3 Pipeline Generation. The final step organizes sensors and actions into a staged execution pipeline. DomAIn uses two greedy algorithms:

(1) *Stage Generation*: Groups compatible device-actions until the stage score stops improving. To generate a stage, we order sensors/actions in descending order based on their scores and select the highest-scoring device-action as the initial element. We then iteratively add compatible device-actions that improve the stage's overall score.
(2) *Stage Evaluation*: Determines if adding the stage to the pipeline improves overall satisfaction of user preferences. We compute a weighted average between the previous pipeline and the newly added stage: $\mathbf{v}' \leftarrow d\mathbf{v}\ 1 - d\mathbf{v}_t$, where $0 < d < 1$ (we found $d = 0.8$ optimal in our experiments).

For example, a privacy-focused pipeline might first use vibration sensors to detect potential falls, then activate a camera only when triggered. This approach preserves privacy while maintaining accuracy. DomAIn continues adding stages until no further improvement is achieved, then deploys the pipeline using a library of pre-configured actions.

The pipeline generation algorithm handles complex trade-offs and creates multi-stage solutions that a single sensor approach could not achieve. For instance, in the fall detection case, a user prioritizing both coverage and privacy will receive a pipeline that uses microphones (good coverage) combined with vibration sensors (good privacy) in the first stage, activating further analysis only when potential falls are detected.

3.2 System Implementation

DomAIn requires three key inputs beyond voice commands:

(1) *Device Availability*: The system detects compatible devices on the home network through a Matter-like protocol. Each device reports its capabilities and sensing modalities (e.g., temperature sensing, video capture) through standardized descriptors, allowing DomAIn to assess available resources without manual configuration.
(2) *Device Location*: Users specify which room each sensor is in during setup. Although this requires initial manual input, it's a one-time process that enables room-specific task execution. Future versions could incorporate automatic localization techniques using signal strength or acoustic features.
(3) *Floor Map*: For mobile devices like drones, DomAIn needs a floor plan, which can be quickly generated using smartphone apps like RoomScan LiDAR. This map enables navigation between rooms and optimal positioning of mobile sensors when needed.

We implemented DomAIn on an Nvidia Jetson Nano with a user interface accessible via smartphone or web browser. The system communicates with sensors through ESP8266 WiFi modules and supports programmable drones for mobile sensing. Actions and models are downloaded from a cloud library and executed locally to preserve privacy.

Our deployment in real home environments used a variety of sensors (vibration, distance, microphones, cameras) and a Tello drone to enable diverse automation scenarios. By keeping computation at the edge and leveraging existing protocols like Matter, DomAIn demonstrates how automatic pipeline generation can be integrated into current and future smart home ecosystems.

4 System Evaluation

We evaluated DomAIn's ability to generate appropriate execution pipelines across a wide range of scenarios. We created a dataset with 1600 different test cases by randomly selecting from 485 diverse voice commands, generating random room configurations with 3-20 sensors from 20 different types, and varying user preferences.

Commands included a wide range of home automation tasks, such as "Let me know when the dog needs to go outside," "Tell me when my package arrives," and "Alert me if someone falls in

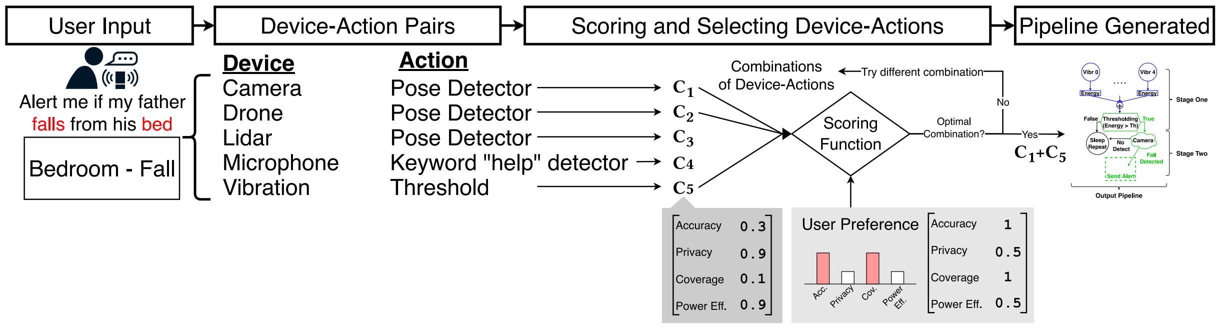


Figure 5: Execution pipeline generation architecture and example of generating a fall detection pipeline in a bedroom. DomAIn aggregates available sensors and actions in the room of interest and greedily attempts to generate and add stages to the pipeline to satisfy user preferences.

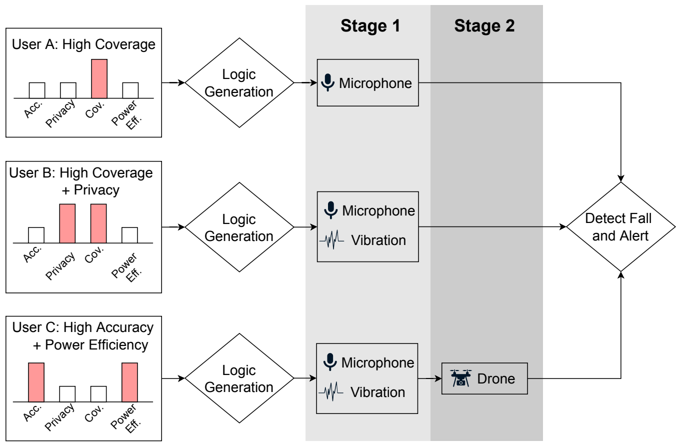


Figure 6: Examples of pipelines generated for "detecting and alerting if someone falls" with different user preferences: (top) high coverage, (middle) high coverage + privacy, (bottom) high accuracy + high power efficiency.

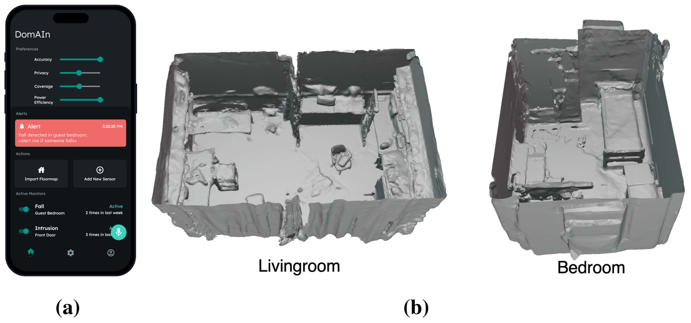


Figure 7: a) User interface of DomAIn, allowing users to view connected devices and input/select their user preferences. b) Floor plan of rooms captured by RoomScan smartphone application.

the bathroom." For each command, we considered various sensing approaches, from simple threshold-based detection to complex computer vision and audio analysis.

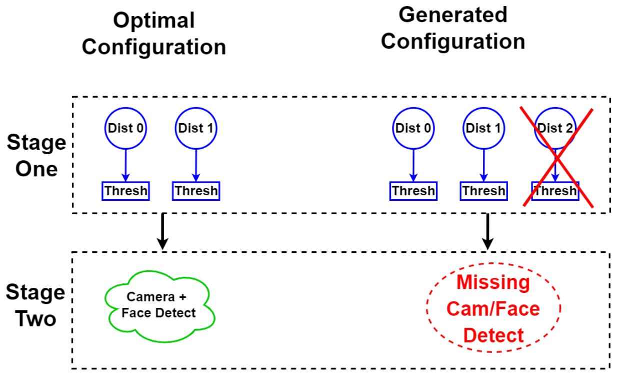


Figure 8: Example of the Levenshtein loss between a generated pipeline and the "ground truth" pipeline. The loss is 2 because we: 1) remove the extra distance sensor with thresholding, and 2) add a camera with a face detector.

Table 2: DomAIn's logic generation pipeline evaluation with four scoring modules across 1600 voice command scenarios. Norm-Lev is the normalized Levenshtein distance (lower is better). "Satisfy" shows the percentage of generated pipelines that could accomplish the task. Best performing method for each preference is highlighted.

	Random Forest		MLP DNN		Baseline	
Preference	norm-Lev	Satisfy	norm-Lev	Satisfy	norm-Lev	Satisfy
Accuracy	0.192	0.888	**0.182**	**0.898**	5.120	0.314
Privacy	**0.350**	**0.900**	0.351	0.876	2.144	0.439
Power	**0.176**	**0.817**	0.205	0.783	3.321	0.398
Coverage	**0.427**	**0.899**	0.432	0.832	3.250	0.455
Overall	**0.286**	**0.876**	0.293	0.847	3.546	0.402

To measure how well a generated pipeline matches the ground truth (human-crafted) pipeline, we used the Levenshtein distance metric (Figure 8), which counts the minimum number of device-action edits needed to transform one pipeline into another. We normalized this distance by dividing by the total number of device-action pairs in the ground truth pipeline.

Table 3: Summary of user study comparing DomAIn (A) with Home Assistant (G).

Setting	Setup Time (min)		Program Time (min)		Pipeline	Usability
	A	G	A	G	A	A vs G
Acc	35.0	22.8	1.8	>30	8.2	7.4 vs 4.6
Pow	33.3	22.9	1.1	>30	7.0	7.6 vs 2.5
Cov	34.0	28.1	1.2	>30	8.0	8.3 vs 4.0
Priv	35.5	25.3	1.5	>30	7.3	7.9 vs 3.4
Avg	34.5	24.8	1.4	>30	7.6	7.8 vs 3.6

Table 2 shows the performance of DomAIn's logic generation pipeline with different scoring methods compared to a baseline that simply selects devices with the highest rating in the user's most important dimension. DomAIn generates pipelines much closer to human-designed ones, with a normalized Levenshtein distance 10-12 times smaller than the baseline across all user preferences.

We also evaluated whether each generated pipeline could logically accomplish its task. The Random Forest scoring method achieved the highest satisfaction rate, successfully generating viable pipelines for 87.6% of tasks compared to only 40.2% for the baseline. This method performed particularly well for privacy-focused (90.0%) and coverage-focused (89.9%) preferences.

To analyze pipeline generation in detail, we examined a fall detection scenario. The system created three distinct pipelines based on user preferences: (1) for accuracy, it deployed cameras with pose detection; (2) for privacy, it used vibration sensors as initial triggers followed by camera activation only when needed; (3) for power efficiency, it used microphones and vibration sensors in a first stage, with selective drone deployment for confirmation.

Based on these results, we adopted the Random Forest scoring method for our physical deployment. The entire pipeline generation process takes approximately 4 seconds on our Jetson Nano platform, which is acceptable for most home automation tasks.

5 User Study and Physical Deployment

We evaluated how users respond to DomAIn and how its generated pipelines perform in real home environments. We recruited 23 participants (ages 20-30) with engineering backgrounds and compared DomAIn to Home Assistant on four smart home tasks: fall detection, child monitoring, stove monitoring, and intruder detection.

Our physical deployment involved a real home with four rooms (living room, bedroom, kitchen, bathroom), equipped with various sensors: vibration sensors (8), distance sensors (6), microphones (4), cameras (2), environmental sensors (temperature, air flow, light), and a programmable drone. Participants were asked to complete two phases:

1. *Setup Phase*: Install sensors and configure the system. DomAIn required slightly longer setup time (34.5 vs 24.8 minutes) due to floor mapping but received similar ease-of-use ratings.

2. *Programming Phase*: Create automation for the four test tasks. DomAIn excelled here, taking just 1.4 minutes on average compared to Home Assistant, where no participant completed programming within 30 minutes. Users simply selected their preferences (accuracy, privacy, power, coverage) in DomAIn, while Home Assistant required Python scripting knowledge.

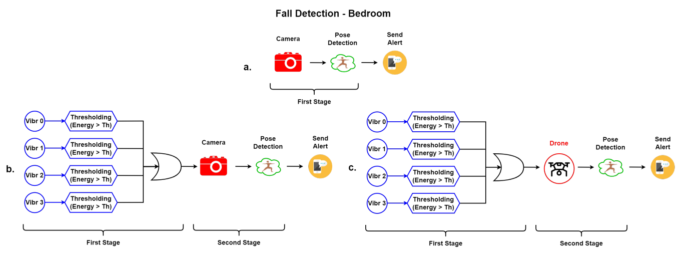


Figure 9: Fall detection pipelines deployed by DomAIn in the bedroom under a) accuracy, b) power, and c) power settings when the camera fails.

DomAIn-generated pipelines received high ratings (7.6/10 on average), with 95% of tasks satisfactorily met. Participants were particularly impressed by the system's ability to adapt to their preferences; for example, privacy-focused users noted that DomAIn minimized camera usage when appropriate.

Before and after using DomAIn, participants rated statements about smart home technologies (1-4 scale, higher meaning more concern). Initially, the average concern score was 3.02, with only 23% of responses indicating low concern (1-2). After using DomAIn, the average decreased to 2.19, with 61% in the low concern range—a 38% improvement. Most users (83%) agreed DomAIn executed tasks well, though privacy concerns remained.

In physical deployments, we tested DomAIn across all user preference settings. For fall detection with accuracy preference, the system achieved 95% true positive and 98% true negative rates. When a camera failed during testing, DomAIn automatically reconfigured the pipeline to use the drone instead, demonstrating its adaptability.

A particularly interesting case was intruder detection, where DomAIn created a multi-sensor first stage combining microphones (for sound detection) with distance sensors (for movement detection), followed by a drone deployment only when triggered. This demonstrated how the system could intelligently combine multiple sensing modalities to create solutions that balanced coverage, privacy, and power efficiency.

5.1 Case Study: Fall Detection Pipeline

To illustrate DomAIn's capabilities in detail, we analyze the fall detection pipeline across different user preference settings. Fall detection is a critical home safety application with significant tradeoffs between accuracy, privacy, and power consumption.

Figure 9a shows the pipeline generated for the bedroom with the accuracy preference setting. DomAIn uses the high-accuracy camera inside the room to continuously monitor for poses of a fallen person. This single-stage approach achieves 95% true positive and 98% true negative rates but has lower privacy ratings (3.2/10) from users.

For power-efficiency preference (Figure 9b), the system creates a two-stage pipeline. The first stage consists of multiple vibration sensors to detect potential falls with minimal power consumption (262 mW base consumption). Only when these sensors detect a potential fall does the system activate the camera for pose detection, significantly reducing overall power usage while maintaining 85% true positive detection.

Most notable is DomAIn's ability to adapt to failures. Figure 9c shows the pipeline automatically regenerated when the bedroom camera became unavailable. The system reconfigured to use the drone with pose detection capabilities as a second-stage replacement, maintaining similar performance characteristics with only minor changes to power efficiency and latency.

This adaptability extends to other scenarios as well. For intruder detection with coverage preference, DomAIn deployed a multi-sensor first stage combining microphones and distance sensors, followed by drone-based visual confirmation. This combination provided 80% true positive detection rate with wide spatial coverage that no single sensor could achieve.

5.2 Impact on Smart Home Adoption Barriers

Our user study revealed significant insights about how automatic pipeline generation impacts smart home adoption barriers. Before using DomAIn, participants identified four major concerns with existing systems:

1. *Programming Complexity*: Users without technical backgrounds found it impossible to implement complex automation tasks.

2. *Limited Device Reusability*: Most devices were perceived as single-purpose, requiring new purchases for each automation task.

3. *Privacy Concerns*: Users worried about continuous monitoring from cameras and microphones.

4. *Economic Burden*: The perceived need for specialized devices for each task created financial barriers.

DomAIn directly addressed these concerns by eliminating programming requirements, dynamically repurposing generic sensors, creating privacy-aware multi-stage pipelines, and reducing the need for specialized devices. The 38% improvement in user perception metrics demonstrates the effectiveness of this approach.

Particularly notable was the change in user attitudes toward privacy. While privacy remained a concern (though reduced), users appreciated the transparency of DomAIn's preference-based approach. As one participant stated: "I like that I can tell it privacy is important to me and it automatically reduces camera usage—it gives me more control without requiring technical knowledge."

These findings suggest that automatic pipeline generation significantly lowers barriers to smart home adoption by addressing both perceived usefulness and perceived ease of use simultaneously, without sacrificing customizability or control.

6 Conclusion

In this work, we present DomAIn, a smart home system that dynamically and intelligently leverages sensors and actuators to provide a wide range of services without requiring users to program logic. Through our novel three-stage logic generation pipeline, DomAIn generates execution pipelines based on available sensors and user preferences, satisfying 87% of tasks across diverse home settings.

Our user study demonstrates that DomAIn significantly improves automation ease compared to existing frameworks, enhancing factors affecting perceived usefulness and ease of use by up to 38%. The system's ability to automatically adapt to both user preferences and changing device availability addresses key barriers to smart home adoption.

DomAIn represents a fundamental shift in how smart home systems operate—from explicitly programmed logic to automatically generated and adaptive execution pipelines. By eliminating programming requirements while maintaining customizability through preference-based generation, DomAIn moves us one step closer to truly intelligent homes that adapt to users' needs rather than requiring users to adapt to technology limitations.

References

[1] Connectivity Standards Alliance. 2022. Build with Matter | Smart Home Device Solution. https://csa-iot.org/all-solutions/matter/.

[2] Amazon.com. [n. d.]. Amazon Echo & Alexa Devices. https://www.amazon.com/smart-home-devices/b?node=9818047011, Last accessed on 2021-12-18.

[3] Home Assistant. 2022. Raspberry Pi - Home Assistant. https://www.home-assistant.io/installation/raspberrypi/.

[4] S.K. Das, D.J. Cook, A. Battacharya, E.O. Heierman, and Tze-Yun Lin. 2002. The role of prediction algorithms in the MavHome smart home architecture. *IEEE Wireless Communications* 9, 6 (2002), 77–84. doi:10.1109/MWC.2002.1160085

[5] Statista Market Forecast. 2021. Smart Home - Worldwide. https://www.statista.com/outlook/dmo/smart-home/worldwide. [Online].

[6] S. Helal, W. Mann, H. El-Zabadani, J. King, Y. Kaddoura, and E. Jansen. 2005. The Gator Tech Smart House: a programmable pervasive space. *Computer* 38, 3 (2005), 50–60. doi:10.1109/MC.2005.107

[7] Areum Hong, Changi Nam, and Seongcheol Kim. 2020. What will be the possible barriers to consumers' adoption of smart home services? *Telecommunications Policy* 44, 2 (2020), 101867. doi:10.1016/j.telpol.2019.101867

[8] IFTTT. [n. d.]. IFTTT. https://ifttt.com/explore/new_to_ifttt, Last accessed on 2021-12-18.

[9] Apple Inc. [n. d.]. HomePod mini. https://www.apple.com/homepod-mini/, Last accessed on 2021-12-18.

[10] Google LLC. [n. d.]. Welcome to Google Nest. Build your helpful home. http://www.nest.com, Last accessed on 2021-12-18.

[11] Chuck Martin. 2019. Smart Home Technology Hits 69% Penetration in U.S. https://www.mediapost.com/publications/article/341320/smart-home-technology-hits-69-penetration-in-us.html. [Online].

[12] Michael C. Mozer. 1998. The Neural Network House: An Environment that Adapts to its Inhabitants.

[13] Shahrokh Nikou. 2019. Factors driving the adoption of smart home technology: An empirical assessment. *Telematics and Informatics* 45 (2019), 101283. doi:10.1016/j.tele.2019.101283

[14] Peter Wei, Xiaoqi Chen, Jordan Vega, Stephen Xia, Rishikanth Chandrasekaran, and Xiaofan Jiang. 2017. Eprints: A real-time and scalable system for fair apportionment and tracking of personal energy footprints in commercial buildings. In *Proceedings of the 4th ACM International Conference on Systems for Energy-Efficient Built Environments*. 1–10.

[15] Peter Wei, Stephen Xia, Runfeng Chen, Jingyi Qian, Chong Li, and Xiaofan Jiang. 2020. A deep-reinforcement-learning-based recommender system for occupant-driven energy optimization in commercial buildings. *IEEE Internet of Things Journal* 7, 7 (2020), 6402–6413.

[16] Peter Wei, Stephen Xia, and Xiaofan Jiang. 2018. Energy saving recommendations and user location modeling in commercial buildings. In *Proceedings of the 26th Conference on User Modeling, Adaptation and Personalization*. 3–11.

[17] Stephen Xia, Rishikanth Chandrasekaran, Yanchen Liu, Chenye Yang, Tajana Simunic Rosing, and Xiaofan Jiang. 2021. A Drone-Based System for Intelligent and Autonomous Homes. In *Proceedings of the 19th ACM Conference on Embedded Networked Sensor Systems (SenSys '21)*. Association for Computing Machinery, New York, NY, USA, 349–350. doi:10.1145/3485730.3492881

[18] Stephen Xia, Peter Wei, Yanchen Liu, Andrew Sonta, and Xiaofan Jiang. 2023. Reca: a multi-task deep reinforcement learning-based recommender system for co-optimizing energy, comfort and air quality in commercial buildings. In *Proceedings of the 10th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*. 99–109.

[19] Stephen Xia, Peter Wei, Yanchen Liu, Andrew Sonta, and Xiaofan Jiang. 2024. A multi-task deep reinforcement learning-based recommender system for co-optimizing energy, comfort, and air quality in commercial buildings with humans-in-the-loop. *Data-Centric Engineering* 5 (2024), e26.

[20] Minghui Zhao, Kaiyuan Hou, Junxi Xia, Stephen Xia, and Xiaofan Jiang. 2024. Connecting Foundation Models with the Physical World using Reconfigurable Drone Agents. In *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking*. 1745–1747.

[21] Minghui Zhao, Junxi Xia, Kaiyuan Hou, Yanchen Liu, Stephen Xia, and Xiaofan Jiang. 2025. FlexiFly: Interfacing the Physical World with Foundation Models Empowered by Reconfigurable Drone Systems. arXiv:2403.12853 [cs.RO] https://arxiv.org/abs/2403.12853