

MULTIMODAL SELF-SUPERVISED ACTIVITY RECOGNITION ON CASAS SMART-HOME SENSOR STREAMS

by

HARMANPREET SINGH

(Under the Direction of Fei Dou)

ABSTRACT

This thesis studies multimodal human activity recognition in smart home environments using sensor streams from the CASAS house datasets. Sparse binary sensor events are transformed into four synchronized modalities: chronological event sequences, deterministic text, raster images, and event graphs. The project also contributes a reproducible software workflow: a single experiment intent configuration system, a smart-home playback simulator and a sensor position zone annotation GUI. The thesis emphasizes the 30 event window set. Across three seeds, the best weighted-F1 proposed models differ by house: Aruba is strongest with graph+image+sequence input, reaching weighted-F1 0.850; Cairo is strongest with graph input, reaching weighted-F1 0.842; Kyoto is strongest with image input, reaching weighted-F1 0.843 and Milan is strongest with all four modalities, reaching macro-F1 0.649. The study also shows that accuracy and weighted-F1 mask failures in minority activities, making macro-F1 and per-class diagnostics essential for smart-home deployment.

INDEX WORDS: Human activity recognition, smart homes, CASAS, multimodal learning, sensor streams, graph neural networks, transformers, self-supervised learning

MULTIMODAL SELF-SUPERVISED ACTIVITY RECOGNITION ON CASAS SMART-HOME
SENSOR STREAMS

by

HARMANPREET SINGH

B.E. Computer Engineering, Savitribai Phule Pune University, 2023

A Thesis Submitted to the Graduate Faculty of the
University of Georgia in Partial Fulfillment of the Requirements for the Degree.

MASTER OF SCIENCE

ATHENS, GEORGIA

2026

©2026
Harmanpreet Singh
All Rights Reserved

MULTIMODAL SELF-SUPERVISED ACTIVITY RECOGNITION ON CASAS SMART-HOME
SENSOR STREAMS

by

HARMANPREET SINGH

Major Professor: Fei Dou

Committee: Lakshmish Ramaswamy
Suchendra M Bhandarkar

Electronic Version Approved:

Ron Walcott
Dean of the Graduate School
The University of Georgia
August 2026

DEDICATION

I dedicate this work to my family, whose patience, encouragement and sacrifices made my education possible. Their support gave me the confidence to keep learning through difficult semesters, long experiments and every uncertain step in between.

I also dedicate it to the teachers and mentors who shaped the way I think. Their guidance taught me not only technical knowledge, but also discipline, curiosity and respect for careful work.

Finally, I dedicate this thesis to my colleagues and classmates, whose conversations, feedback and friendship made the research journey less solitary and more meaningful. This work is part of a larger educational path and I am grateful to everyone who helped make that path possible.

ACKNOWLEDGEMENTS

I want to express my deepest gratitude to my advisor, Dr. Fei Dou, for their invaluable guidance, immense patience and constant encouragement throughout this project. I am equally grateful to Zishuai Liu, whose insightful discussions and clarity helped me navigate complex concepts and resolve critical research doubts. My sincere thanks also go to Junhao Zhao for their invaluable assistance in setting up and running the baseline experiments, which were crucial to the evaluation of this work.

CONTENTS

Acknowledgements	v
List of Figures	vii
List of Tables	viii
1 Introduction	1
1.1 Research Objective	2
1.2 Thesis Questions	3
1.3 Contributions	3
1.4 Headline Findings	4
1.5 Organization	5
2 Background and Related Work	6
2.1 Smart Home Activity Recognition	6
2.2 Window Based Recognition	7
2.3 Multimodal Fusion	8
3 Data and Preprocessing	9
3.1 Datasets	9
3.2 House Metadata	10
3.3 Event Parsing	14
3.4 Sliding Windows	14
3.5 Configuration and Generated Artifacts	16
3.6 Simulator and Data Inspection GUI	16
3.7 Class Spaces and the <i>Other</i> Class	18
3.8 Generated Modalities	18
4 Methodology	20
4.1 System Overview	20
4.2 Why Multimodal Learning	22

4.3	Encoder Architectures	23
4.4	Fusion Model	24
4.5	Confidence Aware Gating	25
4.6	Self Supervised Pretraining	25
4.7	Supervised Finetuning	26
4.8	Evaluation	27
4.9	Selected Baseline Models	28
5	Experimental Results	29
5.1	Result Sources	29
5.2	Two Stage Experimental Protocol	30
5.3	Cross House Summary	31
5.4	Selected Baseline Comparison	32
5.5	Additional Shorter Window Check	33
5.6	Aruba Results	34
5.7	Cairo Results	34
5.8	Kyoto Results	35
5.9	Milan Results	35
5.10	Activity Confusions and Shared Patterns	36
5.11	Best Model Recommendations	37
6	Discussion and Conclusion	38
6.1	Discussion	38
6.2	Limitations	40
6.3	Future Work	41
6.4	Conclusion	42
	Bibliography	43
	Appendices	44
A	Full Multimodal Rankings	44
A.1	Aruba	44
A.2	Cairo	45
A.3	Kyoto	46
A.4	Milan	47
B	Activity Remapping	48

LIST OF FIGURES

3.1	The <i>Sensor Annotator</i> tool. Each of the 34 Aruba sensors (31 motion, 3 door) is placed on the floorplan and assigned to a semantic zone via the zone-definition mode, in which the user draws polygon boundaries (e.g. the office zone, shown). The resulting sensor coordinates and zone assignments are persisted to <code>aruba.yaml</code> , the per-house metadata that drives the spatial heatmap and sensor-graph modalities. The panel reports 34/34 sensor positions and 34/34 zone assignments at native resolution (1612×1012 px).	11
3.2	The <i>CASAS Smart Home Simulator</i> . The tool replays the raw event log over the annotated Aruba floorplan, highlighting active sensors (orange) and recent transitions (coloured edges) as events stream at an adjustable rate. Tabs expose the four modality representations derived from each sliding window—graph, image, text, and sequence. The footer displays the current activity (<code>Meal_Preparation</code>) and the auto-generated textual window summary (time-of-day, day-of-week, dominant zone, dwell and pace features) consumed by the text modality.	17
4.1	Framework architecture for multimodal CASAS activity recognition. The system converts event windows into sequence, text, image and graph views, encodes each view and fuses active modalities for classification.	21
5.1	Top confusion pair visualizations for the four-house result set.	37

LIST OF TABLES

1.1	Best proposed models in <i>main result set</i> , ranked by mean weighted F1 over three seeds. Macro F1 and accuracy are shown as secondary metrics.	4
3.1	The four CASAS houses under the default activity labeling pipeline. Sensor counts and class counts are those used by shared Stage 1 metadata.	9
5.1	Cross house summary using weighted F1 as the primary metric. Macro F1 is shown as the secondary balanced-class diagnostic.	31
5.2	Selected baseline comparison for the main 30 event setting. Weighted F1 is the primary metric.	32
5.3	Best proposed models for the shorter 10 event check and the main 30 event setting. Delta is the macro F1 change from the shorter setting to the main setting.	33
5.4	Aruba top proposed models ranked by weighted F1.	34
5.5	Cairo top proposed models ranked by weighted F1.	34
5.6	Kyoto top proposed models ranked by weighted F1.	35
5.7	Milan top models ranked by macro F1	36
A.1	Aruba full multimodal ranking by weighted F1.	44
A.2	Cairo full multimodal ranking by weighted F1.	45
A.3	Kyoto full multimodal ranking by weighted F1.	46
A.4	Milan full multimodal ranking by macro-F1.	47
B.1	Final default activity classes.	48

CHAPTER I

INTRODUCTION

Human activity recognition in smart homes asks a deceptively difficult question: given a sparse stream of motion, door, item or environmental sensor events, what is the resident doing? The data is not dense video, audio or wearable traces, but is a collection of ambient sensors. They are irregular event logs such as a kitchen motion sensor turning on, a hallway sensor firing shortly afterward and a door sensor opening or closing. In CASAS style homes, these events must be converted into labels such as *Sleep, Eat, Cook, Work, Leave Home, Bed to toilet*, etc. The task matters because unobtrusive sensing can support aging in place, health monitoring, resident routine analysis, and assistive computing without invading the privacy of residents wearing devices or living under cameras.

The main technical challenge is ambiguity. A short local sensor window can contain similar evidence for different activities: kitchen sensors may indicate cooking, eating, taking medicine or general household movement; bedroom sensors may indicate sleeping, bathing transitions or bed to toilet episodes; entry sensors can indicate leaving, returning or simply passing through a transition zone. The challenge is worsened by severe class imbalance. A home contains far more instances of common background behavior than of rare events such as *Take medicine* or *Bed to toilet*. A model can therefore achieve high accuracy or high weighted F1 score while still failing the rare activities that may matter most.

This thesis presents a multimodal approach to this problem using the CASAS datasets: Aruba, Cairo, Kyoto and Milan. The primary experimental setting evaluates 30 event windows across multiple houses,

allowing the study to compare how different sensor layouts, resident structures and activity distributions affect recognition performance. The activity pipeline retains an *Other* class for windows that do not belong to the target activity subset and each house keeps the activity vocabulary supported by its sensor layout and annotations.

In addition to the model and experiments, the project has matured into an end-to-end research environment. A single experiment specification is expanded into concrete per-house and per-seed training configurations. A desktop experiment graphical user interface edits that intent file, previews derived paths and dimensions, manages active baselines and launches local smoke stages. A separate simulator visualizes built windows over each house floor plan, including resident trajectory playback and per-window modality views. The sensor annotation graphical user interface supports direct metadata editing for all supported houses and assigns sensor zones from drawn polygons.

1.1 Research Objective

The objective is to evaluate whether synchronized multimodal representations improve smart home activity recognition across houses with different sensor layouts, resident structures and activity distributions. The system converts each event count window into four views:

1. A chronological sequence of sensor event features.
2. Deterministic summary text that describes zones, pace, counts and temporal context.
3. Behavior images rasterized onto a house-specific floorplan grid.
4. Typed event graphs that connect chronological events to physical sensor super nodes.

The core hypothesis is that these views are not redundant. Sequence encoders preserve order and event timing. Text adds compressed semantic context about zones and temporal motives. Images reveal spatial density and room level activation patterns. Graphs represent repeated sensor identities, temporal proximity and sensor to event relationships. If the hypothesis is correct, the best-performing model should

differ from one house to another, depending on how well its design fits the house’s sensor layout and activity labels.

1.2 Thesis Questions

This thesis addresses four questions.

Q1. Which modality combination gives the best balanced activity recognition for each house under the 30 event window setting?

Q2. Do the best multimodal models outperform the selected thesis baselines when all models are evaluated across the same houses and seeds?

Q3. How do weighted F_1 , macro F_1 and accuracy differ in the conclusions they support?

Q4. Which classes remain difficult after multimodal fusion and what do the figures in *main result set* reveal about those failures?

1.3 Contributions

The thesis makes the following contributions.

- It documents the end to end methodology of the framework: house metadata, event parsing, same label sliding windows, four modality generators, pretrained encoders, multimodal fusion and test evaluation.
- It documents the research software workflow: generated configurations, shared artifacts, the experiment graphical user interface, the smart-home simulator, sensor and zone annotation tooling, smoke tests and result reporting utilities.
- It provides a house-by-house analysis of the default 30 event result set for Aruba, Cairo, Kyoto, and Milan.

- It ranks the best models for each house by weighted F1 and reports macro F1 and accuracy as secondary metrics.
- It compares the multimodal model family against the selected thesis baselines: SICA, CPC, Deep-CASAS and DCNN.
- It interprets class level errors using the stored result figures: summary F1 plots, full confusion matrices, top confusion pairs, calibration curves, support vs F1 scatter plots and ablation summaries.

1.4 Headline Findings

Table 1.1 summarizes the main experimental outcome. The winning model depends strongly on the house. Aruba benefits from graph+image+sequence input, Cairo is graph dominant, Kyoto is strongest with image input, and Milan’s strongest proposed aggregate row uses all four modalities.

Table 1.1: Best proposed models in *main result set*, ranked by mean weighted F1 over three seeds. Macro F1 and accuracy are shown as secondary metrics.

House	Best model	Weighted F1	Macro F1	Accuracy
Aruba	Ours: graph+image+sequence	0.850 ± 0.002	0.735 ± 0.004	0.850 ± 0.003
Cairo	Ours: graph	0.842 ± 0.002	0.524 ± 0.010	0.834 ± 0.002
Kyoto	Ours: image	0.843 ± 0.005	0.766 ± 0.011	0.838 ± 0.007
Milan	Ours: graph+image+sequence+text	0.670 ± 0.023	0.618 ± 0.022	0.667 ± 0.026

The headline table also reveals why a single metric would be misleading. Cairo’s best weighted F1 proposed model is graph+text, not graph only, while Milan’s selected baseline comparison is closer under weighted F1 than under macro F1. Kyoto is the cleanest case: image+text wins both macro F1 and weighted F1. Because the activity distributions are imbalanced, this thesis treats macro F1 as the primary ranking criterion and interprets weighted F1 as a deployment style measure of average performance under the observed test distribution.

1.5 Organization

The document follows the structure of a conventional machine learning thesis. Chapter 2 reviews smart home HAR, windowed recognition and multimodal fusion. Chapter 3 describes the CASAS datasets, house metadata, event windowing and feature materialization pipeline. Chapter 4 details the sequence, text, image, graph and fusion models. Chapter 5 presents the experimental protocol, rankings, baseline comparisons, class diagnostics and figures from *main result set*. The conclusion discusses the implications, limitations and next steps.

CHAPTER 2

BACKGROUND AND RELATED WORK

2.1 Smart Home Activity Recognition

The CASAS project introduced the concept of deployable smart home sensing environments for activity recognition, health monitoring and behavior analysis (Diane J. Cook, Aaron S. Crandall, Brian L. Thomas, Narayanan C. Krishnan, 2013). A CASAS home typically records timestamped events from ambient sensors rather than continuous video. This makes the data comparatively privacy preserving, but also sparse and irregular. Human activity recognition in this setting therefore differs from image classification or wearable inertial sensing. The model must infer activities from a sequence of discrete events, the sampling rate of which is controlled by resident behavior and sensor state changes.

Smart home HAR surveys identify recurring difficulties: variable event density, sensor placement differences, resident specific routines, label noise, overlapping activities and severe class imbalance (Damien Bouchabou, Son Minh Nguyen, Christian Lohr, Benoit LeDuc, Ioannis Kanellos, 2021). The present work encounters all of these. Aruba is a single-resident home with a large motion/door sensor set and strong performance under the observed test distribution. Cairo has two residents and no door sensors, so leave home behavior must be inferred from motion only context. Kyoto contains the richest sensor set and multiple zones, producing strong spatial information but also a complex resident structure. Milan is

a single resident and closer to classic CASAS motion/door layouts, but it has a 10 class default label space and a chronological split that exposes a temporal shift.

2.2 Window Based Recognition

Most sensor based HAR pipelines segment a continuous event stream before classification. One option is a fixed wall clock interval, such as 30 seconds or five minutes. Another is an event count window, such as 10 or 30 events. This thesis uses event count windows because they produce fixed size model inputs even when the timeseries is irregular. A 30-event window may span seconds during dense kitchen activity or many minutes during sparse nighttime behavior. The window length therefore represents a fixed amount of sensor evidence rather than a fixed amount of real time.

Smaller windows are useful when the goal is localized feature learning. A compact window forces convolutional, recurrent or attention layers to focus on short term transitions, nearby sensor dependencies and the most relevant parts of a short input sequence. Smaller inputs are also efficient for self supervised augmentation because transformations such as noise, masking, scaling or local order changes create positive pairs within a focused behavior fragment.

Larger windows are useful when the goal is global and spatial context. A longer event span provides enough evidence to render a meaningful floorplan activity image, capture frequency sensitive movement patterns and represent inter room transitions. It also helps encode daily routine context, since locally similar patterns can indicate different activities depending on the time of day and the surrounding movement history.

2.3 Multimodal Fusion

Multimodal fusion can improve recognition when modalities provide complementary evidence. However, fusion is not guaranteed to be monotonic. A weak or noisy modality can dilute a strong one and different classes may prefer different modalities. The framework addresses this by combining a fusion Transformer with auxiliary unimodal heads. The fusion path learns cross modal interactions, while auxiliary heads keep each encoder independently class discriminative. At inference, learned per-class ensemble weights mix the fusion logits with auxiliary logits, and confidence gating adjusts the strength of modality tokens using the entropy of unimodal predictions.

This design is especially relevant for the *main result set* findings. Under the thesis’s primary weighted-F1 ranking, Aruba benefits from graph+image+sequence input, Cairo from graph input, Kyoto from image input and Milan’s strongest proposed aggregate row from all four modalities. The secondary macro-F1 rankings are not always identical, which supports class aware and house aware fusion rather than a blanket assumption that every added modality will help every dataset.

CHAPTER 3

DATA AND PREPROCESSING

3.1 Datasets

The experiments uses four CASAS datasets: Aruba, Cairo, Kyoto and Milan. Each dataset contains timestamped sensor events and activity labels. The framework stores house specific metadata in YAML files under `data/metadata`. These metadata files define sensor lists, sensor types, floor plan images, pixel positions, zone assignments, activity labels and label remapping rules. This design lets the same preprocessing and model code run across houses with very different sensor sets.

Table 3.1 summarizes the three experimental houses under the default activity labeling pipeline. The result set uses 30 event windows with a stride of 15 for all houses. Cairo, Milan and Kyoto use stratified splitting in their configuration. The baseline JSON files report a shared held out test set and per seed cross validation summaries for the baseline methods.

Table 3.1: The four CASAS houses under the default activity labeling pipeline. Sensor counts and class counts are those used by shared Stage 1 metadata.

House	Sensors	Classes	Character
Aruba	34	9	Single resident-> strong motion/door coverage
Cairo	27	7	Two residents-> dominated by a large <i>Other</i> class
Kyoto	55	7	Two residents-> comparatively balanced
Milan	31	10	Single resident-> many co located area activities

Each raw event is parsed into a five field record.

$$e_t = \{\text{timestamp}, \text{sensor}, \text{sensor_idx}, \text{state}, \text{label}\}$$

These four houses are deliberately not identical. Aruba is a single resident motion/door setting with many sensors and strong distribution-weighted performance. Cairo is a two resident motion only setting, Kyoto is a two resident sensor rich setting with motion, door and sparse item sensing and Milan is a single resident motion/door setting with more target classes. Reporting all three makes the thesis more informative than a single house result because it shows how the preferred representation changes with the physical sensing environment.

3.2 House Metadata

The metadata YAML file is the plug in layer that makes new houses possible. It describes the floor plan, accepted sensors, state mapping, zone names, sensor positions, activity classes and activity remapping. The model code reads these files rather than assuming one fixed house. As a result, the same sequence, image, text, graph, training and evaluation code can be reused for Aruba, Cairo, Kyoto, Milan and future datasets.

Sensor Annotation GUI

To streamline the integration of new smart home topologies and datasets, a graphical metadata configuration utility was developed. This tool mitigates the manual overhead and potential for human error associated with hardcoding spatial coordinates and zone boundaries within configuration files (e.g., YAML metadata files). Utilizing an interactive graphical user interface (GUI), users can superimpose sensor placements directly onto a digital floor plan. The interface dynamically translates these spatial selections into pixel coordinates and writes the spatial metadata directly to the sensor positions schema in real time.

The annotator also supports right click clearing, search/filtering, an “only unannotated” view and polygon based zone assignment. In zone mode, the user can draw a polygon around a room or region; every annotated sensor inside the polygon receives that zone label in sensor zones. This is especially useful for large houses such as those in Kyoto, where dozens of sensors must be installed and audited.

The practical effect is that a new dataset can be added via a shared event CSV file and a metadata file, rather than rewriting the entire image or graph preprocessing. Once sensor positions and zones are annotated, the same metadata automatically feeds graph spatial features, image heatmaps, zone channels and summary text tokens. This makes the pipeline extensible: a new floor plan and sensor list can be converted into a supported multimodal dataset with a visual metadata pass rather than custom code.

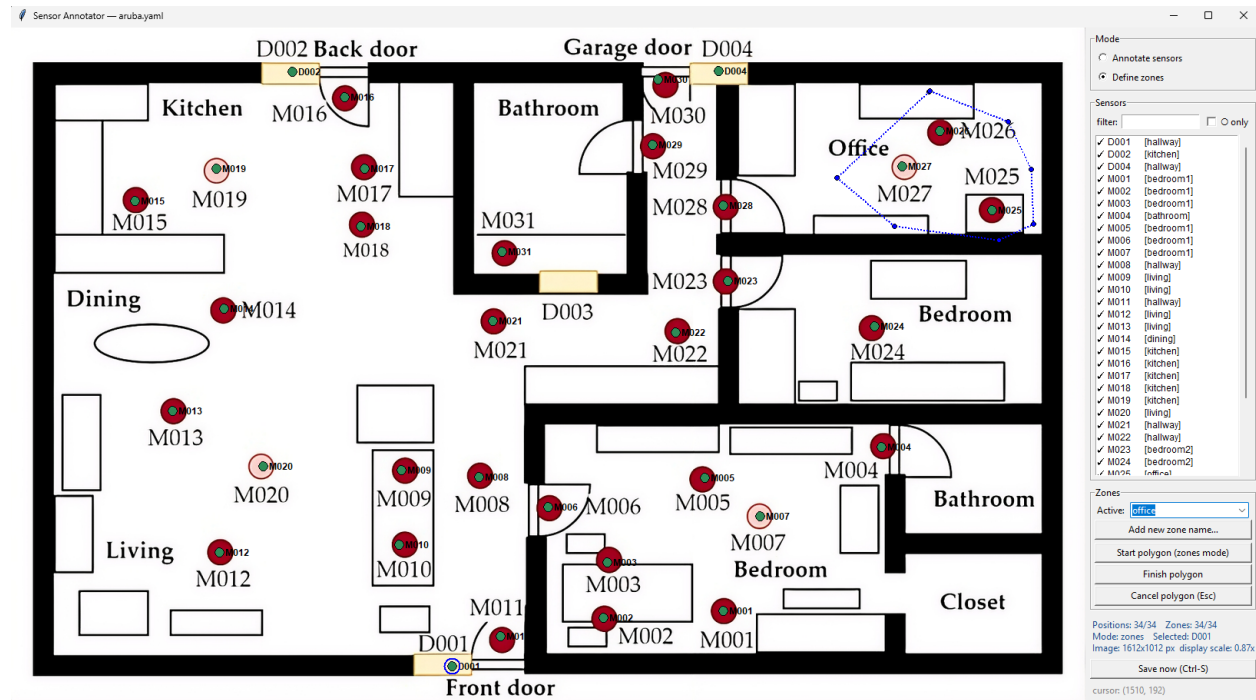


Figure 3.1: The *Sensor Annotator* tool. Each of the 34 Aruba sensors (31 motion, 3 door) is placed on the floorplan and assigned to a semantic zone via the zone-definition mode, in which the user draws polygon boundaries (e.g. the office zone, shown). The resulting sensor coordinates and zone assignments are persisted to `aruba.yaml`, the per-house metadata that drives the spatial heatmap and sensor-graph modalities. The panel reports 34/34 sensor positions and 34/34 zone assignments at native resolution (1612×1012 px).

Aruba

Aruba is a single resident CASAS house with 34 sensors: 31 motion sensors and 3 door sensors. The metadata maps sensors to zones including kitchen, dining, living, hallway, office, two bedrooms and bathroom. The default class set contains nine labels:

Meal_Preparation, Relax, Eating, Work, Sleeping, Bed_to_Toilet, Enter_Home,
Leave_Home, Housekeeping

Aruba is useful because it adds a high-performing single resident setting whose weighted F_1 is higher than the other thesis houses. The best proposed model for Aruba combines graph, image and sequence evidence, suggesting that the house benefits from both spatial floorplan structure and chronological sensor order.

Cairo

Cairo is a two resident CASAS house with 27 motion sensors and no door sensors. The metadata assigns sensors to zones including entry, bedroom, kitchen, laundry and hallway. Because no door sensors are present, the *Leave Home* class must be inferred from motion patterns around entry and transition areas rather than direct open/close events. The default class set contains seven labels:

Other, Work, Take_medicine, Sleep, Leave_Home, Eat, Bed_to_toilet

The absence of doors is a key reason the graph modality is valuable. Event graphs can represent repeated motion sensor identities, temporal proximity and physical sensor super node relationships even when there is no explicit door state.

Kyoto

Kyoto is the richest sensor environment in this result set. Its metadata includes motion sensors, door sensors and a sparse item sensor. Sensors are distributed across zones such as living/TV, kitchen/dining,

utility/entry, resident bedrooms, shared bath and transition areas. The default class set contains seven labels:

Other, Work, Sleep, Relax, Personal_hygiene, Cook, Bed_to_toilet

Kyoto's strong image model performance is consistent with this metadata. The house has many spatially meaningful zones, so rasterized behavior images can capture room occupancy and movement patterns more directly than in Cairo.

Milan

Milan is a single resident environment with motion and door sensors. The metadata maps sensors to zones, including kitchen, dining, living, office, master bedroom, bathrooms, guest bedroom and entry. The default class set contains ten labels:

Other, Work, Take_medicine, Sleep, Relax, Leave_Home, Eat, Cook, Bed_to_toilet,
Bathing

Milan is the only house in the main result set with ten classes. It is therefore a difficult direct comparison: the model must distinguish more activities, many of which occur in nearby or overlapping zones.

3.3 Event Parsing

The parser converts each raw row into a normalized event dictionary. The core fields are timestamp, sensor identifier, sensor index, binary state and activity label. Motion sensors use ON/OFF while door sensors use OPEN/CLOSE. These are mapped to binary values so the same model code can process both motion and door signals. Metadata files specify the valid sensor vocabulary and state map for each house. Unknown sensors and invalid labels are excluded.

The model treats the sensor list as house specific. Aruba has a large single-resident motion/door layout, Cairo has no doors, Kyoto has a larger, more irregular sensor zoo and Milan has a compact motion/door layout. This metadata driven design is important because hard coding Aruba style assumptions would fail when moving across houses.

3.4 Sliding Windows

The labeled window generator uses event count windows rather than fixed duration windows. Event count windows keep the amount of sensor evidence fixed even when real time is irregular. A short daytime kitchen routine may produce 30 events in seconds, while sparse nighttime behavior may require much longer clock time to produce the same number of events. The main experiments use:

$$W = 30 \quad S = 15$$

where W is the window length and S is the stride. This choice sits between two useful extremes. Smaller windows focus the encoder on fine grained local dynamics: short term transitions, nearby sensor dependencies and the most relevant parts of a compact input sequence. They are also computationally efficient for self supervised transformations because scaling, masking, noise or order perturbations affect a smaller local pattern.

Larger windows provide broader context. They give the spatial representation enough events to render a meaningful movement signature on the floor plan, preserve frequency sensitive sensor patterns and expose inter room relationships that may disappear in very small windows. They also help with the context of daily routines because activities such as a nap and nighttime sleep can look similar locally but occur at different times.

The code first partitions the labeled event stream into maximal contiguous runs with the same remapped activity label. If a run contains at least 30 events, the generator slides a 30-event window through the run using a stride of 15. If a run contains fewer than 30 events, the generator pads the short run without introducing cross label evidence.

The short run padding routine has three tiers:

1. borrow same label suffix events when the following events have the same label and are close in time
2. borrow same label prefix events when suffix borrowing is not possible
3. repeat the final real event when no same label context is available

This policy is conservative. It avoids mixed label windows, which is especially important for rare classes such as *Bed to toilet*, *Take medicine* and *Leave Home*. The tradeoff is that some rare short activities contain repeated evidence, so their F_1 should be interpreted with support and split stability in mind. The additional shorter window experiment in Chapter 5 tests the local dynamics side of this tradeoff. At the same time, the main 30 event setting is used as the default because it better supports spatial signatures and cross view alignment.

3.5 Configuration and Generated Artifacts

The current project uses a single-source configuration workflow. The user edits the compact intent file, which contains the selected house, window size, stride, seeds and run flags. The script merges this intent with optional house specific tuning files to derive the model dimensions.

This design removes several fragile manual steps. Changing the window size automatically changes all the configuration. The practical benefit is that experiments are defined by intent rather than by hand edited path strings scattered throughout the codebase.

3.6 Simulator and Data Inspection GUI

The framework includes a smart-home playback simulator that loads built windows and visualizes the selected house over its floor plan. It animates the resident trajectory through sensor activations, supports interactive house and exposes multiple modality views for the active window. These views include the original event sequence, deterministic text tokens, image channels and graph structure.

The simulator is not used to train the model, but it is important for analyzing the data and validating the preprocessing pipeline. It makes it possible to inspect whether sensor coordinates align with the floor plan, whether the images contain plausible spatial signatures, and whether a window label agrees with the visible event trace. This kind of inspection is particularly useful when adding a new house or debugging metadata errors that would otherwise only appear as weak model performance.

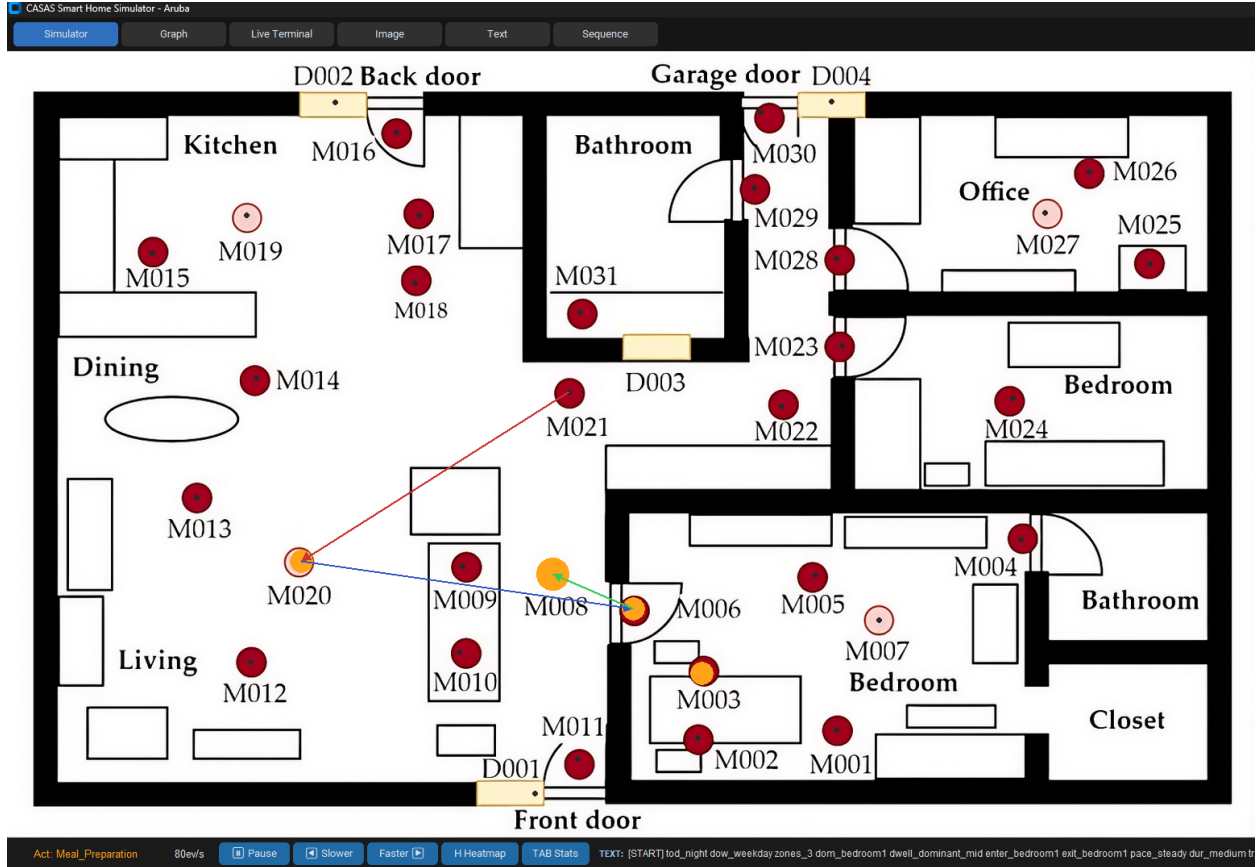


Figure 3.2: The *CASAS Smart Home Simulator*. The tool replays the raw event log over the annotated Aruba floorplan, highlighting active sensors (orange) and recent transitions (coloured edges) as events stream at an adjustable rate. Tabs expose the four modality representations derived from each sliding window—graph, image, text, and sequence. The footer displays the current activity (*Meal_Preparation*) and the auto-generated textual window summary (time-of-day, day-of-week, dominant zone, dwell and pace features) consumed by the text modality.

3.7 Class Spaces and the *Other* Class

The default activity pipeline keeps a house specific target activity subset and retains an *Other* class rather than discarding non target windows. This makes the problem more realistic: a model deployed in a home must decide not only which target activity is occurring, but also whether the current window belongs to none of the target activities.

The *Other* class also changes evaluation. High-support classes may dominate the weighted F1 and accuracy, while macro F1 gives each class equal weight. This thesis uses weighted F1 as the primary ranking metric because it reflects expected performance under the observed test distribution. Macro F1 is retained as the secondary class-balance diagnostic, especially for rare transitions and low-support activities.

3.8 Generated Modalities

Sequence Features

The sequence representation preserves the chronological order of the 30 events. For each event, the sequence encoder receives seven feature groups: sensor identity, binary state, normalized time since the start of the window, sensor type, log scaled inter event gap, state transition indicator and cyclic time features for hour of day and day of week.

The cyclic features use sine and cosine terms so that midnight and late night remain close in feature space. Log scaled inter event gaps preserve timing information without allowing very long gaps to dominate the feature scale.

Text

The text representation is a deterministic summary generated from the window. It includes time of day tokens, weekday/weekend tokens, dominant zone tokens, dwell fraction markers, pace and duration descriptors, zone transition motifs, revisit tokens and per sensor count tokens. The purpose is to provide

the text encoder with a semantic summary complementary to the raw sequence. It is less order sensitive than the sequence representation and more explicitly room aware.

Images

The behavior image representation rasterizes each window into a 96-by-96 image on the house floor plan. The six base channels represent motion activation, door activation, transition count, last-event recency, early-window count, and late-window count. Additional channels represent house specific zones. Therefore the number of image channels varies by house: the six base channels plus the number of zones. Aruba, Cairo, Kyoto and Milan each contribute a different zone structure, so image channel count is derived from metadata rather than hard coded.

Sparse sensor activations are normalized and blurred with Gaussian filters. The blur converts isolated points into spatial heatmaps, making the signal better suited to convolutional encoders. The early/late split gives the image encoder a coarse sense of motion direction without requiring a full video like sequence.

Graphs

The graph modality constructs a PyTorch Geometric graph for each window. Each graph contains event nodes and sensor super nodes. Event nodes carry state, previous gap, cyclic time, position in window, floor plan coordinates and window duration features. Super nodes represent fixed physical sensors and carry location information. Edges include sequential event edges, same sensor skip edges, temporal proximity cross sensor edges and event to sensor super node edges.

This graph design is important for generality across multiple houses. It allows the model to reason about repeated sensors, spatial identity and temporal proximity without assuming that every house has the same number of sensors or the same room layout.

CHAPTER 4

METHODOLOGY

4.1 System Overview

The framework implements a two stage multimodal learning pipeline. First, encoders can be pretrained on unlabeled windows using contrastive alignment and masked sensor prediction. Second, the model is finetuned on labeled windows for activity classification. The same `MultimodalModel` class supports any subset of the four modalities, so ablation experiments instantiate sequence only, text only, image only, graph only, pairwise, three way and all four modality models from the same core implementation.

The implementation is organized so that the research protocol can be run either from cluster scripts or from a local graphical user interface. The canonical batch entry point builds generated configs, creates shared windows, runs baselines, pretrains, executes the 15 modality ablations, evaluates checkpoints and generates visualizations. The graphical user interface exposes the same workflow for smaller local checks.

Figure 4.1 shows the architecture diagram from the existing report. The four modality generators feed modality specific encoders. A Transformer fuses their 256 dimensional embeddings with a learned fusion token. Auxiliary heads and ensemble logits are used during supervised classification.

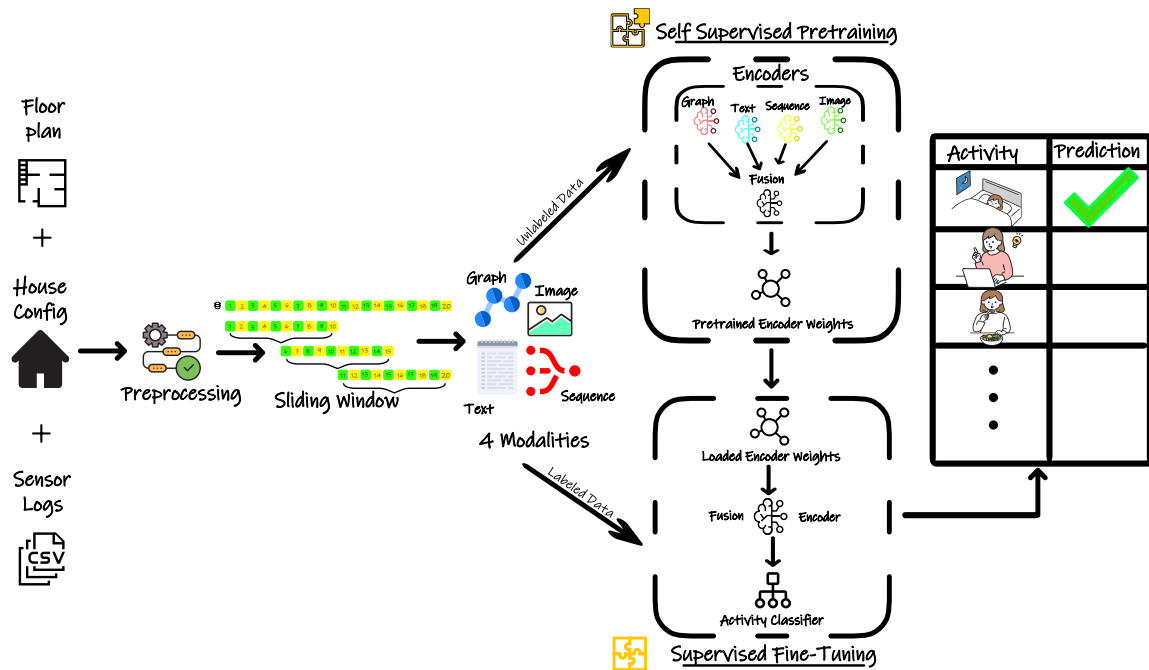


Figure 4.1: Framework architecture for multimodal CASAS activity recognition. The system converts event windows into sequence, text, image and graph views, encodes each view and fuses active modalities for classification.

4.2 Why Multimodal Learning

Any single view of the event stream does not fully describe Smart home activities. A chronological sequence preserves event order and fine timing: it can indicate whether a resident moved from the kitchen to the hallway or from the hallway to the kitchen, and whether events occurred in a burst or gradually over time. This temporal view is essential for transition like classes such as *Leave Home* and *Bed to toilet*.

The behavior image contributes spatial evidence. Placing activations on the floor plan lets a convolutional encoder see room-level occupancy, co-located sensors, early versus late movement, and recurring zone patterns. This is especially useful in houses such as Aruba, Kyoto and Milan, where several activities are separated more by where they occur than by the raw number of events.

The deterministic text summary contributes semantic and contextual evidence. It exposes dominant zones, pace, time of day, revisit patterns and sensor count phrases in a form that is less sensitive to small local permutations. This helps when the same activity can be performed with slightly different sensor orders but with similar high level context.

The graph view contributes relational structure. Event nodes, sensor super nodes, temporal edges, same sensor edges, and physical sensor links enable the model to reason about repeated sensors and neighborhood structure without requiring every house to have the same number of sensors. The fusion model is therefore not simply adding features; it is combining temporal, spatial, semantic and relational evidence produced from the same shared Stage 1 windows.

4.3 Encoder Architectures

Sequence Encoder

The sequence encoder is a Transformer encoder over 30 event positions plus a learned classification token. Each event embedding is formed by summing learned or projected representations of the seven sequence feature groups. The code includes a learnable scale for each feature group, allowing the model to learn, for example, that sensor identity or inter event gap is more informative than another feature for a given task. After adding a learned positional embedding, the sequence is processed by Transformer layers, and the classification token is projected into the shared 256-dimensional embedding space.

The main configuration uses a hidden sequence size of 128, four attention heads, four Transformer layers, a dropout of 0.1 and a maximum sequence length of 30.

Text Encoder

The text encoder is a compact Transformer over tokenized summary text. It uses a 256 dimensional model size, four attention heads, four layers, a feed forward dimension of 1024, maximum length 128 and dropout 0.1. The configuration reserves a BERT compatible vocabulary size of 30,522 with padding index of 0. The summary is deterministically derived from the event window, ensuring reproducibility and avoiding reliance on an external generation model during evaluation.

Image Encoder

The image encoder is a convolutional network over 96 by 96 behavior images. Four convolutional blocks progressively downsample the rasterized sensor activity, followed by adaptive pooling and projection into the shared 256-dimensional embedding space. The image encoder is designed to detect room level activity distributions, door/motion patterns, spatial transitions and early vs late activation changes.

Graph Encoder

The graph encoder uses GPSConv layers with local GINE style message passing and global multi head attention. Raw node features are augmented with learned sensor embeddings and node type embeddings. Edge attributes are augmented with learned edge type embeddings. The readout concatenates a mean pool over sensor super nodes with a mean pool over event nodes, then projects the result to 256 dimensions.

The main configuration uses two GPS layers, a hidden dimension of 128, an edge dimension of 3 and dropout of 0.1. The graph generator also supports a single context neighbor window.

4.4 Fusion Model

Let $z_m \in \mathbb{R}^{256}$ be the embedding for modality m . The fusion module normalizes active modality embeddings, adds learned modality identity embeddings, prepends a learned [FUSE] token and passes the resulting token sequence through a two layer Transformer. The fusion head classifies the fusion token's output.

The model also includes auxiliary classification heads for each active modality. These heads receive normalized unimodal embeddings before fusion. During training, the auxiliary loss ensures each encoder remains independently useful. During inference, the final logits are not only the fusion logits. The model stacks fusion logits and auxiliary logits, applies learned per class ensemble weights and forms a class wise convex mixture. This lets one class rely heavily on graph evidence while another relies more on image or text evidence.

4.5 Confidence Aware Gating

The model computes a confidence gate from the entropy of each auxiliary head. A low entropy auxiliary prediction is treated as more confident, while a high entropy prediction is treated as uncertain. The negative entropies are passed through a softmax with a learned temperature, giving a per sample gate over active modalities. These gates scale modality tokens before fusion. The intention is to let confident modalities influence the fusion Transformer more strongly while dampening uncertain modalities.

This design is motivated by the empirical behavior of multimodal HAR. Adding modalities is useful only when they provide reliable complementary evidence. A class aware mixture and a confidence aware gate are more flexible than forcing all decisions through a single fused representation.

4.6 Self Supervised Pretraining

Pretraining uses unlabeled windows and is label agnostic. The sequence, summary text, behavior image and graph views of the same window are treated as positive views of a single underlying activity fragment. In contrast, views from other windows in the batch are treated as negatives. Because no activity labels are used, the same pretrained encoders can be reused under any later label remapping or finetuning split without leaking supervised information.

Let $z_i^{(m)}$ be the normalized embedding for window i and modality m and let τ be the contrastive temperature. For a positive modality pair (m, n) , the framework’s NT Xent style loss can be written as

$$\mathcal{L}_{m,n} = \frac{1}{2N} \sum_{i=1}^N \left[\log \frac{\exp(\text{sim}(z_i^{(m)}, z_i^{(n)})/\tau)}{\sum_{j=1}^N \exp(\text{sim}(z_i^{(m)}, z_j^{(n)})/\tau)} \log \frac{\exp(\text{sim}(z_i^{(n)}, z_i^{(m)})/\tau)}{\sum_{j=1}^N \exp(\text{sim}(z_i^{(n)}, z_j^{(m)})/\tau)} \right]$$

The full pretraining objective combines cross modal contrastive learning, masked sensor prediction on the sequence encoder, fusion level contrastive alignment between the fused representation and individual modality embeddings and an optional sequential order prediction objective:

$$\mathcal{L}_{\text{pre}} = \lambda_{\text{cm}}\mathcal{L}_{\text{cm}} + \lambda_{\text{msp}}\mathcal{L}_{\text{msp}} + \lambda_{\text{joint}}\mathcal{L}_{\text{joint}} + \lambda_{\text{sop}}\mathcal{L}_{\text{sop}}$$

Masked sensor prediction randomly hides sensor identities and trains the sequence encoder to recover them from surrounding temporal context. Fusion level contrastive alignment prevents the fused embedding from drifting away from the individual modality embeddings. The sequential order objective, when enabled, asks the model to distinguish correctly ordered windows from order corrupted windows, encouraging sensitivity to temporal structure rather than only aggregate counts.

The Aruba, Cairo, Milan and Kyoto main configurations are pretrained for 100 epochs with a batch size of 512, a learning rate of 0.0003, a 10-epoch warmup, a contrastive weight of 1.0, a masked sensor prediction weight of 0.5, a joint contrastive weight of 0.5 and a temperature of 0.07.

4.7 Supervised Finetuning

Finetuning uses the shared labeled windows produced by Stage 1. It first trains with encoders frozen, so only the classifier, fusion head, auxiliary heads and ensemble weights adapt to the supervised task. After this warmup, all encoders are unfrozen and optimized with separate learning rate groups for the encoders, fusion and heads. This staged schedule protects the pretrained representations early in finetuning, then allows house specific adaptation once the classifier has stabilized.

Class imbalance is addressed at multiple levels. The DataLoader uses weighted sampling, the classification loss uses effective number class weighting and focal loss with $\gamma = 2.0$ reduces the effect of easy high support examples:

$$\mathcal{L}_{\text{focal}} = \alpha_y (1p_y)^\gamma \log p_y$$

where p_y is the predicted probability assigned to the true class and α_y is the class weight. The final supervised objective adds the main classification loss and the auxiliary modality head loss:

$$\mathcal{L}_{\text{ft}} = \mathcal{L}_{\text{main}} + \lambda_{\text{aux}} \mathcal{L}_{\text{aux}}$$

Finetuning runs for up to 50 epochs. The first five epochs are a warmup stage during which the encoders are frozen and the classifier/fusion heads are trained at a learning rate of 0.001. After warmup, encoders are unfrozen and trained with a learning rate of 0.0001, while the fusion module is trained with a learning rate of 0.0008. The configuration uses AdamW-style parameter groups, weight decay 0.0001 for encoders, stronger weight decay 0.0005 for fusion and classifier components, gradient clipping at 1.0, modality dropout 0.45, auxiliary loss weight 0.3, effective class weighting, and focal loss with gamma 2.0.

Checkpoints are selected based on the validation macro F1 score. This is an important choice: if checkpoint selection optimizes only accuracy or weighted F1, the model could overfit high support classes and neglect rare activities. Final predictions are the argmax of the model logits. No external decision layer, threshold tuning or sequence level postprocessor is added, so the reported numbers are attributable to the learned model.

4.8 Evaluation

Evaluation reports accuracy, weighted precision, weighted recall, weighted F1, macro precision, macro recall and macro F1. No threshold tuning or Markov filtering is used in the main configuration. The final reported values are the means and standard deviations across seeds 10, 30 and 50. The evaluation package also includes visualization scripts for summary F1 comparisons, ablation plots, full confusion matrices, top confusion pairs, calibration curves, support vs. F1 scatter plots and learning curves. The standalone script converts results into compact F1 tables and figures and extracts diagnostic summaries from result artifacts.

The thesis uses weighted F1 as the primary ranking metric:

$$\text{WeightedF1} = \sum_{c=1}^C \frac{n_c}{N} F1_c$$

where n_c is the support of class c and N is total test support. Weighted F1 reflects expected performance under the observed test distribution and is therefore the primary deployment-oriented metric in this report.

Macro F1 is reported as the secondary class-balance diagnostic. For C classes, macro F1 is

$$\text{MacroF1} = \frac{1}{C} \sum_{c=1}^C F1_c$$

Macro F1 gives every class equal weight, so it remains important for detecting minority-class weaknesses that weighted F1 can hide.

4.9 Selected Baseline Models

The thesis headline comparison uses a focused baseline set:

- **Sequence–Image Contrastive Alignment**(Junhao Zhao, Zishuai Liu, Ruili Fang, Jin Lu, Linghan Zhang, Fei Dou, 2025), reported as `sica`
- **Contrastive Predictive Coding**(Aaron van den Oord, Yazhe Li, Oriol Vinyals, 2018), reported as `cpc`
- **DeepCASAS**(Daniele Liciotti, Michele Bernardini, Luca Romeo, Emanuele Frontoni, 2019), represented by `deepcasas`
- **DCNN**(Munkhjargal Gochoo, Tan-Hsu Tan, Shing-Hong Liu, Fu-Rong Jean, Fady S. Alnajjar, Shih-Chia Huang, 2018), reported as `dcnn`
- **MMSAR**, our proposed model

This selected set keeps the comparison concise while still covering contrastive representation learning, smart home specific deep models, convolutional baselines and the best proposed multimodal configuration.

CHAPTER 5

EXPERIMENTAL RESULTS

5.1 Result Sources

This chapter analyzes the artifacts from the *main result set*. The multimodal ablation rows come from each house’s seed specific ablation files. Baseline rows come from the per house result files and their aggregated summaries. Figures are copied from each house’s result figure directory into this thesis project. The framework also includes utilities for generating reports that rebuild F1 tables and figures from a results directory, keeping the thesis tables aligned with the generated experiment artifacts.

All reported means and standard deviations are based on three seeds: 10, 30, and 50. The thesis uses weighted F1 as the primary ranking metric. Weighted F1 follows the empirical test distribution and is the most deployment-oriented score in this report. Macro F1 is the secondary metric because it gives each class equal weight and exposes minority-class failures that weighted F1 can hide. Accuracy is reported as a familiar auxiliary measure.

5.2 Two Stage Experimental Protocol

The comparison uses a two stage protocol, with stage 1 being the fairness critical stage and shared across models. It performs segmentation, label assignment and train/test splitting. Moreover, the raw event stream is first converted into identical (sample, label) pairs and the same samples are assigned to the same train/test partitions for the proposed model and all selected baselines and experiments.

Stage 2 is representation encoding and remains model specific. Once the shared Stage 1 windows are available, each model can convert a window to the tensor format it was designed to use. Sequence models utilize event sequences, image models render activity images, graph models build event graphs while token models use event token sequences and the proposed multimodal model builds all four of its representations. Moreover, all of these representations are derived from the same Stage 1 windows and labels.

This protocol prevents an unfair comparison in which one model benefits from a different segmentation rule or a different test split. Performance differences in the tables should therefore be interpreted as differences in Stage 2 representation and model learning, not differences in sample construction.

5.3 Cross House Summary

Table 5.1 compares the best weighted-F1 proposed model with the strongest selected baseline under the same primary metric. The selected baselines are SICA, CPC, DeepCASAS represented by `deepcasas_cascadeensemble` and DCNN.

Table 5.1: Cross house summary using weighted F1 as the primary metric. Macro F1 is shown as the secondary balanced-class diagnostic.

House	Best proposed model	Weighted F1	Macro F1	Best selected baseline	Baseline weighted F1	Baseline macro F1
Aruba	Ours: graph+image+sequence	0.850 ± 0.002	0.735 ± 0.004	SICA	0.847 ± 0.001	0.638 ± 0.001
Cairo	Ours: graph	0.842 ± 0.002	0.524 ± 0.010	SICA	0.755 ± 0.005	0.181 ± 0.016
Kyoto	Ours: image	0.843 ± 0.005	0.766 ± 0.011	SICA	0.821 ± 0.004	0.726 ± 0.005
Milan	Ours: graph+image+sequence+text	0.700 ± 0.009	0.702 ± 0.009	CPC	0.559 ± 0.011	0.682 ± 0.004

The four-house pattern is the main empirical result. The proposed model is the best weighted-F1 model for Aruba, Cairo and Kyoto. Milan is different: CPC has the highest weighted F1 in the selected comparison set, while the proposed all-modality model achieves a higher macro F1 than CPC. This illustrates why the thesis reports both metrics. Weighted F1 answers which model performs best under the observed test distribution, while macro F1 shows whether that performance is balanced across classes.

5.4 Selected Baseline Comparison

The thesis baseline set contains four external comparators and the best proposed model. Table 5.2 reports the selected comparison for the main 30 event setting. Table ?? reports the same selected comparison for the additional shorter window check.

Table 5.2: Selected baseline comparison for the main 30 event setting. Weighted F1 is the primary metric.

House	Model	Weighted F1	Macro F1	Accuracy
Aruba	Ours: graph+image+sequence	0.850 ± 0.002	0.735 ± 0.004	0.850 ± 0.003
Aruba	SICA	0.847 ± 0.001	0.638 ± 0.001	0.852 ± 0.001
Aruba	CPC	0.842 ± 0.001	0.740 ± 0.006	0.844 ± 0.001
Aruba	DeepCASAS	0.829 ± 0.002	0.713 ± 0.004	0.831 ± 0.001
Aruba	DCNN	0.823 ± 0.000	0.669 ± 0.000	0.827 ± 0.000
Cairo	Ours: graph	0.842 ± 0.002	0.524 ± 0.010	0.834 ± 0.002
Cairo	SICA	0.755 ± 0.005	0.181 ± 0.016	0.799 ± 0.003
Cairo	CPC	0.754 ± 0.002	0.377 ± 0.009	0.759 ± 0.004
Cairo	DeepCASAS	0.751 ± 0.002	0.374 ± 0.001	0.755 ± 0.002
Cairo	DCNN	0.750 ± 0.009	0.280 ± 0.012	0.803 ± 0.002
Kyoto	Ours: image	0.843 ± 0.005	0.766 ± 0.011	0.838 ± 0.007
Kyoto	SICA	0.821 ± 0.004	0.726 ± 0.005	0.826 ± 0.003
Kyoto	CPC	0.813 ± 0.003	0.712 ± 0.004	0.819 ± 0.003
Kyoto	DeepCASAS	0.812 ± 0.002	0.718 ± 0.004	0.818 ± 0.002
Kyoto	DCNN	0.813 ± 0.009	0.711 ± 0.011	0.822 ± 0.007
Milan	Ours: graph+image+sequence+text	0.649 ± 0.004	0.700 ± 0.009	0.702 ± 0.002
Milan	SICA	0.485 ± 0.006	0.630 ± 0.004	0.631 ± 0.015
Milan	CPC	0.559 ± 0.011	0.682 ± 0.004	0.685 ± 0.008
Milan	DeepCASAS	0.520 ± 0.003	0.636 ± 0.002	0.637 ± 0.009
Milan	DCNN	0.542 ± 0.016	0.685 ± 0.012	0.694 ± 0.003

5.5 Additional Shorter Window Check

The main thesis setting uses 30 event windows with a stride of 15. In addition, as a robustness check, the framework includes a shorter 10-event setting with a stride of 5. The shorter setting tests whether the model benefits from localized features: short term transitions, immediate sensor dependencies and attention over compact event sequences. Smaller windows also make self-supervised transformations such as masking, scaling, perturbations, and local order changes more focused, because each transformed view corresponds to a tight behavioral fragment.

The main 30 event setting tests the opposite side of the tradeoff. It gives the model enough context to construct a floorplan level spatial signature, capture frequency sensitive activation patterns and preserve inter room movement. This broader context is useful when a class is defined by the shape of a routine rather than a single local transition. Table 5.3 compares the best proposed model for each house under both settings.

Table 5.3: Best proposed models for the shorter 10 event check and the main 30 event setting. Delta is the macro F1 change from the shorter setting to the main setting.

House	10 event macro F1	10 event weighted F1	30 event macro F1	30 event weighted F1	Delta
Aruba	0.533 $\pm$ 0.013	0.809 $\pm$ 0.010	0.735 $\pm$ 0.004	0.850 $\pm$ 0.002	+0.182
Cairo	0.405 $\pm$ 0.008	0.829 $\pm$ 0.014	0.524 $\pm$ 0.011	0.842 $\pm$ 0.002	+0.119
Kyoto	0.774 $\pm$ 0.006	0.845 $\pm$ 0.007	0.766 $\pm$ 0.011	0.843 $\pm$ 0.005	-0.008
Milan	0.547 $\pm$ 0.003	0.675 $\pm$ 0.018	0.700 $\pm$ 0.009	0.649 $\pm$ 0.004	+0.153

The alternate 10 event setting is evidently better for Kyoto, according to macro F1. Kyoto gains 0.008 macro F1. The reason for better performance lies in the data, during data analysis it was noted that activities in kyoto are not only shorter but also there were minimal activity overlap in reference to the sensor. The results suggest that longer context is most helpful when activities require relational structure or multiple evidence types, whereas highly spatially separable activities may already be captured by shorter windows.

5.6 Aruba Results

Aruba is the strongest house by weighted F1. The best proposed model is graph+image+sequence, with weighted F1 0.850, macro F1 0.735 and accuracy 0.850. SICA is close by weighted F1 at 0.847 and has slightly higher accuracy, while CPC has the strongest macro F1 at 0.740. This means the proposed model is the best primary-metric model, but CPC is the strongest class-balance comparator.

Table 5.4: Aruba top proposed models ranked by weighted F1.

Rank	Model	Weighted F1	Macro F1	Accuracy
1	Ours: graph+image+sequence	0.850	0.735	0.850
2	Ours: image+sequence	0.847	0.725	0.846
3	Ours: graph+image+sequence+text	0.844	0.728	0.845
4	Ours: graph+image	0.843	0.723	0.844
5	Ours: graph+image+text	0.842	0.713	0.843

5.7 Cairo Results

Cairo’s best proposed model by weighted F1 is graph only, with weighted F1 0.842, macro F1 0.524 and accuracy 0.834. The macro-F1 winner is graph+sequence at 0.550, but its weighted F1 is lower at 0.830. Since this thesis is weighted-F1 first, graph is the recommended Cairo model.

Table 5.5: Cairo top proposed models ranked by weighted F1.

Rank	Model	Weighted F1	Macro F1	Accuracy
1	Ours: graph	0.842	0.524	0.834
2	Ours: graph+text	0.840	0.495	0.840
3	Ours: graph+sequence+text	0.838	0.512	0.832
4	Ours: graph+image+text	0.836	0.487	0.833
5	Ours: graph+sequence	0.830	0.550	0.820

Within the selected thesis baseline set, Cairo’s strongest weighted-F1 comparator is SICA at 0.755. CPC remains the strongest baseline by macro F1 at 0.377. The difference between these two baseline choices shows why metric priority must be explicit.

5.8 Kyoto Results

Kyoto’s best proposed model by weighted F1 is image only, with weighted F1 0.843, macro F1 0.766 and accuracy 0.838. The macro-F1 winner is graph+text at 0.777, with weighted F1 0.841. These two models are very close, but the primary metric selects image.

Table 5.6: Kyoto top proposed models ranked by weighted F1.

Rank	Model	Weighted F1	Macro F1	Accuracy
1	Ours: image	0.843	0.766	0.838
2	Ours: graph+text	0.841	0.777	0.841
3	Ours: graph+image+text	0.833	0.753	0.826
4	Ours: graph+image	0.820	0.768	0.814
5	Ours: text	0.816	0.759	0.814

Within the selected thesis baseline set, Kyoto’s strongest weighted-F1 comparator is SICA at 0.821. The proposed image model exceeds that by about 0.022 weighted F1, while graph+text remains the stronger proposed choice if macro F1 is prioritized.

5.9 Milan Results

Milan is the most revealing fusion case. The best balanced model is the all four modality model, with macro F1 0.649. The next best models also use multiple modalities: graph+image+sequence, graph+sequence+text, image+sequence+text and image+sequence. This indicates that Milan benefits from combining spatial, temporal, textual and relational evidence.

Within the selected thesis baseline set, the CPC is the strongest Milan macro F1 comparator. The all four modality proposed model remains the best-balanced model, as it is more evenly distributed across Milan’s 10 classes.

Within the selected thesis baseline set, the strongest comparator is CPC, with a macro F1 of 0.559, followed by DCNN at 0.542 and DeepCASAS at 0.520. All four modality models improve over the strongest selected baseline by about 0.090 macro F1.

Table 5.7: Milan top models ranked by macro F1

Rank	Model	Macro F1	Weighted F1	Accuracy
1	Ours: graph+image+sequence+text	0.649	0.700	0.702
2	Ours: graph+image+sequence	0.638	0.674	0.680
3	Ours: graph+sequence+text	0.636	0.678	0.682
4	Ours: image+sequence+text	0.633	0.696	0.699
5	Ours: image+sequence	0.629	0.681	0.687

5.10 Activity Confusions and Shared Patterns

The four houses show that smart-home HAR performance depends on both metric choice and house topology. Aruba has high weighted F1 across several methods, suggesting that the observed test distribution is comparatively learnable. Cairo has strong weighted F1 for graph-based proposed models but much lower macro F1, indicating persistent rare-class difficulty. Kyoto is strong under both weighted and macro F1, but the best model changes depending on which metric is prioritized. Milan shows the sharpest metric tradeoff: CPC and DCNN are stronger under weighted F1, while the proposed all-modality model is stronger under macro F1 in the selected table.

The top confusion pair visualizations show that several low F1 activities are not simply underfit; they are genuinely hard to distinguish from nearby activities because the sensor evidence overlaps. Smart home sensors are sparse, so two activities can share the same room, the same sensors, a similar duration and a similar event order. In those cases, the model must infer intent from weak context rather than a clean physical signature.

Cairo demonstrates this most clearly in rare and transition like classes. Without door sensors, *Leave Home* is inferred from motion around entry and transition zones and activities such as *Take medicine*, *Work* and *Bed to toilet* can be confused when their windows contain short or repeated motion patterns. Kyoto is stronger overall, but its *Other* class remains difficult because it aggregates heterogeneous fragments that may resemble several named activities. Milan has many co-located area activities; *Eat*, *Cook*, and *Take*

medicine can all activate kitchen or dining sensors with similar temporal density, so even a multimodal model struggles to distinguish among them consistently.

This explains why macro F1 remains lower than weighted F1 in Cairo and Milan. The high support classes are often spatially stable and easy to recognize, whereas rare or semantically adjacent activities occupy the same sensor neighborhoods. The top confusion pair plots in Figure 5.1 make this failure mode visible: errors concentrate where activity labels differ more by human intention than by observable sensor structure.

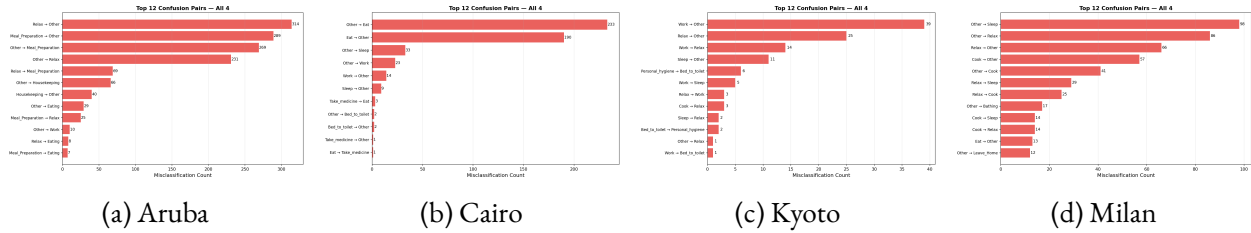


Figure 5.1: Top confusion pair visualizations for the four-house result set.

5.11 Best Model Recommendations

For Aruba, the recommended proposed model is graph+image+sequence. It is the best weighted-F1 model in the four-house result set.

For Cairo, the recommended model is graph only. It is the best weighted-F1 model and substantially outperforms the selected baselines under the primary metric.

For Kyoto, the recommended weighted-F1 model is image only. If balanced minority-class recognition is more important than distribution-weighted performance, graph+text is the better macro-F1 choice.

Lastly, for Milan, the recommendation depends on deployment priorities. If rare or minority classes are important, use graph+image+sequence+text. Within the selected baseline set, it is the best macro F1 model by a clear margin, making the four modalities model the correct headline model for balanced HAR.

CHAPTER 6

DISCUSSION AND CONCLUSION

6.1 Discussion

The main 30 event experiments show that multimodal smart home HAR is not a single model story. Different houses reward different representations. Aruba is best served by graph+image+sequence, Cairo by graph, Kyoto by image and Milan by the full four modality model. This is a strong argument for metadata aware and house aware model selection. It also argues against treating multimodal fusion as automatically superior. Fusion helps Milan, but graph only remains best for Cairo and image+text remains best for Kyoto.

The software system developed for these experiments is also a major contribution. The current framework is not only a collection of model scripts; it is an integrated workflow for defining experiments, deriving paths and dimensions, building shared windows, running baselines, launching ablations, visualizing results and inspecting the underlying house metadata. The experiment GUI, simulator and annotation GUI make the system more usable for iterative research and for extending the work to additional houses.

Why Weighted F₁ Leads

The most important lesson from the evaluation is that weighted F₁ and macro F₁ answer different questions. Weighted F₁ estimates performance under the observed class distribution and is therefore the primary deployment-oriented metric in this thesis. Macro F₁ estimates balanced class recognition and remains the secondary diagnostic. In smart home monitoring, rare activities can still matter even when they have small support, so the thesis reports macro F₁ alongside weighted F₁ rather than replacing it.

Why Macro F₁ is Also Important

The most important lesson from the evaluation is that weighted F₁ and macro F₁ answer different questions. Weighted F₁ estimates performance under the observed class distribution. Macro F₁ estimates balanced class recognition. In smart home monitoring, the rare activities can be the activities that matter most. A system that recognizes common background behavior but misses *Take medicine*, *Bed to toilet* or *Leave Home* is not sufficient, even if weighted F₁ looks strong.

This is clearest in Milan. The all four modality model has the best macro F₁, while selected baselines remain closer under distribution weighted scoring. The thesis therefore recommends the multimodal model for balanced recognition.

House Specific Interpretation

Aruba has a large single-resident motion/door layout. Its graph+image+sequence winner suggests that the house benefits from combining spatial structure with chronological sensor evidence.

Cairo has no door sensors. Its graph only weighted-F₁ winner suggests that relational structure over motion sensors compensates better than image or text alone. The model can encode repeated sensor identities, temporal proximity and spatial super node relationships even without explicit door events.

Kyoto has rich spatial metadata and many sensors. Its image-only weighted-F1 winner suggests that floorplan rasterization captures the most distribution-relevant evidence, while graph+text remains stronger under macro F1.

Milan has 10 classes and a chronological split. Its all four modality winner suggests that no single view is sufficient for balanced recognition. The class level table shows strong performance on several room specific classes but persistent weakness on *Eat*, *Take medicine* and *Leave Home*.

6.2 Limitations

- Default class spaces are not identical across houses. Aruba uses nine metadata classes, Cairo and Kyoto use seven standard classes, and Milan uses ten. This is appropriate for house specific activity support, but it means macro F1 values should be compared carefully across houses.
- Figures are generated from the framework’s visualization pipeline and should be interpreted together with the tables. Some figures emphasize selected checkpoints or validation style diagnostics, while the tables in this thesis report test set means across seeds.
- Rare class scores remain statistically fragile. Some rare activities have limited support and their F1 can shift substantially with the split seed, day selection or label remapping.
- Graphical user interface tools improve inspection and reproducibility but are not themselves evaluated through a formal user study. Their value in this thesis is practical: they reduce manual configuration and metadata errors during experimentation.

6.3 Future Work

Future work should evaluate window lengths beyond 10 and 30 events. The framework already supports event count windows, so a systematic window size study could measure how each activity changes as context grows.

The fusion module should also be analyzed at the class level. Learned per-class ensemble weights could reveal which classes trust sequence, text, image, graph, or fusion logits. This would turn the model from a performance system into a diagnostic tool.

Rare class objectives deserve more attention. Binary auxiliary objectives for *Take medicine*, *Leave Home*, or *Bed to toilet* may improve detection without sacrificing performance on common classes.

The method should be tested on additional CASAS houses and on resident transfer settings. A robust smart home HAR system should not merely perform well within a single house; it should adapt to new sensor layouts with minimal additional labels.

Finally, the simulator and annotation tools should also be extended into a more complete dataset onboarding workflow. A future version could validate sensor coverage, flag missing zone assignments, compare generated behavior images against floor plan geometry and export a metadata audit report before training begins.

6.4 Conclusion

This thesis produced a detailed report on the framework’s *main result set* experiments for CASAS Aruba, Cairo, Kyoto and Milan. The system converts each 30 event sensor window into sequence, text, image and graph representations that encode it with modality specific neural encoders and fuses active modalities with auxiliary heads, confidence gating and class-wise ensemble logits. Around the model, the project now also provides a reproducible experiment configuration system, desktop experiment controls, metadata annotation, smart-home playback simulation, smoke checks and result reporting.

The best balanced models are house specific. Aruba is best with graph+image+sequence, reaching weighted F1 0.850. Cairo is best with graph only input, reaching weighted F1 0.842. Kyoto is best with image input, reaching weighted F1 0.843. Milan performs best across all four modalities, achieving a macro F1 of 0.649. These results support the central claim of the thesis: multimodal representations are valuable for smart home HAR, but the best modality set depends on the house, label space and evaluation objective.

BIBLIOGRAPHY

- Aaron van den Oord, Yazhe Li, Oriol Vinyals. (2018). Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.
- Damien Bouchabou, Son Minh Nguyen, Christian Lohr, Benoit LeDuc, Ioannis Kanellos. (2021). 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. <https://doi.org/10.3390/s21186037>
- Daniele Liciotti, Michele Bernardini, Luca Romeo, Emanuele Frontoni. (2019). A sequential deep learning application for recognising human activities in smart homes. *Neurocomputing*, 396, 501–513. <https://doi.org/10.1016/j.neucom.2018.10.104>
- Diane J. Cook, Aaron S. Crandall, Brian L. Thomas, Narayanan C. Krishnan. (2013). Casas: A smart home in a box. *Computer*, 46(7), 62–69. <https://doi.org/10.1109/MC.2012.328>
- Junhao Zhao, Zishuai Liu, Ruili Fang, Jin Lu, Linghan Zhang, Fei Dou. (2025). Care: Contrastive alignment for adl recognition from event-triggered sensor streams. *arXiv preprint arXiv:2510.16988*.
- Munkhjargal Gochoo, Tan-Hsu Tan, Shing-Hong Liu, Fu-Rong Jean, Fady S. Alnajjar, Shih-Chia Huang. (2018). 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. <https://doi.org/10.1109/JBHI.2018.2833618>

APPENDIX A

FULL MULTIMODAL RANKINGS

This appendix lists all 15 proposed modality combinations for each house, ranked by macro-F1. Values are means and standard deviations over seeds 10, 30 and 50.

A.1 Aruba

Table A.1: Aruba full multimodal ranking by weighted F1.

Rank	Combination	Weighted F1	Macro F1
1	graph+image+sequence	0.850	0.735
2	image+sequence	0.847	0.725
3	graph+image+sequence+text	0.844	0.728
4	graph+image	0.843	0.723
5	graph+image+text	0.842	0.713
6	graph+sequence+text	0.842	0.727
7	image	0.840	0.704
8	image+text	0.837	0.695

Rank	Combination	Weighted F ₁	Macro F ₁
9	graph+sequence	0.837	0.734
10	sequence+text	0.836	0.716
11	graph+text	0.833	0.709
12	sequence	0.832	0.717
13	image+sequence+text	0.831	0.722
14	text	0.820	0.685
15	graph	0.817	0.708

A.2 Cairo

Table A.2: Cairo full multimodal ranking by weighted F₁.

Rank	Combination	Weighted F ₁	Macro F ₁
1	graph	0.842	0.524
2	graph+text	0.840	0.495
3	graph+sequence+text	0.838	0.512
4	graph+image+text	0.836	0.487
5	graph+sequence	0.830	0.550
6	graph+image+sequence	0.823	0.499
7	graph+image+sequence+text	0.823	0.499
8	sequence+text	0.809	0.514
9	graph+image	0.804	0.468
10	image+sequence+text	0.803	0.482
11	image+sequence	0.801	0.472

Rank	Combination	Weighted F ₁	Macro F ₁
12	sequence	0.789	0.516
13	image+text	0.785	0.445
14	text	0.766	0.442
15	image	0.743	0.375

A.3 Kyoto

Table A.3: Kyoto full multimodal ranking by weighted F₁.

Rank	Combination	Weighted F ₁	Macro F ₁
1	image	0.843	0.766
2	graph+text	0.841	0.777
3	graph+image+text	0.833	0.753
4	graph+image	0.820	0.768
5	text	0.816	0.759
6	graph+image+sequence	0.808	0.761
7	graph+image+sequence+text	0.801	0.744
8	sequence+text	0.798	0.741
9	graph+sequence	0.796	0.729
10	graph+sequence+text	0.789	0.742
11	image+text	0.785	0.720
12	image+sequence+text	0.771	0.728
13	image+sequence	0.769	0.725
14	sequence	0.763	0.685

Rank	Combination	Weighted F ₁	Macro F ₁
15	graph	0.743	0.680

A.4 Milan

Table A.4: Milan full multimodal ranking by macro-F₁.

Rank	Combination	Macro-F ₁	Weighted-F ₁
1	graph+image+sequence+text	0.649	0.700
2	graph+image+sequence	0.638	0.674
3	graph+sequence+text	0.636	0.678
4	image+sequence+text	0.633	0.696
5	image+sequence	0.629	0.681
6	graph+image+text	0.628	0.661
7	sequence+text	0.628	0.664
8	graph+image	0.624	0.670
9	graph+text	0.623	0.659
10	sequence	0.610	0.671
11	image+text	0.594	0.656
12	graph	0.586	0.646
13	text	0.570	0.645
14	graph+sequence	0.565	0.596
15	image	0.558	0.640

APPENDIX B

ACTIVITY REMAPPING

This appendix records the activity remapping metadata used to create canonical activity spaces. The first table lists the final default activity classes used for the main thesis results, including the retained *Other* class. The remaining tables list raw-label remaps from each house’s metadata YAML file. Raw labels not listed in a remap table are either already canonical or handled by the house-specific parsing and class-filtering code.

Table B.1: Final default activity classes.

House	Classes
Aruba	Meal_Preparation, Relax, Eating, Work, Sleeping, Bed_to_Toilet, Enter_Home, Leave_Home, Housekeeping
Cairo	Other, Work, Take_medicine, Sleep, Leave_Home, Eat, Bed_to_toilet
Kyoto	Other, Work, Sleep, Relax, Personal_hygiene, Cook, Bed_to_toilet
Milan	Other, Work, Take_medicine, Sleep, Relax, Leave_Home, Eat, Cook, Bed_to_toilet, Bathing