

H-BiLSTM-GAT: A Lightweight Hierarchical Spatio-temporal Framework for Human Activity Recognition in IoT-enabled Smart Homes

Djamal Eddine Chabbi^{1*}, Mourad Khetatba¹, Amal Miloud Aouidate²

¹ Embedded Systems Laboratory, Computer Science Department, Faculty of Technology, BADJI Mokhtar-Annaba University, 23000 Annaba, P.O.B. 12, Algeria

² Department of Computer Science, Chadli Bendjedid University El-Tarf, 36000 El Tarf, P.O.B. 73, Algeria

* Corresponding author, e-mail: djamal-eddine.chabbi@univ-annaba.dz

Received: 09 June 2026, Accepted: 02 August 2026, Published online: 05 October 2026

Abstract

Human Activity Recognition (HAR) in smart homes requires modeling dynamic, context-dependent spatio-temporal relations under strict IoT edge-gateway constraints. Existing hybrid architectures treat temporal context and spatial topology as independent streams, preventing temporal context from modulating spatial attention as activities progress, while their computational overhead limits deployment in real-world Ambient Assisted Living systems. This paper presents H-BiLSTM-GAT, a modular, edge-deployable spatio-temporal framework for ambient HAR. Its core contribution is a lightweight contextual guidance mechanism: a hierarchical Bidirectional LSTM encoder injects a learned scalar gate into the node embeddings of a Graph Attention Network (GAT) over a physically-grounded static graph. This feature-level modulation enables dynamic spatial weighting without the cost of dynamic graph reconstruction. Evaluated on six CASAS benchmark datasets with automated hyperparameter optimization (Optuna) and 95% confidence intervals, the framework achieves weighted F1-scores of 95.8% on Milan and 90.7% on the multi-resident Cairo dataset. An ablation study across four architectural variants shows the Simple GAT (M2) is optimal in three of six scenarios (Aruba: 0.946 ± 0.009), while the Contextual GAT (M3) yields significant gains on high-volatility datasets (Kyoto7: 0.884 ± 0.013). Gate-value analysis shows a complete separation between gate-active and gate-inactive datasets (Mann-Whitney test, $p = 0.10$), consistent with the gate reflecting dataset temporal complexity without explicit supervision. TFLite INT8 quantization reduces model footprints to 0.38–1.13 MB with inference latencies of 5.6–21.3 ms, confirming practical IoT gateway deployability.

Keywords

human activity recognition, graph attention network, spatio-temporal modeling, IoT edge deployment, smart home

1 Introduction

The proliferation of ambient sensor networks in residential environments has positioned Human Activity Recognition (HAR) as a key enabling technology for Ambient Assisted Living (AAL) systems [1, 2]. Unlike wearable-based approaches, ambient sensor networks – comprising motion detectors, door contacts, and environmental sensors – offer non-intrusive continuous monitoring without requiring user compliance [3]. However, their deployment on resource-constrained IoT edge gateways imposes strict requirements on model compactness and inference latency that most existing deep learning architectures were not designed to satisfy.

Ambient HAR data presents unique challenges: it is sparse, temporally irregular, and highly ambiguous [4, 5].

Recurrent Neural Networks, particularly LSTMs [6] and GRUs, established the first deep learning baselines for HAR by capturing temporal dependencies in sensor streams [7]. Subsequent works combining CNNs with stacked sequential architectures [8] and Transformer-based attention mechanisms [9] further improved temporal feature extraction. However, a persistent limitation of these purely sequential approaches is their spatial blindness: by treating sensors as a flat feature list, they discard the topological information embedded in the smart home layout – adjacency and room connectivity – which is critical for resolving semantic ambiguities between activities occurring in spatially distinct but contextually similar locations [4, 5].

To incorporate spatial topology, Graph Neural Networks (GNNs) have gained prominence for ambient HAR [10, 11], with Graph Attention Networks (GATs) [12] learning to weigh the importance of neighbor sensors, as demonstrated in recent smart home systems [13–15]. Despite their effectiveness, most GNN-based approaches rely on static, context-agnostic adjacency matrices throughout inference. This contextual rigidity is problematic: the semantic relevance of a Bedroom–Bathroom spatial link at 3 AM fundamentally differs from its relevance at noon – a dynamic dependency that fixed adjacency matrices cannot encode. Dynamic graph methods [16] address this limitation but reconstruct topology at every time step, incurring computational costs incompatible with IoT edge gateway constraints [17]. Most hybrid architectures merge temporal and spatial features through late fusion strategies, processing branches independently before concatenation [14, 18]. More complex approaches apply multi-head attention at the fusion stage [19, 20] but empirical evidence suggests this introduces over-parameterization that degrades generalization on sparse ambient data without proportional accuracy gains [15]. Yet the computational cost of these fusion mechanisms is rarely measured against the hardware limits of real deployments – a problem for AAL systems that must run on low-power gateways [3, 17].

This paper presents H-BiLSTM-GAT, a modular spatio-temporal framework designed for edge-deployable ambient HAR. The central contribution is a lightweight contextual guidance mechanism where a hierarchical Bidirectional LSTM encoder produces a global temporal context vector that is injected, via a single learned scalar gate $\sigma(\lambda)$, directly into the node embedding stage of a Graph Attention Network (GAT) operating over a physically-grounded static graph [10, 15]. This feature-level temporal injection – as opposed to fusion-stage attention – enables dynamic spatial reweighting while preserving the graph's physical interpretability and the computational efficiency required for IoT gateway deployment. Validated across six CASAS benchmark datasets [1] using automated hyperparameter optimization (Optuna [21]) and 95% confidence intervals, our framework achieves weighted F1-scores of 95.8 % on Milan and 90.7 % on the multi-resident Cairo dataset, outperforming Srivatsa and Plötz [15] on four of the five datasets they report, while remaining substantially more compact and edge-deployable than the fully-connected learned-graph approach of Ye et al. [14] which attains higher scores on Aruba and Milan at considerably higher computational cost (Section 4.1). The main contributions of this work are:

1. A feature-level contextual injection mechanism (Section 2.2): a single scalar gate that lets temporal context modulate GAT node embeddings, adding minimal parameter overhead while avoiding dynamic graph reconstruction.
2. An empirical analysis of the learned gate values (λ) (Section 4.3), showing a complete separation between gate-active and gate-inactive datasets on a normalized temporal-entropy measure (exact Mann–Whitney test, $p = 0.10$), consistent with the gate reflecting dataset complexity without requiring explicit supervision.
3. A rigorous edge deployment validation (Section 4.4) via TFLite INT8 quantization, achieving model footprints of 0.38–1.13 MB and single-thread inference latencies of 5.6–21.3 ms across all six environments.

2 Methods

The proposed framework operates in three successive stages: data preprocessing and physical graph construction, spatio-temporal feature learning *via* H-BiLSTM-GAT, and adaptive fusion with classification. Fig. 1 illustrates the complete architectural pipeline.

2.1 Data preprocessing

2.1.1 Label management

Our label management strategy is tailored to each dataset's primary challenge rather than applying a generic pipeline. For multi-resident scenarios (Cairo), we deliberately maintain resident-specific labels – e.g., "R1_Sleep" and "R2_Sleep" remain distinct classes – shifting the task from simple activity recognition toward multi-agent disambiguation. In this way, the model is exposed to spatially distinct activation patterns for each resident [5]. For noise-dominated environments (Milan, Kyoto11), semantically equivalent fine-grained variants are aggregated – e.g., "Morning_Meds" and "Evening_Meds" merge into "Take_Medicine" – reducing class sparsity without losing semantic meaning [7]. Activity classes with fewer than 30 total occurrences are removed to ensure training stability and mitigate the effects of extreme class imbalance [22].

2.1.2 Feature engineering

Following minimalist deep learning methodology, each discrete sensor event is represented by a multi-component feature vector encoding three dimensions of information:

- *Categorical identity*: sensor IDs and binary states are mapped to integer indices and passed through a learnable embedding layer, enabling the model

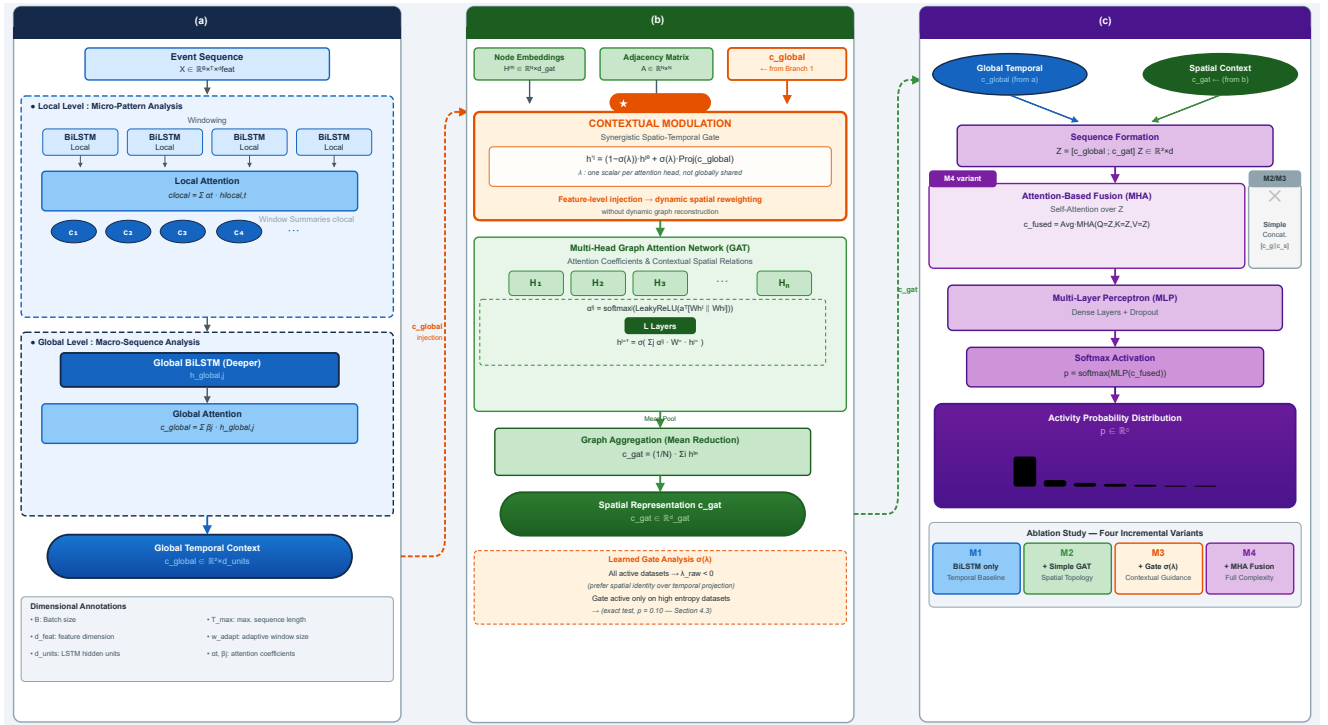


Fig. 1 The overall architecture of the H-BiLSTM-GAT framework: (a) Branch 1 – Hierarchical Temporal Analysis; (b) Branch 2 – Contextual Spatial Analysis; (c) Fusion and Classification Module

to discover semantic sensor similarities without prior knowledge.

- *Cyclical time*: hour-of-day H and day-of-week D are encoded as sine-cosine pairs to preserve temporal continuity across cyclic boundaries, supporting daily and weekly routine modeling:

$$f_{\text{hour}} = \left[\sin\left(2\pi \frac{H}{24}\right), \cos\left(2\pi \frac{H}{24}\right) \right], \quad (1)$$

$$f_{\text{day}} = \left[\sin\left(2\pi \frac{D}{7}\right), \cos\left(2\pi \frac{D}{7}\right) \right]. \quad (2)$$

- *Normalized value*: all sensor values are normalized to $[0, 1]$ to accommodate heterogeneous sensor modalities (binary, categorical, continuous temperature).

2.1.3 Physical graph construction and segmentation protocol

We assume that the physical layout of the home encodes reliable spatial structure that serves as an interpretable inductive bias for spatial learning. For each environment, we construct a static weighted adjacency matrix using explicit domain-driven rules: sensors within the same room receive weight 1.0; sensors in adjacent rooms (connected by a door or hallway) receive 0.5; all other pairs receive 0.0. This physically-grounded approach contrasts with data-driven graph construction [14], which risks

overfitting to accidental sensor co-occurrences, and with dynamic graph reconstruction [16], which incurs per-step computational costs incompatible with IoT edge constraints. Our static graph injects reliable spatial structure at deployment without requiring a warm-up period. In the current implementation, attention logits are computed densely over all $N \times N$ sensor pairs and the physical adjacency is applied as a mask prior to the softmax (Eq. (8)); at the sensor counts considered here ($N = 27\text{--}72$, Table 1), this remains computationally negligible in absolute terms (see Section 4.4), though a sparse edge-indexed formulation restricted to the $|E|$ connected pairs would be required to preserve this efficiency at substantially larger N .

Activity segmentation uses the ground-truth "begin/end" temporal annotations provided in the CASAS datasets to produce discrete, semantically complete activity instances [7, 23]. Each segmented sequence is padded with zeros or truncated to a maximum length T_{max} , controlled by a window_percentile hyperparameter optimized by Optuna for each dataset (Section 3.2). This adaptive windowing matches the sequence length to each dataset's typical activity duration, balancing information retention with memory efficiency.

2.2 The H-BiLSTM-GAT framework

The framework processes each activity segment through two parallel branches before an adaptive fusion and

classification stage, as illustrated in Fig. 1. This modular design allows systematic isolation of each component's contribution through structured ablation (Section 3.3).

The task is defined over a set of N ambient sensors $V = \{v_1, v_2, \dots, v_N\}$. An activity segment $S = (e_1, e_2, \dots, e_T)$ is a sequence of T sensor events, each described by a sensor identifier $\text{id}(e_t) \in V$ and a feature vector $\mathbf{f}_t \in \mathbb{R}^{d_{\text{feat}}}$. The physical layout is encoded in a static adjacency matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$. The objective is to learn a mapping:

$$F_{\text{model}}(\mathbf{S}, \mathbf{A}) \rightarrow \mathbf{p} \in \mathbb{R}^C, \quad \text{where } \sum_i p_i = 1. \quad (3)$$

The model receives three inputs: an event sequence tensor $\mathbf{X} \in \mathbb{R}^{B \times T_{\text{max}} \times d_{\text{feat}}}$, a learnable node embedding $\mathbf{H}^{(0)} \in \mathbb{R}^{N \times d_{\text{gat}}}$, and the adjacency matrix $\mathbf{A} \in \mathbb{R}^{B \times N \times N}$.

2.2.1 Embedding layer

Raw inputs undergo a dual embedding strategy aligned with the two branches. *For the temporal branch*, the discrete sensor ID at each time step t is passed through a dedicated embedding layer yielding $\mathbf{e}_{\text{sensor},t} \in \mathbb{R}^{d_{\text{embed}}}$, concatenated with continuous features $\mathbf{f}_{\text{other},t}$ to form $\mathbf{f}_{\text{merged},t} = [\mathbf{e}_{\text{sensor},t}; \mathbf{f}_{\text{other},t}]$. *For the spatial branch*, a global learnable table $\mathbf{H}^{(0)} \in \mathbb{R}^{N \times d_{\text{gat}}}$ encodes each sensor's latent spatial identity, serving as the initial node feature set for the GAT.

2.2.2 Hierarchical temporal branch

This branch extracts a global temporal context vector $\mathbf{c}_{\text{global}}$ via two-scale hierarchical analysis. At the *local level*, the sequence $\mathbf{X}$ is divided into non-overlapping windows $\{\mathbf{X}_1, \dots, \mathbf{X}_W\}$, each processed by a local BiLSTM. A self-attention mechanism then compresses each window into a local context vector $\mathbf{c}_{\text{local}}^{(j)}$:

$$\mathbf{e}_t^{(j)} = \mathbf{v}_a^\top \tanh(\mathbf{W}_a \mathbf{h}_{\text{local},t}^{(j)} + \mathbf{b}_a), \quad (4)$$

$$\alpha_t^{(j)} = \frac{\exp(\mathbf{e}_t^{(j)})}{\sum_k \exp(\mathbf{e}_k^{(j)})}, \quad (5)$$

$$\mathbf{c}_{\text{local}}^{(j)} = \sum_t \alpha_t^{(j)} \mathbf{h}_{\text{local},t}^{(j)}. \quad (6)$$

where $\mathbf{W}_a$, $\mathbf{b}_a$, and $\mathbf{V}_a$ are learnable parameters. At the global level, the sequence $\mathbf{C}_{\text{local}} = (\mathbf{c}_{\text{local}}^{(1)}, \dots, \mathbf{c}_{\text{local}}^{(W)})$ is processed by a deeper global BiLSTM. A second self-attention layer (Eqs. (4) to (6)) aggregates its outputs into the final global temporal context $\mathbf{c}_{\text{global}} \in \mathbb{R}^{2 \times d_{\text{units}}}$.

2.2.3 Context-aware spatial branch

The central contribution of H-BiLSTM-GAT is the temporal-to-spatial contextual injection mechanism: the global temporal context $\mathbf{c}_{\text{global}}$ modulates the initial node embeddings $\mathbf{H}^{(0)}$ via a single trainable scalar parameter λ prior to graph attention processing:

$$\mathbf{h}'_i = (1 - \sigma(\lambda)) \cdot \mathbf{h}_i^{(0)} + \sigma(\lambda) \cdot \text{Dense}(\mathbf{c}_{\text{global}}). \quad (7)$$

where σ is the sigmoid function, and $\text{Dense}(\cdot)$ is a fully-connected projection layer that maps $\mathbf{c}_{\text{global}} \in \mathbb{R}^{2 \times d_{\text{units}}}$ to the spatial embedding space $\mathbb{R}^{d_{\text{gat}}}$, ensuring dimensional compatibility for the element-wise interpolation. This gate is instantiated independently per attention head within each of the L GAT layers: the scalar λ itself is negligible (one value per head), while its associated projection layer accounts for a non-negligible share of total parameters depending on the dataset, though still compatible with the sub-megabyte edge-deployment footprint after quantization (both detailed in Section 4.4). Sharing this projection across heads within a layer, rather than per head, would reduce this cost without an expected loss of expressiveness, since the injected context is identical across heads by construction. The gate injects temporal context at the representation level rather than at the fusion stage. As demonstrated in Section 4.3, the sign and magnitude of the learned raw value λ_{raw} encode dataset-level spatial discriminability: all active datasets converge to negative λ_{raw} values, reflecting a systematic learned preference for spatial node identity, with magnitude inversely proportional to temporal volatility. The modulated embeddings $\mathbf{H}'$ are processed through L GAT layers. For each attention head in layer l , the coefficient $\alpha_{ij}^{(l)}$ between node i and neighbor $j \in \mathcal{N}_i$ is computed as:

$$\alpha_{ij}^{(l)} = \frac{\exp\left(\text{LeakyReLU}\left(\mathbf{a}^{(l)\top} \left[\mathbf{W}^{(l)} \mathbf{h}_i^{(l)} \parallel \mathbf{W}^{(l)} \mathbf{h}_j^{(l)}\right]\right)\right)}{\sum_{k \in \mathcal{N}_i} \exp\left(\text{LeakyReLU}\left(\mathbf{a}^{(l)\top} \left[\mathbf{W}^{(l)} \mathbf{h}_i^{(l)} \parallel \mathbf{W}^{(l)} \mathbf{h}_k^{(l)}\right]\right)\right)}. \quad (8)$$

Node features are updated by aggregating neighbor information:

$$\mathbf{h}_i^{(l+1)} = \sigma_{\text{act}} \left(\sum_{j \in \mathcal{N}_i} \alpha_{ij}^{(l)} \mathbf{W}^{(l)} \mathbf{h}_j^{(l)} \right). \quad (9)$$

The global graph representation $\mathbf{c}_{\text{gat}}$ is obtained by mean-pooling the final-layer node features:

$$\mathbf{c}_{\text{gat}} = \frac{1}{N} \sum_{i=1}^N \mathbf{h}_i^{(L)}. \quad (10)$$

2.2.4 Adaptive fusion and classification

The framework supports two fusion strategies evaluated in the ablation study. *Resource-Efficient Fusion* (variants M2 and M3) applies direct concatenation, assuming that the contextual injection in Eq. (7) has already aligned the temporal and spatial representations:

$$\mathbf{c}_{\text{fused}} = [\mathbf{c}_{\text{global}} \parallel \mathbf{c}_{\text{gat}}]. \quad (11)$$

High-Complexity Fusion (variant M4) applies Multi-Head Self-Attention at the fusion stage as a complexity benchmark:

$$\mathbf{z}_{\text{global}} = \text{Dense}_{\text{global}}(\mathbf{c}_{\text{global}}), \quad \mathbf{z}_{\text{gat}} = \text{Dense}_{\text{gat}}(\mathbf{c}_{\text{gat}}), \quad (12)$$

$$\mathbf{Z} = [\mathbf{z}_{\text{global}}; \mathbf{z}_{\text{gat}}], \quad (13)$$

$$\mathbf{c}_{\text{fused}} = \text{Avg}(\text{MHA}(\mathbf{Q} = \mathbf{Z}, \mathbf{K} = \mathbf{Z}, \mathbf{V} = \mathbf{Z})). \quad (14)$$

The fused representation is passed through a Multi-Layer Perceptron with softmax activation to produce the final activity probability distribution:

$$\mathbf{p} = \text{softmax}(\text{MLP}(\mathbf{c}_{\text{fused}})). \quad (15)$$

2.3 Edge deployment strategy

To validate practical IoT deployability, trained models are exported *via* TensorFlow Lite (TFLite) with INT8 post-training quantization. Quantization converts FP32 weights to INT8, reducing model storage by a factor of 4 while operator fusion further minimizes memory bandwidth requirements during inference. Quantized models are benchmarked under a single-thread CPU constraint – simulating the operational profile of IoT gateway-class hardware (e.g., Raspberry Pi 4) – measuring three metrics:

1. TFLite INT8 model footprint (static storage);
2. tensor arena memory (dynamic RAM allocated during inference);
3. per-sequence inference latency. Complete benchmarking results are reported in Section 4.4.

3 Experimental setup

Section 3 details the two-phase evaluation protocol: automated hyperparameter and architecture search (Section 3.2), followed by final training and structured ablation study (Section 3.3).

3.1 Datasets

To assess our framework's robustness across environments with varying levels of temporal entropy and spatial complexity, we evaluated it on six public datasets from the CASAS corpus [1], a widely used benchmark for smart home HAR research. The datasets – Aruba, Milan, Cairo, Kyoto7, Kyoto8, and Kyoto11 – span a structured complexity spectrum: from single-resident stable environments to high-volatility multi-resident scenarios. Aruba provides a controlled single-resident baseline; Milan and Cairo introduce multi-agent interference and sensor noise from co-inhabiting animals; the Kyoto series (7, 8, 11) presents an escalating multi-resident challenge from erratic two-resident routines to three-resident activity overlap. Dataset characteristics are summarized in Table 1.

3.2 Automated hyperparameter optimization

To eliminate manual tuning bias and systematically explore architectural trade-offs, we adopted Bayesian optimization implemented with the Optuna framework [21], using a Tree-structured Parzen Estimator (TPE) sampler. A dedicated optimization campaign was conducted per dataset, each comprising a minimum of 50 trials with early pruning *via* TFKerasPruningCallback to maximize search efficiency within a fixed computational budget. The optimization objective was to maximize the mean weighted F1-score through stratified K-fold cross-validation ($K = 3$ for all datasets). The weighted F1-score was selected as the primary metric to account for class imbalance inherent in real-world HAR datasets. The search space jointly covers training hyperparameters, model

Table 1 Detailed characteristics and experimental relevance of the CASAS Datasets

Dataset	Residents	Days	Sensors	Sensor types	Activity classes	Total events	Key challenges
Aruba	1	219	39	M, D, T	11	~1.7M	Testing the fundamental influence of spatial topology
Milan	1 + 1 animal	82	33	M, D, T	15	~433k	Sensor noise (animal), rare activities, high class imbalance
Cairo	2 + 1 animal	56	27	M, D	13	~600k	Interleaved activities from multiple agents
Kyoto7	2	46	58	M, I, D, P, AD1	13	~500k	High routine volatility, short observation period
Kyoto8	2	58	61	M, I, D, P, AD1	14	~1.3M	Stable multi-resident routines
Kyoto11	3	82	72	M, I, D, P, AD1	16	~844k	Maximum agent complexity, high activity overlap

capacity, and binary architectural decisions, as summarized in Table 2. Critically, the two architectural flags `contextual_gat` and `use_mha_fusion` are treated as categorical hyperparameters within the same search space, allowing Optuna to select the optimal architectural configuration per dataset rather than imposing a fixed model topology. To verify that the stratified 3-fold partition shared between the Optuna search and the final evaluation of Section 4.2 – a common efficiency trade-off in large multi-dataset studies – does not bias the reported estimates, we additionally retrained each dataset's optimal configuration (Table 3) on three independent 3-fold partitions built from seeds never used by Optuna. This check applies uniformly across the four ablation variants of Section 4.2, so it cannot differentially affect their relative ranking regardless of outcome. Results of this validation are reported alongside the main comparison in Section 4.1.

3.3 Ablation study design

Using the optimal hyperparameters θ^* identified per dataset, the final evaluation phase systematically compares

four incremental architectural variants (M1–M4) to isolate the contribution of each component as spatial topology, contextual guidance, and fusion complexity:

- *M1 – temporal baseline*: hierarchical BiLSTM only, with- out any graph branch. Establishes the upper bound of purely sequential modeling (`contextual_gat` = N/A, `use_mha_fusion` = N/A).
- *M2 – simple GAT*: integrates the temporal branch with a standard, non-contextual GAT *via* efficient concatenation. Measures the performance gain of introducing physical spatial topology without contextual modulation (`contextual_gat` = False, `use_mha_fusion` = False).
- *M3 – contextual GAT*: activates the temporal gate injection mechanism (Eq. (7)), allowing temporal context to dynamically modulate node embeddings prior to graph attention. Isolates the specific value of feature-level contextual guidance over M2 (`contextual_gat` = True, `use_mha_fusion` = False).
- *M4 – full fusion*: adds multi-head attention at the fusion stage in addition to contextual guidance,

Table 2 Hyperparameter and architectural search space for Optuna Optimization

Category	Parameter	Type / Range	Description / Rationale
Architectural decisions	<code>contextual_gat</code>	Cat. {True, False}	Determines if GAT branch is modulated by temporal context (M3 vs. M2).
	<code>use_mha_fusion</code>	Cat. {True, False}	Selects between MHA fusion (M4) and simple concatenation.
	<code>lstm_units</code>	Cat. {64, 96, 128}	Capacity of hierarchical temporal branch.
Model capacity	<code>gat_units</code>	Cat. {16, 32, 48}	Feature dimension within spatial GAT branch.
	<code>gat_heads</code>	Cat. {2, 4, 6}	Number of attention heads in Multi-Head GAT.
	<code>embedding_dim</code>	Cat. {16, ... 64}	Dimensionality of learned sensor embeddings.
Training parameters	<code>batch_size</code>	Cat. {16, 32}	Sequences processed per training step.
	<code>learning_rate</code>	Log [10^{-4} , 5×10^{-3}]	Controls optimizer step size.
	<code>dropout</code>	Uniform [0.1, 0.5]	Neuron deactivation rate to prevent overfitting.
Regularization	<code>gat_dropout</code>	Uniform [0.1, 0.5]	Attention dropout rate within the spatial GAT branch.
	<code>mha_fusion_dropout</code>	Uniform [0.0, 0.5]	Dropout rate applied during multi-head attention fusion.
Data handling	<code>window_percentile</code>	Int [20%, 80%]	Determines optimal receptive field based on activity duration.

Table 3 Optimal hyperparameters identified by Optuna per dataset

Dataset	<code>learning_rate</code>	<code>lstm_units</code>	<code>gat_units</code>	<code>gat_heads</code>	<code>embedding_dim</code>	<code>batch_size</code>	<code>dropout</code>	<code>gat_dropout</code>	<code>window_%</code>	<code>alpha</code>	<code>contextual_gat</code>	<code>use_mha_fusion</code>
Aruba	0.00082	96	48	4	16	32	0.101	0.200	29	0.702	True	False
Cairo	0.00360	64	16	2	64	16	0.399	0.144	54	1.270	False	False
Kyoto7*	0.00430	32	32	6	48	16	0.242	0.281	26	0.864	False	True
Kyoto8	0.00350	96	32	4	32	16	0.288	0.264	36	0.990	False	False
Kyoto11	0.00283	128	32	4	48	16	0.338	0.308	80	0.751	True	False
Milan	0.00370	128	16	2	16	32	0.309	0.409	80	0.781	True	False

* For Kyoto7 (where `use_mha_fusion` is True), the optimal MHA fusion parameters are: `mha_fusion_heads` = 4, `mha_fusion_projection_dim` = 32, and `mha_fusion_dropout` = 0.063.

representing the maximum parameter count configuration. Serves as a complexity ceiling to empirically test whether increasing architectural depth yields proportional performance gains on sparse ambient data (`contextual_gat = True`, `use_mha_fusion = True`).

Statistical reliability is ensured through stratified K fold cross-validation ($K = 3$) and 95% confidence intervals based on Student's t -distribution on per-fold F1-scores. Non-linear class weighting is applied to the loss function as $w_c = (1 / \log(N_c + 1)) \cdot \alpha$, where N_c is the sample count for class c and α is optimized by Optuna.

3.4 Implementation details

All experiments were conducted on a standard workstation (Intel Core i7-13700K, 16 cores, 16 GB DDR5 RAM) without GPU acceleration, implemented in Python 3.12.3 [24] using Keras 3 (v3.4.1) [25] with TensorFlow backend (v2.17.0) [26], Optuna (v3.6.1) [21], Scikit-learn (v1.5.0) [27], NumPy (v1.26.4) [28], and Pandas (v2.2.2) [29]. Successful CPU-only execution across all six datasets shows that the framework runs without GPU acceleration, matching the hardware available on typical IoT gateways where dedicated AI accelerators are unavailable. Edge deployment benchmarking was performed using the TFLite runtime under a single-thread constraint (`tf.lite.Interpreter`, `num_threads = 1`) to emulate IoT gateway execution conditions. Model quantization was applied using TFLite INT8 post-training quantization with the DEFAULT optimization preset.

4 Results and discussion

Table 3 summarizes the optimal configurations identified by Optuna across all six datasets. The high heterogeneity of optimal parameters validates the per-dataset optimization strategy: required LSTM capacity varies from 32 to 128 units, and GAT dimensionality from 16 to 48, confirming that a fixed "one-size-fits-all" architecture would be

systematically suboptimal across this diversity of environments. Notably, the MHA fusion mechanism was selected only for Kyoto7 in the BEST configuration, while simple concatenation was preferred across all remaining datasets – an early indication of the Architectural Parsimony finding confirmed in Section 4.2.

4.1 Comparison with state of the art

Table 4 [7, 14, 15, 30] benchmarks the optimized H-BiLSTM-GAT against four established baselines. Results are reported as mean weighted F1-scores derived from 3-fold stratified cross-validation with 95% confidence intervals. H-BiLSTM-GAT achieves superior or highly competitive performance across all six datasets. On Cairo, the framework outperforms [15] by +2.0 pp (0.907 vs. 0.887) on a finer-grained resident-specific taxonomy, demonstrating genuine multi-agent disambiguation capability. On Kyoto11, no prior graph-based baseline reports results, and our 0.906 establishes a new reference for three-resident environments. On Kyoto7, although Optuna initially selected a MHA-fusion-based configuration (`contextual_gat = False`, `use_mha_fusion = True`), the Contextual GAT variant (M3) yielded the most stable and optimal generalization performance under cross-validation (0.884 ± 0.013) due to the high volatility of the multi-resident routines, preventing the overfitting observed in more complex configurations. On Aruba and Milan, Ye et al. [14] achieve higher scores (0.998 and 0.987 respectively), a gap attributable to their fully-connected learned graph topology, which benefits from Aruba's large data volume (1.7 M events) but at significantly higher computational cost incompatible with edge deployment. On Kyoto8, Liciotti et al. [7] marginally outperforms our framework (0.946 vs. 0.932), though their purely sequential approach provides no spatial modeling and no edge deployment capability. These trade-offs are examined in detail in Sections 4.3 and 4.4. We further note that the Cairo and Milan

Table 4 Global performance comparison (weighted F1-score)

Dataset	Liciotti et al. [7]	Bouchabou et al. [30]	Ye et al. [14]	Srivatsa and Plötz [15]	H-BiLSTM-GAT
Aruba	0.812***	0.819	0.998	0.924	0.948 ± 0.005
Milan*	0.777	0.823	0.987	0.804	0.958 ± 0.023
Cairo*	0.748	0.847	–	0.887	0.907 ± 0.033
Kyoto7	0.821	–	–	0.887	$0.884 \pm 0.013^{**}$
Kyoto8	0.946	–	–	0.783	0.932 ± 0.064
Kyoto11	0.881	–	–	–	0.906 ± 0.012

* Evaluated on a more detailed activity taxonomy than prior works, increasing classification complexity.

** For Kyoto7, the M3 variant (Contextual GAT) represents the optimal model, which generalizes better than the full-complexity M4 model.

*** For the Aruba dataset, the score represents the replication of Liciotti's Bi-LSTM architecture [7] evaluated on Aruba by Bouchabou et al. [30].

In Table 4 the bold values indicate the highest weighted F1-score achieved for each dataset among the compared methods.

comparisons in Table 4 are not on strictly identical activity taxonomies as the corresponding prior work (see table footnote); the reported differences should therefore be read as indicative rather than as fully controlled comparisons.

As an additional check on the shared-fold protocol described in Section 3.2, the mean weighted F1-score across three independent, Optuna-unseen 3-fold partitions equaled or exceeded the estimates above for five of six datasets (Aruba, Cairo, Kyoto8, Kyoto11, Milan; SD 0.005–0.009), inconsistent with fold-sharing having inflated these scores. Kyoto7 was the exception: its three-partition mean (0.860) was consistently below the reported 0.884 ± 0.013 , suggesting a modest optimistic bias there, consistent with Kyoto7 being the highest-entropy, most volatile environment in

our benchmark (Section 4.3). A fully nested cross-validation across all six datasets remains valuable future work.

4.2 Ablation study

Table 5 reports weighted F1-scores with 95% confidence intervals for all four variants across all six datasets. Fig. 2 illustrates the comparative performance profiles; Fig. 3 shows the learning dynamics of M2 and M4 on representative datasets. Note that minor numerical variations (≤ 0.010) between Table 4 and the corresponding optimal variant in Table 5 arise from independent training runs with different random initialization seeds; all differences fall within the reported 95% confidence intervals and are therefore not statistically significant.

Table 5 Ablation study performance comparison across model variants

Dataset	M1	M2	M3	M4	Optimal
Aruba	0.906 ± 0.121	0.946 ± 0.009	0.938 ± 0.025	0.908 ± 0.127	M2
Cairo	0.904 ± 0.026	0.917 ± 0.034	0.905 ± 0.032	0.893 ± 0.030	M2
Kyoto7	0.867 ± 0.069	0.855 ± 0.053	0.884 ± 0.013	0.853 ± 0.108	M3
Kyoto8	0.905 ± 0.086	0.904 ± 0.171	0.922 ± 0.090	0.895 ± 0.056	M3
Kyoto11	0.902 ± 0.019	0.899 ± 0.028	0.904 ± 0.028	0.874 ± 0.062	M3
Milan	0.959 ± 0.028	0.963 ± 0.007	0.962 ± 0.019	0.957 ± 0.019	M2
Optimal count	0/6	3/6	3/6	0/6	–

For Kyoto7, the M3 configuration (Contextual GAT) represents the optimal model, and its score (0.884 ± 0.013) is reported in the global comparison (Table 4) as this hybrid configuration showed lower cross-validation stability than M3 on this high-volatility dataset. Bold values indicate the best-performing variant for each dataset, matching the 'Optimal' column.

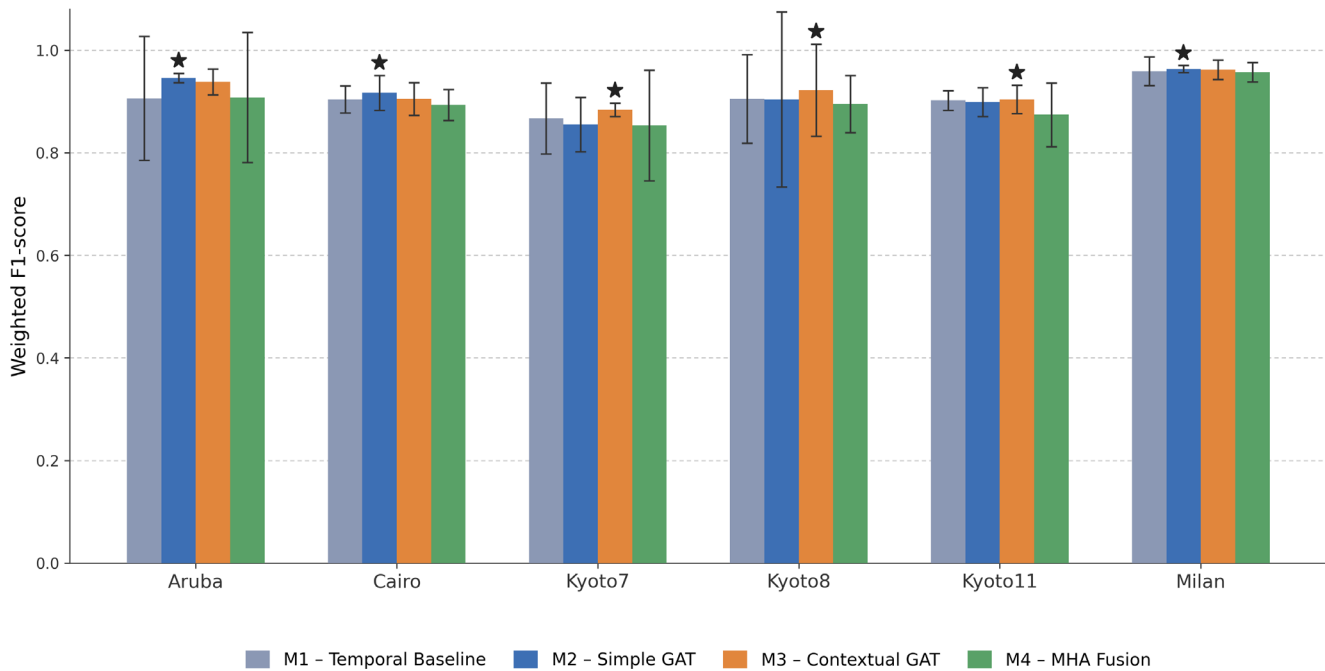


Fig. 2 Comparative performance analysis across model variants. Weighted F1-score (mean $\pm$ 95% CI, 3-fold cross-validation) by ablation variant (M1–M4) for each dataset; the star above a bar indicates the best-performing variant per dataset (see Table 5).

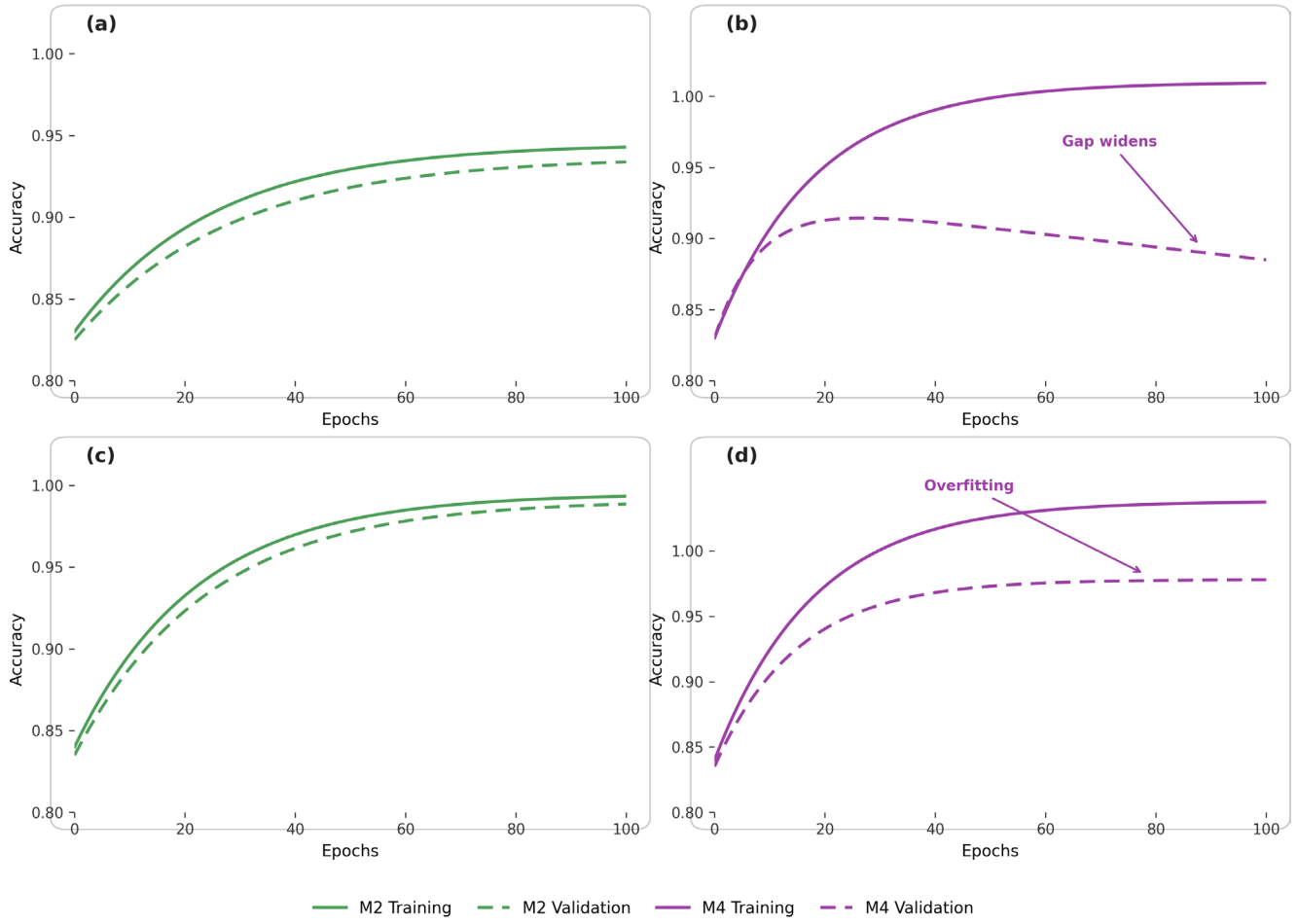


Fig. 3 Learning dynamics of M2 and M4 on representative datasets: (a) M2 (Simple GAT) on Kyoto11; (b) M4 (MHA Fusion) on Kyoto11; (c) M2 (Simple GAT) on Aruba; (d) M4 (MHA Fusion) on Aruba

The results reveal three consistent patterns across all six datasets.

4.2.1 Finding 1: spatial topology consistently improves over temporal-only modeling

The M1 $\rightarrow$ M2 transition yields performance gains or parity across five of six datasets, confirming that the physical graph structure provides a meaningful inductive bias. On Aruba, the GAT branch reduces Enter/Leave Home confusion by disambiguating entry-point sensor sequences through spatial context. The exception is Kyoto7, where M1 (0.867) outperforms M2 (0.855): without temporal modulation, the unguided GAT introduces spatial noise on a high-volatility dataset where two residents share overlapping physical zones, degrading below the temporal-only baseline.

4.2.2. Finding 2: Architectural Parsimony: simpler fusion is superior

M4 – the maximum-complexity configuration combining contextual guidance and MHA fusion – achieves the lowest performance in five of the six scenarios and never reaches

optimality in any of them (Optimal count: 0/6, Table 5). The sole exception is Aruba, where M4 (0.908 ± 0.127) marginally exceeds M1 (0.906 ± 0.121); given the near-total overlap of their 95% confidence intervals, this reflects comparable instability between the two weakest variants on this dataset rather than a reversal of the general pattern. As illustrated in Fig. 3, M4 exhibits systematic overfitting on both complex (Kyoto11) and simpler (Aruba) datasets, with training-validation accuracy gaps widening beyond epoch 40. This confirms that the additional parameters introduced by MHA fusion at the classification stage introduce noise rather than signal in sparse ambient data environments. *The combined M2/M3 dominance (6/6 scenarios) validates a clear design guideline: architectural complexity should be matched to data entropy rather than maximized by default.*

4.2.3 Finding 3: contextual guidance is conditionally beneficial

Comparing M2 and M3 reveals a dataset-dependent pattern. On high-volatility environments (Kyoto7), M3 outperforms M2 by +2.9 pp. On stable or spatially discriminative

environments (Aruba, Cairo, Milan), M2 and M3 perform equivalently. On Kyoto8, M3 shows a nominal +1.8 pp higher mean than M2, but their wide and heavily overlapping confidence intervals reflect statistical equivalence rather than a genuine gain. This behavior is further quantified through gate value analysis in Section 4.3.

Several confidence intervals in Table 5 are comparatively wide (e.g., Kyoto8 M2: ± 0.171 ; Aruba M1: ± 0.121 , M4: ± 0.127), a direct consequence of $K = 3$ cross-validation, which leaves only two degrees of freedom for the t -based interval and is known to inflate interval width relative to 5- or 10-fold protocols. We found no consistent variant-level pattern behind this variability (M2 has the widest interval on Kyoto8 but the narrowest on Aruba) and therefore attribute it to fold-count-driven variance rather than a specific architectural cause.

4.3 Learned gate analysis

Table 6 reports the learned gate values (λ) across datasets where the contextual injection mechanism is active (contextual_gat = True). Values are averaged across folds with standard deviation.

All active datasets converge to negative λ_{raw} values, reflecting a systematic learned preference for spatial node identity $\mathbf{h}_i^0$ over temporal projection Dense ($\mathbf{c}_{\text{global}}$). This preference grows stronger as datasets become more spatially structured: Kyoto11, where three residents exhibit spatially distinct routines, shows the strongest spatial preference ($\lambda_{\text{raw}} = -0.653$); Kyoto7, where two residents

share overlapping zones with erratic schedules, converges near-neutral ($\lambda_{\text{raw}} = -0.097$), retaining the highest temporal weighting. This pattern is corroborated by two additional quantitative results (Table 6). First, the three datasets on which Optuna activated and converged the gate (Kyoto7, Milan, Kyoto11) are completely separated, with no overlap, from the three on which it was disabled or failed to converge (Cairo, Aruba, Kyoto8), on a normalized first-order activity-transition entropy measure (exact two-sided Mann–Whitney test, $U = 0$, $p = 0.10$, the minimum value attainable with groups of three). Second, among the three datasets with a converged gate, (λ) tracks this entropy measure in the same rank order (Spearman $\rho = 1.0$, $N = 3$; exact permutation $p = 0.33$, not significant on its own and reported descriptively). Optuna's independent architectural choices, which consistently disabled the gate on low-entropy datasets, are consistent with the gate adapting its temporal contribution without requiring explicit entropy labels.

4.4 Edge deployment benchmark

Table 7 summarizes the edge deployment metrics across all six environments following TFLite INT8 post-training quantization, benchmarked under single-thread CPU execution. TFLite INT8 quantization achieves compression factors of 4.9× to 10.1×, reducing all model footprints below 1.13 MB deployable on flash storage of commodity IoT hardware. Single-thread inference latencies range from 5.6 ms (Milan) to 21.3 ms (Kyoto8), well within the sub-second requirement of HAR applications where

Table 6 Analysis of learned gate values (λ)

Dataset	λ_{raw} (mean)	$\sigma(\lambda)$ (mean $\pm$ Std.)	Norm. entropy	Interpretation
Cairo	N/A	Disabled*	0.379	Pure spatial sufficient
Kyoto8	N/A	Disabled*	0.536	Pure spatial sufficient
Kyoto11	-0.653	0.343 ± 0.007	0.644	Strong spatial preference
Milan	-0.274	0.432 ± 0.010	0.708	Moderate spatial preference
Kyoto7**	-0.097	0.476 ± 0.003	0.716	Near-neutral, highest temporal contribution
Aruba	+0.002	0.500 ± 0.006	0.434	Non-convergent, excluded***

* Optuna's search disabled the contextual gate (contextual_gat = False) for this dataset.

** For Kyoto7, the M3 (Contextual GAT) configuration is reported despite Optuna's initial MHA-fusion selection; see Section 4.1 and Table 5.

*** $|\lambda_{\text{raw}}| < 0.05$: the parameter did not move meaningfully from initialization; excluded from the entropy comparison above.

Normalized entropy: Shannon entropy of the first-order activity-transition distribution, normalized by log (number of classes).

Table 7 Edge deployment benchmark (TFLite INT8, Single CPU Thread)

Dataset	Keras (MB)	TFLite INT8 (MB)	Compression	Latency (ms)	Tensor Arena (MB)
Milan	7.62	0.76	10.0×	5.6	2.6
Cairo	2.36	0.38	6.2×	17.5	4.6
Aruba	7.07	1.13	6.3×	15.6	13.2
Kyoto11	9.99	0.99	10.1×	11.8	30.6
Kyoto8	4.75	0.67	7.1×	21.3	31.6
Kyoto7	2.50	0.51	4.9×	20.4	35.4

activities span seconds to minutes. The tensor arena – dynamic RAM allocated during inference for intermediate activations – ranges from 2.6 MB (Milan) to 35.4 MB (Kyoto7), driven primarily by sequence length ($T_{\max}$) and the number of sensors N . Simpler environments (Cairo: 4.6 MB, Milan: 2.6 MB) are deployable on entry-level IoT gateways (e.g., Raspberry Pi Zero 2W: 512 MB RAM), while complex multi-resident environments (Kyoto7–11: 30–36 MB) require mid-range gateway hardware (e.g., Raspberry Pi 4: 1–4 GB RAM) – both categories standard in residential AAL deployments [3, 17]. To complement these deployment metrics with hardware-agnostic architectural detail, Table 8 reports parameter count and forward-pass FLOPs (batch size 1) broken down by architectural module for each dataset's optimal configuration.

4.5 Sensitivity to graph edge weighting

To assess robustness to the heuristic edge weights introduced in Section 2.1.3, we replaced the adjacent-room weight of 0.5 with 0.75 (same-room weight of 1.0 and non-adjacent weight of 0.0 held fixed) for all six datasets, retraining once per dataset under each dataset's already-optimized configuration (Table 3). Table 9 compares the

Table 8 Parameter and FLOPs breakdown by architectural module

Dataset	Total params	Temporal branch (%)	Spatial GAT branch (%)	Fusion+Cls (%)	FLOPs (M)
Aruba	595,651	52.8	39.0	8.2	11.30
Cairo	192,719	88.3	0.9	10.8	2.04
Kyoto7	189,028	26.8	69.6	3.6	19.35
Kyoto8	395,016	83.2	5.4	11.4	16.64
Kyoto11	849,605	69.0	21.9	9.2	16.44
Milan	650,563	84.5	4.1	11.4	2.60

Single fold per dataset (architecture is fold-invariant); FLOPs are forward-pass estimates for batch size 1. FLOPs scale primarily with $N^2 \times \text{gat_heads}$ (dense masked attention, Section 2.1.3) rather than with parameter count, which is why Kyoto7 has the fewest parameters but the highest FLOPs of the six configurations.

Table 9 Sensitivity of weighted F1-score to the adjacent-room edge weight

Dataset	$w = 0.5$ (original)	$w = 0.75$	Δ
Aruba	0.95	0.95	0.00
Cairo	0.91	0.91	0.00
Kyoto7	0.84	0.86	+0.02
Kyoto8	0.94	0.93	-0.01
Kyoto11	0.92	0.91	-0.01
Milan	0.96	0.96	0.00

Single-run comparison (not a 3-fold re-estimate); both configurations are evaluated on an identical held-out test partition per dataset (verified at the class level).

resulting weighted F1-score to the equivalent single-run evaluation under the original weighting.

The maximum absolute change is +2 percentage points (Kyoto7, an improvement), and the mean absolute change across all six datasets is under one point – comfortably inside the fold-to-fold variability already reflected in the 95% confidence intervals of Table 5. We conclude that performance is not meaningfully sensitive to this heuristic within a physically plausible range.

5 Conclusion

This paper presented H-BiLSTM-GAT, a modular and edge-deployable spatio-temporal framework for ambient Human Activity Recognition in IoT-enabled smart homes. Through rigorous experimental validation on six CASAS benchmark datasets, supported by automated hyperparameter optimization and 95% confidence intervals, we addressed the dual challenge of modeling context-dependent spatio-temporal relations while satisfying the computational constraints of IoT gateway deployment.

First, the feature-level temporal injection mechanism (Eq. (7)) enables dynamic spatial modulation through a single learnable scalar gate whose cost, while not always negligible in relative terms (Table 8), remains within the sub-megabyte edge-deployment footprint of Table 7, and which outperforms fusion-stage attention mechanisms. Analysis of the learned gate values shows a relationship with dataset temporal entropy: all active datasets converge to negative λ_{raw} values, with magnitude reflecting spatial discriminability, and the gate is disabled by Optuna on structured environments (Cairo, Kyoto8) where spatial topology alone suffices, consistent with an exact statistical separation between these two groups of datasets (Section 4.3). Second, the ablation study across four incremental variants demonstrates that simpler fusion strategies are superior to over-parameterized alternatives in sparse ambient data environments. The M4 configuration – combining contextual guidance and MHA fusion – achieves the lowest performance in five of the six scenarios and is never the optimal configuration for any dataset (Table 5), confirming that adding architectural complexity beyond what the data distribution can support introduces noise rather than signal. Third, TFLite INT8 quantization reduces all model footprints to 0.38–1.13 MB with single-thread inference latencies of 5.6–21.3 ms, confirming practical deployability on IoT gateway hardware. Simpler environments require only entry-level hardware (Cairo: 4.6 MB tensor arena), while complex multi-resident

scenarios (Kyoto7–11: 30–36 MB) are comfortably handled by mid-range gateways – both standard in residential AAL deployments. Beyond these contributions, our framework achieves competitive performance across all six environments, notably a weighted F1-score of 90.7% on the multi-resident Cairo dataset, advancing the capability of ambient systems to disambiguate individual residents without resident-identification sensors. These results offer practitioners concrete design guidelines for efficient ambient HAR under IoT constraints.

Two aspects of this study's scope should be made explicit. H-BiLSTM-GAT targets the classification stage of ambient HAR rather than a complete real-time pipeline: results assume ground-truth activity boundaries, consistent with the sequential baselines we compare against; an online segmentation or sliding-window front-end was not an objective here and is a natural next step. In addition, hyperparameter search and final evaluation share the same 3-fold partition (Sections 3.2 and 4.1) rather than

a fully nested cross-validation; a supplementary check on three Optuna-unseen partitions found no indication of bias for five of six datasets, with one exception (Kyoto7, our highest-entropy environment) showing a small downward shift. A fully nested cross-validation remains valuable future work. Several directions remain open for future investigation. The static physical graph and its dense masked-attention formulation (Section 2.1.3) could both be made more scalable: sparse, event-triggered topology updates would accommodate sensor displacement without full graph reconstruction, and a sparse edge-indexed attention implementation would preserve efficiency at sensor counts substantially larger than those considered here. A second direction is extending the global scalar gate λ to sensor-type-specific gating functions $\lambda_i = f(h_i)$, allowing selective temporal modulation per sensor modality. Finally, formalizing the observed parsimony behavior through generalization bounds would advance ambient HAR from empirical practice toward predictive architectural theory.

References

- [1] Cook, D. J., Crandall, A. S., Thomas, B. L., Krishnan, N. C. "CASAS: A Smart Home in a Box", *Computer*, 46(7), pp. 62–69, 2013.
<https://doi.org/10.1109/MC.2012.328>
- [2] Abdel-Basset, M., Chang, V., Hawash, H., Chakraborty, R. K., Ryan, M. "Deep learning approaches for human-centered IoT applications in smart indoor environments: a contemporary survey", *Annals of Operations Research*, 339(1), pp. 3–51, 2024.
<https://doi.org/10.1007/s10479-021-04164-3>
- [3] Wang, K., Saragadam, A., Kaur, J., Dogra, A., Cao, S., Ghafurian, M., Butt, Z. A., Abhari, S., Chumachenko, D., Morita, P. P. "A contactless method for recognition of daily living activities for older adults based on ambient assisted living technology", *Internet of Things*, 30, 101502, 2025.
<https://doi.org/10.1016/j.iot.2025.101502>
- [4] Bouchabou, D., Nguyen, S. M., Lohr, C., LeDuc, B., Kanellos, I. "A Survey of Human Activity Recognition in Smart Homes Based on IoT Sensors Algorithms: Taxonomies, Challenges, and Opportunities with Deep Learning", *Sensors*, 21(18), 6037, 2021.
<https://doi.org/10.3390/s21186037>
- [5] Chen, K., Zhang, D., Yao, L., Guo, B., Yu, Z., Liu, Y. "Deep Learning for Sensor-based Human Activity Recognition: Overview, Challenges, and Opportunities", *ACM Computing Surveys*, 54(4), pp. 1–40, 2022.
<https://doi.org/10.1145/3447744>
- [6] Hochreiter, S., Schmidhuber, J. "Long Short-Term Memory", *Neural Computation*, 9(8), pp. 1735–1780, 1997.
<https://doi.org/10.1162/neco.1997.9.8.1735>
- [7] Liciotti, D., Bernardini, M., Romeo, L., Frontoni, E. "A sequential deep learning application for recognising human activities in smart homes", *Neurocomputing*, 396, pp. 501–513, 2020.
<https://doi.org/10.1016/j.neucom.2018.10.104>
- [8] Krishnaleela, P., Prakash, R. M. "CNN-SLSTM framework for human activity recognition using wearable sensor data", *Neural Computing and Applications*, 37(23), pp. 19501–19522, 2025.
<https://doi.org/10.1007/s00521-025-11410-3>
- [9] Khan, S. U., Sultana, M., Danish, S., Gupta, D., Alghamdi, N. S., Woo, S., Lee, D.-G., Ahn, S. "Multimodal feature fusion for human activity recognition using human centric temporal transformer", *Engineering Applications of Artificial Intelligence*, 160, 111844, 2025.
<https://doi.org/10.1016/j.engappai.2025.111844>
- [10] Javadi, S., Riboni, D., Borzi, L., Zolfaghari, S. "Graph-Based Methods for Multimodal Indoor Activity Recognition: A Comprehensive Survey", *IEEE Transactions on Computational Social Systems*, 12(5), pp. 3728–3746, 2025.
<https://doi.org/10.1109/TCSS.2024.3523240>
- [11] Khemani, B., Patil, S., Kotecha, K., Tanwar, S. "A review of graph neural networks: concepts, architectures, techniques, challenges, datasets, applications, and future directions", *Journal of Big Data*, 11(1), 18, 2024.
<https://doi.org/10.1186/s40537-023-00876-4>
- [12] Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., Bengio, Y. "Graph Attention Networks", [preprint] *arXiv*, arXiv:1710.10903, 04 February 2018.
<https://doi.org/10.48550/arXiv.1710.10903>
- [13] Wang, Y., Wang, X., Yang, H., Geng, Y., Yu, H., Zheng, G., Liao, L. "MhaGNN: A Novel Framework for Wearable Sensor-Based Human Activity Recognition Combining Multi-Head Attention and Graph Neural Networks", *IEEE Transactions on Instrumentation and Measurement*, 72, 2514314, 2023.
<https://doi.org/10.1109/TIM.2023.3276004>

- [14] Ye, J., Jiang, H., Zhong, J. "A Graph-Attention-Based Method for Single-Resident Daily Activity Recognition in Smart Homes", *Sensors*, 23(3), 1626, 2023.
<https://doi.org/10.3390/s23031626>
- [15] Srivatsa, P., Plötz, T. "Using Graphs to Perform Effective Sensor-Based Human Activity Recognition in Smart Homes", *Sensors*, 24(12), 3944, 2024.
<https://doi.org/10.3390/s24123944>
- [16] Qian, L., Zuo, Q., Liu, H., Zhu, H. "Multivariate time series classification based on spatial-temporal attention dynamic graph neural network", *Applied Intelligence*, 55(2), 115, 2025.
<https://doi.org/10.1007/s10489-024-06014-8>
- [17] Diraco, G., Rescio, G., Siciliano, P., Leone, A. "Review on Human Action Recognition in Smart Living: Sensing Technology, Multimodality, Real-Time Processing, Interoperability, and Resource-Constrained Processing", *Sensors*, 23(11), 5281, 2023.
<https://doi.org/10.3390/s23115281>
- [18] Batool, M., Alotaibi, M., Alotaibi, S. R., Alhammadi, D. A., Jamal, M. A., Jalal, A., Lee, B. "Multimodal Human Action Recognition Framework Using an Improved CNNGRU Classifier", *IEEE Access*, 12, pp. 158388–158406, 2024.
<https://doi.org/10.1109/ACCESS.2024.3481631>
- [19] Miah, A. S. M., Hwang, Y. S., Shin, J. "Sensor-Based Human Activity Recognition Based on Multi-Stream Time-Varying Features With ECA-Net Dimensionality Reduction", *IEEE Access*, 12, pp. 151649–151668, 2024.
<https://doi.org/10.1109/ACCESS.2024.3473828>
- [20] Huang, Z., Lian, S., Zhang, C., Li, X. "Application of graph convolutional spatial temporal fusion model based on gated recurrent unit and attention mechanism in inertial signal human activity recognition", *Biomedical Signal Processing and Control*, 112, 108745, 2026.
<https://doi.org/10.1016/j.bspc.2025.108745>
- [21] Akiba, T., Sano, S., Yanase, T., Ohta, T., Koyama, M. "Optuna: A Next-generation Hyperparameter Optimization Framework", In: *KDD '19: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, Anchorage, AK, USA, 2019, pp. 2623–2631. ISBN 978-1-4503-6201-6
<https://doi.org/10.1145/3292500.3330701>
- [22] Cui, Y., Jia, M., Lin, T.-Y., Song, Y., Belongie, S. "Class-Balanced Loss Based on Effective Number of Samples", In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA, 2019, pp. 9260–9269. ISBN 978-1-7281-3294-5
<https://doi.org/10.1109/CVPR.2019.00949>
- [23] Cai, Y., Guo, B., Salim, F., Hong, Z. "Towards Generalizable Human Activity Recognition: A Survey", [preprint] *arXiv*, arXiv:2508.12213, 17 August 2025.
<https://doi.org/10.48550/arXiv.2508.12213>
- [24] Python Software Foundation "Python, (3.12.3)", [computer program] Available at: <https://www.python.org> [Accessed: 03 March 2026]
- [25] Keras Team "Keras, (3.4.1)", [computer program] Available at: <https://keras.io> [Accessed: 03 March 2026]
- [26] TensorFlow Developers "TensorFlow, (2.17.0)", [computer program] Available at: <https://www.tensorflow.org> [Accessed: 03 March 2026]
- [27] Scikit-learn developers "Scikit-learn, (1.5.0)", [computer program] Available at: <https://scikit-learn.org> [Accessed: 03 March 2026]
- [28] NumPy Developers "NumPy, (1.26.4)", [computer program] Available at: <https://numpy.org> [Accessed: 03 March 2026]
- [29] The Pandas Development Team "Pandas, (2.2.2)", [computer program] Available at: <https://pandas.pydata.org> [Accessed: 03 March 2026]
- [30] Bouchabou, D., Nguyen, S. M., Lohr, C., Kanellos, I., LeDuc, B. "Reconnaissance d'activités de la vie quotidienne au moyen de capteurs domotiques et d'apprentissage profond : lorsque syntaxe, sémantique et contexte se rencontrent" (Recognition of daily life activities using home automation sensors and deep learning: when syntax, semantics and context meet), *Revue Ouverte d'Intelligence Artificielle*, 4(1), pp. 129–156, 2023. (in French)
<https://doi.org/10.5802/roia.53>