Evidence-Grounded Guardrail Extraction for Activity Recognition in Smart Homes using Small Language Models

Victor Romero II[1,2,a)] Tomokazu Matsui[1,b)] Yuki Matsuda[3,1,c)] Hirohiko Suwa[1,d)]
Keiichi Yasumoto[1,e)]

Abstract: Activity recognition remains a significant challenge in pervasive computing, where models must infer user actions from sparse signals but often fail to enforce the contextual constraints required for consistent predictions. This study proposes a failure-triggered method for extracting decision guardrails using Small Language Models (SLMs) to improve the reliability of activity recognition systems. The framework constructs a Knowledge Graph of guardrails from model feedback, which serves as a grounded evidence base for subsequent inference. During inference, these grounded constraints are incorporated to guide predictions toward more contextually consistent activity recognition. We implement a pipeline that generates and reuses these constraints, and evaluate its impact on classification performance and reasoning behaviour. Experiments show that the proposed approach improves top-1 accuracy from 46.4% to 69.7%, with reduced class-level confusion and more consistent predictions. This work offers a training-free mechanism for transitioning from purely pattern-based activity recognition toward more constraint-aware systems.

Keywords: small language models, human activity recognition, knowledge graphs

1. Introduction

Human activity recognition (HAR) in smart-homes underpins a wide range of applications, including energy management, safety, well-being and healthcare [1]. In the canonical formulation, HAR models map windows of sensor events to activity labels [2]. However, sensor inputs are typically sparse and noisy, and only indirectly reflect the underlying activity [3]. As a result, the same activity can generate different sensor patterns while similar patterns may correspond to different activities. Addressing this ambiguity benefits from incorporating contextual constraints, such as room layout, object affordances, and routine patterns. Unfortunately, homes rarely share identical layouts or occupant routines. This heterogeneity makes static, pre-defined knowledge about sensor-activity relationships brittle in practice, motivating the development of context-aware and adaptable HAR techniques.

A recent line of work explores the use of language models as semantic layer for interpreting sensor data [4], [5]. By translating low-level sensor events into textual representations, these models can be leveraged to reason over activity patterns, enabling recognition without extensive additional retraining. However, this approach introduces a practical tension. While large, cloud-based models exhibit strong reasoning capabilities, their deployment is often incompatible with privacy, latency, and resource constrains of smart home environments [6]. Small Language Models (SLM) provide a more viable alternative in such settings due to their efficiency and deployability [7]. Yet this efficiency comes at the cost of weaker contextual grounding making them more prone to producing logically inconsistent predictions [8], [9].

Existing mitigation strategies attempt to address this limitation either by encoding constraints directly in prompts or by injecting external knowledge at inference time [10], [11]. Prompt-based approaches condition the model through handcrafted rules, but these formulations become brittle and difficult to scale as environmental complexity increases. Knowledge-based systems, including retrieval augmented generation and knowledge graph injection improve contextual consistency but typically assume the availability of pre-constructed knowledge sources.

This leaves an unresolved gap: *How to equip SLM-based HAR systems with contextually grounded constraints without relying on static, pre-defined knowledge*. In this work, we treat knowledge not as a fixed input but as an artifact that can be constructed during deployment. We propose a failure-triggered knowledge infusion loop in which the model reflects on its own prediction errors and, together with corrective feedback, extracts structured guardrail triples that capture evidence-grounded constraints. These guardrails are accumulated in a lightweight knowledge graph and selectively retrieved at inference time based on their relevance to the current sensor evidence. By explicitly injecting these constraints into the reasoning process, the model is guided toward predictions that are not only plausible but also contextually feasible, enabling a more transparent and adaptable approach to activity recognition in smart-home environments.

1 Nara Institute of Science and Technology
2 University of the Philippines Tacloban College
3 Okayama University
a) vmromero@up.edu.ph
b) m.tomokazu@is.naist.jp
c) yukimat@okayama-u.ac.jp
d) h-suwa@is.naist.jp
e) yasumoto@is.naist.jp

© 2018 Information Processing Society of Japan

Contributions. Our contributions are as follows:

(1) We introduce a failure-triggered KG-infusion loop for smart-home HAR, where model errors are converted into reusable guardrail triples for later inference;

(2) We show strong accuracy and class-separation gains over zero-shot. Top-1 accuracy improves from 46.4% to 69.7% and confusion quality also improves with macro diagonal (row-normalized class recall) rising from 36.9% to 64.8%;

(3) We show the benefit comes with measurable but manageable runtime cost. Median inference latency increases from 4.77s (zero-shot) to 8.84s (KG-infused), while KG augmentation, the failure-triggered construction and integration of guardrail triples into the KG, adds a median 48.61s.

2. Related Work

This section reviews literature on sensor-based human activity recognition (HAR), prompt-based classification with small language models (SLMs), and knowledge-based augmentation. It identifies key limitations in existing approaches and motivates a self-constructed knowledge graph for guardrail injection.

2.1 Sensor-Based HAR: Data and Structural Constraints

Sensor-based HAR in smart homes is typically formulated as mapping windows sequences of ambient sensor events to activity labels. Inputs are sparse, noisy, and only indirectly related to the underlying activity [4], [12], [13]. Consequently, the same activity can produce different sensor patterns, and similar patterns can correspond to different activities [14].

A central property of this setting is that correct predictions depend on implicit contextual constraints [15]. Sensor activations must be interpreted relative to spatial layout (e.g., room location), object affordances , and routine patterns. Many recognition errors, therefore, arise when a predicted activity is incompatible with observed context (e.g., required sensors are absent or locations are inconsistent) [16]. The challenge is further compounded by heterogeneity, since no two homes share identical layouts, sensor placements, or occupant routines, which makes static, one-size-fits-all knowledge representation brittle in practice [17].

2.2 Prompt-based Classification with Language Models

Recent work shows that pre-trained language models can perform classification without updating model parameters. Methods such as PET [18] and LM-BFF [19] operationalize this idea by recasting classification as a prompt-based task, enabling strong performance with only a small number of labelled examples. In HAR, this requires converting sensor readings into a textual representation that the model can condition on. Prior studies indicate that language-based representation of sensor data can capture useful semantics [5], and recent work suggests that language models can perform zero-shot HAR successfully [4].

However, deploying such approaches in practical settings introduces additional constraints. In smart home environments, models are typically required to operate under limited computational resources, privacy restrictions, and on-device deployment, making small language models (SLMs) the more realistic choice [7], [20]. Yet this efficiency comes at the cost of weaker internal priors and reduced contextual grounding. Empirical studies show that SLMs are more prone to hallucination and exhibit degraded context-sensitive reasoning, often producing outputs that are superficially correct but supported by inconsistent reasoning [21], [22], [23]. In this setting, model behaviour can be influenced either by modifying the model parameters or by conditioning the model at inference time. Fine-tuning allows knowledge to be internalized within the model but requires additional data and computation and may not be practical in privacy-sensitive or continuously evolving environments such as smart homes [24]. As a result, inference-time methods such as prompting become the more practical mechanism for incorporating task-specific information [10], [11]. However, while prompts can bias predictions, they do not provide a reliable way to encode or persist structured contextual constraints. Moreover, directly embedding a large number of constraints into the prompt does not scale and may potentially introduce noise during inference [25], [26], [27].

2.3 Knowledge-based HAR Augmentation

To address the need for explicit context, prior work in HAR has explored knowledge-driven and neuro-symbolic approaches. Ontology-based systems encode domain knowledge explicitly and use reasoning mechanisms to improve recognition and interpretability [28]. Extensions such as Arianna+ further develop modular ontology networks for activity recognition [29]. More recent approaches incorporate knowledge into machine learning pipelines. Methods such as P-NIMBUS [30] integrate contextual knowledge and uncertainty handling into deep learning models, while ContextGPT [31] uses large language models to retrieve common-sense knowledge for HAR tasks.

In parallel, the broader language model literature shows that external knowledge can improve model performance. Retrieval-Augmented Generation (RAG) and knowledge graph injection methods such as K-BERT demonstrate that language models can benefit from retrieving structural or textual knowledge at inference time [32]. Constraint-based prompting methods, including grammar prompting [33], further show that structured constraints can be injected into prompts to guide model outputs [34]. A related line of work concerns knowledge graph construction (KFC) which typically builds triples from large text corpora using supervised or weakly supervised pipelines [35].

2.4 Position of this Study

Despite progress across these areas, current approaches typically rely on implicit knowledge within the model or assume the availability of externally constructed knowledge. Neither addresses how task-specific constraints can be obtained when such knowledge is not available at deployment. We position this work as addressing this gap by treating knowledge as a construct that originates during deployment rather than being assumed a priori. Specifically, we propose a framework in which a small language model uses its own prediction errors, together with corrective feedback, to extract and accumulate structured decision constraints in the form of knowledge graph triples. These constraints are then reintroduced during inference, enabling the model to move beyond purely pattern-based predictions.

© 2018 Information Processing Society of Japan

3. Methodology

This section presents a constraint-aware HAR framework that constructs and reuses symbolic guardrails derived from prediction errors. We consider a privacy-preserving smart-home environment instrumented with ambient sensors, where visual sensing (e.g., cameras) is not used. The sensing infrastructure is composed primarily of event-driven and state-based devices such as contact sensors and room-level location signals that provide indirect observations of user activities through state changes and interactions with objects and spaces.

3.1 Preliminaries and Problem Definition

Let the smart home be partitioned into semantic zones, denoted as $\mathcal{Z} = \{z_1, \cdots, z_{|\mathcal{Z}|}\}$. Similarly, let the deployed sensors be represented as a set $\mathcal{S} = \{s_1, \cdots, s_{|\mathcal{S}|}\}$.

Each sensor has metadata

$$\phi(s) = (\tau(s), m(s), \rho(s), \nu(s)),$$

where $\tau(s)$ denotes type, $m(s) \in \{\text{static}, \text{mobile}\}$ denotes mobility class, $\rho(s)$ denotes the sampling profile, and $\nu(s)$ is an optional associated object or appliance. To represent heterogeneous sensing behaviour, we define

$$\rho(s) = \begin{cases} (\text{periodic}, r_s), & \text{if } s \text{ samples at fixed rate } r_s \\ (\text{event-driven}, \kappa_s), & \text{if } s \text{ emits on trigger condition } \kappa_s \end{cases}$$

Many home sensors are trigger-based (ON/OFF or threshold events), while others are sampled regularly. Here, κ_s is represented as a symbolic trigger such as state transition or threshold crossing. We denote the location of a sensor s as $\lambda(s) \in \mathcal{Z}$. In the case of static sensors, this location is assumed to be fixed.

Classification is performed on fixed-length windows of duration Δ. For sample i with decision time t_i, the active window is

$$I_i = [t_i - \Delta,\ t_i].$$

All sensor readings that occur within I_i are used to construct the evidence E_i for that sample. Let $C = \{c_1, \ldots, c_{|C|}\}$ be the activity label set. The HAR decision problem is to map sample evidence to a predicted class:

$$\hat{c}_i = h(E_i), \quad h : \mathcal{E} \to C,$$

where $\mathcal{E}$ denotes the space of evidence representations. Throughout this work, we denote by $f_{\text{SLM}}(\cdot)$ a Small Language Model (SLM) generation function that produces structured outputs conditioned on an input prompt constructed from its arguments. Task-specific behaviour is indicated through a superscript. For example, $f^{\text{pred}}_{\text{SLM}}(\cdot)$ denotes a prediction function.

3.2 Guardrail KG Construction for HAR

Here we present the principal methodological contribution: a failure-triggered online construction process depicted in **Fig. 1** that turns prediction mistakes into reusable symbolic guardrails.

3.2.1 Inference Formulation

Classification is conditioned on both the current evidence and the retrieved guardrails:

$$\hat{\mathbf{c}}_i^{(K)} = f^{\text{pred}}_{\text{SLM}}(E_i, G_i),$$

where $\hat{\mathbf{c}}_i^{(K)}$ is an ordered top-K prediction list over classes in C and G_i is the retrieved guardrail set for sample i. If no relevant guardrails are found, inference proceeds with $G_i = \varnothing$.

For each sample, the prompt is assembled from five components: (1) sensor infrastructure semantics, (2) time-window context, (3) normalized evidence summary, (4) retrieved guardrails (if available), and (5) candidate label set. The model is required to return strict JSON containing ranked activity predictions and concise evidence-grounded justifications.

3.2.2 Failure Trigger and Reflection Prompting

Let c_i denote ground-truth class and let $\hat{c}_i$ denote top-1 prediction (the first element of $\hat{\mathbf{c}}_i^{(K)}$). Graph update is triggered only when $\hat{c}_i \neq c_i$. Under this condition, reflection is invoked with evidence E_i, the ranked prediction list $\hat{\mathbf{c}}_i^{(K)}$, and the true label c_i. In this formulation, c_i acts as corrective feedback that signals a prediction failure and enables guardrail extraction. The framework does not require continuous supervision and only assumes that such feedback is available at some points during operation. In practice, this feedback may come from delayed human annotation, user correction, or other supervisory processes, consistent with a human-in-the-loop setting.

We define reflection generation as

$$R_i = f^{\text{reflect}}_{\text{SLM}}(E_i, \hat{\mathbf{c}}_i^{(K)}, c_i),$$

where R_i is a structured reflection artifact rather than unconstrained free text. The prompt enforces three sections: (1) `SUPPORTING_EVIDENCE`, (2) `CONTRASTIVE_CUES`, and (3) `GUARDRAILS`, each one serving a distinct function in the update mechanism. `SUPPORTING_EVIDENCE` surfaces the positive evidence pattern for the true class (required vs. supporting signals and location context). `CONTRASTIVE_CUES` surfaces discriminative negatives by making absences and mismatches explicit for confusing alternatives. `GUARDRAILS` converts the earlier sections into reusable decision constraints. An example reflection is shown in **Fig. 2**. This structure is intentionally requested because later symbolic extraction depends on explicit mention of presences, absences, and class-conditional distinctions.

3.2.3 Multiphase Guardrail Construction

Rather than extracting triples directly from free-form reflection, we apply a staged construction chain as illustrated in **Fig. 3**.

Let $\mathcal{A}$, $\mathcal{S}$, and $\mathcal{L}$ denote activity, sensor, and location vocabularies, respectively.

In **Stage 1** (entity extraction), the reflection artifact is projected onto ontology-relevant symbols only:

$$U_i = f^{\text{ent}}_{\text{SLM}}(R_i), \quad U_i \subseteq \mathcal{A} \cup \mathcal{S} \cup \mathcal{L}.$$

This step removes narrative tokens and out-of-scope concepts so downstream processing cannot rely on entities that are outside the activity-sensor-location interface.

© 2018 Information Processing Society of Japan

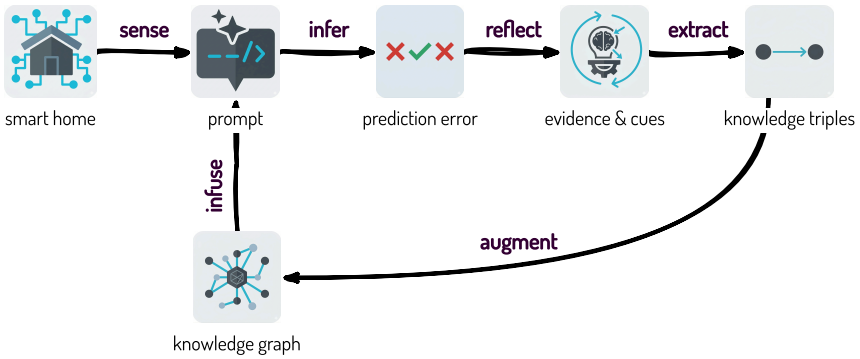

Fig. 1 Knowledge-infused inference loop for smart-home sensor-based HAR, where inferred and reflected outcomes are extracted to augment the knowledge graph, and the updated graph information is infused into subsequent inferences.

Reflection Output for Activity: `preparing_cold_meal`

- **Supporting Evidence:**
  Pantry interaction is observed in the kitchen, which is consistent with preparing a cold meal. This activity typically involves accessing stored food items without using heat-based appliances...
- **Contrastive Cues:**
  Cooking is less likely because there is no evidence of stove usage. Watching TV is also unlikely since no television-related signals are present and the activity is not occurring in the living room...
- **Guardrails:**
  Cooking generally requires use of the cooking stove. Watching TV requires the television to be active and typically occurs in the living room. Preparing a cold meal involves pantry interaction and is usually performed in the kitchen...

Fig. 2 Example reflection artifact R_i produced by $f_{\text{SLM}}^{\text{reflect}}$, showing supporting evidence, contrastive cues, and inferred guardrails.

Stage 1 Output (U_i):

- **activities:** `preparing_cold_meal`, `cooking`, `watching_tv`, ...
- **sensors:** `pantry`, `cooking_stove`, `television`, ...
- **locations:** `kitchen`, `living_room`, ...

Stage 2 Output (D_i):

- `preparing_cold_meal`:
  - required_sensor: `pantry`
  - required_location: `kitchen`
- ...

Stage 3 Output:(T_i^{cand}):

- (`preparing_cold_meal`, `REQUIRES_SENSOR`, `pantry`)
- (`preparing_cold_meal`, `REQUIRES_LOCATION`, `kitchen`)
- ...

Fig. 3 Stepwise transformation from reflection artifact to ontology-constrained representations, including entity extraction (U_i), structured denoising (D_i), and candidate triple construction (T_i^{cand}).

In **Stage 2** (reflection denoising), the same reflection is rewritten using the extracted symbol set as guidance:

$$D_i = f_{\text{SLM}}^{\text{denoise}}(R_i, U_i).$$

Here, D_i is an activity-centered structured representation that stores slots such as required sensors, missing required sensors, required location, forbidden locations, and supporting sensors. The purpose of this stage is to convert explanatory prose into a rule-ready intermediate form.

In **Stage 3** (triple construction), denoised fields are compiled into candidate symbolic rules:

$$T_i^{\text{cand}} = f_{\text{SLM}}^{\text{triple}}(D_i, U_i).$$

Compilation is schema-constrained. Slot semantics determine relation type, while entity arguments are restricted to U_i. This ensures that candidate triples are syntactically valid and semantically grounded in the reflection-derived intermediate output.

This decomposition improves robustness by reducing narrative leakage and increasing schema compliance before candidate triples are considered for acceptance. It also separates concerns across model-mediated stages: $f_{\text{SLM}}^{\text{reflect}}$ performs error analysis, while $f_{\text{SLM}}^{\text{ent}}$, $f_{\text{SLM}}^{\text{denoise}}$, and $f_{\text{SLM}}^{\text{triple}}$ perform symbolic compilation.

3.2.4 Guardrail Ontology and Final Consistency Gate

Guardrail ontologies can be designed in many ways, ranging from narrow task-specific relation sets to broader, dynamically expanded schemas. In this study, we intentionally use a compact relation set that targets the contrastive decision cues required by our HAR setting. Candidate triples are therefore constrained to:

(1) `REQUIRES_SENSOR(activity, sensor)`
(2) `ABSENCE_CONTRADICTS(activity, sensor)`
(3) `REQUIRES_LOCATION(activity, location)`
(4) `FORBIDS_LOCATION(activity, location)`
(5) `SUPPORTS_SENSOR(sensor, activity)`

These five relation types are chosen to encode the minimum contrastive semantics needed by the downstream classifier: necessity, contradiction-by-absence, location requirement, location exclusion, and positive sensor support. The design objective is not ontological completeness, but operational usefulness under online updates. In preliminary development, allowing unrestricted relation extraction from reflection text produced large, semantically diffuse triple sets and did not improve predictive performance. Constraining relation type therefore functions as an explicit bias toward high-utility, decision-facing rules rather than descriptive but non-operational knowledge. At the same time, this ontology is not assumed to be universally complete. Future variants may expand or dynamically induce additional relation families under stronger validation constraints.

After extraction, final consistency gating is implemented as a hybrid two-stage operator that combines model-based adjudication with deterministic validation:

$$\widetilde{T}_i = f_{\text{SLM}}^{\text{gate}}(R_i, T_i^{\text{cand}}),$$

$$T_i^{\text{final}} = \Pi_{\text{valid}}(\widetilde{T}_i),$$

© 2018 Information Processing Society of Japan

where $f_{\text{SLM}}^{\text{gate}}$ selects a reflection-consistent subset from candidate triples, and $\Pi_{\text{valid}}(\cdot)$ enforces deterministic checks: (i) non-empty subject/relation/object, (ii) membership in the allowed relation set, and (iii) admissible subset selection from T_i^{cand}. Speculative or malformed candidates are rejected. Only triples in T_i^{final} are committed as reusable guardrails.

3.3 Guardrail Infusion During Inference

Guardrail infusion is the reuse phase of the lifecycle. At a later sample j, stored triples are ranked against current evidence by lexical overlap. For stored triple g,

$$\text{score}(g, E_j) = \left|\text{tok}(g) \cap \text{tok}(E_j)\right|,$$

where tok$(\cdot)$ denotes the set of normalized tokens obtained by lowercasing the input text, extracting alphanumeric substrings, and removing duplicates. In effect, both evidence and triples are treated as unordered bags of tokens, and relevance is determined by the number of shared tokens. Triples are sorted by score, deduplicated, and truncated to top-R to produce G_j, which is injected into a dedicated guardrail section in the inference prompt. This retrieval policy is intentionally simple, transparent, and auditable such that every infused guardrail can be traced to explicit lexical overlap with current evidence.

4. Experiments

This section evaluates the proposed constraint-aware HAR framework against a zero-shot baseline.

4.1 Dataset

We use the MARBLE [13] dataset, a smart-home activity recognition corpus collected in a fixed-layout environment. It is organized into scenarios, each representing sequences of activities performed by one or more subjects over a period of the day, with multiple recordings per scenario.

4.1.1 The Smart Home Model

Fig. 4 illustrates the smart home layout and the sensor deployment locations in the MARBLE dataset. The home setup is compact but semantically rich, including contact sensors (pantry, drawers, cabinet, fridge), smart plugs (stove, TV), and pressure mats (chairs, couch), along with smartphone events, room-level locations, and activity labels. This configuration supports both spatial and behavioural reasoning. The fixed layout provides stable context, while diverse sensors offer complementary evidence about actions and location, which is essential for context-aware evaluation. Additionally, MARBLE also includes inertial sensor readings from smart watches worn by the smart home subjects, they are however not considered in this study.

4.1.2 Preprocessing and Datapoint

We restrict our experiments to MARBLE's individual subset, which excludes multi occupancy cases, to avoid ambiguity that arise from concurrent occupants in the same environment. We further filter out `TRANSITION` events, since these intervals mark boundaries between activities rather than target activities themselves and would otherwise introduce additional class noise.

The remaining activity interval is converted into a sequence of fixed-duration temporal windows with a length of 16 seconds and 80% overlap between consecutive windows. This windows construction, consistent with the preprocessing pipelines used in prior studies using the MARBLE dataset [4], standardizes variable-length activity intervals into fixed-sized samples while preserving enough temporal context for downstream reasoning. For environmental sensors and smart phone events, the representation includes not only events that occur inside the window but also pre-window context when the most recent state before the window is still relevant at window onset. This is essential because many smart-home signals are stateful, i.e,. a door may already be open or a switch may already be on when the window begins. By carrying forward that context, the representation preserves continuity across adjacent windows and avoids treating each slice as an independent environment.

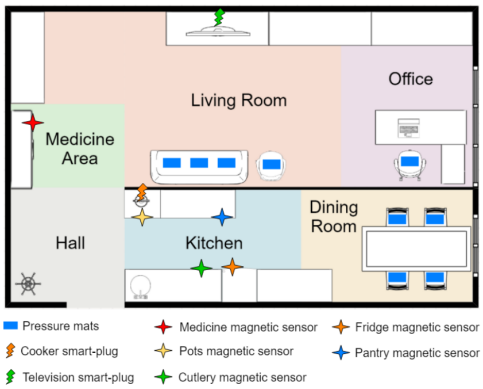

Fig. 4 Smart home layout and sensor placement for MARBLE dataset [13].

Table 1 Class code mapping with dataset counts and percentage.

Code	Description	Count	Percent
WTV	Watching Tv	579	16.07
COK	Cooking	503	13.96
UPC	Using Pc	415	11.51
APH	Answering Phone	370	10.27
MPC	Making Phone Call	360	9.99
EAT	Eating	351	9.74
PCM	Preparing Cold Meal	282	7.82
WSD	Washing Dishes	198	5.49
SUT	Setting Up Table	174	4.83
CLT	Clearing Table	166	4.61
TMD	Taking Medicines	158	4.38
LVH	Leaving Home	25	0.69
ENH	Entering Home	23	0.64

4.1.3 Constructed Dataset Characteristics

The preprocessing pipeline yields 3,604 data points across 36 instances and 12 subjects. The class distribution is long-tailed (see **Table 1**), with activities such as *watching tv* and *cooking* dominating the dataset, while *leaving home* and entering *home together* account for roughly 1.5%. This skew reflects the underlying activity structure of MARBLE, since longer activities naturally contribute more windows. The table also provides 3-letter codes for each class to streamline the presentation of results.

As shown in **Fig. 5**, cross-subject heterogeneity is substantial with different subjects contributing varying numbers of data points. Beyond volume, the left panel shows that activity frequencies differ across individuals, suggesting that subjects vary in how they perform activities and how long they spend on them,

© 2018 Information Processing Society of Japan

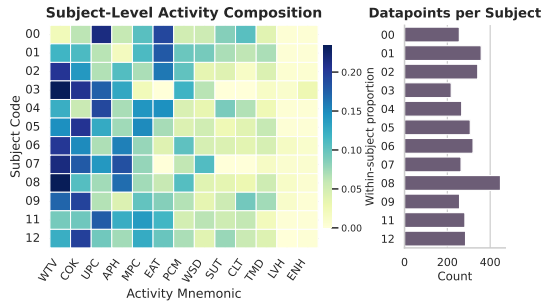

Fig. 5 Subject-level heterogeneity in the dataset: row-normalized activity composition by subject (left) and subject-wise datapoint counts (right), with activities ordered by global frequency.

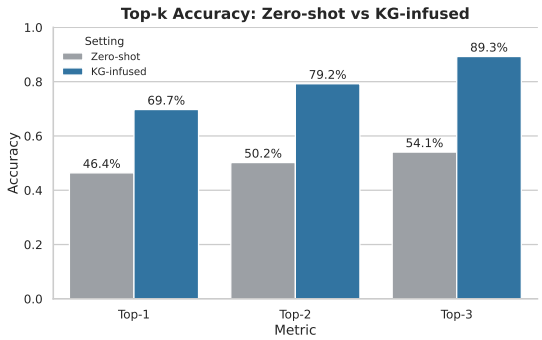

Fig. 6 Top-k accuracy comparison between zero-shot and KG-infused settings. KG infusion consistently improves performance across Top-1, Top-2, and Top-3 metrics.

even when following the same scenarios.

4.2 Agent Implementation

We instantiate the classifier as a CrewAI[*1]-based agent pipeline, where a `smart home agent` is defined with a clear functional role (sensor interpretation), an operational objective (supporting activity recognition), and a domain prior that sensor state changes can be translated into semantic activity hypotheses. We use `qwen3:4b-instruct` as the underlying small language model, selected to balance reasoning capability with computational efficiently, which is important in realistic resource-constrained settings.

The functional components introduced in Section 3 are operationalized as distinct `Tasks` that each correspond to a specific step in the pipeline and is implemented through prompt-based instructions. The tasks are orchestrated sequentially within the agent, enabling a modular execution of the overall inference and KG-augmentation process. Rather than treating the model as a black box, the pipeline exposes intermediate artifacts that make it possible to trace how the system arrives at a decision and, when necessary, how it converts failures into reusable constraints.

4.3 Research Questions and Experimental Setups

We consider three research questions:

- **RQ1**: Does KG infusion improve activity recognition performance over a zero-shot baseline?
- **RQ2**: How does KG infusion affect model behaviour across classes and retrieval patterns?
- **RQ3**: What are the inference-time costs of KG infusion, and are they justified by performance gains?

To answer these questions, we run paired experiments comparing zero-shot and KG-infused inference on the same evaluation stream and align predictions by data point index.

We evaluate accuracy and ranking behaviour using top-k results and confusion analyses, then inspect retrieval dynamics through relation/triple distributions and cumulative growth of unique retrieved triples vs. KG expansion.

Finally, we quantify system overhead by estimating per-step latency from timestamps, separating no-reflection cases from augmentation cases, and relating added latency to the number of triples introduced. This design lets us jointly assess effectiveness, behavioural mechanisms, and computational trade-offs.

[*1] https://crewai.com/

5. Results and Discussions

This section presents the empirical results of the proposed framework and analyzes its impact on recognition performance and model behaviour. We first compare overall accuracy against a zero-shot baseline, then examine retrieval dynamics and efficiency trade-offs, and finally discuss limitations of the study.

5.1 Overall Performance

As shown in **Fig. 6**, KG infusion yields a clear improvement over the zero-shot baseline across all ranking levels. Top-1 accuracy increases from 46.4% to 69.7%, while top-2 accuracy rises from 50.2% to 79.2% and top-3 accuracy from 54.1% to 89.3%. This is not just a modest gain in first-choice prediction: the larger jumps at top-2 and top-3 show that the KG-infused model consistently places the correct activity much higher in its ranked outputs. In relative terms, the improvement is especially pronounced at the top of the list, indicating that the retrieval-augmented pipeline is substantially better at resolving the most likely label rather than merely expanding coverage.

The confusion matrices in **Fig. 7**, reinforce this picture at the class level. The zero-shot model exhibits broader off-diagonal confusion, with several activities being systematically mixed with nearby classes, whereas the KG-infused model concentrates much more mass on the diagonal and reduces several of the strongest misclassification patterns. This suggests that KG infusion improves not only aggregate accuracy but also the structure of the error distribution: predictions become more class-consistent and more semantically aligned with the true activity. In that sense, the gains are not random or isolated; they reflect better ranking quality and more stable class separation. The retrieval analysis in the next subsection helps explain why this happens.

5.2 Retrieval and Behavioural Analysis

Beyond prediction accuracy, the KG-infused system reveals a structured retrieval profile over the predefined relation types (see **Fig. 8**(left)). Retrieval events are dominated by `REQUIRES_LOCATION` and `FORBIDS_LOCATION`, followed by `REQUIRES_SENSOR`, `ABSENCE_CONTRADICTS`, and `SUPPORTS_SENSOR`. This distribution indicates that the system relies primarily on location constraints to prune the activity space, with sensor-based rules providing a secondary but still

© 2018 Information Processing Society of Japan

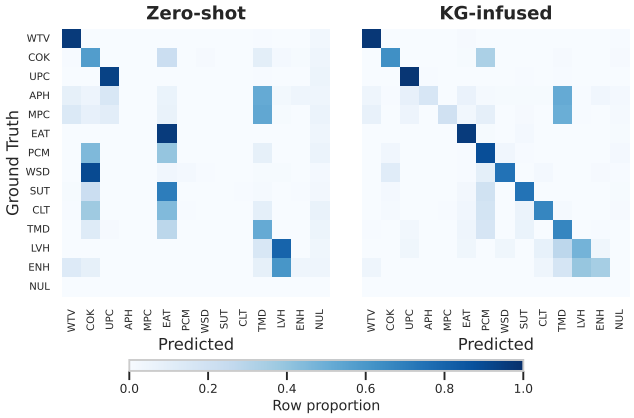

Fig. 7 Row-normalized top-1 confusion matrices show that KG infusion redistributes error mass from dominant off-diagonal confusions toward class-specific predictions.

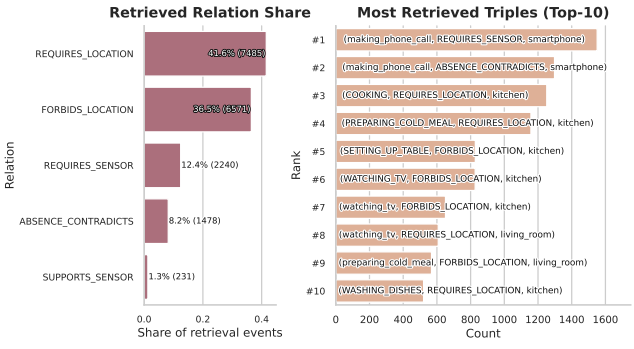

Fig. 8 Retrieved knowledge profile of the KG-infused setting: the left panel shows the distribution of retrieved relation types as a share of all retrieval events, while the right panel ranks the top-10 most frequently retrieved triples.

important layer of evidence.

The top retrieved triples, as shown in Fig. 8(right), further clarify this behaviour. The most frequently retrieved items correspond to compact, activity-specific guardrails, with a notable concentration of location-oriented constraints. Rather than reflecting arbitrary context, these results show that the system repeatedly reuses a small set of structured rules that encode activity–location and activity–sensor relationships. This pattern is consistent with the role of guardrails in disambiguating closely related activities in a smart-home setting.

The growth curve in **Fig. 9** adds a temporal perspective. Unique retrieved triples accumulate quickly at the beginning of inference, then increase more gradually as new samples are processed. By the end of the evaluation stream, the system has already consolidated most of its distinct retrieval content, which suggests an early discovery of dominant guardrail patterns followed by slower refinement. In practical terms, the retrieval mechanism appears to be selective rather than expansive. It repeatedly reuses a compact knowledge set while only occasionally introducing genuinely new triples.

5.3 Efficiency and Trade-offs

The runtime analysis in **Fig. 10** shows that KG infusion introduces measurable but uneven overhead. The aligned latency comparison indicates that KG-infused inference is typically slower

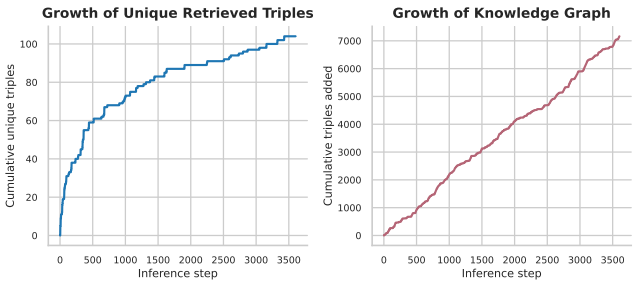

Fig. 9 Retrieval growth dynamics in the KG-infused pipeline: cumulative unique retrieved triples saturate over inference steps while cumulative KG augmentation (triples added) continues to grow.

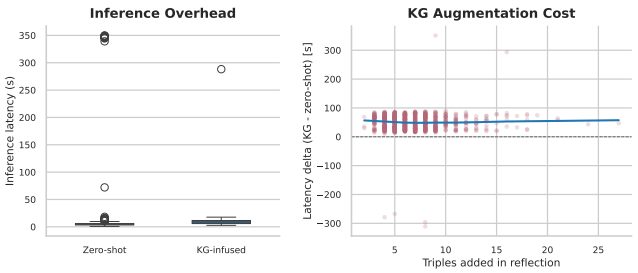

Fig. 10 Inference overhead and KG augmentation cost. Left: Zero-shot vs. KG-infused latency on matched no-reflection indices (Box-plots show median, IQR, and outliers). Right: Latency delta relative to the number of triples added during reflection.

than the zero-shot baseline, although the gap is modest for most datapoints and is dominated by a small number of large outliers. The boxplot therefore suggests that the main cost is not a uniform slowdown, but occasional expensive reflection-heavy cases.

When we isolate KG augmentation cost and relate it to the number of triples introduced, the added latency is generally positive and remains in the same broad range across most samples. The fitted trend is relatively flat, which implies that the overhead is not primarily driven only by triple count.

Taken together, these results indicate that KG infusion is not free, but the overhead is bounded enough to be interpretable in light of the substantial gains in recognition quality. The augmentation process need not be tightly coupled to real-time inference and can instead be offloaded to more capable models or scheduled during less latency-sensitive periods. The practical trade-off is therefore favourable when the application can tolerate some additional latency in exchange for better performance.

5.4 Limitations of the Study

We interpret the results as evidence under the specific experimental conditions of this study, with several limitations that should be considered. The current pipeline does not yet leverage multi-depth retrieval, so both effectiveness and runtime may change under richer retrieval configurations. In addition, data points are processed independently, without explicit modelling of temporal dependencies across consecutive windows, which limits sequence-level reasoning for transition-sensitive activities. Finally, evaluation coverage remains constrained, and the findings should therefore not be taken as indicative of universal performance across all smart-home HAR settings. Within these bounds, the results demonstrate that KG infusion can improve recognition performance while yielding interpretable retrieval behaviour.

© 2018 Information Processing Society of Japan

6. Conclusion

This study examined whether KG infusion can improve zero-shot smart-home activity recognition while preserving interpretable behaviour and acceptable runtime cost. The core mechanism is a self-reflection-driven augmentation loop. When the small model makes an error, it reflects on its own mistake, converts that reflection into structured guardrail triples, and uses those triples as knowledge for subsequent inference. Across the reported experiments, this KG-infused process consistently improved ranking performance, reduced key confusion patterns, and revealed a retrieval profile dominated by compact activity-location and activity-sensor guardrails. These findings suggest that error-triggered symbolic augmentation can strengthen prediction quality in a practical HAR pipeline.

Acknowledgments This study was supported in part by JSPS KAKENHI Grant Number 25K03107.

References

[1] D. Bouchabou, S. M. Nguyen, C. Lohr, B. LeDuc, and I. Kanellos, "A survey of human activity recognition in smart homes based on iot sensors algorithms: Taxonomies, challenges, and opportunities with deep learning," *Sensors*, vol. 21, no. 18, p. 6037, 2021.

[2] Y. Yin, L. Xie, Z. Jiang, F. Xiao, J. Cao, and S. Lu, "A systematic review of human activity recognition based on mobile devices: overview, progress and trends," *IEEE Communications Surveys & Tutorials*, vol. 26, no. 2, pp. 890–929, 2024.

[3] X. Ye, K. Sakurai, N.-K. C. Nair, and K. I.-K. Wang, "Machine learning techniques for sensor-based human activity recognition with data heterogeneity—a review," *Sensors*, vol. 24, no. 24, p. 7975, 2024.

[4] G. Civitarese, M. Fiori, P. Choudhary, and C. Bettini, "Large language models are zero-shot recognizers for activities of daily living," *ACM Transactions on Intelligent Systems and Technology*, vol. 16, no. 4, pp. 1–32, 2025.

[5] Z. Li, S. Deldari, L. Chen, H. Xue, and F. D. Salim, "Sensorllm: Aligning large language models with motion sensors for human activity recognition," in *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, 2025, pp. 354–379.

[6] S. I. Siam, H. Ahn, L. Liu, S. Alam, H. Shen, Z. Cao, N. Shroff, B. Krishnamachari, M. Srivastava, and M. Zhang, "Artificial intelligence of things: A survey," *ACM Transactions on Sensor Networks*, vol. 21, no. 1, pp. 1–75, 2025.

[7] P. Belcak, G. Heinrich, S. Diao, Y. Fu, X. Dong, S. Muralidharan, Y. C. Lin, and P. Molchanov, "Small language models are the future of agentic ai," *arXiv preprint arXiv:2506.02153*, 2025.

[8] F. Corradini, M. Leonesi, and M. Piangerelli, "State of the art and future directions of small language models: a systematic review," *Big Data and Cognitive Computing*, vol. 9, no. 7, p. 189, 2025.

[9] F. Wang, M. Lin, Y. Ma, H. Liu, Q. He, X. Tang, J. Tang, J. Pei, and S. Wang, "A survey on small language models in the era of large language models: Architecture, capabilities, and trustworthiness," in *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, 2025, pp. 6173–6183.

[10] B. Chen, Z. Zhang, N. Langrené, and S. Zhu, "Unleashing the potential of prompt engineering for large language models," *Patterns*, vol. 6, no. 6, 2025.

[11] Z. Hu, P. Yang, F. Liu, Y. Meng, and X. Liu, "Prompting large language models with knowledge-injection for knowledge-based visual question answering," *Big Data Mining and Analytics*, vol. 7, no. 3, pp. 843–857, 2024.

[12] D. J. Cook, A. S. Crandall, B. L. Thomas, and N. C. Krishnan, "Casas: A smart home in a box," *Computer*, vol. 46, no. 7, pp. 62–69, 2012.

[13] L. Arrotta, C. Bettini, and G. Civitarese, "The marble dataset: Multi-inhabitant activities of daily living combining wearable and environmental sensors data," in *International conference on mobile and ubiquitous systems: computing, networking, and services*. Springer, 2021, pp. 451–468.

[14] F. Gu, M.-H. Chung, M. Chignell, S. Valaee, B. Zhou, and X. Liu, "A survey on deep learning for human activity recognition," *ACM Computing Surveys (CSUR)*, vol. 54, no. 8, pp. 1–34, 2021.

[15] M. Thukral, S. G. Dhekane, S. K. Hiremath, H. Haresamudram, and T. Ploetz, "Layout-agnostic human activity recognition in smart homes through textual descriptions of sensor triggers (tdost)," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 9, no. 1, pp. 1–38, 2025.

[16] Y. Lu, L. Zhou, A. Zhang, M. Wang, S. Zhang, and M. Wang, "Research on designing context-aware interactive experiences for sustainable aging-friendly smart homes," *Electronics*, vol. 13, no. 17, p. 3507, 2024.

[17] S. K. Hiremath, Y. Nishimura, S. Chernova, and T. Plötz, "Bootstrapping human activity recognition systems for smart homes from scratch," *Proceedings of the ACM on interactive, mobile, wearable and ubiquitous technologies*, vol. 6, no. 3, pp. 1–27, 2022.

[18] D. Tam, R. R. Menon, M. Bansal, S. Srivastava, and C. Raffel, "Improving and simplifying pattern exploiting training," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 2021, pp. 4980–4991.

[19] E. Park, D. Jeon, S. Kim, I. Kang, and S.-H. Na, "Lm-bff-ms: Improving few-shot fine-tuning of language models based on multiple soft demonstration memory," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2022, pp. 310–317.

[20] A. Yang, A. Li, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Gao, C. Huang, C. Lv *et al.*, "Qwen3 technical report," *arXiv preprint arXiv:2505.09388*, 2025.

[21] L. Advani, "When small models are right for wrong reasons: Process verification for trustworthy agents," *arXiv preprint arXiv:2601.00513*, 2026.

[22] S. O. U. Islam, A. Lauscher, and G. Glavaš, "How much do llms hallucinate across languages? on realistic multilingual estimation of llm hallucination," in *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, 2025, pp. 29 077–29 098.

[23] C. Sun, Y. Li, D. Wu, and B. Boulet, "Onioneval: An unified evaluation of fact-conflicting hallucination for small-large language models," *arXiv preprint arXiv:2501.12975*, 2025.

[24] V. Romero, T. Matsui, Y. Matsuda, H. Suwa, and K. Yasumoto, "Schema-grounded agents for enabling emergent multi-device reasoning in smart environments," in *International Conference on Distributed Computing in Smart Systems and the Internet of Things (DCOSS-IoT)*, 2026, to appear.

[25] N. F. Liu, K. Lin, J. Hewitt, A. Paranjape, M. Bevilacqua, F. Petroni, and P. Liang, "Lost in the middle: How language models use long contexts," *Transactions of the association for computational linguistics*, vol. 12, pp. 157–173, 2024.

[26] Y. Du, M. Tian, S. Ronanki, S. Rongali, S. Bodapati, A. Galstyan, A. Wells, R. Schwartz, E. A. Huerta, and H. Peng, "Context length alone hurts llm performance despite perfect retrieval," *arXiv preprint arXiv:2510.05381*, 2025.

[27] B. Upadhayay, V. Behzadan, and A. Karbasi, "Cognitive overload attack: Prompt injection for long context," *arXiv preprint arXiv:2410.11272*, 2024.

[28] T. Gu, X. H. Wang, H. K. Pung, and D. Q. Zhang, "An ontology-based context model in intelligent environments," *arXiv preprint arXiv:2003.05055*, 2020.

[29] L. Buoncompagni, S. Y. Kareem, and F. Mastrogiovanni, "Human activity recognition models in ontology networks," *IEEE transactions on cybernetics*, vol. 52, no. 6, pp. 5587–5606, 2021.

[30] L. Arrotta, G. Civitarese, and C. Bettini, "Probabilistic knowledge infusion through symbolic features for context-aware activity recognition," *Pervasive and Mobile Computing*, vol. 91, p. 101780, 2023.

[31] L. Arrotta, C. Bettini, G. Civitarese, and M. Fiori, "Contextgpt: Infusing llms knowledge into neuro-symbolic activity recognition models," in *2024 IEEE International Conference on Smart Computing (SMARTCOMP)*. IEEE, 2024, pp. 55–62.

[32] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel *et al.*, "Retrieval-augmented generation for knowledge-intensive nlp tasks," *Advances in neural information processing systems*, vol. 33, pp. 9459–9474, 2020.

[33] B. Wang, Z. Wang, X. Wang, Y. Cao, R. A Saurous, and Y. Kim, "Grammar prompting for domain-specific language generation with large language models," *Advances in Neural Information Processing Systems*, vol. 36, pp. 65 030–65 055, 2023.

[34] W. Liu, P. Zhou, Z. Zhao, Z. Wang, Q. Ju, H. Deng, and P. Wang, "K-bert: Enabling language representation with knowledge graph," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, no. 03, 2020, pp. 2901–2908.

[35] H. Huang, C. Chen, Z. Sheng, Y. Li, and W. Zhang, "Can llms be good graph judge for knowledge graph construction?" in *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, 2025, pp. 10 940–10 959.

© 2018 Information Processing Society of Japan