Ask the Sensors

Grounded, Explainable Activity Question Answering from Wearable Signals

Read this brief in full before you begin. It defines what you will build, the exact form your answers must take, how you will be assessed, and the rules you must follow. A few logistical details, shown in brackets, will be confirmed separately.

The Scenario

Consider Aparna, seventy-two years old, living on her own while her son works in another city. Throughout the day she wears an ordinary smartwatch. Her son does not want a surveillance feed of her life. He wants to ask a few plain questions and trust the answers. Did she take her usual morning walk? How much of the afternoon did she spend resting? She mentioned feeling unsteady around noon, so was she doing anything strenuous at that time? A physiotherapist following her recovery has related questions of his own: is her walking time increasing from one week to the next, and can that claim be checked against the signal itself rather than taken on trust?

The watch already records the raw material for every one of these answers, as continuous accelerometer and gyroscope streams. The difficulty is that raw motion signals mean nothing to a person reading them, and an ordinary activity classifier improves matters only a little: it emits a label such as walking at each instant, but it never says for how long, how often, at what time, or on what basis. In a healthcare setting that last point is decisive. A clinician will not act on the bare verdict that the machine said so. The answer has to arrive with the evidence that supports it, pointing to the exact stretch of signal and the reason it reads as walking rather than standing.

What You Will Build

Your task is to build the system that closes this gap. It is a sensor question-answering service: given a natural-language question and a multimodal sensor recording, it returns an accurate answer about the activity in question and grounds that answer in the specific sensor evidence that justifies it. This grounding is the point of the challenge. An answer produced by language reasoning alone, with no reference to the recorded signal, does not meet the requirement.

The system accepts two inputs. The first is a sensor recording: a timestamped, multivariate time series from a triaxial accelerometer, with channels X, Y and Z, and a triaxial gyroscope, again with channels X, Y and Z. The second is a natural-language query about the activity observed in that recording. For every query the system returns one structured answer, in the format defined in the next section. Depending on the question, that answer may name an activity, confirm or deny a claim, quantify a duration or a count, place an event in time, compare two activities, or reason about a behavior that carries no fixed label. In each case, any claim about activity must be traceable to the signal.

The Data

You will develop and validate your system on the ExtraSensory dataset (http://extrasensory.ucsd.edu/), a large collection of wearable recordings gathered in the wild from sixty users going about their ordinary daily lives. For this challenge you will use two modalities only, the accelerometer and the gyroscope, and you will restrict attention to seven activity and posture classes: lying down, sitting, standing in place, standing and moving, walking, running, and bicycling. Resample every sensor stream to 25 Hz before processing, so that all groups work from a common time base.

Because ExtraSensory was recorded in natural conditions rather than in a laboratory, the data is realistic and, for that reason, awkward. Sampling is uneven, stretches of data are missing, labels are self-reported and can overlap, and the class distribution leans heavily toward sedentary behavior. Treat these properties as part of the problem rather than as blemishes to be sidestepped. A solution that stays reliable in the presence of noise, gaps, and class imbalance is worth more than one that performs well only on a clean subset.

The Output Format

Every response your system produces, for every task, must use the following fields, in this order. The uniform format lets us score answers consistently across all groups.

```
Answer:              <direct answer to the query, or N/A>
Activity/Event:      <activity or event, or N/A>
Evidence:
    Timestamp(s):      <time range or ranges, or N/A>
    Sensor Modality:   <accelerometer, gyroscope, or both, or N/A>
    Sensor Channel(s): <for example Acc X/Y/Z, Gyro X/Y/Z, or All, or N/A>
Explanation:         <reasoning grounded in the observed signal, or N/A>
```

Three points govern the format. Express every timestamp in a single, consistent time base across all of your answers, either as absolute clock time in HH:MM:SS or as seconds from the start of the recording, and state which convention you have chosen. The examples below use seconds from the start. The evidence fields may be reported as N/A in Task 1, where only the answer is scored; they are expected in Task 2; and they are required and directly assessed in Tasks 3 and 4. Wherever you can support an answer with evidence, do so, even when it is not strictly required.

The Four Tasks

The challenge is organized as four tiers of increasing difficulty. A single system should handle all four, since the evaluation will mix questions drawn from every tier.

Task 1: Activity Identification

The first tier tests whether the system reads the sensor stream correctly and names the activity being performed. Questions take two forms: open identification, such as asking what the user is doing, and binary verification, such as asking whether the user is running. Only the answer is scored at this tier, so the evidence fields may be left as N/A, though you are welcome to fill them.

```
Query: "What activity is the user performing?"
Answer: Walking
Activity/Event: Walking
Evidence:
    Timestamp(s): N/A
    Sensor Modality: N/A
    Sensor Channel(s): N/A
Explanation: N/A
```

```
Query: "Is the user running?"
Answer: Yes
Activity/Event: Running
Evidence:
    Timestamp(s): N/A
    Sensor Modality: N/A
    Sensor Channel(s): N/A
Explanation: N/A
```

Task 2: Temporal and Quantitative Reasoning

The second tier asks the system to reason across the whole recording rather than a single window. Questions concern how long an activity lasted, how many times it occurred, when a particular event took place, and how two activities compare in total time. Answers must be computed from the detected activity intervals and must reflect the user's behavior over the full recording. Evidence is expected here: report the intervals your answer rests on.

```
Query: "How long was the user walking?"
Answer: 700 seconds
Activity/Event: Walking
Evidence:
    Timestamp(s): 905 to 1420, 2110 to 2295 (seconds from start)
    Sensor Modality: Accelerometer, Gyroscope
    Sensor Channel(s): All
Explanation: Walking was detected in two separate intervals, of 515 and 185
             seconds, which sum to 700 seconds.
```

```
Query: "Did the user spend more time walking or running?"
Answer: Walking
Activity/Event: Walking, Running
Evidence:
    Timestamp(s): Walking = 3480 seconds total, Running = 1440 seconds total
    Sensor Modality: Accelerometer, Gyroscope
    Sensor Channel(s): All
Explanation: Total walking time exceeded total running time over the recording.
```

Task 3: Evidence Grounding

The third tier makes the evidence itself the object of assessment. A correct answer is necessary but not sufficient: the system must cite the specific part of the signal that justifies the answer and explain why it does. Every response at this tier must supply the answer, the temporal location of the supporting evidence, the sensor modality it is drawn from, the specific channel or channels, and the reasoning that ties the cited pattern to the conclusion.

```
Query: "Did the user begin running at any point, and if so, when?"
Answer: Yes, running began at 1512 seconds
Activity/Event: Onset of running
Evidence:
    Timestamp(s): 1512 to 1980 (seconds from start)
    Sensor Modality: Accelerometer, Gyroscope
    Sensor Channel(s): All
Explanation: A sustained rise in accelerometer magnitude at a higher step
             frequency, together with larger gyroscope oscillations, marks the
             transition from lower-intensity gait to running at the cited time.
```

Task 4: Open-World Activity Reasoning

The final tier moves past the fixed set of labels. Here the questions probe the meaning of the motion rather than its class name, and answering them calls for semantic reasoning over the signal: inferring a broad behavior, judging whether the observed evidence is consistent with a described scenario, explaining why a pattern looks the way it does, or reading body dynamics and temporal structure. The behavior in question may be one the system was never trained to name, so the answer has to be argued from the signal. Both the evidence and the explanation are mandatory at this tier.

```
Query: "Did the user lie down for a prolonged period?"
Answer: Likely yes
Activity/Event: Prolonged lying down
Evidence:
    Timestamp(s): 6300 to 9420 (seconds from start)
    Sensor Modality: Accelerometer, Gyroscope
    Sensor Channel(s): All
Explanation: A long, continuous stretch of near-zero acceleration variance and
             minimal gyroscope activity, well beyond any brief stationary pause,
             is consistent with sustained rest rather than a transient stop.
```

```
Query: "Was the user using a wheeled or pedal-based mode of movement?"
Answer: Yes
Activity/Event: Unknown outdoor physical activity, consistent with cycling
Evidence:
    Timestamp(s): 2400 to 3120 (seconds from start)
    Sensor Modality: Accelerometer, Gyroscope
    Sensor Channel(s): All
Explanation: The segment shows smooth, continuous, cyclic acceleration at a steady
             cadence, without the discrete heel-strike spikes of walking or running,
             accompanied by sustained periodic gyroscope oscillation consistent with
             pedaling and balance, which points to a low-impact wheeled mode.
```

Accuracy and the Cost of Running It

A system that answers well is only half of the goal. Ubiquitous computing runs on devices that are small, battery powered, and often disconnected, and a method that lives only in the cloud gives up the two things that matter most in a health setting: privacy and availability. This challenge therefore asks you to be clear about what your solution costs, not only about how well it answers.

At a minimum, alongside your accuracy figures, report the resource cost of your system: the model size, both in parameters and on disk, the peak memory used during inference, and the time taken to answer a single query. Where you can, add processor utilization and an estimate of the energy spent

per query. Measure these on a defined target and state clearly what that target is, whether a laptop processor, a single-board computer such as a Raspberry Pi, an embedded accelerator such as a Jetson-class board, a mid-range Android phone, or a constrained environment that you emulate. Clear and consistent reporting matters more than the particular hardware you choose.

Extra credit is available for a solution that runs accurately on a small, resource-constrained edge device. To earn it, do not merely assert that the model is small; demonstrate the tradeoff between accuracy and overhead. Present at least two operating points, for instance a full model and a compressed one obtained through quantization, pruning, or knowledge distillation, and plot accuracy against one or more cost axes so that the shape of the tradeoff is visible. A solution that trades a small amount of accuracy for a large reduction in cost, and shows the curve that proves it, is exactly what this part of the challenge rewards.

How You Will Be Evaluated

Both the sensor recordings and the questions used for grading will be released only at evaluation time. Keeping the evaluation set undisclosed lets us measure how well your solution generalizes rather than how well it fits data you have already seen, and the set will include difficult and edge-case questions chosen to probe robustness. Build for the general problem, not for a leaderboard you can study in advance.

This is a design challenge, not a coding test. Marks reward the quality of your design, the correctness of your implementation, and the accuracy and performance your system reaches, not the cleverness or the sheer volume of code. In practice we will weigh four things: whether the design is sound and well reasoned for the problem, whether the implementation is correct and reproducible from your repository, how accurately the system answers across the four tiers, and how convincingly it grounds its answers in real sensor evidence. Efficiency and the accuracy-versus-overhead tradeoff are assessed on top of these and carry the extra credit described above.

The following weighting is indicative and will be confirmed by the instructor.

Dimension	What we look for	Weight
Design and approach	Soundness of the overall design and the reasoning behind it, given the constraints of the problem	25%
Correctness and reproducibility	A correct implementation that the teaching team can rerun from your repository to obtain your reported results	20%
Accuracy across the four tasks	Answer quality on identification, on temporal and quantitative reasoning, and on open-world reasoning	35%
Evidence grounding	Whether the cited timestamps, modalities, and channels are correct and clearly support the stated answer	20%
Efficiency and tradeoff	Accuracy held against measured resource cost, shown as a clear accuracy-versus-overhead curve	up to +10% (extra credit)

Accuracy Reporting and Required Figures

The answers in this challenge are of several kinds: activity labels, yes or no verdicts, numbers such as a duration or a count, time intervals, and free-form judgments. No single number captures performance across all of them, so report a headline figure together with a breakdown by question type, and state clearly which rule defines correctness for each type, since a categorical answer is scored by exact match, a numeric answer by closeness, and a temporal answer by overlap. Define the overall QA accuracy as the fraction of all questions judged correct under their respective rules, macro-averaged across the question types so that the abundant easy sedentary examples do not dominate the score.

At a minimum, present the following five figures. Plotting individual queries along the x-axis is useful for qualitative error analysis, but the reportable figures aggregate by type.

1. **Accuracy by question type**. A grouped bar chart with the question type on the x-axis (identification, verification, duration, count, comparison, grounding, and open-world reasoning) and the accuracy metric on the y-axis, with a final bar for the overall score. The caption must state the correctness rule used for each group, because that rule differs across groups.
2. **Activity confusion matrix.** A heatmap over the seven classes for the recognition backbone, with the true activity on the rows and the predicted activity on the columns, reported alongside per-class precision, recall, and F1. It exposes the systematic confusions that accuracy hides, such as sitting against standing in place, or walking against running.
3. **Accuracy versus strictness.** A line plot in which, for numeric answers, the x-axis is the error tolerance, and for temporal answers and cited intervals, the x-axis is the IoU threshold from 0.1 to 0.9; the y-axis is the fraction of answers accepted. This reveals whether the misses are near or wild, and the value read off at a fixed threshold gives one comparable number.
4. **Accuracy versus overhead.** A scatter plot with a resource-cost axis on the x-axis (model size in MB, single-query latency, peak memory, or energy per query) and the overall QA accuracy on the y-axis, where each point is one configuration, such as the full model, an 8-bit quantized model, a pruned model, or a distilled model, and the non-dominated points are joined into a Pareto frontier. This is the figure that supports the edge extra credit.
5. **Robustness curve**. A line plot with a controlled degradation of the input on the x-axis (added sensor noise, the percentage of dropped samples, or a sampling rate below 25 Hz) and accuracy on a fixed question set on the y-axis. A curve that stays flat as conditions worsen is stronger evidence of a usable system than any single clean number.

The metric behind each figure depends on the kind of answer, and is computed as follows.

For categorical answers, which include activity identification, comparison, and yes or no plausibility, begin with accuracy, the number of exactly matching answers divided by the total. Because the data is heavily imbalanced, also report macro-F1: for each class compute precision as TP/(TP+FP) and recall as TP/(TP+FN), combine them as F1 = 2 × precision × recall / (precision + recall), and average the per-class values; balanced accuracy, the mean of the per-class recall, serves the same end. For binary verification, report precision, recall, and F1 on the positive class together with specificity, since plain accuracy hides a bias toward answering no.

For numeric answers, meaning durations and counts, exact match is too harsh, so use accuracy within tolerance: an answer is correct when its absolute error is at most a chosen threshold, whether absolute, for instance a few seconds, or plus or minus one for a count, or relative, for instance within about ten percent for a duration. Report the size of the error as well, as the mean absolute error, the average of the absolute differences between the predicted and true values, and optionally the mean absolute percentage error for durations, since these convey the typical error independent of any threshold.

For temporal answers and cited intervals, meaning event times, onsets, and evidence spans, use the Intersection over Union between a predicted interval and the ground-truth interval, that is the length of their overlap divided by the length of their union. When several intervals are involved, match the predicted to the true by overlap and average, or compute a temporal precision as overlap over predicted length and a recall as overlap over true length and take their F1. The grounding accuracy is then the fraction of intervals whose IoU meets the threshold, which is what the accuracy-versus-strictness figure traces.

For evidence grounding as a whole, combine the pieces under one rule: an answer counts as grounded and correct only when the answer is correct, the cited interval reaches the IoU threshold, and the cited modality and channels match the reference. Report this figure beside the plain answer accuracy, since the gap between the two is the price of demanding evidence and is worth showing on its own. A lighter check that needs no interval labels is grounding precision, the fraction of answers whose cited interval, according to the ground-truth activity labels, in fact contains the activity that the answer names.

For open-world reasoning, score the answer categorically wherever a reference exists by mapping it to the intended broad behavior. The explanation calls for more than a correctness check, so judge its faithfulness and plausibility with a short rubric on a one to five scale, applied either by human graders or by a language model given a fixed rubric, rating whether the reasoning cites real signal features, whether those features support the conclusion, and whether the conclusion is plausible. Report the mean rubric score, and the level of agreement between graders where the grading is done by humans; embedding similarity to a reference explanation is a cheap automatic proxy but should stay secondary, because it rewards surface overlap.

Rules and Logistics

Groups and the GitHub Repository

Work in your assigned groups of 2-3 members each. Every member is expected to contribute, and your report must include a short statement of what each member did. Each group keeps its own GitHub repository for the challenge, created at the start and used throughout, and grants access to the teaching team [sandipc-iitkgp, sayantan-kuila, debjit2001]. The commit history should reflect real, incremental work over the period of the challenge rather than a single upload at the end, since that history is part of how we judge each group's effort. The repository must contain your source code, a README that explains how to set up the environment and reproduce your results, the scripts or configuration needed to run the system on a fresh recording in the required output format, and your report. Do not commit the raw dataset itself; commit the code and instructions that fetch or prepare it.

Academic Integrity: AI and Plagiarism

The AI-use and plagiarism policies announced for CS60055 apply to this challenge in full and without exception. The design and the reasoning must be your group's own. Every external resource that you build upon, whether a dataset, a pretrained model, a software library, or a fragment of code, must be cited precisely at the point where you use it. Copying another group's work, in whole or in part, is not permitted. Where you use an AI tool at any stage of development, disclose it: include a short statement in your report that names the tools and describes what each was used for. Fabricated results, misattributed data, and undisclosed reuse are treated as violations under the course policy. If you are unsure whether a particular use is allowed, ask before you rely on it.

Deliverables

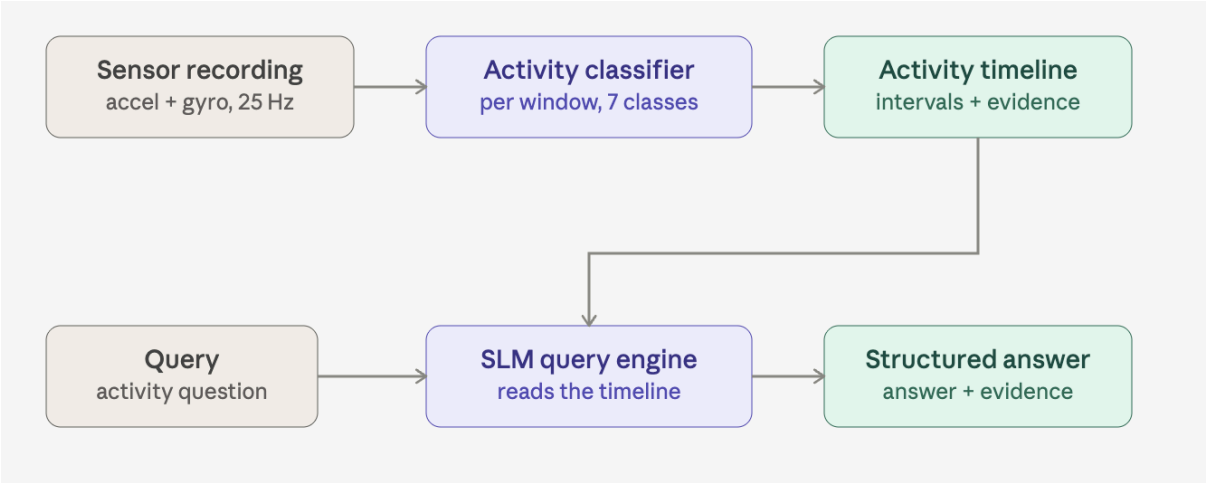
Submit the following by the deadline given below.

6. The GitHub repository, containing your code, the README, and the reproducible setup described above.
7. A concise technical report of about 10-12 pages, covering the problem as you frame it, your design and the reasons for it, the implementation, your results across the four tasks, the accuracy-versus-overhead analysis if you attempt the extra credit, the per-member contribution statement, and the AI-use disclosure.
8. A runnable system that, given a data recording and a set of questions at evaluation time, produces answers in the required output format.
9. A short demonstration or presentation, based on the schedule announced in the class.

A Suggested Starting Point

One reasonable way to approach the problem, though certainly not the only one, is to think of the system in layers. A preprocessing layer cleans and resamples the streams and divides them into analysis windows. A recognition layer assigns activity evidence to those windows, using either engineered features with a light classifier or a compact neural model; keeping this layer small is what will later let you target the edge. An aggregation layer turns the per-window evidence into the intervals, durations, counts, and transitions that the temporal questions ask about. A final interface layer maps a natural-language question to the right operation over that structured evidence and writes the result in the required format. Whatever design you settle on, the grounding requirement constrains this last layer: the language it produces must be tied to evidence that the earlier layers found, not generated freely. Begin with a small, correct pipeline that runs end to end over the seven activities, confirm that it produces well-formed output for a few questions of each type, and only then push on accuracy, on the quality of your evidence, and on efficiency.

The next figure shows a possible pipeline that you can explore. However, put up your innovative thoughts here – this is just a starting point for you.



Related Readings:

[1] https://dl.acm.org/doi/10.1145/3699765

[2] https://dl.acm.org/doi/abs/10.1145/3810210

[3] https://dl.acm.org/doi/pdf/10.1145/3749496

[4] https://dl.acm.org/doi/abs/10.1145/3699747

Language Models:

It is suggested that you explore open-source SLMs such as Ministral-3, Llama 3.2 (1B & 3B), or Qwen2.5/Qwen3 (0.5B, 1.5B, 3B) or something similar. You are free to choose your design and model, but justify your choice in the report.