Concept Bottleneck Model

Concepts and labeling guidelines

- Choose 3–5 reference windows per concept to define what you consider “0”, “0.5”, and “1”.
- Label everything else *relative* to those anchors to keep consistency.
- The following guidelines are defined for an ordinal scale of measurement, if the data is too noisy, use an nominal scale to simplify the labeling process.

1. Motion Intensity

What you’re labeling: how strong the movements appear over the 3 s window.

Visual cue: amplitude and spread of x/y/z signals.

Guideline:

- Almost flat → `0.0–0.1`
- Gentle oscillations (e.g., slow walking) → `0.4–0.6`
- Large, frequent oscillations (e.g., jogging/running) → `0.8–1.0`
- If intensity changes within the window, take the *average perceived strength*.

Rationale:

The CNN will learn a regression mapping from signal variance and energy patterns to a continuous motion intensity score.

2. Periodicity

What you’re labeling: how regular and rhythmic the motion is.

Visual cue: consistency of peaks and troughs over time.

Guideline:

- Random/no rhythm → `0.0–0.2`
- Some repeating patterns but inconsistent → `0.4–0.6`
- Clear, repeating oscillation with stable frequency → `0.8–1.0`

Rationale:

The CNN (and especially an LSTM if you add one later) will capture repeating temporal motifs.

You're effectively giving it a continuous supervision signal about how *cyclic* the window looks

3. Vertical Dominance

What you're labeling: whether vertical motion dominates over horizontal motion in the window.

Visual cue: compare amplitude of z-axis to x and y.

Guideline (binary):

- 1 → vertical axis *clearly* stronger than x and y (e.g., stairs, jumps, vertical displacement visible).
- 0 → vertical comparable to or weaker than x/y (e.g., walking, jogging, sitting, lying).

Rationale:

You only want to flag windows where vertical movement is *unambiguously dominant*. Ambiguous or mixed windows count as 0.

4. Static Posture

What you're labeling: whether the subject remains essentially still.

Visual cue: all axes are flat or near-constant, low amplitude, no distinct oscillation.

Guideline (binary):

- 1 → signals are nearly flat (e.g., sitting, standing, lying still).
- 0 → visible motion, any repeated oscillation or bursts (e.g., walking, jogging, stairs).

Rationale:

We're marking clearly stationary segments, not just "low intensity." If there's movement, however small, it's a 0.

ML Pipeline

1. Manually label 200 windows of 3s each
2. Development Workflow (70/15/15 Split)

Stage	Data Used	Purpose	What Happens	Output / Decision
1. Data Split	100% of dataset	Partition data into training, validation, and test sets	Split data randomly (or stratified, if classification) into:• 70% Train• 15% Validation• 15% Test	3 non-overlapping subsets for modeling
2. Cross-Validation (Tuning Phase)	*Within the 70% training set only*	Optimize hyperparameters and estimate model generalization	Perform **k-fold CV** (usually k=5):• Split 70% data into 5 folds• Train on 4 folds, validate on 1 fold• Repeat for all folds and average scores	Performance estimate for each hyperparameter configuration
3. Hyperparameter Selection	*CV results from training set*	Identify the best configuration	Compare mean CV scores across parameter sets	Select best hyperparameter set (e.g., best learning rate, depth, etc.)
4. Validation Check	15% validation set	Check generalization on unseen data after tuning	Retrain model on **full 70% training data** with chosen hyperparameters, then evaluate on validation set	Validation performance — confirms whether the model generalizes or overfits
5. Optional Fine-Tuning	Training + Validation (if needed)	Slightly adjust or confirm model behavior	If validation underperforms, revisit tuning	Possibly refined hyperparameters

Stage	Data Used	Purpose	What Happens	Output / Decision
			space or adjust regularization	
6. Final Training	85% (Train + Validation)	Train final production model	Retrain model using best hyperparameters on combined Train+Validation data	Fully trained final model
7. Final Evaluation	15% test set	Get unbiased performance estimate	Evaluate the final model once on the held-out test set	Final accuracy / F1 / MSE — reported metric

3. **Human consistency check**: relabel 30-50 examples at random and calculate Cohen's Kappa to test the degree of subjectivity.
4. **Full CBM training**: Freeze the weights of each concept model, input raw data into a concept vector which yields all concept activations, train a classifier on these vectors to predict the activity label
5. **Model evaluation:** Compare CBM against a black-box CNN trained on raw data, visualize feature activations per label (Grad-CAM), concept-label correlation (SHAP), assess concept completeness (residual analysis)