

# Time-Series Fidelity Evaluation

## UCI Human Activity Recognition with DFM-Mosaic Simulation

Independent Study: Information-Theoretic Fidelity of Ground Truth vs Simulated Data

### 1 Overview

This document describes the end-to-end pipeline for time-series fidelity analysis on the UCI Human Activity Recognition (HAR) dataset. It covers data characteristics and units, windowing and sampling, preprocessing, the DFM-mosaic simulator, and a multi-domain fidelity suite. Each metric is defined mathematically and its computation protocol is specified, including the axes over which averages are taken and how windows are pooled. Results are produced globally and per activity class.

### 2 Dataset, Signals, and Windowing

#### 2.1 Signals and Units

We use the *UCI Human Activity Recognition Using Smartphones* dataset. Each example is a fixed-length multivariate time series window with the following channels:

[body\_acc\_x, body\_acc\_y, body\_acc\_z, body\_gyro\_x, body\_gyro\_y, body\_gyro\_z, total\_acc\_x, total\_acc\_y, total\_acc\_z]

- Sampling frequency:  $f_s = 50$  Hz  $\Rightarrow$  sampling period  $\Delta t = 0.02$  s.
- Window length:  $T = 128$  samples  $\Rightarrow$  2.56 s per window.
- Reported units in the original dataset: accelerations in  $g$  and angular velocities in rad/s. After standardization, metrics operate on unitless standardized signals.

#### 2.2 Activities and Labels

There are six activity classes:

WALKING, WALKING\_UPSTAIRS, WALKING\_DOWNSTAIRS, SITTING, STANDING, LAYING.

Labels are mapped to integers  $y \in \{0, \dots, 5\}$  using the dataset’s provided split. We preserve the official train and test partition.

### 3 Preprocessing

#### 3.1 Loading and Shapes

From the raw text files (`Inertial Signals/*_train.txt` and `*_test.txt`), we build:

$$X_{\text{train}} \in \mathbb{R}^{N_{\text{tr}} \times 128 \times 9}, \quad X_{\text{eval}} \in \mathbb{R}^{N_{\text{ev}} \times 128 \times 9}, \quad y_{\text{train}} \in \{0, \dots, 5\}^{N_{\text{tr}}}, \quad y_{\text{eval}} \in \{0, \dots, 5\}^{N_{\text{ev}}}.$$

### 3.2 Standardization

We compute per-channel mean and standard deviation on the training set only and apply the same normalization to train and eval:

$$\hat{X}_{\text{train}}[i, t, c] = \frac{X_{\text{train}}[i, t, c] - \mu_c}{\sigma_c}, \quad \hat{X}_{\text{eval}}[i, t, c] = \frac{X_{\text{eval}}[i, t, c] - \mu_c}{\sigma_c}.$$

All subsequent metrics are computed on standardized windows  $\hat{X}$ . Cached arrays and scalar statistics are stored under `experiments/time_series_har/data.cache` for reproducibility.

## 4 Simulated Data: DFM-Mosaic

### 4.1 Motivation

The Deviation-From-Mean (DFM) mosaic is a non-neural simulator that preserves activity-specific dynamics by recombining empirical deviations from class templates. Simulation is class-conditional.

### 4.2 Smoothing and Templates

For each window and channel, apply a Savitzky–Golay filter with window  $w_{\text{sg}} = 15$  samples and polynomial order  $p_{\text{sg}} = 3$ , producing a smoothed signal  $\tilde{x}(t)$ . For each class  $k$  and channel  $c$ , define class envelopes and template mean over all smoothed windows in class  $k$ :

$$\min_{k,c}(t), \max_{k,c}(t), \text{mean}_{k,c}(t) = \frac{1}{2}(\min_{k,c}(t) + \max_{k,c}(t)).$$

### 4.3 Deviation-From-Mean and Mosaic

Define normalized deviation for a smoothed window, using a small  $\varepsilon$  for stability,

$$\text{DFM}_c(t) = \frac{\tilde{x}_c(t) - \text{mean}_{k,c}(t)}{\max(|\text{mean}_{k,c}(t)|, \varepsilon)}.$$

To synthesize a new window for class  $k$ , choose  $S \sim \{3, \dots, 6\}$  multi-channel DFM segments, each of length  $L_s \in [0.15T, 0.35T]$  samples. Concatenate the selected DFM segments with short linear ramps at seams to avoid discontinuities. Reconstruct the signal multiplicatively and add realistic residual noise:

$$x_c^{\text{sim}}(t) = \text{mean}_{k,c}(t)(1 + \widetilde{\text{DFM}}_c(t)) + \alpha r_{k,c}(t),$$

where  $\widetilde{\text{DFM}}$  is the mosaicked DFM,  $r_{k,c}(t)$  is a residual from a randomly selected real window in class  $k$ , and  $\alpha$  is a residual scale (default  $\alpha = 1.0$ ). Class counts are sampled in proportion to the class prior of the reference split and total  $n$  is matched to the evaluation set.

### 4.4 Parameter Summary

Table 1: DFM-Mosaic parameters and units

Parameter	Meaning	Default
$w_{\text{sg}}, p_{\text{sg}}$	Savitzky–Golay window and polynomial order	15 samples, 3
Segments	Number of DFM segments per window	3 to 6 (integer)
Segment length	Fraction of window length $T = 128$	0.15 to 0.35
Reconstruction	Additive or multiplicative	multiplicative
Residual scale $\alpha$	Noise reinjection multiplier	1.0
Seam smoothing	Linear ramp at boundaries	enabled

## 5 Fidelity Metrics and Exact Computation Protocol

Metrics are computed on standardized windows. Two evaluation scopes are produced:

1. **Global:** pool all evaluation windows across classes, size-match GT and SIM to the same  $n$ .
2. **Per class:** for each activity  $k$ , size-match GT and SIM counts to the minimum available and compute metrics on that subset. Results are reported per channel and per class.

### 5.1 Power Spectral Density Jensen–Shannon (PSD–JS)

**Definition.** For each channel  $c$ , compute Welch PSDs on each window using  $f_s = 50$  Hz,  $n_{\text{perseg}} = 64$ . Average PSDs over windows to obtain mean spectra  $S_c^{(\text{gt})}(f)$  and  $S_c^{(\text{sim})}(f)$ , then normalize them to probability mass functions:

$$P_c^{(\text{gt})}(f) = \frac{S_c^{(\text{gt})}(f)}{\sum_f S_c^{(\text{gt})}(f)}, \quad P_c^{(\text{sim})}(f) = \frac{S_c^{(\text{sim})}(f)}{\sum_f S_c^{(\text{sim})}(f)}.$$

The Jensen–Shannon divergence is

$$D_{\text{JS}}(P_c^{(\text{gt})}, P_c^{(\text{sim})}) = \frac{1}{2}D_{\text{KL}}(P_c^{(\text{gt})} \parallel M_c) + \frac{1}{2}D_{\text{KL}}(P_c^{(\text{sim})} \parallel M_c), \quad M_c = \frac{1}{2}(P_c^{(\text{gt})} + P_c^{(\text{sim})}).$$

**Protocol.** Welch is applied window-wise, then averaged across windows in the chosen scope. Frequency axis is in Hz. PSDs are normalized to remove units and to compare shapes only.

### 5.2 Autocorrelation Difference (ACF– $\Delta$ )

**Definition.** For each channel  $c$ , compute the normalized autocorrelation for lags  $\ell = 0, \dots, L$  with  $L = 20$  on each window:

$$\rho_{c,i}(\ell) = \frac{\sum_t (x_{c,i}(t) - \bar{x}_{c,i})(x_{c,i}(t + \ell) - \bar{x}_{c,i})}{\sum_t (x_{c,i}(t) - \bar{x}_{c,i})^2},$$

where  $i$  indexes windows. Average over windows to obtain  $\bar{\rho}_c^{(\text{gt})}(\ell)$  and  $\bar{\rho}_c^{(\text{sim})}(\ell)$ . Report mean absolute difference for lags 1 to  $L$ :

$$\Delta_{\text{ACF},c} = \frac{1}{L} \sum_{\ell=1}^L \left| \bar{\rho}_c^{(\text{gt})}(\ell) - \bar{\rho}_c^{(\text{sim})}(\ell) \right|.$$

**Protocol.** We cap the number of windows used in the per-window ACF averaging to at most 512 for efficiency. Lags map to physical time as  $\ell \cdot \Delta t$  with  $\Delta t = 0.02$  s so  $L = 20$  corresponds to 0.4 s.

### 5.3 Cross-Channel Correlation Delta (Frobenius)

**Definition.** Flatten time within windows and stack windows, then compute  $9 \times 9$  Pearson correlation matrices across channels for GT and SIM:

$$C^{(\text{gt})} = \text{corr}(\text{reshape}(X^{(\text{gt})}, -1, 9)), \quad C^{(\text{sim})} = \text{corr}(\text{reshape}(X^{(\text{sim})}, -1, 9)).$$

Distance is the Frobenius norm:

$$\Delta_{\text{corr}} = \left\| C^{(\text{gt})} - C^{(\text{sim})} \right\|_F = \sqrt{\sum_{i,j} \left( C_{ij}^{(\text{gt})} - C_{ij}^{(\text{sim})} \right)^2}.$$

**Protocol.** Correlations are computed on standardized values pooled across all samples and timepoints in the given scope. This tests whether inter-sensor coupling is preserved.

#### 5.4 Maximum Mean Discrepancy (MMD) with RBF kernel

**Definition.** Flatten each window to  $x \in \mathbb{R}^{T \cdot C} = \mathbb{R}^{1152}$ . With kernel  $k(x, y) = \exp(-\gamma \|x - y\|^2)$ ,

$$\text{MMD}^2(X, Y) = \mathbb{E}_{x, x'}[k(x, x')] + \mathbb{E}_{y, y'}[k(y, y')] - 2 \mathbb{E}_{x, y}[k(x, y)].$$

**Protocol.** We size-match sets, optionally subsample to at most 2000 windows for tractability, and set  $\gamma$  using the median heuristic on pairwise squared distances over the concatenated sample set. Kernel matrices use zeroed diagonals for the within-set terms and the standard average for cross terms.

## 6 Scopes, Pooling, and Size Matching

- **Global scope:**

- Choose  $n = \min(N_{\text{eval}}^{\text{gt}}, N_{\text{sim}})$ .
- Take the first  $n$  windows from each set after standardization.
- Compute PSD–JS per channel, ACF– $\Delta$  per channel,  $\Delta_{\text{corr}}$  on all channels, and MMD on flattened windows.

- **Per-class scope:**

- For each class  $k$ , let  $n_k = \min(N_k^{\text{gt}}, N_k^{\text{sim}})$ .
- Use the first  $n_k$  windows for GT and SIM in class  $k$ .
- Compute the same metrics as above. PSD–JS and ACF– $\Delta$  are reported per channel.  $\Delta_{\text{corr}}$  is a single scalar per class. MMD is per class using flattened windows.

## 7 Outputs and Figures

### 7.1 Artifacts

- **results\_sim.json:** experiment metadata, simulator parameters, global metrics, per-class metrics, counts.
- Figures in **experiments/time\_series\_har/figures:**
  - Global overlays: PSD and ACF for channel 0.
  - Per-class grids in **figures/per\_class/**:  $3 \times 3$  overlays for PSD and ACF across all channels.

### 7.2 Interpreting Scales

- PSD–JS  $\in [0, \ln 2]$ : lower is more similar spectral shape.
- ACF– $\Delta \geq 0$ : lower indicates closer short-lag temporal dependencies, here up to 0.4 s.
- $\Delta_{\text{corr}} \geq 0$ : lower means inter-sensor relationships are closer.
- MMD  $\geq 0$ : lower indicates closer global distributions in the full window space.

## 8 Reproducibility and Execution

### 8.1 CLI

From the repository root:

```
python -m experiments.time_series_har.run \
  --seed 42 \
  --n_sim 4000 \
  --sg_window 15 --sg_poly 3 \
  --segments_min 3 --segments_max 6 \
  --seglen_min_frac 0.15 --seglen_max_frac 0.35 \
  --residual_scale 1.0 \
  --reconstruct mul \
  --out results_sim.json
```

### 8.2 Determinism

Seeding is applied for simulation and evaluation selection. Standardization parameters are computed on train only and cached. The evaluation size matching step ensures that compared sets have equal counts.

## 9 Current Findings at a Glance

Dynamic activities that contain clear periodic motion, such as **WALKING**, **WALKING\_UPSTAIRS**, and **WALKING\_DOWNSTAIRS**, show low PSD-JS and MMD with modest ACF- $\Delta$  and small  $\Delta_{\text{corr}}$ . Stationary activities such as **SITTING** and **STANDING** exhibit larger deviations, indicating that small-amplitude, low-variance dynamics and subtle cross-channel couplings are harder to mimic with the current DFM-mosaic configuration. These effects are visible in the per-class  $3 \times 3$  grids.

## 10 Next Steps

1. Class-specific simulator tuning for stationary classes: shorter segments, more segments, larger residual scale, or an explicit micro-jitter process.
2. Add learned synthetic generators for 3-way comparisons: GT vs SIM vs SYN with identical metrics.
3. Export LaTeX tables directly from `results_sim.json` for inclusion in manuscripts.
4. Optional utility tests: cross-train accuracy using a lightweight classifier to relate fidelity to downstream performance.

## References

- [1] Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13:723–773, 2012.