

# Supporting Information for “AI-enabled Micro Motion Sensors for Revealing the Random Daily Activities of Caged Mice”

Yifan Liu<sup>1</sup>, Meng Chen<sup>1</sup>, Chishing Chan<sup>2</sup>, Ho-yin Chan<sup>1</sup>, Jianping Wang<sup>1</sup>, Xinge Yu<sup>1</sup>, Xinyue Li<sup>1</sup>, and Wen Jung Li<sup>1</sup>

<sup>1</sup>City University of Hong Kong

<sup>2</sup>CASCUBE Limited

February 22, 2024

## Abstract

This Supporting Information includes:

*Supplementary text describing Preliminary Status Classifier, segmentation methods, model training and validation details; two supplementary tables, two supplementary figures and one supplementary video.*

Corresponding author(s) Email: [hoychan@cityu.edu.hk](mailto:hoychan@cityu.edu.hk); [xinyueli@cityu.edu.hk](mailto:xinyueli@cityu.edu.hk); [wenjli@cityu.edu.hk](mailto:wenjli@cityu.edu.hk)

## Supplementary Text

### Preliminary Status Classifier

FFT energy  $E_f$  of a signal sequence is defined as follows:

$$E_f = \sum_{i=1}^N F_i^2 \quad (1)$$

where  $F_i$  is the  $i^{th}$  amplitude of the Fourier transform of  $a_t$  and  $N$  is the length of the amplitude array.  $a_t$  is the total acceleration, which combines the accelerations in three axes (i.e., x, y, and z) as follows:

$a_t = \sqrt{a_{xt}^2 + a_{yt}^2 + a_{zt}^2}$  (2) The other parameter  $Var(a_t)$  is the acceleration variance of the signal sequence, which is expressed as follows:

$$Var(a_t) = \frac{\sum_{i=1}^n (a_{ti} - \bar{a_t})^2}{n} \quad (3)$$

In Equation (3),  $n$  is the length of the acceleration array.

### Segmentation Method

#### Fixed-window Segmentation (FWS)

The IMU reading of  $t^{th}$  sampling is denoted as  $x_t, y_t, z_t$ , and all the readings from  $1^{st}$  sampling to  $t^{th}$  are  $X_t$ ; therefore, the relationship between  $x_t$  and  $X_t$  is  $X_t = \{x_1, x_2, \dots, x_t\}$ . Each reading consists of 3-axis

acceleration and 3-axis angular velocity, which is expressed as follows:

$x_t = [a_{xt}, a_{yt}, a_{zt}, g_{xt}, g_{yt}, g_{zt}]$  (4) Then, the segmentation process divides the signal  $X_t$  into slices, which is a set of contiguous readings. The slice set is denoted as  $S$  and  $s_1, s_2 \dots s_t \in S$ , and the  $n^{th}$  slice  $s_n$  consisting of the readings between  $a^{th}$  and  $b^{th}$  is as follows:

$s_n = [x_a, x_{a+1}, \dots x_b]$  (5) The window size of the  $n^{th}$  slice is denoted as  $L_n$ . Since FWS usually divides raw sensor signals into same-length slices,  $L_n$  is a constant value for all slices, denoted as  $L$ . In CSM, we set  $L$  to 1 second, and the readings passed into the FWS is divided into 1-second slices. Since the sampling frequency of our sensor is 25, one FWS slice can contain 25 readings.

### Probability Density Function Segmentation (PDFS)

However, not all of the mice behaviors end in or last for 1 second, and the sample length would be less than 1 second. Thus, if a behavior's duration deviates from the length of a predefined fixed window, the quality of the features extracted from that sample would be low, as they could not represent the entire sample. For instance, one of the features is the mean value of accelerations in the x-direction of a slice containing a behavior within a fixed window (e.g., 1 second). If the duration of that behavior is much shorter than 1 second, the slice must contain information of other behavior(s), which would dominate the mean value. Thus, to solve this problem, we design the first Adaptive Window Segmentation (AWS) named the probability density function segmentation (PDFS), which utilizes a probability density function (PDF) with a multivariate Gaussian distribution. The PDF function defines the probability of a reading belonging to a slice and decides whether to extend the slice to include the reading (Mickey & Rencher, 1998).

Due to the uncertainty in the window size and their boundary positions, during the processing, we define the stop boundary point of the previous slice as the changepoint and elapsed time since the last changepoint as the "run".  $r_t$  is the length of the current run if it stops at time  $t$ , and the reading set with run length  $r_t$  is  $x_t^{(r)}$ . The PDF for all the readings of the current run  $x_t^{(r)}$  at time  $t$  falls within a particular range, which can be expressed as follows:

$$f_n(x_t^{(r)}) = \frac{1}{(\sqrt{2\pi})^d \cdot \left| \frac{r_t-1}{r_t} K \right|^{\frac{1}{2}}} e^{-\frac{1}{2} \sum_{i=1}^{r_t} (x_i - \mu)^T \cdot \frac{r_t}{(r_t-1)S} \cdot (x_i - \mu)} \quad (6)$$

First, we define two windows: a base window  $W_b$  with a fixed size ( $b$ ) and an expanding window  $W_e$  with another fixed size ( $e$ ) that is much smaller than that of the base window. Each run starts from the beginning of a  $W_b$ , and we first calculate the probability density ( $p_1$ ) of the data in current run. Then, the run is extended by adding one  $W_e$  to  $W_b$  to form a new temporary segmentation window ( $b + e$ ), and the probability density ( $p_2$ ) of current run is calculated. By comparing the two probability density values, the decision of extending the current window is made. If  $p_1$  is smaller than  $p_2$ , the base window is taken as the segmentation window. Otherwise, the extending window is kept. For the next round, the probabilities before and after adding an expanding window will be calculated again for the comparison and determination of the expanding.

We define the expanding times of the temporary segmentation window as  $n_e$ , and the run length in PDFS at time  $t$  is determined by those three parameters.

$$r_t = b + n_e \cdot e \quad (7)$$

### Bayesian Online Online Changepoint Changepoint Detection Detection Segmentation (BOCDS)

We design the second AWS utilizing one of the popular changepoint detection techniques, the Bayesian Online Changepoint Detection (BOCD). Therefore, this segmentation method is named the Bayesian Online

Changepoint Detection Segmentation (BOCDS). It can generate an accurate posterior prediction of the next data point  $x_{t+1}$  in the sequence based on the received sensor readings  $x_{1:t}$ , which can be expressed as follows:

$$P(x_{t+1}|x_{1:t}) = \sum_{r_t} P(x_{t+1}|r_t, x_t^{(r)}) \cdot P(r_t|x_{1:t}) \quad (8)$$

The details about the equation derivation are referred to in Adams et al.'s work (Adams & MacKay, 2007). Based on this equation, we could predict the next data point and detect the abrupt change in the signal sequence. We first input the four primary parameters required by the model and set one of the boundaries at the stop point of the last slice. Then, we start to calculate several probabilities, including predictability, growth probability, and changepoint probability, starting from the next point. The related prediction is conducted based on these probabilities. If this point is predicted as the changepoint, the current run would stop at this point; otherwise, the run length would increase, and we would perform a similar step for the next data point. Using the BOCDS, the continuous readings are divided into windows of different sizes in real-time.

### CSM-1(PSC+FWS+PDFS) & CSM-2 (PSC+FWS+BOCDS)

Based on the PSC and three segmentation methods mentioned above, we construct two combined segmentation methods (CSMs). The first CSM consists of the PSC, FWS and PDFS and the second is formed by the PSC, FWS and BOCDS. In the first CSM, PSC decides the readings belongs to static or dynamic behaviors. The static readings are divided into same-size segments in FWS, while the dynamic readings are processed by PDFS. For the second CSM, BOCDS replaces PDFS while keeping other setting same with the first CSM.

## Model training and validation

For model training and validation, the confusion matrix is calculated to evaluate the performance of our trained model, along with another three metrics defined as follows:

$$Recall(A) = \frac{L_{TP}(A)}{L_{TP}(A) + L_{FN}(A)} \quad (9)$$

$$Precision(A) = \frac{L_{TP}(A)}{L_{TP}(A) + L_{FP}(A)} \quad (10)$$

$$F_1(A) = \frac{2 \cdot Precision(A) \cdot Recall(A)}{Precision(A) + Recall(A)} \quad (11)$$

where  $L_{TP}(A)$  is the total time length of the true positive case for behavior  $A$ ,  $L_{FN}(A)$  is the total time length of a false negative error for behavior  $A$ , and  $L_{FP}(A)$  is the total time length of a false-positive error for behavior  $A$ . In the usual classification evaluation, the three metrics are calculated based on the true or false predicted case number. In our case, readings are divided into slices of different lengths with the AWSs. Therefore, we evaluate prediction accuracy based on the time lengths of the correctly and incorrectly classified behaviors.

A high recall value indicates that most of the behavior samples are correctly classified, while a high precision value means the percentage of correct prediction is large among the total related prediction (Powers, 2020). To evaluate the overall performance, we calculate the accuracy for the whole training model as follows:

$$Accuracy(total) = \frac{L_{TP}(total)}{L_{total}} \quad (12)$$

where  $L_{total}$  is the total time length of the validation set and  $L_{TP}(total)$  is the total time length of all the true positive cases in the validation set. The overall accuracy is the key parameter representing the overall

effectiveness and accuracy of a classification system, but this accuracy is highly affected by the ratio of samples between different behaviors. If a dataset is not well-balanced between behaviors, the accuracy is dominated by the behaviors with high numbers of samples. Thus, we adapt the average recall value (macro-recall)  $Recall_{avg}$  as the main parameter to represent the model accuracy, which is expressed as follows:

$$Recall_{avg} = \frac{1}{M} \sum_{A=1}^M Recall(A) \quad (13)$$

where  $M$  is the number of behavior types, which equals 8 in our system. As a 4-fold cross validation is employed in this paper to resample our dataset and evaluate our machine learning model, the average recall value among 4-fold could be defined as follows:

$$Recall_{4-fold} = \frac{1}{4} \sum_{K=1}^4 Recall_{avg-K} \quad (14)$$

here  $Recall_{avg-K}$  is the macro-recall value of the  $K^{th}$  fold.

## Supplementary Tables

| Behavior       | Label | Description  |
|----------------|-------|--|
| Resting        | RE    | Stay still and motionless  |
| Walking        | W     | Walking around in the cage                                       |
| Rearing        | RA    | Standing on its hind legs and raise forelimbs from the ground    |
| Digging        | DI    | Using forelimbs to dig material put on the cage bottom           |
|                |       | Swings its front legs fast to throw the bedding particles around |
| Eating         | E     | Eating food placed in cage                                       |
| Grooming       | G     | Grooming itself to keep clean                                    |
| Drinking water | DW    | Drinking water placed in cage                                    |
| Scratching     | S     | Using hindlimb to scratch its body                               |

Table 1: Labels and descriptions for eight common mouse behaviors

| Time domain          | Frequency domain    |
|----------------------|---------------------|
| 30 features (6 axes) | 8 features (2 axes) |
| Maximum              | DC component        |
| Minimum              | FFT amplitude peak  |
| Variance             | FFT entropy         |
| Average              | FFT energy          |
| Root Mean Square     |                     |

Table 2: The calculated features in the recognition system.

# Supplementary Figures

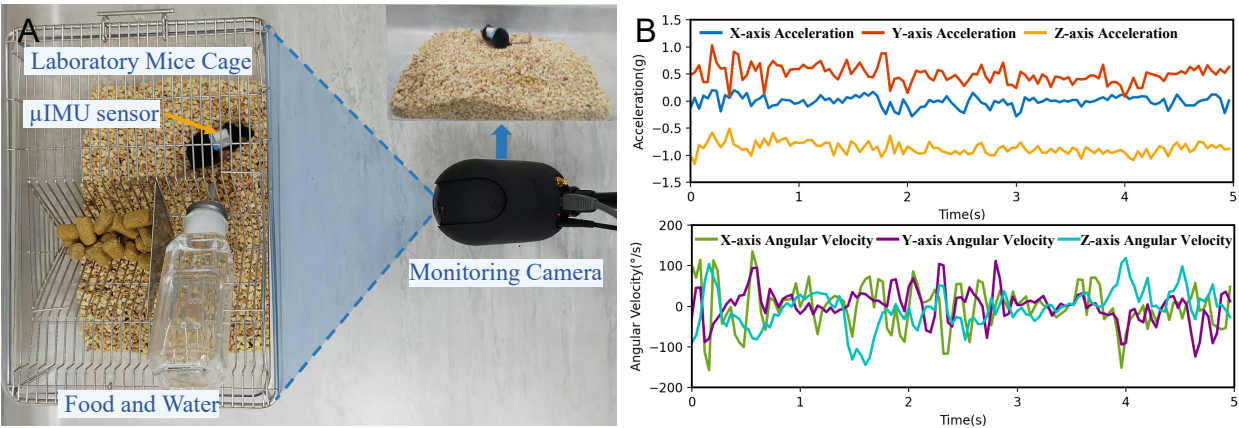


Figure 1: **Fig. S1.** Mouse tracking system hardware setup. (A) Experimental setup. (B) The collected acceleration and angular velocity signals

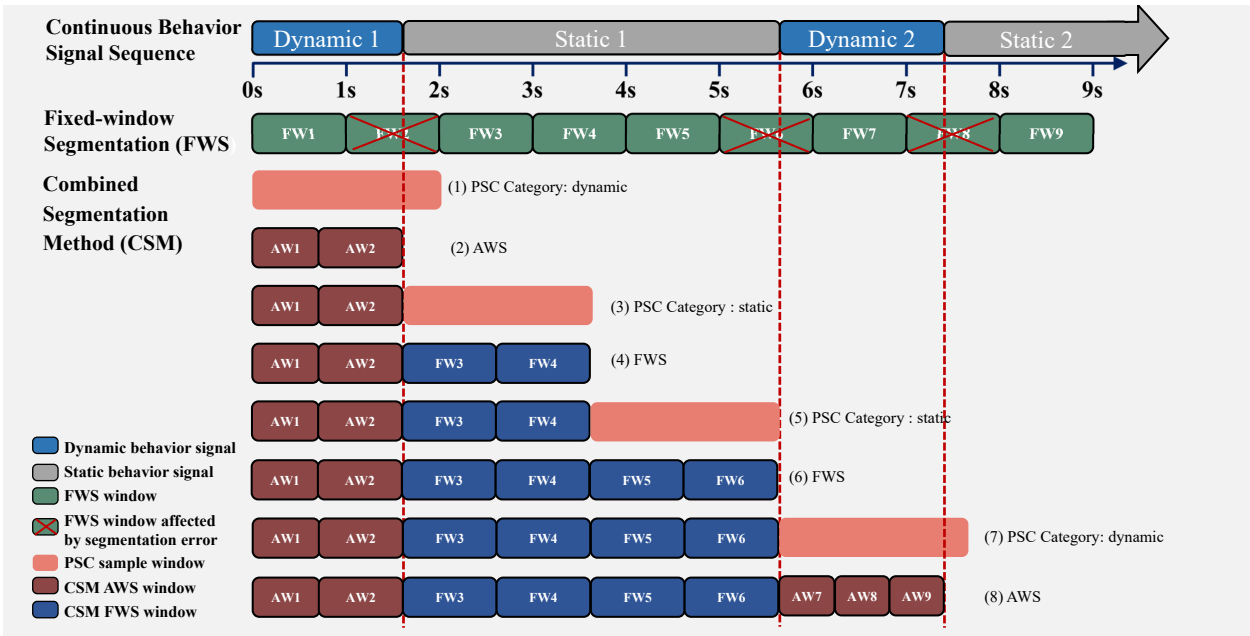


Figure 2: **Fig. S2.** The working principle of the Combined Segmentation Method (CSM)

# Supplementary Video

Rich media available at <https://drive.google.com/file/d/1p3rGwQ0969xrQChddNo0BshdFSEYHzeH/view?usp=sharing>

## References

Methods of Multivariate Analysis. (1998). *Journal of the American Statistical Association*, 93(443), 1239.  
<https://doi.org/10.2307/2669873>

Bayesian online changepoint detection. (2007). *ArXiv Preprint ArXiv:0710.3742*.

Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. (2020).  
*ArXiv Preprint ArXiv:2010.16061*.