IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 00.0000/ACCESS.2025.DOI

Sequential Unsupervised Motion Segmentation for Skill Evaluation and Comparison

KEN KAJIWARA[1], ISSEI SAITO[1], and TOMOAKI NAKAMURA[1], DAICHI MOCHIHASHI[2] and KOKI MIMURA[3]

[1]The University of Electro-Communications, Tokyo, Japan
[2]The Institute of Statistical Mathematics, Tokyo, Japan
[3]National Center of Neurology and Psychiatry, Tokyo, Japan

Corresponding author: Ken Kajiwara (e-mail: k_kajiwara@radish.ee.uec.ac.jp).

ABSTRACT In recent years, the evaluation of motion skills in fields such as dance and exercise has benefited from the proliferation of motion-capture technologies. This study proposes a computationally efficient method for segmenting and comparing time-series human motion data across individuals with varying skill levels. The method extends the Gaussian process hidden semi-Markov model (GP-HSMM) by incorporating random Fourier features (RFF), enabling motion segments to be modeled through linear regression and significantly reducing computational complexity. We also introduce sequential learning to support the incremental adaptation of the model to new individuals, which facilitates quantitative motion comparisons on a per-segment and per-joint basis. To preserve segmentation accuracy despite the approximation introduced by RFF, we optimize the kernel parameters using the Metropolis-Hastings algorithm within a Markov chain Monte Carlo (MCMC) framework. Experiments on publicly available exercise and dance datasets show that the proposed method achieves segmentation performance similar to GP-HSMM while reducing training time from several days to minutes. Moreover, the method supports the quantitative visualization of motion-skill differences, illustrated through a joint-level analysis of choreographic elements performed by dancers with different skill levels.

INDEX TERMS computational efficient motion analysis, Gaussian process, random Fourier features, time-series modeling, unsupervised segmentation, sequential learning, motion comparison, dance skill improvement, MCMC optimization.

I. INTRODUCTION

TRADITIONALLY, motion instruction in fields such as dance and sports has relied on the subjective experience and intuition of instructors. With increasing competitiveness in these fields, there is a growing need to enhance instruction efficiency. This can be achieved by automatically and quantitatively evaluating human motions using high-precision and continuous recordings of these complex motions. Recent advances in motion capture technology enable the acquisition of such data [1]–[6]. In this study, we propose a method that automatically and quantitatively compares and evaluates sequential motions of multiple individuals with varying skill levels on a unit-action basis. To achieve this, it is first necessary to extract common unit actions from sequential motion data and compare the differences in each individual's performance. The technique of automatically extracting unit actions from such data is known as segmentation. One powerful method for this task is the Gaussian process hidden semi-Markov model (GP-HSMM) [7].

GP-HSMM is a highly expressive method that enables accurate segmentation of continuous motions into unit actions. However, applying GP-HSMM to motion analysis presents two major challenges:

- The computational cost of the Gaussian process is high and requires significant training, which limits its suitability for high-dimensional or long-duration data.
- Because each unit action is modeled using a nonparametric probabilistic model—a Gaussian process—it remains difficult to quantitatively evaluate differences in motion across individuals.

To address these challenges, we propose a motion analysis method that extends the GP-HSMM. The method, termed random Fourier features GP-HSMM (RFF-GP-HSMM) [8], introduces basis functions derived from an approximation of the radial basis function (RBF) kernel using RFF [9]. In RFF-GP-HSMM, each unit action is modeled using a linear

regression framework based on these basis functions. This formulation replaces the computationally intensive Gaussian process with a simpler linear regression model, thereby reducing computational cost and addressing the first challenge. The second challenge is addressed by incorporating sequential learning [10]. Specifically, after training the model on the motion data of one participant, the motion data of a new participant is incorporated incrementally. Because the linear regression model represents each unit action with a low-dimensional set of parameters, it becomes possible to quantitatively compare the parameters before and after this incremental update. This enables a quantitative assessment of motion differences in each body part for the same unit action across different participants.

Although RFF offers substantial advantages, approximating the kernel function may degrade performance compared with a standard Gaussian process. To mitigate this, the parameters used for the basis function approximation are optimized using the Metropolis-Hastings(MH) algorithm [11].

In our experiments, we used publicly available motion-capture datasets for exercise and dance to evaluate the effectiveness of the method. On the exercise dataset, we demonstrate that by incorporating Markov chain Monte Carlo (MCMC) methods for parameter estimation of the RFF, the method achieves segmentation performance comparable to that of GP-HSMM. Furthermore, using a large-scale dance dataset, we show that the method enables faster analysis than conventional GP-HSMM and supports quantitative comparison and evaluation of motions performed by two dancers with different skill levels on a per-choreography basis.

The main contributions of this paper are as follows:

- Achieving significantly faster motion analysis compared to GP-HSMM.
- Enabling the comparison of unit-action differences through sequential learning in Bayesian linear regression.
- Maintaining high segmentation accuracy by optimizing basis function parameters using the MH algorithm.

II. RELATED WORK

In recent years, advances in motion-capture technology have spurred active research into the high-precision analysis of human motion [1]–[6]. For instance, Balazia and Sojka [1] used the Carnegie Mellon University (CMU) MoCap database to extract features from 3D time-series data from 31 joints, combining linear discriminant analysis (LDA) with the maximum margin criterion (MMC) to achieve higher accuracy than conventional methods in pedestrian identification.

Although optical motion capture systems provide high-precision motion data and are widely used, they suffer from inherent limitations such as high equipment cost, the need for a dedicated laboratory environment, and susceptibility to occlusion. These constraints may hinder the deployment of motion-capture-based algorithms in broader and more practical scenarios, such as daily training grounds or performance venues. Recently, various smart material-based actuation technologies, including liquid crystal elastomers [12], dielectric elastomer actuators [13], electromagnetic actuators [14], and piezoelectric actuators [15], [16], have been developed to support fast, flexible, and compact motion generation. Such advancements are expected to facilitate the integration of motion-capture and actuation systems, thereby extending the applicability of data-driven motion analysis to more practical environments.

Research targeting complex motions such as dance has also progressed. DanceMVP [6] introduced the ImperialDance dataset, which includes 69,300 seconds of dance motion with corresponding music, covering 5 genres, 20 music pieces, and 20 choreographies. This dataset was developed for a self-supervised learning framework that jointly handles multiple tasks, including vocabulary recognition, performance scoring, and rhythm alignment evaluation, aiming to capture skill differences through repeated motions by dancers with varying proficiency. However, the primary focus of this research is on overall motion scoring and classification. In classification tasks, manual annotation by experts is essential, which makes analyzing new data costly and time-consuming. Therefore, there is a clear need for analytical methods that do not require labeled data and can be processed automatically.

Furthermore, various methods have been proposed for motion skill evaluation [26]–[29]. However, many of these rely on supervised learning with labeled performance scores or external evaluation metrics. In contrast, the proposed method enables skill evaluation in a fully unsupervised manner without relying on any labeled data.

In response, unsupervised segmentation [17]–[22] has emerged as a promising approach, as it supports the automatic extraction of recurring patterns and facilitates the structural analysis of complex time-series data. Although it is important for real-world applications, unsupervised segmentation requires simultaneous estimation of segment boundaries and class labels. This leads to a combinatorial increase in possible configurations and, consequently, substantial computational cost. To mitigate this issue, many approaches incorporate heuristic assumptions. However, these heuristics often exhibit strong task dependency, which restricts their broader applicability. To overcome this limitation, GP-HSMM was introduced as a probabilistic generative model for segmentation [7]. It captures complex recurring patterns using Gaussian processes and represents their duration through a HSMM. This model leverages the flexibility of nonparametric modeling to achieve accurate segmentation of time-series data. GP-HSMM has since been applied across various domains. For example, Mimura et al. [23] integrated a hierarchical Dirichlet process into GP-HSMM to enable unsupervised extraction of characteristic behaviors in marmosets. In industrial contexts, it has been used to analyze worker behavior, revealing not only task structures but also individual traits and differences in skill levels [24]. In robotic systems, the model has been employed to segment continuous tasks into primitive actions for learning [25].

Although GP-HSMM is an effective, high-accuracy un-

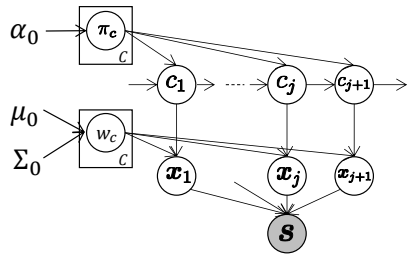


FIGURE 1. Graphical model of RFF-GP-HSMM.

supervised segmentation method, its use of Gaussian processes results in substantial computational overhead. As an alternative, RFF [9] was introduced to generate basis functions that approximate the kernel function. RFF has been incorporated into models such as the Gaussian process latent variable model (GPLVM) [30], [31] and the hidden Markov model (HMM) [32], demonstrating improved computational speed and scalability. Building on this, we previously proposed the RFF-GP-HSMM, which integrates RFF into the GP-HSMM framework [8]. However, the study [8] provided only a preliminary evaluation of computational efficiency and segmentation accuracy and did not examine motion characteristics. Furthermore, performance degradation caused by RFF approximation errors remained unresolved.

III. METHOD

A. GENERATIVE PROCESS

Fig. 1 shows a graphical model of the proposed RFF-GP-HSMM. This model assumes that time-series data are generated through the following process:

First, the transition probabilities π_c are generated from a Dirichlet distribution parameterized by α_0:

$$\pi_c \quad \sim \quad P(\pi|\alpha_0). \tag{1}$$

The j-th segment class c_j is generated using the $j-1$-th class c_{j-1} and the transition probability π_c:

$$c_j \quad \sim \quad P(c|c_{j-1},\pi_c). \tag{2}$$

In conventional GP-HSMM, each segment $\boldsymbol{x}_j$ is assumed to be generated by a Gaussian process. By contrast, RFF-GP-HSMM assumes that each segment is generated through linear regression. The regression weights $\boldsymbol{w}$ are generated from a Gaussian distribution with mean $\boldsymbol{\mu}_0$ and covariance matrix $\boldsymbol{\Sigma}_0$:

$$\boldsymbol{w}_c \quad \sim \quad \mathcal{N}(\boldsymbol{w}|\boldsymbol{\mu}_0,\boldsymbol{\Sigma}_0). \tag{3}$$

Segment $\boldsymbol{x}_j$, corresponding to class c_j, is generated using linear regression with weights $\boldsymbol{w}_{c_j}$:

$$\boldsymbol{x}_j \quad \sim \quad p(\boldsymbol{x}|\boldsymbol{w}_{c_j}). \tag{4}$$

The observed time-series data $\boldsymbol{S}$ are generated by concatenating the segments $\boldsymbol{x}_j$. However, simple linear regression alone cannot sufficiently represent complex time-series structure. To address this, RFF-GP-HSMM employs linear regression based on basis functions derived from RFF [9].

B. GAUSSIAN PROCESS

In conventional GP-HSMM, a Gaussian process models class-specific time-series variability and predicts an output value x' at time t'. A Gaussian process is a probabilistic model that marginalizes the regression weights $\boldsymbol{w}$ in the linear formulation $x' = \boldsymbol{w}^T\boldsymbol{\phi}(t')$, and applies the kernel trick, where $\boldsymbol{\phi}$ denotes the basis functions. Given a set of segments $\boldsymbol{X}_c$ assigned to class c and the corresponding set of time steps $\boldsymbol{t}_c$, the predictive distribution of the output x' at time t' is expressed as the following Gaussian distribution:

$$p(x'|t',\boldsymbol{X}_c,\boldsymbol{t}_c) = \mathcal{N}(\boldsymbol{k}^T\boldsymbol{K}^{-1}\boldsymbol{X}_c, k(\hat{t'},\hat{t'}) - \boldsymbol{k}^T\boldsymbol{K}^{-1}\boldsymbol{k}), \tag{5}$$

where $k(\cdot,\cdot)$ denotes the kernel, and $\boldsymbol{K}$ is a covariance matrix whose (p,q)-th element:

$$K(p,q) = k(t_p,t_q) + \beta^{-1}\delta_{pq}, \tag{6}$$

where t_p and t_q are the p-th and q-th elements of $\boldsymbol{t}_c$, respectively. β represents the noise variance, and δ_{pq} is the Kronecker delta function. $\boldsymbol{k}$ is a vector whose p-th element is $k(t_p,t')$. In this paper, we use the RBF kernel defined as follows:

$$k(t_p,t_q) = \exp(-\frac{1}{2}||t_p - t_q||^2), \tag{7}$$

The kernel enables flexible representation of nonlinear and high-dimensional input–output relationships by directly computing the inner product $k(t_p,t_q)$ $(= \boldsymbol{\phi}(t_p)^T\boldsymbol{\phi}(t_q))$, thereby avoiding explicit computation of the basis function $\boldsymbol{\phi}(\cdot)$. In Gaussian processes, computing (5) requires inverting the matrix $\boldsymbol{K}$. If the number of data points in $\boldsymbol{X}_c$ is denoted by N_c, then $\boldsymbol{K}$ is an $N_c \times N_c$ matrix. As the number of data points N_c increases, the size of $\boldsymbol{K}$ grows accordingly, making the matrix inversion increasingly computationally expensive. This creates a major computational bottleneck in conventional GP-HSMM.

C. GAUSSIAN PROCESS APPROXIMATION USING RFF

The primary computational bottleneck in GP-HSMM is the inversion of the kernel matrix $\boldsymbol{K}$, whose size increases with the number of data points N_c. However, if a basis function $\boldsymbol{\phi}$ satisfies $k(t_p,t_q) \approx \boldsymbol{\phi}(t_p)^T\boldsymbol{\phi}(t_q)$ for the RBF kernel, the linear regression model $x' = \boldsymbol{w}^T\boldsymbol{\phi}(t')$ can be applied without explicitly computing $\boldsymbol{K}$. RFF [9] provides a method for constructing such basis functions. For the RBF kernel, the basis functions are defined as:

$$\boldsymbol{\phi}(t) = \sqrt{\frac{2}{M}}\begin{bmatrix}\cos(\omega^{(1)}t + b^{(1)})\\ \cos(\omega^{(2)}t + b^{(2)})\\ \vdots \\ \cos(\omega^{(M)}t + b^{(M)})\end{bmatrix}, \tag{8}$$

where $\omega^{(m)} \sim p(\omega)$ and $b^{(m)} \sim \mathrm{Uniform}(0,2\pi)$. Using these basis functions enables Gaussian processes to be computed

through linear regression with significantly reduced computational cost, thereby eliminating the main bottleneck.

D. MCMC-OPTIMIZED RFF KERNEL

However, in RFF, the quality of the kernel approximation can vary substantially depending on the sampled values of $\omega^{(m)}$ and $b^{(m)}$. To improve approximation accuracy, we optimize these parameters so that the error between the true kernel function $k(t_p, t_q)$ in (7) and the approximated kernel function $\hat{k}(t_p, t_q)$ obtained from (8) is minimized. The approximated kernel function is defined as:

$$\hat{k}(t_p, t_q) = \frac{1}{M}\sum_{m=1}^{M}\sqrt{2}\cos(\omega^{(m)}t_p + b^{(m)}) \times \sqrt{2}\cos(\omega^{(m)}t_q + b^{(m)}). \tag{9}$$

The RBF kernel takes the value 1 when $t_p = t_q$ and decays smoothly toward 0 as $|t_p - t_q|$ increases, reflecting high correlation for nearby time points and near-independence for distant ones. Because the RBF kernel depends only on the time difference $|t_p - t_q|$, evaluating the approximation error at $t_q = 0$ is theoretically equivalent to evaluating the error over all pairs (t_p, t_q). Therefore, minimizing the error at $t_q = 0$ ensures overall kernel approximation accuracy for all time pairs under the assumption of stationarity. Based on this property, the error function is defined as follows:

$$E(\boldsymbol{\omega}, \boldsymbol{b}) = \sum_{p=1}^{N_c}\big(k(t_p, 0) - \hat{k}(t_p, 0)\big)^2, \tag{10}$$

$$\boldsymbol{\omega} = \{\omega^{(1)}, \omega^{(2)}, \ldots, \omega^{(M)}\}, \tag{11}$$

$$\boldsymbol{b} = \{b^{(1)}, b^{(2)}, \ldots, b^{(M)}\}. \tag{12}$$

Minimizing this error function can be interpreted as maximizing the probability of the following Gaussian distribution:

$$p(\boldsymbol{\omega}, \boldsymbol{b}) \propto \exp\big[-E(\boldsymbol{\omega}, \boldsymbol{b})\big], \tag{13}$$

$$= \prod_{p=1}^{N_c}\exp\Big[-\big(k(t_p, 0) - \hat{k}(t_p, 0)\big)^2\Big], \tag{14}$$

$$\propto \prod_{p=1}^{N_c}\frac{1}{\sqrt{2\pi}}\exp\Big[-\big(k(t_p, 0) - \hat{k}(t_p, 0)\big)^2\Big], \tag{15}$$

$$= \prod_{p=1}^{N_c}\mathcal{N}\big(k(t_p, 0)|\hat{k}(t_p, 0), 1\big). \tag{16}$$

We then estimate the parameters $\boldsymbol{\omega}$ and $\boldsymbol{b}$ that maximize this probability using the MH algorithm [11], an MCMC method in which the target distribution is defined by (16). For the proposal distribution, we use a multivariate normal distribution centered at the current values $(\boldsymbol{\omega}, \boldsymbol{b})$ to sample new candidates $(\boldsymbol{\omega}', \boldsymbol{b}')$:

$$\boldsymbol{\omega}' \sim \mathcal{N}(\boldsymbol{\omega}'|\boldsymbol{\omega}, \sigma_\omega \boldsymbol{I}), \tag{17}$$

$$\boldsymbol{b}' \sim \mathcal{N}(\boldsymbol{b}'|\boldsymbol{b}, \sigma_b \boldsymbol{I}). \tag{18}$$

Here, $\boldsymbol{I}$ is the identity matrix, and σ_ω and σ_b are variance parameters. The acceptance rate for a sampled point $(\boldsymbol{\omega}', \boldsymbol{b}')$ is given by:

$$A = \min\Big\{1, \frac{p(\boldsymbol{\omega}', \boldsymbol{b}')}{p(\boldsymbol{\omega}, \boldsymbol{b})}\Big\}. \tag{19}$$

By repeatedly proposing new samples using (17) and (18) and accepting or rejecting them according to (19), the algorithm efficiently explores regions of high probability in (16).

E. SEQUENTIAL LEARNING OF BAYESIAN LINEAR REGRESSION

Using RFF, we can explicitly obtain the basis function $\boldsymbol{\phi}(t) \in \mathbb{R}^M$ that approximates the RBF kernel. This enables the Gaussian process regression in (5) to be approximated using Bayesian linear regression with this basis function, resulting in a substantial reduction in computational cost. Furthermore, Bayesian linear regression supports sequential learning—previously infeasible with GP-HSMM— and allows for quantitative comparison of parameter changes before and after learning.

Consider a case where the r-th dataset $\boldsymbol{D}^{(r)}$ $(r = 1, 2, \ldots)$ is segmented by RFF-GP-HSMM, yielding segments $\boldsymbol{X}_c^{(r)}$ and corresponding time steps $\boldsymbol{t}_c^{(r)}$ for class c. In this setting, the predictive distribution of the output x' at time t' is given by:

$$p\big(x'|t', \boldsymbol{X}_c^{(r)}, \boldsymbol{t}_c^{(r)}\big) = \mathcal{N}\big(x' \mid \mu^{(r)}(t'), \sigma^{2\,(r)}(t')\big), \tag{20}$$

$$\mu_c^{(r)}(t') = \big(\boldsymbol{m}_c^{(r)}\big)^\top \boldsymbol{\phi}(t'), \tag{21}$$

$$\sigma_c^{2\,(r)}(t') = \beta^{-1} + \boldsymbol{\phi}(t')^\top \boldsymbol{S}_c^{(r)} \boldsymbol{\phi}(t'), \tag{22}$$

where $\boldsymbol{S}_c^{(r)}$ and $\boldsymbol{m}_c^{(r)}$ are the parameters of the posterior distribution of the weights $\boldsymbol{w}$, computed as:

$$\boldsymbol{S}_c^{(r)} = \Big\{\big(\boldsymbol{S}_c^{(r-1)}\big)^{-1} + \beta\sum_{p=1}^{N_c}\boldsymbol{\phi}(t_p^{(r)})\boldsymbol{\phi}(t_p^{(r)})^\top\Big\}^{-1}, \tag{23}$$

$$\boldsymbol{m}_c^{(r)} = \boldsymbol{S}_c^{(r)}\Big\{\big(\boldsymbol{S}_c^{(r-1)}\big)^{-1}\boldsymbol{m}_c^{(r-1)} + \beta\sum_{p=1}^{N_c}x_p^{(r)}\boldsymbol{\phi}(t_p^{(r)})\Big\}, \tag{24}$$

Here, $x_p^{(r)}$ and $t_p^{(r)}$ are the p-th elements of $\boldsymbol{X}_c^{(r)}$ and $\boldsymbol{t}_c^{(r)}$, respectively. The prior is defined by $\boldsymbol{S}^{(r-1)}$ and $\boldsymbol{m}^{(r-1)}$, which are obtained from the previous dataset $\boldsymbol{D}^{(r-1)}$. When $r = 1$, because no prior information is available, we set $\boldsymbol{m}^{(0)} = \boldsymbol{0}$ and $\boldsymbol{S}^{(0)} = \alpha^{-1}\boldsymbol{I}$.

In the predictive distribution of Gaussian process regression (5), one must compute the inverse of the $N_c \times N_c$ kernel matrix $\boldsymbol{K}$. In contrast, when using RFF, the predictive distribution can be obtained by inverting the $M \times M$ matrix in (23). Here, N_c is the number of data points in class c, and the cost of computing $\boldsymbol{K}^{-1}$ increases with data size. However, M is a fixed value chosen in advance; therefore, the matrix inversion in RFF-based linear regression can be computed in constant time, independent of the number of data points. Moreover, unlike Gaussian processes, which do not explicitly represent segment-characterizing parameters, our linear

regression model represents these characteristics with the M-dimensional mean parameter vector $\boldsymbol{m}^{(r)}$. Thus, by comparing $\boldsymbol{m}^{(r-1)}$ and $\boldsymbol{m}^{(r)}$, it becomes possible to quantify differences in segments between the $(r-1)$-th and r-th datasets.

F. PARAMETER INFERENCE

1) Blocked Gibbs sampler

We infer the segments and their corresponding classes in each observation sequence using a blocked Gibbs sampler. The inference procedure is summarized as follows:

1) Randomly segment all observation sequences $\boldsymbol{s}_1, \boldsymbol{s}_2, \ldots, \boldsymbol{s}_n, \ldots$, and initialize all model parameters.
2) For the n-th observation sequence $\boldsymbol{s}_n$, remove its segments $\boldsymbol{x}_{nj}$ from their previously assigned classes and update the linear regression parameters $(\boldsymbol{\Sigma}_c, \boldsymbol{m}_c)$ and the transition probabilities $P(c|c')$ for each class.
3) Sample new segments $\boldsymbol{x}_{nj}$ $(j = 1, \ldots, J)$ and the corresponding class labels c_{nj} $(j = 1, \ldots, J)$ using the forward filtering-backward sampling procedure described below. Then update the linear regression parameters $\boldsymbol{\Sigma}_c, \boldsymbol{m}_c$ and transition probabilities $P(c|c')$.
4) Repeat steps 2 and 3 for all sequences until sampler converges.

Using this iterative procedure, the parameters of the RFF-GP-HSMM are optimized, and the observation sequences are segmented.

2) Forward filtering-backward sampling

We use forward filtering-backward sampling (FFBS) to sample the segments and their corresponding class labels. FFBS is a dynamic programming algorithm that efficiently evaluates the probabilities of all valid segmentation–classification configurations and then draws samples according to these probabilities.

In the forward filtering step, we compute the forward probability α, which represents the probability that a subsequence of length k ending at time t in the observation sequence $\boldsymbol{s}_n$ forms a segment assigned to class c:

$$\alpha[t][k][c] = \left(\prod_{t'=t-k}^{t} p(\boldsymbol{s}_{n,t'} | t', \boldsymbol{X}_c, \boldsymbol{t}_c) \right) p(k|\lambda) \times \sum_{k'=K_{\min}}^{K_{\max}} \sum_{c'=1}^{C} p(c|c') \alpha[t-k][k'][c'], \quad (25)$$

where $\boldsymbol{s}_{n,t'}$ denotes the data point at time step t' in sequence $\boldsymbol{s}_n$. C is the number of segment classes, and K_{min} and K_{max} are the minimum and maximum allowable segment lengths, respectively. The term $p(k|\lambda)$ represents the probability that a segment has length k, modeled using a Poisson distribution with mean segment length λ:

$$P(k|\lambda) = \frac{\lambda^k e^{-\lambda}}{k!}. \quad (26)$$

Algorithm 1 Forward filtering-backward sampling.

```
// Forward filtering
for t = 1 to T do
    for k = 1 to K do
        for c = 1 to C do
            Compute α[t][k][c]
        end for
    end for
end for

// Backward sampling
t = T, j = 1
while t > 0 do
    k, c ~ p(c_{j+1} | c) α[t][k][c]
    x_j = s_{t-k:t}
    c_j = c
    t = t - k
    j = j + 1
end while
return (x_{J_n}, x_{J_n-1}, ..., x_1), (c_{J_n}, c_{J_n-1}, ..., c_1)
```

TABLE 1. Computer used in experiments.

CPU	Apple M4
Memory	16GB

The term $p(c|c')$ represents the transition probability between classes and is given by the following Dirichlet multinomial model:

$$p(c|c') = \frac{N_{c'c} + \alpha}{N_{c'} + C\alpha}, \quad (27)$$

where $N_{c'}$ is the number of segments assigned to class c', and $N_{c'c}$ is the number of transitions from class c' to class c. In (25), k' and c' denote the length and class of the segment preceding $\boldsymbol{s}_{t-k:t}$. Because these values are not known during the forward pass, they are marginalized out over all possible segment lengths and classes.

Next, backward sampling is performed based on the computed forward probabilities. In this step, the class and segment lengths are sampled in reverse temporal order, starting from the end of the time-series data ($t = T$), according to the forward probabilities.

$$k, c \sim p(c_{j-1}|c)\alpha[t][k][c]. \quad (28)$$

This procedure determines the final segments and their corresponding classes in each observation sequence. The complete algorithm is summarized in Algorithm 1.

IV. EXPERIMENTS

We performed a segmentation accuracy evaluation using an exercise dataset and conducted a motion analysis using a dance dataset. The specifications of the computer used for both experiments are shown in Table 1.

A. EXPERIMENT 1: SEGMENTATION ACCURACY ON THE EXERCISE DATASET

In this experiment, we quantitatively evaluated the segmentation performance of the proposed method using labeled motion data.

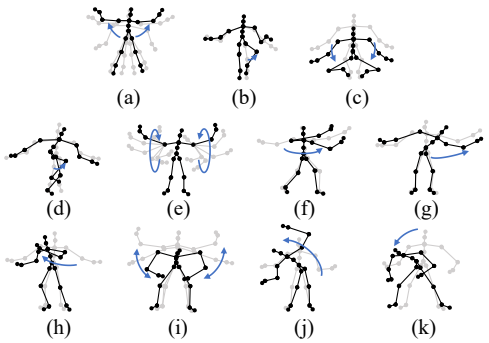


FIGURE 2. Motions included in the exercise dataset: (a) jumping jack, (b) jogging, (c) squatting, (d) knee raise, (e) arm circle, (f) twist, (g) side reach, (h) boxing, (i) arm wave, (j) side bend, and (k) toe touch.

1) Experimental setup

We used three sequences from Subject 14 in the CMU Motion Capture dataset[1]. These sequences include various activities such as running and jumping, as shown in Fig. 2. The data were downsampled to 4 fps. We used eight-dimensional coordinate data comprising the 2D positions of the left hand, right hand, left foot and right foot, and normalized the data to the range $[-1, 1]$ using min–max normalization.

In this experiment, we compared three methods: the conventional GP-HSMM, a basic RFF-GP-HSMM without parameter optimization, and an RFF-GP-HSMM with parameter optimization via MCMC. For both RFF-GP-HSMM models, the number of random features was set to $M = 20$. The parameter λ was selected empirically [2]. For the exercise data, visual inspection of the motion videos indicated that one unit motion typically spans about 20 frames. Therefore, for all methods, the segment length parameters to $\lambda = 20$, $K_{\min} = 15$, and $K_{\max} = 30$. The blocked Gibbs sampler was run for 5 iterations.

Because all methods are sensitive to initialization, each method was executed 10 times with different initial values. We computed the average normalized Hamming distance (NHD) between the segmentation results and the ground-truth labels. The NHD ranged from $[0, 1]$, where values closer to 0 indicate higher consistency between the segmentation results and ground truth.

2) Results

The mean and standard deviation of the NHD across 10 trials for each method are shown in Table 2. The basic RFF-GP-HSMM showed slightly lower accuracy than the conventional GP-HSMM. In contrast, the RFF-GP-HSMM with MCMC improved the approximation accuracy of the kernel function, resulting in a lower mean NHD and smaller variance, thereby achieving segmentation performance close to that of the conventional GP-HSMM.

[1] http://mocap.cs.cmu.edu

[2] We acknowledge that this parameter requires tuning for each dataset; however, we observed that model performance is not overly sensitive to variations in this parameter [33].

TABLE 2. Normalized Hamming distance (mean ± SD) from 10 trials for each method.

Method	NHD (Mean ± SD)
GP-HSMM	0.384 ± 0.0589
RFF-GP-HSMM	0.445 ± 0.0759
RFF-GP-HSMM w/ MCMC	0.408 ± 0.0427

B. EXPERIMENT 2: EFFICIENT ANALYSIS AND COMPARATIVE EVALUATION ON THE DANCE DATASET

This experiment demonstrates that sequential learning with RFF-GP-HSMM enables the analysis of motion differences between expert and intermediate dancers.

1) Experimental setup

We used the dance dataset introduced in DanceMVP [6], presented at AAAI 2024.[3] This dataset contains motion data from participants at several skill levels performing a 10-second dance from the song "What is Love?" by TWICE. According to Zhong et al., participants were classified according to their skill levels by two professional dance evaluators invited from a dance organization. The classification was based on evaluation criteria established by their affiliated institutions, which included the following aspects:

- speed and timing of movements
- range of motion
- continuity and smoothness of movements
- synchronization with music

The data consisted of 63-dimensional coordinate values representing the 3D positions (x, y, z) of 21 joints. To obtain coordinates relative to the torso, we subtracted the position of the chest joint from the coordinates of all other joints. This converted the 63-dimensional data into 60-dimensional data centered on the chest. The dancers in the dataset all faced the same front direction, so orientation normalization or rotational alignment was unnecessary and was not performed.

The dataset used in this experiment consisted of motion data from two participants— one expert and one intermediate dancer—each performing the dance 20 times. As each trial contained 1,200 frames and each frame consisted of 60-dimensional coordinate data, the total dataset size was 2,880,000 time-series data points. We compared the computation time of the RFF-GP-HSMM with MCMC-based parameter optimization against that of the conventional GP-HSMM. In our RFF-GP-HSMM, the number of random features was set to $M = 20$. For the dance dataset, we visually examined the motion capture videos and found that one choreography sequence spans roughly 120 frames[IV-A1]. Therefore, for both methods, we set the segment length parameters to $\lambda = 120$, $K_{\min} = 100$, and $K_{\max} = 140$. The number of iterations for the blocked Gibbs sampler was 5.

Furthermore, we analyzed the motions of the expert and intermediate dancers using RFF-GP-HSMM. As each segmented motion in the dance sequence can be interpreted

[3] https://github.com/YunZhongNikki/ImperialDance-Dataset

TABLE 3. Comparison of training times.

Trial	GP-HSMM (s)	RFF-MCMC-GP-HSMM (s)
1	5.25×10^5	3.44×10^2
2	5.34×10^5	3.57×10^2
3	5.21×10^5	3.46×10^2
4	5.17×10^5	3.46×10^2
5	5.57×10^5	3.51×10^2
Mean	$\mathbf{5.31 \times 10^5}$	$\mathbf{3.49 \times 10^2}$

as a unit of choreography, we refer to each segment as a choreographic element. First, the expert dancer's data $\boldsymbol{D}^{(1)}$ were segmented, and the posterior distribution parameters $\boldsymbol{S}_c^{(1)}$ and $\boldsymbol{m}_c^{(1)}$ for each choreographic element were computed using Equations (23) and (24). For this initial segmentation, we set the prior distribution parameters to $\boldsymbol{m}^{(0)} = 0$ and $\boldsymbol{S}^{(0)} = \alpha^{-1}\boldsymbol{I}$, where the prior precision parameter was fixed to $\alpha = 1.0$. Next, using $\boldsymbol{S}_c^{(1)}$ and $\boldsymbol{m}_c^{(1)}$ as the prior distribution parameters, the intermediate dancer's data $\boldsymbol{D}_c^{(2)}$ were segmented, and the posterior distribution parameters $\boldsymbol{S}^{(2)}$ and $\boldsymbol{m}^{(2)}$ for each choreographic element were computed following Equations (23) and (24). The resulting vectors $\boldsymbol{m}^{(1)}$ and $\boldsymbol{m}^{(2)}$ represent the mean weights of the motion characteristics for the expert and intermediate dancers, respectively. Using these vectors, we quantitatively visualized the skill differences for each choreographic element on a per-joint basis.

2) Computation time

For each method, we conducted five trials and measured the training time for each. The results are shown in Table 3. A comparison of the average training times shows that the conventional GP-HSMM required approximately 148 hours (about 6 days) to complete training, whereas the proposed method completed it in approximately 6 minutes, demonstrating its practicality for large-scale analysis.

3) Motion analysis based on segmentation

Through segmentation with the proposed method, the dance motions were automatically classified into six choreographic elements. An example of these six extracted choreographic elements is shown in Fig. 3. For these six choreographic elements, we selected three body parts for comparison: the right forearm, which corresponded to the choreography with the largest distance between the parameters of the expert and intermediate dancers, left knee and the hip. In dance, knee and hip movements are known to be particularly important. The knee is essential for maintaining rhythm through actions such as "up" and "down" motions, whereas the hip is critical for enhancing expressiveness through movements such as "hip swings," which require coordination with the trunk and hip joints. Because these movements demand a high degree of physical control, we hypothesized that differences in skill level would be clearly reflected in the accuracy and smoothness of these motions.

For each choreographic element c, to compare the motions

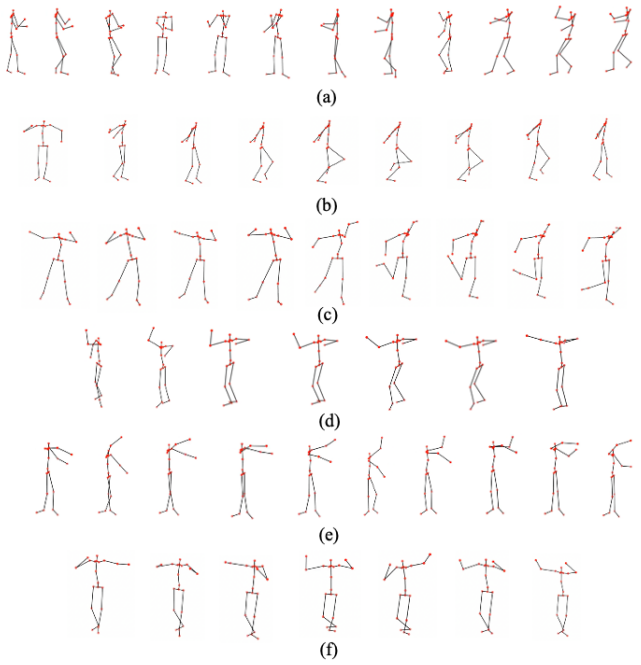


FIGURE 3. Choreographies: (a) Choreography 1: Swing both arms to sides; (b) Choreography 2: Raise and lower left leg while bending knee; (c) Choreography 3: Swing right arm back and forth while lifting right heel; (d) Choreography 4: Extend left arm forward; (e) Choreography 6: Swing arm to left side; and (f) Choreography 7: Stretch both arms to sides.

of the expert and intermediate dancers for each choreographic element c, we normalized the 60-dimensional weight vectors $\boldsymbol{m}_c^{(1)}$ and $\boldsymbol{m}_c^{(2)}$ for each joint to have a unit L2 norm and reduced them to two dimensions using principal component analysis (PCA). In Figs. 5–6, the expert's parameters are shown as blue circles and the intermediate dancer's as orange crosses, plotted separately for each joint. The numbers indicate the class index c of each choreographic element.

For each joint and each choreography, we quantitatively calculated the distances between the parameters of the expert and intermediate dancers, which represent the skill differences. For each joint, we focused on the two choreographies with the largest distances between the dancers' parameters. To enable a relative comparison of motion differences across choreographic elements and body joints within the dancer, we additionally computed a rank-based percentile that indicates the proportion of all distance values that are equal to or larger than a given distance (i.e., the percentage position when ranked in descending order).

As shown in Fig. 4, larger distances between the expert and intermediate dancers were observed for choreographic elements 5 and 7, suggesting that their motions differ. Element 5 exhibited the largest distance, whereas element 7 ranked second and fell within the upper 0.83% of the distance distribution. The segmented motions of the right forearm for elements 5 and 7 are shown in Fig. 7 (a) and (b), respectively. All of these choreographic elements involved movements of the right forearm. In these figures, the upper row represents the expert dancer's motion and the lower row represents the intermediate dancer's motion in temporal order. The analyzed

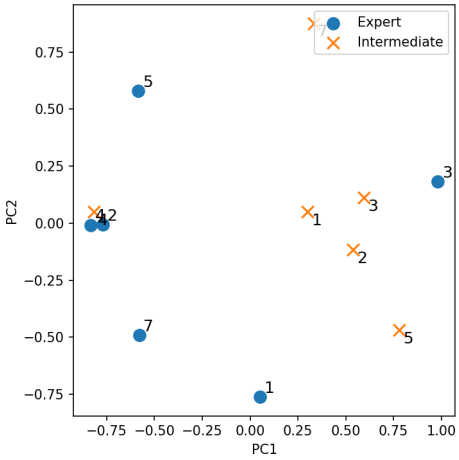


FIGURE 4. Reduced parameters of right forearm motion. Circles indicate the expert dancer and crosses indicate the intermediate dancer. Each number represents the index of the choreography.

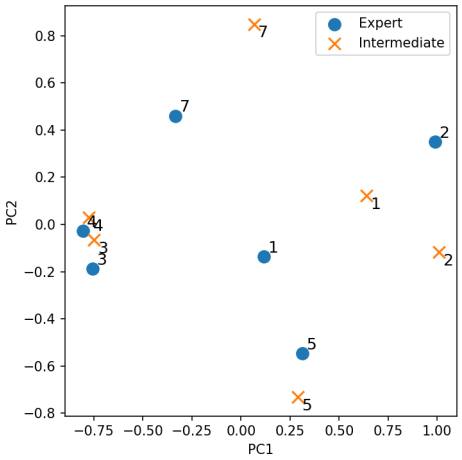


FIGURE 5. Reduced parameters of left knee motion. Circles indicate the expert dancer and crosses indicate the intermediate dancer. Each number represents the index of the choreography.

joint is indicated by a blue circle for the expert dancer and an orange circle for the intermediate dancer. In Fig. 7 (a), which shows the right forearm motion in choreographic element 5, the expert swings the right forearm widely, whereas the intermediate dancer exhibits only a small swing. In Fig. 7 (b), which shows the motion in element 7, the expert moves the right forearm while swinging the arm sideways, whereas the intermediate dancer keeps the forearm fixed while swinging the arm. These observations indicate that the intermediate dancer is unable to utilize the right forearm as effectively as the expert.

As shown in Fig. 5, a larger difference in parameter distributions between the expert and intermediate dancers was observed for choreographic elements 1 and 7, suggesting that their motions differ. Elements 1 and 7 were positioned within the top 43.3% and 47.5% of the distance distribution, respectively. The segmented motions of the left knee for elements 1 and 7 are shown in Fig. 8 (a) and (b), respectively. All of these choreographic elements involved movements of

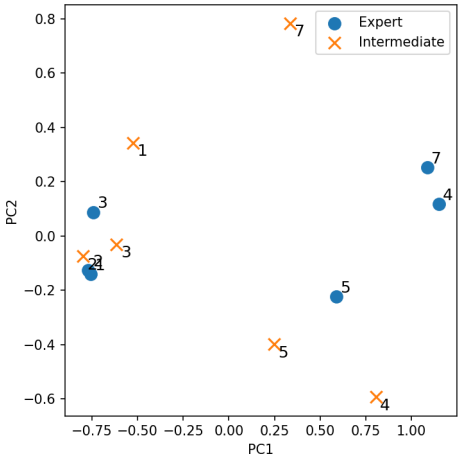


FIGURE 6. Reduced parameters of hip motion. Circles indicate the expert dancer and crosses indicate the intermediate dancer. Each number represents the index of the choreography.

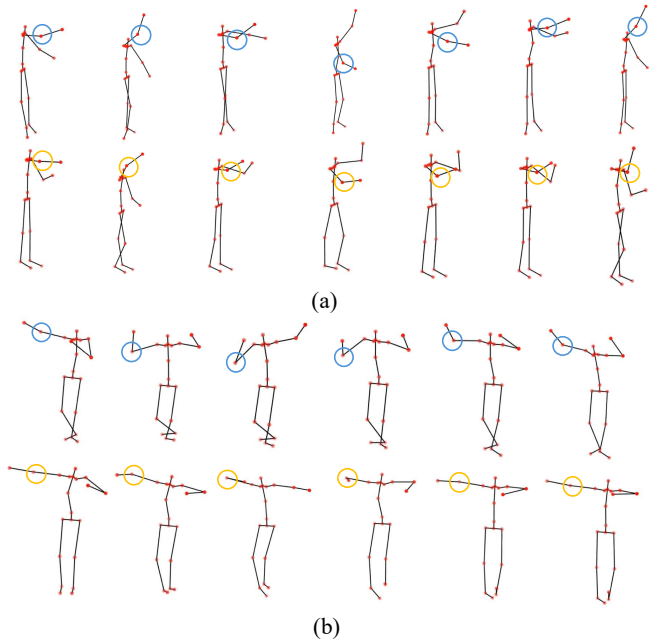


FIGURE 7. (a) Visualized movement of the right forearm in Choreography 5, segmented for detailed comparison. (b) Visualized movement of the right forearm in Choreography 7, segmented for detailed comparison.

the knee. Comparing the left knee motions in (a) and (b), the expert clearly bends the left knee, whereas the intermediate dancer shows less flexion, indicating limited use of the left knee compared to the expert.

As shown in Fig. 6, a larger difference in the parameters between the expert and intermediate dancers was observed for choreographic elements 4 and 7, suggesting that their motions differ. Elements 4 and 7 fell within the top 30.0% and 21.7% of the distance distribution, indicating relatively large expert–intermediate differences. The segmented motions of the hip for elements 4 and 7 are shown in Fig. 9 (a) and (b), respectively. Both of these choreographic elements involved hip movements. In Fig. 9 (a), which shows the hip

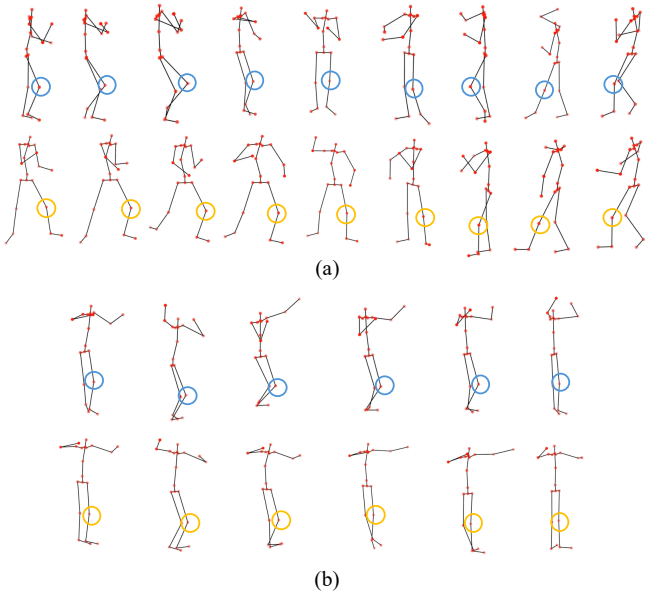


FIGURE 8. **(a) Visualized movement of the left knee in Choreography 1, segmented for detailed comparison. (b) Visualized movement of the left knee in Choreography 7, segmented for detailed comparison.**

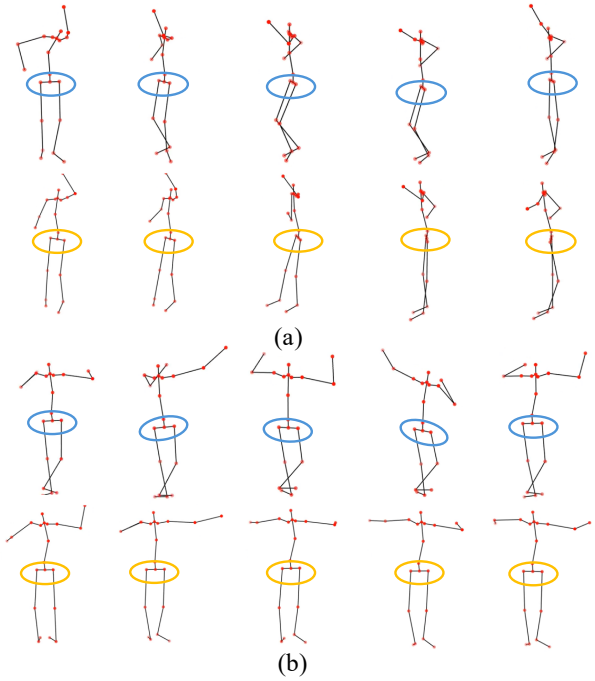


FIGURE 9. **(a) Visualized movement of the hip in Choreography 4, segmented for detailed comparison. (b) Visualized movement of the hip in Choreography 7, segmented for detailed comparison.**

motion in choreographic element 4, the expert dancer moves the hip vertically and front to back in coordination with the motion of the knee, whereas the intermediate dancer shows no observable hip movement. In Fig. 9 (b), which shows the motion in element 7, the expert dancer swings the hip to the side, whereas the intermediate dancer again exhibits no hip movement. These observations indicate that the intermediate dancer is unable to utilize the hip as effectively as the expert.

These results demonstrate that the proposed method, by incorporating sequential learning, enables the quantitative comparison of motions from multiple participants with different skill levels on a per-joint and per-choreography basis.

V. CONCLUSION

This study aimed to develop a system for quantitative and visual comparison of motions from multiple individuals with different skill levels at the level of individual action units in domains such as dance and exercise. To this end, we developed a framework by extending GP-HSMM. The method employs RFF for kernel approximation and optimizes parameters via MCMC, achieving a substantial reduction in computation time while maintaining segmentation accuracy comparable with that of conventional GP-HSMM. In addition, by incorporating sequential learning, the framework enables the model trained on an expert's motion to be updated with data from new participants, allowing for flexible modeling of individual motion characteristics.

Experiments on public motion-capture datasets for exercise and dance demonstrated the effectiveness of stabilizing kernel approximation via MCMC and adaptive model updating via sequential learning. The results also showed that the method can quantitatively evaluate joint-level motion differences in unit actions among participants with varying skill levels.

Although the framework enables segment-wise, joint-level comparison of motions using a computationally efficient extension of GP-HSMM, several limitations remain. First, the motion-comparison experiment was conducted using only two dancers—one expert and one intermediate participant. Although each dancer performed the dance 20 times and the learned parameter vectors therefore reflect stable, participant-specific averages rather than frame-level noise, the current dataset does not allow statistically generalizable conclusions.

Furthermore, a limitation of this study is that it only demonstrated the ability to identify which joints exhibit motion differences, without analyzing the specific nature of those differences. In our RFF-based linear regression model, motions are represented as weighted sums of cosine waves with varying frequencies and phases. This implies that further analysis of the learned regression weights could reveal the specific components of the motion that differ. Because the main focus of this paper is the sequential learning framework for skill-level comparison, we did not include detailed analyses of weight-vector interpretation. Nevertheless, such analysis is important. We plan to investigate in future work how the PCA-projected directions of the learned weight vectors relate to kinematic properties such as range of motion and flexion, thereby clarifying the specific motion components that contribute to skill differences.

While the RFF-based linear model provides substantial computational advantages, its representational power inherently depends on M; smaller values may limit the ability to capture high-frequency or non-stationary motion patterns. We therefore regard a systematic investigation of the relationship

among M, model performance, and approximation limits as an important direction for future work.

Building on this research, we plan to develop a system based on the method for applications in sports and rehabilitation, where the assessment of motion quality and skill comparison is essential. Such a system would also enable personalized feedback based on comparisons with expert motion.

ACKNOWLEDGMENT

This research was supported by AMED under Grant Number JP24wm0625124. The authors acknowledge the use of the CMU Motion Capture Database and the ImperialDance dataset in this study.

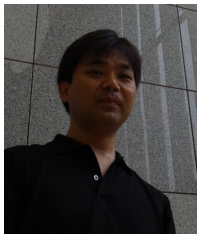
REFERENCES

[1] M. Balazia and P. Sojka, "Walker-independent gait recognition with CNN," in *Proc. 25th IEEE Int. Conf. Image Process. (ICIP)*, 2018, pp. 4163–4167.
[2] F. von Marcard, B. Rosenhahn, M. J. Black, and G. Pons-Moll, "3D people in the wild: From videos to accurate 3D human pose and shape," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2018, pp. 2670–2679.
[3] F. M. Thoker, H. Doughty, and C. G. M. Snoek, "Skeleton Contrastive 3D Action Representation Learning," in *Proc. 29th ACM Int. Conf. Multimedia (MM '21)*, Oct. 20–24, 2021, pp. 1655–1663.
[4] W. T. Lam, Y. M. Tang, and K. N. K. Fong, "A systematic review of the applications of markerless motion capture technology for clinical measurement in rehabilitation," *J. NeuroEng. Rehabil.*, vol. 20, no. 1, p. 1, 2023.
[5] X. Suo, W. Tang, and Z. Li, "Motion capture technology in sports scenarios: A survey," *Sensors*, vol. 24, no. 9, art. no. 2947, 2024.
[6] Y. Zhong and Y. Demiris, "DanceMVP: Multi-task dance performance assessment via text prompting," in *Proc. AAAI Conf. Artif. Intell.*, vol. 38, no. 9, 2024, pp. 10270–10278.
[7] T. Nakamura, T. Nagai, D. Mochihashi, I. Kobayashi, H. Asoh, and M. Kaneko, "Segmenting continuous motions with hidden semi-Markov models and Gaussian processes," *Front. Neurorobot.*, vol. 11, art. no. 67, 2017.
[8] I. Saito, M. Nagano, T. Nakamura, D. Mochihashi, and K. Mimura, "Scalable unsupervised segmentation via random Fourier feature-based Gaussian process," *arXiv preprint arXiv:2507.10632*, 2025.
[9] A. Rahimi and B. Recht, "Random features for large-scale kernel machines," in *Adv. Neur. Info. Process. Syst.*, J. C. Platt, D. Koller, Y. Singer, and S. Roweis, Eds., vol. 20., 2007.
[10] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer, 2006.
[11] W. K. Hastings, "Monte Carlo sampling methods using Markov chains and their applications," *Biometrika*, vol. 57, no. 1, pp. 97–109, 1970.
[12] M.-Y. Choi, K. Kim, K. Kim, S.-K. Ahn, and J.-H. Na, "Rapid and spatially programmed electrostatic actuation of anisotropic polymers," *Chem. Eng. J.*, vol. 471, Art. no. 146237, 2023. [Online]. Available: https://doi.org/10.1016/j.cej.2023.146237
[13] J. Prechtl, M. Baltes, K. Flaßkamp, and G. Rizzello, "Sensorless proprioception in multi-DoF dielectric elastomer soft robots via system-level self-sensing," *IEEE/ASME Trans. Mechatron.*, vol. 29, no. 6, pp. 4365–4376, Dec. 2024. [Online]. Available: https://doi.org/10.1109/TMECH.2024.3375923
[14] Y. Wang, X. Hu, L. Cui, X. Xiao, K. Yang, Y. Zhu, and H. Jin, "Bioinspired handheld time-share driven robot with expandable DoFs," *Nat. Commun.*, vol. 15, no. 1, Art. no. 768, Jan. 2024. [Online]. Available: https://doi.org/10.1038/s41467-024-44993-x
[15] Y. Wang and L. Wang, "Design, fabricate, and experimental verification of an ultrasonic linear motor derived from V-type motors," *Rev. Sci. Instrum.*, vol. 91, Art. no. 045002, Apr. 2020. [Online]. Available: https://doi.org/10.1063/1.5129586
[16] Y. Wang, S. Ma, J. Deng, S. Zhang, W. Chen, and Y. Liu, "Focus-switchable piezoelectric actuator: A bionic thin-plate design inspired by conch structure," *Sens. Actuators A Phys.*, vol. 379, Art. no. 115921, 2024. [Online]. Available: https://doi.org/10.1016/j.sna.2024.115921
[17] E. B. Fox, E. B. Sudderth, M. I. Jordan, and A. S. Willsky, "Joint modeling of multiple related time series via the beta process," *arXiv preprint arXiv:1112.1414*, 2011.
[18] Y. Matsubara, Y. Sakurai, and C. Faloutsos, "Autoplait: Automatic mining of co-evolving time sequences," in *Proc. ACM SIGMOD Int. Conf. Manag. Data*, 2014, pp. 193–204.
[19] F. Sener and A. Yao, "Unsupervised learning and segmentation of complex activities from video," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2018, pp. 8368–8376.
[20] P. Bojanowski, R. Lajugie, F. Bach, I. Laptev, J. Ponce, C. Schmid, and J. Sivic, "Weakly supervised action labeling in videos under ordering constraints," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2014, pp. 628–643.
[21] D.-A. Huang, L. Fei-Fei, and J. C. Niebles, "Connectionist temporal modeling for weakly supervised action labeling," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2016, pp. 137–153.
[22] A. Richard, H. Kuehne, and J. Gall, "Weakly supervised action learning with RNN based fine-to-coarse modeling," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2017, pp. 754–763.
[23] K. Mimura, J. Matsumoto, D. Mochihashi, T. Nakamura, H. Nishijo, M. Higuchi, T. Hirabayashi, and T. Minamimoto, "Unsupervised decomposition of natural monkey behavior into a sequence of motion motifs," *Commun. Biol.*, vol. 7, no. 1, art. no. 1080, 2024.
[24] I. Saito, T. Nakamura, T. Hatta, W. Fujita, S. Watanabe, and S. Miwa, "Unsupervised work behavior analysis using hierarchical probabilistic segmentation," in *Proc. 49th Annu. Conf. IEEE Ind. Electron. Soc. (IECON)*, 2023, pp. 1–6.
[25] Y. Mo, H. Sasaki, T. Matsubara, and K. Yamazaki, "Multi-step motion learning by combining learning-from-demonstration and policy-search," *Adv. Robot.*, vol. 37, no. 9, pp. 560–575, 2023. [Online]. Available: https://doi.org/10.1080/01691864.2022.2163187
[26] Y. Sharma, V. Bettadapura, T. Plötz, N. Hammerla, S. Mellor, R. McNaney, P. Olivier, S. Deshmukh, A. McCaskie, and I. Essa, "Video based assessment of OSATS using sequential motion textures," in *Georgia Institute of Technology*, 2014.
[27] G. Bertasius, H. Soo Park, S. X. Yu, and J. Shi, "Am I a baller? Basketball performance assessment from first-person videos," in *Proc. IEEE Int. Conf. Comput. Vis.*, pp. 2177–2185, 2017.
[28] H. Doughty, W. Mayol-Cuevas, and D. Damen, "The pros and cons: Rank-aware temporal attention for skill determination in long videos," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pp. 7862–7871, 2019.
[29] Y. Tang, Z. Ni, J. Zhou, D. Zhang, J. Lu, Y. Wu, and J. Zhou, "Uncertainty-aware score distribution learning for action quality assessment," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pp. 9839–9848, 2020.
[30] M. M. Zhang, G. W. Gundersen, and B. E. Engelhardt, "Bayesian non-linear latent variable modeling via random Fourier features," *arXiv preprint arXiv:2306.08352*, 2023.
[31] Y. Li, Z. Lin, F. Yin, and M. M. Zhang, "Preventing model collapse in Gaussian process latent variable models," *arXiv preprint arXiv:2404.01697*, 2024.
[32] Y. Jung and J. Park, "Scalable hybrid HMM with Gaussian process emission for sequential time-series data clustering," *arXiv preprint arXiv:2001.01917*, 2020.
[33] M. Nagano, T. Nakamura, T. Nagai, D. Mochihashi, and I. Kobayashi, "Spatio-temporal categorization for first-person-view videos using a convolutional variational autoencoder and Gaussian processes," Frontiers in Robotics and AI, vol. 9, Frontiers Media S.A., Sep. 2022.

KEN KAJIWARA received his Bachelor's degree from the University of Electro-Communications in 2025. He is currently enrolled in a Master's program at the Graduate School of Informatics and Engineering at the University of Electro-Communications. His research interests include intelligent robotics and machine learning.

ISSEI SAITO received his Bachelor's degree from the University of Electro-Communications in 2023. He received his Master's degree from the University of Electro-Communications in 2025. He is currently in a Doctor 's course at the Graduate School of Informatics and Engineering, the University of Electro-Communications. His research interests include intelligent robotics and machine learning.

TOMOAKI NAKAMURA received his BE, ME, and Dr. of Eng. degrees from the University of Electro-Communications in 2007, 2009, and 2011. From April 2011 to March 2012, He was a research fellow of the Japan Society for the Promotion of Science. In 2013, he worked for Honda Research Institute Japan Co., Ltd. From April 2014 to March 2018, he was an Assistant Professor at the Department of Mechanical Engineering and Intelligent Systems, the University of Electro-Communications. Since April 2019, he has been an Associate Professor at the same department. His research interests are intelligent robotics and machine learning. He has received the IROS Best Paper Award Finalist, the Advanced Robotics Best Paper Award, and the JSAI Best Paper Award.



DAICHI MOCHIHASHI obtained BS from University of Tokyo and PhD from Nara Institute of Science and Technology in 1998 and 2005, respectively. He is currently a professor at The Institute of Statistical Mathematics, and jointly at National Institute for Japanese Language and Linguistics, Japan. His research interest includes natural language processing, machine learning, and Bayesian statistics.

KOKI MIMURA received his BS and PhD from Tokyo University of Agriculture and Technology in 2008 and 2013, respectively. He is currently a section chief at the National Center of Neurology and Psychiatry (NCNP), Japan. His research interests include primate neuroscience, computational modeling of behavior, and the pathophysiology of autism spectrum disorders.