

# Estimating 3D Trajectories from 2D Projections via Disjunctive Factored Four-Way Conditional Restricted Boltzmann Machines

Decebal Constantin Mocanu<sup>a,\*</sup>, Haitham Bou Ammar<sup>b</sup>, Luis Puig<sup>c</sup>, Eric Eaton<sup>d</sup>,  
Antonio Liotta<sup>a</sup>

<sup>a</sup>*Department of Electrical Engineering, Eindhoven University of Technology, Eindhoven, the Netherlands*

<sup>b</sup>*Department of Computer Science, American University of Beirut, Beirut, Lebanon*

<sup>c</sup>*Department of Computer Science & Engineering, University of Washington, Seattle, USA.*

<sup>d</sup>*Department of Computer and Information Science, University of Pennsylvania, Philadelphia, USA.*

---

## Abstract

Estimation, recognition, and near-future prediction of 3D trajectories based on their two dimensional projections available from one camera source is an exceptionally difficult problem due to uncertainty in the trajectories and environment, high dimensionality of the specific trajectory states, lack of enough labeled data and so on. In this article, we propose a solution to solve this problem based on a novel deep learning model dubbed *disjunctive factored four-way conditional restricted Boltzmann machine* (DFFW-CRBM). Our method improves state-of-the-art deep learning techniques for high dimensional time-series modeling by introducing a novel tensor factorization capable of driving forth order Boltzmann machines to considerably lower energy levels, at no computational costs. DFFW-CRBMs are capable of accurately estimating, recognizing, and performing near-future prediction of three-dimensional trajectories from their 2D projections while requiring limited amount of labeled data. We evaluate our method on both simulated and real-world data, showing its effectiveness in predicting and classifying complex ball trajectories and human activities.

*Keywords:* Deep learning, restricted Boltzmann machines, 3D trajectories estimation, activity recognition.

---

\*Corresponding author

Email address: d.c.mocanu@tue.nl (Decebal Constantin Mocanu)

## 1. Introduction

Estimating and predicting trajectories in three-dimensional spaces based on two-dimensional projections available from *one* camera source is an open problem with wide-ranging applicability including entertainment [1], medicine [2], biology [3], physics [4], etc. Unfortunately, solving this problem is exceptionally difficult due to a variety of challenges, such as the variability of states of the trajectories, partial occlusions due to self articulation and layering of objects in the scene, and the loss of 3D information resulting from observing trajectories through 2D planar image projections. A variety of techniques have considered variants of this problem by incorporating additional sensors, e.g., cameras [5], sonar radars [6], which provide new data for geometric solvers allowing for accurate estimation and prediction. Though compelling, the success of these methods arrives at increased costs (e.g., incorporating new sensors) and computational complexities (e.g., handling more inputs geometrically).

The problem above, however, can be framed as a time-series estimation and prediction one, for which numerous machine learning algorithms can be applied. An emerging trend in machine learning for computer vision and pattern recognition is deep learning (DL) which has been successfully applied in a variety of fields, e.g., multi-class classification [7], collaborative filtering [8], image quality assessment [9], reinforcement learning [10], transfer learning [11], information retrieval [12], depth estimation [13], face recognition [14], and activity recognition [15]. Most related to this work are *temporal-based* deep learners, e.g., [16, 17], which we briefly review next. Extending on standard restricted Boltzmann machines (RBMs)[18], temporal RBMs (TRBMs) consider a succession of RBMs, one for each time frame, allowing them to perform accurate prediction and estimation of time-series. Due to their complexity, such naive extensions require high computational effort before acquiring acceptable behavior. Conditional RBMs (CRBMs) remedy this problem by proposing an alternative extension of RBMs [19]. Here, the architecture consists of two separate visible layers, representing history (i.e., values from previous time frames), and current values, and a hidden layer for latent correlation discovery. Though successful, CRBMs are only capable of modeling time series data with relatively “smooth” variations and

similarly with other state-of-the-art neural network architectures for time series, e.g. recurrent neural networks, they can not learn within the same model different types of time-series. Thus, to model different types of non-linear time variations within the same model, the authors in [19] extend CRBMs by allowing for a three-way weight tensor connection among the different layers. Computational complexity is then reduced by adapting a factored version (i.e., FCRBMs) of the weight tensor, which leads to a construction exhibiting accurate modeling and prediction results in a variety of experiments, including human motion styles [20]. However, these methods fail to perform both classification and regression in one unified framework. Recently, Factored Four-Way Conditional Restricted Boltzmann Machines (FFW-CRBMs) have been proposed [21]. These extend FCRBMs by incorporating a label layer and a four-way weight tensor connection among the layers to modulate the weights for capturing subtle temporal differences. This construction allowed FFW-CRBMs to perform both, i.e. classification and real-valued predictions, within the same model, and to outperform state-of-the-art specialized methods for classification or prediction [21].

**Contributions:** In this paper we, first, propose the use of FFW-CRBMs to estimate 3D trajectories from their 2D projections, while at the same time being also capable to classify those trajectories. Though successful, we discovered that FFW-CRBMs require substantial amount of *labeled data* before achieving acceptable performance when predicting three-dimensional trajectories from two-dimensional projections. As FFW-CRBMs require *three-dimensional labeled* information for accurate predictions which is not typically available, secondly, in this paper, we remedy these problems by proposing an extension of FFW-CRBMs, dubbed Disjunctive FFW-CRBMs (DFFW-CRBMs). Our extension refines the factoring of the four-way weight tensor connecting the machine layers to settings where labeled data is scarce. Adopting such a factorization “specializes” FFW-CRBMs and ensures lower energy levels (approximately three times less energy on the overall dataset). This yields the sufficiency of a reduced training dataset for DFFW-CRBMs to reach similar classification performance to state-of-the-art methods and to at least double the performance on real-valued predictions. Importantly, such accuracy improvements come at the same computational cost of  $\mathcal{O}(n^2)$  compared to FFW-CRBMs. Precisely, our machine requires limited la-

beled data (less than 10 % of the overall dataset) for: *i*) simultaneously classifying and predicting three-dimensional trajectories based on their two-dimensional projections, and *ii*) accurately estimating three-dimensional postures up to an arbitrary number of  
65 time-steps in the future.

We have extensively tested DFFW-CRBMs on both, simulated and real-world data, to show that they are capable of outperforming state-of-the-art methods in real-valued predictions and classifications. In the first set of experiments we evaluate its performance by predicting and classifying simulated three-dimensional ball trajectories  
70 (based on a real-world physics simulator) thrown from different initial spins. Given these successes, in the second set of experiments we predict and classify high-dimensional human poses and activities (up-to 32 human skeleton joints in 2D and 3D coordinates systems, corresponding to 160 dimensions) using real-world data showing that DFFW-CRBMs acquire double accuracy results at reduced labeled data sizes.

## 75 **2. Background**

This section provides relevant background knowledge essential to the remainder of the paper. Firstly, restricted Boltzmann machines (RBMs), being at the basis of our proposed method, are surveyed. Secondly, Contrastive Divergence, a training algorithm for Deep Learning methods, is presented. The section concludes with a brief  
80 description of deep-learning based models for time series prediction and classification.

### *2.1. Restricted Boltzmann Machines*

Restricted Boltzmann machines (RBMs) [18] are energy-based models for unsupervised learning. They use a generative model of the distribution of training data for  
85 prediction [22]. These models employ stochastic nodes and layers, making them less vulnerable to local minima [20]. Further, due to their stochastic neural configurations, RBMs possess excellent generalization and density estimation capabilities [23, 24].

Formally, an RBM consists of visible and hidden binary layers connected by an undirected bipartite graph. More exactly, the visible layer  $\mathbf{v} = [v_1, \dots, v_{n_v}]$  collects all

visible units  $v_i$  and represents the real-data, while the hidden layer  $\mathbf{h} = [h_1, \dots, h_{n_h}]$  representing all the hidden units  $h_j$  increases the learning capability by enlarging the class of distributions that can be represented to an arbitrary complexity.  $n_v$  and  $n_h$  are the number of neurons in the visible and hidden layers, respectively.  $W_{ij}$  denotes the weight connection between the  $i^{th}$  visible and  $j^{th}$  hidden unit, and  $v_i$  and  $h_j$  denote the state of the  $i^{th}$  visible and  $j^{th}$  hidden unit, respectively. The matrix of all weights between the layers is given by  $\mathbf{W} \in \mathbb{R}^{n_h \times n_v}$ . The energy function of RBMs is given by

$$E(v, h) = - \sum_{i=1}^{n_v} \sum_{j=1}^{n_h} W_{ij} v_i h_j - \sum_{i=1}^{n_v} a_i v_i - \sum_{j=1}^{n_h} b_j h_j \quad (1)$$

where,  $a_i$  and  $b_j$  represent the biases of the visible and hidden layers, respectively. The joint probability of a visible and hidden configuration can be written as  $P(v, h) = \frac{\exp(-E(v, h))}{Z}$  with  $Z = \sum_{x, y} \exp(-E(x, y))$ . The marginal distribution,  $p(v) = \sum_h p(v, h)$ , can be used to determine the probability of a data point represented by a state  $v$ .

## 2.2. Training an RBM via Contrastive Divergence

The RBMs parameters are trained by maximizing the likelihood function, typically by following the gradient of the energy function. Unfortunately, in RBMs, maximum likelihood estimation can not be applied directly due to intractability problems. These problems can be circumvented by using Contrastive Divergence (CD) [25] to train the RBM. In CD, learning follows the gradient of:

$$CD_n \propto D_{KL}(p_0(\mathbf{x}) || p_\infty(\mathbf{x})) - D_{KL}(p_n(\mathbf{x}) || p_\infty(\mathbf{x})) \quad (2)$$

where,  $p_n(\cdot)$  is the distribution of a Markov chain running for  $n$  steps and  $D_{KL}$  symbolizes the Kullback-Leibler divergence [26]. To find the update rules for the free parameters of the RBM (i.e weights and biases), the RBM's energy function from Equation 1 has to be differentiated with respect to those parameters. Thus, in  $CD_n$  the weight updates are done as follows:  $w_{ij}^{\tau+1} = w_{ij}^\tau + \alpha (\langle \langle h_j v_i \rangle_{p(\mathbf{h}|\mathbf{v}; \mathbf{W})} \rangle_0 - \langle h_j v_i \rangle_n)$  where  $\tau$  is the iteration number,  $\alpha$  is the learning rate,  $\langle \langle h_j v_i \rangle_{p(\mathbf{h}|\mathbf{v}; \mathbf{W})} \rangle_0 = \frac{1}{N_I} \sum_{k=1}^{N_I} v_i^{(q)} P(h_j^{(q)} = 1 | \mathbf{v}^{(q)}; \mathbf{W})$  and  $\langle h_j v_i \rangle_n = \frac{1}{N_I} \sum_{k=1}^{N_I} v_i^{(q)(n)} P(h_j^{(q)(n)} = 1 | \mathbf{v}^{(q)(n)}; \mathbf{W})$  where  $N_I$  is the

total number of input instances, and the superscript  $(q)$  shows the  $q^{th}$  input instance. The superscript  $(n)$  indicates that the states are obtained after  $n$  steps of Gibbs sampling on a Markov chain which starts at the original data distribution  $p_0(\cdot)$ . In practice, learning can be performed using just one step Gibbs sampling, which is carried in four sub-steps: (1) initialize visible units, (2) infer all the hidden units, (3) infer all the visible units, and (4) update the weights and the biases.

### 2.3. Factored Conditional Restricted Boltzmann Machine

Conditional Restricted Boltzmann Machines (CRBM) [20] are an extension of RBMs used to model time series data, for example, human activities. They use an undirected model with binary hidden variables connected to real-valued visible ones. At each time step  $t$ , the hidden and visible nodes receive a connection from the visible variables at the last  $L$  time-steps. The history of the real-world values until time  $t$  is collected in the real-valued history vector  $\mathbf{v}_{<t}$  with  $n_{v_{<t}} = n_v(L - 1)$  being the number of elements in  $\mathbf{v}_{<t}$ . The total energy of the CRBM is given by:

$$E = \sum_{i=1}^{n_v} \frac{(\hat{a}_{i,t} - v_{i,t})^2}{2\sigma_i^2} - \sum_{j=1}^{n_h} \hat{b}_{j,t} h_{j,t} - \sum_{i=1}^{n_v} \sum_{j=1}^{n_h} W_{ij} \frac{v_{i,t}}{\sigma_i} h_{j,t} \quad (3)$$

where  $\hat{a}_{i,t} = a_i + \sum_{k=1}^{n_{v_{<t}}} A_{ki} v_{k,<t}$  and  $\hat{b}_{j,t} = b_j + \sum_{k=1}^{n_{v_{<t}}} B_{kj} v_{k,<t}$  represent the “dynamic biases”, with  $k$  being the index of the elements from  $\mathbf{v}_{<t}$ .

Taylor and Hinton introduced the Factored Condition Restricted Boltzmann Machine (FCRBM) [20], which permits the modeling of different styles of time series within the same model, due to the introduction of multiplicative, three-way interactions and of a *preset* style label,  $\mathbf{y}_t$ . To reduce the computational complexity of this model, they factored the third order tensors between layer in products of matrices. Formally, FCRBM defines a joint probability distribution over the visible  $\mathbf{v}_t$  and hidden  $\mathbf{h}_t$  neurons. The joint distribution is conditioned on the past  $L$  observations,  $\mathbf{v}_{<t}$ , model parameters,  $\Theta$ , and the *preset* style label,  $\mathbf{y}_t$ . Interested readers are referred to [20] for a more comprehensive discussion on CRBMs and FCRBMs.

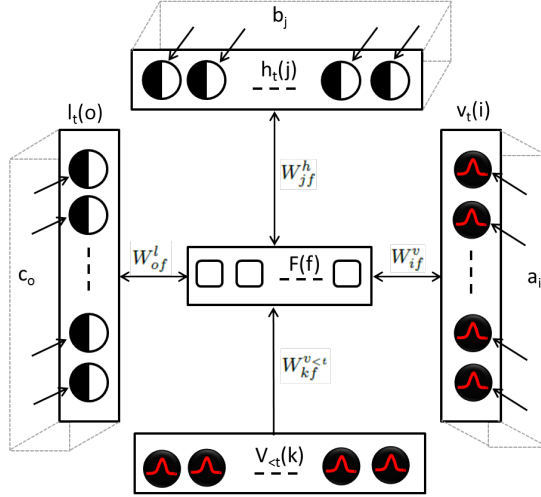


Figure 1: A high level depiction of the FFW-CRBM showing the four layer configuration and the factored weight tensor connection among them. Gaussian nodes shown on the history and visible layers represent real-valued inputs, while sigmoidal nodes on the hidden and label layers demonstrate binary values.

#### 2.4. Four-Way Conditional Restricted Boltzmann Machines

120 Due to the limitations exhibited by FCRBMs, e.g., the impossibility of performing classification without extensions, we proposed the four-way conditional restricted Boltzmann machines (FW-CRBMs) for performing prediction and classification in one unified framework [21]. FW-CRBMs introduced an additional layer and a four-way multiplicative weight tensor interaction between neurons. Please note that, later on,  
 125 other four-way models have been proposed but they can perform just classification and no prediction [27].

FW-CRBMs extended FCRBMs to include a label layer  $\mathbf{I}_t$  and a fourth order weight tensor connection  $\mathbf{W}_{ijk\ell} \in \mathbb{R}^{n_v \times n_h \times n_{v<t} \times n_l}$ , where  $n_v$ ,  $n_h$ ,  $n_{v<t}$ ,  $n_l$  represent the number of neurons from the present, hidden, history and label layers, respectively. Though successful, FW-CRBMs exhibited high computational complexities (i.e.,  $\mathcal{O}(n^4)$ ) for tuning free parameters. Circumventing these problems, we factored the weight tensor into sums of products leading to more efficient machines (i.e.,  $\mathcal{O}(n^2)$ ) labeled as factored four-way conditional restricted Boltzmann machines (FFW-CRBMs). FFW-

CRBMs, shown in Figure 1, minimize the following energy functional

$$\begin{aligned}
 \mathbf{E}(\mathbf{v}_t, \mathbf{h}_t, \mathbf{l}_t | \mathbf{v}_{<t}, \Theta) = & \quad (4) \\
 & - \sum_{i=1}^{n_v} \frac{(v_{i,t} - a_i)^2}{\sigma_i^2} - \sum_{j=1}^{n_h} h_{j,t} b_j - \sum_{o=1}^{n_l} l_{o,t} c_o \\
 & - \sum_{f=1}^{n_F} \sum_{i=1}^{n_v} W_{if}^v \frac{v_{i,t}}{\sigma_i} \sum_{j=1}^{n_h} W_{jf}^h h_{j,t} \sum_{k=1}^{n_{v<t}} W_{kf}^{v<t} \frac{v_{k,<t}}{\sigma_k} \sum_{o=1}^{n_l} W_{of}^l l_{o,t} .
 \end{aligned}$$

where  $n_F$  is number of factors and  $i, j, k$ , and  $o$  are the indices of the visible layer neurons  $\mathbf{v}_t$ , the hidden layer neurons  $\mathbf{h}_t$ , the history layer neurons  $\mathbf{v}_{<t}$  and the labeled layer neurons  $\mathbf{l}_t$  respectively.  $\mathbf{W}^v, \mathbf{W}^h, \mathbf{W}^l$  symbolize the bidirectional and symmetric weights from the visible, hidden and label layers to the factors, respectively, while  $\mathbf{W}^{v<t}$  represents the directed weights from the history layer to the factors. As in the case of the three-way models [28], standard CD is unsuccessful in training also the four-way models, due to the need of predicting two output layers (i.e. label and present layers). Thus, in [21] we proposed a sequential variant of CD, named sequential Markov chain contrastive divergence, more suitable for tuning the free parameters in FW-CRBMs.

FFW-CRBMs have shown good generalization and time series latent feature learning capabilities compared to state-of-the-art techniques including but not limited to, support vector machines, CRBMs, and FCRBMs [21]. It is for these reasons that we believe that FFW-CRBMs can serve as a basis for predicting three-dimensional trajectories from two-dimensional projections. Unfortunately, FFW-CRBMs are not readily applicable to such a problem as they require substantial amount of labeled data for successful tuning. In this paper, we extend FFW-CRBMs to Disjunctive FFW-CRBMs (DFFW-CRBMs) by proposing a novel factoring process essential for predicting and classifying 3D trajectories from 2D projections. Our model, detailed next, reduces sample complexities of current methods and allows for lower energy levels compared to FFW-CRBMs leading to improved performance.



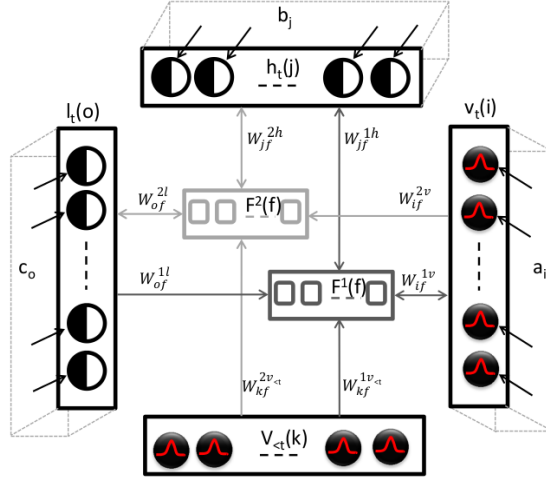


Figure 2: A high level depiction of DFFW-CRBMs showing the four layer configuration and the refined tensors factoring for increased accuracy and efficiency.

### 3. Disjunctive Factored Four Way Conditional restricted Boltzmann Machines

This section details disjunctive factored four way conditional restricted Boltzmann machines (DFFW-CRBMs), shown in Figure 2. Similarly to FFW-CRBMs, our model consists of four layers to represent visible, history, hidden, and label units. Contrary to the factoring adopted by FFW-CRBMs, however, our model incorporates two new factoring layers. The first, i.e.,  $F^1(f)$  in the figure, is responsible for specializing the machine to real-valued predictions through  $\mathbf{W}^{1l}$ ,  $\mathbf{W}^{1v}$ ,  $\mathbf{W}^{1h}$ , and  $\mathbf{W}^{1v<t}$ , while the second,  $F^2(f)$ , specializes the machine to classification through the corresponding weight tensor collections. Such a specialization is responsible for reducing sample complexities needed by DFFW-CRBMs for successful parameter tuning as demonstrated in Section 4, while the computational complexity of DFFW-CRBM remains the same as for FFW-CRBM (i.e.,  $\mathcal{O}(n^2)$ ). Given our novel construction, DFFW-CRBMs require their own special mathematical treatment. Next, we detail each of the energy functional and learning rules needed by DFFW-CRBMs.

### 3.1. DFFW-CRBM's Energy Function

The energy function of DFFW-CRBMs consists of three major terms. The first, i.e.,  $-\sum_{i=1}^{n_v} \frac{(v_{i,t} - a_i)^2}{\sigma_i^2} - \sum_{j=1}^{n_h} h_{j,t} b_j - \sum_{o=1}^{n_l} l_{o,t} c_o$  corresponds to the standard energy representing a specific submachine of DFFW-CRBMs (i.e. the energy given by the neurons of each layers and their biases) while the second two denote energies related to the first and second factoring layers, respectively:

$$\begin{aligned}
\mathbf{E}(\mathbf{v}_t, \mathbf{h}_t, \mathbf{l}_t | \mathbf{v}_{<t}, \Theta) = & \quad (5) \\
& - \underbrace{\sum_{i=1}^{n_v} \frac{(v_{i,t} - a_i)^2}{\sigma_i^2} - \sum_{j=1}^{n_h} h_{j,t} b_j - \sum_{o=1}^{n_l} l_{o,t} c_o}_{\text{standard three-layer energy}} \\
& - \underbrace{\sum_{f=1}^{n_{F1}} \sum_{i=1}^{n_v} W_{if}^{1v} \frac{v_{i,t}}{\sigma_i} \sum_{j=1}^{n_h} W_{jf}^{1h} h_{j,t} \sum_{k=1}^{n_{v<t}} W_{kf}^{1v<t} \frac{v_{k,<t}}{\sigma_k} \sum_{o=1}^{n_l} W_{of}^{1l} l_{o,t}}_{\text{first factoring layer}} \\
& - \underbrace{\sum_{f=1}^{n_{F2}} \sum_{i=1}^{n_v} W_{if}^{2v} \frac{v_{i,t}}{\sigma_i} \sum_{j=1}^{n_h} W_{jf}^{2h} h_{j,t} \sum_{k=1}^{n_{v<t}} W_{kf}^{2v<t} \frac{v_{k,<t}}{\sigma_k} \sum_{o=1}^{n_l} W_{of}^{2l} l_{o,t}}_{\text{second factoring layer}}.
\end{aligned}$$

Here,  $n_{F1}$  denotes the total number of factors for the weight tensor collection specializing DFFW-CRBMs to regression, while  $n_{F2}$  is total the number of factors responsible  
165 for classification.  $i, j, k,$  and  $o$  represent the indices of the visible layer neurons  $\mathbf{v}_t$ , the hidden layer neurons  $\mathbf{h}_t$ , the history layer neurons  $\mathbf{v}_{<t}$  and the labeled layer neurons  $\mathbf{l}_t$ , respectively. Furthermore,  $\mathbf{W}^{1v}$  and  $\mathbf{W}^{1h}$  represent the bidirectional and symmetric weight connections from the visible and hidden layers to the factors, while  $\mathbf{W}^{1l}$  and  $\mathbf{W}^{1v<t}$  denote the *directed* weights from the label and history layers to the factors  
170 Similarly,  $\mathbf{W}^{2l}$  and  $\mathbf{W}^{2h}$  represent the bidirectional and symmetric weights from the label and hidden layers to the factors, while  $\mathbf{W}^{2v}$  and  $\mathbf{W}^{2v<t}$  denote the directed weights from the visible and history layers to the factors. Finally, the two groups of four weight matrices each noted with  $\mathbf{W}^{1\cdot}$  and  $\mathbf{W}^{2\cdot}$  belong to the factorized tensor specialization in regression and classification, respectively.

Inference for DFFW-CRBM corresponds to determining values of the activation probabilities for each of the units. As shown in Figure 2, units within the same layer do not share connections. This allows for parallel probability computation for all units within the same layer. The overall input of each hidden  $s_{j,t}^h$ , visible  $s_{i,t}^v$ , and labelled  $s_{o,t}^l$  unit is given by:

$$\begin{aligned}
s_{j,t}^h &= \sum_{f=1}^{n_{F1}} W_{jf}^{1h} \sum_{i=1}^{n_v} W_{if}^{1v} \frac{v_{i,t}}{\sigma_i} \sum_{k=1}^{n_{nv}<t} W_{kf}^{1v}<t} \frac{v_{k,<t}}{\sigma_k} \sum_{o=1}^{n_l} W_{of}^{1l} l_{o,t} \\
&+ \sum_{f=1}^{n_{F2}} W_{jf}^{2h} \sum_{i=1}^{n_v} W_{if}^{2v} \frac{v_{i,t}}{\sigma_i} \sum_{k=1}^{n_{nv}<t} W_{kf}^{2v}<t} \frac{v_{k,<t}}{\sigma_k} \sum_{o=1}^{n_l} W_{of}^{2l} l_{o,t} \\
s_{i,t}^v &= \sum_{f=1}^{n_{F1}} W_{if}^{1v} \sum_{j=1}^{n_h} W_{jf}^{1h} h_{j,t} \sum_{k=1}^{n_{nv}<t} W_{kf}^{1v}<t} \frac{v_{k,<t}}{\sigma_k} \sum_{o=1}^{n_l} W_{of}^{1l} l_{o,t} \\
s_{o,t}^l &= \sum_{f=1}^{n_{F2}} W_{of}^{2l} \sum_{j=1}^{n_h} W_{jf}^{2h} h_{j,t} \sum_{k=1}^{n_{nv}<t} W_{kf}^{2v}<t} \frac{v_{k,<t}}{\sigma_k} \sum_{i=1}^{n_v} W_{if}^{2v} \frac{v_{i,t}}{\sigma_i} . \quad (6)
\end{aligned}$$

Consequently, for each of the  $j^{th}$  hidden,  $i^{th}$  visible, and  $o^{th}$  labelled units, the activation probabilities can be determined as

$$\begin{aligned}
p(h_{j,t} = 1 | \mathbf{v}_t, \mathbf{v}_{<t}, \mathbf{l}_t) &= \frac{1}{1 + e^{-(b_j + s_{j,t}^h)}} \quad (7) \\
p(v_{i,t} = x | \mathbf{h}_t, \mathbf{v}_{<t}, \mathbf{l}_t) &= \mathcal{N}(a_i + s_{i,t}^v, \sigma_i^2) \\
p(l_{o,t} = 1 | \mathbf{v}_t, \mathbf{v}_{<t}, \mathbf{h}_t) &= \frac{1}{1 + e^{-(c_o + s_{o,t}^l)}} ,
\end{aligned}$$

where  $\mathcal{N}(\cdot)$  represents the standard Gaussian distribution.

### 3.3. Parameter Tuning: Update Rules & Algorithm

#### 3.3.1. Update Rules

Generally, parameters,  $\Theta$ , are updated according to:

$$\Theta_{\tau+1} = \Theta_{\tau} + \underbrace{\rho \tilde{\Theta}_{\tau} + \alpha(\Delta \Theta_{\tau+1} - \gamma \Theta_{\tau})}_{\tilde{\Theta}_{\tau+1} \text{ update}} \quad (8)$$

where  $\tau$  represents the update iteration,  $\rho \in (0, 1)$  is the momentum,  $\alpha \in (0, 1)$  denotes the learning rate, and  $\gamma \in (0, 1)$  is the weight decay. A more detailed discussion on

the choice of these parameters is provided by Hinton in [29]. Therein, the update rules are attained by deriving the energy functional with respect to free parameters (i.e., weights matrices, and the biases of each of the layers). In DFFW-CRBMs, a set of eight free parameters, corresponding to the connections between the factors and each of the layers, has to be inferred. These are presented below. Intuitively, each of these update equations, aims at minimizing the reconstruction error (i.e., the error between the original inputs and these reconstructed through the model). Moreover, each of the update equations include three main terms representing the connections between the factored weights and the corresponding layer of the machine, as per Figure 2. For instance, connections to only the hidden, history, and label layers suffice for updating  $W_{if}^{1v}$ . Thus, the update rules  $\Delta\Theta_\tau$  for each of the weights corresponding to the first factored layer, can be computed as:

$$\begin{aligned}
\Delta W_{if}^{1v} &\propto \left\langle v_{i,t} \sum_{j=1}^{n_h} W_{jf}^{1h} h_{j,t} \sum_{k=1}^{n_{v<t}} W_{kf}^{1v<t} v_{k,<t} \sum_{o=1}^{n_l} W_{of}^{1l} l_{o,t} \right\rangle_0 \\
&\quad - \left\langle v_{i,t} \sum_{j=1}^{n_h} W_{jf}^{1h} h_{j,t} \sum_{k=1}^{n_{v<t}} W_{kf}^{1v<t} v_{k,<t} \sum_{o=1}^{n_l} W_{of}^{1l} l_{o,t} \right\rangle_\lambda \\
\Delta W_{kf}^{1v<t} &\propto \left\langle v_{k,<t} \sum_{j=1}^{n_h} W_{jf}^{1h} h_{j,t} \sum_{i=1}^{n_v} W_{if}^{1v} v_{i,t} \sum_{o=1}^{n_l} W_{of}^{1l} l_{o,t} \right\rangle_0 \\
&\quad - \left\langle v_{k,<t} \sum_{j=1}^{n_h} W_{jf}^{1h} h_{j,t} \sum_{i=1}^{n_v} W_{if}^{1v} v_{i,t} \sum_{o=1}^{n_l} W_{of}^{1l} l_{o,t} \right\rangle_\lambda \\
\Delta W_{of}^{1l} &\propto \left\langle l_{o,t} \sum_{k=1}^{n_{v<t}} W_{kf}^{1v<t} v_{k,<t} \sum_{j=1}^{n_h} W_{jf}^{1h} h_{j,t} \sum_{i=1}^{n_v} W_{if}^{1v} v_{i,t} \right\rangle_0 \\
&\quad - \left\langle l_{o,t} \sum_{k=1}^{n_{v<t}} W_{kf}^{1v<t} v_{k,<t} \sum_{j=1}^{n_h} W_{jf}^{1h} h_{j,t} \sum_{i=1}^{n_v} W_{if}^{1v} v_{i,t} \right\rangle_\lambda \\
\Delta W_{jf}^{1h} &\propto \left\langle h_{j,t} \sum_{k=1}^{n_{v<t}} W_{kf}^{1v<t} v_{k,<t} \sum_{i=1}^{n_v} W_{if}^{1v} v_{i,t} \sum_{o=1}^{n_l} W_{of}^{1l} l_{o,t} \right\rangle_0 \\
&\quad - \left\langle h_{j,t} \sum_{k=1}^{n_{v<t}} W_{kf}^{1v<t} v_{k,<t} \sum_{i=1}^{n_v} W_{if}^{1v} v_{i,t} \sum_{o=1}^{n_l} W_{of}^{1l} l_{o,t} \right\rangle_\lambda,
\end{aligned}$$

while for the second factoring we have:

$$\begin{aligned}
\Delta W_{if}^{2v} &\propto \left\langle v_{i,t} \sum_{j=1}^{n_h} W_{jf}^{2h} h_{j,t} \sum_{k=1}^{n_v} W_{kf}^{2v} v_{k,<t} \sum_{o=1}^{n_l} W_{of}^{2l} l_{o,t} \right\rangle_0 \\
&\quad - \left\langle v_{i,t} \sum_{j=1}^{n_h} W_{jf}^{2h} h_{j,t} \sum_{k=1}^{n_v} W_{kf}^{2v} v_{k,<t} \sum_{o=1}^{n_l} W_{of}^{2l} l_{o,t} \right\rangle_\lambda \\
\Delta W_{kf}^{2v} &\propto \left\langle v_{k,<t} \sum_{j=1}^{n_h} W_{jf}^{2h} h_{j,t} \sum_{i=1}^{n_v} W_{if}^{2v} v_{i,t} \sum_{o=1}^{n_l} W_{of}^{2l} l_{o,t} \right\rangle_0 \\
&\quad - \left\langle v_{k,<t} \sum_{j=1}^{n_h} W_{jf}^{2h} h_{j,t} \sum_{i=1}^{n_v} W_{if}^{2v} v_{i,t} \sum_{o=1}^{n_l} W_{of}^{2l} l_{o,t} \right\rangle_\lambda \\
\Delta W_{of}^{2l} &\propto \left\langle l_{o,t} \sum_{k=1}^{n_v} W_{kf}^{2v} v_{k,<t} \sum_{j=1}^{n_h} W_{jf}^{2h} h_{j,t} \sum_{i=1}^{n_v} W_{if}^{2v} v_{i,t} \right\rangle_0 \\
&\quad - \left\langle l_{o,t} \sum_{k=1}^{n_v} W_{kf}^{2v} v_{k,<t} \sum_{j=1}^{n_h} W_{jf}^{2h} h_{j,t} \sum_{i=1}^{n_v} W_{if}^{2v} v_{i,t} \right\rangle_\lambda \\
\Delta W_{jf}^{2h} &\propto \left\langle h_{j,t} \sum_{k=1}^{n_v} W_{kf}^{2v} v_{k,<t} \sum_{i=1}^{n_v} W_{if}^{2v} v_{i,t} \sum_{o=1}^{n_l} W_{of}^{2l} l_{o,t} \right\rangle_0 \\
&\quad - \left\langle h_{j,t} \sum_{k=1}^{n_v} W_{kf}^{2v} v_{k,<t} \sum_{i=1}^{n_v} W_{if}^{2v} v_{i,t} \sum_{o=1}^{n_l} W_{of}^{2l} l_{o,t} \right\rangle_\lambda .
\end{aligned}$$

and for the biases are:

$$\begin{aligned}
\Delta a_i &\propto \langle v_{i,t} \rangle_0 - \langle v_{i,t} \rangle_\lambda \\
\Delta b_j &\propto \langle h_{j,t} \rangle_0 - \langle h_{j,t} \rangle_\lambda \\
\Delta c_o &\propto \langle l_{o,t} \rangle_0 - \langle l_{o,t} \rangle_\lambda
\end{aligned}$$

where  $\lambda$  represents a Markov chain step running for a total of  $n$  steps and starting at  
180 the original data distribution,  $\langle \cdot \rangle_0$  denotes the expectation under the input data, and  $\langle \cdot \rangle_\lambda$   
represents the model's expectation.

### 3.3.2. Sequential CD for DFFW-CRBMs

Algorithm 1 presents a high-level description of the sequential Markov chain con-  
trastive divergence [21] adapted to train DFFW-CRBMs. It shows the two main steps  
185 needed for training such machines. Firstly, the visible layer is inferred by fixing the his-  
tory and label layers. While in the second step the label layer is reconstructed by fixing  
the history and the present layers. Updating the weights involves the implementation  
of the rules derived in the previous section. These two procedures are then repeated

190 for a pre-specified number of epochs, where at each epoch the reconstruction error is decreasing to reach the minimum of the energy function, guaranteeing a minimized divergence between the original data distribution and the one given by the model.

---

**Inputs:** TD - training data,  $n$  - number of Markov Chain steps;

---

**Initialization:**  $\Theta \leftarrow \mathcal{N}(0, \sigma^2)$ , Set  $\alpha, \rho, \gamma$ ;

---

**for all epochs do**

**for each Sample  $\in$  TD do**

*%%First Markov Chain to reconstruct  $\mathbf{v}_t$ ;*

        Init  $\mathbf{v}_t \leftarrow 0$ ,  $\mathbf{l}_t = \text{Sample.Label}$ ,  $\mathbf{v}_{<t} = \text{Sample.History}$ ;

$\mathbf{h}_t = \text{InferHiddenLayer}(\mathbf{v}_t, \mathbf{l}_t, \mathbf{v}_{<t}, \Theta)$ ;

**for  $\lambda = 0; \lambda < n; \lambda++$  do**

*%%Positive phase;*

$\mathbf{pSt} = \text{GetPosStats}(\mathbf{h}_t, \text{Sample.Present}, \mathbf{l}_t, \mathbf{v}_{<t}, \Theta)$ ;

*%%Negative phase;*

$\mathbf{v}_t = \text{InferPresentLayer}(\mathbf{h}_t, \mathbf{l}_t, \mathbf{v}_{<t}, \Theta)$ ;

$\mathbf{h}_t = \text{InferHiddenLayer}(\mathbf{v}_t, \mathbf{l}_t, \mathbf{v}_{<t}, \Theta)$ ;

$\mathbf{nSt} = \text{GetNegStats}(\mathbf{h}_t, \mathbf{v}_t, \mathbf{l}_t, \mathbf{v}_{<t}, \Theta)$ ;

$\Theta = \text{UpdateWeights}(\mathbf{pSt}, \mathbf{nSt}, \Theta, \alpha, \rho, \gamma)$ ;

**end**

*%%Second Markov Chain to reconstruct  $\mathbf{l}_t$ ;*

        Init  $\mathbf{l}_t \leftarrow 0$ ,  $\mathbf{v}_t = \text{Sample.Present}$ ,  $\mathbf{v}_{<t} = \text{Sample.History}$ ;

$\mathbf{h}_t = \text{InferHiddenLayer}(\mathbf{v}_t, \mathbf{l}_t, \mathbf{v}_{<t}, \Theta)$ ;

**for  $\lambda = 0; \lambda < n; \lambda++$  do**

*%%Positive phase;*

$\mathbf{pSt} = \text{GetPosStats}(\mathbf{h}_t, \text{Sample.Label}, \mathbf{v}_t, \mathbf{v}_{<t}, \Theta)$ ;

*%%Negative phase;*

$\mathbf{l}_t = \text{InferLabelLayer}(\mathbf{h}_t, \mathbf{v}_t, \mathbf{v}_{<t}, \Theta)$ ;

$\mathbf{h}_t = \text{InferHiddenLayer}(\mathbf{v}_t, \mathbf{l}_t, \mathbf{v}_{<t}, \Theta)$ ;

$\mathbf{nSt} = \text{GetNegStats}(\mathbf{h}_t, \mathbf{v}_t, \mathbf{l}_t, \mathbf{v}_{<t}, \Theta)$ ;

$\Theta = \text{UpdateWeights}(\mathbf{pSt}, \mathbf{nSt}, \Theta, \alpha, \rho, \gamma)$ ;

**end**

**end**

**end**

---

**Algorithm 1:** Sequential Contrastive Divergence for DFFW-CRBMs

## 4. Experiments and Results

This section extensively tests the performance of DFFW-CRBMs on both simulated as well as on real-world datasets. The major goal of these experiments was to assess the capability of DFFW-CRBM to predict three-dimensional trajectories from two-dimensional projection, given small amounts of labeled data (i.e., in the order of 9-10 % of the total dataset). As a secondary objective, the goal was to classify such trajectories to different spins (ball trajectories) or activities (human pose estimation). In the real-valued prediction setting, we compared our method to state-of-the-art FFW-CRBMs and FCRBMs, while for classification our method’s performance was tested against FFW-CRBMs and support vector machines with radial basis functions (SVM-RBFs) [30].

**Evaluation Metrics:** To assess the models’ performance, a variety of standard metrics were used. For classification, we used accuracy [31] in percentages, while for estimation tasks, we used the Normalized Root Mean Square Error (NRMSE) estimating distance between the prediction and ground truth, Pearson Correlation Coefficient (PCC) reflecting the correlations between predictions and ground truth, and the P-value to arrive at statistically significant predictions.

### 4.1. Ball Trajectory Experiments

We generated different ball trajectories thrown with different spins using the Bullet Physics Library<sup>1</sup>. With this simulated dataset we targeted three objectives using small amounts (9 %) of labeled training data. First, we estimated 3D ball coordinates based on their 2D projections at each time-step  $t$  (i.e., one-step prediction). Second, we aimed at predicting near-future (i.e., couple of time steps in the future) 3D ball coordinates recursively, while giving limited 2D sequence of coordinates as a starting point. Third, we classified various ball spins based on just 2D coordinates. We used four trajectory classes corresponding to four different ball spin types. For each class, a set of 11 trajectories each containing approximately 400 time-steps (amounting to a total of

---

<sup>1</sup><http://bulletphysics.org>, Last accessed on November 8<sup>th</sup> 2016

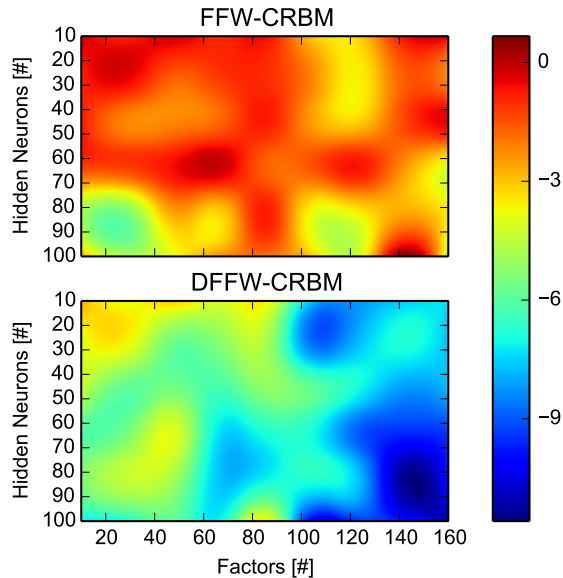


Figure 3: Averaged energy levels of FFW-CRBM and DFFW-CRBM over all ball trajectories when the parameters (i.e. number of hidden neurons and factors) are varying. The training was done for 100 epochs.

17211 data instances) were sampled. To assess the performance of DFFW-CRBM, we  
 220 performed 11-fold cross validation and reported mean and standard deviation results.  
 Precisely, from each class of trajectories we used only *one labeled trajectory*<sup>2</sup> to train  
 the models and the other 10 were used for testing.

**Deep Learner Setting:** The visible layers of both models (i.e. FFW-CRBM and  
 DFFW-CRBM) were set to 5 neurons, three denoting 3D ball center coordinates (i.e.  
 225 x, y, z), and two for its 2D projection at time  $t$ . The label layer consisted of 4 neu-  
 rons (one for each of the different spins classes), while the history layers included 100  
 neurons corresponding to the last 50 history frames. One frame incorporates the 2D  
 coordinates of the center of the ball projected in a two dimensional space. The number  
 of hidden neurons was set to 10, and the number of factors to 100, as discussed in the  
 230 next paragraph, and in Subsection 4.2. A learning rate of  $10^{-4}$  and momentum of 0.5

<sup>2</sup>A labeled trajectories has complete information: the 3D ball coordinates, their 2D projections, and the spin (i.e. class).



Task		Metrics	Methods			
			SVM-RBF	FCRBM	FFW-CRBM	DFFW-CRBM
Classification		Accuracy[%]	39.26±4.63	N/A	37.49±3.66	<b>39.51±4.47</b>
Present Step 3D estimation		NRMSE[%]	N/A	18.38±8.07	19.53±33.24	<b>11.24±8.53</b>
		PCC	N/A	-0.06±0.70	0.31±0.72	<b>0.62±0.61</b>
		P-value	N/A	0.51±0.29	0.40±0.29	<b>0.28±0.27</b>
Multi-Step 3D prediction	After 1 step	NRMSE[%]	N/A	25.61±3.25	23.53±2.48	<b>9.52±6.12</b>
		PCC	N/A	0.14±0.69	0.31±0.74	<b>0.95±0.14</b>
		P-value	N/A	0.50±0.29	0.38±0.25	<b>0.12±0.18</b>
3D prediction	After 50 steps	NRMSE[%]	N/A	31.38±7.99	29.49±8.14	<b>19.93±10.27</b>
		PCC	N/A	0.05±0.69	-0.05±0.72	<b>0.20±0.66</b>
		P-value	N/A	0.51±0.26	0.47±0.26	0.51±0.26

Table 1: Classification, present step 3D estimation, and multi-step 3D prediction for the balls trajectories experiment. Results, cross-validated and presented with mean and standard deviation, show that our method is capable of outperforming state-of-the-art techniques on all evaluation metrics.

were chosen. Weight decay factors were set to 0.0002, and the number of the Markov chain steps for CD in the training phase, but also for the Gibbs sampling in the testing phase, was set to 3. All weights were initialized with  $\mathcal{N}(0, 0.3)$ . Finally, data were normalized to have 0 mean and unit variance as explained in [29], and the models were trained for 100 epochs.

**Importance of disjunctive Factoring:** To find the optimal number of hidden neurons and factors, we have performed exhaustive search by varying the number of hidden neurons from 10 to 100 and the number of factors from 10 to 160. To gain some insights on the behavioral differences between FFW-CRBMs and DFFW-CRBMs, even if the energy equation of DFFW-CRBM has an extra tensor, in Figure 3 we illustrate on the same scale the heat-map of the averaged energy levels. They were computed using Equation 4 for FFW-CRBM and Equation 5 for DFFW-CRBM, after both models were trained for 100 epochs. Though both models acquire the lowest energy levels in a configuration starting with 10-20 hidden neurons and a number of factors larger than 100, analyzing these results signifies the importance of the disjunctive factoring introduced in the paper. Namely, DFFW-CRBMs always acquire lower energy levels compared to FFW-CRBMs due to its “specialized” tensor factoring. Moreover, by averaging the

energy levels from the aforementioned figure, we found that the average energy level of DFFW-CRBM is approximately three times smaller than the one of FFW-CRBM (i.e.  $-6.63 \pm 2.09$  for DFFW-CRBM, and  $-2.04 \pm 1.43$  for FFW-CRBM), thus anticipating the more accurate performance results, as showed next.

Figures 4 and 5 compare the capabilities of DFFW-CRBMs on estimating different 3D trajectories of balls picked at random to FFW-CRBMs, showing that our method is capable of achieving closely correlated transitions to the real trajectory. Interestingly, DFFW-CRBMs can handle discontinuities “less abruptly” compared to FFW-CRBMs. The cross-validation results showing the performance of all models of all ball trajectories are summarized in Table 1. In terms of classification, SVM-RBF, FFW-CRBM, and DFFW-CRBM perform almost similarly, with a slightly advantage of DFFW-CRBM<sup>3</sup>. In the case of 3D coordinates estimation from 2D projection at a time-step  $t$ , DFFW-CRBM clearly outperforms state-of-the-art methods with a NRMSE almost twice smaller than FCRBMs and FFW-CRBMs. Besides that, in this case, the mean value of the correlation coefficient for DFFW-CRBM is 0.62, double than that for FFW-CRBM, while the one for FCRBM is powerless (i.e below zero). For the multi-step prediction of near-future 3D point coordinates, DFFW-CRBM has an even more significant improvement. It is worth highlighting that in this scenario, the average PCC value after one step prediction is almost perfectly 0.95, while after 50 steps predicted into the future the mean PCC value is still positive and larger than those of the other methods. In a final set of experiments we tested the change in the accuracy of classification as a number of data points used. These are summarized in the bar-graph in Figure 6, showing that our method slightly outperforms the state-of-the-art techniques in all cases.

#### 4.2. Human Activity Recognition

Given the above successes, next we evaluate the performance of our method on real-world data representing a variety of human activities. In each set of experiments,

---

<sup>3</sup>It is worth noting that in this scenario the random guess for classification would have an accuracy of 25%.

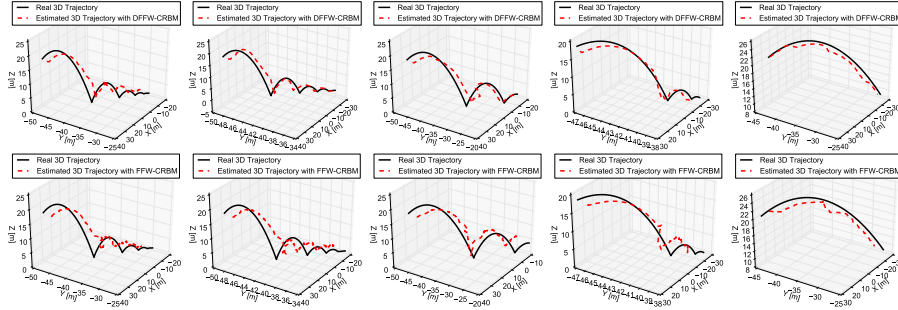


Figure 4: Estimation of different 3D balls trajectories from their 2D counterparts with DFFW-CRBM (top) and FFW-CRBM (bottom) showing that our method outperforms state-of-the-art techniques while requiring less data.

275 we targeted two main objectives and a third secondary one. The first two corresponded to estimating three-dimensional joint coordinates from two-dimensional projections as well as predicting such coordinates in near future, while the third involved classifying activities based on only two-dimensional joint coordinates. Please note that the third experiment is exceptionally hard due to the loss of three-dimensional information  
 280 making different activities more similar.

**Human 2.6m dataset.** For all experiments, we used the real-world comprehensive benchmark database [32, 33], containing 17 activities performed by 11 professional actors (6 males and 5 females) with over 3.6 million 3D human poses and their corresponding images. Further, for 7 actors, the database accurately reports 32 human  
 285 skeleton joint positions in 3D space, together with their 2D projections acquired at 50 frames per seconds (FPS).

We used these seven actors being Subject 1 (S1), Subject 5 (S5), Subject 6 (S6), Subject 7 (S7), Subject 8 (S8), Subject 9 (S9), Subject 11 (S11) accompanied with their corresponding joint activities, such as Purchasing (A1), Smoking (A2), Phoning (A3),  
 290 Sitting-Down (A4), Eating (A5), Walking-Together (A6), Greeting (A7), Sitting (A8), Posing (A9), Discussing (A10), Directing (A11), Walking (A12), and Waiting (A13). To avoid computational overhead, we have also reduced the temporal resolution of the data to 5 FPS leading to a total of 46446 training and testing instances. The instances were split between different subjects as: S1 (5514 instances), S5 (8748 instances), S6

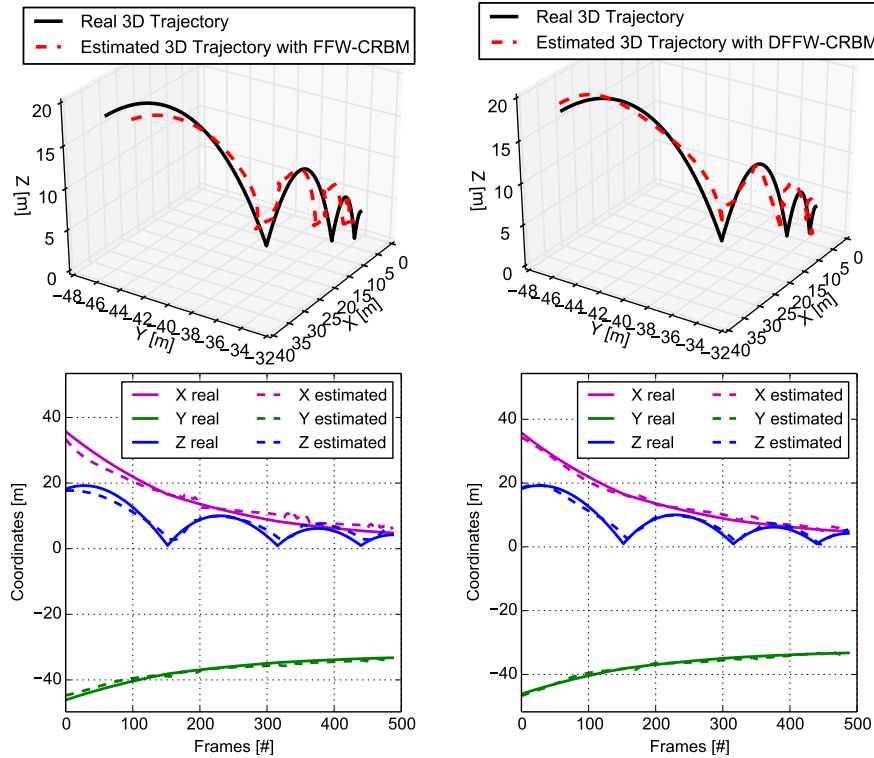


Figure 5: Estimation of the 3D trajectory for the center of one ball from its 2D projection using FFW-CRBM (left) and DFFW-CRBM (right). The top figure presents the trajectory in the 3D space, while the bottom figure presents the  $O_x$ ,  $O_y$ ,  $O_z$  coordinates of the same trajectory in a 2D plot.

295 (5402 instances), S7 (9081 instances), S8 (5657 instances), S9 (6975 instances), and S11 (5069 instances).

**Deep Learner Setting:** The visible layers of both the FFW-CRBM and DFFW-CRBM were set to 160 neurons corresponding to 96 neurons for the 3D coordinates of the joints, and 64 for their 2D projections at time  $t$ . The label layer consisted of 13  
300 neurons (one for each of the activities), and the history layers included 320 neurons corresponding to 5 history frames each incorporating 2D joint coordinates. The size of the hidden layer was set to 10 neurons, and the number of factors to 100, as explained in the next paragraph. Furthermore, a learning rate of  $10^{-5}$  was used to guarantee bounded reconstruction errors. The number of the Markov Chain steps in the training

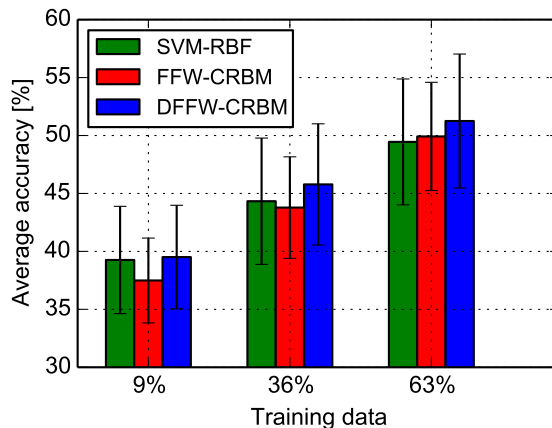


Figure 6: Average classification accuracies with mean and standard deviation, over all balls trajectories, when the amount of training data is increased.

305 phase and of the Gibbs sampling in the testing phase were set to 3, and the weights were initialized with  $\mathcal{N}(0, 0.3)$ . Further particularities, such as momentum and weight decay were set to 0.5 and 0.0002. Also, all data were normalized to have a 0 mean and unit standard deviation.

**Importance of disjunctive Factoring:** Similarly with the previous experiment on  
 310 simulated balls trajectories, we searched for the optimal number of hidden neurons and factors, by performing exhaustive search and varying the number of hidden neurons and factors from 10 to 100 and from 10 to 160, respectively. Figure 7 depicts on the same scale the averaged energy levels for both FFW-CRBM and DFFW-CRBM, after being trained for 100 epochs. As before, in the balls experiment, the energy  
 315 levels of both models are more affected by the number of factors than the number of hidden neurons. Even if we are scrutinizing unnormalized energy levels, the fact that the energy levels of DFFW-CRBM are always much lower than the energy levels of FFW-CRBM reflects the importance of the disjunctive factoring. By quantifying and averaging all the energy levels for each model, we may observe that DFFW-CRBM has  
 320 in average approximately three times less energy than FFW-CRBM (i.e.  $-153.32 \pm 17.31$  for DFFW-CRBM, and  $-48.57 \pm 33.92$  for FFW-CRBM).

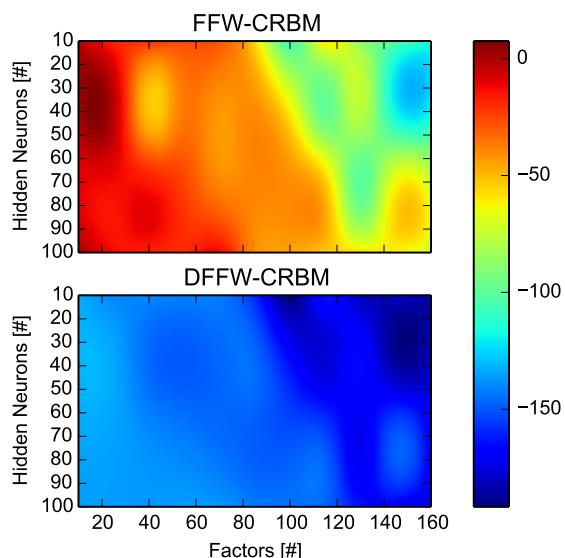


Figure 7: Averaged energy levels of FFW-CRBM and DFFW-CRBM for the human activities experiments when the parameters (i.e. number of hidden neurons and factors) are varying. The training was done for 100 epochs.

#### 4.2.1. Training and Testing on The Same Person

Here, data from the same subject has been used for both training and testing. Emulating real-world 3D trajectory prediction settings where labeled data is scarce, we made use of only 10% of the available data for training and 90% for testing with the aim of performing accurate one and multi-step 3D trajectory predictions.

Results in Tables 2, 3, and 4 show that DFFW-CRBMs are capable of achieving better performance than state-of-the-art techniques in both classification and prediction even when only using a small amount of training data. These results provide a proof-of-concept to the fact that DDFW-CRBMs are capable of accurately predicting (in both one-step and multi-step scenarios) 3D trajectories from their 2D projections by using only 10% of the data for training and 90% for testing.

**Activity Recognition (Classification)** The goal in this set of experiments was to classify the 13 activities based on only their 2D projections. Please note that such a task is substantially difficult to solve due to the loss of information exhibited by the

Persons	SVM-RBF	FFW-CRBM	DFFW-CRBM
S1	49.77	50.53	49.34
S5	36.92	38.82	40.21
S6	30.68	31.51	30.18
S7	38.50	37.03	37.94
S8	26.49	30.41	31.32
S9	24.69	28.12	22.63
S11	34.56	34.21	32.16
Average	34.51	<b>35.80</b>	34.83

Table 2: Classification accuracies in percentages for the human activities experiments, when training and testing data belong to the same person.

Persons	FCRBM			FFW-CRBM			DFFW-CRBM		
	NRMSE [%]	PCC	P-value	NRMSE [%]	PCC	P-value	NRMSE [%]	PCC	P-value
S1	8.41 ± 3.75	0.02 ± 0.10	0.49 ± 0.29	9.93 ± 7.47	0.13 ± 0.37	0.15 ± 0.26	6.36 ± 3.45	0.54 ± 0.29	0.05 ± 0.16
S5	6.70 ± 2.44	-0.03 ± 0.09	0.54 ± 0.28	6.95 ± 3.21	0.10 ± 0.33	0.16 ± 0.26	4.30 ± 2.30	0.68 ± 0.25	0.02 ± 0.10
S6	4.41 ± 1.93	0.03 ± 0.09	0.53 ± 0.28	4.50 ± 2.37	0.01 ± 0.28	0.21 ± 0.29	3.19 ± 1.64	0.50 ± 0.32	0.05 ± 0.16
S7	9.14 ± 3.46	0.02 ± 0.10	0.49 ± 0.29	9.16 ± 4.30	0.13 ± 0.35	0.14 ± 0.25	6.19 ± 3.11	0.71 ± 0.24	0.01 ± 0.09
S8	8.31 ± 3.37	-0.00 ± 0.11	0.47 ± 0.29	8.23 ± 4.42	0.02 ± 0.27	0.26 ± 0.31	4.96 ± 2.57	0.62 ± 0.26	0.05 ± 0.18
S9	7.25 ± 2.74	0.00 ± 0.09	0.55 ± 0.28	8.40 ± 5.05	0.01 ± 0.27	0.22 ± 0.29	4.63 ± 2.34	0.54 ± 0.32	0.05 ± 0.17
S11	9.62 ± 4.05	-0.00 ± 0.10	0.54 ± 0.28	9.89 ± 5.94	0.06 ± 0.32	0.15 ± 0.25	6.82 ± 3.82	0.53 ± 0.35	0.04 ± 0.14
Average	≈7.69 ± 3.10	≈0.01 ± 0.09	≈0.51 ± 0.28	≈8.15 ± 4.68	≈0.07 ± 0.31	≈0.18 ± 0.27	≈5.21 ± 2.75	≈0.59 ± 0.29	≈0.04 ± 0.14

Table 3: The 3D estimation of the human joints from their 2D counterpart at the present time, when the training and testing are done on the same person. The results are presented with mean and standard deviation.

Steps Predicted	Persons	FCRBM			FFW-CRBM			DFFW-CRBM		
		NRMSE [%]	PCC	P-value	NRMSE [%]	PCC	P-value	NRMSE [%]	PCC	P-value
After 1 step	S1	7.68 ± 3.71	0.02 ± 0.09	0.51 ± 0.28	7.78 ± 4.95	0.17 ± 0.32	0.17 ± 0.27	5.91 ± 3.45	0.55 ± 0.26	0.06 ± 0.16
	S5	6.72 ± 2.48	-0.05 ± 0.09	0.50 ± 0.28	6.41 ± 2.70	0.19 ± 0.37	0.18 ± 0.29	3.88 ± 1.68	0.68 ± 0.25	0.01 ± 0.08
	S6	4.40 ± 2.17	0.04 ± 0.09	0.52 ± 0.28	4.27 ± 2.31	0.11 ± 0.31	0.21 ± 0.30	3.17 ± 1.83	0.48 ± 0.32	0.05 ± 0.16
	S7	9.07 ± 3.20	0.02 ± 0.11	0.46 ± 0.29	8.78 ± 3.52	0.27 ± 0.35	0.06 ± 0.17	6.52 ± 3.05	0.73 ± 0.17	0.00 ± 0.02
	S8	7.16 ± 3.08	0.01 ± 0.12	0.48 ± 0.31	6.42 ± 3.51	0.04 ± 0.23	0.30 ± 0.31	3.93 ± 1.87	0.69 ± 0.19	0.01 ± 0.05
	S9	6.98 ± 2.64	-0.01 ± 0.09	0.54 ± 0.29	6.91 ± 3.17	0.08 ± 0.26	0.22 ± 0.28	4.40 ± 1.69	0.64 ± 0.20	0.01 ± 0.08
	S11	9.55 ± 4.05	-0.00 ± 0.08	0.56 ± 0.25	8.92 ± 4.66	0.10 ± 0.31	0.18 ± 0.27	7.02 ± 4.00	0.51 ± 0.44	0.04 ± 0.14
Average	≈7.37 ± 3.05	≈0.00 ± 0.09	≈0.51 ± 0.28	≈7.07 ± 3.55	≈0.14 ± 0.31	≈0.18 ± 0.27	≈4.96 ± 2.51	≈0.61 ± 0.26	≈0.03 ± 0.1	
After 50 steps	S1	10.88 ± 3.17	0.01 ± 0.10	0.55 ± 0.32	10.23 ± 5.39	0.10 ± 0.41	0.17 ± 0.26	8.22 ± 4.56	-0.03 ± 0.29	0.14 ± 0.24
	S5	7.50 ± 2.20	0.02 ± 0.10	0.53 ± 0.28	8.40 ± 3.20	0.01 ± 0.43	0.19 ± 0.30	6.86 ± 2.47	0.12 ± 0.43	0.08 ± 0.19
	S6	5.38 ± 1.92	0.01 ± 0.12	0.45 ± 0.29	4.77 ± 2.65	-0.02 ± 0.24	0.22 ± 0.30	4.44 ± 2.17	0.12 ± 0.32	0.12 ± 0.23
	S7	11.07 ± 3.61	-0.03 ± 0.09	0.52 ± 0.30	10.68 ± 4.04	0.11 ± 0.44	0.15 ± 0.26	9.31 ± 3.60	0.17 ± 0.33	0.12 ± 0.25
	S8	15.41 ± 1.66	0.01 ± 0.11	0.45 ± 0.29	11.33 ± 5.81	0.09 ± 0.24	0.26 ± 0.29	9.91 ± 5.73	0.08 ± 0.38	0.10 ± 0.21
	S9	9.25 ± 2.10	0.01 ± 0.11	0.48 ± 0.28	8.56 ± 3.23	0.03 ± 0.25	0.25 ± 0.28	7.48 ± 3.60	-0.02 ± 0.39	0.12 ± 0.22
	S11	14.39 ± 2.71	-0.01 ± 0.09	0.52 ± 0.28	10.93 ± 5.78	0.17 ± 0.37	0.18 ± 0.29	8.56 ± 4.47	0.05 ± 0.51	0.12 ± 0.26
Average	≈10.55 ± 2.48	≈0.00 ± 0.10	≈0.5 ± 0.29	≈9.27 ± 4.3	≈0.07 ± 0.34	≈0.20 ± 0.28	≈7.82 ± 3.8	≈0.07 ± 0.37	≈0.11 ± 0.23	

Table 4: Multi-step 3D prediction for the human activities experiments, when the training and testing are done on the same person. The results are presented with mean and standard deviation.

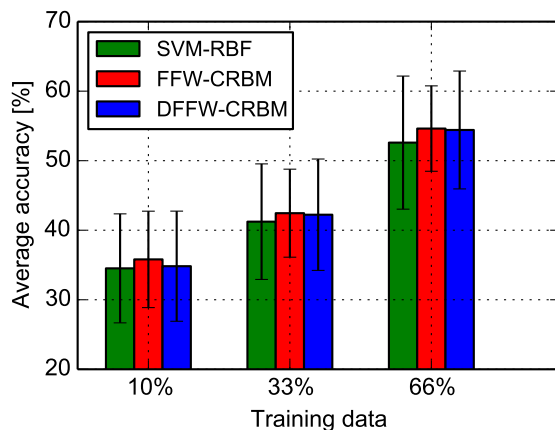


Figure 8: Average classification accuracies with mean and standard deviation for the human activities experiments, over all subjects, when the data for training and testing the models come from the same person and the amount of training data is increased.

performed projection. Namely, activities different in 3D space might resemble high similarities in their 2D projections leading to low classification accuracies. Table 2 reports the accuracy performance of DFFW-CRBMs, against state-of-the-art methods including SVMs and FFW-CRBMs. By averaging the results over all subjects, we can observe that all three models perform comparable. It is worth mentioning that the classification accuracy for random choice in this scenario would be 7.69% and all models performs approximately 5 times better.

We also performed two more experiments to classify activities with more input data points to prove the correctness of the presented methods and show DFFW-CRBMs is capable of achieving state-of-the-art classification results. Here, we used 33% and 66% of the data to train the models, and the remaining for test. It is clear from Figure 8 that all models increase in performance as the amount of training data increases, reaching around 55% accuracy when 66% of the data is used for training.

**Estimating 3D Skeleton Coordinates from 2D Projections (Present Step Prediction)** In this task, we estimate the 3D joint coordinates from their 2D counterpart while using 10% training data. Results depicted in Table 3 show that DFFW-CRBMs achieves better performance than FFW-CRBMs and FCRBM. Though FFW-CRBMs



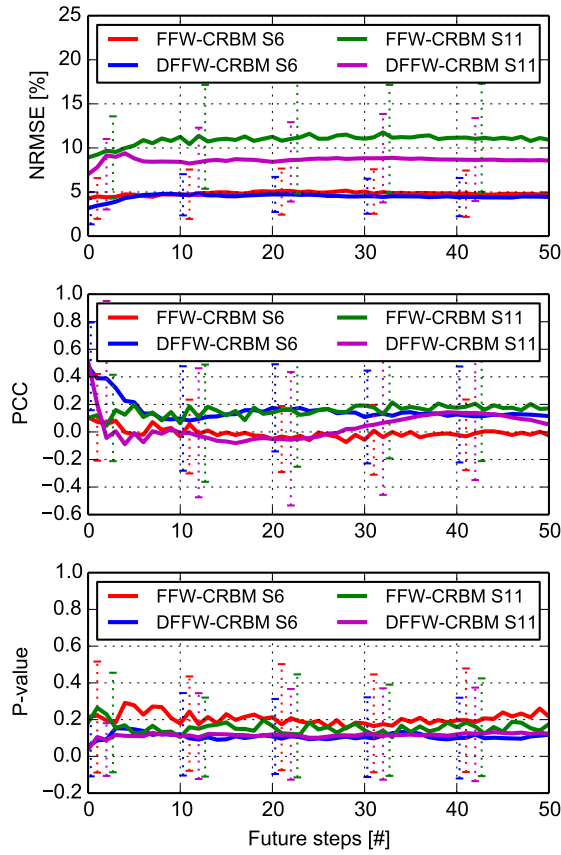


Figure 9: Multi-step 3D prediction on the worst performer (S11) and best performer (S6) subjects, when the data for training and testing the models come from the same person.

perform comparatively, it is worth noting that the PCC and P-values signify the fact that DFFW-CRBMs drastically outperform FFW-CRBMs in the sense that the predic-  
 355 tions are correlated with ground truth, a property essential for accurate and reliable predictions.

**Prediction of 3D Skeleton Trajectories (Multi-Step Prediction)** Here, the goal was to perform multi-step predictions of the 3D skeleton joints based on only 2D pro-  
 jections. Starting from a 2D initial state, the model was executed autonomously by  
 360 recursively feeding-back 2D outputs to perform next-step predictions. Definitely, the performance is expected to degrade since the prediction errors accumulate with time.

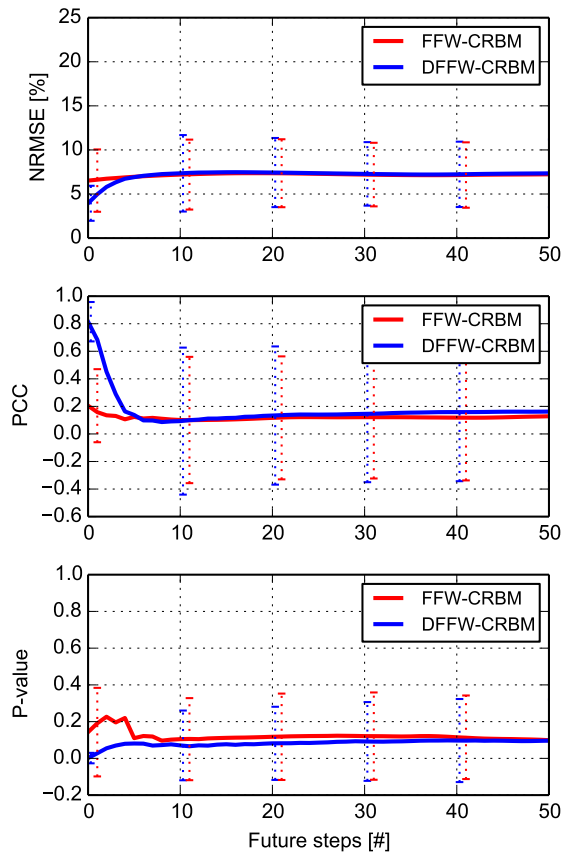


Figure 10: Multi-step 3D prediction using cross-validation on all subjects, when the data for training and testing the models come from different persons.

Table 4, showing the performance of the models after 1 and 50 step predictions, validate this phenomenon since all metrics show a decrease in both models' performance over time. Table 4, however, also signify that DFFW-CRBMs outperform FFW-CRBMs in both one and multi-step predictions achieving an average NRMSE of 7.82 compared to 9.27 NRMSE for FFW-CRBMs. Further results are summarized by Figure 9 showing the minimum and maximum performance results of both models. In these experiments, clearly, DFFW-CRBM is the best performer in both, prediction errors and correlations.

Task		Metrics	Methods			
			SVM-RBF	FCRBM	FFW-CRBM	DFFW-CRBM
Classification		Accuracy[%]	37.93±5.04	N/A	<b>44.96±2.68</b>	44.49±6.60
Present Step 3D estimation		NRMSE[%]	N/A	7.58±3.62	7.52±3.63	<b>3.93±1.75</b>
		PCC	N/A	-0.00±0.09	0.14±0.24	<b>0.79±0.16</b>
		P-value	N/A	0.52±0.28	0.21±0.28	<b>0.01±0.03</b>
Multi-Step	After 1 step	NRMSE[%]	N/A	6.60±3.53	6.52±3.54	<b>3.95±1.99</b>
		PCC	N/A	-0.01±0.11	0.21±0.27	<b>0.81±0.14</b>
		P-value	N/A	0.49±0.29	0.14±0.24	<b>0.01±0.03</b>
3D prediction	After 50 steps	NRMSE[%]	N/A	7.27±3.81	<b>7.24±3.84</b>	7.34±3.84
		PCC	N/A	0.01±0.11	0.13±0.46	<b>0.16±0.50</b>
		P-value	N/A	0.49±0.31	0.10±0.20	0.10±0.22

Table 5: Classification, present step 3D estimation, and multi-step 3D prediction, for the human activities experiments, when the training and the testing are done on different persons. The results are cross-validated and presented with mean and standard deviation.

#### 4.2.2. Testing Generalization Capabilities

370 **Motivation:** In the second set of human activities experiments, our goal was to determine to what extent can DFFW-CRBMs generalize across different human subjects and activities. The main motivation is that in reality subject-specific data is scarce, while data available from different users or domains is abundant. Results reported in Table 5 and Figure 10 show that DFFW-CRBMs are capable of generalizing beyond  
375 specific subjects due to their ability in learning latent features shared among a variety of tasks.

**Experiments:** Here, data from 6 subjects was used to train the models, and predictions on an unseen subject were performed. The procedure was then repeated to cross-validate the results. Further, to emulate real-world settings only 10% of the data  
380 was used for training. During testing, however, all data from the all testing subjects was used increasing the tasks' difficulty. The same three goals of the previous experiments were targeted.

**Activity Recognition (Classification):** Results reported in Table 5 show that DFFW-CRBMs achieve comparable results to FFW-CRBMs at an accuracy of 44.5 % both  
385 outperforming SVMs. Clearly, these classification results resemble higher accuracies

when compared to these in Table 2. The reasons can be attributed back to the availability of similar domain data from other subjects signifying the latent feature similarities automatically learn by DFFW-CRBMs.

**Estimation of 3D Skeleton Coordinates from 2D Projections (Present Step Prediction):** Again, DFFW-CRBMs achieve better performance than FFW-CRBMs in present step estimation of the 3D skeleton joints from 2D projections, while both outperform FCRBM. It is worth highlighting that DFFW-CRBMs are capable of attaining a high average prediction correlation to ground-truth of almost 0.8.

**Prediction of 3D skeleton Trajectories (Multi-Step Prediction):** Finally, Figure 10 shows that DFFW-CRBMs are capable of surpassing FFW-CRBMs in multi-step predictions on unseen subjects achieving low prediction errors and high ground truth correlation.

## 5. Conclusion

In this paper we proposed *disjunctive factored four-way conditional restricted Boltzmann machines* (DFFW-CRBMs). These novel machine learning techniques can be used for estimating 3D trajectories from their 2D projections using limited amounts of labeled data. Due to the new tensor factoring introduced by DFFW-CRBMs, these machines are capable of achieving substantially lower energy levels than state-of-the-art techniques leading to more accurate predictions and classification results. Furthermore, DFFW-CRBMs are capable of performing classification and accurate near-future predictions simultaneously in one unified framework.

Two sets of experiments, one on a simulated ball trajectories dataset and one on a real-world benchmark database, demonstrate the effectiveness of DFFW-CRBMs. The empirical evaluation showed that our methods are capable of outperforming state-of-the-art machine learning algorithms in both classification and regression. Precisely, DFFW-CRBM were capable of achieving substantially lower energy levels (approximately three times less energy on the overall datasets, independently on the number of factors or hidden neurons) than FFW-CRBM. This leads to at least double accuracies for real-valued predictions, while acquiring similar classification performance, at no

415 increased computational complexity costs.

## References

- [1] H. Silva, A. Dias, J. Almeida, A. Martins, E. Silva, Real-time 3d ball trajectory estimation for robocup middle size league using a single camera, in: T. Rfer, N. Mayer, J. Savage, U. Saranl (Eds.), RoboCup 2011: Robot Soccer World Cup XV, Vol. 7416 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 420 2012, pp. 586–597. doi:10.1007/978-3-642-32060-6\_50.
- [2] S. Qiu, Y. Yang, J. Hou, R. Ji, H. Hu, Z. Wang, Ambulatory estimation of 3d walking trajectory and knee joint angle using marg sensors, in: Innovative Computing Technology (INTECH), 2014 Fourth International Conference on, 2014, 425 pp. 191–196. doi:10.1109/INTECH.2014.6927742.
- [3] S. Maleschlijski, G. Sendra, A. Di Fino, L. Leal-Taix, I. Thome, A. Terfort, N. Aldred, M. Grunze, A. Clare, B. Rosenhahn, A. Rosenhahn, Three dimensional tracking of exploratory behavior of barnacle cyprids using stereoscopy, *Biointerphases* 7 (1-4). doi:10.1007/s13758-012-0050-x.
- 430 [4] M. Jauzac, E. Jullo, J.-P. Kneib, H. Ebeling, A. Leauthaud, C.-J. Ma, M. Limousin, R. Massey, J. Richard, A weak lensing mass reconstruction of the large-scale filament feeding the massive galaxy cluster macsj0717.5+3745, *Monthly Notices of the Royal Astronomical Society* 426 (4) (2012) 3369–3384. doi:10.1111/j.1365-2966.2012.21966.x.
- 435 [5] J. Tao, B. Risse, X. Jiang, R. Klette, 3d trajectory estimation of simulated fruit flies, in: Proceedings of the 27th Conference on Image and Vision Computing New Zealand, IVCNZ '12, ACM, New York, NY, USA, 2012, pp. 31–36. doi:10.1145/2425836.2425844.
- 440 [6] J. Pinezich, J. Heller, T. Lu, Ballistic projectile tracking using cw doppler radar, *Aerospace and Electronic Systems, IEEE Transactions on* 46 (3) (2010) 1302–1311. doi:10.1109/TAES.2010.5545190.

- [7] H. Larochelle, Y. Bengio, Classification using discriminative restricted boltzmann machines, in: Proceedings of the 25th International Conference on Machine Learning, ICML '08, ACM, 2008, pp. 536–543. doi:10.1145/1390156.1390224.
- [8] R. Salakhutdinov, A. Mnih, G. Hinton, Restricted boltzmann machines for collaborative filtering, in: In Machine Learning, Proceedings of the Twenty-fourth International Conference (ICML 2004). ACM, AAAI Press, 2007, pp. 791–798.
- [9] D. C. Mocanu, G. Exarchakos, A. Liotta, Deep learning for objective quality assessment of 3d images, in: Image Processing (ICIP), 2014 IEEE International Conference on, 2014, pp. 758–762. doi:10.1109/ICIP.2014.7025152.
- [10] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, D. Hassabis, Human-level control through deep reinforcement learning, Nature 518 (7540) (2015) 529–533. doi:10.1038/nature14236.
- [11] H. Bou-Ammar, D. C. Mocanu, M. E. Taylor, K. Driessens, K. Tuyls, G. Weiss, Automatically mapped transfer between reinforcement learning tasks via three-way restricted boltzmann machines, in: Machine Learning and Knowledge Discovery in Databases, Vol. 8189 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2013, pp. 449–464. doi:10.1007/978-3-642-40991-2\_29.
- [12] P. V. Gehler, A. D. Holub, M. Welling, The rate adapting poisson model for information retrieval and object recognition, in: In Proceedings of 23rd International Conference on Machine Learning (ICML06), ACM Press, 2006, p. 2006.
- [13] D. Eigen, C. Puhrsch, R. Fergus, Depth map prediction from a single image using a multi-scale deep network, in: Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada, 2014, pp. 2366–2374.

- 470 [14] P. Rasti, T. Uiboupin, S. Escalera, G. Anbarjafari, Convolutional Neural Network Super Resolution for Face Recognition in Surveillance Monitoring, Springer International Publishing, Cham, 2016, pp. 175–184. doi:10.1007/978-3-319-41778-3\_18.
- [15] K. Nasrollahi, S. Escalera, P. Rasti, G. Anbarjafari, X. Baro, H. J. Escalante, T. B. Moeslund, Deep learning based super-resolution for improved action recognition, in: 2015 International Conference on Image Processing Theory, Tools and Applications (IPTA), 2015, pp. 67–72. doi:10.1109/IPTA.2015.7367098.
- 475 [16] J. Shotton, R. Girshick, A. Fitzgibbon, T. Sharp, M. Cook, M. Finocchio, R. Moore, P. Kohli, A. Criminisi, A. Kipman, A. Blake, Efficient human pose estimation from single depth images, IEEE Transactions on Pattern Analysis and Machine Intelligence 35 (12) (2013) 2821–2840. doi:http://doi.ieeecomputersociety.org/10.1109/TPAMI.2012.241.
- 480 [17] I. Sutskever, G. E. Hinton, Learning multilevel distributed representations for high-dimensional sequences., in: M. Meila, X. Shen (Eds.), AISTATS, Vol. 2 of JMLR Proceedings, JMLR.org, 2007, pp. 548–555.
- 485 [18] P. Smolensky, Information processing in dynamical systems: Foundations of harmony theory, in: D. E. Rumelhart, J. L. McClelland, et al. (Eds.), Parallel Distributed Processing: Volume 1: Foundations, MIT Press, Cambridge, 1987, pp. 194–281.
- 490 [19] G. W. Taylor, G. E. Hinton, Factored conditional restricted boltzmann machines for modeling motion style, in: Proceedings of the 26th Annual International Conference on Machine Learning, ICML '09, 2009, pp. 1025–1032.
- [20] G. W. Taylor, G. E. Hinton, S. T. Roweis, Two distributed-state models for generating high-dimensional time series, Journal of Machine Learning Research 12 (2011) 1025–1068.
- 495 [21] D. C. Mocanu, H. Bou-Ammar, D. Lowet, K. Driessens, A. Liotta, G. Weiss, K. Tuyls, Factored four way conditional restricted boltzmann machines for activ-

- ity recognition, *Pattern Recognition Letters* doi:<http://dx.doi.org/10.1016/j.patrec.2015.01.013>.
- 500 [22] D. C. Mocanu, M. T. Vega, E. Eaton, P. Stone, A. Liotta, Online contrastive divergence with generative replay: Experience replay without storing data, *CoRR* abs/1610.05555.
- [23] Y. Bengio, Learning deep architectures for ai, *Found. Trends Mach. Learn.* 2 (1) (2009) 1–127.
- 505 [24] D. C. Mocanu, E. Mocanu, P. H. Nguyen, M. Gibescu, A. Liotta, A topological insight into restricted boltzmann machines, *Machine Learning* 104 (2) (2016) 243–270. doi:[10.1007/s10994-016-5570-z](https://doi.org/10.1007/s10994-016-5570-z).
- [25] G. E. Hinton, Training Products of Experts by Minimizing Contrastive Divergence, *Neural Computation* 14 (8) (2002) 1771–1800.
- 510 [26] M. Ponti, J. Kittler, M. Riva, T. de Campos, C. Zor, A decision cognizant kullbackleibler divergence, *Pattern Recognition* 61 (2017) 470 – 478. doi:<http://dx.doi.org/10.1016/j.patcog.2016.08.018>.
- [27] S. Elaiwat, M. Bennamoun, F. Boussaid, A spatio-temporal rbm-based model for facial expression recognition, *Pattern Recognition* 49 (2016) 152 – 161. doi:  
515 <http://dx.doi.org/10.1016/j.patcog.2015.07.006>.
- [28] G. Hinton, M. Pollefeys, J. Susskind, R. Memisevic, Modeling the joint density of two images under a variety of transformations, *2013 IEEE Conference on Computer Vision and Pattern Recognition* 00 (undefined) (2011) 2793–2800. doi:  
doi.[ieeecomputersociety.org/10.1109/CVPR.2011.5995541](https://doi.org/10.1109/CVPR.2011.5995541).
- 520 [29] G. Hinton, A practical guide to training restricted boltzmann machines, in: *Neural Networks: Tricks of the Trade*, Vol. 7700 of *Lecture Notes in Computer Science*, Springer, 2012, pp. 599–619. doi:[10.1007/978-3-642-35289-8\\_32](https://doi.org/10.1007/978-3-642-35289-8_32).
- [30] C. Cortes, V. Vapnik, Support-Vector Networks, *Mach. Learn.* 20 (3) (1995) 273–297.



- 525 [31] T. Fawcett, An introduction to ROC analysis, *Pattern Recognition Letters* 27 (8) (2006) 861–874.
- [32] C. S. Catalin Ionescu, Fuxin Li, Latent structured models for human pose estimation, in: *International Conference on Computer Vision*, 2011.
- 530 [33] C. Ionescu, D. Papava, V. Olaru, C. Sminchisescu, Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36 (7) (2014) 1325–1339.