Towards Natural and Accurate Future Motion Prediction of Humans and Animals

Zhenguang Liu*[1], Shuang Wu*[2,3], Shuyuan Jin[4], Qi Liu[4], Shijian Lu[3], Roger Zimmermann[4], Li Cheng[2,5]
[1]Zhejiang Gongshang University, [2]Bioinformatics Institute, A*STAR,
[3]Nanyang Technological University, [4]National University of Singapore, [5]University of Alberta
liuzhenguang2008@gmail.com, wushuang@bii.a-star.edu.sg, shuyuanjin@u.nus.edu,
leuchine@gmail.com, shijian.lu@ntu.edu.sg, rogerz@comp.nus.edu.sg, chengli@bii.a-star.edu.sg

Abstract

Anticipating the future motions of 3D articulate objects is challenging due to its non-linear and highly stochastic nature. Current approaches typically represent the skeleton of an articulate object as a set of 3D joints, which unfortunately ignores the relationship between joints, and fails to encode fine-grained anatomical constraints. Moreover, conventional recurrent neural networks, such as LSTM and GRU, are employed to model motion contexts, which inherently have difficulties in capturing long-term dependencies. To address these problems, we propose to explicitly encode anatomical constraints by modeling their skeletons with a Lie algebra representation. Importantly, a hierarchical recurrent network structure is developed to simultaneously encodes local contexts of individual frames and global contexts of the sequence. We proceed to explore the applications of our approach to several distinct quantities including human, fish, and mouse. Extensive experiments show that our approach achieves more natural and accurate predictions over state-of-the-art methods.

1. Introduction

For a human being, it is usually not very hard to predict short-term future motions of the moving objects around them. Without this ability, it would be extremely difficult for us to walk in a crowded street, get past defenders in a football game, or avoid imminent dangers in movement. Similarly, anticipating the movements of articulate objects, especially humans and animals, is crucial for a machine to adjust its behavior, plan its action, and properly allocate its attention when interacting with humans and animals. Natural and accurate future motion prediction is also highly valuable for a wide range of applications including high-fidelity animal simulation in games and movies, human or animal tracking, and intelligent driving [4, 25, 21, 31].

In this paper, we focus on the problem of predicting future 3D poses of an articulate object given its prior skeleton sequence. The problem is challenging due to the non-linear dynamics, high dimensionality, and stochastic nature of human or animal movements. Conventional approaches utilize latent-variable models, such as hidden Markov models [18], Gaussian processes [29], and restricted Boltzmann machine [26] to capture the temporal dynamics of human motions. Recently, recurrent neural networks (RNNs) based methods are introduced with improved performance. For example, [9] uses an Encoder-Recurrent-Decoder network where the long short-term memory (LSTM) is utilized in the recurrent layer. [17] divides human body into spine, arms, and legs, and uses multiple RNNs to model interactions between different body parts. Further, [21] and [25] resort to residual Gated Recurrent Unit (GRU) and Modified Highway Unit (MHU) to capture motion contexts.

Scrutinizing the released implementations of existing methods [17, 21, 9], one observes that current methods often encounter difficulties in obtaining *natural* and *accurate* future motion prediction. Specifically, for relatively long-term prediction, existing methods tend to degrade into motionless states or drift away to non-human like motions. For short-term prediction, there often exists a clear discontinuity between the prior pose sequence and the first prediction [17]. Interestingly, quantitative evaluations revealed that many existing methods may be outperformed by a trivial baseline that simply predicts the future as its last observed pose [21].

We believe these issues are mainly due to the following reasons. *First*, current algorithms do not respect the physical laws of motions based on the skeletal anatomy. This often leads to strange distortions in the predicted motion. *Second*, In modelling temporal motion dynamics, current approaches rely on conventional recurrent units, such as

*Denotes equal contribution

LSTM and GRU, where the hidden state sequentially reads in a frame and updates its value. The hidden state then tends to be overwhelmed by the inputs in recent time steps and such recurrent units are known to have difficulties in capturing long-term dependencies [2]. Moreover, the sequential updating characteristic of these architectures may lend itself to undesirable computational bottlenecks in practice.

To address these issues, we propose a novel architecture, which consists of a dedicated Hierarchical Motion Recurrent (HMR) network incorporated with a Lie algebra representation. Specifically, we characterize the pose of an articulate object, being e.g. a human, a mouse or a fish, as a kinematic tree consisting of one or multiple kinematic chains based on the mathematical formalism of Lie algebra. Fine-grained anatomical constraints are explicitly and naturally encoded. The pose sequence is then fed into our proposed HMR network to learn its temporal evolution. Motion contexts are jointly modeled by a state hierarchy consisting of local states for individual frames and an overall sequence-level state. At each recurrent step, a frame exchanges contextual information with its neighboring frames in two directions, as well as with the sequence-level state. Differing from the traditional RNN architecture, the number of recurrent steps in our network does not scale with the sequence length. Empirically, our method can effectively model motion contexts in around 10 recurrent steps.

Moreover, state-of-the-art motion prediction methods typically focus on humans. The principled approach that we have developed generalizes well across object categories and is easily adapted to modeling animal motions. Empirically, our approach achieves state-of-the-art results on the H3.6m benchmark dataset with a much enhanced long-term proficiency, capable of predicting natural human-like motions over 50 seconds, and works well on animal datasets such as fish and mouse.

To summarize, our key contributions are: 1) A novel hierarchical RNN structure is proposed to effectively model global and local motion contexts. 2) A Lie algebra skeletal representation is formalized following the kinematic body structure, which explicitly encodes the anatomical constraints, and is applicable to a range of articulate objects including but not limited to human body. 3) Our method sets the new state-of-the-art on short-term and long-term motion predictions, and overall provides insights into the challenges in motion context modeling using RNNs. Our implementation can be found on https://github.com/BII-wushuang/Lie-Group-Motion-Prediction.

2. Related Work

Skeleton-based Human Pose Representation Human pose representation is a fundamental problem in computer vision and graphics. Skeleton-based human pose representations have attracted intense attention due to their robustness to viewpoint change, human body scale and motion speed as well as real-time performance [11, 12]. Many existing approaches such as [4, 13] directly utilize raw 3D joint positions to represent human skeleton. [8] follows this line of work but divides the human skeleton into hierarchical body parts, while [5] selects only a subset of most informative joints. [17] and [6] characterize the orientation of a joint by an exponential map introduced in [10]. Displacement based skeletal representation has also been explored either as displacements between pairwise skeletal joints such as in [28] or displacements w.r.t. a global reference joint (hip center) as in [19]. [15] & [27] model the relative geometry between each pair of joints with the Special Euclidean group $SE(3)$.

Motion Prediction Conventional motion prediction approaches have typically utilized shallow models including hidden Markov models [18], Gaussian processes [29], and restricted Boltzmann machine [26] to learn temporal dynamics of human motions. Recently, deep learning based methods have attracted increasing interests due to their superior performance. For instance, [9] presents an Encoder-Recurrent-Decoder network where LSTM is utilized for the recurrent layer and non-linear transformations are incorporated in the encoder and decoder. [21] employs GRU as the RNN unit and estimates the joint velocities instead of directly predicting body pose. [17] represents human body parts as a structured graph of node RNNs linked by edge RNNs. [25] introduces a motion context modeling network using modified highway unit, while [3] develops a probabilistic human motion prediction network employing generative adversarial networks.

Animal Datasets Now, let us consider the other two articulated objects that we investigate in this work, i.e., fish and mouse. They are important model organisms in the life science community and there are increasing interests and efforts in development of visual behavioral analysis toolkits adopting computer vision and machine learning. Existing literature [7, 23, 22] mostly focus on pose estimation and tracking. [14, 30] are two recent work analyzing mice social behaviors, where a mouse is characterized by a straight-line and an ellipse, respectively.

3. Our Approach

Problem Formulation Presented with an observed 3D pose sequence $\langle \mathbf{p}_1, \mathbf{p}_2, \cdots, \mathbf{p}_t \rangle$ of an articulate object, we are interested in predicting its future pose sequence $\langle \mathbf{p}_{t+1}, \mathbf{p}_{t+2}, \cdots, \mathbf{p}_{t+T} \rangle$. Note pose sequences can now be conveniently acquired by commodity motion capture systems or extracted from depth images and videos using pose estimation algorithms (e.g., [24, 32]).

There have been efforts to cast a pose as 3D coordinates of all skeletal joints [4], which in fact regards joints

as independent entities and fails to capture intrinsic geometric constraints. The prediction results may suffer from severe body distortion (shown in Subsection 4.2). More importantly, most existing approaches simply adopt LSTM or GRU, which cannot model motion contexts, especially long-term dependencies, effectively [2, 25].

Method Overview To tackle these problems, we propose an approach that consists of two key components: 1) a hierarchical motion recurrent network (HMR) and 2) a unified Lie algebra representation formalism. Specifically, we develop a Lie algebra representation for articulate objects, which follows the kinematic structure of the body and explicitly encodes the geometric constraints and actual DoFs (degrees of freedom) of individual joints. Then the pose sequences, with each pose represented in the compact Lie algebra space, are fed into the proposed HMR network to model the dynamic evolution of poses.

3.1. Lie Algebra Representation

An articulate object can be characterized as a kinematic tree of rigid bones connected by joints. As depicted in Fig. 1(a), a human full-body is represented by a kinematic tree consisting of five kinematic chains: the spine and the four limbs, with a total of 57 DoFs. Similarly, a fish and a mouse are both represented as a single kinematic chain along the spine with 44 and 12 DoFs respectively.

We utilize the theory of Lie groups [32, 27] to characterize the relative 3D geometry between two successive bones. Given two successive bones b_{i-1} and b_i, their relative geometry is modeled as the 3D rigid transformation (translation and rotation) required to take b_i to the position and orientation of b_{i-1}. Mathematically, 3D rigid transformations are elements of the Special Euclidean group $SE(3)$, which is a Lie group. Therefore, the relative geometry between b_{i-1} and b_i is represented as a point in $SE(3)$, while the entire skeletal pose is represented as a point in $SE(3) \times SE(3) \times \cdots \times SE(3)$ [27], which is a Lie group endowed with a manifold structure. A motion corresponds to a curve on this manifold, and motion prediction amounts to regressing the future curves. However, regression in this curved manifold is non-trivial, hence we map the curves from $SE(3) \times SE(3) \times \cdots \times SE(3)$ to its Lie algebra space. Below we elaborate the mathematical details.

Take the simplified fish kinematic model depicted in Fig. 1(b) as an example, a local coordinate system is attached to each of the bones such that the x-axis is aligned with the bone and the origin is aligned with the start joint of the bone. Descending along the kinematic chain, a 3D rigid transformation relates the local coordinate systems between two successive bones, which is represented as a 4×4 matrix of the form $\begin{pmatrix} \mathrm{R} & \mathbf{t} \\ 0 & 1 \end{pmatrix}$, with R being a 3×3 rotation matrix, and $\mathbf{t}$ a 3D translation vector. This 3D rigid transformation is an element of $SE(3)$. Specifically, a joint with coordinates $\mathbf{x} = (x, y, z)^\intercal$ w.r.t. coordinate system $i+1$ will have coordinates $\mathbf{x}' = (x', y', z')^\intercal$ w.r.t coordinate system i with $\begin{pmatrix} \mathbf{x}' \\ 1 \end{pmatrix} = \begin{pmatrix} \mathrm{R}_i & \mathbf{t}_i \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ 1 \end{pmatrix}$.

Therefore, the entire forward chain pose is naturally represented as the product of rigid transformations. Technically, a skeletal pose is an element in $SE(3) \times SE(3) \times \cdots \times SE(3)$ which is a Lie group. A pose corresponds to a point in this Lie group manifold, while a motion amounts to a curve on the manifold. Being a Lie group, the manifold also comes with its associated tangent space or Lie algebra $\mathfrak{se}(3)$ which possesses a vector space structure, so our familiar linear algebra techniques can work.

Lie algebra $\mathfrak{se}(3)$ The tangent space at the identity of $SE(3)$ is referred to as its Lie algebra $\mathfrak{se}(3)$. The $SE(3) \to \mathfrak{se}(3)$ association is effectuated by the logarithm map [1] $\log$:

$$\begin{pmatrix} \mathrm{R} & \mathbf{t} \\ 0 & 1 \end{pmatrix} \mapsto \xi_\times = \begin{pmatrix} 0 & -\omega_3 & \omega_2 & \nu_1 \\ \omega_3 & 0 & -\omega_1 & \nu_2 \\ -\omega_2 & \omega_1 & 0 & \nu_3 \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$

This has the closed-form solution [20]

$$\omega = \begin{pmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{pmatrix} = \frac{\theta}{2 \sin \theta} \begin{pmatrix} \mathrm{R}(3,2) - \mathrm{R}(2,3) \\ \mathrm{R}(1,3) - \mathrm{R}(3,1) \\ \mathrm{R}(2,1) - \mathrm{R}(1,2) \end{pmatrix}, \qquad (1)$$

where $\theta = \arccos\left(\frac{\mathrm{Tr}(\mathrm{R})-1}{2}\right)$,

$$\nu = \begin{pmatrix} \nu_1 \\ \nu_2 \\ \nu_3 \end{pmatrix} = \left[\frac{(\mathrm{I}_3 - \mathrm{R})\omega_\times + \omega\omega^\intercal}{\|\omega\|}\right]^{-1} \mathbf{t}.$$

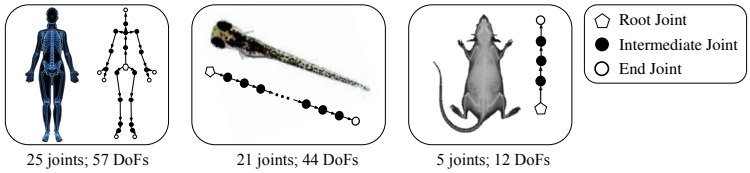
$\xi_\times$ is conveniently mapped to a vector form $\xi = \begin{pmatrix} \omega \\ \nu \end{pmatrix}$.

The above formalizes the procedure of recasting a skeletal pose as a $\mathfrak{se}(3)$ parameterized vector, $\mathbf{p} = \left(\xi_1^{1\intercal}, \cdots, \xi_{m_1}^{1\ \intercal}, \cdots, \xi_1^{K\intercal}, \cdots, \xi_{m_K}^{K\ \intercal}\right)^\intercal$. K denotes the number of kinematic chains, m_k the number of joints in chain k, and ξ_i^k the Lie algebra parameter vector of joint i in chain k. Considering the bone length invariance, the 3 translational DoFs for all bones except the first bone are fixed. If a bone is anatomically constrained to rotate along one axis, our scheme obtains zero variance of the rotational parameters along the other two axis. This nails down on the exact DoF of the joint and we explicitly encode this by fixing its non-rotational elements as constants.

3.2. Hierarchical Motion Context Modeling

The future pose prediction problem can now be formulated as follows: Given as input the sequence of Lie-algebra

[1] The inverse transformation is given by the exponential map $\exp : \xi_\times \in \mathfrak{se}(3) \mapsto \begin{pmatrix} \mathrm{R} & \mathbf{t} \\ 0 & 1 \end{pmatrix} \in SE(3)$.



(a) (b)

Figure 1: (a) A display of the three articulate objects, with their respective joints and skeletons. The first bones of the skeletons are of 6 DoFs, while all other bones in the fish or mouse skeleton have 2 DoFs. The first bone amounts to the bone located in the spine and starts from the root joint. (b) An illustration of a simplified fish kinematic chain. b_i and J_k stands for the i^{th} bone and k^{th} joint, respectively. Each bone is assigned with a local coordinate system describing its rigid transformation relative to its parent (preceding) bone, the sequence of rigid transformations characterizes a pose. Specifically, the rigid transformation of the first bone is relative to the global coordinate system.

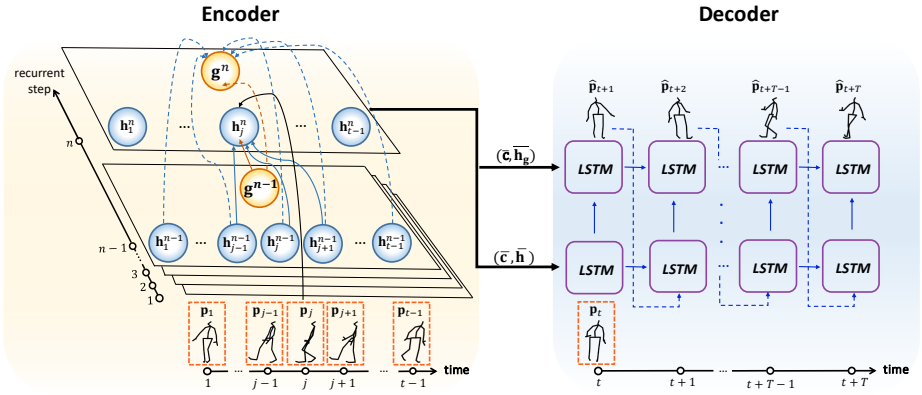


Figure 2: The proposed neural network unfolded over recurrent steps. Local hidden state $\mathbf{h}_j^n$ is updated as a function of $\mathbf{h}_{j-1}^{n-1}, \mathbf{h}_j^{n-1}, \mathbf{h}_{j+1}^{n-1}, \mathbf{g}^{n-1}, \mathbf{c}_j^{n-1}$, and $\mathbf{p}_j$. Global state $\mathbf{g}^n$ is updated as a function of $\mathbf{g}^{n-1}$ and $\mathbf{h}_1^{n-1}, \cdots, \mathbf{h}_{t-1}^{n-1}$.

parameterized poses, $\langle \mathbf{p}_1, \cdots, \mathbf{p}_t \rangle$, generate predictions for $\langle \mathbf{p}_{t+1}, \cdots, \mathbf{p}_{t+T} \rangle$. In conventional motion prediction models, the encoder and decoder usually consist of single or stacked layers of LSTM or GRU cells. Poses are input successively into the encoder cells to model motion contexts into hidden states. The inputs must be processed sequentially and the final hidden state is largely affected by the inputs at recent frames [25], which cannot properly capture long-term dependencies [2]. To avoid this issue, we consider a new encoder-decoder architecture, where a Hierarchical Motion Recurrent (HMR) network is proposed as the encoder, and the entire input sequence of poses is fed one-shot instead of successively. Motion contexts are jointly modeled by a hierarchical state $\mathbf{S}$ consisting of local states $\mathbf{h}_j$ for individual frames and an overall sequence-level state $\mathbf{g}$. At each recurrent step n, the j^{th} frame updates its motion context $\mathbf{h}_j^n$ by exchanging information with its neighboring local states $\mathbf{h}_{j+1}^n$ and $\mathbf{h}_{j-1}^n$, as well as with the global state $\mathbf{g}^n$. As the number of recurrent steps increases, the number of frames that have information exchange with $\mathbf{h}_j^n$ becomes larger, which enriches state representations incrementally.

Fig. 2 illustrates the proposed encoder network unfolded over recurrent steps. At recurrent step 0, the network is initialized with $\mathbf{h}_j^0 = \mathbf{c}_j^0 = W\mathbf{p}_j + \mathbf{b}$ and $\mathbf{g}^0 = \mathbf{c}_g^0 = \frac{1}{t-1}\sum_{j=1}^{t-1} \mathbf{h}_j^0$, where $\mathbf{c}_j^0$ and $\mathbf{c}_g^0$ respectively denote the cell states of $\mathbf{h}_j^0$ and $\mathbf{g}^0$. Matrix W and vector $\mathbf{b}$ are network parameters. Subsequently, at each recurrent step n, the state transition process is performed to update $\mathbf{h}_j^n, \mathbf{g}^n, \mathbf{c}_j^n, \mathbf{c}_g^n$ as functions of $\mathbf{h}_j^{n-1}, \mathbf{g}^{n-1}, \mathbf{c}_j^{n-1}, \mathbf{c}_g^{n-1}$. Fig. 3 illustrates the one-step state transition process with equations formulating the process and figures visualizing it.

Update frame-level state $\left(\mathbf{h}_j^n, \mathbf{c}_j^n\right)$ As illustrated in the left panel of Fig. 3, at recurrent step n, $\mathbf{h}_j^{n-1}$ is updated (to $\mathbf{h}_j^n$) by exchanging information with $\mathbf{h}_{j-1}^{n-1}, \mathbf{h}_{j+1}^{n-1}$, and $\mathbf{g}^{\mathbf{n-1}}$. There are a total of 4 types of *forget* gates: $\mathbf{f}^n, \mathbf{l}^n, \mathbf{r}^n$, and $\mathbf{q}^n$ (forward, left, right, and global forget gates), which respectively control the information flows from the current cell state $\mathbf{c}_j^{n-1}$, left cell state $\mathbf{c}_{j-1}^{n-1}$, right cell state $\mathbf{c}_{j+1}^{n-1}$, and global cell state $\mathbf{c}_g^{n-1}$ to the final cell state $\mathbf{c}_j^n$. The *input* gate $\mathbf{i}^n$ controls the information flow from the pose input $\mathbf{p}_j$. Finally, the j^{th} frame hidden state $\mathbf{h}_j^n$ is obtained by a Hadamard product of the *output* gate $\mathbf{o}_j^n$ with the $\tanh$

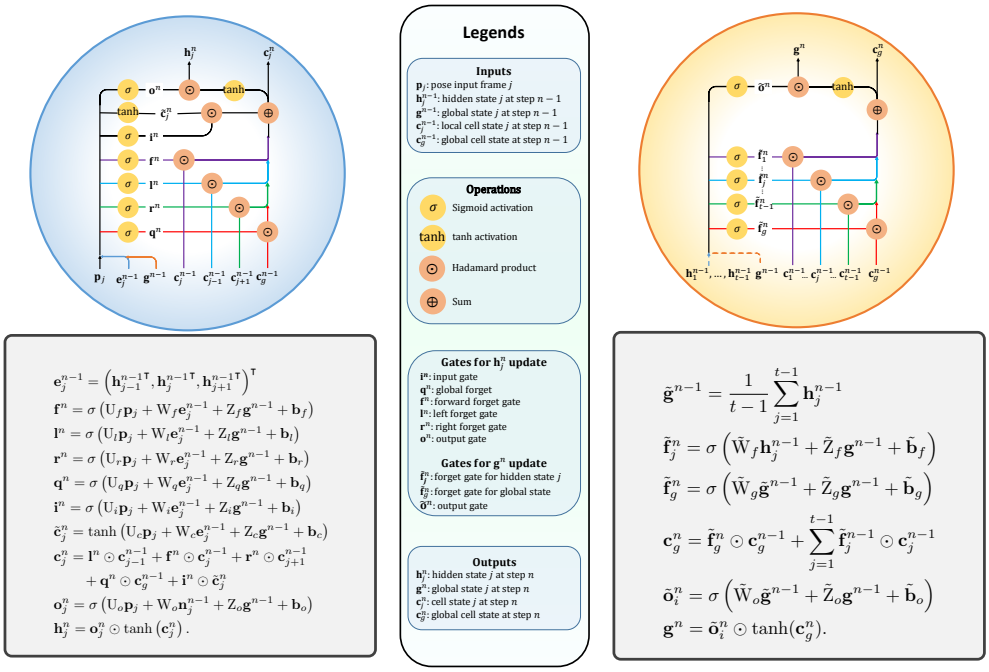


Figure 3: The left panel shows the update process of frame-level state $\left(\mathbf{h}_j^n, \mathbf{c}_j^n\right)$, the right panel shows the update process of sequence-level state $\left(\mathbf{g}^n, \mathbf{c}_g^n\right)$. The equations in the two panels formulate the process while the figures visualize the gates.

activated cell state $\mathbf{c}_j^n$. Matrices U_k, W_k, Z_k and biases $\mathbf{b}_k$ are parameters to be learned where $k \in \{f, l, r, q, i, o\}$.

Update sequence-level state $\left(\mathbf{g}^n, \mathbf{c}_g^n\right)$ The update process from $\mathbf{g}^{n-1}$ to $\mathbf{g}^n$ is demonstrated in the right panel of Fig. 3. $\tilde{\mathbf{f}}_g^n$ and $\tilde{\mathbf{f}}_j^n$ are the respective *forget* gates that filter information from $\mathbf{c}_g^{n-1}$ and $\mathbf{c}_j^{n-1}$ to global cell state $\mathbf{c}_g^n$. The global state $\tilde{\mathbf{g}}^n$ is obtained by a Hadamard product of the *output* gate $\tilde{\mathbf{o}}_j^n$ with the $\tanh$ activated $\mathbf{c}_g^n$. Matrices $\tilde{W}_k, \tilde{Z}_k$ and biases $\tilde{\mathbf{b}}_k$ with index $k \in \{g, f, o\}$ are the parameters to be learned.

Encoder & Decoder In the proposed HMR approach, our encoder learns a two-level representation of the entire input sequence. It is subsequently passed to the decoder that recursively outputs the future motion sequence. As displayed in Fig. 2, our decoder engages a two-layer stacked LSTM network. For both layers of the decoder, the cell state input is $\overline{\mathbf{c}} = \frac{1}{t-1}\sum_{j=1}^{t-1}\mathbf{c}_j^n$, namely the average over all frame-level cell states at the final recurrent step n. In particular, for the first layer, its hidden state input is set as $\overline{\mathbf{h}} = \frac{1}{t-1}\sum_{j=1}^{t-1}\mathbf{h}_j^n$. Similarly, the hidden state input of the second layer is configured as $\overline{\mathbf{h_g}} = \frac{1}{t}\left(\sum_{j=1}^{t-1}\mathbf{h}_j^n + \mathbf{g}^n\right)$. Finally, the pose $\mathbf{p}_t$ at time t serves as the initial input pose to the decoder. The decoder is executed following the directed links shown in Fig. 2, producing pose predictions in a recursive manner.

Loss function Given a kinematic chain of m joints with prescribed Lie algebra pose $\mathbf{p} = (\xi_1^\intercal, \cdots, \xi_m^\intercal)^\intercal$, the location of joint $\mathbf{J}_i$ can be obtained by forward kinematics

$$\begin{pmatrix}\mathbf{J}_i\\1\end{pmatrix} = \left[\prod_{j=1}^{i}\exp(\xi_{j\times})\right]\begin{pmatrix}\mathbf{0}\\1\end{pmatrix}. \tag{2}$$

Existing works such as [17, 21] adopted a simple L2 loss function for training, which unfortunately treats all joints equally, and ignores this important kinematic chain hierarchy. One immediate consequence is that Lie algebraic parameter estimation errors will accumulate rapidly down the chain. To account for this, we propose the following loss:

$$\mathrm{Loss}(\mathbf{p}, \hat{\mathbf{p}}) = \sum_{i=1}^{m-1}(m-i)l_i\|\xi_i - \hat{\xi}_i\|_2. \tag{3}$$

where $\hat{\mathbf{p}} = \left(\hat{\xi}_1^\intercal, \cdots, \hat{\xi}_m^\intercal\right)^\intercal$ denotes the predicted pose, and l_i denotes the length of bone i. Now higher losses would be incurred if there are errors in the preceding joints of a chain. As will be illustrated in Subsection 4.5, this setting improves prediction performance.

4. Experiments

4.1. Experimental Settings

Datasets Experiments are conducted on three large and complex datasets of distinct articulate objects, namely human, fish, and mouse. For human, the 3D human full-body motion dataset H3.6m [16] is used. H3.6m contains

3.6 million 3D human poses with 15 activities performed by 7 subjects. Following existing works [25, 17], we downsampled the motion sequence by 2 to 25 frames per second (FPS). For animals, we consider the fish and mouse datasets of [32], which contain 14 fish videos (50 FPS) of 6 different fish, and 8 mouse videos (25 FPS) of 4 lab mice. In general, the continuous sequences in these videos vary from 2,250 frames to 24,000 frames. For all datasets, comparison with existing methods were done with pose sequences parameterized using our Lie algebra representation.

Parameter Settings The hidden state size, i.e. length of state vectors $\mathbf{h}$ and $\mathbf{g}$ is set to 300, 800, and 100 respectively for human, fish, and mouse motion prediction. All other settings and hyperparameters are constant across different objects. The default number of recurrent steps is set to 10 and neighboring context window size 3. Following previous works [21, 25], we do not model global translation and utilize $t = 50$ observed frames as inputs to predict future $T = 10$ frames in training. The Adam optimizer is employed with an intial learning rate of 0.001 which decays by 10% every 10,000 iterations. A batch size of 16 is used and the gradient clipping threshold is set to 5.

4.2. Evaluation on H3.6m dataset

First, we benchmark our approach against state-of-the-art methods on the H3.6m dataset [16] with the mean angle error (MAE) metric adopted in previous works [17, 25, 21].

In Table 1, the performance of different methods are presented in terms of MAE for 4 complex activities, namely "Discussion", "Greeting, "Posing" and "Walking Dog". A total of 10 methods are compared, including ERD [9], LSTM-3LR [9], SRNN [17], Res-GRU [21], zero-velocity [25], MHU [25], our HMR network, and 3 variants of HMR. Zero-velocity is a baseline that constantly outputs the last observed pose. We reproduced the existing methods following their released codes on GitHub[2]. For all methods, training is done over all activity types with a training output window size of $T = 10$ frames. From the quantitative results in Table 1, the first observation is that our HMR network delivers state-of-the-art results for both short-term and long-term predictions on complex activities. Interestingly, many existing methods, such as ERD and SRNN, tend to be outperformed by the simple baseline zero-velocity, which reconfirms the findings reported in [21]. We can also observe that prediction error increases as we predicts deeper into the future. We further conducted full experiments on all other 11 activities, which reconfirms that our methods consistently outperforms existing methods. We put them in the supplementary files due to space limitations.

Besides quantitative evaluation, we further compare the performance of state-of-the-art methods visually. Exemplar visual results for long-term forecasting on the walking activity are demonstrated in Fig. 4, where the predicted poses for future 50 seconds (1,250 frames) are presented. A XYZ baseline is engaged in the comparison, which employs 3 stacked LSTM layers as the encoder and uses raw 3D joint coordinates instead of Lie algebra parameters as inputs. Here, training for all methods is done over a single activity type for a longer training output window size of $T = 100$ frames. Our insights are that it is unrealistic to expect accurate forecasting in the long-term and a more reasonable goal is to achieve human-like motion. As shown in Fig. 4 as well as in the supplementary video, LSTM-3LR converges to a motionless state within 1 sec; ERD exhibits jittery (nonsmooth) and unrealistic motion; Res-GRU converges to a motionless pose after 5 sec. XYZ yields good short term predictions but suffers from bone length deformation (e.g., longer or shorter arms), leading to horrendous predictions in the long term. HMR is capable of producing natural pose predictions throughout the entire forecast window. In this regard, an important highlight of our architecture is the capability to generate long-term natural, bone-length-invariant, and human-like motions.

For raw 3D joint coordinates representation as inputs, besides the XYZ baseline presented, we also tried other encoders such as those in Res-GRU [21], ERD [9], and HMR. Empirical evidences demonstrate similar severe body distortions in long-term prediction. Slight deformation at earlier predictions propagate out of control, resulting in extreme deterioration in long term performance. This suggests the limitations (or challenges) of raw coordinates representation and the necessity of explicitly encoding the kinematic body structure and anatomical constraints.

4.3. Evaluation on Fish and Mouse datasets

Whereas the human dataset poses the challenge of having to model multiple kinematic chains simultaneously, the fish and mouse datasets of [32] raise different issues such as 1) long kinematic chain of 21 joints for fish and 2) stochastic nature of the animal motions and lack of activity type classification in the datasets.

For both datasets, we evaluate the performance of state-of-the-art methods with quantitative results reported in Table 2. HMR consistently and significantly outperforms other methods on both datasets.

It would be more instructive to look at the visual results. A sample forecasting result for the fish dataset is shown in Fig. 5. The long kinematic chain in the fish skeletal anatomy resulted in modeling difficulties for the competing methods. In both ERD and Res-GRU, the predicted fish pose demonstrate unnatural distortions and a zigzagged contour, which is especially the case for ERD. LSTM-3LR suffers from the issue of quickly converging to a motionless state. In contrast, HMR retains streamlined shapes and the curvature of

[2]The implementation of [25] is not available and we report the experimental results in their paper.

Methods	Discussion								Greeting							
	80ms	160ms	320ms	400ms	560ms	640ms	720ms	1000ms	80ms	160ms	320ms	400ms	560ms	640ms	720ms	1000ms
ERD [9]	2.22	2.38	2.58	2.69	2.89	2.93	2.94	3.11	1.70	2.04	2.60	2.81	3.29	3.47	3.55	3.43
LSTM-3LR [9]	1.80	2.00	2.13	2.13	2.29	2.32	2.36	2.44	0.93	1.51	2.27	2.54	2.97	3.05	3.12	3.09
SRNN [17]	1.16	1.40	1.75	1.85	2.06	2.07	2.08	2.19	1.33	1.60	1.83	1.98	2.27	2.28	2.30	2.31
Res-GRU [21]	0.31	0.69	1.03	1.12	1.52	1.61	1.70	1.87	**0.52**	0.86	1.30	1.47	1.78	1.75	1.82	1.96
Zero-velocity [21]	0.31	0.67	0.97	1.04	1.41	1.56	1.71	1.96	0.54	0.89	1.30	1.49	1.79	1.74	1.77	1.80
MHU [25]	0.31	0.66	0.93	1.00	1.37	1.51	1.66	1.88	0.54	0.87	1.27	1.45	1.75	1.71	1.74	1.87
HMR (Remove $\mathbf{l}$, $\mathbf{r}$ from update of $\mathbf{h}$)	0.30	0.59	0.93	1.02	1.42	1.56	1.67	1.80	0.54	0.89	1.30	1.44	1.68	1.66	1.70	1.85
HMR (Remove $\mathbf{g}$)	0.30	0.57	0.87	0.96	1.38	1.54	1.69	1.89	0.53	0.86	1.28	1.45	1.71	1.70	1.76	1.95
HMR (Remove 2nd decoder layer)	0.30	0.60	0.94	1.03	1.40	1.55	1.69	1.86	0.57	0.92	1.33	1.49	1.75	1.79	1.82	1.88
HMR	**0.29**	**0.55**	**0.83**	**0.94**	**1.35**	**1.49**	**1.61**	**1.72**	**0.52**	**0.85**	**1.25**	**1.40**	**1.65**	**1.62**	**1.67**	**1.73**
Methods	Posing								Walking Dog							
	80ms	160ms	320ms	400ms	560ms	640ms	720ms	1000ms	80ms	160ms	320ms	400ms	560ms	640ms	720ms	1000ms
ERD [9]	2.42	2.77	3.26	3.39	3.43	3.42	3.45	3.87	1.58	1.78	2.02	2.10	2.31	2.37	2.48	2.60
LSTM-3LR [9]	1.22	1.89	3.02	3.53	4.25	4.57	4.83	4.60	0.76	1.29	1.91	2.18	2.72	3.01	3.30	3.78
SRNN [17]	1.74	1.89	2.23	2.43	2.67	2.73	2.79	3.42	1.57	1.73	1.93	1.96	2.13	2.17	2.23	2.20
Res-GRU [21]	0.41	0.84	1.53	1.81	2.06	2.21	2.24	2.53	0.56	0.95	1.33	1.48	1.78	1.81	1.88	1.96
Zero-velocity [21]	0.28	0.57	1.13	**1.38**	1.81	2.14	2.23	2.78	0.60	0.98	1.36	1.50	1.74	1.80	1.87	1.96
MHU [25]	0.33	0.64	1.22	1.47	1.82	2.11	2.17	2.51	0.56	0.88	1.21	1.37	1.67	1.72	1.81	1.90
HMR (Remove $\mathbf{l}$, $\mathbf{r}$ from update of $\mathbf{h}$)	0.27	0.56	1.22	1.52	1.76	1.91	2.08	2.60	0.56	0.89	1.33	1.49	1.73	1.82	1.90	2.00
HMR (Remove $\mathbf{g}$)	0.25	0.54	1.19	1.48	1.93	2.10	2.23	2.65	0.56	**0.87**	1.23	1.42	1.84	1.90	1.94	2.06
HMR (Remove 2nd decoder layer)	0.30	0.59	1.26	1.49	1.87	2.04	2.20	2.66	0.60	0.90	1.24	1.48	1.79	1.86	1.94	2.07
HMR	**0.24**	**0.53**	**1.12**	1.42	**1.75**	**1.89**	**2.02**	**2.50**	**0.55**	**0.87**	**1.20**	**1.36**	**1.65**	**1.70**	**1.77**	**1.84**

Table 1: Performance evaluation (in MAE) of comparison methods over 4 different action types on the H3.6m dataset.

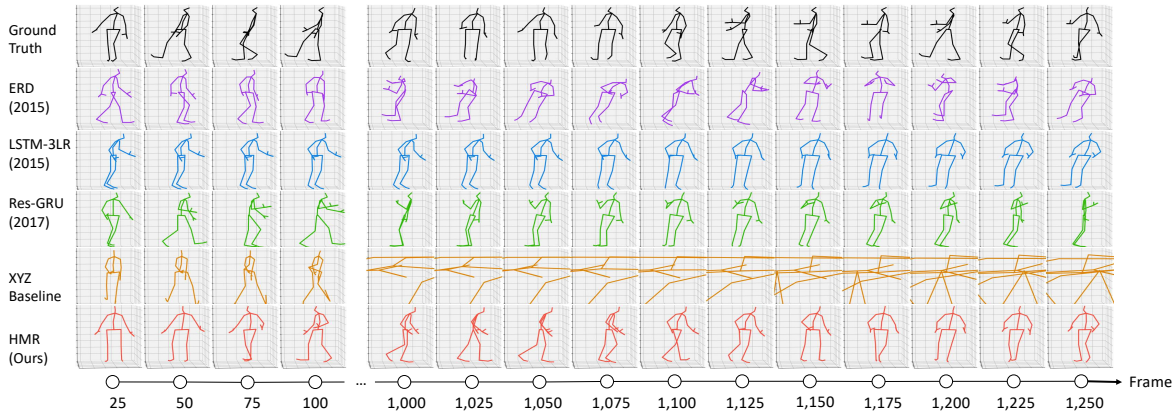


Figure 4: Long-term motion forecasting of walking activity by the comparison methods on the H3.6m dataset. 25 frames correspond to 1 sec. The complete visual results can be found in the supplementary video file.

Methods	Fish								Mouse							
	80ms	160ms	320ms	400ms	560ms	640ms	720ms	1000ms	80ms	160ms	320ms	400ms	560ms	640ms	720ms	1000ms
ERD [9]	0.62	0.59	0.54	0.69	0.79	0.85	0.87	1.20	0.77	0.62	0.67	0.77	0.86	0.83	0.88	0.91
LSTM-3LR [9]	0.91	0.59	0.45	0.39	0.25	0.26	0.30	0.29	0.68	0.61	0.81	0.84	0.85	0.81	0.85	0.80
Res-GRU [21]	0.52	0.56	0.52	0.39	0.26	0.25	0.26	0.26	0.40	0.48	0.66	0.70	0.74	0.71	0.72	0.74
HMR (ours)	**0.40**	**0.48**	**0.44**	**0.28**	**0.13**	**0.12**	**0.13**	**0.11**	**0.39**	**0.44**	**0.56**	**0.63**	**0.69**	**0.68**	**0.67**	**0.69**

Table 2: Performance evaluation (in MAE) of the comparison methods for the Fish and Mouse datasets of [32].

the predicted pose remains smooth and natural.

The mouse motion is highly stochastic, leading to difficulties in accurate forecasting. Compared methods all tend to converge to motionless states. A sample forecasting sequence on the mouse dataset is displayed in Fig. 6. ERD, LSTM-3LR and Res-GRU converge to motionless states after 30, 15, 45 frames respectively. HMR remains in motion throughout with fairly accurate prediction for the first 40 frames. In particular, it can be seen that the HMR predicted mouse body orientation remains in alignment with the ground truth whereas compared methods all wrongly predict the mouse orientation.

4.4. Computational Efficiency

The training and testing time as well as number of training parameters required for different methods are shown in Table 3. Our architecture is implemented using TensorFlow 1.8 [1]. All experiments were performed on a Nvidia GeForce GTX TITAN X GPU. In brief, HMR requires less parameters than existing methods, and its computation speed is significantly faster.

4.5. Loss Function Study

In this subsection we compare our loss function presented in Eq. (3) against conventional L2 loss. In addition

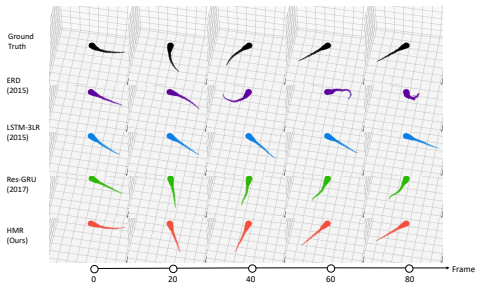


Figure 5: Motion forecasting on Fish dataset. The head of the fish is rendered wider (for resemblance with the actual zebrafish).

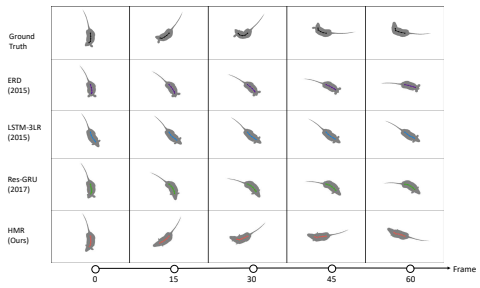


Figure 6: Motion forecasting on Mouse dataset. The mouse shape is rendered in gray color with joints marked out along the spine.

Methods	# Parameters	Train time / 1,000 iterations (s)	Test FPS
ERD	17,348,054	428	52.4
LSTM-3LR	20,831,054	632	33.7
SRNN	22,817,888	947	14.4
Res-GRU	6,684,726	65	173.5
HMR (Ours)	**4,422,654**	**33**	**406.1**

Table 3: Number of training parameters and efficiency.

to the MAE metric used, we also use the mean joint error (MJE) as a supplementary metric. Experiments were conducted on H3.6m dataset with results averaged over all 15 activities. In the experiments, we kept the network structure fixed as HMR, while using different losses. As shown in Table 4, the proposed loss consistently outperforms the L2 loss. By respecting the hierarchical nature of kinematic chains, our proposed loss reduces errors in the root joint predictions, which translates to a better estimate of the global orientation and thus significant improvements on the MJE metric. Experimental results on animals (see Subsection 4.3) also indicate that employing the proposed loss function (in our work) instead of the L2 loss (employed in compared works) is more successful in capturing the anatomical features of the skeleton and its motion dynamics, such as better prediction of the mouse orientation.

Time(ms)		80	160	320	400	560	640	1000
MAE	L2	0.34	0.57	0.87	0.96	1.15	1.24	1.40
	Our	**0.33**	**0.55**	**0.83**	**0.93**	**1.13**	**1.21**	**1.36**
MJE	L2	67.6	78.3	83.1	86.4	97.6	105.7	113.4
	Our	**9.3**	**17.1**	**28.1**	**32.7**	**40.9**	**43.5**	**46.4**

Table 4: Comparison of our loss function against L2 loss in terms of MAE and MJE, respectively.

4.6. Extensions and Reductions in HMR Network

By default in HMR the number of neighboring context window size is fixed to 3. As the number of steps increases, the number of local states that have information exchange with $\mathbf{h}_j^n$ becomes larger. It is interesting to see the effect of enlarging or reducing neighboring window size, which we report in Table 5. We observe that enlarging the neighboring context window size does not necessarily leads to improvement in accuracy. The optimal number of recurrent steps and hidden state size on the H3.6m dataset is also empirically determined. This allows us to settle on the default hidden state size 300 and number of recurrent steps 10.

Hidden Size	Val. Loss	Rec. Steps	Val. Loss	Neigh. Win.	Val. Loss
100	0.177	1	0.169	1	0.169
200	0.165	5	0.162	**3**	**0.151**
300	**0.151**	**10**	**0.151**	5	0.155
400	0.172	15	0.159	7	0.159
500	0.175	20	0.161	11	0.158

Table 5: Validation Loss on H3.6m on varying internal parameters, hidden states size, number of recurrent steps n, and context window size.

We also report 3 variants of our HMR network: 1) Removal of the left and right forget gates in the update of the frame level states $\mathbf{h}_j$ from the encoder; 2) Removal of the sequence level state $\mathbf{g}$ from the encoder; and 3) Removal of the second decoder layer in Table 1. Removal of the sequence level state $\mathbf{g}$ from our HMR encoder is observed not to severely affect forecasting before 400 ms (i.e. 10 frames) but performance beyond 400 ms declines noticeably. Prediction accuracy also suffered upon removal of the left and right forget gates or the removal of the second decoder layer.

5. Conclusion

We have proposed a hierarchical motion recurrent network, which can effectively model motion contexts and significantly surpasses existing work in both short-term and long-term motion predictions. The proposed network incorporates our Lie algebra representation naturally preserves the skeletal articulation of the underlying objects. Extensive results on human, fish and mouse datasets demonstrate the competency of our approach. Future work includes further investigation into group-level motion predictions.

6. Acknowledgments

This paper is partly supported by the National Key R&D Program of China (No. 2017YFB1401304), the Natural Science Foundation of Zhejiang Province, China (Grant No. LQ19F020001), the JCO grants of A*STAR Singapore, and the support from University of Alberta-Huawei Joint Innovation Center in Canada. We thank Yunphant Ltd. for their constructive suggestions.

References

[1] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, et al. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, pages 265–283, 2016. 7

[2] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473, 2014. 2, 3, 4

[3] E. Barsoum, J. Kender, and Z. Liu. HP-GAN: probabilistic 3d human motion prediction via GAN. *CoRR*, abs/1711.09561, 2017. 2

[4] J. Butepage, M. Black, D. Kragic, and H. Kjellstrom. Deep representation learning for human motion prediction and classification. In *CVPR*, 2017. 1, 2

[5] A. A. Chaaraoui, J. R. Padilla-López, P. Climent-Pérez, and F. Flórez-Revuelta. Evolutionary joint selection to improve human action recognition with RGB-D devices. *ESWA*, 41(3):786–794, 2014. 2

[6] K. Cho, B. van Merrienboer, Ç. Gülçehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *EMNLP*, pages 1724–1734, 2014. 2

[7] P. Dollár, P. Welinder, and P. Perona. Cascaded pose regression. In *CVPR*, pages 1078–1085, 2010. 2

[8] Y. Du, W. Wang, and L. Wang. Hierarchical recurrent neural network for skeleton based action recognition. In *CVPR*, pages 1110–1118, 2015. 2

[9] K. Fragkiadaki, S. Levine, P. Felsen, and J. Malik. Recurrent network models for human dynamics. In *ICCV*, 2015. 1, 2, 6, 7

[10] F. S. Grassia. Practical parameterization of rotations using the exponential map. *J. Graphics, GPU, & Game Tools*, 3(3):29–48, 1998. 2

[11] F. Han, B. Reily, W. Hoff, and H. Zhang. Space-time representation of people based on 3d skeletal data: A review. *CVIU*, 158:85–105, 2017. 2

[12] D. Hoiem and S. Savarese. *Representations and Techniques for 3D Object Recognition and Scene Interpretation*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers, 2011. 2

[13] D. Holden, J. Saito, and T. Komura. A deep learning framework for character motion synthesis and editing. *ACM Trans. Graph.*, 35(4):138:1–138:11, 2016. 2

[14] W. Hong, A. Kennedy, X. Burgos-Artizzu, M. Zelikowsky, S. Navonne, P. Perona, and D. Anderson. Automated measurement of mouse social behaviors using depth sensing, video tracking, and machine learning. *PNAS*, 112(38):E5351–E5360, 2015. 2

[15] Z. Huang, C. Wan, T. Probst, and L. V. Gool. Deep learning on lie groups for skeleton-based action recognition. In *CVPR*, pages 1243–1252, 2017. 2

[16] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE TPAMI*, 36(7):1325–39, 2014. 5, 6

[17] A. Jain, A. R. Zamir, S. Savarese, and A. Saxena. Structural-rnn: Deep learning on spatio-temporal graphs. In *CVPR*, pages 5308–5317, 2016. 1, 2, 5, 6, 7

[18] A. M. Lehrmann, P. V. Gehler, and S. Nowozin. Efficient nonlinear markov models for human motion. In *CVPR*, pages 1314–1321, 2014. 1, 2

[19] J. Luo, W. Wang, and H. Qi. Group sparsity and geometry constrained dictionary learning for action recognition from depth maps. In *ICCV*, pages 1809–1816, 2013. 2

[20] Y. Ma, S. Soatto, J. Koseck, and S. Sastry. *An Invitation to 3-D Vision: From Images to Geometric Models*. Springer, 2010. 3

[21] J. Martinez, M. Black, and J. Romero. On human motion prediction using recurrent neural networks. In *CVPR*, 2017. 1, 2, 5, 6, 7

[22] A. Mathis, P. Mamidanna, K. M. Cury, T. Abe, V. N. Murthy, M. W. Mathis, and M. Bethge. Deeplabcut: markerless pose estimation of user-defined body parts with deep learning. Technical report, Nature Publishing Group, 2018. 2

[23] G. Salem, J. Krynitsky, M. H. H. III, T. Pohida, and X. P. Burgos-Artizzu. Cascaded regression for 3d pose estimation for mouse in fisheye lens distorted monocular images. In *2016 IEEE Global Conference on Signal and Information Processing, GlobalSIP 2016, Washington, DC, USA, December 7-9, 2016*, pages 1032–1036, 2016. 2

[24] J. Shotton, R. Girshick, A. Fitzgibbon, T. Sharp, M. Cook, M. Finocchio, R. Moore, P. Kohli, A. Criminisi, A. Kipman, and A. Blake. Efficient human pose estimation from single depth images. *TPAMI*, 35(12):2821–2840, 2013. 2

[25] Y. Tang, L. Ma, W. Liu, and W. Zheng. Long-term human motion prediction by modeling motion context and enhancing motion dynamic. *CoRR*, abs/1805.02513, 2018. 1, 2, 3, 4, 6, 7

[26] G. W. Taylor, G. E. Hinton, and S. T. Roweis. Modeling human motion using binary latent variables. In *NIPS*, pages 1345–1352, 2006. 1, 2

[27] R. Vemulapalli, F. Arrate, and R. Chellappa. Human action recognition by representing 3d skeletons as points in a lie group. In *CVPR 2014*, pages 588–595, 2014. 2, 3

[28] J. Wang, Z. Liu, Y. Wu, and J. Yuan. Mining actionlet ensemble for action recognition with depth cameras. In *CVPR*, pages 1290–1297, 2012. 2

[29] J. M. Wang, D. J. Fleet, and A. Hertzmann. Gaussian process dynamical models. In *NIPS*, pages 1441–1448, 2005. 1, 2

[30] A. Wiltschko, M. Johnson, G. Iurilli, R. Peterson, J. Katon, S. Pashkovski, V. Abraira, R. Adams, and S. Datta. Mapping sub-second structure in mouse behavior. *Neuron*, 88(6):1121–1135, 2015. 2

[31] J. Xie, S. Zhu, and Y. N. Wu. Synthesizing dynamic patterns by spatial-temporal generative convnet. In *CVPR*, pages 1061–1069, 2017. 1

[32] C. Xu, L. Govindarajan, Y. Zhang, and L. Cheng. Lie-X: Depth image based articulated object pose estimation, tracking, and action recognition on Lie groups. *IJCV*, 123(3):454–78, 2017. 2, 3, 6, 7