Abstract

Predicting human motion from a historical pose sequence is at the core of many applications in computer vision. Current state-of-the-art methods concentrate on learning motion contexts in the pose space, however, the high dimensionality and complex nature of human pose invoke inherent difficulties in extracting such contexts. In this paper, we instead advocate to model motion contexts in the joint trajectory space, as the trajectory of a joint is smooth, vectorial, and gives sufficient information to the model. Moreover, most existing methods consider only the dependencies between skeletal connected joints, disregarding prior knowledge and the hidden connections between geometrically separated joints. Motivated by this, we present a semi-constrained graph to explicitly encode skeletal connections and prior knowledge, while adaptively learn implicit dependencies between joints.

We also explore the applications of our approach to a range of objects including human, fish, and mouse. Surprisingly, our method sets the new state-of-the-art performance on 4 different benchmark datasets, a remarkable highlight is that it achieves a 19.1% accuracy improvement over current state-of-the-art in average. To facilitate future research, we have released our code at https://github.com/Pose-Group/MPT.

1. Introduction

The ability for machines to anticipate and model human motion dynamics is very much coveted [29] in a wide range of applications such as autonomous driving, human tracking, and regulating the response of a robot when interacting with humans. As a result, future motion prediction has attracted considerable attention in the past decade [9, 41, 46, 6, 39].

Whereas existing methods achieve accurate prediction for a few immediate future frames, it is still difficult to expect accurate and natural forecasting in the long-term since the information hidden in the conscious activity of a human is complex and high-dimensional [15]. To tackle the challenge, we seek to reduce the complexity of motion context modeling at the base level, i.e. representation space level, and capture long range dependencies to yield accurate and natural prediction on both short-term and long-term.

Fundamentally, human motion prediction aims to learn a mapping function that bridges the historical skeleton pose sequence to the future pose sequence. Pioneering approaches adopt Gaussian Processes [40], Hidden Markov Models [20], and Restricted Boltzmann Machine [38], to predict future human skeleton poses. Unfortunately, these models impose strong assumptions such as Gaussian distributions on the motion dynamics, leading to unsatisfactory results.

Recent approaches explored using different sorts of deep neural networks to address the issue [24, 16, 36, 23, 19, 4, 45, 1, 35]. One line of work [8, 12, 13, 22] utilized recur-
The proposed trajectories representation has the crucial advantage of being smooth and low dimensional.

Another severe limitation of existing works is that they consider only the connectivity between adjacent joints while ignoring the movement coordination between geometrically separated joints. Dissevering these additional cues results in insufficient context modeling and inaccurate prediction. To address this issue, [32] incorporates dense connections between each pair of joints, [7] engages in a dynamic graph, and [22] adopts a multiscale graph to model the relations. However, the problem is still not efficiently addressed and useful prior knowledge, such as limb mirror symmetry tendency (e.g., symmetry tendency between two arms) and cross sides synchronization tendency (e.g., synchronization tendency between left arm and right leg), are ignored. In this paper, we propose a new graph convolutional network that uses a semi-constrained graph to explicitly encode skeletal connection and useful prior knowledge, while adaptively learn flexible connections between joints. We would like to highlight that the proposed convolutional network has an edge in adopting efficient matrix operations and maintaining constraints that facilitate the training.

Interestingly, most existing methods typically focus on 3D human motion prediction. In this paper, we explore applying our approach to a range of objects including human, fish, and mouse. Extensive experiments are conducted on large benchmark datasets including H3.6M and CMU MoCap, and on animal datasets that involve motions of fish and mouse. Empirically, our approach outperforms state-of-the-art methods by a large margin (more than 19.1% accuracy gain) in both short-term and long-term motion predictions. Our code is released, hoping to inspire future research.

**Contributions** To summarize, the key contributions of this paper are: 1) A new motion representation is proposed, which models motion contexts in the trajectory space in stead of the traditional pose space. 2) A semi-constrained graph convolution network is presented to comprehensively learn the relationships between joints, which simultaneously considers skeletal connection, prior knowledge, and implicit dependencies between joints. 3) Our method sets the new state-of-the-art, is applicable to a range of objects, and provides more interesting insights overall.

**2. Related Work**

**Human Motion Prediction** Traditional methods tackle the human motion prediction task by utilizing shallow models such as Gaussian Processes [40], Hidden Markov Models [20], and Restricted Boltzmann Machine [38]. With the success of deep learning in various fields [47, 10, 42, 25, 44, 28, 11], and the availability of large-scale public datasets including Human3.6M [14] and CMU MoCap [5], various deep learning methods have been proposed recently to address this problem, which can be roughly classified into three categories: RNN, GCN, and...
Figure 2. The architecture of the motion context modeling network, which includes a graph semantic enriched GCN module, a GRU layer, and a pose reconstruction block. The GCN module encodes skeletal and known prior connections between joints, and learns implicit connections. The GRU layer deals with the sequence data, and pose reconstruction block converts the prediction results to pose space.


Structural Connection Modeling Human motion is a coordinated movement involving multiple joints. Recently, a set of models attempt to encode spatial dependencies or physical constraints between joints, which contain useful information for prediction. [15] proposes a spatio-temporal graph to explicitly model the structural information of human pose. [30] characterizes the pose as a kinematic tree based on the representation of Lie algebra to explicitly model the anatomical constraints. [43] divides human joints into several body parts and constructs a graph to capture joint dependencies. [7] and [32] design novel GCN architectures for capturing spatial dependencies via treating a pose as a generic graph. [22] develops a novel representation for human body, characterizing a body at multiple scales to capture more comprehensive correlations. [3] applies a global attention mechanism and a progressive decoding strategy to extract the long-range structural relations among the joints.

3. Our approach

Problem Definition Presented with a historical pose sequence $P_{0:T} = \langle p_0, p_1, \cdots, p_T \rangle$, we are interested in predicting its future pose sequence $\langle \hat{p}_{t+1}, \hat{p}_{t+2}, \cdots, \hat{p}_{t+T} \rangle$. A pose $p_i$ can be conveniently considered as the 3D coordinates of all body joints.

Method Overview The proposed method MPT (Motion Prediction leveraging Trajectory cues) consists of two key components. (1) MPT casts the historical trajectory of a joint $j$ as its frame-wise velocities and its final (last observed) position. (2) Trajectory cues are then fed into a novel motion context modeling network for future trajectory prediction, which considers rich semantic dependencies between joints. In what follows, we will elaborate the two components, respectively.

3.1. Trajectory Representation Conventionally, the human posture is described as the 3D coordinates or angles of all joints, then a recurrent neural network is engaged to absorb the historical pose sequence and output the future sequence. This characterizes the pose
of each frame statically and all joints are mixed together, bringing inherent difficulties in extracting motion dynamics. In contrast, a joint trajectory directly conveys temporal motion dynamics of per joint [32], which naturally reduces the complexity of motion context modeling at the base level. Inspired by these facts, we represent the pose sequence in the joint trajectory space.

Formally, given the historical pose sequence \( \langle p_0, p_1, \cdots, p_t \rangle \), the trajectory of a joint \( j \) can be formulated as:

\[
\Gamma = (v_1, v_2, \cdots, v_t, s_t),
\]

where \( v_i \in \mathbb{R}^3 \) denotes the position displacement of \( j \) between the adjacent \( i^{th} \) and \( i-1^{th} \) frames, and \( s_t \in \mathbb{R}^3 \) is the position of \( j \) in the \( t^{th} \) frame (the last observed frame). Put differently, \( \{v_i\}_{i=1}^t \) model the frame-wise velocities while \( s_t \) describes the final (last observed) position of \( j \). We further decompose velocity \( v_i \) into velocity magnitude \( m_i \in \mathbb{R} \) and velocity orientation \( o_i \in \mathbb{R}^3 \). Finally, we arrive at the formulation:

\[
\Gamma = (\{m_i\}_{i=1}^t, \{o_i\}_{i=1}^t, s_t).
\]

Overall, there exist \( n \) joints in the human skeleton and the \( n \) joints are represented by \( n \) historical trajectories.

The proposed trajectory representation in Eq. (2) has the following benefits. (1) Using Eq. (2), we can easily restore the entire joint trajectory with no information loss. Meanwhile, explicitly modeling of velocities and position of a joint leads to a richer motion context for predicting its future. (2) Mathematically, in this problem the Lagrangian corresponds to the product space of joint position and joint velocity, and learning dynamical evolution amounts to solving the Euler-Lagrange equation. Position \( s_t \) corresponds to potential energy while velocities \( v_i \) correspond to kinetic energy. By incorporating them, we have a complete characterization of the trajectory configuration space, which is consistent with the Lagrangian formulation of dynamical systems. Empirically, the representation also translates to significantly better performance compared to conventional models.

### 3.2. Semantic Enriched GCN For Motion Context Modeling and Pose Sequence Prediction

Up to this point, we have discussed reducing the motion prediction problem to extrapolating the trajectories of all joints. However, it is crucial to take into account the interdependence and interaction among these joints when we consider motion. To tackle the challenge, we model the human body as a semi-constrained graph. In particular, to adequately describe the rich spatial dependencies between joints, we explicitly consider three types of joint connections.

(1) **Skeletal** The natural skeletal connection between joints is obviously meaningful in motion context modeling. We model such connections using the skeletal adjacency matrix \( A_s \).

(2) **Motion Prior Knowledge** Most existing methods tend to consider merely the skeletal connections. However, geometrically separated joints may also show strong correlations [7, 32]. For example, the two arms always coordinate each other when clapping, walking, and swimming. Ignoring these valuable prior knowledge may lead to severe performance degradation. Therefore, we explicitly encode these useful prior knowledge in a semantic adjacency matrix \( A_p \). More specifically, in \( A_p \) we encode connections between two arms and two legs respectively in consideration of mirror symmetry tendency, and connections between a arm (e.g., left arm) and a leg (e.g., right leg) in opposite sides regarding synchronization tendency. It is easy to see that the model is extendable to other prior knowledge.

(3) **Learned** Besides fixed connections encoded in \( A_s \) and \( A_p \), we parameterize a trainable matrix \( A_f \), which is adaptively tuned to learn flexible and implicit dependencies between joints, providing important complementary connections.

Further, the connection strengths between joints are learned during training instead of being constant, which are captured by a weight matrix \( W \) [32]. The diagonal elements in the skeletal adjacency matrix \( A_s \) is set to 1 to take account of self-adjacency.

Typically, the operation of a general graph convolutional layer is given by:

\[
X^{r+1} = \sigma(\hat{A}X^{r}M^r)
\]

where \( X^r \in \mathbb{R}^{n \times l_r} \) and \( X^{r+1} \in \mathbb{R}^{n \times l_{r+1}} \) are the features of the \( r^{th} \) and \( r + 1^{st} \) layers, respectively. \( n \) is the number of nodes in the graph, which translates to the number of joints in this problem. \( l_r \) is the length of joint features at the \( r^{th} \) layer. \( \sigma(\cdot) \) is an activation function, e.g., ReLU. Matrix \( M^r \in \mathbb{R}^{l_r \times l_{r+1}} \) is network parameter (transformation matrix). Filter matrix \( \hat{A} \) is computed based on the adjacency matrix \( A \) by \( \hat{A} = \hat{D}^{-1/2}A\hat{D}^{-1/2} \), with \( \hat{A} = A + I \) and \( \hat{D} \in \mathbb{R}^{n \times n} \) being the degree matrix, \( \hat{D}_{i,i} = \sum_j \hat{A}_{i,j} \).

Similarly, one layer graph convolution in our GCN module is formulated as:

\[
X^{r+1} = \sigma((A_s + A_p + A_f) \circ W^{r}X^rM^r)
\]

where \( A_s \in \mathbb{R}^{n \times n} \) and \( A_p \in \mathbb{R}^{n \times n} \) encode skeletal connections and prior knowledge connections respectively. Trainable \( A_f \) captures implicit joint dependencies. Symbol \( \circ \) denotes element-wise product and \( W^{r} \in \mathbb{R}^{n \times n} \) is the trainable connection weight matrix.

**Benefits.** \( A_f \) adaptively extract flexible and implicit connections between joints, while fixed \( A_s \) and \( A_p \) constrains the training. \( A_f, A_s, \) and \( A_p \) complement each other, cap-

13302
turing rich joint dependencies. \( W^r \) enables learnable connection weights instead of constant ones.

Upon graph convolution, the rich dependencies between joints are considered. Mathematically, the trajectory features of a joint \( j \) is updated by incorporating trajectory features of other joints that are correlated to \( j \). The updated frame-wise trajectory features are then passed through a GRU layer, as shown in Fig. 2, to output future trajectories in the form of frame-wise velocities (namely velocity magnitudes \( \{m_i\}_{i=t+1}^{t+T} \) and orientations \( \{o_i\}_{i=t+1}^{t+T} \) for all joints. Finally, a simple pose reconstruction block is used to restore 3D poses from the predicted future frame-wise velocities.

**Loss functions.** We use weighted trajectory loss and bone length loss to acquire accurate motion prediction. Trajectory loss ensures that the predicted trajectory is consistent with the ground truth. Existing methods such as [32, 7] adopted equal weights for all joints in each predicted frame. This fails to attend to the spatial aspect that different joints engage differently in motion and the temporal aspect that later predictions rely on earlier predictions. Therefore, we assign higher weights to the joints possessing wider motion ranges and to earlier frames in the prediction. Formally,

\[
L_{\text{Traj}} = \frac{1}{n \cdot T} \sum_{p=t+1}^{t+T} \sum_{k=1}^{n} \left\| (\tilde{J}_k^p - J_k^p) \circ \lambda_k^p \right\|_2^2 \tag{5}
\]

where \( J_k^p \) denotes the ground truth of the \( k^{th} \) joint in the \( p^{th} \) frame, while \( \tilde{J}_k^p \) denotes the corresponding estimation. \( J_k^p \) is represented in the trajectory space by velocity of the \( k^{th} \) joint from \( p - 1^{th} \) frame to \( p^{th} \) frame. \( \lambda_k^p \) is the associated weight. Specifically, the spatial weights are designed following kinematic chain configurations while temporal weights decay as prediction goes futher.

**Bone length loss** enforces the bone length invariance across frames, which can be formulated as:

\[
L_{\text{Bone}} = \frac{1}{n \cdot T} \sum_{p=t+1}^{t+T} \sum_{b=1}^{n-1} \left\| (L_b^p - \tilde{L}_b^p) \circ \lambda_b^p \right\|_2^2 \tag{6}
\]

where \( \tilde{L}_b^p \) and \( L_b^p \) is the estimated and ground truth bone lengths of the \( b^{th} \) bone in the \( p^{th} \) frame. \( \lambda_b^p \) is the associated weight.

**4. Experiments**

In this section, we evaluate the proposed method on large benchmark datasets of three distinct articulate objects, namely human, mouse, and fish. We seek to answer the following research questions.

- **RQ1:** How is the proposed method comparing to state-of-the-art motion prediction approaches?
- **RQ2:** How much do different components of MPT contribute to its performance?
- **RQ3:** What interesting insights and findings can we obtain from the empirical results?

Next, we first present the experimental settings, followed by answering the above research questions one by one.

**4.1. Experimental Settings**

**Datasets** For human motion prediction, the large benchmark motion capture datasets Human3.6M (H3.6M) [14] and CMU MoCap [5] are engaged. For animal motion prediction, we utilize the public datasets of [30].

**Human 3.6M** H3.6M dataset is the most widely used and largest public dataset for evaluating human motion prediction methods. It contains 3.6 million 3D poses and videos for 7 subjects, each subject performs 15 different actions, such as eating, sitting, and purchases. Following the data processing schema of prior works [8, 30, 15], we downsample the motion sequence to 25 frames per second (FPS), use 6 subjects (S1, S6, S7, S8, S9, S11) for training, and test with subject 5 (S5).

**CMU MoCap** The CMU MoCap dataset contains 3D skeletal motion data of 40 objects under multiple infrared cameras. We adopt the same training/test split strategy as [21, 7]. For fair comparison, the sequences are also downsampled to 25 FPS.

**Fish and Mouse Datasets** The two datasets of [30] contain eight 3D fish pose sequences (50 FPS) and four 3D mouse pose sequences (25 FPS), respectively. In general, the sequence lengths vary from 298 frames to 15,387 frames. We follow [30] for data preprocessing.

**Parameter Settings** We implemented our methods on PyTorch [34] and experimented on a Nvidia GeForce Titan V GPU. The size of the convolution kernel for semantic enriched GCN is \( 25 \times 25 \). The hidden unit size of GRUs is 128. The Adam Optimizer [17] is employed with an initial learning rate of 0.001 which decays by 10% every 10 epochs. Batch size is set to 16 and the gradient clipping is used at a threshold of 5 and trained for 50 epochs. We utilize \( t = 10 (400ms) \) historical frames as inputs to predict future \( T = 25 (1,000ms) \) frames in training. In the loss functions of Eqs.(5) & (6), we assign gradually decreasing temporal weights to the predicted frames. The spatial weights for different joints are computed based on their spatial moving ranges, where joints undergoing wider range of motions are assigned higher weights.

**4.2. Comparison with Existing Motion Prediction Methods (RQ1)**

**Human motion prediction** We first compare our method with the state-of-the-art approaches on the H3.6M and CMU datasets. The performance of all approaches are evaluated using the widely adopted metric MPJPE (Mean Per Joint Position Error) in millimeter [32, 7, 30], i.e., the
Table 1. Comparisons of position error (in millimeter) for short-term and long-term predictions on H3.6m dataset. Our method consistently outperforms state-of-the-art methods by a large margin and keeps delivering the best performance on different actions. Surprisingly, compared to the current state-of-the-art, our method achieves a remarkable 17.5% accuracy improvement in average. It is noteworthy that our method consistently achieves the best results for both short-term and long-term predictions. Our second observation is that more complex actions such as “Walking Dog” and “Purchase” are harder to be predicted, leading to performance decay for all methods.

In addition to quantitative comparisons, we further visualize the prediction results of state-of-the-art methods. Fig. 3 demonstrates the prediction results for “Eating” and “Talking Photo”, where the first 3 frames (in blue) in each line are historical frames and the subsequent frames in red spatial distance between ground truth and prediction. Following the literature convention [33, 37], we evaluate our method on both short-term (< 400ms) and long-term (400-1,000ms) predictions.

The performance of different models on the H3.6M dataset is evaluated on all kinds of actions, including “Directions”, “Eating”, “Greeting”, “Purchases”, “Sitting Down”, “Walking Dog”, etc. A total of 10 methods are compared, including LSTM3LR [8], Res-GRU [33], ConSeq-Seq [21], HMR [30], FC-GCN [32], LDR [7], TrajNet [27], SDMTL [26], HRI [31], and our MPT model. We present the results of 11 various actions and the overall average results for all actions.

The short-term and long-term prediction results are presented in Table 1. The first observation is that our MPT
are future motion predictions. Specifically, we can observe that in the “Talking Photo” action, our model successfully captured the descending trend of the right leg while other methods did not. As a result, their predicted positions of the right leg are significantly distinct from the ground truth. In the “Eating” action, LSTM3LR tends to predict freezed motion in the long term, Res-GRU yields wrong movements of the two hands, FC-GCN has problems in predicting the position of the right leg, while HRI has errors in predicting the position of the left hand. In contrast, the movements of the hands and head are more coordinated and smooth in our results, which are also more consistent with the ground truth. Similar results are observed for other actions. Motivated readers may refer to https://github.com/Pose-Group/MPT for more visualized results. The empirical evidences reveal that our model can better discover subtle movement trends and achieve more accurate forecasts. Overall, the predicted pose sequences generated from our model are closer to the ground truth.

Furthermore, we conducted extensive experiments on the CMU MoCap dataset. The results are shown in Table 2. Consistently, we found that our model significantly outperforms existing methods for both short-term and long-term predictions. For example, on the “Basketball Signal” and “Soccer” actions, our model achieves an average of 20.8% and 22.6% improvements over state-of-the-art method, respectively. This reconfirms the effectiveness of working in trajectory space for motion prediction.

Animal Motion prediction We use the fish and mouse datasets to further evaluate our model and other methods. Whereas the human datasets pose the challenge of having to model multiple kinematic chains, the fish and mouse datasets raise different issues, i.e., 1) long kinematic chain of 21 joints for fish; 2) animals exhibit faster and highly stochastic movements than humans; 3) relatively smaller datasets for training. The quantitative results are reported in Table 3. Empirical results suggest that mouse is more easier to be predicted than fish, which may due to the fact that fish is more active in action and the fish contain 21 joints which is significant longer than the 5 joints of mouse. Moreover, our method is shown to consistently outperform state-of-the-art methods on the animal datasets. Specifically, the proposed method achieves a 24.1% accuracy improvement over HMR on Fish dataset, and 13.8% on Mouse dataset. This validates the effectiveness and generalizability of the method.

4.3. Ablation Study (RQ2)

We further study the influence of individual components in the proposed framework through the following ablation studies. Experiments are performed on the H3.6M dataset, with empirical results reported in Table 4. In the table, the motion prediction accuracies for 80, 160, 320, 400, 560 and 1,000 ms are presented. First, to verify the impact of the proposed trajectory representation in modeling temporal motion contexts, we replace it with the conventional pose sequence representation (adopting 3D coordinates of joints). As presented in the second and fifth lines of Table 4, empirical results reveal...
that using our trajectory representation to encode temporal dynamics significantly boost accuracy for both short-term and long-term predictions. Specifically, using the trajectory representation achieves more accuracy gain on short-term prediction than on long-term prediction.

Next, we evaluate the impact of explicit relations, namely using only the skeletal connection and prior knowledge while removing the adaptively learning of hidden joint connections. The results in the second line of Table 4 show that when implicit relation matrix is removed, the accuracies of the short-term and long-term prediction are significantly affected. However, the performance drop is relatively smaller than that of replacing trajectory representation.

We also explore using merely the semantic enriched implicit relations. Towards this aim, we directly remove the articulated relations. Towards this aim, we directly remove the articulated relations. The results in Table 4 show that replacing trajectory representation accomplishing the complete characterization of the trajectory configuration space and ultimately facilitate learning the Euler-Lagrange equation, i.e. modeling motion context. Extensive experiments confirm that our method significantly surpasses existing work on 4 different benchmark datasets. Interestingly, the method can also be generalized to fish and mouse datasets.

### 4.4. Discussion (RQ3)

Experiments on 4 different benchmark datasets suggest that representing 3D skeleton motion sequence in trajectory space achieves significantly improved accuracy over representations in conventional pose space. Meanwhile, the generated visualization results are more natural and exhibit better inter-frame continuity.

Interestingly, a point that attracts our attention is: for long-term prediction, we find that although our proposed method still outperforms state-of-the-art approach, but as the prediction horizon goes deeper, the advantage of trajectory representation decreases. We plan to dive deeper into this phenomenon and come up with new solutions.

## 5. Conclusion

In this paper, we have proposed a trajectory representation consisting of position and frame-wise velocities, where position corresponds to potential energy and velocities correspond to kinetic energy. We further engage in a semi-constrained graph model to graph the constraints. These components together formulate a complete characterization of the trajectory configuration space and ultimately facilitate learning the Euler-Lagrange equation, i.e. modeling motion context. Extensive experiments confirm that our method significantly surpasses existing work on 4 different benchmark datasets. Interestingly, the method can also be generalized to fish and mouse datasets.

### 6. Acknowledgments

Corresponding authors: Pengxiang Su, Shuang Wu, Xuanjing Shen, Haipeng Chen.

This research is supported by the National Key Research and Development Program of China under Grant No.2020AAA0140004, the Natural Science Foundation of Zhejiang Province, China (Grant No. LQ19F020001), the National Natural Science Foundation of China (No. 61902348 & No. 61876070), and the Key R&D Program of Zhejiang Province (No. 2021C01104).

### Table 2. Prediction results (in MPJPE) on CMU-MoCap dataset.

<table>
<thead>
<tr>
<th>Method</th>
<th>Time (ms)</th>
<th>80</th>
<th>160</th>
<th>320</th>
<th>400</th>
<th>500</th>
<th>1,000</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERD[8]</td>
<td>80</td>
<td>4.4</td>
<td>6.5</td>
<td>9.5</td>
<td>11.1</td>
<td>13.6</td>
<td>19.2</td>
</tr>
<tr>
<td>LSTM[6]</td>
<td>80</td>
<td>5.3</td>
<td>7.8</td>
<td>14.1</td>
<td>16.7</td>
<td>22.0</td>
<td>51.9</td>
</tr>
<tr>
<td>HMR[30]</td>
<td>80</td>
<td>5.8</td>
<td>12.3</td>
<td>27.8</td>
<td>38.2</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>Res-GRU [33]</td>
<td>80</td>
<td>7.2</td>
<td>8.5</td>
<td>10.2</td>
<td>11.0</td>
<td>13.6</td>
<td>18.0</td>
</tr>
<tr>
<td>FC-GCN [32]</td>
<td>80</td>
<td>8.0</td>
<td>14.5</td>
<td>31.9</td>
<td>41.9</td>
<td>59.4</td>
<td>102.7</td>
</tr>
<tr>
<td>FC-GCN [32]</td>
<td>80</td>
<td>9.4</td>
<td>18.8</td>
<td>31.6</td>
<td>43.2</td>
<td>51.1</td>
<td>93.6</td>
</tr>
<tr>
<td>ConSeq2Seq [21]</td>
<td>80</td>
<td>9.8</td>
<td>17.6</td>
<td>35.7</td>
<td>45.1</td>
<td>70.3</td>
<td>152.4</td>
</tr>
<tr>
<td>ConSeqSeq [21]</td>
<td>80</td>
<td>10.3</td>
<td>21.1</td>
<td>42.7</td>
<td>50.9</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>ConSeqSeq [21]</td>
<td>80</td>
<td>11.3</td>
<td>21.7</td>
<td>44.2</td>
<td>59.6</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>ConSeqSeq [21]</td>
<td>80</td>
<td>11.6</td>
<td>21.7</td>
<td>44.2</td>
<td>57.3</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>ConSeqSeq [21]</td>
<td>80</td>
<td>12.9</td>
<td>27.6</td>
<td>73.5</td>
<td>92.2</td>
<td>176.6</td>
<td></td>
</tr>
<tr>
<td>ConSeqSeq [21]</td>
<td>80</td>
<td>14.0</td>
<td>25.4</td>
<td>49.6</td>
<td>61.4</td>
<td>139.9</td>
<td>191.2</td>
</tr>
<tr>
<td>ConSeqSeq [21]</td>
<td>80</td>
<td>15.2</td>
<td>30.5</td>
<td>65.1</td>
<td>84.0</td>
<td>153.5</td>
<td>176.6</td>
</tr>
<tr>
<td>ConSeqSeq [21]</td>
<td>80</td>
<td>16.9</td>
<td>34.4</td>
<td>76.3</td>
<td>96.8</td>
<td>176.6</td>
<td>191.2</td>
</tr>
</tbody>
</table>

### Table 3. Evaluation (in MPJPE) on Fish and Mouse datasets.

<table>
<thead>
<tr>
<th>Method</th>
<th>Fish</th>
<th>Mouse</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERD[8]</td>
<td>4.4</td>
<td>85.9</td>
</tr>
<tr>
<td>LSTM[6]</td>
<td>5.3</td>
<td>127.1</td>
</tr>
<tr>
<td>HMR[30]</td>
<td>5.8</td>
<td>153.5</td>
</tr>
<tr>
<td>Res-GRU [33]</td>
<td>8.0</td>
<td>162.6</td>
</tr>
<tr>
<td>FC-GCN [32]</td>
<td>9.4</td>
<td>176.6</td>
</tr>
<tr>
<td>ConSeq2Seq [21]</td>
<td>9.8</td>
<td>191.2</td>
</tr>
<tr>
<td>ConSeqSeq [21]</td>
<td>10.3</td>
<td>222.4</td>
</tr>
<tr>
<td>ConSeqSeq [21]</td>
<td>11.3</td>
<td>244.0</td>
</tr>
<tr>
<td>ConSeqSeq [21]</td>
<td>11.6</td>
<td>244.0</td>
</tr>
<tr>
<td>ConSeqSeq [21]</td>
<td>12.9</td>
<td>276.3</td>
</tr>
<tr>
<td>ConSeqSeq [21]</td>
<td>14.0</td>
<td>298.7</td>
</tr>
<tr>
<td>ConSeqSeq [21]</td>
<td>15.2</td>
<td>331.8</td>
</tr>
<tr>
<td>ConSeqSeq [21]</td>
<td>16.9</td>
<td>374.3</td>
</tr>
<tr>
<td>ConSeqSeq [21]</td>
<td>18.4</td>
<td>441.6</td>
</tr>
</tbody>
</table>

### Table 4. Impact of GCN module and trajectory representation.

<table>
<thead>
<tr>
<th>Fish</th>
<th>Mouse</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERD[8]</td>
<td>4.4</td>
</tr>
<tr>
<td>LSTM[6]</td>
<td>5.3</td>
</tr>
<tr>
<td>HMR[30]</td>
<td>5.8</td>
</tr>
<tr>
<td>Res-GRU [33]</td>
<td>8.0</td>
</tr>
<tr>
<td>FC-GCN [32]</td>
<td>9.4</td>
</tr>
<tr>
<td>ConSeq2Seq [21]</td>
<td>9.8</td>
</tr>
<tr>
<td>ConSeqSeq [21]</td>
<td>10.3</td>
</tr>
<tr>
<td>ConSeqSeq [21]</td>
<td>11.3</td>
</tr>
<tr>
<td>ConSeqSeq [21]</td>
<td>11.6</td>
</tr>
<tr>
<td>ConSeqSeq [21]</td>
<td>12.9</td>
</tr>
<tr>
<td>ConSeqSeq [21]</td>
<td>14.0</td>
</tr>
<tr>
<td>ConSeqSeq [21]</td>
<td>15.2</td>
</tr>
<tr>
<td>ConSeqSeq [21]</td>
<td>16.9</td>
</tr>
<tr>
<td>ConSeqSeq [21]</td>
<td>18.4</td>
</tr>
</tbody>
</table>

In this paper, we have proposed a trajectory representation consisting of position and frame-wise velocities, where position corresponds to potential energy and velocities correspond to kinetic energy. We further engage in a semi-constrained graph model to graph the constraints. These components together formulate a complete characterization of the trajectory configuration space and ultimately facilitate learning the Euler-Lagrange equation, i.e. modeling motion context. Extensive experiments confirm that our method significantly surpasses existing work on 4 different benchmark datasets. Interestingly, the method can also be generalized to fish and mouse datasets.
References


