Long Term Motion Prediction Using Keyposes

Sena Kiciroglu1 Wei Wang1,2 Mathieu Salzmann1,3 Pascal Fua1
1 CVLab, EPFL 2 MHUG, University of Trento 3 Clearspace
sena.kiciroglu@epfl.ch

Abstract

Long term human motion prediction is essential in safety-critical applications such as human-robot interaction and autonomous driving. In this paper we show that to achieve long term forecasting, predicting human pose at every time instant is unnecessary. Instead, it is more effective to predict a few keyposes and approximate intermediate ones by interpolating the keyposes.

We demonstrate that our approach enables us to predict realistic motions for up to 5 seconds in the future, which is far longer than the typical 1 second encountered in the literature. Furthermore, because we model future keyposes probabilistically, we can generate multiple plausible future motions by sampling at inference time. Over this extended time period, our predictions are more realistic, more diverse and better preserve the motion dynamics than those state-of-the-art methods yield.

1. Introduction

Human motion prediction is a key component of many vision-based applications, such as automated driving [17, 12, 34], surveillance [31, 21], accident prevention [37, 42], and human-robot interaction [15, 7]. Its goal is to forecast the future 3D articulated motion of a person given their previous 3D poses. While recurrent neural networks [14, 30] and graph convolutional networks [29, 23, 28] are effective for short-term predictions, typically up to one second in the future, their prediction accuracy degrades quickly beyond that, and addressing this shortcoming remains an open problem.

This paper focuses on longer-term prediction, which is critical in many areas, such as providing an autonomous system sufficient time to react to human motions. Our key insight is that, for this task, predicting the pose in every future frame is unnecessary. For example, consider a boxing jab motion. The most significant poses are the ones where the hand is closest to the chest and where the arm is the most extended. The in-between poses are transition ones that can be interpolated from these two. Therefore instead of treating a motion as a sequence of consecutive poses, we downsample it to a set of keyposes from which all other poses can be interpolated up to a given precision. We then use these keyposes for long-term motion prediction.

The simplest way to so would be to replace the poses in existing frameworks by our keyposes. However, while all keyposes are unique, some tend to be similar to each other. We therefore cluster those we extract from a training set and develop a framework that treats keypose prediction as a classification problem. This has two main advantages. First, it overcomes the tendency of regression-based prediction methods to converge to the mean pose in the long term. Second, it allows us not only to predict the most likely future motion by selecting the most probable clusters but also to generate multiple plausible predictions by sampling the relevant probability distributions. This is useful because people are not entirely predictable, as in the case of a pedestrian standing on the curb who may, or may not, cross the street.

In summary, our contributions are threefold. (i) We introduce a keypose extraction algorithm to represent human motion in a compact way. (ii) We formulate motion prediction as a classification problem and design a framework to predict keypose labels and durations. (iii) We demonstrate that our approach enables us to predict multiple realistic motions for up to 5 seconds in the future, which is far longer than the typical 1 second encountered in the literature. The motions we generate preserve the dynamic nature of the observations, whereas the methods designed for shorter timespans tend to degenerate to static poses. Our code and an overview video can be accessed via our project website, https://senakicir.github.io/projects/keyposes.

2. Related Work

The complexity of human motion makes deep learning an ideal framework for tackling the task of motion prediction. In this section, we first review the two main classes of
deep models that have been used in the field and then discuss approaches that depart from these main trends. Finally, we discuss the use of keyposes for different tasks.

**Human Motion Prediction using RNNs.** Recurrent neural networks (RNN) are widely used architectures for modeling time-series data, for instance for natural-language processing [41] and music generation [36, 35]. Since the work of Fragiadaki et al. [13], these architectures have become highly popular for human motion forecasting. In this context, the S-RNN of Jain et al. [19] transforms spatiotemporal graphs to a feedforward mixture of RNNs; the Dropout Autoencoder LSTM (DAE-LSTM) of Ghosh et al. [14] synthesizes long-term realistic looking motion sequences; the recent Generative Adversarial Imitation Learning (GAIL) of Wang et al. [39] was employed to train an RNN-based policy generator and critic networks. HPGAN [4] uses an RNN-based GAN architecture to generate diverse future motions of 30 frames.

Despite their success, using RNNs for long-term motion prediction suffers from drawbacks. As shown by Martinez et al. [30], they tend to produce discontinuities at the transition between observed and predicted poses, and often yield predictions that converge to the mean pose of the ground-truth data in the long term. In [30], this was circumvented by adding a residual connection so that the network only needs to predict the residual motion. Here, we also develop an RNN-based architecture. However, because we treat keypose prediction as a classification task, our approach does not suffer from the accumulated errors that such models tend to generate when employed for regression.

**Human Motion Prediction using GCNs.** Mao et al. [29] proposed to overcome the weaknesses of RNNs by encoding motion in discrete cosine transform (DCT) space, to model temporal dependencies, and learning the relationships between the different joints via a GCN, Lebailly et al. [23] build on top of this work by combining a GCN architecture with a temporal inception layer. The temporal inception layer serves to process the input at different subsequence lengths, so as to exploit both short-term and long-term information. Alternatively, [28, 20] combine the GCN architecture with an attention module aiming to learn the repetitive motion patterns. These methods constitute the state of the art for motion prediction. Nevertheless, they were designed for forecasting up to 1 second in the future. As will be shown by our experiments, for longer timespans, they tend to degenerate to static predictions.

**Other Human Motion Prediction Approaches.** Several other architectures have been proposed for human motion prediction. For example, Bütepage et al. [6] employ several fully-connected encoder-decoder models to encode different properties of the data. One of the models is a time-scale convolutional encoder, with different filter sizes. In [7], a conditional variational autoencoder (CVAE) is used to probabilistically model, predict and generate future motions. This probabilistic approach is extended in [8] to incorporate hierarchical action labels. Aliakbarian et al. [3] also perform motion generation and prediction by encoding their inputs using a CVAE. They are able to generate diverse motions by randomly sampling and perturbing the conditioning variables. Similarly, Yuan et al. [40] also use a CVAE based approach to generate multiple futures. Li et al. [24] use a convolutional neural network for motion prediction, producing separate short-term and long-term embeddings. In [10, 1], interactions between humans and objects in the scene are learned for context-aware motion prediction. Akse et al. [2] use transformer networks to predict up to 20 seconds in the future, but only for cyclic motions. Zhou et al. [43] also target long term predictions, but provide only qualitative results for sequences from walking, dancing, and martial arts, which tend to follow well-structured patterns. Concurrent to our work Diller et al. [11] use characteristic 3D poses resembling our keyposes for long-term motion prediction. However, these poses are manually annotated rather than automatically extracted from sequences. A different related task is to generate realistic motions by conditioning on the action label, rather than the past motion [32, 16]; in our work, we show that regressing the future pose at every time instant is unnecessary and truly long-term prediction can be achieved more accurately by focusing on the essential poses, or keyposes, in a sequence. These poses are extracted automatically from the sequence, without manual annotations.

**Keyposes Applied to Other Tasks.** Keyposes have been used for different tasks, such as action recognition. For example, in [27], 2D keyposes are used for single view action recognition. In [26], Adaboost is used to select keyposes that are discriminative for each action. In [5], linear latent low-dimensional features extracted from sequences for action recognition and action prediction. Furthermore, [22] focus on generating realistic transitions between nodes in a motion graph, which resembles our notion of keyposes, to synthesize short animated sequences. However, none of these works predict future keyposes given past ones.

### 3. Methodology

Classically, the task of motion prediction is defined as producing the sequence of 3D poses from \( t = 1 \) to \( t = N \), denoted as \( P_{1:N} \). Given the sequence of poses from \( t = 0 \) to \( t = \), denoted as \( P_{−M:0} \). Each pose value \( P_t \) is of dimension \( 3 \times J \), where \( J \) is the total number of joints. Therefore, motion prediction is written as

\[
P_{1:N} = F(P_{−M:0})
\]

where \( F \) is the prediction function.
Our overall pipeline for predicting future motions via keyposes. It consists of the following steps: keypose extraction, keypose prediction, linear interpolation to reconstruct the sequence, and refining the final sequence.

Our approach departs from this classical formalism by predicting keyposes from keyposes. As will be discussed in more detail in Section 3.1, keyposes encode the important poses in a sequence \( P_{1:T} \), such that the remaining poses can be obtained by linear interpolation between subsequent keyposes. Therefore, our keypose-to-keypose framework takes as input a motion \( P_{-M:0} \) defined by its keyposes \( K_{-I_1:0} \), where \( I_1 \ll M \) is the number of keyposes in the past sequence. We then predict \( K_{1:I_2} \), where \( I_2 \ll N \) is the number of keyposes in the future sequence. We write this as

\[
K_{1:I_2} = G(K_{-I_1:0})
\]

where \( G \) is the keypose-to-keypose prediction function.

Our overall pipeline, illustrated in Figure 3, consists of extracting keyposes from input sequences, feeding them to the keypose prediction network, reconstructing the predicted sequence via linear interpolation, and refining the final result via a refinement network. We describe each of these steps in detail below.

### 3.1. Keyposes

Let us now discuss how we obtain keyposes \( K_i, i \in [1, T] \), given a sequence of poses \( P_1 \), \( t \in [1, T] \). We define the keyposes as the poses in \( P_1 \) such that linear interpolation can be used to obtain the remaining poses. We therefore employ an optimization-based strategy to identify the poses from which the L2 error between the original sequence \( P \) and the sequence reconstructed by linear interpolation is minimized. Our method proceeds as follows:

- We set \( P_1 \) and \( P_T \) to be the initial keyposes.
- We reconstruct the sequence by linearly interpolating the set of keyposes. We denote the reconstruction as \( \hat{P}_t, t \in [1, T] \).
- We select the pose \( P_t \) at position \( t \) which has the highest L2 error with respect to \( \hat{P}_t \), the pose reconstructed by linear interpolation at the same time index. We add \( P_t \) to our set of keyposes.
- The algorithm continues recursively, selecting keyposes from the sequences between \([1, t]\) and \([t, T]\). The recursion ends once the average reconstruction error of the linear interpolation is below a threshold, yielding a set of keyposes.

### 3.2. Motion Prediction with Keyposes

In principle, we could directly use the above-mentioned keyposes for prediction, by simply learning to regress keypose values. However, for long-term prediction, this would exhibit the same tendency as existing frameworks to converge to a static pose. To overcome this, we propose to cluster the training keyposes and treat keypose prediction as a classification task, where the clusters act as categories.

To this end, we extract the keyposes for every training motion individually, and cluster all the resulting training keyposes into \( K \) clusters via k-means. Each keypose is then given a label determined by the cluster it is assigned to. Finally, we prune the keyposes by removing the unnecessary intermediate ones that have the same label as their preceding and succeeding keypose. An example distribution of keyposes in a sequence is shown in Figure 2.

This formalism allows us to cast keypose prediction as a classification problem. Specifically, instead of predicting the future keypose values, we predict their labels. Given the labels, \( l_i \) and \( l_{i+1} \), of two subsequent keyposes, \( K_i \) and \( K_{i+1} \), we can simply estimate the intermediate poses via linear interpolation between the corresponding cluster centers. However, this requires the duration \( d_{i+1} \) between the two keyposes, indicating the number of intermediate poses, which we therefore also predict.

#### 3.2.1 Network Design and Training

We have designed an RNN based neural network as our keypose-to-keypose prediction framework, as shown in Figure 3. At each time step, in addition to the hidden representation of the previous time step, our recurrent unit takes as input the previous keypose label \( l_i \) and duration \( d_i \). Specifically, we represent the label as a distribution \( L_i \) computed as follows.

1. If we know the true keypose value (i.e., for observed past keyposes): We compute the proximity between the keypose value \( V_i \) and every cluster center \( C_j \), \( j \in [1, K] \) as the negative average Euclidean distance between the corresponding joints in \( V_i \) and \( C_j \). These values form a \( K \)-dimensional proximity vector for each keypose \( i \).
Figure 3. **Keypose-to-keypose network structure**. (a) Overall architecture. At each time step $i$, a keypose GRU (KP-GRU) unit predicts the keypose labels and durations of the next step $i + 1$. The time of the last observation is denoted by $i = 0$. Before this time-step, the network is given ground-truth keyposes as conditioning signal. The label distribution $L_i$ for past keyposes is found using the keypose value $V_i$. After time-step $i = 0$, the network is given its own predictions as input rather than the ground truth. The label distribution $L_i$, in this case, is found using the predicted label $l_i$. The orange T blocks represent the transformation to compute the distributions. (b) Inner structure of the KP-GRU unit, which consists of a three layer GRU network followed by a fully connected layer.

2. If we do not know the keypose value (i.e., for inferred future keyposes): We compute the proximities between the cluster center corresponding to the predicted label $l_i$, $C_{l_i}$, and all cluster centers $C_j$, $j \in [1, K]$.

3. We pass the resulting proximity vector through a softmax operation with a temperature of 0.03 to obtain a distribution $L_i$ over the labels. To also treat duration prediction as a classification task, we categorize the durations into very short (less than 4 frames), short (between 5 and 10 frames), medium (between 10 and 14 frames), long (between 14 and 25 frames), and very long (more than 25 frames). We then encode the duration $d_i$ of a keypose as a one-hot encoding $D_i$ over these categories and output a distribution $L_i$ for the future keyposes.

Therefore, our network predicts a pair of distributions: one over the labels and one over the duration categories. We train the network using two loss functions:

- $E_{\text{labels}}$: The cross-entropy loss between the ground-truth label and the predicted label distribution;
- $E_{\text{dur}}$: The cross-entropy loss between the predicted duration distribution and ground-truth duration category.

The overall loss of our network therefore is

$$E = w_{\text{labels}} E_{\text{labels}} + w_{\text{dur}} E_{\text{dur}},$$

where $w_{\text{labels}}$ and $w_{\text{dur}}$ weigh the different loss terms.

During training, the label of the next keypose $l_{i+1}$ is determined as the one with the highest predicted probability. We then compute a distribution $L_{i+1}$ from this label as described above. This procedure prevents error accumulation as the prediction progresses and guarantees that the network will never see anything very different from what it was trained on. The duration of the next keypose $d_{i+1}$ is determined similarly: According to the category with the highest probability, the duration is set to 3 for very-short, 6 for short, 12 for medium, 16 for long, and 25 for very long. Using the predicted label and duration of each time-step, we can reconstruct the sequence via linear interpolation between the corresponding cluster centers, as described previously.

During training, we observe 7 past keyposes and predict 12 future keyposes. At test time, we predict until we reach 5 seconds. Weights of the loss terms are set to $w_{\text{labels}} = 1.0$, $w_{\text{dur}} = 0.1$. Our network is trained for 100 epochs with a batch size of 64. We use an Adam optimizer with a learning rate of 0.0001 and a 0.01 weight decay. We report the results of the model with the highest validation score.

### 3.2.2 Inference and Interpolation

Our network produces distributions over the keypose clusters. Hence, at inference time, for each iteration of the recurrent network, we can sample the future label and duration from the predicted distributions. In practice, before sampling, we smooth the predicted distributions via a softmax with a temperature of 0.3. This sampling scheme allows us to produce multiple future sequences given a single observation.

Once we have predicted a set of keypose labels and their durations, we can interpolate the intermediate poses and reconstruct the future sequence. Denoting by $t$ the time index of keypose $K_t$ in the sequence, the intermediate pose at time $t_1 > t$ is computed as

$$P_{t_1} = C_{l_t} + (t_1 - t) \frac{C_{l_{t+1}} - C_{l_t}}{d_{t+1}},$$

where $C_{l_t}$ and $C_{l_{t+1}}$ are the cluster centers corresponding to labels $l_t$ and $l_{t+1}$.

The sequences obtained by linear interpolation can then be refined using a pretrained refinement network trained to produce sequences that preserve the poses of the original sequence. Formally this operation can be written as

$$P_{t, \text{ref}} = R(P_{1:N}) ,$$

where $R$ denotes the refinement function and $P_{t, \text{ref}}$ denotes the refined pose sequence. We describe this network in more detail in the appendix.

### 4. Experiments

#### 4.1. Datasets

**Human3.6M** [18] is a standard 3D human pose dataset and has been widely used in the motion prediction literature [30, 19, 29]. It contains 15 actions performed by 7 subjects. Human pose is represented using the 3D coordinates of 32 joints. As previous work [29, 23, 28], we load the exponential map representation of the dataset, remove global
rotation and translation, and generate the Cartesian 3D coordinates of each joint mapped onto a uniform skeleton. Following the implementation of existing works [29, 23, 28], subject 5 is reserved for testing, subject 11 for validation and the remaining subjects are used for training. We test each method on the same 64 sequences formed using indices randomly selected from Subject 5’s sequences. Note that the observed keyposes are extracted using the sequence only up to the present time index as opposed to the entire sequence. The threshold used for keypose extraction is 500mm, and we cluster the keyposes into 1000 clusters.

CMU-Mocap [9] is another standard benchmark dataset for motion prediction and was used in [24, 29, 23]. As explained in [24], the eight action categories with enough trials are used for motion prediction. We used six out of eight actions, basketball, basketball signal, directing traffic, jumping, soccer, and wash window, as the sequences for running and walking were too short to provide enough input keyposes for our method. One sequence of each action is reserved for testing, one for validation and the rest are used for training. The dataset is loaded and processed in the same manner as Human3.6M. The threshold used for keypose extraction is 250mm, as some sequences are quite short, and we found that extracting more keyposes increases validation accuracy. We cluster the keyposes into 100 clusters, as this dataset is much smaller than Human3.6M and contains only 6 action classes as opposed to 15.

4.2. Baselines

We selected the following baselines for comparison purposes: HisRep [28] and TIM-GCN [23] constitute the SOTA among the methods designed for long-term prediction. For HisRep, we evaluate two versions. The first one, HisRep10, was presented as the best model in [28]. It is trained to output 10 frames and iteratively use the predicted frames as input for longer term prediction. We also evaluate HisRep125, which directly predicts 125 frames by taking 150 past frames as input. For TIM-GCN, we trained a model that observes subsequences of lengths 10, 50 and 100 and predicts 125 frames, hence tailoring the architecture to longer-term predictions of 5 seconds. Finally we compare against Mix&Match [3] and DLow [40], the SOTA methods for multiple-long-term motion prediction, trained to predict 125 future frames using 100 past frames. For all the baselines, we used the model that gave the best validation accuracy, to be consistent with our model selection strategy.

4.3. Metrics

As in [14], we evaluate the quality and plausibility of the generated motions by passing them through an action classifier trained to predict the action category of a given motion. If the predicted motion is plausible, such a classifier should output the correct class. To focus our evaluation on the quality of the predicted motions, we designed a Motion-Only Action Classifier (MOAC) based on the architecture of [25], with the pose stream removed and only the motion stream remaining. It takes as input motions encoded as the difference between poses in consecutive time-steps. This eliminates the scenario of a static prediction scoring very high under this metric. We have trained it on the training sequences of Human3.6M and CMU-Mocap separately. We report the top-K action recognition accuracy in percentages obtained with this classifier. For our method, Mix&Match, and DLow, which can output multiple future predictions, we report the average accuracy over 100 predictions.

We also report the PSKL metric [33], which is the KL divergence between the power spectrums of the ground-truth future motions and the predictions. As the KL divergence is asymmetric, we evaluate it in both directions and denote the results as ‘gt-pred’ and ‘pred-gt’ respectively. These values being close indicates that the ground truth and predicted motions are similarly complex.

The mean per-joint position error (MPJPE) is the most commonly used metric to evaluate motion prediction. We report the MPJPE errors at 1 second, which is the conventional long-term timestamp, and at 5 seconds. For multiple-prediction methods, we report the MPJPE results of the closest predicted sequences. We present two results: the MPJPE calculated by finding the sequence with the minimum average MPJPE (denoted as “ave”) and the sequence with the minimum MPJPE at the second being evaluated (denoted as “best”).

Finally, for multiple-prediction methods we report the results of a diversity metric [40, 3] for 100 predictions, calculated by finding the average pairwise $L2$ distances between all pairs of generated sequences.

4.4. Comparative Results

We compare our approach to the baselines on Human3.6M and CMU-Mocap in Table 1 on the MOAC metric. In both cases, our method outperforms the others by a large margin. In Table 2, we report the results for the PSKL metric and show that we outperform the other methods by having both lower PSKL values and having very close ‘gt-pred’ and ‘pred-gt’ values.

In Table 3, we evaluate the diversity and MPJPE losses of the predicted sequences. We observe that the diversity value of our method increases as we increase the softmax temperature used for sampling during inference. Increased diversity allows us to achieve lower MPJPE values since we now have a higher chance of sampling the correct future motion. However this also leads to a drop in average MOAC accuracy. This clearly shows the tradeoff between predicting diverse motions and motions that represent the action of interest, or are close to the ground truth. Therefore, in our evaluations we choose to set the temperature
Figure 4. **Qualitative evaluation of our results** on the Human3.6M (Figures a, b, c) and CMU-Mocap (Figure d) datasets. We present the results of: TIM-GCN (green), HisRep10 (dark blue), HisRep125 (light blue), Mix&Match (violet), DLow (pink), Ours (red). For the multiple prediction methods, we display the prediction that has the lowest average MPJPE error with respect to the ground truth. The top black row depicts the ground truth, and the first 5 poses are the conditioning ones. The numbers at the top indicate the future timestamp in seconds. We highlight the segments and body parts that undergo significant motion in green, and the areas that are static for long stretches in red. Our approach yields more dynamic poses for discussion, walking dog and directing traffic, which are acyclic motions. For cyclic motions such as walking, the other methods are also able to produce dynamic poses.
Figure 5. **Qualitative results of our multiple motion prediction** obtained by sampling the predicted label distribution. The numbers at the top indicate the future timestamp in seconds. The top row in black depicts the ground truth, and the remaining rows in color are our multiple generated motions. The sampled motions are diverse, yet can all still be classified as “sitting down”.

| Human3.6M | CMU-MoCap |
|-----------|-----------|
|           | top-1     | top-2 | top-3 | top-5 | top-1 | top-2 | top-3 | top-5 |
| oracle    | 51        | 70    | 79    | 91    | 86    | 88    | 90    | 100   |
| TIM-GCN [23] | 16 | 26 | 36 | 55 | 44 | 69 | 85 | 95 |
| HisRep10 [28] | 21 | 32 | 39 | 53 | 42 | 54 | 62 | 88 |
| HisRep125 [28] | 20 | 32 | 44 | 60 | 34 | 48 | 57 | 82 |
| Mix&Match [3] | 18 | 32 | 45 | 61 | 30 | 39 | 58 | 85 |
| DLow [40] | 16 | 26 | 39 | 56 | 36 | 49 | 60 | 79 |
| Ours    | 32        | 44    | 54    | 69    | 74    | 81    | 88    | 99    |

Table 1. **Results of the motion-only action classifier (MOAC) on the Human3.6M and CMU-MoCap datasets**. We compare the classification accuracies for the motions predicted with our method and with the SOTA ones. We also report the accuracies of the oracle, which evaluates the ground-truth future motions, as an upper bound. We report the top-1, top-2, top-3 and top-5 accuracies. The results indicate that the motions predicted by our keypose network are more realistic than those produced by the competing methods.

| Human3.6M | CMU-MoCap |
|-----------|-----------|
|           | gt-pred | pred-gt | average | difference | gt-pred | pred-gt | average | difference |
| TIM-GCN [23] | 0.0069  | 0.0098 | 0.0083 | 0.0029 | 0.0073 | 0.0101 | 0.0087 | 0.0028 |
| HisRep10 [28] | 0.0076  | 0.0129 | 0.0103 | 0.0053 | 0.0061 | 0.0081 | 0.0071 | 0.0020 |
| HisRep125 [28] | 0.0070  | 0.0097 | 0.0083 | 0.0027 | 0.0065 | 0.0093 | 0.0079 | 0.0028 |
| Mix&Match [3] | 0.0067  | 0.0075 | 0.0071 | 0.0008 | 0.0090 | 0.0104 | 0.0097 | 0.0014 |
| DLow [40] | 0.0062  | 0.0080 | 0.0071 | 0.0018 | 0.0069 | 0.0073 | 0.0071 | 0.0008 |
| Ours    | 0.0059  | 0.0061 | 0.0060 | 0.0002 | 0.0057 | 0.0062 | 0.0059 | 0.0005 |

Table 2. **PSKL results** on both the Human3.6M and CMU-MoCap datasets; lower numbers indicate better results. We report the PSKL values between ground truth and predictions (‘gt-pred’) and vice-versa (‘pred-gt’), their average, and their absolute difference. For the multiple prediction methods, Mix&Match, DLow and Ours, we report the best PSKL value, obtained from the predictions that have the most similar power spectrum to the ground truth future motion. We observe that the trend is similar to the MOAC results, with our method outperforming the SOTA.

| diversity ↑ | accuracy ↑ | 1s ave ↓ | 1s best ↓ | 5s ave ↓ | 5s best ↓ |
|-------------|-------------|----------|----------|----------|----------|
| TIM-GCN [23] | -           | 16       | 143      | 143      | 196      | 196      |
| HisRep10 [28] | -           | 21       | 116      | 116      | 197      | 197      |
| HisRep125 [28] | -           | 20       | 136      | 136      | 191      | 191      |
| Mix&Match [3] | 1002       | 18       | 161      | 156      | 244      | 237      |
| DLow [40] | 3501       | 16       | 136      | 136      | 189      | 171      |
| Ours (0.1) | 6936       | 34       | 177      | 168      | 208      | 173      |
| Ours (0.3) | 10328      | 32       | 157      | 138      | 196      | 151      |
| Ours (0.5) | 12362      | 30       | 154      | 125      | 191      | 137      |
| Ours (0.7) | 13491      | 27       | 144      | 118      | 190      | 130      |
| Ours (1.0) | 14995      | 20       | 145      | 116      | 194      | 127      |

Table 3. **Results on the diversity metric, top-1 MOAC accuracy and MPJPE errors** on the Human3.6 dataset. Higher diversity values indicate more variation in the multiple future predictions and lower MPJPE values indicate closer predictions to ground truth future motion. We have highlighted close best results in bold. We provide several results of our method with varying sampling softmax temperature, indicated in parentheses. As the temperature increases, the diversity values of the predictions increase and MPJPE values decrease, however the average top-1 MOAC accuracy begins to decrease as well.

to 0.3, trading a bit of accuracy for more diverse predictions. Our method outperforms the others in having both high diversity, the best average MOAC accuracies, and low MPJPE. For MPJPE, at 1 second we are comparable to the other methods, but at 5 seconds, especially for the “best” MPJPE, our performance is noticeably better.
Note that we present the MPJPE results to give a complete picture but do not believe it to be the best metric for evaluating long-term prediction methods, especially for acyclic motions. Consider, for example, the walking dog action, where the subject, in the middle of the walk, kneels down to pet the dog and stands back up. Our method, in contrast to others, is able to predict the order of these motions, as reflected by our high MOAC score. By contrast, the MPJPE is highly sensitive to the timing of the motions and can be thrown off by slight shifts in timing. For instance, the MPJPE error between two phase shifted sinusoids, or sinusoids of slightly different frequencies, would be high. For such cases, the MPJPE between a flat signal and a sinusoidal might even be lower, but the flat signal would be completely incorrect.

Fig. 4 depicts qualitative results for the discussion, walking dog and walking actions of Human3.6M, and for the directing traffic action of CMU-Mocap. Close visual inspection reveals that, while all methods work reasonably well on cyclic motions such as walking, ours does better on the acyclic ones, such as walking dog. It produces wider motion ranges than the others that tend to predict less dynamic motions. Fig. 5 depicts qualitative results for multiple predictions. Our method is capable of generating diverse, yet still plausible motions.

4.5. Ablation Study on Keypose Retrieval Methods

We evaluate the effect of using keypoints obtained via different strategies: sampling, clustering and ours. The naive-sampling method evenly samples the motion every 15 frames, which is the average keypose duration from our method. The keyposes are then clustered without any keypose pruning. This method also doesn’t require predicting durations, as the duration will always be 15. We also evaluate naive-sampling-pruned, where the keyposes are found through naive sampling, and then pruned. The clustering method performs clustering on every pose in the sequence, rather than only on the poses found via our optimization strategy and pruned afterwards.

As shown in Table 4, our keypose method achieves the highest MOAC accuracies. The comparison with the naive-sampling method emphasizes the importance of having variable-duration keyposes, as opposed to evenly sampling the motion. The comparison with the clustering method emphasizes the importance of optimizing for the keyposes.

| Method                | top-1 | top-2 | top-3 | top-5 |
|-----------------------|-------|-------|-------|-------|
| Naive-sampling        | 28    | 38    | 51    | 67    |
| Naive-sampling-pruned | 30    | 42    | 52    | 63    |
| Clustering            | 24    | 37    | 48    | 66    |
| Ours                  | 32    | 44    | 54    | 69    |

Table 4. Analysis of the method to obtain keyposes. We compare the MOAC accuracies of different keypose methods. Our method achieves higher classification accuracies than the other ones, indicating that the quality of the keyposes affects the performance.

Figure 6. Examples of failure to predict the correct keypose labels. Top: Example from the “posing” category. Once our model detects the extended arms, it switches to the waving motion resembling the poses from the “greeting” category. Bottom: Example from the “sitting” category. Although the leg motion is plausible, our prediction lifts a hand to its head, resembling a motion from the “phoning” category.

5. Conclusion

We have presented an approach to long-term motion prediction. To the best of our knowledge, our work constitutes the first attempt at long-term prediction out to 5 seconds in the future. To this end, we have reformulated motion prediction as a classification problem that guesses in which one of a set of keypose clusters the subject will be. To validate our approach, we have introduced a new action classifier, MOAC, that specifically focuses on the transitions between poses, thus placing an emphasis on the correctness of motion, rather than that of poses. Our experiments show that our method yields more dynamic and realistic poses than state-of-the-art techniques, even when they are tailored to learn patterns for long-term prediction. Furthermore, our approach lets us easily propose multiple possible outcomes.

Altogether, we believe that our approach could be highly beneficial for autonomous systems, such as an autonomous car that needs more than 1 second window into the future to react to pedestrian motions. Furthermore, the ability to sample many alternative future situations can be exploited to aid the motion planning of autonomous systems. Ultimately, long-term and short-term predictions should be used in a complementary manner, the former to produce long-term probabilistic scenarios for better action planning, and the latter to predict fine details in the immediate future.

Acknowledgements. This work was supported in part by the Swiss National Science Foundation.
References

[1] V. Adeli, E. Adeli, I. Reid, J.C. Niebles, and H. Rezatofighi. Socially and Contextually Aware Human Motion and Pose Forecasting. In *International Conference on Intelligent Robots and Systems*, 2020.

[2] E. Aksan, M. Kaufmann, P. Cao, and O. Hilliges. A Spatio-Temporal Transformer for 3D Human Motion Prediction. In *2021 International Conference on 3D Vision (3DV)*, 2021.

[3] S. Aliakbarian, F. S. Saleh, M. Salzmann, L. Petersson, and S. Gould. A Stochastic Conditioning Scheme for Diverse Human Motion Prediction. In *Conference on Computer Vision and Pattern Recognition*, 2020.

[4] E. Barsoum, J. Kender, and Z. Liu. HP-GAN: Probabilistic 3D Human Motion Prediction via GAN. In *Conference on Computer Vision and Pattern Recognition*, 2017.

[5] V. Bloom, V. Argyriou, and D. Makris. Linear Latent Low Dimensional Space for Online Early Action Recognition and Prediction. *Pattern Recognition*, 72:532–547, 2017.

[6] J. Butepage, M.J. Black, D. Kragic, and H. Kjellstrom. Deep Representation Learning for Human Motion Prediction and Classification. In *Conference on Computer Vision and Pattern Recognition*, 2017.

[7] Judith Büttepage, Hedvig Kjellström, and Danica Kragic. Anticipating Many Futures: Online Human Motion Prediction and Generation for Human-Robot Interaction. In *International Conference on Robotics and Automation*, 2018.

[8] J. Büttepage, H. Kjellström, and D. Kragic. Predicting the What and How - A Probabilistic Semi-Supervised Approach to Multi-Task Human Activity Modeling. In *Conference on Computer Vision and Pattern Recognition*, pages 2923–2926, 2019.

[9] CMU Graphics Lab Motion Capture Database, 2010. http://mocap.cs.cmu.edu/.

[10] E. Corona, A. Pumarola, G. Alenyà, and F. Moreno-Noguer. Context-Aware Human Motion Prediction. In *Conference on Computer Vision and Pattern Recognition*, 2020.

[11] C. Diller, T. Funkhouser, and A. Dai. Forecasting Characteristic 3D Poses of Human Actions. In *Conference on Computer Vision and Pattern Recognition*, 2022.

[12] J. Fan, X. Shen, and Y. Wu. Scribble Tracker: A Matting-Based Approach for Robust Tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34:1633–1644, 2012.

[13] K. Fragkiadaki, S. Levine, P. Felsen, and J. Malik. Recurrent Network Models for Human Dynamics. In *International Conference on Computer Vision*, 2015.

[14] P. Ghosh, J. Song, E. Aksan, and O. Hilliges. Learning Human Motion Models for Long-Term Predictions. In *International Conference on 3D Vision*, 2017.

[15] L. Gui, K. Zhang, Y. Wang, X. Liang, J. M. F. Moura, and M. Veloso. Teaching Robots to Predict Human Motion. In *International Conference on Intelligent Robots and Systems*, 2018.

[16] C. Guo, X. Zuo, S. Wang, S. Zou, Q. Sun, A. Deng, M. Gong, and L. Cheng. Action2motion: Conditioned Generation of 3D Human Motions. In *Proceedings of the 28th ACM International Conference on Multimedia (MM ’20)*, 2020.

[17] G. Habibi, N. Jaipuria, and J. P. How. Context-Aware Pedestrian Motion Prediction in Urban Intersections. In *arXiv Preprint*, 2018.

[18] C. Ionescu, I. Papava, V. Olaru, and 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*, 2014.

[19] A. Jain, A.R. Zamir, and S. Savarese A. adn Saxena. Structural-Rnn: Deep Learning on Spatio-Temporal Graphs. In *Conference on Computer Vision and Pattern Recognition*, 2016.

[20] I. Katircioglu, H. Rhodin, J. Spöri, M. Salzmann, and P. Fua. Dyadic Human Motion Prediction. In *arXiv Preprint*, 2022.

[21] S. Kiciroglu, H. Rhodin, S. Sinha, M. Salzmann, and P. Fua. Activemocap: Optimized Viewpoint Selection for Active Human Motion Capture. In *Conference on Computer Vision and Pattern Recognition*, 2020.

[22] L. Kovar, M. Gleicher, and F. Pighin. Motion Graphs. In *ACM SIGGRAPH*, pages 473–482, July 2002.

[23] T. Lebailly, S. Kiciroglu, M. Salzmann, P. Fua, and W. Wang. Motion Prediction Using Temporal Inception Module. In *Asian Conference on Computer Vision*, 2020.

[24] M. Li, S. Chen, Y. Zhao, Y. Zhang, Y. Wang, and Q. Tian. Dynamic Multiscale Graph Neural Networks for 3D Skeleton Based Human Motion Prediction. In *Conference on Computer Vision and Pattern Recognition*, June 2020.

[25] Y. Li, L. Yuan, and N. Vasconcelos. Co-Occurrence Feature Learning from Skeleton Data for Action Recognition and Detection with Hierarchical Aggregation. In *International Joint Conference on Artificial Intelligence*, 2018.

[26] L. Liu, L. Shao, X. Zhen, and X. Li. Learning Discriminative Key Poses for Action Recognition. *IEEE Transactions on Cybernetics*, 43(6):1860–1870, 2013.

[27] F. Lv and R. Nevatia. Single View Human Action Recognition Using Key Pose Matching and Viterbi Path Searching. In *Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007.

[28] W. Mao, M. Liu, and M. Salzmann. History Repeats Itself: Human Motion Prediction via Motion Attention. In *European Conference on Computer Vision*, 2020.

[29] W. Mao, M. Liu, M. Salzmann, and H. Li. Learning Trajectory Dependencies for Human Motion Prediction. In *International Conference on Computer Vision*, 2019.

[30] J. Martinez, M.J. Black, and J. Romero. On Human Motion Prediction Using Recurrent Neural Networks. In *Conference on Computer Vision and Pattern Recognition*, 2017.

[31] R. Morais, V. Le, T. Tran, B. Saha, M. Mansour, and S. Venkatesh. Learning Regularity in Skeleton Trajectories for Anomaly Detection in Videos. In *Conference on Computer Vision and Pattern Recognition*, 2019.

[32] M. Petrovich, M. J. Black, and G. Varol. Action-Conditioned 3D Human Motion Synthesis with Transformer VAE. In *International Conference on Computer Vision*, 2021.
[33] A. H. Ruiz, J. Gall, and F. Moreno-Noguer. Human Motion Prediction via Spatio-Temporal Inpainting. In *International Conference on Computer Vision*, 2019.

[34] L. Shi, L. Wang, C. Long, S. Zhou, M. Zhou, Z. Niu, and G. Hua. Sparse Graph Convolution Network for Pedestrian Trajectory Prediction. In *Conference on Computer Vision and Pattern Recognition*, 2021.

[35] Ian Simon and Sageev Oore. Performance RNN: Generating Music with Expressive Timing and Dynamics, 2017.

[36] Bob Sturm, João Santos, Oded Ben-Tal, and Iryna Korschunova. Music Transcription Modelling and Composition Using Deep Learning. *Conference on Computer Simulation of Musical Creativity*, 2016.

[37] S. Tang, M. Golparvar-Fard, M. Naphade, and M. Gopalakrishna. Video-Based Motion Trajectory Forecasting Method for Proactive Construction Safety Monitoring Systems. *Journal of Computing in Civil Engineering*, 34:04020041, 2020.

[38] A. van den Oord, O. Vinyals, and K. Kavukcuoglu. Neural Discrete Representation Learning. In *Advances in Neural Information Processing Systems*, 2017.

[39] B. Wang, E. Adeli, H.-K. Chiu, D.-A. Huang, and J. C. Niebles. Imitation Learning for Human Pose Prediction. In *International Conference on Computer Vision*, pages 7123–7132, 2019.

[40] Y. Yuan and K. Kitani. Dlow: Diversifying Latent Flows for Diverse Human Motion Prediction. In *European Conference on Computer Vision*, 2020.

[41] J. Zhang and C. Zong. Deep Neural Networks in Machine Translation: An Overview. *IEEE Intelligent Systems*, 30(5), 2015.

[42] Z. Zhao, S. Kiciroglu, H. Vinzant, Y. Cheng, I. Katircioglu, M. Salzmann, and P. Fua. 3D Pose Based Feedback for Physical Exercises. In *arXiv Preprint*, 2022.

[43] Y. Zhou, Z. Li, S. Xiao, C. He, Z. Huang, and H. Li. Auto-Conditioned Recurrent Networks for Extended Complex Human Motion Synthesis. In *International Conference on Learning Representations*, 2018.