Human Motion Prediction via Pattern Completion in Latent Representation Space

Yi Tian Xu[1,2], Yaqiao Li[2] and David Meger[1,2]
[1]*Mobile Robotics Lab*
[2]*School of Computer Science*
McGill University
Montreal, Canada
{*yi.t.xu, yaqiao.li*}*@mail.mcgill.ca, dmeger@cim.mcgill.ca*

***Abstract*—Inspired by ideas in cognitive science, we propose a novel and general approach to solve human motion understanding via pattern completion on a learned latent representation space. Our model outperforms current state-of-the-art methods in human motion prediction across a number of tasks, with no customization. To construct a latent representation for time-series of various lengths, we propose a new and generic autoencoder based on sequence-to-sequence learning. While traditional inference strategies find a correlation between an input and an output, we use pattern completion, which views the input as a partial pattern and to predict the best corresponding complete pattern. Our results demonstrate that this approach has advantages when combined with our autoencoder in solving human motion prediction, motion generation and action classification.**[1]

***Keywords*-Human Motion Prediction, Motion Generation, Action Classification, Pattern Completion, Recurrent Neural Network, Representation Learning**

I. Introduction

Knowledge of how humans move can help intelligent robots in tasks involving an interactive human environment, such as navigating through a crowded street or playing sports and tabletop games with humans. Capturing human motion requires feature detection and tracking, as well as modeling a complex dynamical structure which is highly nonlinear, spontaneous and entangled with physical constraints, intention and high-level semantics. With the arrival of large motion capture databases, such as the Human3.6M dataset [1], and 3D pose estimation algorithms, research has focused on the core patterns present in human motion rather than the distracting visual features. Recently, a series of skeleton-based deep learning methods have greatly increased the performance for human motion prediction, while introducing increasingly specialized and sophisticated model designs.

We attack skeleton-based human motion prediction using a *general* representation learning approach [2] without relying on specialized architectures or external knowledge on the data structure. Our method consists of two coordinated steps. First, we learn a latent representation space using a hierarchical sequence-to-sequence (Seq2Seq) architecture, to reveal the underlying structure in the complex Human3.6M dataset. Then, we use the learned representations for motion prediction and for two related tasks: motion generation and action classification, through a process called *pattern completion* [3]. The latter is the core idea in our method; instead of viewing inference as finding the correlation between an input and an output, we view the input as a partial pattern that is to be completed.

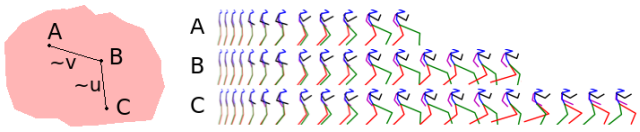


Figure 1: By encoding motion sequences into a well-structured latent space, we are able to complete the pattern A to a pattern that approximates B by simply using vector addition, e.g. $A + v \approx B$, where v is a vector that can be directly computed. The process is recursive and can be repeated to extend to motion C.

As proposed in situated conceptualization in cognitive science [3], pattern completion can support diverse forms of intelligent tasks and provides an important grounding of new situations into experienced situations, rendering structure to the latent representations. To our knowledge, we are the first to attempt this conceptual contribution in the domain of human motion understanding. See an illustration of pattern completion in Fig. 1.

In summary, our main contributions are:

1) We propose a new and generic autoencoder that uses a hierarchical Seq2Seq structure to construct latent representations of time-series of various lengths while maintaining a well-structured embedding.
2) We implement pattern completion on the latent representation space with either a single-layer network or vector addition for human motion prediction, motion generation and action classification. We often achieve high performance.

Our results show competitive and sometimes higher performance compared to state-of-the-art methods in human motion prediction. In particular, our method outperforms the

state of the arts for some aperiodic actions in Human3.6M such as *greeting*, *sitting* and *taking photo*, which are notoriously challenging. Furthermore, the representations learned by our method also allow for the motion generation and action classification tasks to be performed effectively.

II. Related works

A. Human motion prediction with Human3.6M

The Human3.6M dataset [1] is one the largest and most challenging benchmarks to evaluate human motion understanding. Its large variety of poses are recorded from 7 professional actors doing 15 activities, including walking, eating, smoking and engaging in a discussion. Due to the stochastic nature of human movement, previous authors have separated motion prediction into two sub-tasks: short-term and long-term prediction. Short-term prediction is commonly compared quantitatively with mean angle error, while long-term predictions are usually assessed qualitatively. This is because even human intelligence is not able to uniquely determine the motion of a character tens of seconds into the future but rather only capture a sampling of plausible outcomes.

Recurrent neural networks (RNN) are commonly used to solve both short-term and long-term prediction. Earlier works [4], [5] suffer from a noticeable discontinuity between the end of the input (last observed frame) and the first predicted frame. Martinez et al. [6] alleviates this problem using a Seq2Seq model with a residual connection, boosting the performance for short-term prediction.

Recent works focus on long-term prediction which often collapses to an undesired common pose, especially for aperiodic motions. This failure is possibly caused by the common use of mean squared error (MSE), a high-tailed loss that discourages making risky predictions [6]. Additionally, MSE, as well as other traditional losses, treat each joint with equal weight. In reality, the impact of joints on any given motion is non-uniform. This motivates many complex or data-specialized loss function. Pavllo et al. [7] propose to perform Forward Kinematics during training to compute the loss in Cartesian space. Gui et al. [8] introducing geodesic loss to capture the geometric structure of the angles. Several authors [7], [9], [10] also propose to tackle the short-term and long-term prediction tasks separately, or to have two model variations, each adapted and optimized for one of the tasks.

Furthermore, RNNs of all flavors, including both Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU), are hypothesized to have difficulty in keeping track of long-term information [11] and spatial correlation (e.g. between left and right arms) [10]. Tang et al. [11] approach this problem by adding an attention unit and their Modified Highway Unit (MHU) to summarize motion history and to focus on joints with large motions. Li et al. [10] use convolutional filters to learn spatio-temporal dependencies.

In our work, we design a single model to address both short-term and long-term prediction without additional data-specialized or task-specialized architectures. We also employ a Hierarchical Seq2Seq architecture [12] to avoid the loss of long-term information.

B. Representation learning methods

Several deep learning methods include objectives that encourage reproduction of the input data, including autoencoders, Variational Autoencoders (VAE) [13] and Generative Adversarial Networks (GAN) [14]. These approaches are known to produce a latent space that has certain meaningful structures. Nearby latent representations are similar in the original data representation space. Thus, interpolation on the latent space usually produces smooth transitions on the original space, and clustering on the latent space often produces semantically meaningful groups. In particular, successful distributional semantics models in NLP [15]–[17] demonstrate the additive compositionality property, enabling the word analogy task to be solvable using vector addition on the latent space.

Previous works in human motion representation learning such as [18], [19] show how the learned representation can be used for various tasks such as fixing corrupted data or performing action classification. However, to our knowledge, none of them demonstrate performance comparable to the current state of the art in human motion prediction.

III. Method

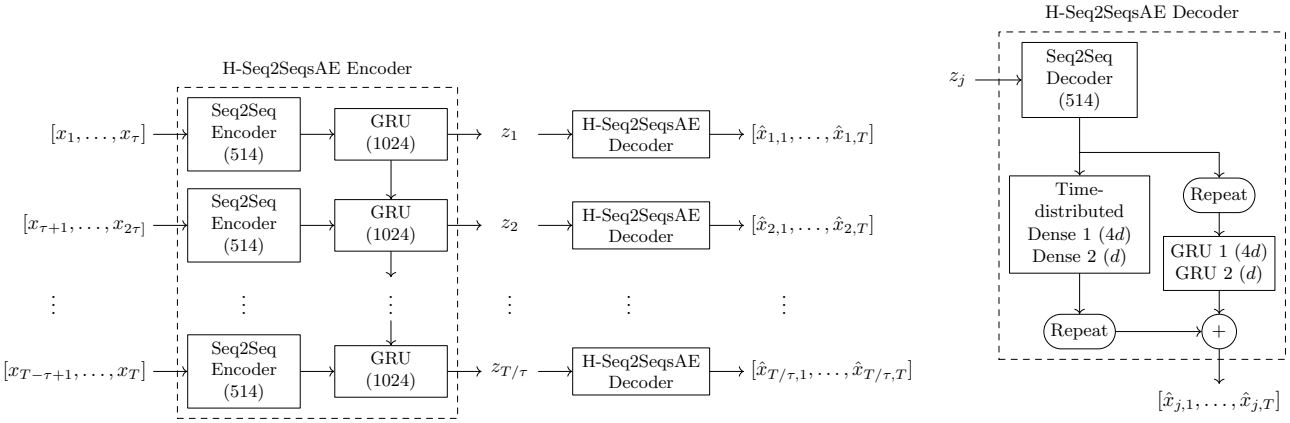
Our method consists of two steps: representation learning and pattern completion. In the representation learning step, we learn encoding and decoding functions E and D, which map observations to corresponding latent representations, and back. In our novel pattern completion step, we train a pattern completion function G to map inputs to their most sensible full patterns in the well-structured latent space. To make overall predictions, we apply the function $D \circ G \circ E$ to encode, complete and decode. Specifically, given an input data pair (X, Y), the function G aims to predict the latent representation of XY from the latent representation of X. This is in contrast to the common practice that predicts Y directly from X. Below, we explain the details of our method.

Let $S = \{(X_i, Y_i)\}_{i=1}^N$ be a given dataset of input and ground truth pairs. For the motion prediction task, X_i is the beginning of a motion that is observed in order to complete the unseen portion Y_i. Each X_i and each Y_i are represented as a sequence of joint angles. We wish to learn a prediction function F such that

$$\{X_i\}_{i=1}^N \xrightarrow{F} \{\widehat{Y}_i\}_{i=1}^N \tag{1}$$

where the difference between Y_i and $\widehat{Y}_i$ is minimized.

In the representation learning step, we construct $S' = \{X_i, X_iY_i\}_{i=1}^N$ where X_iY_i denotes X_i concatenated with



(a) H-Seq2SeqsAE architecture.

(b) Details for the decoder.

Figure 2: An illustration of our Hierarchical Sequence-to-Sequence Autoencoder (H-Seq2SeqsAE). We use common Seq2Seq encoder and decoder [20] in our model. The Repeat unit in (b) takes input a sequence of length T/τ and outputs a sequence of length T, where each element from the input is repeated consecutively τ times. The output dimensionalities are specified in parenthesis. When training for $T = 60$, we use 1500 dimensions for the gated recurrent unit (GRU) in (a). In (b), d denotes the number of features. Note that in (a), the weights are shared across rows.

Y_i. We learn an encoding and a decoding function, E and D, respectively, such that E maps every element in S' to a latent representation in some lower-dimensional space $Z \subseteq \mathbb{R}^n$,

$$S' \xrightarrow{E} Z \xrightarrow{D} \widehat{S}'. \tag{2}$$

In the pattern completion step, we view $E(X_i)$ as a partial pattern and $E(X_iY_i)$ as a complete pattern, respectively, and use the pre-trained E and D to learn a *pattern completion* function G on the representation space. Specifically, let $P = \{p_i = E(X_i)\}_{i=1}^N$ be the set of partial patterns, and $C = \{c_i = E(X_iY_i)\}_{i=1}^N$ be the set of complete patterns. Note that $P, C \subseteq Z$. We learn a function G such that

$$P \xrightarrow{G} \widehat{C}, \tag{3}$$

where the difference between c_i and $\widehat{c_i}$ is minimized. That is, we wish to predict the complete pattern. Finally we obtain $F = D \circ G \circ E$. That is,

$$\{X_i\}_{i=1}^N \xrightarrow{E} P \xrightarrow{G} \widehat{C} \xrightarrow{D} \{\widehat{X}_i\widehat{Y}_i\} \subseteq \widehat{S}'. \tag{4}$$

Note that this step reflects our choice to map X_i to $\widehat{X}_i\widehat{Y}_i$ as suggested by pattern completion in cognitive science [3], rather than to map X_i to $\widehat{Y}_i$ directly as in standard learning methods. While many tasks only require $\widehat{Y}_i$, we show that training for completion yields improved performance (see Section IV-G). To distinguish between completing X with XY and matching X to Y, we call the latter *pattern matching*.

To this point, our discussion has treated the input motion data as simple vectors, but it is crucial to capture their time series nature. In order for E to encode sequences of different lengths, we modify the standard autoencoding framework so that subsequences of the input sequence can be mapped to latent representations. Section III-A presents the details of our autoencoder.

One crucial assumption in our method is that after learning E and D, G can be modeled with a much simpler function than completing in raw space and can be learned in a short amount of time. In our experiments, we model G using either a single dense layer network or vector addition. In Section III-B, we describe how G can be implemented using vector addition.

A. Hierarchical Sequence-to-Sequences Autoencoder (H-Seq2SeqsAE)

Our model takes a sequence as its input and outputs multiple sequences, each corresponding to a reconstruction of a subsequence from the original input. It is based on Hierarchical Seq2Seq model [12] in order to avoid losing long-term historical information [10], [11]. We name our autoencoder *Hierarchical Sequence-to-Sequences Autoencoder (H-Seq2SeqsAE)*.

More specifically, let T be the length of an input sequence and τ be a divisor of T. Given a sequence of joint angles $X = [x_1, x_2, \ldots, x_T]$, we partition X into T/τ subsequences $[x_1, \ldots, x_\tau], \ldots, [x_{T-\tau+1}, \ldots, x_T]$.

For the encoder, our model first obtains sub-encodings for these subsequences using a standard Seq2Seq encoder [20]. Next, the T/τ sub-encodings are fed into a higher-level encoder which outputs T/τ encodings such that the jth encoding, z_j, corresponds to $[x_1, \ldots, x_{j\tau}]$. See Fig. 2a.

For the decoder, we modify the Hierarchical Seq2Seq decoder [12] using residual connections, which were shown to improve the performance in [6], [7]. Given z_j, we first apply the standard Seq2Seq decoder to obtain a sequence of length T/τ. Each element in the sequence is then passed to two dense layers to obtain a pose. Another pathway leads the

entire sequence to two RNNs to obtain T residuals. Finally, the decoder outputs the combined poses and residuals. See Fig. 2b.

Since an important part of our decoder is based on the residual angles, a natural output for our model when autoencoding a subsequence of length $j\tau < T$ is zero motion after the $j\tau^{th}$ frame. Therefore, we use the following loss function

$$\frac{1}{T/\tau}\sum_{j=1}^{T/\tau} l([x_1, x_2, \ldots, x_{j\tau}, x_{j\tau}, \ldots, x_{j\tau}], [\hat{x}_{j,1}, \ldots, \hat{x}_{j,T}]), \tag{5}$$

in which $[x_1, x_2, \ldots, x_{j\tau}, x_{j\tau}, \ldots, x_{j\tau}]$ is also a sequence of length T. We encode the moment when a motion stops rather than the exact length of the sequence. We use mean absolute error (MAE) for l as it does not pose specific assumptions or constraints on the data format, and we find that it performs better compared to MSE in our method.

Finally, we define our functions E and D as follows. Given input sequence X of length $j\tau \leq T$, we construct X' by appending X with placeholders to reach length T. We feed X' to our H-Seq2SeqsAE encoder and take the j^{th} output from it as the output for E. Note this allows E to take input of various lengths: $\tau, 2\tau, \ldots, j\tau, \ldots, T$. For D, we define it as the H-Seq2SeqsAE decoder.

B. Pattern completion using vector addition

The emerging structure in the latent representation space allows for simple and intuitive vector addition to accurately predict human motion. See Fig. 1 for an illustration of this operation at work.

Given a set of input and ground truth pairs $S_j = \{(X, Y) : |X| = j\tau, |XY| = T\}$ for some j. We define

$$d_j(X, Y) = E(XY) - E(X). \tag{6}$$

For an arbitrary input sequence X' such that $|X'| = j\tau$ and $|X'Y'| = T$, we simply use the following vector addition

$$E(X') + v_j \tag{7}$$

to approximate $E(X'Y')$, where

$$v_j = \frac{1}{|S_j|}\sum_{(X,Y)\in S_j} d_j(X, Y). \tag{8}$$

In other words, v_j is the average difference between the latent representations of all X and XY seen in S_j. As we observe, the variance of $d_j(X, Y)$ is low (see Section IV-G), and each v_j can be computed using a small sample (e.g. using 1000 samples as in Fig. 3) to obtain high quality results.

Such additive relationship between X and XY in our latent representation space is analogous to the additive compositionality defined by Mikolov et al. [16]. As our H-Seq2SeqsAE captures the robust features of X and XY, we find a stronger correlation between $E(X)$ and $E(XY)$ than between $E(X)$ and $E(Y)$ (as shown in Section IV-G).

C. Action classification and label recovery

To include action label information, we concatenate a one-hot encoded action type vector with each pose, similar to recent literature [6], [8], [21]. With the action label and human motion learned by our autoencoder, this knowledge can be used to solve the action classification task. We apply our pattern completion method to action classification in two variations.

In the first variation, our H-Seq2SeqsAE learns to encode both the supervised and unsupervised motion sequences, and the action label itself. To achieve this, at each epoch of the training, we randomly choose a third of the data and set the label vector to zero. Another third is randomly chosen with the poses set to zeros.

In the second variation, our H-Seq2SeqsAE learns to encode the supervised motion sequences. Classification by this variation is, therefore, more similar to fixing corrupted data or filling missing information, thus we call it label recovery.

IV. Experiments

We trained our method on the Human3.6M dataset [1] for each of the following three tasks: (1) short-term motion prediction, (2) long-term motion prediction and motion generation, and (3) action classification and label recovery. For each, we perform pattern completion with both a forward neural network with a single dense layer (FN) and vector addition (ADD). We distinguish motion generation from long-term motion prediction by requiring a generative model to be able to output multiple different valid results. We measure the performance of all short-term prediction methods using the community standard metric: mean joint angle error.

A. Baselines

We follow the same evaluation method for short-term prediction as in [4]–[8], [10], [11], [21]. We cite the results from the most relevant works to compare with our method, which are Res-Seq2Seq [6], the model by Tang et al. [11], VGRU-rl [21] and AGED [8] which is the current state of the art. We also compare against the naive zero-velocity baseline proposed by [6] and use their code to generate long-term predictions.

B. Data preprocessing

Following the same settings as our baselines, we down-sample the dataset from 50 to 25 frames per second (fps), and use subject 5 for testing and the rest for training. Joint angles with small standard deviation are ignored, resulting in an input size of 54.

We use two normalization methods depending on the baseline that we are comparing against: (1) subtract the mean and normalized between -1 and 1, which is used in [11], and (2) subtract the mean and divide by the standard deviation, which is used by the other methods [6], [8], [21].

Table I: Comparison of mean angle error between our method and top performing baselines for short-term motion prediction. The "Average*" column is the average error over all 15 actions.

(a) Short-term prediction with 30 input frames and normalized angles between -1 and 1.

	Walking				Eating				Smoking				Discussion				Average*			
miliseconds	80	160	320	400	80	160	320	400	80	160	320	400	80	160	320	400	80	160	320	400
Zero-velocity	0.39	0.68	0.99	1.15	0.27	0.48	0.73	0.86	**0.26**	0.48	0.97	0.95	0.31	0.67	0.94	1.04	0.40	0.71	1.07	1.21
Tang et al. [11]	0.32	0.53	0.69	0.77	-	-	-	-	-	-	-	-	0.31	0.66	0.97	1.04	0.39	0.68	1.01	1.13
Ours-ADD ($T = 40$)	0.37	0.51	0.77	0.90	0.32	0.44	0.70	0.82	0.36	0.54	1.02	0.96	0.40	0.72	1.09	1.21	0.50	0.74	1.09	1.21
Ours-FN ($T = 40$)	**0.21**	**0.33**	**0.54**	**0.61**	**0.20**	**0.31**	**0.53**	**0.67**	0.28	**0.47**	**0.83**	**0.86**	**0.30**	**0.61**	**0.84**	**0.94**	**0.35**	**0.59**	**0.92**	**1.06**

(b) Short-term prediction with 50 input frames and normalized angles by the standard deviation.

	Walking				Eating				Smoking				Discussion				Average*			
miliseconds	80	160	320	400	80	160	320	400	80	160	320	400	80	160	320	400	80	160	320	400
Zero-velocity	0.39	0.68	0.99	1.15	0.27	0.48	0.73	0.86	**0.26**	0.48	0.97	0.95	0.31	0.67	0.94	1.04	0.40	0.71	1.07	1.21
Res-Seq2Seq (sup.) [6]	0.28	0.49	0.72	0.81	0.23	0.39	0.62	0.76	0.33	0.61	1.05	1.15	0.31	0.68	1.01	1.09	0.36	0.67	1.02	1.15
VGRU-rl [21]	0.34	0.47	0.64	0.72	0.27	0.40	0.64	0.79	0.36	0.61	0.85	0.92	0.46	0.82	0.95	1.21	-	-	-	-
AGED (w/o adv) [8]	0.28	0.42	0.66	0.73	0.22	0.35	0.61	0.74	0.30	0.55	0.98	0.99	0.30	0.63	0.97	1.06	0.32	0.62	0.96	1.07
AGED [8]	**0.22**	**0.36**	**0.55**	0.67	**0.17**	**0.28**	**0.51**	**0.64**	0.27	**0.43**	**0.82**	0.84	**0.27**	**0.56**	**0.76**	**0.83**	**0.31**	**0.54**	**0.85**	**0.97**
Ours-ADD ($T = 60$)	0.30	0.45	0.74	0.88	0.21	0.37	0.65	0.78	0.31	0.50	0.95	0.89	0.33	0.65	0.91	1.03	0.38	0.64	0.99	1.12
Ours-FN ($T = 60$)	0.29	**0.36**	0.57	**0.64**	0.24	0.32	0.52	0.67	0.36	0.51	0.85	**0.83**	0.33	0.60	0.84	0.95	0.41	0.62	0.92	1.03

C. Training Procedure

For the representation learning step, we use gated recurrent unit (GRU) for our H-Seq2SeqsAE. For the higher-level encoder, we use $tanh$ activation. For the rest, we use $tanh$ activation when the data is normalized between -1 and 1, otherwise, we use $linear$ activation. We train using Nadam optimizer with a learning rate of $8e{-}4$ and a decay rate of $4e{-}3$. We use a batch size of 64, 5-fold cross-validation and $1e4$ samples per epoch, for 300 epochs.

Except Tang et al. [11] which uses 30 input time-steps and no label information, all compared methods have reported results for short-term prediction using 50 input time-steps, 10 output time-steps and appended action label information. Hence, we train two H-Seq2Seqs-AE, one with $T = 40$ and 1024 latent dimensions, the other with $T = 60$, 1500 latent dimensions and label information. We use $\tau = 10$ for both. Fig. 3 shows an example of a training curve for the latter. The latter model is also used for long-term prediction and motion generation.

For the pattern completion step with a single dense layer network, we train such network to map encodings of sequences of length 30 or 50 to their corresponding encodings of sequences of length 40 or 60, respectively, for short-term prediction. For long-term prediction and motion generation, we train with 10 input time-steps and 50 output time-steps. We use the same settings as for training H-Seq2SeqsAE, except a faster step-decay with a rate of 0.5 and 50 epochs.

For action classification, we train a separate H-Seq2Seqs-AE with $T = 40$, $\tau = 10$, 1024 latent dimensions and label information with only walking and sitting actions. Our pattern completion function maps unlabeled to labeled motions of length 40.

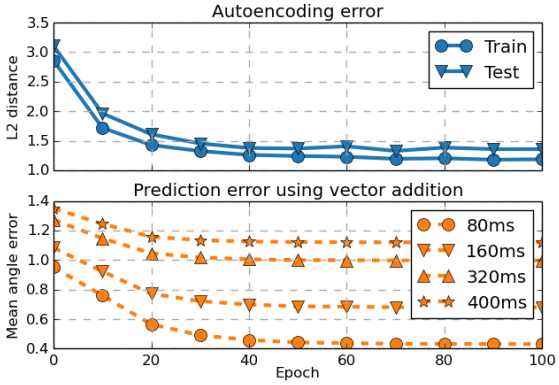


Figure 3: Convergence of autoencoding error and short-term prediction error is observed when training H-Seq2SeqsAE with $T = 60$ and $\tau = 10$. Vector addition is used as the pattern completion function for this prediction task, and it is computed using 1000 training samples. Note that the autoencoding error includes the global rotations and transitions while the prediction error does not.

D. Short-term motion prediction

Table I shows our results for short-term prediction compared against baseline methods. We observe that our method is better than all approaches that utilize stationary loss functions. This includes the core approach of the AGED [8] method. However, the unique addition of adversarial loss within that method has led to boosted performance – a feature we have not yet implemented in our method, but which could easily augment the core improvements demonstrated here.

Our method sometimes has difficulty in the first few output time-steps. This is expected since we are adding the residual angle to reconstructed poses rather than the last

Table II: Short-term motion prediction for some notoriously challenging aperiodic actions where our method outperforms all baselines.

(a) With 30 input frames and normalized angles between -1 and 1.

	Greeting				Sitting				Taking Photo			
miliseconds	80	160	320	400	80	160	320	400	80	160	320	400
Zero-velocity	0.54	0.89	1.30	1.49	**0.40**	0.63	1.02	1.18	0.25	0.51	0.79	0.92
Tang et al. [11]	0.54	0.87	1.27	1.45	-	-	-	-	0.27	0.54	0.84	0.96
Ours-ADD	0.57	0.85	1.26	1.44	0.49	0.67	1.01	1.16	0.29	0.49	0.74	0.87
Ours-FN	**0.40**	**0.69**	**1.11**	**1.28**	**0.40**	**0.58**	**0.90**	**1.09**	**0.24**	**0.45**	**0.67**	**0.77**

(b) With 50 input frames and normalized angles by the standard deviation.

	Greeting				Sitting				Taking Photo			
miliseconds	80	160	320	400	80	160	320	400	80	160	320	400
Zero-velocity	0.54	0.89	1.30	1.49	0.40	0.63	1.02	1.18	0.25	0.51	0.79	0.92
Res. [6]	0.75	1.17	1.74	1.83	0.41	1.05	1.49	1.63	0.24	0.51	0.90	1.05
AGED (w/o adv.) [8]	0.61	0.95	1.44	1.61	0.46	0.87	1.23	1.51	0.24	0.52	0.92	1.01
AGED [8]	0.56	0.81	1.30	1.46	0.41	0.76	1.05	1.19	**0.23**	0.48	0.81	0.95
Ours-ADD	0.47	0.78	1.21	1.40	**0.37**	**0.58**	**0.94**	**1.10**	**0.23**	**0.46**	**0.69**	0.80
Ours-FN	**0.46**	**0.74**	**1.14**	**1.34**	0.43	0.62	**0.94**	**1.10**	0.31	0.49	**0.69**	**0.79**

pose from the original input sequence as done in [6], [8]. However, our method excels for longer temporal horizons. We also observe that ADD outperforms FN on the first few predicted frames when the data is normalized by the standard deviation.

One advantage of our model is that it can capture the structure in several aperiodic motions better than our baselines. Prior works have difficulty in modeling complicated and highly stochastic motions that even the zero-velocity baseline can easily outperform them, as observed by Martinez et al. [6]. The adversarial loss in AGED [8] also leads to a significant performance improvement in aperiodic motions, but here we outperform them all, as can be seen in the tasks greeting, sitting and taking photo (see Table II).

E. Long-term motion prediction and generation

Fig. 4 shows our qualitative results for long-term motion prediction and motion generation with outputs of 50 time-steps compared with the long-term predictions by [6]. For these tasks, we use the same model that resulted Table Ib, and the input sequence length is set to 10 time-steps rather than 50 time-steps. Martinez et al. [6] propose the hypothesis that using MSE as the loss function forces the prediction to converge to a mean pose. Although our MAE loss has similar theoretical properties to MSE, we observe that our method can produce more plausible motions over a longer time horizon, even for aperiodic actions like greeting. We also observe that ADD often generates motionless sequences, albeit it can preserve some general structure of the motions.

To obtain diverse generated solutions, we demonstrate that we can generate different motion sequences by adding noise to the output of our forward network. The amount of noise is computed using the standard deviation of the distance d (see Section III-B). This results in a slight variation for our long-term prediction (see Fig. 4), but the latter is less smooth and may contain unnatural poses.

Our method can also demonstrate motion generation using interpolation on the latent representation space. In Fig. 5, given a walking and a sitting motion, we generate 8 motion sequences between the two, which can be combined to create smooth and realistic motion of a person sitting down.

Animations and quantitative evaluations of our results are available on our project webpage[2].

F. Action classification and label recovery

To our knowledge, we are the first to perform action classification using solely the Human3.6M dataset. Prior works on skeleton-based action classification either use other datasets entirely [19], [22] or combine Human3.6M with additional data [23] for training. Human3.6M is a difficult dataset for action classification due to the large variety of poses and motions that overlap between the action categories. Therefore, we choose to select only two actions to perform action classification and label recovery on: walking and sitting.

Table III shows our results compared with three simple baseline methods. Our results show that our method demonstrates a high performance improvement compared to our baselines. We also observe that our performance in label recovery is slightly lower. This reflects the difference in performance between seeing and not seeing the motions without labels during training.

G. Ablation studies

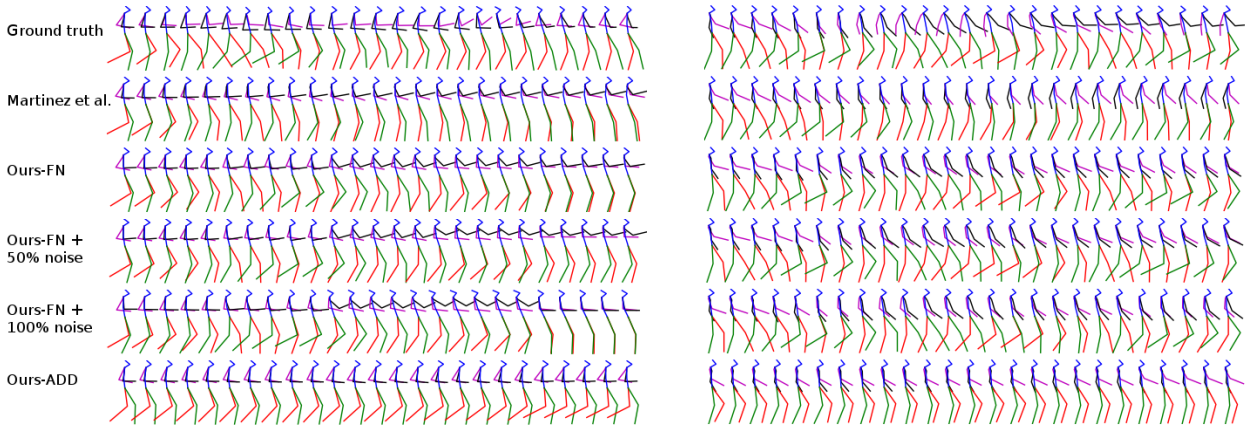
The core idea in our method is that the latent representation of the input can be completed to get a solution to a task. In Table IV, we compare pattern completion and pattern matching for short-term prediction. Note that for pattern matching, the distance measure from Section III-B becomes

$$d'_j(X, Y) = E(Y) - E(X). \tag{9}$$

We observe that the performance boosts when applying pattern completion for both ADD and FN. The average standard deviation of d_j (used in pattern completion) is also smaller than d'_j. Our results also show that our pattern completion approach outperforms a standard hierarchical sequence-to-sequence (H-Seq2Seq) model with encoder and decoder similar to the ones in our H-Seq2SeqsAE, while our autoencoder combined with pattern matching cannot. This demonstrates the advantage of pattern completion over pattern matching in our method.

Furthermore, Table IV also compares our H-Seq2Seqs-AE with a basic Hierarchical Seq2Seq autoencoder (Basic) that pads input sequences with the last input frame to input varying length sequences. Our results suggest that our autoencoder results more effective latent representation for pattern completion, since the basic autoencoder cannot distinguish between long sequences with no motion towards the end and their shorter counterparts.

[2]http://www.cim.mcgill.ca/~yxu219/human-motion-prediction.html



(a) Greeting. (b) Walking.

Figure 4: Long-term motion prediction and generation of 50 time-steps for greeting and walking. Note that we skip every second frame.

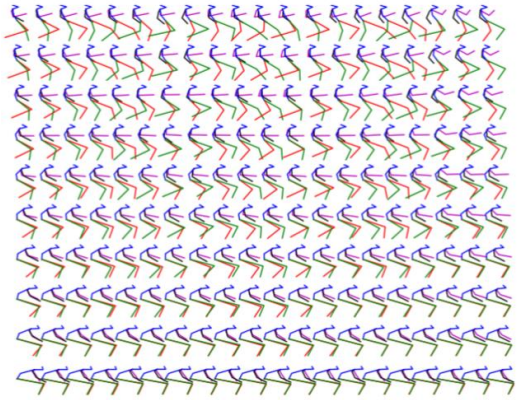


Figure 5: Interpolation between walking (top row) and sitting (bottom row). Each row represents a generated motion sequence using our model. Each column appears to be a smooth and realistic motion of a person sitting down.

Table III: Comparison of the average predicted probability between our method and our baselines for classification of walking and sitting. Note that our method outputs a valid probability vector for two classes. Our baselines are Last+Dense: a single dense layer network trained to classify based on the last pose of the input motion, Flatten+Dense: a single dense layer network trained to classify based on all poses of the input motion, and GRU+Dense: a GRU unit connected with a dense layer.

		Walking	Sitting
	Last+Dense	0.61	0.61
	Flatten+Dense	0.61	0.61
	GRU+Dense	0.61	0.61
Label recovery	Ours-ADD	0.55	0.52
	Ours-FN	**0.72**	**0.71**
Action classification	Ours-ADD	0.63	0.69
	Ours-FN	**0.73**	**0.74**

V. Discussion

We have presented our Hierarchical Sequence-to-Sequences Autoencoder (H-Seq2SeqsAE), a new and generic representation learning model for time-series data of various lengths. Combined with our novel pattern completion approach, we have shown in the context of skeleton-based human motion, that the learned representations enable short-term and long-term motion prediction, motion generation, action classification and label recovery with high quality. In particular, our performance in short-term prediction is competitive with state of the arts and outperforms in certain aperiodic actions. Why can our model attain such performance without specialized architectures and external knowledge that other state-of-the-art methods use?

One possible explanation is: for some complicated and diverse data such as the Human3.6M dataset [1], representation learning can extract important and robust features that are very suitable to the pattern completion approach. Although the lower-dimensional latent space potentially provides less information to our forward network or to vector addition during the pattern completion step, the structure gain through the autoencoding process results in a simpler learning problem from the completion perspective: given input and ground truth pairs $\{(X_i, Y_i)\}_{i=1}^{N}$, robust features in X_i are also present in X_iY_i. The pattern completion approach on the latent space captures these cues, stabilizing learning and allowing stronger connections across the implied sequence.

The impact of the hyperparameter τ can be further studied. Aside from understanding the properties of our latent representation space, future works also include improving H-Seq2SeqsAE, finding more effective representation learning models suitable for pattern completion and applications in other domains.

Table IV: Ablation analysis on the performance difference between pattern completion and pattern matching, and our H-Seq2Seqs-AE and a basic Seq2Seq autoencoder (Basic). X_{30} denotes an input sequence of length 30, and Y_{10}, an output sequence of length 10.

(a) With $T = 40$ and normalized angles between -1 and 1.

		Average				Mean STD	STD STD
	miliseconds	80	160	320	400		
$X_{30} \longrightarrow Y_{10}$	H-Seq2Seq	0.47	0.68	0.95	1.06	-	-
$X_{30} \longrightarrow X_{30}Y_{10}$		0.57	0.73	0.97	1.07		
$X_{30} \longrightarrow X_{30}Y_{10}$	Basic-ADD	0.39	0.66	0.98	1.11	0.003	0.003
	Basic-FN	0.38	0.62	0.93	**1.04**		
$X_{40} \longrightarrow Y_{10}$	Ours-ADD	1.90	1.84	1.76	1.72	0.017	0.006
	Ours-FN	0.59	0.78	1.08	1.18		
$X_{30} \longrightarrow Y_{10}$	Ours-ADD	1.75	1.72	1.59	1.50	0.015	0.005
	Ours-FN	0.41	0.65	0.99	1.11		
$X_{30} \longrightarrow X_{30}Y_{10}$	Ours-ADD	0.50	0.74	1.09	1.21	0.003	0.006
	Ours-FN	**0.35**	**0.59**	**0.92**	1.06		

(b) With $T = 60$ and normalized angles by the standard deviation.

		Average				Mean STD	STD STD
	miliseconds	80	160	320	400		
$X_{50} \longrightarrow Y_{10}$	H-Seq2Seq	0.51	0.70	0.98	1.09	-	-
$X_{50} \longrightarrow X_{50}Y_{10}$		0.66	0.79	1.01	1.10		
$X_{50} \longrightarrow X_{50}Y_{10}$	Basic-ADD	0.41	0.67	0.99	1.12	0.015	0.007
	Basic-FN	0.45	0.67	0.95	1.07		
$X_{50} \longrightarrow Y_{10}$	Our-ADD	1.73	1.79	1.88	1.91	0.027	0.051
	Our-FN	0.54	0.72	0.98	1.10		
$X_{50} \longrightarrow X_{50}Y_{10}$	Our-ADD	**0.38**	0.64	0.99	1.12	0.018	0.007
	Our-FN	0.41	**0.62**	**0.92**	**1.03**		

REFERENCES

[1] 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 transactions on pattern analysis and machine intelligence*, vol. 36, no. 7, pp. 1325–1339, 2014.

[2] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[3] L. W. Barsalou, "Situated conceptualization," in *Handbook of categorization in cognitive science*. Elsevier, 2005, pp. 619–650.

[4] K. Fragkiadaki, S. Levine, P. Felsen, and J. Malik, "Recurrent network models for human dynamics," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 4346–4354.

[5] A. Jain, A. R. Zamir, S. Savarese, and A. Saxena, "Structural-rnn: Deep learning on spatio-temporal graphs," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 5308–5317.

[6] J. Martinez, M. J. Black, and J. Romero, "On human motion prediction using recurrent neural networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017, pp. 4674–4683.

[7] D. Pavllo, D. Grangier, and M. Auli, "Quaternet: A quaternion-based recurrent model for human motion," *arXiv preprint arXiv:1805.06485*, 2018.

[8] L.-Y. Gui, Y.-X. Wang, X. Liang, and J. M. Moura, "Adversarial geometry-aware human motion prediction," in *ECCV*, 2018, pp. 823–842.

[9] X. Lin and M. R. Amer, "Human motion modeling using dvgans," *arXiv preprint arXiv:1804.10652*, 2018.

[10] C. Li, Z. Zhang, W. S. Lee, and G. H. Lee, "Convolutional sequence to sequence model for human dynamics," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5226–5234.

[11] Y. Tang, L. Ma, W. Liu, and W. Zheng, "Long-term human motion prediction by modeling motion context and enhancing motion dynamic," *arXiv preprint arXiv:1805.02513*, 2018.

[12] J. Li, M.-T. Luong, and D. Jurafsky, "A hierarchical neural autoencoder for paragraphs and documents," *arXiv preprint arXiv:1506.01057*, 2015.

[13] C. Doersch, "Tutorial on variational autoencoders," *arXiv preprint arXiv:1606.05908*, 2016.

[14] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[15] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[16] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013, pp. 3111–3119.

[17] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.

[18] D. Holden, J. Saito, T. Komura, and T. Joyce, "Learning motion manifolds with convolutional autoencoders," in *SIGGRAPH Asia 2015 Technical Briefs*. ACM, 2015, p. 18.

[19] J. Bütepage, M. J. Black, D. Kragic, and H. Kjellström, "Deep representation learning for human motion prediction and classification," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017, p. 2017.

[20] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in neural information processing systems*, 2014, pp. 3104–3112.

[21] A. Gopalakrishnan, A. Mali, D. Kifer, C. L. Giles, and A. G. Ororbia, "A neural temporal model for human motion prediction," *arXiv preprint arXiv:1809.03036*, 2018.

[22] L. L. Presti and M. La Cascia, "3d skeleton-based human action classification: A survey," *Pattern Recognition*, vol. 53, pp. 130–147, 2016.

[23] D. C. Luvizon, D. Picard, and H. Tabia, "2d/3d pose estimation and action recognition using multitask deep learning," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, 2018.