The Pennsylvania State University
The Graduate School
College of Engineering

NEURAL TEMPORAL MODELS FOR HUMAN MOTION PREDICTION

A Thesis in
Electrical Engineering
by
Anand Gopalakrishnan

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Master of Science

May 2019

The thesis of Anand Gopalakrishnan was reviewed and approved* by the following:

C. Lee Giles
Professor of Information Science & Technology
Thesis Co-Advisor

David J. Miller
Professor of Electrical Engineering
Thesis Co-Advisor

Minghui Zhu
Assistant Professor of Electrical Engineering

Kultegin Aydin
Professor of Electrical Engineering and Department Head

*Signatures are on file in the Graduate School.

Abstract

This work proposes novel neural temporal models for predicting and synthesizing human motion, achieving state-of-the-art in modeling long-term motion trajectories while being competitive with prior work in short-term prediction, with significantly less computational expense. Key aspects of the proposed system include: 1) a novel, two-level processing architecture that helps in generating "guiding" trajectories, 2) a set of easily computable features that incorporate motion derivative information into the model, and 3) a novel multi-objective loss function that helps the model to incrementally progress from the simpler task of next-step prediction to the harder task of multi-step closed-loop prediction. The results demonstrate that these innovations facilitate improved modeling of long-term motion trajectories. Finally, a novel metric, called Normalized Power Spectrum Similarity (NPSS) is proposed, to evaluate the long-term predictive ability of motion synthesis models, complementing the popular mean-squared error (MSE) measure of the Euler joint angles over time. A user study is conducted to determine if the proposed NPSS correlates with human evaluation of long-term motion more strongly than MSE and finds that it indeed does.

Table of Contents

List of Figures

List of Tables

Acknowledgments

I would like to take this opportunity to thank my advisors Dr. C. Lee Giles and Dr. David J. Miller for their mentorship and guidance. I thank my collaborators Dr. Dan Kifer, Dr. Alexander Ororbia and Ankur Mali for their help with ideation, error analysis and encouragement throughout. I'm grateful to all the participants in the user study who took their time to give us valuable user data. Finally, I'm thankful to my parents for giving me the freedom to pursue graduate school and my passions unburdened by the financial implications.

Chapter 1
Human Motion Prediction: A background

1.1 Problem Definition

This work addresses the problem of building predictive models of human motion using motion capture data. Specifically, the models explored can be successfully used in forecasting the 3D pose of a human subject conditioned on a initial history (or a set of seed frames). Work on this problem has generally focused on two separate but complementary sub-tasks: 1) short-term motion prediction, where models are generally evaluated quantitatively by measuring mean squared error (MSE) over a short horizon, and 2) long-term motion prediction, where models are evaluated qualitatively by visual inspection of samples to see if they are able to generate plausible trajectories of human motion over long spans of time (i.e. $>$ 2-3 seconds).

Solving the above two problems in human motion prediction is challenging given the high dimensionality of the input data as well as the challenge of capturing the nonlinear dynamics and stochasticity inherently present in human motion. Furthermore, human motion, in strong contrast to the motion of other inanimate objects, depends on the subject's underlying intent and high-level semantic concepts which are tremendously difficult to model computationally.

1.2 Application

Short-term motion prediction models are useful in applications of motion tracking while long-term models are useful as generative models in areas like computer graphics [4–6]. Models successful in these tasks are also valuable for human gait analysis, studies in the kinematics of human motion, and in human-computer interaction applications [7, 8].

1.3 Related Work

Traditionally, models were built in the framework of expert systems and made use of strong simplifying assumptions, such as treating the underlying process as if it was Markovian and smooth or using low-dimensional embeddings [9, 10]. Such approaches often led to less-than-satisfactory performance. With the modern successes of artificial neural networks [11] in a variety of application domains, ranging from computer vision [12] to machine translation [13] and language modeling [14], many current, newer models of motion have been become increasingly based on neural architectures. Here recent work on recurrent neural network-based models are reviewed.

Fragkiadaki et. al [1] proposed two architectures: 1) the LSTM-3LR and 2) the ERD (Encoder-Recurrent Decoder as shown in Figure 1.1). The LSTM-3LR consists of 3 layers of 1000 Long Short-Term Memory (LSTM) units whereas the ERD model uses 2 layers of 1000 LSTM units and nonlinear multilayer feedforward networks for encoding and decoding. However, the authors observed that, during inference, the models would quickly diverge and produce unrealistic motion. They alleviate this by gradually adding noise to the input during training which helps in generating plausible motion over longer horizons.

Jain et. al [2] proposed Structural-RNNs (SRNN as shown in Figure 1.2), which take a manually designed spatio-temporal graph and convert it into a multilayer RNN architecture with body RNNs being assigned to model specific body parts and edge-RNNs to model interactions between body parts. This work also uses the noise scheduling technique employed earlier by [1] to alleviate drifting. They

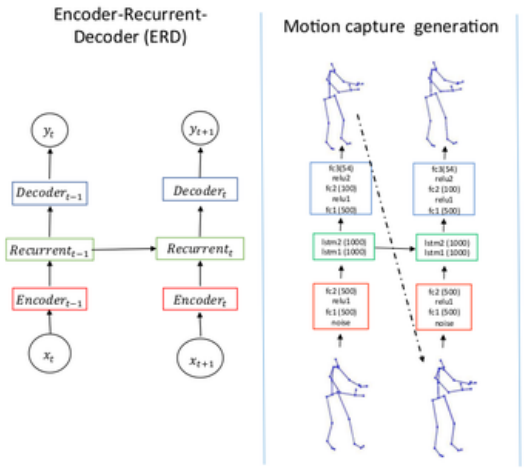

Figure 1.1. ERD model architecture of Fragkiadaki et. al [1]

show that their network outperforms previous methods in both short-term motion prediction as well as long-term qualitative motion.

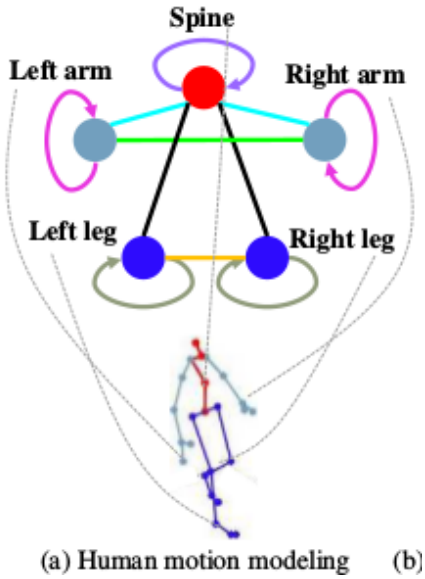

Figure 1.2. SRNN model architecture of Jain et. al [2]

More recently, [3] proposed simple but hard-to-beat baselines on short-term motion prediction as well as a 1-layer seq2seq model [15] with 1024 Gated Recurrent Unit (GRU) units and a linear output decoder for short-term and long-term motion prediction (as shown in Figure 1.3). Additionally, they trained their long-term

model using a sampling-loss as a simpler alternative to noise scheduling in order to alleviate drifting.

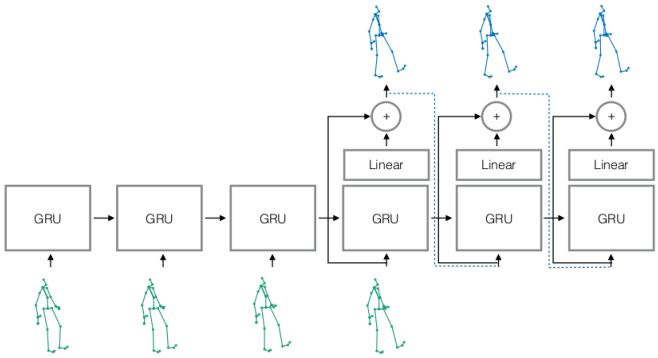

Figure 1.3. GRU seq2seq model architecture of Julietta et. al [3]

Chapter 2
Data and Preprocessing

2.1 Dataset

In order to stay consistent with previous work on human motion synthesis [1], [2], [3], the Human 3.6 Million (h3.6m) dataset [16] is used, which is currently the largest publicly available motion capture (mocap) database. The h3.6m dataset consists of 7 actors performing 15 different actions. Previous work [1], [2], [3] has particularly focused on 4 out of these 15 categories, e.g., walking, eating, smoking, and discussion when evaluating model performance. Subject #5 is split into disjoint subsets used to create both the validation and test set. To create the test-set, staying consistent with prior work by extracting 8 motion sequences per action type from subject #5, yielding the exact same 32 test sequences as used by [1], [2], [3]. The remaining sequences for subject #5 are then placed into a validation set that is used for tuning hyper-parameters. Further, the data of the other 6 subjects is then used as a training set.

2.2 Preprocessing

Again I adopt the pose representation and evaluation metrics as used previously in [1], [2], [3] to allow for experimental comparability. Pose is represented as an exponential map of each joint . To evaluate our models, we measure the Euclidean distance between predictions and ground-truth in Euler angle-space at various time slices along the predicted sequence. Elaborating briefly, each mocap vector consists of a set of 3D body joint angles in a kinematic tree representation. The

orientation of each joint is represented by an exponential map in the coordinate frame of its parent, corresponding to 3 degrees of freedom per joint. The global position of the body in the x-y plane and the global orientation about the vertical z axis are predicted relative to the previous frame, since each clip has an arbitrary global position. This is similar to the approach taken in previous work [17]. We standardize our input by mean subtraction and division by the standard deviation along each dimension. Please refer to [17] and [18] for further details.

Chapter 3
Methods and Models

3.1 Neural Temporal Models

The architecture being proposed for human motion prediction and synthesis is called the Verso-Time Label Noise-RNN model (VTLN-RNN), which consists of a top-level and a bottom-level RNN. Combined, the 2 RNNs have fewer parameters than prior motion deep learning motion synthesis models. The top-level RNN is meant to serve as a learnable noise process inspired by the work of [19] (which runs backwards in time), starting from a sampled initial hidden state (z_ϕ) and is conditioned on the one-hot encoding of the action label. This noise process is used to generate a sequence of K "guide vectors" (where K is the number of future frames we want to predict, or the prediction horizon) that will be subsequently used by the lower-level RNN. The lower-level RNN, or the Body-RNN, runs forward in time, taking in as input at each time step the joint angle vector $\mathbf{x}_t$ as well as the corresponding guide vector $\mathbf{p}_t$ to generate a prediction of the mocap angles for time-step $t + 1$. In essence, running the VTLN-RNN involves first using the top-level noise process RNN to generate the guide vectors and then using the Body-RNN to integrate both the bottom-up mocap input vectors and the top-down guide vectors to compute the final hidden states $\mathbf{h}_t$ and the next-step predictions $\hat{\mathbf{x}}_t$. The unrolled model is shown in Figure 3.1. The loss is computed using the Body-RNN's predicted outputs and the corresponding ground-truth mocap vectors.

In order to sample the initial hidden state of the top-level noise process, it is structured to work like a multivariate Gaussian distribution, drawing inspiration

from the re-parameterization trick [20] and the adaptive noise scheme proposed in [19]. The initial state z_ϕ of the top-level noise process is computed as follows:

$$z_\phi = \mu + \Sigma \otimes \epsilon \tag{3.1}$$

where $\epsilon \sim \mathcal{N}(0, \mathrm{I})$, μ the mean of the random variable, and Σ is its covariance, specifically a diagonal covariance. μ and Σ are parameters that are learned along with the rest of the neural network weights using back-propagated gradients during training. This formulation of the hidden state allows the designer to input samples from a simple base distribution, e.g., a standard Gaussian, instead of having to tune the noise parameters, such as its variance manually.

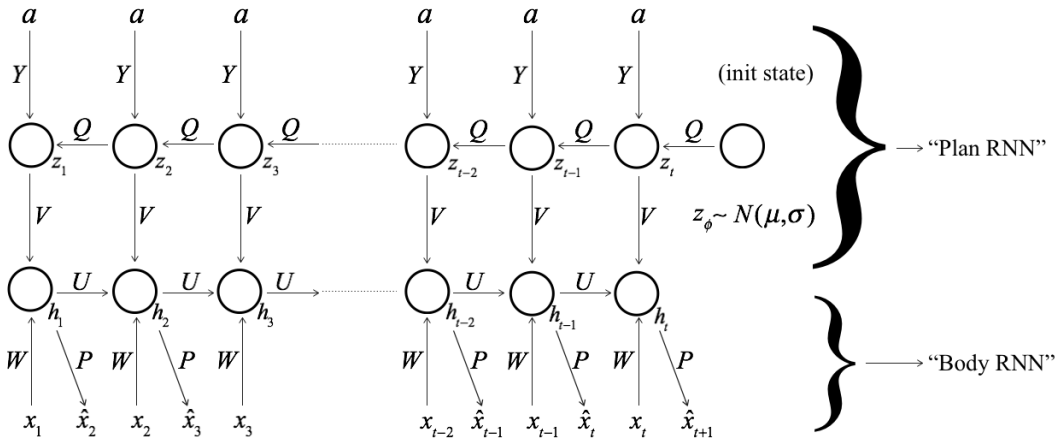

Figure 3.1. VTLN-RNN architecture

In this paper, we use the Gated Recurrent Unit (GRU) [21] to instantiate both the top-level and bottom-level RNNs of the VTLN-RNN due to its simplicity, competitive performance, and ease of training compared to the LSTM [22]. The cell update equations for the top-level GRU remain the same as described in [21] except that we note its non-state input is the action label (which remains fixed over the length of the sequence). However, the cell update equations for the Body-GRU are as follows:

$$r_j = \sigma\big([W_r x_t]_j + [U_r h_{t-1}]_j + [V_r p_t]_j\big) \tag{3.2}$$

$$z_j = \sigma\big([W_z x_t]_j + [U_z h_{t-1}]_j + [V_z p_t]_j\big) \tag{3.3}$$

$$\tilde{h_j}^t = \Phi\big([W x_t]_j + [U(r \otimes h_{t-1})]_j + [V p_t]_j\big) \tag{3.4}$$

$$h_j^t = z_j h_j^{t-1} + (1 - z_j)\tilde{h_j}^t. \tag{3.5}$$

The motivation behind the VTLN-RNN structure was to hierarchically decompose the motion synthesis problem down to a two-level process, as has been successfully done in neural-based dialogue modeling [23], [24]. The top-level RNN would serve to roughly sketch out a course trajectory that the lower-level RNN would take, further conditioned on actual data and its own internal state. However, unlike the hierarchical neural dialogue models that served as inspiration, in this case the top-level process runs in the backwards temporal direction. This was chosen considering the gradient flow in the unrolled computation graph. If the top-level process starts at time $t = K$ and works backward to time $t = 1$, the parameter updates of the top-level model will depend more heavily on information from the future (or from far later on in a sequence) and this information would be encoded in the synaptic weights related to a specific action type/label. When the top-level process is used to generate the sequence of guide vectors, it creates "hints" or coarsely defined states that the lower-level Body-RNN can then refine based on actual input data or its own closed-loop predictions.

While it is hard to theoretically prove that the top-level RNN is truly "planning" out the ultimate trajectory of the model's predictions, experiments conducted will show that this two-level process appears to offer some useful regularization, improving model generalization over simpler mechanisms such as drop-out.

Additionally, our model used for both short-term and long-term motion prediction has significantly less parameters compared to [1], [2], [3] and yet achieves state-of-art results as shown in Table 5.5 on long-term motion prediction and is competitive with [3] state-of-art results on short-term motion prediction as shown in Table 5.4.

Models	No. of parameters
ERD [1]	14,842,054
LSTM-3LR [1]	20,282,054
SRNN [2]	18,368,534
MBR-long [3]	3,425,334
GRU-d (ours)	**2,735,670**
VGRU-d (ours)	**3,413,047**

Table 3.1. Number of parameters of various RNN-based motion models.

3.2 Incorporating Derivatives

Motion derivatives contain crucial feature information which can be used to model local (near past) motion information. From Newtonian mechanics, we know that a body's future trajectory can be calculated accurately given its current position but also its velocity and acceleration (i.e. first and second-order derivatives of position w.r.t time). These features are cheap to compute and do not require any additional model parameters. Motivated by this, motion derivatives are extracted by using a finite backward difference approximation, calculated as follows:

$$\nabla_h^n[f](x) = \sum_{i=0}^{n} (-1)^n \binom{n}{i} f(x - ih) \tag{3.6}$$

where i is the order of the derivative we would like to approximate, up to n, and h is a non-zero spacing constant.

Motion derivatives of $n = \{1, 2, 3\}$ with $h = 1$ are extracted using the above equation and appended to the vector of joint angles. The linear decoder of our recurrent model outputs only joint angles for the next timestep. During closed loop (i.e. iterative multi-step prediction), we calculate these motion derivatives on-the-fly using the equations above and append them to the joint angles before passing the concatenated vector of joint angles + derivatives as input to RNN at $t + 1$ timestep.

3.3 Alleviating Drifting

The standard method to train RNNs for sequence prediction tasks is to feed the ground-truth inputs at every timestep t during training. This is known as open loop prediction. Then, at test time, the model's previous prediction at t is fed in as input to the model at timestep $t + 1$. This known as closed-loop (or iterative) prediction. However, a key issue with this method is that the model is unable to recover from accumulation of errors and the RNN predictions degrade significantly over time. This is due to the significant mismatch in the inputs it receives during train (i.e. ground-truth inputs) and test time (i.e. its own noisy predictions from previous timesteps) commonly referred to as "exposure bias". This causes synthesized long-term motion trajectories to quickly diverge from the manifold of plausible

motion trajectories. As mentioned earlier [1] and [2] alleviate this issue by injecting gradually increasing magnitudes of Gaussian noise to inputs during training. [3] used a sampling loss where, during training, the model outputs are fed back to itself. Lamb et. al [25] introduced the method of "Professor Forcing" which addresses this issue by using an adversarial training regime to ensure the hidden states of the RNN are constrained to be similar during train and test time. However, this method is computationally expensive, needs careful hyperparameter tuning, and suffers from stability issues normally encountered in the training of Generative Adversarial Networks. More recently, [26] showed that their method, or auto-conditioning, helps the RNN models produce good qualitative long-term motion by alternating between feeding in ground-truth samples and the model's own outputs during training.

We can view this problem of using the RNN for multi-step iterative prediction at test time through the prism of multi-task and/or curriculum learning. We ultimately require the RNNs to achieve good performance on the hard-task of multi-step iterative prediction starting from the simple task of one-step prediction. An intuitive way to achieve this would be to gradually make the RNN progress from the simple task of one-step prediction (ground truth fed in at every timestep) to the final goal of multi-step iterative prediction. Defining a composite loss function with separate terms for measuring one-step prediction and multi-step iterative prediction losses, and weighting these terms, would ensure that the network slowly adapts from being able to only predict one-step ahead to becoming capable of multi-step iterative prediction during the course of the training cycle. This intuition forms the basis of our multi-objective loss function defined as follows,

$$L(\widehat{y}, y) = \frac{1}{T}\sum_{t=0}^{T}(\widehat{y_o}^{t} - y^{t})^2 + \frac{\lambda}{T'}\sum_{t_1=0}^{T'}(\widehat{y_c}^{t_1} - y^{t_1})^2 \tag{3.7}$$

where y^t = ground-truth output at t, $\widehat{y_o}^{t}$ = model output in open-loop mode at t, $\widehat{y_c}^{t_1}$ = model output in closed-loop mode at t_1. For clarity, I use *"open-loop"* mode to refer to feeding ground-truth inputs at every timestep to the RNN in order to produce outputs and *"closed-loop"* mode refers to feeding the model's own output at t as input to it at $t+1$. For every input sequence of data this loss function requires us to perform the forward pass twice on the RNN network, i.e., i)

to compute $\widehat{y_o}^t$ in open-loop mode and ii) to compute $\widehat{y_c}^t$ in closed-loop mode. The weight term λ is gradually increased using a step schedule over the training cycle starting with a zero value at the beginning. This schedule therefore gradually places greater importance to the loss-term contributed by making closed-loop predictions as the network has learned to make better one-step predictions. From our long-term motion synthesis experiments, we can see that our multi-objective loss function outperforms noise scheduling [1,2], auto-conditioning [26] and sampling loss devised by [3].

3.4 Long-Term Multi-Action Motion Model

Prior work on long-term motion prediction such as that of Fragkiadaki et. al [1], Jain et. al [2] and Julietta et. al [3] have all trained their top-performing models on single-action data. Apart from being computationally expensive to train a new model from scratch per action class, it also highlights shortcomings of the current state-of-art RNN-based long-term motion models. One of the challenges in building a multi-action model is that a given current pose and past trajectory can result in multiple equally plausible/valid future poses (possibly across several action classes), and since these RNN-based models are trained in a maximum likelihood setting, they're trained to estimate the *"average"* or *"mean"* pose from the training data they have seen. This results in the model sometimes averaging poses across action classes and resulting in physically impossible poses and unsatisfactory performance.

In this work, an attempt is made towards this end of building a RNN-based long-term multi-action motion model competitive with the various single-action models (methods) described previously. The multi-action RNN model consists of a 2-layer 512 unit GRU network wherein the RNNs have 2 input embedding matrices; one which maps the input joint angles to the hidden state and the second which receives a 1-hot encoding of the action class. The GRU cell expressions are similar to that of the body-RNN described in section 3.1. Only difference between this multi-action model (referred to as MA-RNN in Table 5.6) and the VTLN-RNN architecture is that both layers operate in forward time. The intuition behind it is that in training the synaptic weights of this second embedding matrix (which takes a 1-hot action label as input) will learn useful action conditional information.

Since this term is also involved in the hidden state transition expression, it can provide useful action conditioned information thereby help alleviate the model from averaging poses across action classes during long-term prediction. Similar to our earlier models, we append motion derivatives of input joint angles and train the model with the novel multi-objective loss function. Further training details is given in section 5.1. As seen from Table 5.6, the MA-RNN model achieves the best (or close to) for all actions except eating, indicating the promise of this model architecture to serve as a strong baseline model for the multi-action task setting.

Chapter 4
Evaluation Metrics

4.1 Background

The use of mean-squared error (MSE) as an evaluation metric for models has been the standard practice [1–3] on both the short-term motion prediction and long-term motion synthesis tasks. In short-term motion prediction the evaluation metric needs to capture how well various models are able to mimic the ground-truth data over short-term horizons (i.e 0-500 milliseconds) as these models are used for motion tracking applications.

However in the long-term motion synthesis task, models need to be evaluated on how well they generate plausible future motion over long-term horizons given some seed frames of motion. Since human motion is inherently stochastic over long time horizons, models can significantly deviate from the ground-truth trajectories and have a large MSE despite producing qualitatively good human motion. This problem has been noted in prior work [1], [2], [3]. There are a variety of causes. For example: if the predictions correspond to walking at a slower pace, the joint angles will be misaligned (frequency-shift) and MSE computed will diverge over time. In the short term, the joint angles may still be similar enough for MSE to meaningfully capture similarity, but in the long term they will become significantly different. Similarly, if a few extra frames of motion are added or removed (phase-shift) compared to ground-truth sequence will result in high MSE values because frames are again misaligned. Therefore, as noted in prior work [1], [2], [3], the use of MSE as an evaluation metric is not appropriate in the long-term task. However,

no attempt had previously been made to suggest another quantitative metric for evaluation of long-term motion synthesis models.

4.2 Normalized Power Spectrum Similarity

In this work, we propose such a metric based on the following intuition. We can say that the qualitative essence of any action such as walking, eating, running etc. can be captured through the frequency signature of joint angles of the body while performing that action. For walking at a slower pace example, the power spectrum (obtained from a discrete Fourier transform) would show spikes at a slightly lower frequency and the addition or removal of few frames would show up as a phase-shift in the frequency domain. The examples of slow/fast or phase-shifted walking involve periodic sub-actions, whereas aperiodic actions such as discussion will show a more uniform spread of power in the frequency domain (this indicates a lack of periodicity in the action which is also being picked up by the power spectrum). Measuring similarity of power spectrum between between ground truth sequence and corresponding generated sequence for the same motion type would account for these phenomena and correlate better with the visual quality (see user study details in the following section) of samples compared to MSE. The field of content-based image retrieval have used EMD [27, 28] to quantify perceptual similarity of images using the EMD distance between their color histograms. Using the intuitions from above examples and inspired by this success, we propose an EMD-based metric over the power spectrum that overcomes many of the shortcomings of MSE as an evaluation metric on the long-term task.

First, let's introduce some notation and formulation. For a given action class in the test set, let there be k sequences each of T length and output vector of joint angles at each time-step be D dimensional. We define $x_{i,j}[t]$ to be the ground-truth value at time t for j^{th} feature dimension for i^{th} sequence and $y_{i,j}[t]$ to be the corresponding model prediction. Also, let $X_{i,j}[f]$ and $Y_{i,j}[f]$ be the squared magnitude spectrum of Discrete Fourier Transform coefficients (per sequence i per feature dimension j) of $x_{i,j}[t]$ and $y_{i.j}[t]$ respectively. First we normalize $X_{i,j}[f]$ and $Y_{i,j}[f]$

w.r.t f as,

$$X_{i,j}^{\text{norm}}[f] = \frac{X_{i,j}[f]}{\sum_f X_{i,j}[f]}; Y_{i,j}^{\text{norm}}[f] = \frac{Y_{i,j}[f]}{\sum_f Y_{i,j}[f]} \tag{4.1}$$

We then compute,

$$\text{emd}_{i,j} = \left\| X_{i,j}^{\text{norm}}[f] - Y_{i,j}^{\text{norm}}[f] \right\| \tag{4.2}$$

where, $\|.\|$ is the L_1-norm. Finally, we use a power weighted average over all i and j of 1-d EMD distances computed in (4.2) as shown below,

$$NPSS = \frac{\sum_i \sum_j p_{i,j} * \text{emd}_{i,j}}{\sum_i \sum_j p_{i,j}} \qquad p_{i,j} = \sum_f X_{i,j}^{\text{norm}}[f] \tag{4.3}$$

where $p_{i,j}$ = total power of i^{th} feature in j^{th} sequence
to arrive at our scalar evaluation metric for an evaluation set of sequences for a given action class. This metric is referred to as normalized power spectrum similarity (NPSS).

Another interpretation is that, long-term motion synthesis can be viewed as a generative modeling task. Under this interpretation, the evaluation metric must capture differences in the distributions of the ground-truth and predicted motion samples. NPSS captures distributional differences in the power spectrum of joint angles of the ground-truth and predicted sequences. As a result, it is better equipped to model differences in visual quality of motion trajectories. The section 5.2 outlines the details of the user study conducted to support the claim that NPSS is more correlated to qualitative human perception of long-term motion compared to MSE.

Chapter 5
Results and Discussion

5.1 Training Details

For the short-term model, the VGRU-r1 (MA), it was trained on all action classes using our proposed multi-objective cost, calculating gradients over mini-batches of 32 samples (clipping gradient norms to 5) and optimizing parameters over $100,000$ iterations RMSprop [29] with initial learning rate $\lambda = 0.0001$ and decayed by 0.8 every 5000 iterations until 60,000 iterations. Drop-out [30, 31], with probability of 0.3, was applied only to the Body-RNN, which was further modified to use skip connections that connect input units to output units, as in [3]. The model was given 50 seed frames and tasked with predicting the next 10 subsequent frames (400 milliseconds). When training for this, the VTLN-RNN is unrolled backwards while the Body-RNN is unrolled forwards, in time, over 60 steps. (Note: MA stands for multi-action, SA for single-action.)

For the long-term models, which were trained on single-action data, parameter optimization was done using RMSprop ($\lambda = 0.0002$, decayed by 0.6 every 2000 iterations) over $10,000$ iterations with mini-batches of 32 (clipping gradient norms to 1), using, again, our proposed cost function. Models were fed in 50 seed frames and made to predict the next 100 frames (4 sec), which meant that the VTLN-RNN was unrolled backwards and the Body-RNN forwards 150 steps. The input vector to the Body-RNN consisted of joint angles appended with motion derivatives.

For the multi-action long-term model, it was trained with RMSProp optimizer

($\lambda = 0.0001$, decayed by 0.8 every 2500 iterations over $15,000$ iterations with mini-batch size of 32 (clipping gradient norms to 5) using the proposed loss function. Model was fed 50 seed frames and made to predict 100 frames (consistent with prior work on single-action long-term models).

In Tables 5.5, 5.6, 5.7, VGRU-d refers to our proposed VTLN-RNN architecture where the VTLN-RNN and Body-RNN both contain only a single layer of 512 GRU cells. GRU-d refers to a 2-layer GRU model (512 units in each). Both VGRU-d and GRU-d models are trained with our proposed loss and make use of inputs augmented with motion derivatives. VGRU-ac refers to our VTLN-RNN architecture trained with auto-conditioning [26], using the recommended length of 5, serving as a baseline. For all models (short and long-term), hyper-parameters were tuned on the validation set (subset of sequences sampled from subject #5).

5.2 User Study Analysis

We conducted a user study to understand how human judgment of long-term motion correlates with MSE as well as our proposed NPSS. A desirable quantitative evaluation metric for long-term human motion would be one that strongly agrees with human judgment. In order to conduct this study, we considered the 6 models from Table 5.6 (i.e. VGRU-r1(SA), MBR-unsup (SA), MBR-long, VGRU-ac, GRU-d and VGRU-d). In each trial, a user was presented videos of the ground-truth motion and corresponding model predictions from a randomly chosen pair of models (from the list above) for a given test-set action sequence (the ordering of the models was random with identities were hidden from the users). Users were asked to compare model predicted motion trajectories with the ground truth, based on which one possessed better "motion quality". The users were informed that the phrase "motion quality" referred to similarity/closeness in overall skeletal pose (i.e. overall posture) and joint motion dynamics over the entire sequence, rather than simple point-to-point matches in time, and made their decisions based on this criteria. A sample screenshot of the user survey video is shown in Figure 5.1.

For each of the 4 action classes (i.e. walking, eating, smoking, and discussion) we presented 20 video sequences of the ground-truth with the A versus B comparison as shown in Figure 5.1. Video samples were selected uniformly and randomly

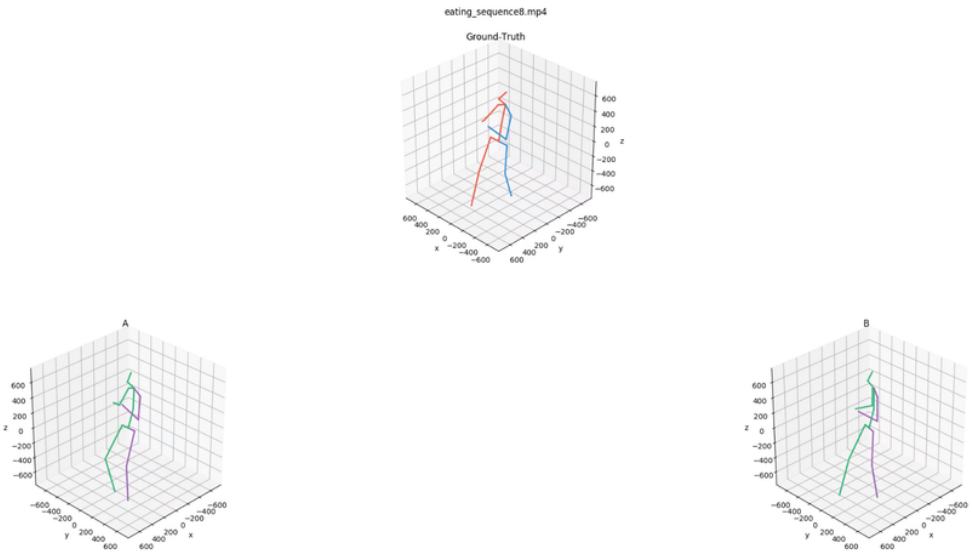

Figure 5.1. A sample screenshot taken from the user survey

(without replacement) from all possible, pairwise combinations of the 6 models. We then selected a test sequence for an action class, via uniform random sampling with replacement, and presented the ground-truth motion sequence and previously picked paired model predictions for that sequence. This process is repeated to generate 20 videos (i.e. 20 questions) for each of the 4 action classes. The study involved 20 participants for each of the 4 action class surveys.

Now for the 2 evaluation metrics (i.e. MSE and NPSS) rankings of the models used in the user study were derived. For the MSE metric ranking, the sum of MSE over all timeslices for the long-term window (i.e. 80, 160, 320, 400, 560, 1000 milliseconds is computed which is consistent with prior work [2]). For NPSS, the results in Table 5.1 is used to arrive at ranking of the models for all 4 actions.

Then, using these rankings (MSE and NPSS) for each action class to make predictions for each question in the user survey. As shown in Table 5.2, probabilities of agreement and disagreement with users is computed for MSE and NPSS. Further, a Binomial test of proportions was conducted to test the claim that NPSS agrees better with user judgment than MSE. In this test, p_1 is defined as the probability that, on a random sample, NPSS will agree with human ordering/choice while p_2 is the probability that MSE will agree with human ordering/choice. The null hypothesis is taken to be $H_0 : p_1 \leq p_2$ and the alternative

Metrics	Walking	Eating	Smoking	Discussion
MSE rankings	1. VGRU-r1(SA)	1. VGRU-r1(SA)	1. VGRU-r1(SA)	1. VGRU-r1(SA)
	2. MBR-unsup (SA)	2. VGRU-d	2. MBR-unsup (SA)	2. VGRU-d
	3. VGRU-d	3. MBR-unsup (SA)	3. VGRU-d	3. GRU-d
	4. MBR-long	4. VGRU-ac	4. VGRU-ac	4. VGRU-ac
	5. VGRU-ac	5. GRU-d	5. GRU-d	5. MBR-unsup (SA)
	6. GRU-d	6. MBR-long	6. MBR-long	6. MBR-long
NPSS rankings	1. VGRU-d	1. GRU-d	1. VGRU-d	1. VGRU-ac
	2. GRU-d	2. VGRU-ac	2. GRU-d (SA)	2. GRU-d
	3. VGRU-ac	3. VGRU-d (SA)	3. VGRU-ac	3. VGRU-d
	4. VGRU-r1 (SA)	4. VGRU-r1 (SA)	4. VGRU-r1 (SA)	4. MBR-unsup (SA)
	5. MBR-long	5. MBR-unsup (SA)	5. MBR-unsup (SA)	5. MBR-long
	6. MBR-unsup (SA)	6. MBR-long	6. MBR-long	6. VGRU-r1 (SA)

Table 5.1. MSE and NPSS rankings of long-term motion models

	MSE	NPSS
Agree	0.4875 (39/80)	0.8125 (65/80)
Disagree	0.5125 (41/80)	0.1875 (15/80)

Table 5.2. User agreement ratios for MSE and NPSS aggregated across all actions taking majority user vote as ground-truth. a/b = number of times user's answers agrees with metric's answer/ total sequences in user survey across 4 actions

hypothesis to be $H_A : p_1 > p_2$ and seek to test the null against the alternative hypothesis. Scientific studies typically set the threshold of statistical significance for p-values to be below 0.01 (smaller p-values would better support the claim that NPSS is a better metric, confirming that p_1 is statistically larger than p_2). The value obtained is significantly lower than this threshold, i.e., a p-value of $\mathbf{1.7 \times 10^{-5}}$.

Further analysis of the user study results was conducted to support the claim that the proposed NPSS evaluation metric correlates more strongly with human judgment over MSE for specific timeslices $\{80, 160, 320, 400, 560, 1000\}$ milliseconds (used previously by [2], [1], [3]) as well as the sum of MSE scores over all timeslices. There are 80 pairs of generated sequences (and, for each pair, a ground truth sequence). For every comparison between two generated sequences A and B, there are 20 human judgements determining which one is closer to the ground truth sequence.

A good error measure should correlate well with human judgment in the following manner: if the vast majority of users prefer sequence A over Sequence B, then the error of B should be much higher than that of A. On the other hand, if the preference is almost evenly split, then Sequences A and B should have similar error.

This kind of correlation is tested as follows: For each A/B comparison, the *Disapproval* of A over B is defined as, denoted by *Disapproval*(A,B), to be the fraction of subjects who preferred B minus the fraction of subjects who preferred A. Thus if A is much worse than B, *Disapproval*(A,B) will be close to 1 and, if A and B are equally good, *Disapproval*(A,B) will be 0. If A is much better than B, then the *Disapproval*(A,B) will be close to -1.

For each A/B comparison, we can also compute

$$\text{NPSS(A)} - \text{NPSS(B)} \tag{5.1}$$

$$\text{MSE}_1\text{(A)} - \text{MSE}_1\text{(B)} \tag{5.2}$$

$$\text{MSE}_2\text{(A)} - \text{MSE}_2\text{(B)} \tag{5.3}$$

where, NPSS(A) is the NPSS error for sequence A with respect to the ground truth, $\text{MSE}_1(A)$ is the sum of MSE scores (with respect to ground truth) over timeslices $= \{80, 160, 320, 400, 560, 1000\}$ and $\text{MSE}_2(A)$ is the sum of MSE scores over all of the time slices. The reason for two MSE calculcations is that prior work only evaluated MSE at select time slices (so MSE_1 was also computed to ensure consistency with prior work).

For each Equation (Eq. 5.1, 5.2, or 5.3), strong positive correlation means they strongly agree with human judgment while a correlation close to 0 means they do not appear to be related to human judgment. Spearman's Rank Correlation is used for this task. Both the correlation coefficient and the p-value are reported. The p-value is designed to test the null hypothesis that the correlation is 0. A low p-value indicates that there is evidence against the null hypothesis, or, in other words, a low p-value indicates that there is a correlation and that it is statistically significant. Typically, statistical significance is claimed when the p-value is less than 0.01. We show the results of our correlation test in Table 5.3.

	MSE_1 (Eq. 5.2)	MSE_2 (Eq. 5.3)	NPSS (Eq. 5.1)
Correlation	-0.143	-0.0638	0.5635
p-value	0.2049	0.5738	5.23×10^{-8}

Table 5.3. Spearman Correlation Results.

The analysis conducted shows that the proposed NPSS has a reasonably large, positive correlation and a very small p-value, meaning that it is strongly correlated with human judgment and the correlation found is statistically significant. Meanwhile MSE_1 empirically shows a small negative correlation and has a relatively high p-value which means that it is still possible for it to be completely unrelated to human judgment (which would be consistent with the observations made in prior work [2], [3]). When the MSE is computed across all time slices (i.e. MSE_2) the empirical correlation is even closer to 0. In conclusion, with this analysis included, the user study strongly suggests that the proposed NPSS metric is a much more suitable quantitative metric for evaluating motion sequences generated by statistical motion-synthesis models.

5.3 Short-Term Motion Prediction

	Walking				Eating				Smoking				Discussion			
milliseconds	80	160	320	400	80	160	320	400	80	160	320	400	80	160	320	400
Zero-velocity [3]	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**
MBR-unsup (MA) [3]	**0.27**	**0.47**	0.70	0.78	0.25	0.43	0.71	0.87	0.33	0.61	1.04	1.19	0.31	0.69	1.03	1.12
MBR-sup (MA)	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
VGRU-r1 (MA) (ours)	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
	± 1e-3	± 1e-3	± 2e-3	± 2e-3	± 2e-3	± 1e-3	± 2e-3	± 2e-3	± 6e-4	± 1e-3	± 1e-3	± 1e-3	± 2e-3	± 1e-3	± 3e-3	± 5e-3

Table 5.4. Short-term results: MSE on test sequences for short-term motion prediction. All models in this Table are trained on multiple actions. VGRU-r1 (MA) refers to our VTLN-RNN with 1 layer (512 GRU unit) and a Body-RNN with 1 layer 512 GRU cells, where the Body-RNN sports residual input-to-output connections as in [3]. For the VGRU-r1, model we have computed mean and standard error over 30 trials.

5.4 Long-Term Motion Prediction

	Walking						Eating						Smoking						Discussion					
models	80	160	320	400	560	1000	80	160	320	400	560	1000	80	160	320	400	560	1000	80	160	320	400	560	1000
MBR-unsup (SA) [3]	**0.37**	0.655	0.987	1.095	1.286	1.476	0.411	0.781	1.375	1.630	1.926	2.106	0.472	0.891	1.497	1.726	2.077	2.581	0.701	1.326	2.134	2.433	2.996	2.950
VGRU-r1 (SA) (ours)	0.410	**0.570**	**0.807**	**0.868**	**1.026**	**1.231**	**0.285**	**0.441**	**0.668**	**0.829**	**0.995**	**1.531**	**0.378**	**0.656**	**0.916**	**0.994**	**1.147**	**1.837**	**0.504**	**0.909**	**1.074**	**1.282**	**1.653**	2.168
	± 1e-3	± 1e-3	± 2e-3	± 3e-3	± 3e-3	± 3e-3	± 2e-3	± 2e-3	± 2e-3	± 3e-3	± 3e-3	± 3e-3	± 1e-3	± 1e-3	± 1e-3	± 2e-3	± 2e-3	± 2e-3	± 1e-3	± 2e-3	± 4e-3	± 5e-3	± 6e-3	± 7e-3
ERD [1]	1.30	1.56	1.84	-	2.00	2.38	1.66	1.93	2.28	-	2.36	2.41	2.34	2.74	3.73	-	3.68	3.82	2.67	2.97	3.23	-	3.47	2.92
LSTM-3LR [1]	1.18	1.50	1.67	-	1.81	2.20	1.36	1.79	2.29	-	2.49	2.82	2.05	2.34	3.10	-	3.24	3.42	2.25	2.33	2.45	-	2.48	2.93
SRNN [2]	1.08	1.34	1.60	-	1.90	2.13	1.35	1.71	2.12	-	2.28	2.58	1.90	2.30	2.90	-	3.21	3.23	1.67	2.03	2.20	-	2.39	2.43
VGRU-ac	1.180	1.210	1.247	1.236	1.291	1.363	1.150	1.210	1.310	1.400	1.490	1.700	1.81	1.950	2.080	2.140	2.240	2.440	1.720	1.970	1.930	1.870	2.050	**2.147**
	± 3e-4	± 3e-4	± 2e-4	± 3e-4	± 6e-4	± 7e-4	± 2e-4	± 1e-4	± 1e-4	± 1e-4	± 1e-4	± 1e-4	± 1e-4	± 1e-4	± 1e-4	± 1e-4	± 1e-4	± 1e-4	± 1e-4	± 1e-4	± 1e-4	± 1e-4	± 1e-4	± 1e-4	± 1e-4
MBR-long [3]	0.93	1.05	1.24	1.29	1.43	1.56	1.13	1.35	1.75	1.91	2.07	2.28	1.29	2.07	2.53	2.56	2.76	3.39	1.63	2.03	2.57	2.72	2.96	2.94
GRU-d (ours)	1.311	1.333	1.369	1.364	1.350	1.370	1.275	1.305	1.386	1.466	1.530	1.702	1.943	2.062	2.201	2.255	2.342	2.486	1.744	1.980	2.026	1.994	2.214	2.172
VGRU-d (ours)	1.108	1.146	1.211	1.200	1.220	1.280	1.090	1.160	1.240	1.330	1.370	1.560	1.670	1.800	1.940	1.980	2.060	2.320	1.749	2.037	2.011	1.868	2.088	2.318
	± 1e-4	± 1e-4	± 2e-4	± 2e-4	± 3e-4	± 2e-4	± 2e-4	± 1e-4	± 1e-4	± 1e-4	± 1e-4	± 1e-4	± 1e-4	± 1e-4	± 1e-4	± 1e-4	± 1e-4	± 1e-4	± 1e-4	± 1e-4	± 1e-4	± 1e-4	± 1e-4	± 1e-4	± 1e-4

Table 5.5. Long-term motion synthesis results: All models in this table are trained on single-action data (SA = single-action). Top set show short-term models including the (MBR-unsup(SA) = Residual unsup. (MA) from [3] re-trained on single-action) and as well as ours sampled for longer duration to match long-term duration. Bottom set shows long-term models by MBR-long = sampling-based loss (SA) from [3], ERD and LSTM-3LR from [1], SRNN from [2]), our GRU-d and VGRU-d and VGRU-ac. Since the VTLN-RNN architecture samples from a noise distribution for each forward pass, the table shows mean and standard deviation in predictions over 30 trials.

Models	Walking	Eating	Smoking	Discussion
VGRU-r1(SA) (ours)	1.217	1.312	1.736	4.884
MBR-unsup (SA) [3]	1.809	1.481	2.794	2.258
MBR-long [3]	1.499	1.621	4.741	2.882
VGRU-ac	1.032	0.842	1.426	**1.651**
GRU-d (ours)	0.931	**0.836**	<u>1.274</u>	1.688
VGRU-d (ours)	0.887	<u>0.846</u>	**1.235**	1.777
MA-RNN (ours)	**0.845**	0.966	1.292	<u>1.659</u>

Table 5.6. Test-set NPSS scores (lower is better). Above the top double line: short-term models, i.e., MBR-unsup (SA), MBR-unsup. (MA) [3] (re-trained on single-action), and ours, sampled for long-term durations. Below the line: long-term models, i.e., MBR-long (SA) [3], and ours, such as GRU-d, VGRU-d, & VGRU-ac. MA-RNN refers to our multi-action model

Given the results of our user study, we argue that NPSS should be preferred (over MSE) for measuring model generation quality over long sequences (for predictions made over longer horizons). However, to holistically evaluate a motion synthesis model, we recommend using NPSS in tandem with MSE when evaluating a model's ability to make both short-term and long-term predictions. The results of our user study for NPSS is promising, however, further studies should be conducted to further validate and strengthen our findings.

Models	Short-Term			
	Walking	Eating	Smoking	Discussion
VGRU-r1 (SA) (ours)	0.120	**0.091**	**0.052**	0.258
MBR-unsup (SA) [3]	0.238	0.249	0.183	0.416
MBR-long [3]	0.161	0.214	0.265	0.703
VGRU-ac	0.118	0.113	0.075	0.256
GRU-d (ours)	0.127	0.095	0.126	**0.185**
VGRU-d (ours)	**0.117**	0.121	0.084	0.194
	Medium-Term			
VGRU-r1 (ours)(SA)	0.194	0.093	0.079	0.375
MBR-unsup (SA) [3]	0.206	0.178	0.237	0.439
MBR-long [3]	0.237	0.160	0.405	0.477
VGRU-ac	0.188	0.103	0.097	0.298
GRU-d (ours)	**0.170**	0.096	0.083	**0.258**
VGRU-d (ours)	0.179	**0.080**	**0.067**	0.331
	Long-Term			
VGRU-r1 (SA) (ours)	0.544	0.764	0.948	2.72
MBR-unsup (SA) [3]	0.884	0.684	1.077	0.943
MBR-long [3]	0.549	0.754	1.403	1.245
VGRU-ac	0.460	0.459	1.051	0.811
GRU-d (ours)	0.406	0.332	0.723	**0.785**
VGRU-d (ours)	**0.359**	**0.288**	**0.577**	1.001

Table 5.7. NPSS at 3 different time scales i.e 1) short-term: 0-1 second 2) medium-term: 1-2 seconds 3) long-term: 2-4 seconds window prediction on test set

For compatibility with prior work, Table 5.5 compares the MSE of Euler angles, measured at particular time slices on test sequences, with competing methods such as LSTM-3LR and ERD [1], SRNN by [2], and MBR-long [3]. Although the short-term model, VGRU-r1 (SA), displays the best performance (lowest mse) until the 1 second mark, it has been noted by [2], [3] and further now corroborated by the results of the user study that MSE is not appropriate for the task of long-term motion synthesis. Table 5.6 shows the results of the proposed NPSS metric for various models evaluated on the test set.

In order to discern the strengths and weaknesses of short-term and long-term models, the NPSS metric is computed on test sequences at 3 different timescales, i.e., 1) short-term: 0-1 s, 2) medium-term: 1-2 s, 3) long-term: 2-4 s along the

prediction timeline for test sequences shown in Table 5.7. Observe that the short-term models (above double line) VGRU-r1 (SA) and MBR-unsup (SA) perform competitively with long-term models (below double line) in the short-term timescale. In the medium-term prediction horizon, the short-term models degrade slightly more than the long-term models, as evidenced by a small gap in the measured NPSS values. However, in the long-term prediction horizon (of 2-4 s), the short-term models degrade significantly relative to the long-term models. This is evidenced by wider gaps in NPSS values. GRU-d and VGRU-d models perform best across all actions and time-horizons, effectively outperforming MBR-long and VGRU-ac.

Finally, Table 5.4 shows MSE results for short-term motion prediction experiments on multi-action data on test set sequences. Zero-velocity is a simple, yet hard-to-beat baseline, introduced in [3], which uses the previous frame as the prediction for current one. As we can see VGRU-r1 model is competitive with the state-of-art short-term MBR model as well the quite powerful, zero-velocity baseline. These results show that our proposed VTLN-RNN architecture, augmented with motion-derivative features and our novel multi-objective loss function, can serve as useful models for short-term motion prediction as well as powerful long-term motion synthesizers.

5.5 Ablation Study

An ablation study was conducted to determine the value of each component of the overall neural architecture for long-term motion synthesis. The components investigated were: 1) the two-level processing mechanism, 2) the integration of the finite-difference motion derivative approximation features, and 3) the multi-objective cost function used to guide parameter optimization.

VGRU-d in Tables 5.8, 5.9 refers to the full two-level processing network (i.e. VTLN-RNN) with derivatives appended and trained using our multi-objective loss. Both the RNNs in the two-level system contain a single layer of 512 GRU units as described in chapter 3. Dropout [30] with a probability of 0.3 was applied only to the Body-RNN (as shown in Figure 3.1). GRU-d in Tables 5.8, 5.9 refers to a regular two-layer network (512 GRU units/cells in each layer) trained with the

proposed loss and derivatives appended. Dropout [30] with probability of 0.3 was applied to both layers of the *GRU-d* model.

VGRU-d + no-loss in Tables 5.8, 5.9 refers to a system that is identical to the VGRU-d system (described above) but trained without the proposed multi-objective loss. During the training phase, when predicting the data at time $t+1$, the ground-truth at time t was fed in as input to the model while at test-time, the model's own output at time t was used instead, i.e., standard Teacher Forcing. The rest of the training setup was identical to that described for the *VGRU-d* model. *GRU + no-d* in Tables 5.8, 5.9 refers to a system that was identical to that of *GRU-d* except that the finite difference approximations of the $\{1, 2, 3\}$-order derivatives, as described in section 3.2 of this document, were not appended to the input vector.

Looking at the NPSS results in Table 5.9, we can see that the VGRU-d model, with all of the proposed components, achieves the lowest score on 2 out of 4 actions, e.g., walking and smoking, and with a score that is quite close for the act of eating. Discussion itself is a highly aperiodic and extremely difficult-to-model action, especially when only pure joint angle information is exclusively used, which was also noted in prior work [2]. The *GRU-d* achieves the overall second best performance across all 4 actions. Furthermore, when the approximate derivative features are dropped, the performance of the *GRU + no-d* drops significantly across all 4 actions. This indicates that approximate joint angle derivatives play a crucial role in guiding the model to producing smooth, realistic plausible (long-term) motion trajectories (with the added benefit that these finite-difference equations are parameter-free and thus readily/easily calculated). Lastly, observe that for the *VGRU-d + no-loss* in Table 5.9 there is a drastic drop in performance on periodic actions like walking and smoking when compared to aperiodic actions like discussion. Interestingly enough, for discussion, this model ablation achieves the best NPSS score. This possibly indicates that although progress has been made highly aperiodic actions such as discussion where there are no cyclic or obvious cues before the movement of a hand or a leg, purely using motion capture data alone is not a complete solution. Audio-visual information of the surroundings in such cases can give important information in such cases to help the model get a complete picture of the actor and his actions.

	Walking						Eating						Smoking						Discussion					
models	80	160	320	400	560	1000	80	160	320	400	560	1000	80	160	320	400	560	1000	80	160	320	400	560	1000
GRU + no-d	1.410	1.436	1.412	1.419	1.471	1.541	1.318	1.366	1.459	1.531	1.627	1.771	2.108	2.215	2.327	2.382	2.452	2.614	1.847	2.095	2.083	1.989	2.186	2.144
VGRU-d + no-loss	1.210	1.294	1.408	1.424	1.477	1.550	1.139	1.230	1.346	1.427	1.503	1.635	1.689	1.930	2.273	2.350	2.433	2.533	1.499	1.837	1.974	1.970	2.327	2.507
	± 1e-5	± 1e-5	± 2e-5	± 5e-5	± 5e-5	± 6e-5	± 1e-5	± 1e-5	± 1e-5	± 2e-5	± 4e-5	± 4e-5	± 1e-4	± 1e-4	± 3e-4	± 3e-4	± 4e-4	± 2e-4	± 1e-5	± 1e-5	± 1e-5	± 1e-5	± 1e-5	± 1e-5
GRU-d	1.311	1.333	1.369	1.364	1.350	1.370	1.275	1.305	1.386	1.466	1.530	1.702	1.943	2.062	2.201	2.255	2.342	2.486	1.744	1.980	2.026	1.994	2.214	2.172
VGRU-d	1.108	1.146	1.211	1.200	1.220	1.280	1.090	1.160	1.240	1.330	1.370	1.560	1.670	1.800	1.940	1.980	2.060	2.320	1.749	2.037	2.011	1.868	2.088	2.318
	± 1e-4	± 1e-4	± 2e-4	± 2e-4	± 3e-4	± 2e-4	± 2e-4	± 1e-4	± 1e-4	± 1e-4	± 1e-4	± 1e-4	± 1e-4	± 1e-4	± 1e-4	± 1e-4	± 1e-4	± 1e-4	± 1e-4	± 1e-4	± 1e-4	± 1e-4	± 1e-4	± 1e-4	± 1e-4

Table 5.8. Ablation study on long-term motion synthesis models. The MSE of euler angles on test set sequences is shown.

Models	Walking	Eating	Smoking	Discussion
GRU + no-d	1.138	1.147	1.443	1.812
VGRU-d + no-loss	1.541	0.911	1.474	**1.621**
GRU-d	0.931	**0.836**	<u>1.274</u>	<u>1.688</u>
VGRU-d	**0.887**	<u>0.846</u>	**1.235**	1.777

Table 5.9. Test-set NPSS scores for ablation study models (lower is better).

5.6 Future Work

Possible future avenues to explore in this problem would be to investigate generative variants of the current models that are capable of producing multiple and/or hybrid/unseen actions for long-term motion. The idea of attention model (conditioned on current state and action class) applied to RNN-based models during decoding is also an interesting avenue of exploration for improving the performance in the multi-action setting. A broader study on multiple motion capture datasets would give the NPSS metric more credence. The loss function with separate terms for *"open loop"* and *"closed loop"* predictions can be explored in greater detail and depth to help alleviate the exposure bias while using RNNs on any general problem/dataset.

Bibliography

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

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

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

[4] Sidenbladh, H., M. J. Black, and L. Sigal (2002) "Implicit probabilistic models of human motion for synthesis and tracking," in *European conference on computer vision*, Springer, pp. 784–800.

[5] Levine, S., J. M. Wang, A. Haraux, Z. Popović, and V. Koltun (2012) "Continuous character control with low-dimensional embeddings," *ACM Transactions on Graphics (TOG)*, **31**(4), p. 28.

[6] Kovar, L., M. Gleicher, and F. Pighin (2002) "Motion Graphs," *ACM Trans. Graph.*, **21**(3), pp. 473–482.
URL `http://doi.acm.org/10.1145/566654.566605`

[7] Boulic, R., N. M. Thalmann, and D. Thalmann (1990) "A global human walking model with real-time kinematic personification," *The visual computer*, **6**(6), pp. 344–358.

[8] Song, Y., D. Demirdjian, and R. Davis (2012) "Continuous Body and Hand Gesture Recognition for Natural Human-computer Interaction," *ACM Trans. Interact. Intell. Syst.*, **2**(1), pp. 5:1–5:28.
URL `http://doi.acm.org/10.1145/2133366.2133371`

[9] Wang, J. M., D. J. Fleet, and A. Hertzmann (2008) "Gaussian process dynamical models for human motion," *IEEE transactions on pattern analysis and machine intelligence*, **30**(2), pp. 283–298.

[10] Pavlovic, V., J. M. Rehg, and J. MacCormick (2001) "Learning switching linear models of human motion," in *Advances in neural information processing systems*, pp. 981–987.

[11] LeCun, Y., Y. Bengio, and G. Hinton (2015) "Deep learning," *nature*, **521**(7553), p. 436.

[12] Krizhevsky, A., I. Sutskever, and G. E. Hinton (2012) "ImageNet Classification with Deep Convolutional Neural Networks," in *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'12, Curran Associates Inc., USA, pp. 1097–1105.
URL http://dl.acm.org/citation.cfm?id=2999134.2999257

[13] Bahdanau, D., K. Cho, and Y. Bengio (2014) "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*.

[14] Ororbia II, A. G., T. Mikolov, and D. Reitter (2017) "Learning simpler language models with the differential state framework," *Neural computation*, **29**(12), pp. 3327–3352.

[15] Sutskever, I., O. Vinyals, and Q. V. Le (2014) "Sequence to Sequence Learning with Neural Networks," in *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, MIT Press, Cambridge, MA, USA, pp. 3104–3112.
URL http://dl.acm.org/citation.cfm?id=2969033.2969173

[16] Ionescu, C., D. Papava, V. Olaru, and C. Sminchisescu (2014) "Human3.6M: Large Scale Datasets and Predictive Methods for 3D Human Sensing in Natural Environments," *IEEE Trans. Pattern Anal. Mach. Intell.*, **36**(7), pp. 1325–1339.
URL https://doi.org/10.1109/TPAMI.2013.248

[17] Taylor, G. W., G. E. Hinton, and S. T. Roweis (2007) "Modeling Human Motion Using Binary Latent Variables," in *Advances in Neural Information Processing Systems 19* (B. Schölkopf, J. C. Platt, and T. Hoffman, eds.), MIT Press, pp. 1345–1352.

[18] Bregler, C. and J. Malik (1998) "Tracking People with Twists and Exponential Maps," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, CVPR '98, IEEE Computer Society, Washington, DC, USA, pp. 8–.
URL http://dl.acm.org/citation.cfm?id=794191.794776

[19] II, A. G. O. and A. Mali (2018) "Biologically Motivated Algorithms for Propagating Local Target Representations," *CoRR*, **abs/1805.11703**, 1805.

11703.
URL http://arxiv.org/abs/1805.11703

[20] Kingma, D. P., T. Salimans, and M. Welling (2015) "Variational Dropout and the Local Reparameterization Trick," in *Advances in Neural Information Processing Systems 28* (C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, eds.), Curran Associates, Inc., pp. 2575–2583.

[21] Chung, J., Ç. Gülçehre, K. Cho, and Y. Bengio (2014) "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling," *CoRR*, **abs/1412.3555**, 1412.3555.
URL http://arxiv.org/abs/1412.3555

[22] Hochreiter, S. and J. Schmidhuber (1997) "Long Short-Term Memory," *Neural Comput.*, **9**(8), pp. 1735–1780.
URL http://dx.doi.org/10.1162/neco.1997.9.8.1735

[23] Serban, I. V., A. Sordoni, R. Lowe, L. Charlin, J. Pineau, A. C. Courville, and Y. Bengio (2017) "A Hierarchical Latent Variable Encoder-Decoder Model for Generating Dialogues." in *AAAI*, pp. 3295–3301.

[24] Serban, I. V., A. Ororbia II, J. Pineau, and A. Courville (2017) "Piecewise Latent Variables for Neural Variational Text Processing," in *Proceedings of the 2nd Workshop on Structured Prediction for Natural Language Processing*, pp. 52–62.

[25] Lamb, A. M., A. G. ALIAS PARTH GOYAL, Y. Zhang, S. Zhang, A. C. Courville, and Y. Bengio (2016) "Professor Forcing: A New Algorithm for Training Recurrent Networks," in *Advances in Neural Information Processing Systems 29* (D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, eds.), Curran Associates, Inc., pp. 4601–4609.

[26] Zhou, Y., Z. Li, S. Xiao, C. He, Z. Huang, and H. Li (2018) "Auto-Conditioned Recurrent Networks for Extended Complex Human Motion Synthesis," in *International Conference on Learning Representations*.
URL https://openreview.net/forum?id=r11Q2SlRW

[27] Rubner, Y., C. Tomasi, and L. J. Guibas (2000) "The earth mover's distance as a metric for image retrieval," *International journal of computer vision*, **40**(2), pp. 99–121.

[28] Grauman, K. and T. Darrell (2004) "Fast contour matching using approximate earth mover's distance," in *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, vol. 1, IEEE, pp. I–I.

[29] Tieleman, T. and G. Hinton (2012), "Lecture 6.5—RmsProp: Divide the gradient by a running average of its recent magnitude," COURSERA: Neural Networks for Machine Learning.

[30] Zaremba, W., I. Sutskever, and O. Vinyals (2014) "Recurrent Neural Network Regularization," *CoRR*, **abs/1409.2329**, 1409.2329.
URL http://arxiv.org/abs/1409.2329

[31] Pham, V., T. Bluche, C. Kermorvant, and J. Louradour (2014) "Dropout improves recurrent neural networks for handwriting recognition," in *Frontiers in Handwriting Recognition (ICFHR), 2014 14th International Conference on*, IEEE, pp. 285–290.