Under review for publication in IJRR
XX(X):1–21
©The Author(s) 2017

Learning a bidirectional mapping between human whole-body motion and natural language using deep recurrent neural networks

Matthias Plappert, Christian Mandery and Tamim Asfour

Abstract

Linking human whole-body motion and natural language is of great interest for the generation of semantic representations of observed human behaviors as well as for the generation of robot behaviors based on natural language input. While there has been a large body of research in this area, most approaches that exist today require a symbolic representation of motions (e.g. in the form of motion primitives), which have to be defined a-priori or require complex segmentation algorithms. In contrast, recent advances in the field of neural networks and especially deep learning have demonstrated that sub-symbolic representations that can be learned end-to-end usually outperform more traditional approaches, for applications such as machine translation. In this paper we propose a generative model that learns a bidirectional mapping between human whole-body motion and natural language using *deep recurrent neural networks* (RNNs) and *sequence-to-sequence learning*. Our approach does not require any segmentation or manual feature engineering and learns a distributed representation, which is shared for all motions and descriptions. We evaluate our approach on 2 846 human whole-body motions and 6 187 natural language descriptions thereof from the *KIT Motion-Language Dataset*. Our results clearly demonstrate the effectiveness of the proposed model: We show that our model generates a wide variety of realistic motions only from descriptions thereof in form of a single sentence. Conversely, our model is also capable of generating correct and detailed natural language descriptions from human motions.

Keywords

human whole-body motion, natural language, sequence-to-sequence learning, recurrent neural network

1 Introduction

An intriguing way to instruct a robot is to first demonstrate the task at hand. In such a setup, a human teacher performs the necessary steps while the robot observes the human's motion. This way of robot programming is commonly referred to as *programming by demonstration* (Kuniyoshi et al. 1994; Dillmann et al. 2000; Billard et al. 2008) and has been extensively studied. However, observing only the motion of a human teacher is often not sufficient as the demonstrator will often include additional or corrective instructions to the student using natural language. In other words, the teacher-student interaction is inherently multi-modal.

Natural language presents itself as an intuitive way of communicating with the robot since it can be used to describe even rather complex motions and their parameterizations. For example, the description *"A person waves with the left hand five times."* encodes the motion (*waving*), the body part that should perform it (*left hand*) and the number of repetitions (*five times*). Enabling a robot to combine such rich descriptions in natural language with human whole-body motion therefore facilitate a much richer human-robot communication.

In recent years, deep learning (LeCun et al. 2015; Goodfellow et al. 2016) has proven to be very successful in computer vision (Krizhevsky et al. 2012; He et al. 2015), natural language processing (Sutskever et al.

All authors are with the High Performance Humanoid Technologies (H^{2}T) lab, Karlsruhe Institute of Technology (KIT), Germany

Corresponding author:
Matthias Plappert, High Performance Humanoid Technologies (H^{2}T), Institute for Anthropomatics and Robotics (IAR), Karlsruhe Institute of Technology (KIT), Adenauerring 2, 76131 Karlsruhe, Germany
Email: matthias.plappert@partner.kit.edu

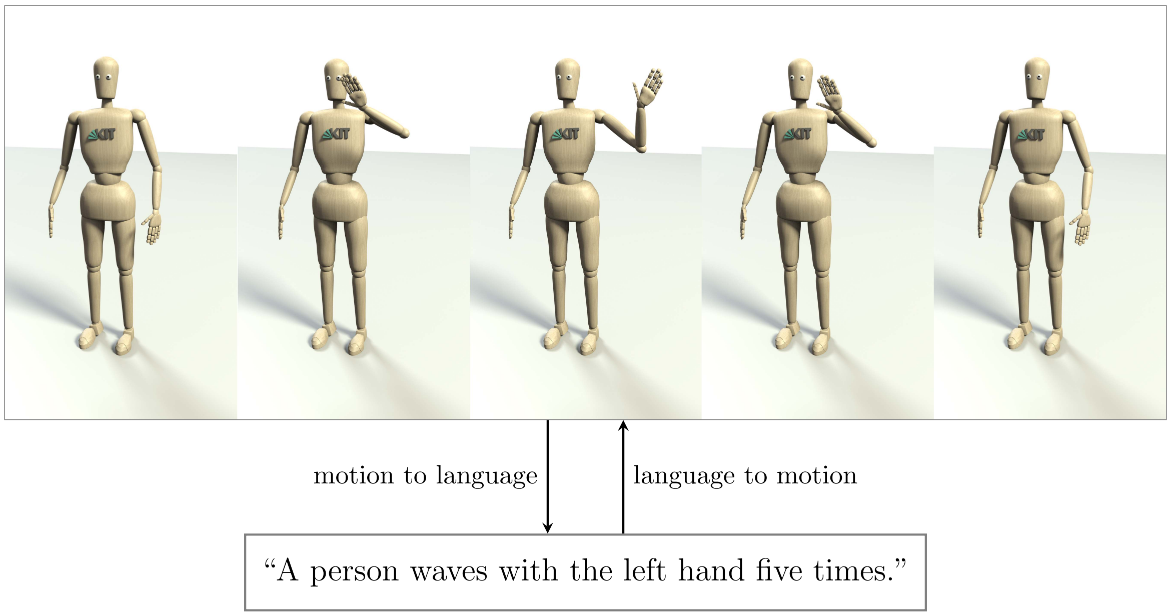


Figure 1. Illustration of the desired bidirectional mapping between human whole-body motion (top) and natural language (bottom).

2014; Glorot et al. 2011; Wu et al. 2016) and speech recognition (Graves et al. 2013). More recently, researchers have also reported promising results when applying deep learning techniques to problems in robotics (Levine et al. 2016; Gu et al. 2016; Levine et al. 2015).

In this paper, we use deep learning techniques to link human whole-body motion and natural language. More specifically, we make use of *sequence-to-sequence learning* (Sutskever et al. 2014) to learn a bidirectional mapping between human whole-body motion and natural language. Human whole-body motion is represented in joint space under the *Mater Motor Map* (MMM) (Terlemez et al. 2014) framework and descriptions thereof are in the form of complete English sentences. Figure 1 illustrates the desired mapping.

On one hand, this mapping allows us to generate rich descriptions of observed human motion, which can, for example, be used in a motion database. On the other hand, our model is capable of generating a wide range of different motions only from a description thereof in natural language. Even more so, the proposed system is capable of successfully synthesizing certain variations of motion, e.g. waving with the left or the right hand as well as walking quickly or slowly simply by specifying this parametrization in the natural language description.

The remainder of this paper is organized as follows. In Section 2 we review work that is related to our approach of combining human motion and natural language. Section 3 describes in detail how we represent both modalities, human motion and natural language, for use in the proposed bidirectional mapping. The model that is used to learn this mapping is presented in Section 4. In Section 5 we show that the proposed approach is capable of learning the desired bidirectional mapping. We also analyze the model and its learned representations in depth. Finally, Section 6 summaries and discusses our results and points out promising areas for future work.

2 Related work

Different models to encode human motion have been proposed in the literature. Takano et al. (2006), Kulic et al. (2008) and Herzog et al. (2008) use *hidden Markov models* (HMMs) to learn from observation. Their model can also be used to generate motion sequences by sampling from it. Taylor et al. (2006) and Taylor and Hinton (2009) propose *conditional restricted Boltzmann machines* (CRBMs) to learn from and then generate human whole-body motion. Calinon et al. (2007) use a *Gaussian mixture model* (GMM) to encode motion data. A different approach is proposed by Schaal (2006), who uses *dynamic movement primitives* (DMPs) to model motion using a set differential equations. More recently, Fragkiadaki et al. (2015) and Jain et al. (2015) have used *recurrent neural networks* (RNNs) to learn to generate

human motion from observation. Mordatch et al. (2015) have combined *trajectory optimization* and *deep neural networks* to generate complex and goal-directed movement for a diverse set of characters.
Bütepage et al. (2017) use deep feed-forward neural networks and an encoder-decoder architecture to learn a lower-dimensional latent representation that can be used for classification and motion generation.

While many models have been proposed to encode human motion, less research has been conducted on the question how to combine human motion and natural language. Takano et al. (2007) and Takano and Nakamura (2015b) describe a system that allows to learn a mapping between human motion and word labels. The authors segment human motion and encode the resulting motion primitives into hidden Markov models, which then form the *motion symbol space* or *proto symbol space*. Similarly, the *word space* is constructed from the associated word labels. Finally, the authors describe a projection between the motion symbol space and word space, which allows to obtain a sequence of motion symbols from a sequence of words and vice versa.

Takano and Nakamura (2008, 2009, 2012, 2015a) learn a bidirectional mapping between human motion and natural language in the form of a complete sentence. The authors propose two components that, when combined, realize the desired mapping. In the *motion language model*, motion primitives, which are encoded into HMMs, are probabilistically related to words using latent variables. These latent variables represent non-observable properties like the semantics. The conditional probabilities that govern the association between motion and language are obtained by using the EM algorithm. The second part, the *natural language model*, captures the syntactical structure of natural language. Different approaches to realize this model have been described by the authors, e.g. HMMs (Takano and Nakamura 2008, 2009) or bigram models (Takano and Nakamura 2012, 2015a). Both models are finally combined to generate natural language descriptions of motion by first recognizing the motion, then obtaining an unordered set of likely words associated with that motion (using the motion language model) and finally finding a likely sequence of these words (using the natural language model). This approach is commonly referred to as *bag-of-words*. Similarly, the most likely motion can be obtained from a description thereof in natural language by searching for the motion symbol with maximum likelihood given the word sequence.

Medina et al. (2012) present an approach where motion symbols in the form of motion primitives are learned using *parametric hidden Markov models* (PHMMs). Adverbs (e.g. *slowly*) are used to parametrize the PHMMs and the natural language is modeled in a similar way to aforementioned works using a bigram language model. Although their approach can enable the generation of textual representations from motions, the presented evaluation only covers the other direction of motion generation from textual descriptions. This evaluation considers a rather limited set of 7 different motion primitives and 13 words, and is based on a virtual scenario of a human-robot cooperation task with a 2 degrees of freedom haptic interface. A different approach is described by Sugita and Tani (2005) and Ogata et al. (2007a,b) where the authors use a *recurrent neural network with parametric bias* (RNNPB) model to combine movement of simple robots (e.g. a robot platform or a robot arm) with simple commands in the form of words (e.g. *push red*). The authors demonstrate that their approach can generate the appropriate trajectories corresponding to a given command and vice versa. Attempts have been made to gradually increase the complexity and supported variety of the commands (Arie et al. 2010; Ogata and Okuno 2013).

In recent years, there has been extensive work in the field of deep learning to combine natural language with other modalities like images (Karpathy and Li 2015; Vinyals et al. 2015) and videos (Donahue et al. 2015; Venugopalan et al. 2015). A recent survey on proposed solutions to the problem of visualizing natural language descriptions is given in Hassani and Lee (2016).

3 Data representation

3.1 Human whole-body motion

In this work, we only consider human whole-body motion that has been recorded with an optical marker-based motion capture system. Briefly speaking, such a system observes a set of reflective markers placed on specific anatomical landmarks of the human subject using multiple cameras that are positioned around the subject. The Cartesian coordinates of each marker can then be reconstructed using triangulation. For an in-depth discussion of motion capture techniques, we refer the reader to Field et al. (2011).

While this approach allows for highly accurate motion acquisition, the resulting representation of motion in the form of marker trajectories has several drawbacks. First, the positions of the markers depend on the reference coordinate system, which varies across recordings and thus would require some sort of normalization to obtain an invariant representation. Second, different marker sets with a varying number

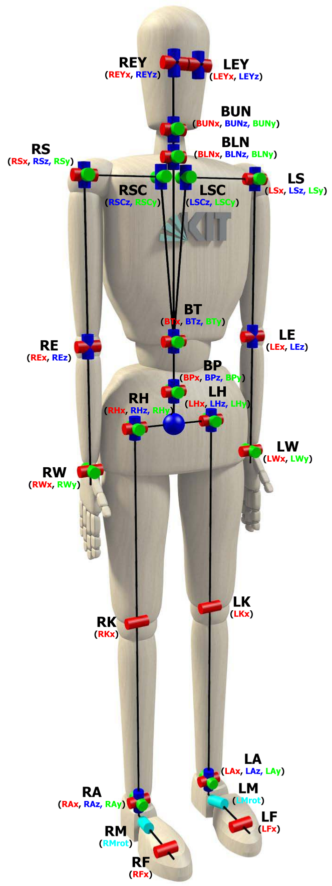


Figure 2. The kinematic model of the *Master Motor Map* (MMM) framework (Mandery et al. 2016).

of markers or different marker locations on the human body may be used. Third, the data is high-dimensional since each marker position requires three dimensions and usually more than 50 markers are used.

We therefore use the *Master Motor Map* (MMM) framework (Mandery et al. 2016; Terlemez et al. 2014; Azad et al. 2007) to represent human whole-body motion in *joint space*. This is realized by the MMM reference model, which specifies the kinematics of the human body (see Figure 2). The conversion from Cartesian to joint space is then achieved by minimizing the squared distance between the physical markers on the human subject and the virtual markers on the reference model w.r.t. to the joint angles of the kinematic model. The resulting joint representation is no longer dependent on a reference coordinate system, abstracts away the concrete marker set that was used during recording and has significantly lower dimensionality than the Cartesian representation.

Throughout this paper, we use $J = 44$ joints of the MMM reference model to represent human motion, which are distributed over the torso, arms and legs of the model. The remainder of the degrees of freedom that the model features (e.g. individual fingers and eyes) are not used since they are less important for this work and also hard to track.

We also introduce an additional binary feature which is enabled as long as the motion is ongoing. This feature is necessary for two reasons: First, we pad all sequences to have equal length due to implementation details and the binary flag indicates the active part of the motion. Second and more importantly, the generative part of our proposed model will also predict the binary flag and thus can be used to indicate if the generation is still ongoing or if it has finished.

More formally, each motion $\boldsymbol{M}$ is thus represented as a sequence of length N

$$\boldsymbol{M} = \left(\boldsymbol{m}^{(1)}, \boldsymbol{m}^{(2)}, \ldots, \boldsymbol{m}^{(t)}, \ldots, \boldsymbol{m}^{(N)}\right), \qquad (1)$$

where each timestep $\boldsymbol{m}^{(t)}$ is defined as

$$\boldsymbol{m}^{(t)} \in \mathbb{R}^J \times \{0, 1\}. \qquad (2)$$

Furthermore, we scale the individual joints to have zero mean and variance one. We also down-sample each motion from 100 Hz to 10 Hz, which results in shorter sequences that are less resource-expensive during training and evaluation of our model. However, we do not discard the remaining motion data since we split each original motion sequence into 10 down-sampled sequences by applying a variable offset $t_0 \in \{0, \ldots, 9\}$. We then treat these as additional data during training, effectively introducing noise into the training process. Since we do this for all sequences, the original distribution of different motion types in the

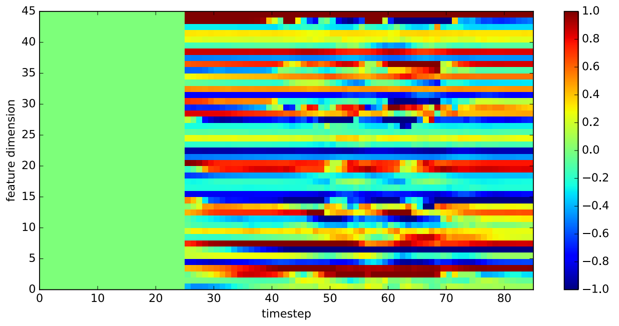


Figure 3. Representation of an exemplary human motion in which the subject walks forward. The first $J = 44$ dimensions are the joint values of the MMM model and dimension $J + 1 = 45$ is the binary active feature, plotted over time. The motion is padded, which explains the constant values for the first 25 timesteps. Notice that the motion is not active during this part, as indicated by the binary feature.

dataset is maintained. Figure 3 illustrates the described representation of a single human whole-body motion.

3.2 Natural language descriptions

We represent the natural language annotations on the word-level. More concretely, we first normalize by transforming each sentence to lower case letters, remove all punctuation and apply minor spelling corrections for commonly misspelled words. We also pad all sequences to have equal length by introducing a special `PAD` word and, similarly, use special `SOS` (start of sentence) and `EOS` (end of sentence) words to mark the start and end of a sentence, respectively. Next, we tokenize each sentence into individual words and assign each word a unique integer index.

More formally, each sentence $\boldsymbol{w}$ is thus represented as a sequence of length M

$$\boldsymbol{w} = \left(w^{(1)}, w^{(2)}, \ldots, w^{(t)}, \ldots, w^{(M)}\right), \tag{3}$$

where each word is represented as an integer $w^{(t)} \in \mathbb{N}$. Note that, typically, $M \ll N$ (recall that N denotes the motion sequence length), i.e. the motion and language sequences are not required to have equal length.

In practice, we encode each word using one-hot encoding $\bar{\boldsymbol{w}}^{(t)} \in \{0,1\}^V$ over a vocabulary of size V instead of using the integer representation (since this would imply some order):

$$\bar{w}_i^{(t)} = \begin{cases} 1, & \text{if } i = w^{(t)} \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

However, this representation has the problem that its dimensionality grows linearly with V (the number of words in the vocabulary). *Word embeddings* (Mikolov et al. 2013) resolve this problem by projecting the one-hot encoded word to a continuous, but much lower-dimensional *embedding space*. Depending on the training procedure, these embeddings have also been shown to group semantically similar words closer together.

In this work, we learn this projection end-to-end when training the entire model. However, in future work we might consider to use the weights of an embedding layer that was pre-trained on a large text corpus, e.g. using the open source `word2vec` implementation.[*]

4 Model

We describe the two directions of the mapping between human motion and natural language separately. However, since both models share a common approach commonly referred to as *sequence-to-sequence learning* (Sutskever et al. 2014), we explain this shared mode of operation before describing each model individually.

4.1 Shared mode of operation

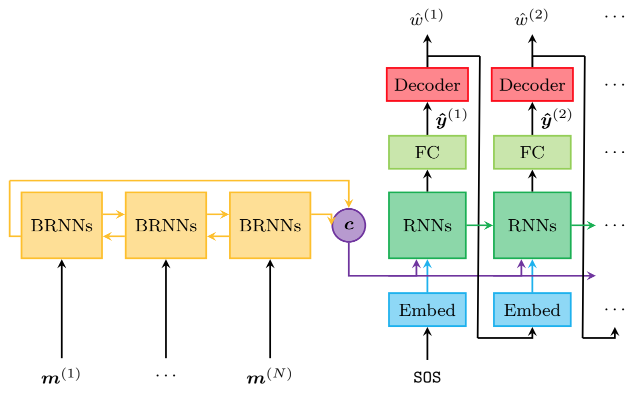
Sequence-to-sequence learning has been used with great success, e.g. in large-scale machine translation (Wu et al. 2016). As the name suggests, the goal of such models is to generate a target sequence from an input sequence, where the sequences can differ in length and modality. This property makes sequence-to-sequence learning an excellent fit for the purpose of learning a mapping between human motion and natural language.

We model each direction, that is from human motion to natural language as well as from natural language to human motion, individually. Figure 4 depicts the details of both models. In both cases, the input sequence is first transformed into a latent *context vector* $\boldsymbol{c}$ by a recurrent neural network (RNN) or a stack thereof (meaning that many recurrent layers are stacked). The output of the recurrent encoder network after the last timestep of the input sequence has been processed is used as the context vector. This context vector is then decoded by the *decoder network* and, therefore, acts as the coupling mechanism between encoder and decoder network. Different approaches have been proposed for this (e.g. initializing the hidden state of the decoder with the context vector) but we provide the context vector as input to the decoder network at each timestep. The decoder then produces the desired target sequence step by step.

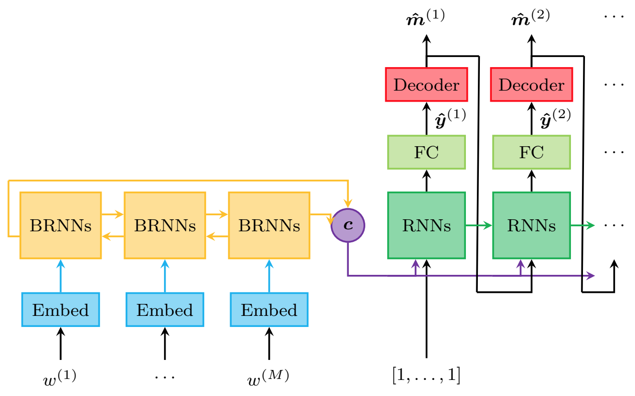
Typically, more advanced architectures than a vanilla RNN like *long short-term memory* (LSTM) (Hochreiter and Schmidhuber 1997; Gers et al. 2000) or *gated recurrent units* (GRUs) (Cho et al. 2014; Chung et al. 2014) are used. The encoder often uses *bidirectional RNNs* (BRNNs) (Graves and Schmidhuber 2005), which process the input sequence in both directions and then combine the computed latent representations, e.g. by concatenation. While we use GRUs and a bidirectional encoder in this work, the proposed model can be used with any recurrent network architecture.

We model the decoder output probabilistically, which means that the network predicts the parameters of some probability distribution instead of predicting the output value directly. This intermediate output is denoted as $\hat{\boldsymbol{y}}^{(t)}$ and its form depends on the

[*]`https://code.google.com/archive/p/word2vec/`



(a) Motion-to-language model.



$\hat{\boldsymbol{m}}^{(1)}$
$\hat{\boldsymbol{m}}^{(2)}$
Decoder
$\hat{\boldsymbol{y}}^{(1)}$
$\hat{\boldsymbol{y}}^{(2)}$
FC
BRNNs
$\boldsymbol{c}$
RNNs
Embed
$w^{(1)}$
$w^{(M)}$
$[1,\dots,1]$

(b) Language-to-motion model.

Figure 4. Overview of the proposed models for both directions. Figure 4a depicts the model that learns a mapping from a motion to language by first *encoding* the motion sequence $\boldsymbol{M} = (\boldsymbol{m}^{(1)} \dots \boldsymbol{m}^{(N)})$ into a *context vector* $\boldsymbol{c}$ (in purple) using a *stack of bidirectional RNNs* (BRNNs, in yellow). The context vector is then *decoded* by *another stack of unidirectional RNNs* (in green), which also takes the *embedded word* (in blue) generated in the previous timestep as input. A fully-connected layer (FC, in light green) produces the parameters of the output probability distribution, denoted as $\hat{\boldsymbol{y}}^{(t)}$. The *decoder* (in red) finally takes $\hat{\boldsymbol{y}}^{(t)}$ and transforms it into a concrete word $\hat{w}^{(t)}$. Combined, the model thus generates the corresponding description word by word until the special EOS token is emitted and the description $\hat{\boldsymbol{w}} = (\hat{w}^{(1)}, \hat{w}^{(2)}, \dots)$ is obtained. The other direction from language to motion is depicted in Figure 4b. The model works in similar fashion but uses a description in natural language $\boldsymbol{w} = (w^{(1)} \dots w^{(M)})$ to generate the corresponding whole-body motion $\hat{\boldsymbol{M}} = (\hat{\boldsymbol{m}}^{(1)}, \hat{\boldsymbol{m}}^{(2)}, \dots)$. The models for both directions are trained individually and share no weights.

specific mapping (i.e. motion-to-language vs. language-to-motion) and are therefore described in Section 4.2 and Section 4.3 in more detail.

The *decoder*[†] finally decodes this probabilistic representation to a concrete deterministic instance. One way to do this would be to greedily select the instance with highest probability under the distribution. However, such a strategy does not necessarily yield the sequence with highest probability. On the other hand, expanding each possible node is computational expensive for the discrete case and intractable for the continuous case. We therefore use a common middle-ground between these two extremes: *beam search* (Huang et al. 2001, chapter 12). Beam search is a modification of best-first search, where in each step, only a limited set of stored nodes is considered for expansion. Since the output of the network should depend on the decision of the decoder (which is made *outside* of the decoder RNNs), we feed back this decision in each timestep. The input at timestep t is thus the concatenation of the decoded output from the previous timestep and the context vector, which is constant for all timesteps.

Finally, the encoder and decoder network can jointly be trained end-to-end using *back-propagation through time* (BPTT) (Werbos 1990).

4.2 *Motion-to-language mapping*

Having described the general sequence-to-sequence framework that is used throughout this work (depicted in Figure 4a), we now explain the concrete case of mapping from human motion to natural language. To this end, the encoder network takes a motion sequence $\boldsymbol{M}$ as its input and encodes it into a context vector $\boldsymbol{c}$. The decoder network then, step by step, produces the desired description in natural language $\hat{\boldsymbol{w}} = (\hat{w}^{(1)}, \hat{w}^{(2)}, \dots)$ from this context vector, as described in the previous section.

4.2.1 Model architecture: The architecture of the encoder network is straightforward. We use stacked bidirectional RNNs to compute the context vector given the motion sequence. More concretely, we set the context vector to be the output of the last RNN layer after it has processed all timesteps of the input sequence. In all layers, the outputs of the forward and backward processing (due to the bidirectional model) are concatenated before being passed to the next layer as input.

As mentioned before, our approach uses a probabilistic formulation of the decoder. Fortunately, this can be

[†]Note that the decoder is an element of the larger decoder network and the two are *not* the same

achieved by defining a discrete probability distribution over the entire vocabulary. This is realized in our model by a softmax layer as the final layer of the decoder network:

$$\hat{y}_i^{(t)} = \frac{\exp z_i^{(t)}}{\sum_j \exp z_j^{(t)}}, \tag{5}$$

where $z_i^{(t)}$ denotes the unnormalized activation of the i-th output neuron, which corresponds to the i-th item in the vocabulary. This can be interpreted as the probability of the i-th item in the vocabulary conditioned on the input motion encoded by the context vector and on the previously emitted words encoded by the hidden state of the recurrent decoder network:

$$P\left(\hat{w}^{(t)} = i \;\middle|\; \boldsymbol{M}, \hat{w}^{(t-1)}, \ldots, \hat{w}^{(1)}\right) := \hat{y}_i^{(t)}. \tag{6}$$

The decoder network also uses stacked RNNs but connects them in such a way that each RNN has access to the context vector, the embedding of the previously emitted word and the output of the preceding RNN (if applicable). Finally, the fully-connected softmax layer that produces the discrete probability distribution over the vocabulary as described above has access to the output of each RNN layer. Details on the number of layers, units per layer and other hyperparameters are detailed in Section 5. A detailed schematic of the model architecture can also be found in Appendix A.

4.2.2 Training: We can train the entire model end-to-end by minimizing the categorical cross-entropy loss:

$$\mathcal{L} = -\frac{1}{T}\sum_{t=1}^{T}\sum_i y_i^{(t)} \log \hat{y}_i^{(t)}. \tag{7}$$

$\boldsymbol{y}^{(t)}$ denotes the ground truth at timestep t in the form of a reference natural language description and $\hat{\boldsymbol{y}}^{(t)}$ denotes the corresponding prediction of the model. We use BPTT with mini-batches to update the network parameters. The exact hyperparameters of the training procedure are described in Section 5.

4.2.3 Decoding: During prediction, we face the problem of deciding on a concrete word $\hat{w}^{(t)}$ given the predicted probabilities $\hat{\boldsymbol{y}}^{(t)}$. As mentioned before, we use beam search as a middle ground between greedily selecting the word with highest probability and performing an exhaustive search in the space of possible word sequences. More concretely, we expand each of the candidate sequences by predicting W different probability vectors $\hat{\boldsymbol{y}}_1^{(t)}, \ldots, \hat{\boldsymbol{y}}_K^{(t)}$ (beam search of width W) for each timestep. Given a vocabulary of size V, this yields $W \cdot V$ new candidates, of which we only keep around the W most likely sequences under our model. This process is then repeated iteratively until each candidate has been completed as indicated by the `EOS` token, resulting in W different descriptions.

Importantly, we also obtain the probability of each sequence by accumulating the product of all corresponding step-wise probabilities. This in turn allows us to rank the candidates according to their probability under the model.

4.3 Language-to-motion mapping

Although the approach for the language-to-motion mapping is similar to the previously describe motion-to-language mapping, the architecture of the decoder network is necessarily different since the output modality is now multi-dimensional and continuous.

4.3.1 Model architecture: The architecture of the encoder network is similar to what was already described in Section 4.2. A network of stacked bidirectional RNNs encode the input description into a context vector. The decoder uses stacked RNNs where each RNN layer has access to the context vector computed by the encoder network, the previously generated motion timestep and the output of the preceding RNN layers. The complete model is depicted in Figure 4b.

However, there is a significant difference in the final fully-connected layer since the problem now requires to output a multi-dimensional and continuous frame of the motion $\hat{\boldsymbol{m}}^{(t)}$ in each timestep (compared to the discrete word $\hat{w}^{(t)}$ as in Section 4.2). Furthermore, our approach uses a probabilistic decoder network, which allows us to generate non-deterministic outputs and also provides likelihood scores under our model for each generated sequence.

Recall that each motion frame is defined as $\hat{\boldsymbol{m}}^{(t)} \in \mathbb{R}^J \times \{0, 1\}$, where the first J dimensions are the joint values (and therefore continuous) and the last dimension is the binary flag that indicates the active parts of the motion. We use an approach similar to Graves (2013) based on a mixture of Gaussians for the continuous part and a Bernoulli distribution for the discrete part.

More concretely, the final fully-connected layer produces the following parameters, assuming a mixture model with K components and omitting time indices for better readability:

- K component weights $\hat{\alpha}_1, \ldots, \hat{\alpha}_K \in [0, 1]$ produced by a softmax activation, and therefore $\sum_k \hat{\alpha}_k = 1$).
- K mean vectors $\hat{\boldsymbol{\mu}}_1, \ldots, \hat{\boldsymbol{\mu}}_K \in \mathbb{R}^J$ produced by a linear activation.

- K variance vectors $\hat{\boldsymbol{\sigma}}_1, \dots, \hat{\boldsymbol{\sigma}}_K \in [0, \infty)^J$, assuming Gaussians with diagonal covariance matrices, produced by a softplus activation.
- The probability of being active $\hat{p} \in [0, 1]$ produced by a sigmoid activation.

To summarize, at each timestep the network predicts

$$\hat{\boldsymbol{y}}^{(t)} = [(\alpha_i^{(t)}, \hat{\boldsymbol{\mu}}_i^{(t)}, \hat{\boldsymbol{\sigma}}_i^{(t)})_{i=1,\dots,K}, \hat{p}^{(t)}],$$

which are the parameters for the mixture of Gaussians and Bernoulli distributions.

Given this formulation, we can define the likelihood of a given motion frame $\hat{\boldsymbol{m}}^{(t)}$ conditioned on the input description, as encoded by the context vector, as well as all previously emitted frames, encoded by the hidden state of the recurrent decoder network, as:

$$p\left(\hat{\boldsymbol{m}}^{(t)} \,\middle|\, \boldsymbol{w}, \hat{\boldsymbol{m}}^{(1)}, \dots, \hat{\boldsymbol{m}}^{(t-1)}\right) = \sum_{k=1}^{K} \left[\hat{\alpha}_k \, \mathcal{N}\left(\hat{\boldsymbol{m}}_{1:J}^{(t)} \,\middle|\, \hat{\boldsymbol{\mu}}_k^{(t)}, \hat{\boldsymbol{\sigma}}_k^{(t)}\right)\right] \cdot \mathcal{B}\left(\hat{m}_{J+1}^{(t)} \,\middle|\, \hat{p}^{(t)}\right). \tag{8}$$

Since we use diagonalized Gaussians, we can write the likelihood under each multivariate Gaussian as the product of J one-dimensional Gaussians, one for each joint:

$$\mathcal{N}\left(\boldsymbol{x} \,\middle|\, \boldsymbol{\mu}, \boldsymbol{\sigma}\right) = \prod_{j=1}^{J} \frac{1}{\sqrt{2\pi\sigma_j^2}} \exp\left(-\frac{(x_j - \mu_j)^2}{2\sigma_j^2}\right). \tag{9}$$

Finally, the probability mass for the discrete Bernoulli distribution is defined as

$$\mathcal{B}\left(x \,\middle|\, p\right) = \begin{cases} p, & \text{if } x = 1 \\ 1 - p, & \text{otherwise.} \end{cases} \tag{10}$$

Details on the number of layers, units per layer and other hyperparameters are detailed in Section 5. A detailed schematic of the model architecture can also be found in Appendix B.

4.3.2 Training: Again, we can train the entire model end-to-end by minimizing the following loss:

$$\mathcal{L} = \frac{1}{T} \sum_{t=1}^{T} \left[-\log \sum_{k} \hat{\alpha}_k \, \mathcal{N}\left(\boldsymbol{m}_{1:J}^{(t)} \,\middle|\, \hat{\boldsymbol{\mu}}^{(t)}, \hat{\boldsymbol{\sigma}}^{(t)}\right) - m_{J+1}^{(t)} \log \hat{p}^{(t)} - \left(1 - m_{J+1}^{(t)}\right) \log\left(1 - \hat{p}^{(t)}\right) \right]. \tag{11}$$

This loss consists of two parts. The first part describes the likelihood of the ground truth joint values (denoted as $\boldsymbol{m}_{1:J}^{(t)}$) under the predicted mixture distribution. The second part is the binary cross-entropy between the ground truth of the active flag $m_{J+1}^{(t)}$ and the predicted parameter of the Bernoulli distribution $\hat{p}^{(t)}$. By minimizing the loss $\mathcal{L}$, we jointly maximize the likelihood of the ground truth under the mixture of Gaussians and minimize the cross-entropy.

However, in practice the formulation in Equation 11 has numerical stability issues. This is because computing the likelihood under each multi-variate Gaussian requires computing the product of J individual likelihoods (compare Equation 9), which can easily be subject to both numerical under- and overflow.

We therefore define the following surrogate loss function:

$$\widetilde{\mathcal{L}} = \frac{1}{T} \sum_{t=1}^{T} \left[-\sum_{k} \hat{\alpha}_k^{(t)} \, \log \mathcal{N}\left(\boldsymbol{m}_{1:J}^{(t)} \,\middle|\, \hat{\boldsymbol{\mu}}^{(t)}, \hat{\boldsymbol{\sigma}}^{(t)}\right) - m_{J+1}^{(t)} \log \hat{p}^{(t)} - \left(1 - m_{J+1}^{(t)}\right) \log\left(1 - \hat{p}^{(t)}\right) \right] \tag{12}$$

This re-formulation allows us to replace the product from Equation 9 with a sum of logarithms, making the computation more robust against numerical problems. We find that minimizing $\widetilde{\mathcal{L}}$ instead of $\mathcal{L}$ works well and yields the desired results.

Like before, we use BPTT with mini-batches to train the network end-to-end. Again, the exact hyperparameters of the training procedure are described in Section 5.

4.3.3 Decoding: Similar to motion-to-language mapping (Section 4.2.3), we again face the problem of decoding a concrete motion frame $\hat{\boldsymbol{m}}^{(t)}$ from the prediction vector $\hat{\boldsymbol{y}}^{(t)} = [(\alpha_i^{(t)}, \hat{\boldsymbol{\mu}}_i^{(t)}, \hat{\boldsymbol{\sigma}}_i^{(t)})_{i=1,\dots,K}, \hat{p}^{(t)}]$. This can be solved by sampling the joint values from the multivariate Gaussian mixture distribution (by first sampling a mixture component with probabilities $\alpha_1^{(t)}, \dots, \alpha_K^{(t)}$ and then sampling from the selected multivariate Gaussian) as well as sampling from the Bernoulli distribution for the binary active indicator. Like before we use beam search to obtain a couple of candidates at the end of the decoding process as previously described in Section 4.2.3. The key difference here is that we cannot compute the likelihood for each discrete possibility anymore. We resolve this problem by sampling a couple of candidates for each hypothesis from the respective distributions and then truncating them to keep around a fixed number of hypotheses for the next timestep.

5 Experiments

5.1 Dataset

While large datasets for human motion exist (see also Mandery et al. (2016) for a recent review), our evaluation requires a dataset that also contains

Table 1. The hyperparameters for both models that were used for all experiments. When using bidirectional RNNs, the number of units per layer is given in the form $2 \times x$, where x is the number of units for each processing direction.

Hyperparameter	**Motion-to-Language**	**Language-to-Motion**
Encoder RNN type	bidirectional GRU	bidirectional GRU with layer normalization
Encoder layers	2 layers $(2 \times 64, 2 \times 64)$	2 layers $(2 \times 64, 2 \times 64)$
Decoder RNN type	GRU	GRU with layer normalization
Decoder layers	2 layers $(128, 128)$	3 layers $(400, 400, 400)$
Embedding dimension	64	64
Dropout rate	0.4	0.1
Optimizer	Adam with Nesterov	Adam with Nesterov
Learning rate	10^{-3}	10^{-3}
Gradient clipping	∞	25
Batch size	128	128
Training epochs	100	100
Vocabulary size (V)	1 344	1 344
Motion joints (J)	44	44
Mixture components	*n/a*	20
Trainable parameters	843 456	5 446 517
excl. embedding & FC layers	414 720	3 221 520

descriptions of such motion in natural language. We have recently proposed the *KIT Motion-Language Dataset* (Plappert et al. 2016), which uses human whole-body motions from the *KIT Whole-Body Human Motion Database*[‡] (Mandery et al. 2015b) and the *CMU Graphics Lab Motion Capture Database*[§]. For each motion, a set of descriptions in the form of a single English sentence was collected using a crowd-sourcing approach.

We use the `2016-10-10` release of the KIT Motion-Language Dataset for all experiments. This version of the dataset contains 3 911 recordings of human whole-body motion in the aforementioned MMM representation and 6 278 annotations in natural language. The dataset is publicly available[¶] so that all results in this paper can be reproduced.

For our evaluation, we filter out motions that have a duration of 30 seconds or more in order to reduce the computational overhead due to excessive padding. This results in 2 846 usable motion samples with a total duration of 5.3 hours and 6 187 natural language annotations that consist of 46 561 words in total. We randomly split the remaining data into training, validation and test sets with a ratio of 0.8, 0.1 and 0.1, respectively. All results are reported using the test set, if not otherwise indicated. The processing steps described in Section 3 are performed to obtain the representations suitable for training the model.

5.2 Setup

The model architectures for motion-to-language (Section 4.2) and language-to-motion (Section 4.3) have already been described. However, we have not yet described the specific hyperparameters that we used for our experimental results.

For both the encoder and decoder parts of each model, we use *gated recurrent units* (GRUs) (Cho et al. 2014) as the recurrent neural network architecture. We regularize all models with *dropout* (Srivastava et al. 2014; Gal 2015) in the embedding, recurrent and fully-connected layers. In both cases, we train our models using the *Adam* optimizer with Nesterov momentum (Kingma and Ba 2014; Dozat 2015). Training of the language-to-motion model proved to be more difficult, presumably due to the more complex dynamics of the recurrent model, which made it necessary to use both *gradient clipping* (Bengio et al. 2012) as well as *layer normalization* (Ba et al. 2016). We also experimented with *batch normalization* (Ioffe and Szegedy 2015; Cooijmans et al. 2016) instead of layer normalization but were unable to get good results

[‡] `https://motion-database.humanoids.kit.edu/`
[§] `http://mocap.cs.cmu.edu/`
[¶] `https://motion-annotation.humanoids.kit.edu/dataset/`

with it, presumably due to the known problems of batch normalization with padded sequences.[‖]

Table 1 summaries the aforementioned and lists all other hyperparameters for both models, language-to-motion and motion-to-language. These hyperparameters were used for all experiments throughout this work. The code that was used to obtain all following results is available online: `https://gitlab.com/h2t/DeepMotionLanguageMapping/`

5.3 Generating language from motion

5.3.1 Training. Optimizing the model turned out to be straightforward and did not require special measures. Figure 5 depicts the change in loss during training for the training and validation split, respectively. As can be seen, both training and validation loss continuously decrease, indicating that the model does not overfit on the training data. Additionally, the validation loss seems to have converged after 100 training epochs, indicating that our training time was sufficiently long.

5.3.2 Qualitative results: To provide insight into the style and quality of the natural language descriptions generated by our model, we present a few examples in Figure 6. For each depicted human whole-body motion, we compute five natural language description hypotheses (thus, we perform beam search with $W = 5$) and sort them in descending order by their respective log probability under the model (depicted in the tables below each motion; due to space constraints, we only present the top three descriptions per motion). As can be seen from these examples, all descriptions are complete English sentences with valid grammatical structure. This is a pattern that we observe throughout. Interestingly, the model also produces semantically identical descriptions with varying grammatical structure. For example, when describing standing up (Figure 6f), the model creates a description using present simple (*"a person stands up from the ground"*) and another one using present continuous (*"a person is standing up from the ground"*).

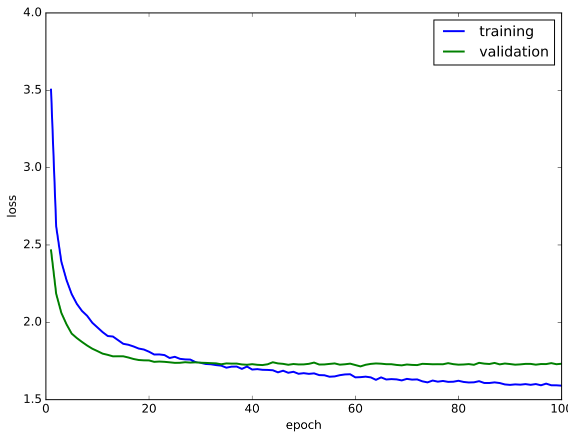


Figure 5. Training and validation loss during training of the motion-to-language model.

The model also uses synonyms interchangeably. For example, the model refers to the subject in the scene as a *"human"*, *"person"* or *"someone"*. This is especially apparent in Figure 6d, where the produced sentences are identical except for this variation.

The generated natural language descriptions are also rich in detail. For example, the model successfully differentiates between wiping a surface with the right (Figure 6h) vs. left (Figure 6i) hand and creates corresponding descriptions that mention the handedness of the motion. Similar behavior can be observed for the waving motion (Figure 6d), stomping motion (Figure 6e) and pushing motion (Figure 6c), for which the descriptions all correctly mention the correct hand, foot and perturbation direction, respectively.

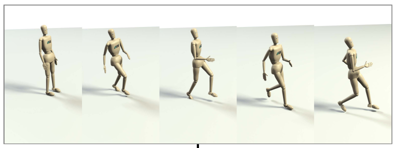
Overall, the contents of the generated descriptions are highly encouraging and demonstrate the capabilities of the model to not only generate syntactically valid descriptions, but also semantically meaningful and detailed ones.

5.3.3 Quantitative results: While the previously presented results allow us to gather some understanding of the generated descriptions, they do not necessarily represent the overall behavior of the model. Therefore, we provide in the following a quantitative evaluation of the performance of the proposed model over the training and test splits of our dataset.

We select the BLEU score (Papineni et al. 2002) to measure the performance of the proposed model. Briefly speaking, the BLEU score is a metric that was initially proposed to measure the performance of machine translation systems and has also been used in work targeting the similar problem of learning a mapping between human motion and natural language (Takano et al. 2016).

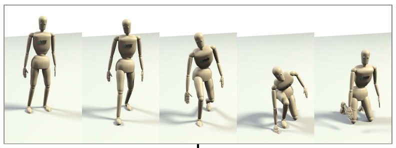
The BLEU score is obtained by first counting unigrams, bigrams, . . . , n-grams (in this work up to $n = 4$) in the provided reference and then computing the n-gram precision (with exhaustion) of the hypothesis. Essentially, it is this modified and weighted n-gram precision with some additional smoothing. The BLEU score is defined to be between

[‖]`https://github.com/cooijmanstim/recurrent-batch-normalization/issues/2`



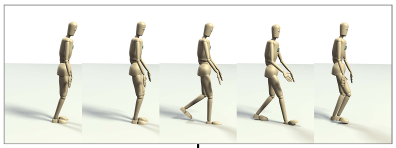
Description	log P
a person runs forward	-3.23
a person runs forwards	-4.13
a person is running	-4.16

(a) Running



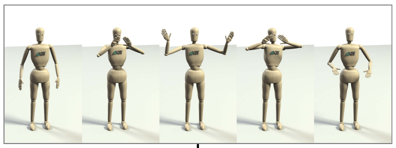
Description	log P
a person kneels down	-3.26
someone is kneeling down	-3.34
a person goes down on his knees	-3.63

(b) Kneeling down



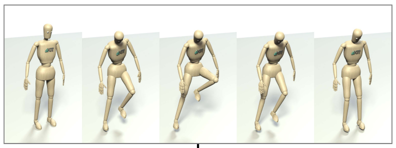
Description	log P
a person gets pushed backwards	-2.62
a person is pushed backwards	-3.21
a person stumbles backwards	-3.21

(c) External perturbation



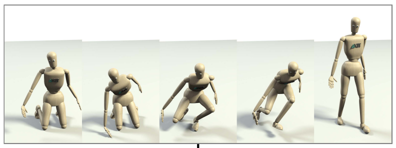
Description	log P
a person waves with both hands	-0.91
someone waves with both hands	-2.47
a human waves with both hands	-3.12

(d) Waving



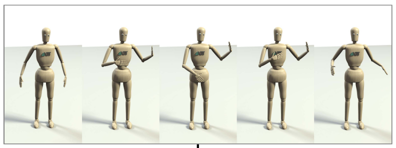
Description	log P
a person stomps the left foot	-3.01
a person stomps his left foot	-3.12
a person stomps with his left foot	-3.35

(e) Stomping



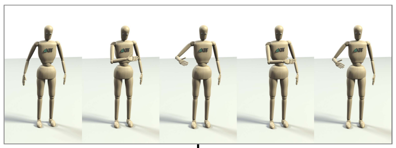
Description	log P
a person stands up from the ground	-2.30
a person stands up from a kneeling position	-3.14
a person is standing up from the ground	-3.63

(f) Standing up



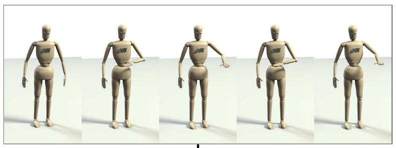
Description	log P
a person plays the guitar	-2.57
a person plays the air guitar	-2.76
someone plays air guitar	-2.93

(g) Playing guitar



Description	log P
a person wipes something with its right hand	-3.32
a person wipes a desk with his right hand	-3.64
a person wipes a surface with their right hand	-4.19

(h) Wiping (right hand)



Description	log P
a person wipes something with its left hand	-3.77
a person wipes a surface with the left hand	-3.82
a person wipes a surface with his left hand	-3.88

(i) Wiping (left hand)

Figure 6. Exemplary natural language descriptions as generated by the proposed model for nine different human whole-body motions. The description hypotheses are sorted in descending order by their log probability under the model.

0 and 1, with 1 indicating a perfect match, i.e. the hypothesis is word for word identical to one of the references.

We compute the BLEU score on a corpus-level, in contrast to sentence-level as in other works on this problem. This is an important distinction, since sentence-level computation estimates the n-gram model only on very few reference sentences, leading to bad estimates and therefore unreliable scores. More concretely, we generate five descriptions, that is

Table 2. Corpus-level Bleu scores for the motion-to-language model.

	Bleu scores				
	1st	2nd	3rd	4th	5th
Train	0.387	0.355	0.330	0.329	0.302
Test	0.338	0.283	0.295	0.277	0.250

the hypotheses, for each human motion and sort in descending order by their log probability under the model. We then compute five different Bleu scores that correspond to selecting the 1st, 2nd, ..., 5th best hypothesis. In all cases, we use all available annotations, which have been created by the human annotators, as the ground truth. Bleu scores are computed separately for the training and test split.

Table 2 lists the achieved Bleu scores. A couple of observations are noteworthy. First, the Bleu scores are lower than one would expect from a machine translation system, for example. This is because there is much more ambiguity when generating descriptions of human motion than when translating a text into a different language. In the former case, different levels of details and different styles are also semantically correct (e.g. *"A human walks"* vs. *"Someones takes a couple of steps"*), whereas the latter case is much more constrained. Second, the Bleu scores are clearly correlated with the order of the hypotheses (as defined by their log probabilities under the model) even though the loss that was used to train the model and the Bleu score are completely separate. This means that the log probability is suitable as a measure of quality and, in turn, that the model indeed captures some understanding of what an objectively high-quality (as measured by Bleu), description is.

Third, the model achieves slightly worse, but still comparable performance on the test split. This, again, demonstrates that the model does not overfit on the training data and generalizes to previously unseen motions. Additionally, the second point, that Bleu scores and their ranking as defined by the respective log probabilities of the hypotheses are correlated, still holds; however, the pattern is a bit more noisy.

5.4 *Generating motion from language*

5.4.1 Training: Optimizing the language-to-motion model proved to be more complex, which was presumably due to the much more complex decoder network. In order to successfully optimize the model, we found that gradient clipping and layer normalization play a crucial role. The learning curves for the training and validation splits are depicted in Figure 7.

Although the curve for the validation loss is more noisy than in the motion-to-language case, the model still appears to learn the problem at hand without overfitting on the training data.

5.4.2 Qualitative results: Similar to before, we provide some insight into the type of motion the proposed model generates by visualizing several examples in Figure 8. For each given natural language description, we compute five human whole-body motion hypotheses (thus $W = 5$). Due to space constraints, we only depict the motion with the highest loglikelihood under the model.

Most interestingly, the results also demonstrate that our model is not only capable of generating the correct motion primitive, but also to adjust the generated motion to a desired parametrization, e.g. the model generates motions for waving with the right, left and both hands (Figure 8g, Figure 8h and Figure 8i. We observed similar behavior in other examples such as walking motions with different speed.

For periodic movements, we observed that the model does generate motions with a number of repetitions (e.g. waving seven times) that were not indeed included in the training set, suggesting that it does discover the underlying periodic structure of the motion. However, we were unable to parametrize the number of repetitions using language. We hypothesize that this is because the training data does not contain enough training examples to learn counting.

Another limitation of the proposed model is caused by the fact that we represent the motion using only joint angles. This, in turn, means that the model does

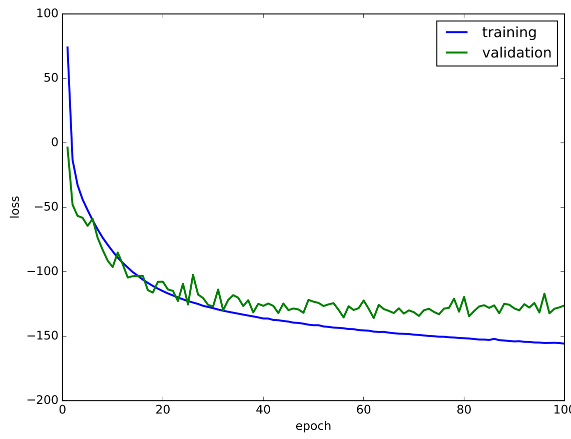
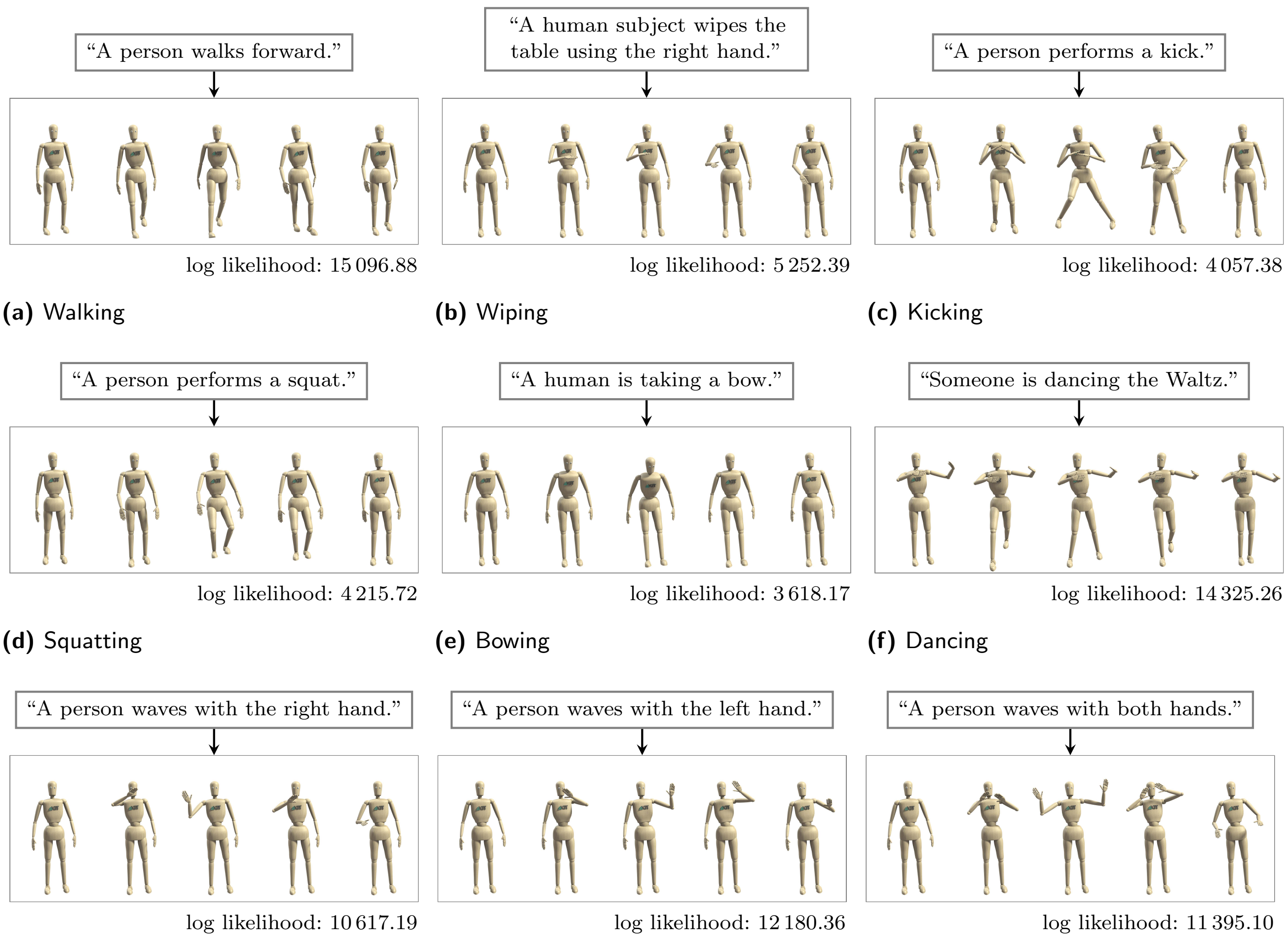


Figure 7. Training and validation loss during training of the language-to-motion model.


"A person walks forward."
log likelihood: 15 096.88
"A human subject wipes the table using the right hand."
log likelihood: 5 252.39
"A person performs a kick."
log likelihood: 4 057.38
"A person performs a squat."
log likelihood: 4 215.72
"A human is taking a bow."
log likelihood: 3 618.17
"Someone is dancing the Waltz."
log likelihood: 14 325.26
"A person waves with the right hand."
log likelihood: 10 617.19
"A person waves with the left hand."
log likelihood: 12 180.36
"A person waves with both hands."
log likelihood: 11 395.10

(a) Walking **(b)** Wiping **(c)** Kicking

(d) Squatting **(e)** Bowing **(f)** Dancing

(g) Waving (right hand) **(h)** Waving (left hand) **(i)** Waving (both hands)

Figure 8. Exemplary human whole-body motions as generated by the proposed model from different natural language descriptions.

not predict the pose of the human subject in space, which can be observed for the bowing, squatting and kicking motions (Figure 8e, Figure 8d and Figure 8c), where the root pose of the human model remains fixed. Furthermore, since we only consider the kinematic aspects of a motion, the generated motions are not necessarily dynamically stable. Similarly, since we do not consider contact information with objects at this point, generated manipulation motions are violating constraints that would be necessary to achieve the desired outcome. This can be observed for the wiping motion (Figure 8b), where the wiping diverges from the (imaginary) table surface over time. Including these dynamic properties and contact information is an important area of future work.

Overall, we clearly demonstrate the capabilities of the model to generate a wide variety of realistic human whole-body motions specified only by a description in natural language. We also visualize the entirety of the generated motions in a supplementary video: `https://youtu.be/2UQWOZtsg-8`.

5.4.3 Quantitative results: Providing quantitative results for the performance of the language-to-motion model is complicated since defining an appropriate metric is non-trivial. The reason for this is that a motion can be performed in a large variety of different styles that can all be semantically correct for the given description but may have very different joint-level characteristics. Computing a metric like the mean squared error between reference and hypotheses is therefore ill-suited to judge how correct the motion hypothesis is *semantically*.

To resolve this problem, we exploit the fact that we already can compute a semantic description of a given motion using our previously evaluated motion-to-language model. Since we have already evaluated

Table 3. Corpus-level Bleu scores for the language-to-motion model.

	Bleu scores				
	1st	2nd	3rd	4th	5th
Train	0.256	0.277	0.289	0.286	0.278
Test	0.249	0.242	0.288	0.249	0.240

this model separately, we can use its performance as a baseline to judge the quality of the other direction, namely language-to-motion.

More concretely, we essentially chain the two models. First, we use the language-to-motion model, which we want to evaluate here, to compute a motion given a description in natural language. Next, we use the previously trained and evaluated motion-to-language model to transform this generated motion back into a description in natural language. Finally, we can compute the Bleu score as described in Section 5.3.3 to quantitatively measure the performance of the language-to-motion model. The results of this approach are given in Table 3.

To make it easier to compare the performance, we propose to measure the language-to-motion model relative to the performance of the motion-to-language model. More formally, we relate the Bleu score of the language-to-motion model (to be evaluated, see Table 3) to the highest Bleu score of the motion-to-language model (the baseline, which is 0.387) by dividing the two. We thus measure the percentage of performance that is retained after transforming the natural language into a motion hypothesis and then transforming this generated motion back into a description. Table 4 lists this relative performance for the language-to-motion model.

A couple of observations. First, the results clearly demonstrate that our model is capable of generating the correct human motion beyond the few examples presented in the qualitative analysis described before. Compared to the baseline, the model achieves a performance of $71.6\% \pm 3.0$ for both the training and $65.5\% \pm 4.5$ for the test split. As expected, the Bleu scores for the training split are slightly higher than for the test split, but still comparable. This suggests minimal overfit and generalization capabilities to previously unseen examples. Second, the ranking of hypotheses as defined by their loglikelihood under the model seems to be less correlated with the performance of the model under the proposed evaluation metric. Currently, it is difficult to identify the prime origin and cause for this since this could also be caused by the error of the motion-to-language model.

Table 4. Relative performance of the language-to-motion model compared to the baseline performance of the motion-to-language model.

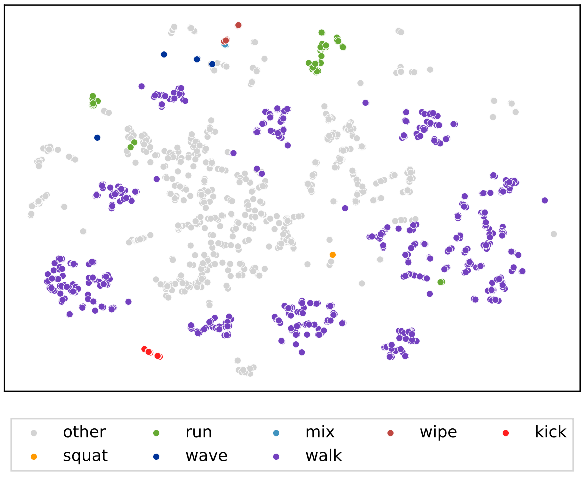
	Relative Performance				
	1st	2nd	3rd	4th	5th
Train	66.1%	71.6%	74.7%	73.9%	71.8%
Test	64.3%	62.5%	74.4%	64.3%	62.0%

5.5 *Understanding the model*

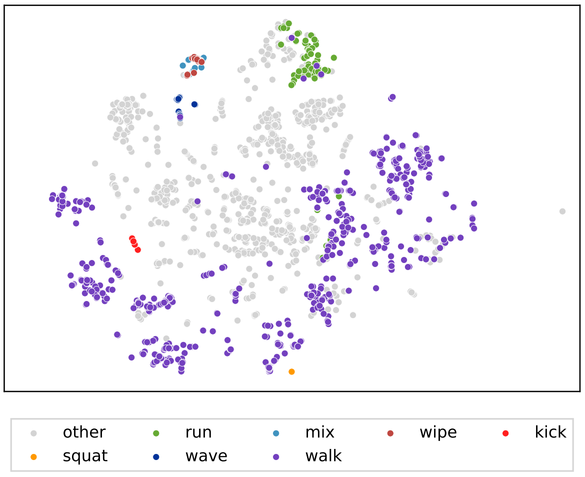
Our final experiment is concerned with providing insight into the latent representations that the proposed model uses internally. In order to do so, we analyze the context vectors produced by both the motion-to-language model and language-to-motion model. Since they are extremely high-dimensional ($\boldsymbol{c} \in \mathbb{R}^{128}$), we use *t-distributed Stochastic Neighbor Embedding* (t-SNE) (Maaten and Hinton 2008) to project all context vectors into a 2-dimensional representation. t-SNE is particular well-suited for this task since it is known to maintain the structure of the original high-dimensional vector space in the low-dimensional projection.

We then color this low-dimensional projection of the context vector space according to the type of motion performed. Due to the large variety of different motion, we select to color motions of type walking, running, waving, wiping, mixing and squatting and use a combined color for all other motions. The labels are obtained from the KIT Whole-Body Human Motion Database, which provides multiple labels for each motion record. Figure 9 depicts this visualization for both models, motion-to-language and language-to-motion.

Both visualization exhibit a clear structure and contain clusters of motions of the same type. Interestingly, the visualization of the motion-to-language model (Figure 9a) has denser clusters with less variance and cleaner separation between clusters than the visualization of the language-to-motion model (Figure 9b). This makes intuitive sense since describing a motion in natural language has far more ambiguities compared to observing the motion directly. Comparing running and walking motions, this observation is especially apparent: In the motion-to-language case, the two types of motion are nicely separated wheres in the language-to-motion case the



(a) Motion-to-language context vectors.



(b) Language-to-motion context vectors.

Figure 9. Visualization of the context vectors, colored by their respective motion type.

two types often fall into the same cluster (top-most green cluster in Figure 9b).

Another interesting observation is that motions of type wiping and mixing (in the sense of mixing something in a bowl) are used interchangeably in both directions. Since we do not include object information (to neither the model nor the human annotators that created the dataset), mixing and wiping appear to be the same.

We also investigate the structure *within* a given type of motion, in this case walking. For this purpose, we use the same projection as in the previous motion-to-language visualization. However, this time we color motions by their respective direction (left, right, forward, backward) and only those that are walking motions. The resulting 2-dimensional t-SNE projection is depicted in Figure 10, which shows that walking motions that have the same direction are grouped in the same cluster. However, the separation is often less clear than for motions of completely different type.

Overall, the analysis of the context vector space clearly suggests that our proposed models extract semantic meaning from the given input (for both cases, motion or language) and encode it into the context vector. The decoder part of the models can then generate the correct sequence using this semantic representation.

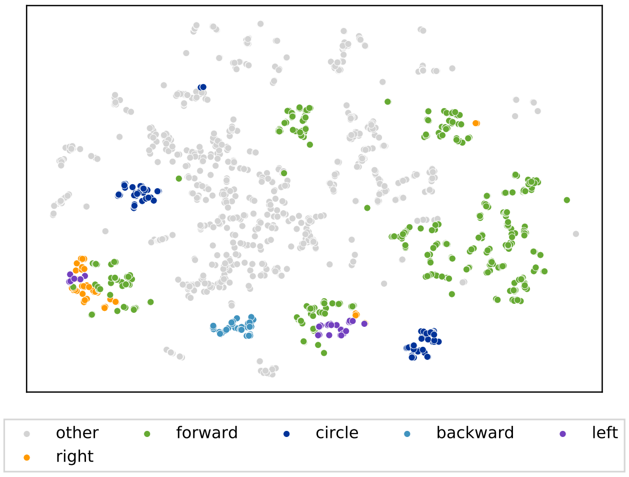


Figure 10. Visualization of contexts vectors associated with walking, colored by their respective direction.

6 Conclusion

In this paper, we proposed the use of deep recurrent neural networks in a sequence-to-sequence setting to learn a bidirectional mapping between human whole-body motion and descriptions in natural language. We presented models that can be used to model each direction of the bidirectional mapping individually. An important property of our proposed models is that they are probabilistic, allowing us to produce different candidate hypotheses and ranking them accordingly. Additionally, our system makes minimal assumptions about both the natural language descriptions and human whole-body motions, requiring minimal preprocessing and no explicit motion segmentation into motion or action primitives or clusters thereof *a-priori*. Furthermore, each model makes use of a distributed representation, which is shared for all types of motions.

In our experiments, we clearly demonstrated the capabilities of our proposed system to generate rich and detailed descriptions of a large variety of different human whole-body motions. Conversely, we showed that our model is capable of generating a similarly large variety of realistic human whole-body motion given a description thereof in natural language. We quantified and reported the performance of our system using the BLEU score, which is well-known and frequently used in the context of machine translation. We also presented results that indicate that each model successfully learns distributed and semantically meaningful latent representations of the given input to produce the desired output.

A limitation of our proposed system is that the input sequence needs to be encoded into a single vector (the context vector $\boldsymbol{c}$). This becomes especially problematic as the sequence length increases. To overcome this problem, *attention mechanisms* have been proposed in the literature(Bahdanau et al. 2014). Integrating such mechanisms in the future is likely going to improve the performance of our system. Similarly, *hierarchical RNNs* (Jain et al. 2015; Du et al. 2015) have been proposed and were used successfully to model human motion. Integrating these ideas into our system would likewise be an interesting experiment.

Increasing the size of the KIT Motion-Language Dataset (Plappert et al. 2016) is an important area of future work as well. More data would allow us to use more complex models and reduces the risk of overfitting. Additionally, including more complex motions in which a subject performs a sequence of distinct steps (e.g. when cooking) would allow for interesting experiments to further test the generalization capabilities of our proposed system by permutating the order of the steps.

Lastly, representing human whole-body motion using only the joint values of the kinematic model is insufficient. In future work, we therefore intend to incorporate dynamic properties of the motion as well as contact information with the environment and objects that are involved in the execution of the motion. Such multi-contact motions have recently been studied by Mandery et al. (2015a).

Acknowledgements

The research leading to these results has received funding from the H^2T at KIT, the German Research Foundation (Deutsche Forschungsgemeinschaft: DFG) under the Priority Program on Autonomous Learning (SPP 1527), and the European Union's Horizon 2020 Research and Innovation Programme and Seventh Framework Programme under grant agreements No. 643666 (I-Support) and No. 611832 (WALK-MAN).

References

Arie H, Endo T, Jeong S, Lee M, Sugano S and Tani J (2010) Integrative learning between language and action: A neuro-robotics experiment. In: *Artificial Neural Networks - ICANN 2010, 20th International Conference, Thessaloniki, Greece, September 15-18, 2010, Proceedings, Part II*. pp. 256–265.

Azad P, Asfour T and Dillmann R (2007) Toward an unified representation for imitation of human motion on humanoids. In: *2007 IEEE International Conference on Robotics and Automation, ICRA 2007, 10-14 April 2007, Roma, Italy*. pp. 2558–2563.

Ba LJ, Kiros R and Hinton GE (2016) Layer normalization. *CoRR* abs/1607.06450.

Bahdanau D, Cho K and Bengio Y (2014) Neural machine translation by jointly learning to align and translate. *CoRR* abs/1409.0473.

Bengio Y, Boulanger-Lewandowski N and Pascanu R (2012) Advances in optimizing recurrent networks. *CoRR* abs/1212.0901.

Billard A, Calinon S, Dillmann R and Schaal S (2008) Robot programming by demonstration. In: *Springer Handbook of Robotics*. pp. 1371–1394.

Bütepage J, Black MJ, Kragic D and Kjellström H (2017) Deep representation learning for human motion prediction and classification. *CoRR* abs/1702.07486.

Calinon S, Guenter F and Billard A (2007) On learning, representing, and generalizing a task in a humanoid robot. *IEEE Trans. Systems, Man, and Cybernetics, Part B* 37(2): 286–298.

Cho K, van Merrienboer B, Bahdanau D and Bengio Y (2014) On the properties of neural machine translation: Encoder-decoder approaches. *CoRR* abs/1409.1259.

Chung J, Gülçehre Ç, Cho K and Bengio Y (2014) Empirical evaluation of gated recurrent neural networks on sequence modeling. *CoRR* abs/1412.3555.

Cooijmans T, Ballas N, Laurent C and Courville AC (2016) Recurrent batch normalization. *CoRR* abs/1603.09025.

Dillmann R, Rogalla O, Ehrenmann M, Zollner R and Bordegoni M (2000) Learning robot behaviour and skills based on human demonstration and advice: the machine learning paradigm. In: *ROBOTICS RESEARCH-INTERNATIONAL SYMPOSIUM-*, volume 9. pp. 229–238.

Donahue J, Hendricks LA, Guadarrama S, Rohrbach M, Venugopalan S, Darrell T and Saenko K (2015) Long-term recurrent convolutional networks for visual recognition and description. In: *IEEE Conference* *on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*. pp. 2625–2634.

Dozat T (2015) Incorporating nesterov momentum into adam. Technical report, Stanford University, Tech. Rep., 2015.

Du Y, Wang W and Wang L (2015) Hierarchical recurrent neural network for skeleton based action recognition. In: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*. pp. 1110–1118.

Field M, Pan Z, Stirling D and Naghdy F (2011) Human motion capture sensors and analysis in robotics 38(2): 163–171.

Fragkiadaki K, Levine S and Malik J (2015) Recurrent network models for kinematic tracking. *CoRR* abs/1508.00271.

Gal Y (2015) A theoretically grounded application of dropout in recurrent neural networks. *arXiv preprint arXiv:1512.05287* .

Gers FA, Schmidhuber J and Cummins F (2000) Learning to forget: Continual prediction with lstm. *Neural computation* 12(10): 2451–2471.

Glorot X, Bordes A and Bengio Y (2011) Domain adaptation for large-scale sentiment classification: A deep learning approach. In: *Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, Washington, USA, June 28 - July 2, 2011*. pp. 513–520.

Goodfellow I, Bengio Y and Courville A (2016) Deep learning. Book in preparation for MIT Press.

Graves A (2013) Generating sequences with recurrent neural networks. *CoRR* abs/1308.0850.

Graves A, Mohamed A and Hinton GE (2013) Speech recognition with deep recurrent neural networks. In: *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2013, Vancouver, BC, Canada, May 26-31, 2013*. pp. 6645–6649.

Graves A and Schmidhuber J (2005) Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks* 18(5-6): 602–610.

Gu S, Holly E, Lillicrap TP and Levine S (2016) Deep reinforcement learning for robotic manipulation. *CoRR* abs/1610.00633.

Hassani K and Lee W (2016) Visualizing natural language descriptions: A survey. *ACM Comput. Surv.* 49(1): 17.

He K, Zhang X, Ren S and Sun J (2015) Deep residual learning for image recognition. *CoRR* abs/1512.03385.

Herzog D, Ude A and Krüger V (2008) Motion imitation and recognition using parametric hidden markov models. In: *8th IEEE-RAS International Conference on Humanoid Robots, Humanoids 2008, Daejeon, South Korea, December 1-3, 2008*. pp. 339–346.

Hochreiter S and Schmidhuber J (1997) Long short-term memory. *Neural computation* 9(8): 1735–1780.

Huang X, Acero A and Hon HW (2001) *Spoken language processing: A guide to theory, algorithm, and system development*. 1st ed. edition. Englewood Cliffs, NJ, USA: Prentice Hall.

Ioffe S and Szegedy C (2015) Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*. pp. 448–456.

Jain A, Zamir AR, Savarese S and Saxena A (2015) Structural-rnn: Deep learning on spatio-temporal graphs. *CoRR* abs/1511.05298.

Karpathy A and Li F (2015) Deep visual-semantic alignments for generating image descriptions. In: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*. pp. 3128–3137.

Kingma DP and Ba J (2014) Adam: A method for stochastic optimization. *CoRR* abs/1412.6980.

Krizhevsky A, Sutskever I and Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In: *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States*. pp. 1106–1114.

Kulic D, Takano W and Nakamura Y (2008) Incremental learning, clustering and hierarchy formation of whole body motion patterns using adaptive hidden markov chains. *I. J. Robotics Res.* 27(7): 761–784.

Kuniyoshi Y, Inaba M and Inoue H (1994) Learning by watching: Extracting reusable task knowledge from visual observation of human performance. *IEEE Transactions on Robotics and Automation* 10: 799–822.

LeCun Y, Bengio Y and Hinton G (2015) Deep learning. *Nature* 521(7553): 436–444.

Levine S, Finn C, Darrell T and Abbeel P (2015) End-to-end training of deep visuomotor policies. *CoRR* abs/1504.00702.

Levine S, Pastor P, Krizhevsky A and Quillen D (2016) Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. *CoRR* abs/1603.02199.

Maaten Lvd and Hinton G (2008) Visualizing high-dimensional data using t-SNE. *Journal of Machine Learning Research* 9(Nov): 2579–2605.

Mandery C, Sol JB, Jöchner M and Asfour T (2015a) Analyzing whole-body pose transitions in multi-contact motions. In: *15th IEEE-RAS International Conference on Humanoid Robots, Humanoids 2015, Seoul, South Korea, November 3-5, 2015*. pp. 1020–1027.

Mandery C, Terlemez Ö, Do M, Vahrenkamp N and Asfour T (2015b) The KIT whole-body human motion database. In: *International Conference on Advanced Robotics, ICAR 2015, Istanbul, Turkey, July 27-31, 2015*. pp. 329–336.

Mandery C, Terlemez Ö, Do M, Vahrenkamp N and Asfour T (2016) Unifying representations and large-scale whole-body motion databases for studying human motion. *IEEE Trans. Robotics* 32(4): 796–809.

Medina JR, Shelley M, Lee D, Takano W and Hirche S (2012) Towards interactive physical robotic assistance: Parameterizing motion primitives through natural language. In: *The 21st IEEE International Symposium on Robot and Human Interactive Communication, IEEE RO-MAN 2012, Paris, France, September 9-13, 2012*. pp. 1097–1102.

Mikolov T, Sutskever I, Chen K, Corrado GS and Dean J (2013) Distributed representations of words and phrases and their compositionality. In: *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States*. pp. 3111–3119.

Mordatch I, Lowrey K, Andrew G, Popovic Z and Todorov E (2015) Interactive control of diverse complex characters with neural networks. In: *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*. pp. 3132–3140.

Ogata T, Matsumoto S, Tani J, Komatani K and Okuno HG (2007a) Human-robot cooperation using quasi-symbols generated by RNNPB model. In: *2007 IEEE International Conference on Robotics and Automation, ICRA 2007, 10-14 April 2007, Roma, Italy*. pp. 2156–2161.

Ogata T, Murase M, Tani J, Komatani K and Okuno HG (2007b) Two-way translation of compound sentences and arm motions by recurrent neural networks. In: *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems, October 29 - November 2, 2007, Sheraton Hotel and Marina, San Diego, California, USA*. pp. 1858–1863.

Ogata T and Okuno HG (2013) Integration of behaviors and languages with a hierarchal structure self-organized in a neuro-dynamical model. In: *2013 IEEE Workshop on Robotic Intelligence In Informationally Structured Space, RiiSS 2013, Singapore, April 16-19, 2013*. pp. 89–95.

Papineni K, Roukos S, Ward T and Zhu W (2002) Bleu: a method for automatic evaluation of machine translation. In: *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, July 6-12, 2002, Philadelphia, PA, USA*. pp. 311–318.

Plappert M, Mandery C and Asfour T (2016) The KIT motion-language dataset. *CoRR* abs/1607.03827.

Schaal S (2006) *Dynamic Movement Primitives -A Framework for Motor Control in Humans and Humanoid Robotics*. Tokyo: Springer Tokyo, pp. 261–280.

Srivastava N, Hinton GE, Krizhevsky A, Sutskever I and Salakhutdinov R (2014) Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research* 15(1): 1929–1958.

Sugita Y and Tani J (2005) Learning semantic combinatoriality from the interaction between linguistic and behavioral processes. *Adaptive Behaviour* 13(1): 33–52.

Sutskever I, Vinyals O and Le QV (2014) Sequence to sequence learning with neural networks. In: *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*. pp. 3104–3112.

Takano W, Kulic D and Nakamura Y (2007) Interactive topology formation of linguistic space and motion space. In: *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems, October 29 - November 2, 2007, Sheraton Hotel and Marina, San Diego, California, USA*. pp. 1416–1422.

Takano W, Kusajima I and Nakamura Y (2016) Generating action descriptions from statistically integrated representations of human motions and sentences. *Neural Networks* 80: 1–8.

Takano W and Nakamura Y (2008) Integrating whole body motion primitives and natural language for humanoid robots. In: *8th IEEE-RAS International Conference on Humanoid Robots, Humanoids 2008, Daejeon, South Korea, December 1-3, 2008*. pp. 708–713.

Takano W and Nakamura Y (2009) Statistically integrated semiotics that enables mutual inference between linguistic and behavioral symbols for humanoid robots. In: *2009 IEEE International Conference on Robotics and Automation, ICRA 2009, Kobe, Japan, May 12-17, 2009*. pp. 646–652.

Takano W and Nakamura Y (2012) Bigram-based natural language model and statistical motion symbol model for scalable language of humanoid robots. In: *IEEE International Conference on Robotics and Automation, ICRA 2012, 14-18 May, 2012, St. Paul, Minnesota, USA*. pp. 1232–1237.

Takano W and Nakamura Y (2015a) Statistical mutual conversion between whole body motion primitives and linguistic sentences for human motions. *I. J. Robotics Res.* 34(10): 1314–1328.

Takano W and Nakamura Y (2015b) Symbolically structured database for human whole body motions based on association between motion symbols and motion words. *Robotics and Autonomous Systems* 66: 75–85.

Takano W, Yamane K, Sugihara T, Yamamoto K and Nakamura Y (2006) Primitive communication based on motion recognition and generation with hierarchical mimesis model. In: *Proceedings of the 2006 IEEE International Conference on Robotics and Automation, ICRA 2006, May 15-19, 2006, Orlando, Florida, USA*. pp. 3602–3609.

Taylor GW and Hinton GE (2009) Factored conditional restricted boltzmann machines for modeling motion style. In: *Proceedings of the 26th Annual International Conference on Machine Learning, ICML 2009, Montreal, Quebec, Canada, June 14-18, 2009*. pp. 1025–1032.

Taylor GW, Hinton GE and Roweis ST (2006) Modeling human motion using binary latent variables. In: *Advances in Neural Information Processing Systems 19, Proceedings of the Twentieth Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 4-7, 2006*. pp. 1345–1352.

Terlemez Ö, Ulbrich S, Mandery C, Do M, Vahrenkamp N and Asfour T (2014) Master motor map (MMM) - framework and toolkit for capturing, representing, and reproducing human motion on humanoid robots. In: *14th IEEE-RAS International Conference on Humanoid Robots, Humanoids 2014, Madrid, Spain, November 18-20, 2014*. pp. 894–901.

Venugopalan S, Xu H, Donahue J, Rohrbach M, Mooney RJ and Saenko K (2015) Translating videos to natural language using deep recurrent neural networks. In: *NAACL HLT 2015, The 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Denver, Colorado, USA, May 31 - June 5, 2015*. pp. 1494–1504.

Vinyals O, Toshev A, Bengio S and Erhan D (2015) Show and tell: A neural image caption generator. In: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*. pp. 3156–3164.

Werbos PJ (1990) Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE* 78(10): 1550–1560.

Wu Y, Schuster M, Chen Z, Le QV, Norouzi M, Macherey W, Krikun M, Cao Y, Gao Q, Macherey K, Klingner J, Shah A, Johnson M, Liu X, Kaiser L, Gouws S, Kato Y, Kudo T, Kazawa H, Stevens K, Kurian G, Patil N, Wang W, Young C, Smith J, Riesa J, Rudnick A, Vinyals O, Corrado G, Hughes M and Dean J (2016) Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR* abs/1609.08144.

A Detailed motion-to-language model architecture

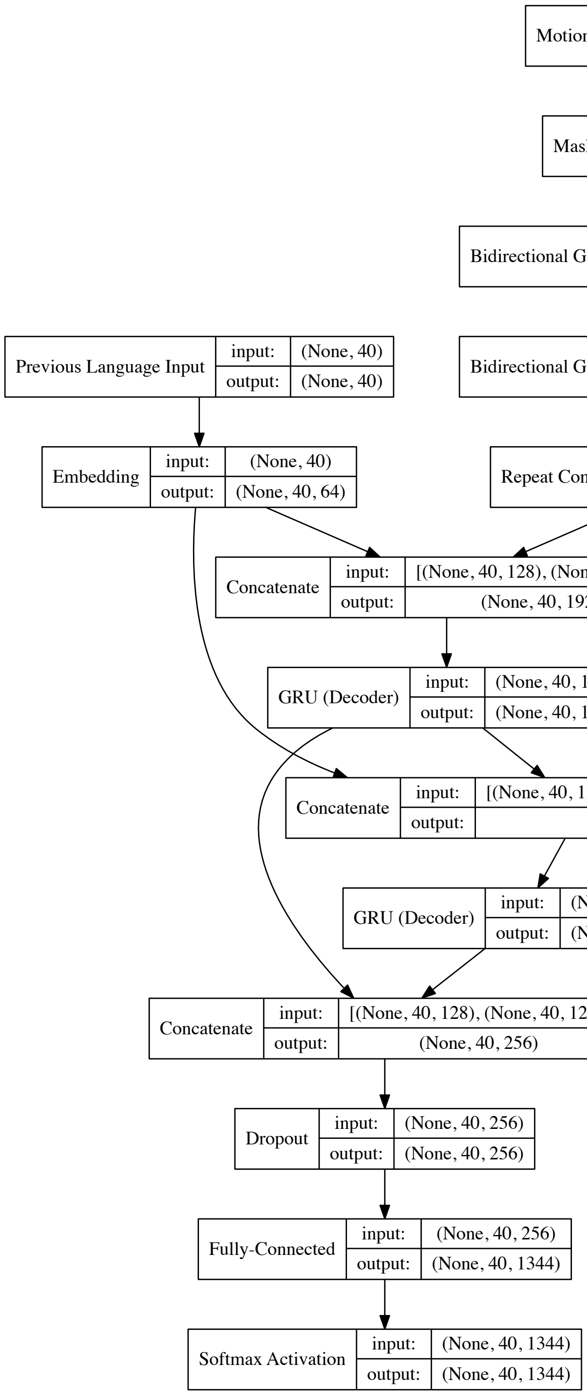

Motion Input | input: (None, 300, 45) | output: (None, 300, 45)
Masking | input: (None, 300, 45) | output: (None, 300, 45)
Bidirectional GRU (Encoder) | input: (None, 300, 45) | output: (None, 300, 128)
Bidirectional GRU (Encoder) | input: (None, 300, 128) | output: (None, 128)
Previous Language Input | input: (None, 40) | output: (None, 40)
Embedding | input: (None, 40) | output: (None, 40, 64)
Repeat Context Vector | input: (None, 128) | output: (None, 40, 128)
Concatenate | input: [(None, 40, 128), (None, 40, 64)] | output: (None, 40, 192)
GRU (Decoder) | input: (None, 40, 192) | output: (None, 40, 128)
Concatenate | input: [(None, 40, 128), (None, 40, 128), (None, 40, 64)] | output: (None, 40, 320)
GRU (Decoder) | input: (None, 40, 320) | output: (None, 40, 128)
Concatenate | input: [(None, 40, 128), (None, 40, 128)] | output: (None, 40, 256)
Dropout | input: (None, 40, 256) | output: (None, 40, 256)
Fully-Connected | input: (None, 40, 256) | output: (None, 40, 1344)
Softmax Activation | input: (None, 40, 1344) | output: (None, 40, 1344)

Figure 11. The detailed architecture of the motion-to-language model. Each node specifies the shape of its input tensor and output tensor. The first dimension is the batch size (128 in this case) followed by the time dimension for 3-dimensional tensors (padded to 300 timesteps at 10 Hz for the motion case and padded to 41 words for the language case). The last dimension is the feature dimension, of which the meaning depends on the specific modality and layer.

B Detailed language-to-motion model architecture

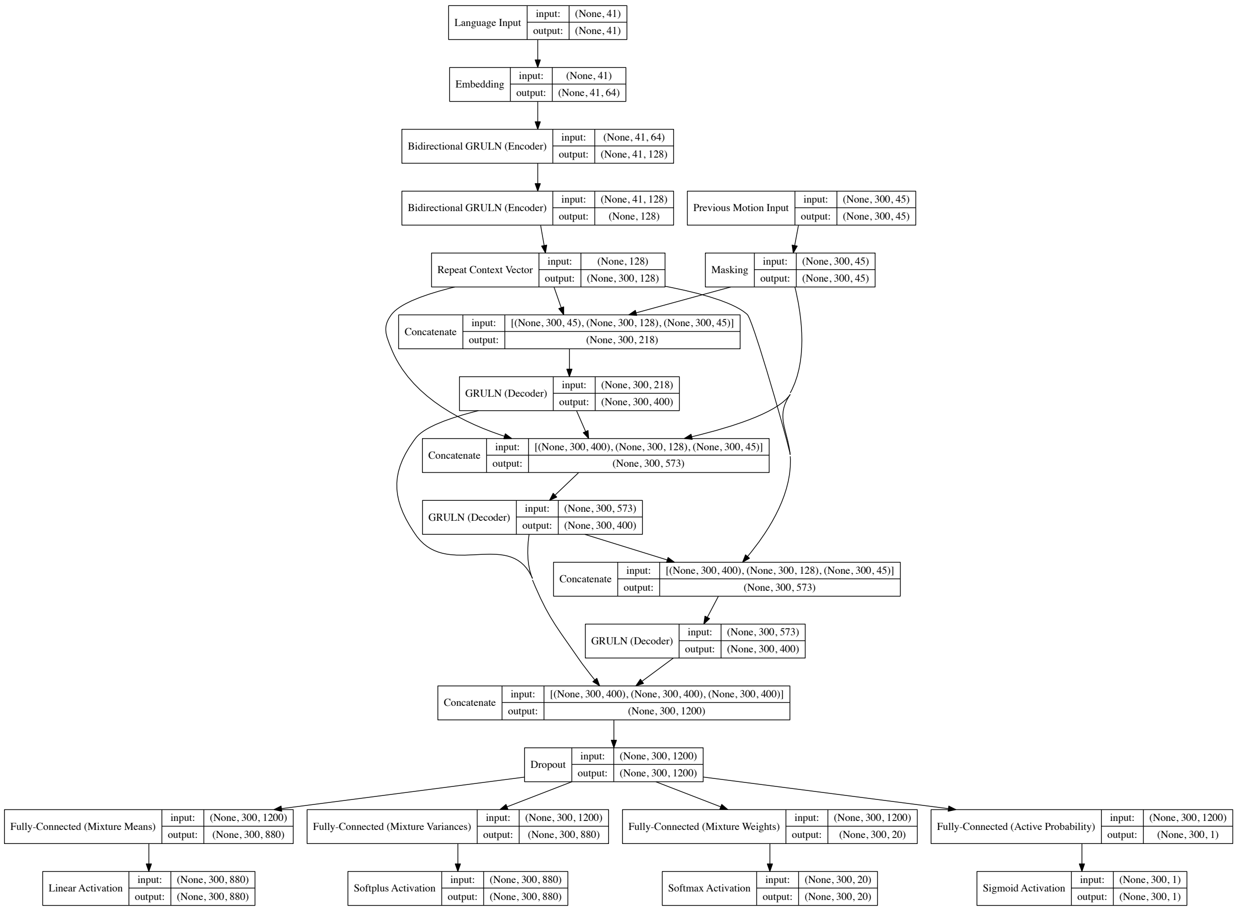

Language Input input: (None, 41) output: (None, 41)
Embedding input: (None, 41) output: (None, 41, 64)
Bidirectional GRULN (Encoder) input: (None, 41, 64) output: (None, 41, 128)
Bidirectional GRULN (Encoder) input: (None, 41, 128) output: (None, 128)
Previous Motion Input input: (None, 300, 45) output: (None, 300, 45)
Repeat Context Vector input: (None, 128) output: (None, 300, 128)
Masking input: (None, 300, 45) output: (None, 300, 45)
Concatenate input: [(None, 300, 45), (None, 300, 128), (None, 300, 45)] output: (None, 300, 218)
GRULN (Decoder) input: (None, 300, 218) output: (None, 300, 400)
Concatenate input: [(None, 300, 400), (None, 300, 128), (None, 300, 45)] output: (None, 300, 573)
GRULN (Decoder) input: (None, 300, 573) output: (None, 300, 400)
Concatenate input: [(None, 300, 400), (None, 300, 128), (None, 300, 45)] output: (None, 300, 573)
GRULN (Decoder) input: (None, 300, 573) output: (None, 300, 400)
Concatenate input: [(None, 300, 400), (None, 300, 400), (None, 300, 400)] output: (None, 300, 1200)
Dropout input: (None, 300, 1200) output: (None, 300, 1200)
Fully-Connected (Mixture Means) input: (None, 300, 1200) output: (None, 300, 880)
Fully-Connected (Mixture Variances) input: (None, 300, 1200) output: (None, 300, 880)
Fully-Connected (Mixture Weights) input: (None, 300, 1200) output: (None, 300, 20)
Fully-Connected (Active Probability) input: (None, 300, 1200) output: (None, 300, 1)
Linear Activation input: (None, 300, 880) output: (None, 300, 880)
Softplus Activation input: (None, 300, 880) output: (None, 300, 880)
Softmax Activation input: (None, 300, 20) output: (None, 300, 20)
Sigmoid Activation input: (None, 300, 1) output: (None, 300, 1)

Figure 12. The detailed architecture of the language-to-motion model. Each node specifies the shape of its input tensor and output tensor. The first dimension is the batch size (128 in this case) followed by the time dimension for 3-dimensional tensors (padded to 300 timesteps at 10 Hz for the motion case and padded to 41 words for the language case). The last dimension is the feature dimension, of which the meaning depends on the specific modality and layer.