Paired Recurrent Autoencoders for Bidirectional Translation Between Robot Actions and Linguistic Descriptions

Tatsuro Yamada, Hiroyuki Matsunaga, and Tetsuya Ogata

***Abstract*—We propose a novel deep learning framework for bidirectional translation between robot actions and their linguistic descriptions. Our model consists of two recurrent autoencoders (RAEs). One RAE learns to encode action sequences as fixed-dimensional vectors in a way that allows the sequences to be reproduced from the vectors by its decoder. The other RAE learns to encode descriptions in a similar way. In the learning process, in addition to reproduction losses, we create another loss function whereby the representations of an action and its corresponding description approach each other in the latent vector space. Across the shared representation, the trained model can produce a linguistic description given a robot action. The model is also able to generate an appropriate action by receiving a linguistic instruction, conditioned on the current visual input. Visualization of the latent representations shows that the robot actions are embedded in a semantically compositional way in the vector space by being learned jointly with descriptions.**

***Index Terms*—Deep learning in robotics and automation, AI-based methods, neurorobotics.**

I. Introduction

ROBOTS in everyday environments are required to work while communicating with people in a linguistic way. Unlike most situations in industrial factories, our living environment is highly changeable; the current situation almost always differs from the previous ones. Therefore, robots have to flexibly ground linguistic descriptions in their own actions conditioned on the current situation [1]. One linguistic skill required of robots is obviously the ability to generate actions that are instructed by people in a linguistic way. Meanwhile, a robot's paired skill is to describe the current situation, event, or action with a linguistic phrase or a sentence. In particular, the ability of a robot to describe its own actions is essential for it to communicate with people effectively; this ability makes it easier to interpret the produced actions. Just as people can translate linguistic descriptions into actions and vice versa, so should robots have equivalent bidirectional linguistic skills.

This study proposes a novel deep recurrent neural network (RNN) architecture that can learn (i) to produce linguistic descriptions from robot actions and conversely (ii) to generate robot actions from linguistic commands conditioned on the current visual input. The key idea is making vector representations that encode robot actions and their descriptions by paired recurrent autoencoders (RAEs). One RAE encodes action sequences as fixed-dimensional vectors in a way that allows the sequences to be reproduced from the vectors by its decoder. The other RAE encodes descriptions in a similar way. We propose to train the paired RAEs with an additional loss function that forcibly binds the representations of actions and their paired descriptions. Thanks to this additional loss, the representations of an action and its corresponding description are embedded in an area close to each other in the shared space. Across the shared representation, the trained model can produce a linguistic description from a given robot action. The model is also able to generate an appropriate action upon receiving a linguistic instruction. In other words, through this shared space, these two modalities are bidirectionally translatable.

This paper is organized as follows. In Section II, we make a short survey of related work. In Section III, we describe our proposed neural architecture and the learning algorithm. In Section IV, we present the results of a robot learning experiment to evaluate our proposed method. We also visualize the acquired latent representations of the actions and their descriptions. In Section V, we perform another experiment with a larger dataset of annotated human body motions to evaluate the scalability of the proposed method. We discuss the results and our method in Section VI and finally conclude our study in Section VII.

II. Related Work

A. Encoder–Decoder Model

Many studies have endeavored to train machine learning models to convert linguistic instructions into robot or simulated agent actions [2]–[10], to convert actions into descriptions [11], or to convert both bidirectionally [12]–[14]. In recent years, the deep RNN model has achieved impressive performance thanks

Manuscript received February 23, 2018; accepted June 17, 2018. Date of publication July 4, 2018; date of current version August 2, 2018. This paper was recommended for publication by Associate Editor E. E. Aksoy and Editor T. Asfour upon evaluation of the reviewers' comments. This work was supported in part by a Japan Society for the Promotion of Science (JSPS) Grant-in-Aid for JSPS Research Fellow (17J10580), in part by the JST CREST under Grant JPMJCR15E3, and in part by the Program for Leading Graduate Schools, "Graduate Program for Embodiment Informatics" of the Ministry of Education, Culture, Sports, Science and Technology. T. Yamada is a Research Fellow of the JSPS. *(Corresponding author: Tetsuya Ogata.)*

T. Yamada is with the Department of Intermedia Art and Science, Waseda University, Tokyo 169-8555, Japan (e-mail: yamadat@idr.ias.sci.waseda.ac.jp).

H. Matsunaga and T. Ogata are with the Department of Intermedia Art and Science, Waseda University, Tokyo 169-8555, Japan (e-mail: matsuna4-10@akane.waseda.jp; ogata@waseda.jp).

Digital Object Identifier 10.1109/LRA.2018.2852838

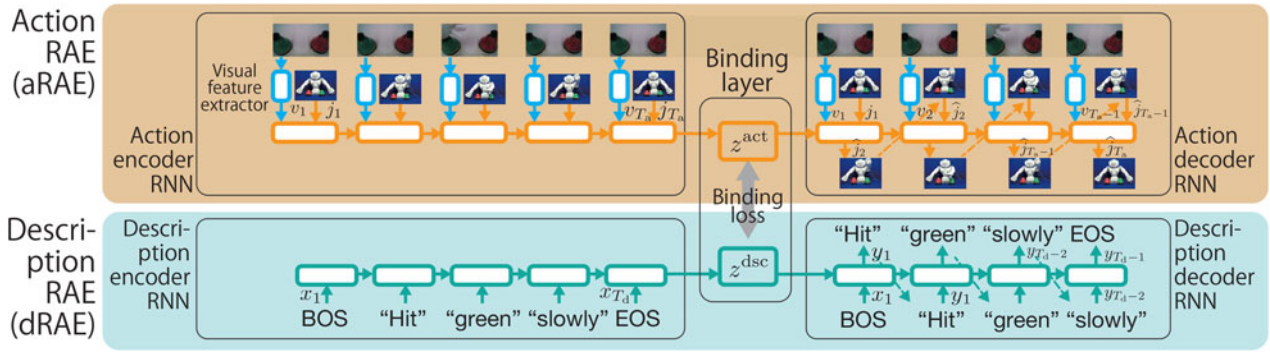

Fig. 1. Model overview. The model consists of paired RAEs, one for descriptions (lower part) and the other for robot actions (upper part). Each learns to minimize the reconstruction error of the original sequence. In this process, each RAE acquires the ability to encode sequences as fixed-dimensional vectors (z^{dsc} or z^{act}) in a way that allows the original sequences to be reproduced by its decoder. In addition to the reconstruction errors, we create an additional loss function that forces the representation of an action and its corresponding description to be close to each other. By these two types of loss (i.e., reconstruction error and representation distance) converging sufficiently, it is expected that bidirectional translation between robot actions and their descriptions would be achieved.

to it being highly capable of dealing with long-term dependency, which is required to process both linguistic sequences and robot actions. One common architecture is the recurrent encoder–decoder structure, also known as the sequence-to-sequence model [15]. In this structure, the encoder RNN first embeds a sequence (e.g., a linguistic instruction) into a fixed-dimensional representation, whereupon the decoder RNN produces a corresponding sequence (e.g., a robot action) by decoding the latent representation. This architecture is very general and thus is used not only for converting between language and agent control [11], [16] but also for such tasks as translating from one language to another [15], [17], translating from human body gestures to robot actions [18], predicting future images from previous ones [19], and speech recognition [20]. Some recent studies have applied a learning framework of generative adversarial networks [21] to the encoder–decoder architecture in order to generate more diverse and realistic samples [4].

From another perspective, the recurrent encoder–decoder architecture can be used as a type of autoencoder for sequential or temporal data [22], [23]. When this is done, the encoder–decoder architecture is called an RAE. The loss function is defined as the distance between the original input sequence and the model's output sequence. In other words, the model is trained so that its decoder can reconstruct an original input sequence from the fixed-dimensional representation of that sequence encoded by its encoder. In this training process, important features or patterns of sequential data can be extracted in an unsupervised manner. The present study uses two RAEs, one to make representations of robot actions and the other for descriptions.

B. Representation Sharing

As mentioned above, we are able to extract important features of sequential data as fixed-dimensional vectors by training (deep) neural networks. As a way to exploit such representations, there have been some studies on cross-modal retrieval in which the paired representations of different modalities (text, sound, video) are bound with each other by creating an additional loss function to make the representations of corresponding audio, visual, and textual information close to each other [24]–[26]. Through such shared representations, each modality can be retrieved from the others. However, the previous research dealt with retrieval only and thus did not generate novel tokens from the representations. Our proposed model can produce sequences of these modalities directly from the representations because in the learning process the shared representations are structured for decoding by the RAE decoders. Through the representations, the model can translate bidirectionally between robot actions and their descriptions.

III. Proposed Method

A. Model Overview

This study proposes a neural network architecture and learning algorithm that learns from a paired dataset of robot action sequences and their linguistic descriptions to enable a robot both to (i) produce actions in response to linguistic instructions and, in contrast, to (ii) produce linguistic descriptions given its own actions. In this study, we define robot actions as sequences of joint angle values, and we suppose that the appropriate choice of translation depends on the situation, which is given as visual information (explained in detail in Section III-C).

As shown in Fig. 1, our model consists of paired RAEs [22], one for descriptions (i.e., word sequences) and the other for robot action sequences. Each RAE learns to minimize the reconstruction error of the original sequences. In this process, each RAE acquires the ability to encode sequences as fixed-dimensional vectors (z^{dsc} or z^{act}) in a way that allows the original sequences to be reproduced by its decoder.

Here, the characteristics of action sequences and those of their descriptions may differ, something that Saussure [27] described as "arbitrariness" of language. Therefore, the shapes of their embedding spaces would also differ. To bind them, we create an additional loss function that forces the representation of an action and that of its corresponding description to be close to each other in the vector space.

By these two types of loss (i.e., reconstruction error and representation distance) converging sufficiently, it is expected that the two aforementioned capabilities would be achieved. Produc-

ing actions in response to linguistic instructions is realized by using the encoder of the description RAE (dRAE) to encode a description and using the decoder of the action RAE (aRAE) to expand the representation. By contrast, producing linguistic descriptions given actions is realized by having the encoder of the aRAE encode an action sequence and having the decoder of the dRAE expand the representation.

B. RAE for Descriptions

To encode descriptions, we use the dRAE (the lower part of Fig. 1). The dRAE consists of an encoder RNN and a decoder RNN. The encoder RNN embeds a description of length $T_{\rm d}$, $(x_1, x_2, ..., x_{T_{\rm d}})$ into a fixed-dimensional vector $z^{\rm dsc}$ as follows:

$$h_t^{\rm enc} = {\rm EncCell}(x_t, h_{t-1}^{\rm enc}) \qquad (1 \le t \le T_{\rm d}), \tag{1}$$

$$z^{\rm dsc} = W^{\rm enc} h_{T_{\rm d}} + b^{\rm enc}, \tag{2}$$

where the function of EncCell represents any type of trainable recurrent cell (e.g., a gated recurrent unit [GRU] [28] or long short-term memory [LSTM] [29]) and $h_t^{\rm enc}$ is the encoder cell state at time step t. $W^{\rm enc}$ and $b^{\rm enc}$ are a trainable weight matrix and a bias vector, respectively, to project the final state $h_{T_{\rm d}}$ of the encoder onto the shared space. We set h_0 as a zero vector.

After encoding, the decoder RNN generates a sequence by recursively expanding the vector representation $z^{\rm dsc}$ as follows:

$$h_0^{\rm dec} = W^{\rm dec} z^{\rm dsc} + b^{\rm dec}, \tag{3}$$

$$h_t^{\rm dec} = {\rm DecCell}(y_{t-1}, h_{t-1}^{\rm dec}) \qquad (1 \le t \le T_{\rm d} - 1), \tag{4}$$

$$y_t = f(W^{\rm out} h_t^{\rm dec} + b^{\rm out}) \qquad (1 \le t \le T_{\rm d} - 1), \tag{5}$$

where the function DecCell represents any type of trainable recurrent cell in a similar way to EncCell, and $h_t^{\rm dec}$ is the decoder cell state at time step t. $W^{\rm dec}$ and $b^{\rm dec}$ are a trainable weight matrix and a bias vector, respectively, to project the representation $z^{\rm dsc}$ onto the decoder initial cell state. Likewise, $W^{\rm out}$ and $b^{\rm out}$ are a trainable weight matrix and a bias vector, respectively, to project the decoder cell state onto the vocabulary size output. f is an activation function. In this study, we represent each word as a one-hot vector that has the value 1 at the element corresponding to the word and 0 at the other elements. We therefore choose the softmax function for f. We give a vector representing the <beginning of sequence (BOS)> symbol instead of y_0.

The target function of training is for the decoder to generate an original description received by the encoder, in other words, minimization of the cross entropy between the input and output:

$$L_{\rm dsc} = \frac{1}{T_{\rm d} - 1} \sum_{t=1}^{T_{\rm d}-1} \left(- \sum_{w}^{W} x_{t+1}(w) \log y_t(w) \right). \tag{6}$$

Here, W is the vocabulary size. We optimize all the trainable parameters by the gradient descent method. The derivative of $L_{\rm dsc}$ with respect to each parameter can be calculated by the back-propagation through time algorithm [30]. In this unsupervised learning, a vector space that effectively embeds the given description set would be learned.

C. RAE for Robot Actions

To encode robot actions, we use the aRAE (the upper part of Fig. 1). Like the dRAE, the aRAE consists of an encoder RNN and a decoder RNN. An action sequence consists of a series of length $T_{\rm a}$, $(j_1, j_2, ..., j_{T_{\rm a}})$ of robot joint angles and a visual stream $(v_1, v_2, ..., v_{T_{\rm a}})$ accompanying it. Here, to start learning from some advantageous stage, we extract low-dimensional visual features from raw images in advance (shown as a visual feature extractor in Fig. 1). The type of extractor can be chosen depending on the tasks. For example, in Section IV, we use a convolutional neural network. The encoder of the aRAE encodes a sequence $((j_1; v_1), (j_2; v_2), ..., (j_{T_{\rm a}}; v_{T_{\rm a}}))$ that concatenates joint angles and visual features into a fixed-dimensional vector $z^{\rm act}$. The architecture and the behavior of the encoder are almost the same, as follows:

$$h_t^{\rm enc} = {\rm EncCell}(v_t, j_t, h_{t-1}^{\rm enc}) \qquad (1 \le t \le T_{\rm a}), \tag{7}$$

$$z^{\rm act} = W^{\rm enc} h_{T_{\rm a}}^{\rm enc} + b^{\rm enc}. \tag{8}$$

However, the input/output of the decoder is somewhat different:

$$h_0^{\rm dec} = W^{\rm dec} z^{\rm act} + b^{\rm dec}, \tag{9}$$

$$h_t^{\rm dec} = {\rm DecCell}(v_t, \widehat{j}_t, h_{t-1}^{\rm dec}) \qquad (1 \le t \le T_{\rm a} - 1), \tag{10}$$

$$\widehat{j}_{t+1} = f(W^{\rm out} h_t^{\rm dec} + b^{\rm out}) \qquad (1 \le t \le T_{\rm a} - 1). \tag{11}$$

Equations (10) and (11) mean that the decoder generates only $\widehat{j}_{t+1}$ as a prediction of joint angles at the next time step j_{t+1}. At each time step, the visual information is given as an external input, as in teacher forcing. In contrast, joint angles predicted by the decoder itself are given to the joint input nodes at the next time step (i.e., it is free running). At the first step only, we give the initial robot posture j_1 instead of $\widehat{j}_1$. The loss function for the aRAE is the mean squared error between the prediction and the target, namely,

$$L_{\rm act} = \frac{1}{T_{\rm a} - 1} \sum_{t=1}^{T_{\rm a}-1} ||j_{t+1} - \widehat{j}_{t+1}||_2^2. \tag{12}$$

We build the model in this way, which predicts only joint angles, not vision, because we want it to be able to deal with context-dependent actions. In other words, the ambiguities between instructions and robot actions will be resolved by receiving visual information. The following explanation is in the form of a concrete example. Action sequences that actualize the description "push the red ball" can differ from each of the other tokens depending on the position of the red ball. As described in Section III-D, the representations of these various action sequences that are encoded by the encoder of the aRAE are bound with the unique representation of the description "push the red ball" that is encoded by the encoder of the dRAE. As a result, these various action sequences are not represented as ones that differ from each other but are embedded close to each other as sequences that have the same meaning, namely "push the red ball." Therefore, the decoder of the aRAE produces an appropriate action sequence by integrating such a semantic

Algorithm 1: Learning of Paired RAEs.

Require: $X^{\mathrm{dsc}}, X^{\mathrm{act}}$: paired dataset
Require: α, β, γ, A: hyperparameters
randomly initialize learnable parameters: θ
while not done **do**
Sample a random batch $\{x_i^{\mathrm{dsc}}, x_i^{\mathrm{act}}\}$ from $X^{\mathrm{dsc}}, X^{\mathrm{act}}$
Calculate $L_{\mathrm{dsc}}, L_{\mathrm{act}}, L_{\mathrm{shr}}$ by forward-path
Compute total loss: $L_{\mathrm{all}} \Leftarrow \alpha L_{\mathrm{dsc}} + \beta L_{\mathrm{act}} + \gamma L_{\mathrm{shr}}$
Compute gradients $\nabla_\theta L_{\mathrm{all}}$ by backward-path
Apply the gradients $\theta \Leftarrow \theta - A\nabla_\theta L_{\mathrm{all}}$
end while

representation and the current situation given as external visual input during decoding.

D. Representation Binding

To bind the encodings of robot actions and their corresponding descriptions, we use the other loss function L_{shr} in addition to the reconstruction losses $L_{\mathrm{dsc}}, L_{\mathrm{act}}$. We denote a batch of encodings of robot actions as $\{z_i^{\mathrm{act}} | 1 \leq i \leq N\}$ and the encodings of the corresponding descriptions as $\{z_i^{\mathrm{dsc}} | 1 \leq i \leq N\}$, where N is the batch size. The binding loss is as follows:

$$L_{\mathrm{shr}} = \sum_i^N \psi\left(z_i^{\mathrm{act}}, z_i^{\mathrm{dsc}}\right) + \sum_i^N \sum_{j \neq i} max\left\{0, \Delta + \psi\left(z_i^{\mathrm{act}}, z_i^{\mathrm{dsc}}\right) - \psi\left(z_i^{\mathrm{act}}, z_j^{\mathrm{dsc}}\right)\right\}. \quad (13)$$

Here, ψ is some function that calculates the distance or dissimilarity between two variables. The first term makes the representation of an action (resp., description) be close to that of its paired description (resp., action). By contrast, the second term makes the representation of an action (resp., description) be far from that of its unpaired description (resp., action) if the distance between them is less than that from the paired description (resp., action). The scalar Δ is the margin added to the distance between paired representations to enhance the loss.

E. Learning Procedure

All the learnable parameters are optimized by the gradient descent method with a random batch sampled from the paired dataset as described in Algorithm 1. Here, we control the importance of each loss function by introducing hyperparameters α, β, and γ. The term A controls the learning rate; we can choose any constant or adaptive rate, such as one controlled by Adam [31].

IV. Experiment I

A. Task Design

To evaluate the bidirectional translation capability of our proposed method, we performed a robot learning experiment using a small humanoid robot NAO. We put two colored cubes (red, green, or yellow; the cubes were always of different colors) in front of the robot (Fig. 2) in fixed positions; the number of possible cube arrangements was ${}_3P_2 = 6$. For each arrangement, the robot could perform the 12 actions listed in Table I.

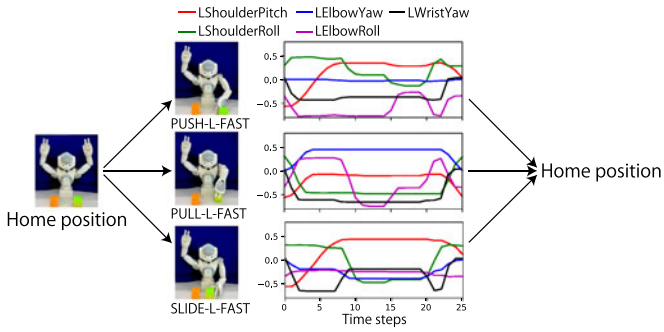

Fig. 2. Some examples of actions (only left-arm joints are plotted).

TABLE I
List of Actions

Action name	Details
PUSH-L-SLOW	Push the left cube slowly
PUSH-L-FAST	Push the left cube fast
PUSH-R-SLOW	Push the right cube slowly
PUSH-R-FAST	Push the right cube fast
PULL-L-SLOW	Pull the left cube slowly
PULL-L-FAST	Pull the left cube fast
PULL-R-SLOW	Pull the right cube slowly
PULL-R-FAST	Pull the right cube fast
SLIDE-L-SLOW	Slide the left cube to the right slowly
SLIDE-L-FAST	Slide the left cube to the right fast
SLIDE-R-SLOW	Slide the right cube to the left slowly
SLIDE-R-FAST	Slide the right cube to the left fast

We made the description corresponding to each action depend on the cube arrangement. More precisely, each descriptive sentence consists of a verb, an object, and an adverb, in that order. The verb and adverb do not depend on the cube arrangement, only on the action type, whereas the object does depend on the cube arrangement. The color subject to the action is assigned to the object. For example, if a red cube was placed on the left and a green cube was placed on the right, the description of the PUSH-L-SLOW action was "push red slowly" and that of the SLIDE-R-FAST action was "slide green fast". Therefore, there are $3 \times 3 \times 2 = 18$ possible sentences, each a combination of push/pull/slide + red/green/yellow + slowly/fast. Note that the numbers of possible actions and possible sentences are not the same due to this task setting.

B. Data and Parameters

We predesigned the action trajectories on a computer in advance. For each of the six cube arrangements, we made the robot produce each of the 12 action sequences while we recorded 10 joint angles on the robot arms as well as images (H: 120, W: 160, RGB) from a head-mounted camera every 300 ms. FAST actions and SLOW actions took approximately 26 and 39 time steps, respectively. We collected all 72 patterns (six arrangements with 12 actions each) six times.

TABLE II
DETAILED ARCHITECTURE OF THE VISUAL FEATURE EXTRACTOR

Layer type	Layers
Input (height, width, n_channels)	(120,160,3)
Convolution (n_channels, kernel size, stride)	- (8,4,2) - (16,4,2) - (32,4,2) - (64,8,5)
Fully connected (n_units)	-(384)-(192)-(10)-(192)-(384)
Deconvolution (n_channels, kernel size, stride)	- (32,8,5) - (16,4,2) - (8,4,2) - output (3,4,2)

After recording, we trained a convolutional autoencoder (CAE) from scratch with the collected images to use it as a visual feature extractor. We extracted 10-dimensional visual features from its centrer layer and used them as visual inputs for the aRAE. Table II explains the details of the CAE architecture.

The description sentences were represented as a series of one-hot vectors. We pre-/post-fixed <BOS/EOS> symbols to every sentence.

Finally, we divided the 72 possible patterns into two sets: 54 for training and 18 for testing[1]. The encoder and decoder of the dRAE were each a one-layer LSTM with 100 nodes, and those of the aRAE were each a two-layer LSTM with 100 nodes per layer. The dimensionality of the binding layer was also 100. For binding loss, we used Euclidean distance as the distance function ψ. The margin Δ was 1.0. The mixing rates of losses α, β, and γ were all 1.0. We used Adam as the optimizer (learning rate: 0.001), the batch size was 50, and the number of learning iterations was 20,000. With this hyper parameter set, we trained the model on a single GPU (Nvidia Titan X Pascal), which took 2,260 s. The source code of our model is available at https://github.com/ogata-lab/PRAE.

C. Result 1: Translation From Actions to Descriptions

First, we evaluated the model's ability to produce descriptions of given actions. As mentioned in Section III, this translation is performed by embedding an action sequence with the encoder of the aRAE and expanding the representation with the decoder of the dRAE. Here, the dRAE outputs the probabilistic distributions over the vocabulary as given by its softmax activation. We interpreted a word that corresponded to the element that took the maximum value as the model's output at each time step. If the output sentence was perfectly the same as the correct sentence, we judged the model to have succeeded in describing the action. For this criterion, the model succeeded to produce the correct descriptions for all the 54 trained patterns and the 18 untrained patterns.

D. Result 2: Translation From Descriptions to Actions

Next, we evaluated whether the model could produce appropriate robot actions by receiving an instruction and visual information. This translation is performed by embedding an instruction with the encoder of the dRAE and by decoding the representation with the decoder of the aRAE, conditioned on the visual information. We judged the action production to have been successful if the robot moved the cube indicated by the object word in the direction indicated by the verb more than 3 cm. Here, we also regarded the produced action as having been the FAST one if the robot returned to the neighborhood of its initial posture within 30 time steps; otherwise, we regarded it as the SLOW one.

For this criterion, the model produced appropriate actions for 36 out of the 54 trained patterns and for 12 out of the 18 untrained patterns. Fig. 3 shows the failed examples of joint trajectories produced by the model. Even in the failed cases, the model seems to have been able to produce trajectories that were rather similar to the predesigned reference trajectories even though the object was not moved in the correct direction because the produced trajectory differed slightly and the robot did not touch the correct side of the cube. For a detailed evaluation, we employed the dynamic time warping (DTW) that is used to measure similarities between time-series data. With DTW, we confirmed that in each failed case the produced trajectory was most similar to the correct one of the 12 reference trajectories.

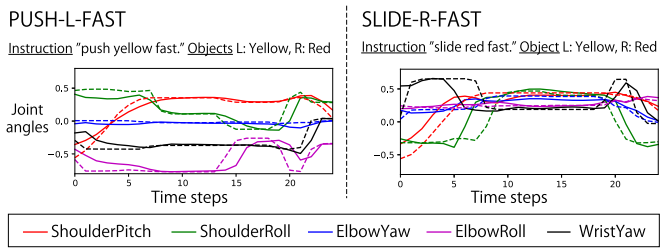

Fig. 3. Cases in which the robot failed to produce the appropriate action. The solid lines indicate the joint angles produced by the model; the broken lines indicate the predesigned target trajectories. Only five joints on the arm that moved are plotted. Even in these failed cases, the model seems to have been able to produce joint trajectories that were rather similar to the target trajectories.

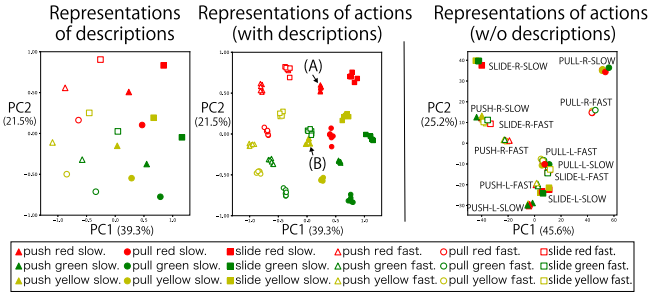

Fig. 4. [Left] Visualization of description encoding. [Center] Visualization of action encoding for being jointly learned with descriptions. The action sequences are bound with their descriptions and are thus represented semantically and compositionally. [Right] Visualization of action encoding for being learned alone. In this case, the encodings of the action sequences are not semantically organized.

E. Analysis of Shared Representations

Finally, we visualized the latent representations of the actions and their descriptions by using principal component analysis. First, the left panel of Fig. 4 shows the encodings of all 18

[1]Here, we removed patterns regularly so that each action, sentence, and cube arrangement would appear uniformly. Therefore, all possible actions, sentences, and cube arrangements are experienced during training, but there are still 18 unexperienced patterns that are "combinations" of them.

possible sentences. The symbol shape, color, and fill express the verb, object, and adverb, respectively. This panel shows that each part of speech was embedded systematically. The compositional structure of the sentences (i.e., verb, object, and adverb) is strongly represented in this space.

The center panel of Fig. 4 shows the encodings of the action sequences projected onto the same two-dimensional space. This panel shows that the action sequences were bound with their descriptions as expected, and thus represented compositionally. Here, note that the same type of action could be bound with different descriptions. For example, the symbol indicated by (A) represents an encoding of PUSH-R-SLOW in a situation in which there was a yellow cube on the left and a red one on the right; therefore, the action was bound with the description "push red slowly." Meanwhile, the symbol indicated by (B) also represents an encoding of PUSH-R-SLOW in a situation in which there was a green cube on the left and a yellow one on the right, and thus the action was bound with the description "push yellow slowly." Although (A) and (B) represent the same action type, they are far from each other because they are semantically different because of the differing cube arrangements. Likewise, different types of action can be bound with the same description.

To verify that these semantic representations truly arose from the joint learning with their descriptions, we performed an additional experiment in which the aRAE was trained alone (i.e., only the reconstruction error was used). In this case, the encodings of the action sequences were organized as 12 clusters that basically depended on the joint trajectories, as shown in the right panel of Fig. 4. The visual information (i.e., which color was acted upon) did not influence the latent representation greatly. This comparison reveals that the binding loss can substantially influence the latent representations of the action sequences to be grounded semantically in the paired descriptions.

V. Experiment II

In the previous experiment, the task space had to be limited because we used a real robot for the phases from data collection to evaluation. In this section, we evaluate the scalability of the proposed method and perform an ablation study to clarify which term of the binding loss function Eq. (13) is important. However, to our knowledge, there is no suitable large dataset of robot actions with paired descriptions. We instead use the KIT motion–language dataset [32], which consists of human whole-body motions and their annotations.

A. Data and Parameters

The KIT motion–language dataset contains 3,911 human whole-body motion sequences, comprising 44 joint angles and 6,278 annotations in natural language[2] (vocabulary size: 1,345). Note that this dataset does not include visual information. We therefore evaluate the proposed model on binding motions and descriptions only, without the visual condition. In accordance with the dataset size, we also scaled up the model. The changed hyper parameters are listed in Table III. We determined the model scale by referring to a model used in [12], although these are not precisely the same. We also changed the word representation form from one-hot to word embedding which was learned together, and changed the output form of the decoder of the aRAE into a mixture of Gaussians (for details, refer to [33]). At last, we replaced the Euclidian distance for the binding loss with the negative cosine similarity, following [24].

TABLE III
Learning Detail of the Paired RAEs

Hyperparameter	Value
Description encoder	Bidirectional LSTM (2 layers, 64 nodes)
Description decoder	LSTM (2 layers, 128 nodes)
Action encoder	Bidirectional LSTM (2 layers, 64 nodes)
Action decoder	LSTM (3 layers, 400 nodes)
Binding layer	512 nodes
Margin (Δ)	0.7
Batchsize	128
Learning epochs	100 (38 batches per epoch)
Word embedding dim.	64
Mixture components	20 Gaussians

TABLE IV
BLEU Scores for Test Data (Max [mean])

(i) full	(ii) w/o discrim.	(iii) only discrim.	(iv) unidir.(m2l)
.303 [.278]	.186 [.151]	.300 [.282]	.328 [.318]

In this experiment, we compare five models: (i) the proposed method (full); (ii) the proposed method without the second term of Eq. (13) (without discriminating) and (iii) without the first term of the Eq. (13) (only discriminating); (iv) a unidirectional model from motion to language (motion2language); and (v) a unidirectional model from language to motion (language2motion)[3]. We left out 10% of the dataset; this omitted part was used for testing. We trained each model five times from differently initialized parameters.

B. Results

We first quantitatively evaluated the ability to generate descriptions from motions[4]. Similarly to [12], we evaluated the performance by calculating the corpus-level BLEU score, which is modified, uniformly weighted from 1-gram to 4-gram, and smoothed. Table IV shows the results. We also show the samples generated by the full model (case (i)) and the unidirectional model (case (iv)). See Table V. Although the bidirectional models could not outperform the unidirectional model on BLEU score and sometimes generated grammatically and/or semantically wrong sentences, these models still succeeded in generating appropriate sentences in other examples. Moreover, from the

[2]Similarly to [12], we use only 2,846 motions that have a duration of less than 30 s (300 time steps) and 6,187 annotations (max length is 41 words). The data representation form is also the same as that used in [12].

[3]Here, unlike the bidirectional models that have no explicit experiences of translation during learning phase, the unidirectional models explicitly learn to translate sequences to their corresponding ones.

[4]In this experiment, we used the beam search (width: 5) to obtain candidates, and finally chose the sentence with the highest probability as the model output.

TABLE V
GENERATED DESCRIPTION SAMPLES

reference	(i) full	(iv) unidir.(m2l)
A person walks 3 fourths of a circle to the left.	A person walks in a circle counterclockwise.	A person walks in a circle to the left.
A person stand still and then gets pushed from behind.	A person is pushed back.	A person is pushed in the back and therefore makes a few steps forward.
A person waves a few times with both hands.	A person waves with the right hand.	A person waves with both hands.

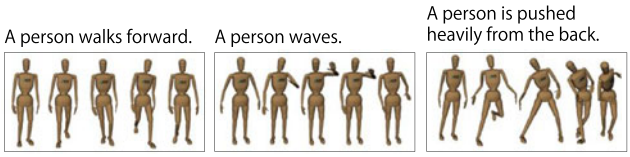

Fig. 5. Generated action samples for full model.

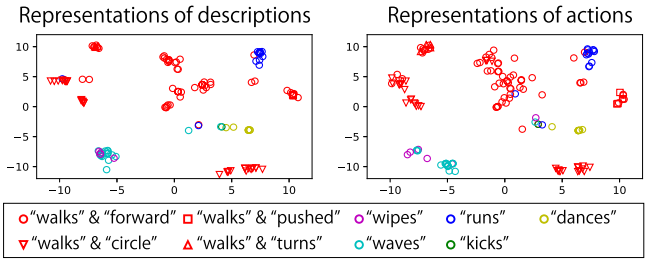

Fig. 6. t-SNE visualization of the shared representations. [Left] Each plot represents a sentence, with color differences reflecting differences in included keywords. [Right] Each plot represents a motion.

comparison among cases (i)–(iii), it seems that in the binding loss, the second term for discrimination has a critical effect on achieving better generation.

We next show the samples of motions generated from descriptions by the full model (case (i); see Fig. 5). We qualitatively observed that walking motions, which are included in the dataset more than any other motion, and some other basic motions described in a simple sentence were generated relatively well. Other motions were not very sophisticated: they sometimes looked meaningless or differed from the described motion type. We also visualize some generated samples in a supplementary video:

https://youtu.be/BN5vWy73vPA. However, in the current experiment, even in the case of a unidirectional model (case (v)), the quality of the generated samples looked quite similar. This implies that the learning of the decoder in the aRAE was insufficient, rather than the binding. Fig. 6 shows the representations of motions and descriptions in the binding space, projected by t-SNE. This visualization shows that the motions and their corresponding descriptions are embedded close to each other and semantically clustered. This suggests that the proposed binding method has the potential to be applied to large-scale datasets if the decoder module could be optimized together. Scalability should be explored in more detail in future work.

VI. DISCUSSION

We proposed a novel deep encoder–decoder architecture to bidirectionally translate between robot actions and their linguistic descriptions. In previous studies, the encoder–decoder models were mainly used to translate sequences into different sequences in a unidirectional and cross-modal way. By contrast, our paired RAE architecture and binding method can learn bidirectional translation between robot actions and their descriptions from paired action–description datasets. Moreover, although we did not perform such evaluation, our model would be able to translate an action sequence directly to another action sequence that has the same meaning through the semantic representation. For example, the PUSH-L-FAST action when there is a red cube on the left can be embedded as the semantic representation "push red fast." Therefore, if the red cube was to be relocated on the right in the decoding phase, the model would be able to convert the representation into the PUSH-R-FAST action.

Ogata *et al.* [14] proposed a neural network architecture to bidirectionally translate between robot actions and their descriptions. However, their model requires an iterative back-propagation calculation to get a latent representation with which to produce an action sequence given a sentence, or vice versa. Therefore, it takes time and the convergence might be unstable. In contrast, our model requires only one forward-path calculation to get a latent vector.

Plappert *et al.* [12] also proposed a method for bidirectional mapping between human body motions and their descriptions. Their model consisted of two distinct unidirectional models. This method always requires a parallel corpus of motions and descriptions. In contrast, our model has the advantage that we can first train the two RAEs independently with single modal datasets and then fine-tune them with a parallel corpus by using binding loss.

However, the present model has some limitations. The most important one is that it binds an action and its description in a point-to-point way. In essence, people can express the same action by various phrases and use the same description for various actions. Even if the situation remains perfectly the same, the relationships between actions and descriptions can still be many to many. One way to deal with this inevitable ambiguity would be to introduce a Bayesian method similar to the variational autoencoder [34].

A second limitation is that the model scalability is not well understood. Although in the second experiment the proposed method sometimes translated between motions and descriptions appropriately, the performance should be improved. The scalability in the case with visual information should also be investigated.

The third one is that in the first experiment we preprocessed the raw images before the learning of the RAEs. As a preliminary check, we analyzed the visual feature compressed by the pre-trained CAE. Doing so revealed that the light condition during the data collection had a large bias on the vision data space, although it did not have serious effects in the current experiment. In future work, to provide the model with more-effective and noise-robust representations, the training should be performed in an end-to-end manner from raw images to robot control.

VII. Conclusion

We proposed a novel paired RAE architecture in which the vector representations of robot actions and their descriptions are bound with each other. This binding enables the model to bidirectionally translate between the actions and the descriptions, conditioned on the current visual input. Visualizations of the shared representations showed that the robot actions were represented semantically and compositionally in the vector space by being learned jointly with descriptions. In future work, we will evaluate the model capacity for tasks that are more complex. We also must consider how to deal with the intrinsic ambiguity of linguistic expressions.

References

[1] S. Harnad, "The symbol grounding problem," *Phys. D:*, vol. 42, no. 1–3, pp. 335–346, 1990.

[2] J. Hatori, Y. Kikuchi, S. Kobayashi, and K. Takahashi, "Interactively picking real-world objects with unconstrained spoken language instructions," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2018, pp. 3774–3781.

[3] A. Das, S. Datta, G. Gkioxari, S. Lee, D. Parikh, and D. Batra, "Embodied question answering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.,* 2018, pp. 1–10.

[4] H. Ahn, T. Ha, Y. Choi, H. Yoo, and S. Oh, "Text2Action: Generative adversarial synthesis from language to action," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2018, pp. 5915–5920. http://arxiv.org/abs/1710.05298

[5] K. M. Hermann *et al.*, "Grounded language learning in a simulated 3D world," arXiv:1706.06551, 2017.

[6] D. S. Chaplot, K. M. Sathyendra, R. K. Pasumarthi, D. Rajagopal, and R. Salakhutdinov, "Gated-attention architectures for task-oriented language grounding," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 2819–2826. http://arxiv.org/abs/1706.07230

[7] H. Mei, M. Bansal, and M. R. Walter, "Listen, attend, and walk: Neural mapping of navigational instructions to action sequences," in *Proc. Natl. Conf. Artif. Intell.*, 2016, pp. 2772–2778.

[8] D. Misra, J. Langford, and Y. Artzi, "Mapping instructions and visual observations to actions with reinforcement learning," in *Proc. Conf. Empir. Methods Natural Lang. Process.*, 2017, pp. 1004–1015. http://arxiv.org/abs/1704.08795

[9] T. Yamada, S. Murata, H. Arie, and T. Ogata, "Dynamical integration of language and behavior in a recurrent neural network for humanrobot interaction," *Frontiers Neurorobotics*, vol. 10, no. 5, pp. 1–17, 2016.

[10] E. Tuci, T. Ferrauto, A. Zeschel, G. Massera, and S. Nolfi, "An experiment on behavior generalization and the emergence of linguistic compositionality in evolving robots," *IEEE Trans. Auton. Mental Develop.*, vol. 3, no. 2, pp. 176–189, Jun. 2011.

[11] S. Heinrich and S. Wermter, "Interactive language understanding with multiple timescale recurrent neural networks," in *Artificial Neural Networks and Machine Learning – ICANN 2014(Lecture Notes in Computer Science 8681)*, Wermter S. *et al.*, Eds. Berlin, Germany: Springer-Verlag, 2014, pp. 193–200.

[12] M. Plappert, C. Mandery, and T. Asfour, "Learning a bidirectional mapping between human whole-body motion and natural language using deep recurrent neural networks," arXiv:1705.06400, 2017. http://arxiv.org/abs/1705.06400

[13] W. Takano, S. Hamano, and Y. Nakamura, "Correlated space formation for human whole-body motion primitives and descriptive word labels," *Robot. Auton. Syst.*, vol. 66, pp. 35–43, 2015. http://dx.doi.org/10.1016/j.robot.2014.11.020

[14] T. Ogata, M. Murase, J. Tani, K. Komatani, and H. G. Okuno, "Two-way translation of compound sentences and arm motions by recurrent neural networks," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*. 2007, pp. 1858–1863.

[15] I. Sutskever, O. Vinyals, and V. Q. Le, "Sequence to sequence learning with neural networks," in *Proc. Neural Inf. Process. Syst.*, 2014, pp. 3104–3112.

[16] P. Anderson, D. Teney, J. Bruce, M. Johnson, S. Niko, and I. Reid, "Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 3674–3683.

[17] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proc. Int. Conf. Learn. Representations*, 2015, pp. 1–15. http://arxiv.org/pdf/1409.0473v6.pdf

[18] G. Park and J. Tani, "Development of compositional and contextual communicable congruence in robots by using dynamic neural network models," *Neural Netw.*, vol. 72, pp. 109–122, 2015. http://dx.doi.org/10.1016/j.neunet.2015.09.004

[19] N. Srivastava, "Unsupervised learning of video representations using LSTMs," *Proc. 32nd Int. Conf. Mach. Learn.*, 2015, pp. 843–852.

[20] L. Lu, X. Zhang, K. Cho, and S. Renals, "A study of the recurrent neural network encoder-decoder for large vocabulary speech recognition," in *Proc. INTERSPEECH 16th Annu. Conf. Int. Speech Commun. Assoc.*, 2015, pp. 3249–3253.

[21] I. Goodfellow *et al.*, "Generative adversarial nets," *Adv. Neural Inf. Process. Syst. 27*, 2014, pp. 2672–2680. http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf

[22] O. Fabius and J. R. V. Amersfoort, "Variational recurrent auto-encoders," in *Proc. Int. Conf. Learn. Representations Workshop*, 2015, pp. 1–5.

[23] A. Tikhonov and I. P. Yamshchikov, "Music generation with variational recurrent autoencoder supported by history," in *Proc. 13th Int. Symp. Comput. Music Multidiscip. Res.*, 2017, pp. 527–537.

[24] Y. Aytar, C. Vondrick, and A. Torralba, "See, hear, and read: Deep aligned representations," arXiv:1706.00932, 2017. https://arxiv.org/pdf/1706.00932.pdf

[25] A. Duarte, D. Surís, A. Salvador, and X. Giró, "Temporal-aware cross-modal embeddings for video and audio retrieval," in *Proc. Neural Inf. Process. Syst.*, 2017. http://arxiv.org/abs/1609.08675

[26] R. Arandjelović and A. Zisserman, "Objects that Sound," arXiv:1712.06651, 2017. http://arxiv.org/abs/1712.06651

[27] F. Saussure, *Course in General Linguistics*. New York, NY, USA: Philosophical Library, 1959.

[28] K. Cho *et al.*, "Learning phrase representations using RNN encoderdecoder for statistical machine translation," in *Proc. Conf. Empir. Methods Natural Lang. Process.*, 2014, pp. 1724–1734.

[29] F. A. Gers and J. Schmidhuber, "Recurrent nets that time and count," *Proc. IEEE-INNS-ENNS Int. Joint Conf. Neural Netw.*, 2000, vol. 3, pp. 189–194.

[30] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*. Cambridge, MA, USA: MIT Press, 1986, pp. 318–362.

[31] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int.Conf. Learn. Representations*, Dec. 2015, pp. 1–15. http://arxiv.org/abs/1412.6980

[32] M. Plappert, C. Mandery, and T. Asfour, "The KIT motion-language dataset," *Big Data*, Mary Ann Liebert, Inc., vol. 4, no. 4, Dec., 2016, doi: 10.1089/big.2016.0028.

[33] A. Graves, "Generating sequences with recurrent neural networks," arXiv:1308.0850, pp. 1–43, 2013.

[34] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," in *Proc. Int. Conf. Learn. Represent.*, 2014, pp. 1–14.