Embodied Language Learning with Paired Variational Autoencoders

1st Ozan Özdemir
Knowledge Technology Group
Department of Informatics
University of Hamburg
Hamburg, Germany
oezdemir@informatik.uni-hamburg.de

2nd Matthias Kerzel
Knowledge Technology Group
Department of Informatics
University of Hamburg
Hamburg, Germany
kerzel@informatik.uni-hamburg.de

3rd Stefan Wermter
Knowledge Technology Group
Department of Informatics
University of Hamburg
Hamburg, Germany
wermter@informatik.uni-hamburg.de

2021 IEEE International Conference on Development and Learning (ICDL) | 978-1-7281-6242-3/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICDL49984.2021.9515668

***Abstract*—Language acquisition is an integral part of developmental robotics, which aims at understanding the key components in human development and learning to utilise them in artificial agents. Similar to human infants, robots can learn language while interacting with objects in their environments and receiving linguistic input. This process, also coined as embodied language learning, can enhance language acquisition in robots via multiple modalities including visual and sensorimotor input. In this work, we explore ways to translate a simple action in a tabletop environment into various linguistic commands based on an existing approach which exploits the idea of multiple autoencoders. While the existing approach focuses on strict one-to-one mappings between actions and descriptions by implicitly binding two standard autoencoders in the latent space, we propose a variational autoencoder model to facilitate one-to-many mapping between actions and descriptions. Additionally, for extracting visual features, we employ channel-separated convolutional autoencoders to better handle complex visual input. The results show that our model outperforms the existing approach in associating multiple commands with the corresponding action.**

***Index Terms*—embodied language learning, variational and recurrent autoencoders, one-to-many mapping, robot actions**

I. Introduction

The linguistic capabilities of robots are still substantially inferior to humans although there have been many attempts at natural human-robot communication in recent years. According to Bisk et al. [1], embodiment (action taking in the environment) is the needed next step after perception (using multimodal input) in language acquisition and production. An embodied agent must be able to relate language to physical control via sensory perception as action and control open up new dimensions to understanding and actively learning about the world [1]. With embodiment, natural language processing can be brought to a level at which it can be deployed in realistic HRI contexts [1].

Embodied language learning is one of the main research topics in embodied robotics [2]–[5]. In a typical scenario, a robot would execute an action and receive a description of the action. In well-structured environments, the complexity and ambiguity of language can be overcome by strictly defining the corpus, with each word having a distinct meaning. In this fashion, it is possible to translate actions into descriptions. Nevertheless, this strict one-to-one mapping between robot actions and linguistic descriptions is not natural in human-to-human communication since we may use different words to describe the same action. In order to break the premise of one-to-one binding, alternative descriptions can be used to define an action, thereby facilitating one-to-many binding between action and language.

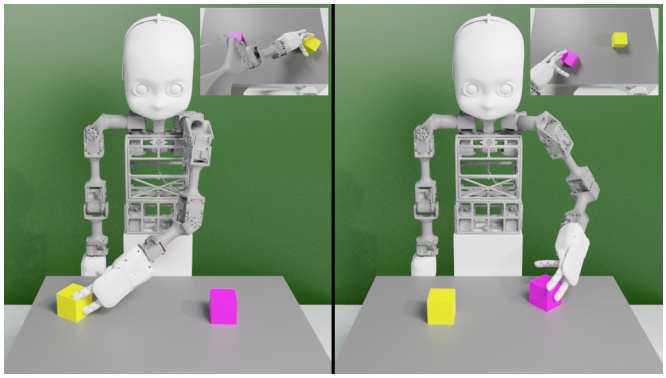

Fig. 1. The NICO robot in the simulation environment: on the left, NICO is sliding the yellow cube; on the right, NICO is pulling the violet cube. In both segments, NICO's field of view is shown in the top right insets.

In this study, we have a robot, i.e. NICO (Neuro-Inspired COmpanion [6], [7]), interacting with cubes of different colours on a table in a simulation environment using its arm and hand while descriptions of the actions are provided - see Figure 1. In this setup, we control the complexity of language and motion with appropriate actions and descriptions (e.g. "push the red cube fast", "pull the green cube slowly"). Inspired by the work of Yamada et al. [8] with paired recurrent autoencoders (PRAE), we propose a novel paired variational autoencoder (PVAE) architecture to enable our robot to translate from action to language in a one-to-many fashion. Our architecture is composed of two variational autoencoders: one for language and one for action. These two autoencoders, which consist of LSTMs, are integrated using Yamada et al.'s [8] binding loss in the latent space. PVAE extends the action-

The authors gratefully acknowledge support from the German Research Foundation DFG, project CML (TRR 169). We thank Cornelius Weber, Jae Hee Lee and Mengdi Li for their valuable feedback on the document.

Authorized licensed use limited to: Bibliothekssystem Universitaet Hamburg. Downloaded on August 27,2021 at 10:09:52 UTC from IEEE Xplore. Restrictions apply.

to-description translation capability of PRAE [8] by being capable of producing alternative versions of a description from an action (one-to-one vs one-to-many association). Aiming to address the research question of one-to-many association of actions and descriptions, i.e. an action can be translated into different variations of a description, the novelty of our PVAE model is exploiting a Bayesian method, i.e. variational autoencoders, to deal with the inexactness of relationships between actions and descriptions [9]. PVAE learns from a dataset[1] that pairs visual observations and kinematics of actions with their corresponding textual descriptions. The robot actions are described as sequences of joint angle values whilst the visual input is gathered from the egocentric perspective of the robot to be extracted via a novel channel-separated convolutional autoencoder (CAE). Besides, the textual descriptions are fed into the network word by word as sentences with one-hot encoding.

Our contribution is two-fold:

1) We show that employing variational autoencoders instead of standard autoencoders leads to a better one-to-many action-to-description translation accuracy, especially with a larger corpus and more data, hence addresses the linguistic ambiguity between an action and its probable descriptions.
2) The experiment results also indicate the superiority of channel separation (channel-separated CAE) in visual feature extraction, leading to a more accurate recognition of object colours where the objects cover only a small portion of the visual field.

II. Related Work

Recently, many studies have been conducted in embodied language learning where a robot interacts with several objects on a table, given verbal instructions [10], [11], [12], [13], [8]. Shao et al. [10] enable a robot to acquire manipulation concepts which can be defined as a mental representation of verbs in a sentence. They propose a robot learning framework to obtain manipulation concepts from human video demonstrations. To be specific, a model which takes a language instruction and a scene image as input and produces a robot motion trajectory as output is trained by computing a reward for the executed trajectory using a video classifier. The approach combines neural networks with reinforcement learning.

Heinrich et al. [11] introduce a novel neural network architecture for embodied crossmodal language grounding: the adaptive multiple timescale recurrent neural network (MTRNN) model. The architecture is a neurocognitive model, with auditory, sensorimotor and visual perception capabilities, that produces spoken language while the learner interacts with objects in the environment. The approach shows promising results towards generalisation and hierarchical concept decomposition.

In their work [12], Shridhar et al. develop a robot system to manipulate objects based on visual and linguistic input. Unlike prior work, their model has the freedom of numerous object categories without constraints for the robot to interact with a diverse set of objects available for everyday use. The authors choose the grounding by generation approach INGRESS (interactive visual grounding of referring expressions) which is an architecture for language grounding with two neural networks trained on large datasets to generate a linguistic expression from the input image. The generated expression is compared with the input expression to spot the referred object.

Hatori et al. [13] focus on the interaction between a human operator and a system that handles objects through speech input. Over 100 objects are scattered across four bins in the environment. The robot is instructed to pick up an object and move it to a specific bin. The authors prioritise realistic, highly cluttered environments with many objects being occluded. Their neural network model receives multimodal input, i.e. a spoken instruction and an RGB image. The model has two modules: object recognition and language understanding. By training these two modules jointly, the system learns to associate object names with actual objects. Moreover, it learns different attributes of an object like its colour, texture or size.

Yamada et al. [8] propose the paired recurrent autoencoder (PRAE) model to address the problem of bidirectional translation between robot actions and linguistic descriptions. The model is trained with a dataset of robot action and textual description pairs. PRAE has two main components: action and description autoencoders. Moreover, the two autoencoders are trained together by exploiting an implicit binding of actions and descriptions as the autoencoders do not have an explicit connection. Thereby, the model can generate actions from textual descriptions and vice versa (bidirectional translation). This simple yet powerful approach of action-description mapping, which facilitates bidirectional translation, makes PRAE more favourable over complex architectures. However, Yamada et al. [8] associate an action with a description in a one-to-one way, although actions can be expressed in many different ways.

In this work, we extend the PRAE architecture [8] by replacing regular autoencoders with variational autoencoders, hence allowing the network to learn one-to-many mapping between robot actions and instructions. Furthermore, we modify the visual feature extraction by separately training the channels of the convolutional autoencoder, which leads to a more accurate recognition of object colours.

III. Method

Following the PRAE approach [8], we use two recurrent autoencoders to learn robot actions and their descriptions and the mapping between them in the latent space. Both autoencoders have a very similar architecture with LSTMs for temporal sequence processing. Different from PVAE [8], instead of regular autoencoders, we employ variational recurrent autoencoders (VRAE) [14], in which latent vectors are randomly sampled from a normal distribution over latent variables. Combining the capabilities of LSTMs and the Stochastic Gradient Variational Bayes (SGVB) [9] allows us to have efficient unsupervised learning on sequential data [14]. Moreover, following the

[1]https://www.inf.uni-hamburg.de/en/inst/ab/wtm/research/corpora.html

Authorized licensed use limited to: Bibliothekssystem Universitaet Hamburg. Downloaded on August 27,2021 at 10:09:52 UTC from IEEE Xplore. Restrictions apply.

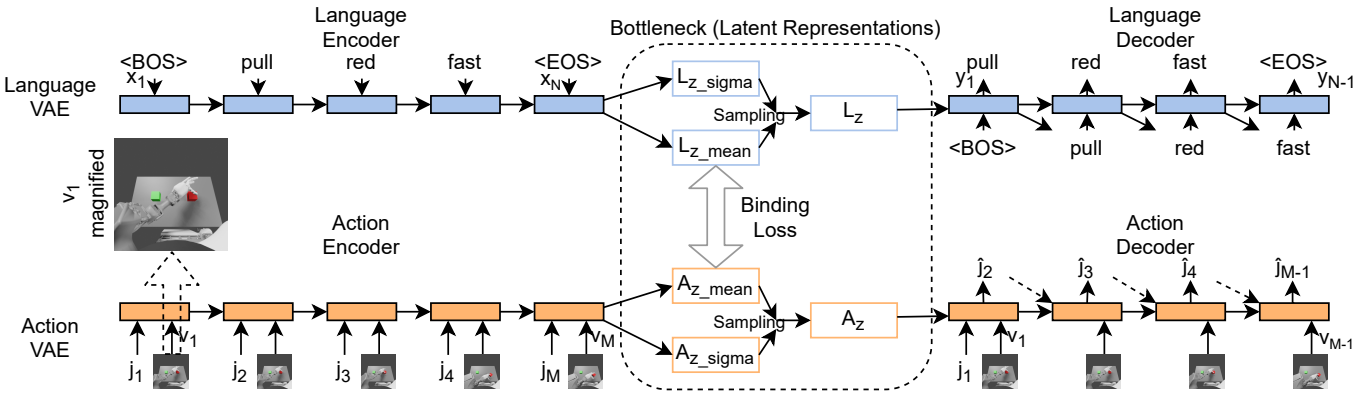

Fig. 2. The architecture of the proposed PVAE: the language VAE (depicted with blue rectangles which denote unfolded LSTMs) is responsible for reconstructing descriptions while the action VAE (depicted with orange rectangles which denote unfolded LSTMs) is responsible for reconstructing the joint angles at each time step. The input to the language VAE is a one-hot encoded word of a given description at a time whereas the action VAE takes as input joint angle values and visual features (v_1 is magnified for visualisation purposes) at a time. The two VAEs are implicitly bound via a binding loss between their latent representations.

advice of Yamada et al. [8] to use a Bayesian method, we overcome the inevitable ambiguity of relations between actions and descriptions. Thus, our method is able to map an action to multiple language instructions. The two variational autoencoders that form the PVAE architecture are the language VAE and action VAE. The language VAE learns descriptions in an unsupervised manner while the action VAE learns joint angles conditioned on the visual input similarly.

A. Architecture

As shown in Figure 2, the architecture is composed of two VAEs: language and action VAE. The input to the language VAE is a sentence describing a robot action. The sentence is fed into the language encoder word by word using one-hot encoding. After the encoding phase, the encoded representation is used to extract latent representations using the reparameterisation trick [9] following the VRAE approach [14]. These latent representations are exploited in the decoder to reproduce the sentence describing the action.

The action VAE has two types of input: robot joint angles and visual input from the perspective of the robot. After the encoding, similar to the language VAE, latent representations are extracted from encoded actions in the bottleneck. The action decoder reproduces the joint angles from the latent representations, conditioned on the visual features. The two VAEs have no explicit connection, but they are integrated with a binding loss reducing the distance between two latent variables, binding actions to descriptions bidirectionally [8].

B. Language Autoencoder

The language VAE encodes descriptions so that they can be reproduced by decoding. It has two components which are the language encoder and decoder. The language encoder embeds a description of length N (x_1, x_2, ..., x_N) into two fixed-dimensional vectors z_{mean} and z_{sigma} with the following equations:

$$h_t^{\text{enc}} = \text{EncCell}(x_t, h_{t-1}^{\text{enc}}) \quad (1 \leq t \leq N), \tag{1}$$

$$z_{\text{mean}} = W_{\text{mean}}^{\text{enc}} \cdot h_N + b_{\text{mean}}^{\text{enc}}, \tag{2}$$

$$z_{\text{sigma}} = W_{\text{sigma}}^{\text{enc}} \cdot h_N + b_{\text{sigma}}^{\text{enc}}, \tag{3}$$

$$z_{\text{lang}} = z_{\text{mean}} + z_{\text{sigma}} \cdot \mathcal{N}(\mu, \sigma^2), \tag{4}$$

where EncCell is an LSTM, h_t is the state of the LSTM at time step t, h_0 is set as a zero vector and $\mathcal{N}$ is a Gaussian distribution. z_{lang} is the latent representation of a description. The language decoder generates a sequence by recursively expanding z_{lang}:

$$h_0^{\text{dec}} = W^{\text{dec}} \cdot z_{\text{lang}} + b^{\text{dec}}, \tag{5}$$

$$h_t^{\text{dec}} = \text{DecCell}(y_{t-1}, h_{t-1}^{\text{dec}}) \quad (1 \leq t \leq N-1), \tag{6}$$

$$y_t = f(W^{\text{out}} \cdot h_t^{\text{dec}} + b^{\text{out}}) \quad (1 \leq t \leq N-1), \tag{7}$$

where DecCell is an LSTM and f is the softmax activation function. y_0 is given as a first symbol indicating the beginning of the sentence.

C. Action Autoencoder

The action VAE encodes robot actions so that they can be reproduced by its decoder. Similar to the language VAE, it has two components which are action encoder and decoder. The action encoder encodes a sequence of length M ((j_1, v_1), (j_2, v_2), ..., (j_M, v_M)) that concatenates joint angles j with visual features v (extracted by the channel-separated convolutional autoencoder):

$$h_t^{\text{enc}} = \text{EncCell}(v_t, j_t, h_{t-1}^{\text{enc}}) \quad (1 \leq t \leq M), \tag{8}$$

$$z_{\text{mean}} = W_{\text{mean}}^{\text{enc}} \cdot h_M + b_{\text{mean}}^{\text{enc}}, \tag{9}$$

$$z_{\text{sigma}} = W_{\text{sigma}}^{\text{enc}} \cdot h_M + b_{\text{sigma}}^{\text{enc}}, \tag{10}$$

$$z_{\text{act}} = z_{\text{mean}} + z_{\text{sigma}} \cdot \mathcal{N}(\mu, \sigma^2), \tag{11}$$

where EncCell is an LSTM, h_t is the state of the LSTM at time step t, h_0 is set as a zero vector and $\mathcal{N}$ is a Gaussian distribution. z_{act} is the latent representation of a robot action. The action decoder reconstructs the joint angles:

$$h_0^{\text{dec}} = W^{\text{dec}} \cdot z_{\text{act}} + b^{\text{dec}}, \tag{12}$$

Authorized licensed use limited to: Bibliothekssystem Universitaet Hamburg. Downloaded on August 27,2021 at 10:09:52 UTC from IEEE Xplore. Restrictions apply.

$$h_t^{\text{dec}} = \text{DecCell}(v_t, \hat{j}_t, h_{t-1}^{\text{dec}}) \quad (1 \leq t \leq M-1), \quad (13)$$

$$\hat{j}_{t+1} = f(W^{\text{out}} \cdot h_t^{\text{dec}} + b^{\text{out}}) \quad (1 \leq t \leq M-1), \quad (14)$$

where DecCell is an LSTM, f is the hyperbolic tangent activation function and $\hat{j}_1$ is equal to j_1.

D. Visual Feature Extraction

We follow the visual feature extractor architecture provided by Yamada et al. [8]. Accordingly, our CAE accepts 120×160 RGB images gathered from the egocentric view of the robot; it consists of a convolutional encoder, a fully-connected bottleneck (latent representations) and a deconvolutional decoder. However, our CAE is trained separately for each channel (red, green and blue) to recognise different colours more accurately, i.e. channel separation. After training, we extract the visual features of each image for all channels from the bottleneck situated between the encoder and the decoder. The visual features extracted from each channel are then concatenated to form the ultimate visual features v.

E. Sampling and Binding

The two VAEs have identical random sampling procedures. After producing the latent variables z_{mean} and z_{sigma} using fully connected layers, a normal distribution $\mathcal{N}(\mu, \sigma^2)$ is used to derive random latent representations, which are, in turn, used with z_{mean} and z_{sigma} to arrive at the final sample latent representation z [9]:

$$z = z_{\text{mean}} + z_{\text{sigma}} \cdot \epsilon \quad (15)$$

where ϵ is the approximation of $\mathcal{N}(0, 0.01)$.

Similar to [8], in order to bind the encodings of the language and action VAEs, we use an extra loss term that brings z_{mean} values of two VAEs closer. This allows the network to bidirectionally translate actions to descriptions and vice versa after training without an explicit fusion of the two modalities. The loss term [8] is given below:

$$L_{\text{binding}} = \sum_{i}^{B} \psi(z_{\text{mean}_i}^{\text{lang}}, z_{\text{mean}_i}^{\text{act}}) + \sum_{i}^{B} \sum_{j \neq i} max \left\{0, \Delta + \psi(z_{\text{mean}_i}^{\text{lang}}, z_{\text{mean}_i}^{\text{act}}) - \psi(z_{\text{mean}_j}^{\text{lang}}, z_{\text{mean}_i}^{\text{act}})\right\} \quad (16)$$

where B stands for the batch size and ψ is the Euclidean distance. The first term in the equation aligns the corresponding instructions and actions whereas the second term helps distinguish irrelevant actions from descriptions. Hyperparameter Δ is used to adjust the second term.

F. Loss Function

The total loss function has three main components: reconstruction, regularisation and binding loss. The binding loss is calculated for both VAEs together. In contrast, the reconstruction and regularisation losses are calculated independently for each VAE. The respective reconstruction loss for the language VAE L_{lang} and action VAE L_{act} are given as:

$$L_{\text{lang}} = \frac{1}{N-1} \sum_{t=1}^{N-1} \left(- \sum_{w}^{W} x_{t+1}(w) \log y_t(w) \right), \quad L_{\text{act}} = \frac{1}{M-1} \sum_{t=1}^{M-1} \| j_{t+1} - \hat{j}_{t+1} \|_2^2, \quad (17)$$

where W is the vocabulary size. The regularisation loss, which is specific to variational autoencoders, is defined as Kullback–Leibler divergence for language $D_{\text{KL}_{\text{lang}}}$ and action $D_{\text{KL}_{\text{act}}}$. Therefore, the overall loss function can be defined as:

$$L_{\text{all}} = \alpha L_{\text{lang}} + \beta L_{\text{act}} + \gamma L_{\text{binding}} + \alpha D_{\text{KL}_{\text{lang}}} + \beta D_{\text{KL}_{\text{act}}} \quad (18)$$

where α, β and γ are weighting factors for different terms in the loss function. In our experiments, α and β are set to 1 whilst γ is set to 2 in order to sufficiently bind the two modalities.

G. Training Details

The model was trained with both networks (the language and action VAEs) together for 15,000 iterations and the gradient descent method was employed to update the weights using the Adam optimiser [15]. The learning rate was set to 10^{-4} and the batch size was chosen as 100 (100 description and action pairs) after preliminary studies.

IV. Evaluation and Results

The proposed PVAE is evaluated with two experiments of varying data sizes to display the advantage of using variational autoencoders over regular autoencoders, i.e. PRAE in [8], and the advantage of using channel separation technique in visual feature extraction. We also want to analyse the impact of a larger corpus on action-to-language translation. Therefore, in both experiments, our PVAE and Yamada et al.'s PRAE [8] are trained on the same datasets and we evaluate their accuracy in translating actions into descriptions. In addition, to see the effect of channel separation on the overall translation accuracy, we train our architecture with visual features provided by a regular CAE without channel separation. In all experiments, two cubes of different colours are placed on a table at which the robot is seated to interact with the cubes. For the first experiment, each cube is one of three colours (red, green, blue) and for the second, one of six colours (red, green, blue, yellow, cyan, violet). The words (vocabulary) that make up the descriptions are given in Table I. Two-word phrases like 'move up' are considered as one word for the one-hot encoding. We introduce a more diverse vocabulary by adding an alternative word for each word in the original vocabulary.

Each cube arrangement has two cubes and these cubes are never of the same colour. There are three action types ('PUSH', 'PULL', 'SLIDE'), two positions ('L', 'R') and two speed settings ('SLOW', 'FAST'): 12 possible actions. Each sentence has three words (excluding the <BOS/EOS> tags

TABLE I
VOCABULARY

Original	Alternative	Original	Alternative
push	move up	yellow	blonde
pull	move down	cyan	greenish blue
slide	move sideways	violet	purple
red	scarlet	slowly	unhurriedly
green	harlequin	fast	quickly
blue	azure		

Authorized licensed use limited to: Bibliothekssystem Universitaet Hamburg. Downloaded on August 27,2021 at 10:09:52 UTC from IEEE Xplore. Restrictions apply.

TABLE II
ACTION-TO-DESCRIPTION TRANSLATION ACCURACY

Method	Experiment 1 (3 colours) Training - Test
PRAE + regular CAE	33.33 ± 1.31% - 33.56 ± 3.03%
PVAE + regular CAE	66.6 ± 1.31% - 65.28 ± 6.05%
PVAE + channel-separated CAE	**100.00 ± 0.00% - 90.28 ± 4.61%**
Method	Experiment 2 (6 colours) Training - Test
PRAE + regular CAE	33.64 ± 1.13% - 33.3 ± 0.98%
PVAE + regular CAE	69.60 ± 0.46% - 61.57 ± 2.01%
PVAE + channel-separated CAE	**100.00 ± 0.00% - 100.00 ± 0.00%**

which indicate the beginning or end of a sentence) with the first word indicating the action, the second the cube colour and the last the speed at which the action is taken (e.g. "push green slowly"). Therefore, without the alternative words, there are 18 possible sentences (3 action verbs $\times$ 3 colours $\times$ 2 adverbs). As a result, our dataset consists of six cube arrangements (12 for the second experiment), $18 \times 8 = 144$ possible sentences ($36 \times 8 = 288$ for the second experiment - the factor of eight because of eight alternatives for each sentence) and 12 actions ($3 \times 2 \times 2$). We have 72 patterns for the first experiment (12 actions with six cube arrangements each) and 144 patterns for the second. Following Yamada et al. [8], we choose the patterns (action-description-arrangement combinations) rigorously ensuring that combinations of action, description and cube arrangements selected for the test set do not exist in the training set although every action, description and cube arrangement is shown during training. Therefore, 54 patterns are used for training while the remaining 18 for testing (second experiment: 108 for training, 36 for testing). Each pattern is collected six times in the simulation with random variations on the action execution resulting in different joint trajectories. Additionally, we use 4-fold cross-validation to provide more reliable results (consult Table II).

The robot used in our experiments is NICO (Neuro-Inspired COmpanion) [6], [7] in a virtual environment created with the Blender software - see Figure 1. NICO is a humanoid robot, has a height of approximately one metre and a weight of approximately 20 kg. We use the left arm of NICO to interact with the objects utilising five joints. Actions are realised with an inverse kinematics solver. NICO has a camera in each of its eyes which is used to extract egocentric visual images.

A. Experiment I - Three Colour Alternatives

We use the same instructions and actions as in [8], e.g. "PUSH-R-SLOW" which can be interpreted as "push the right object slowly". We use three colour options for the cubes as in [8]. However, the instructions are extended by adding an alternative for each word in the vocabulary. Hence, the vocabulary size of 9 is extended to 17 (we do not add an alternative for <BOS/EOS> tags.) As every sentence is composed of three words, we extend the number of sentences by a factor of eight ($2^3 = 8$).

After training our PVAE and PRAE, we test them for action-to-description translation. For the reproduced description to count as correct, all three words (plus the <BOS/EOS> tag) have to be correctly predicted. As each description has seven more alternatives, predicting any of the eight correct descriptions is considered correct. As can be seen in Table II, our model is able to translate approximately 90% of the patterns in the test set (last row) whilst PRAE could translate only one third of the patterns. Thus, our model outperforms PRAE in one-to-many mapping.

We also test the impact of channel separation on the translation accuracy by training our model with visual features extracted with the regular CAE as described in Yamada et al.'s approach [8]. In Table II, we can see that using our variational approach alone increases the accuracy significantly. Nevertheless, using PVAE with channel-separated CAE improves the results further, indicating the superiority of channel separation in our tabletop setting. Therefore, our approach with variational autoencoders and a channel-separated CAE is superior to both PRAE and PVAE with regular visual feature extraction in this experiment with three colours.

B. Experiment II - Six Colour Alternatives

For testing the limits of our PVAE and the impact of more data with a larger corpus, we add three more colour options for the cubes: yellow, cyan and violet. These secondary colours are combined amongst themselves for the arrangements in addition to the colour combinations used in the first experiment, i.e. a cube of a primary colour and a cube of a secondary colour do not co-occur. Therefore, this experiment has 12 arrangements. Moreover, the vocabulary size is extended to 23 compared to 17 in Experiment I (two alternative words for each colour - see Table I). As in the first experiment, each sentence has eight alternative ways to be described.

We train both PVAE and PRAE [8] on the extended dataset from scratch and test both architectures. As shown in Table II, PVAE succeeds in performing 100% by translating every pattern from action to description correctly, even for the test set. In contrast, PRAE performs poorly in this setting and manages to translate only one third of the descriptions correctly in the test set. Compared with the accuracy values reached in the first experiment with less data and a smaller corpus, extension of the dataset helps PVAE to perform better in translation whilst PRAE is not able to take advantage of more data.

As in Experiment I, we also test the influence of channel separation on the translation accuracy by training PVAE with visual features provided by a regular CAE. In this setting, PVAE only achieves around 61% of accuracy in the test set. This highlights again the importance of channel separation in visual feature extraction for our setup. Whilst the improvement by using our PVAE over PRAE is significant, further improvement is made by utilising the channel-separated CAE.

V. DISCUSSION

The results from both experiments show that our variational autoencoder approach with a channel-separated CAE visual feature extraction outperforms the standard autoencoder

Authorized licensed use limited to: Bibliothekssystem Universitaet Hamburg. Downloaded on August 27,2021 at 10:09:52 UTC from IEEE Xplore. Restrictions apply.

approach, i.e. PRAE [8], in the one-to-many translation of actions into language commands. Our approach not only proved more successful in the case of three colour alternatives per cube but also in the case of six colour alternatives by a large margin. Specifically, when the dataset and the corpus were extended, our PVAE model performed better, proving that a Bayesian method like variational autoencoders can scale up with more data for generalisation, whereas standard autoencoders cannot capitalise on more data. Moreover, standard autoencoders are fairly limited when it comes to handling ambiguity in linguistic input. In contrast, variational autoencoders yield remarkably better results in one-to-many mapping between actions and descriptions, because stochastic generation (random normal distribution) within the latent feature extraction allows latent representations to slightly vary, leading to VAEs learning not only one specific description but various descriptions for each action. Moreover, analysing the specific case in which we train our PVAE with visual features extracted by the standard CAE demonstrates that separating the channels of CAE helps to increase distinguishing objects of different colours significantly with a visual input in our setup in which the objects cover only a modest portion of the visual field.

On the one hand, the translation accuracy results of PRAE show that adding more data (i.e. more colour options) does not help regular autoencoders to bind actions with translations more successfully in the case of one-to-many mapping. In fact, the results of PRAE on both experiments are very similar, which also indicates that regular autoencoders are not suitable for this task as they do not respond well to data with more variations.

On the other hand, our PVAE performs even better with more colour alternatives when trained with visual features extracted by the channel-separated CAE. This shows the importance of larger and more various data in one-to-many mapping. When compared with our PVAE and channel-separated CAE approach, the 'PVAE + regular CAE' option yields significantly lower translation accuracy, which exhibits the importance of visual input since the channel-separated CAE performs a more accurate visual feature extraction than a standard CAE in our setup. A significant portion of errors for the 'PVAE + regular CAE' case was caused by failing to distinguish colours, e.g. "push green slowly" instead of "push red slowly".

Lastly, although it is not in the scope of this paper, we have also verified that our PVAE is able to translate descriptions to actions like PRAE. The preliminary studies show that PVAE succeeds in reconstructing the joint angle values from descriptions.

VI. Conclusion

In this work, we have extended one-to-one mapping between actions and descriptions to one-to-many mapping by employing variational autoencoders [9] to tackle the ambiguity of various descriptions describing the same action. We have also shown that, in table-top environments with the objects covering only a small portion of the visual field, it is plausible to use a channel-separated CAE to distinguish objects of different colours. Although we have not tested the limits of PVAE, e.g. training it for many-to-many mapping between robot actions and their descriptions, the results show that our approach with variational and channel-separated autoencoders outperforms the standard autoencoder approach PVAE [8] in one-to-many action-to-description translation. Moreover, our PVAE network performs even better with more data and a larger corpus, suggesting that Bayesian methods like VAEs may scale up well with more data. In real applications, exploiting Bayesian methods may lead to profiting from larger and more complex data. In the future, we will extend our model to address the many-to-many association of action and language. Moreover, we will further investigate the description-to-action translation in the real world. Finally, we will extend our approach for the interpretation of more realistic unconstrained language from human users.

References

[1] Y. Bisk et al., 'Experience Grounds Language', Empirical Methods in Natural Language Processing, pp. 8718-8735, 2020.

[2] S. Heinrich, C. Weber, S. Wermter, R. Xie, Y. Lin, and Z. Liu, 'Crossmodal language grounding, learning, and teaching', NIPS2016 Workshop on Cognitive Computation, pp. 62-68, 2016.

[3] Ng, H.G. et al., 'Hey Robot, Why Don't You Talk to Me?' Proc. 26th IEEE International Symposium on Robot and Human Interactive Communication, 2017.

[4] S. Heinrich et al., 'Embodied Multi-modal Interaction in Language learning: the EMIL data collection', ICDL-EpiRob Workshop on Active Vision, Attention, and Learning, 2018.

[5] S. Heinrich and S. Wermter, 'Interactive natural language acquisition in a multi-modal recurrent neural architecture', Connection Science, vol. 30, no. 1, pp. 99–133, 2018.

[6] M. Kerzel, E. Strahl, S. Magg, N. Navarro-Guerrero, S. Heinrich, and S. Wermter, 'NICO - Neuro-Inspired COmpanion: A developmental humanoid robot platform for multimodal interaction', Proceedings of the IEEE International Symposium on Robot and Human Interactive Communication, 113–120, 2017.

[7] M. Kerzel, T. Pekarek-Rosin, E. Strahl, S. Heinrich and S. Wermter, 'Teaching NICO How to Grasp: An Empirical Study on Crossmodal Social Interaction as a Key Factor for Robots Learning From Humans', Front. Neurorobot. 14:28. 2020.

[8] T. Yamada, H. Matsunaga, and T. Ogata, 'Paired Recurrent Autoencoders for Bidirectional Translation Between Robot Actions and Linguistic Descriptions', IEEE Robotics and Automation Letters, vol. 3, no. 4, pp. 3441–3448, Oct. 2018.

[9] D. P. Kingma and M. Welling, 'Auto-Encoding Variational Bayes', in Proc. Int. Conf. on Learning Representations, pp. 1–14, 2014.

[10] L. Shao, T. Migimatsu, Q. Zhang, K. Yang, and J. Bohg, 'Concept2Robot: Learning Manipulation Concepts from Instructions and Human Demonstrations', Robotics: Science and Systems, 2020.

[11] S. Heinrich et al., 'Crossmodal Language Grounding in an Embodied Neurocognitive Model', Front. Neurorobot., vol. 14, p. 52, Oct. 2020.

[12] M. Shridhar, D. Mittal, and D. Hsu, 'INGRESS: Interactive visual grounding of referring expressions', The International Journal of Robotics Research 2020, 39(2–3), pp. 217–232.

[13] J. Hatori, Y. Kikuchi, S. Kobayashi, K. Takahashi, Y. Tsuboi, Y. Unno, W. Ko, J. Tan., 'Interactively Picking Real-World Objects with Unconstrained Spoken Language Instructions', Proceedings of International Conference on Robotics and Automation, 2018.

[14] O. Fabius and J. R. van Amersfoort, 'Variational Recurrent Auto-Encoders', arXiv:1412.6581 [cs, stat], Jun. 2015, Available: http://arxiv.org/abs/1412.6581.

[15] Kingma, Diederik P. and Ba, Jimmy, 'Adam: A method for stochastic optimization', arXiv preprint arXiv:1412.6980, 2014.

Authorized licensed use limited to: Bibliothekssystem Universitaet Hamburg. Downloaded on August 27,2021 at 10:09:52 UTC from IEEE Xplore. Restrictions apply.