In: *Proceedings of the 31st International Conference on Artificial Neural Networks (ICANN 2022)*, LNCS, Vol 13531, pp. 417-428, Bristol, UK, Sep 6–9, 2022

More Diverse Training, Better Compositionality! Evidence from Multimodal Language Learning*

Caspar Volquardsen, Jae Hee Lee, Cornelius Weber, and Stefan Wermter

Knowledge Technology, Department of Informatics, University of Hamburg
{caspar.volquardsen, jae.hee.lee, cornelius.weber,
stefan.wermter}@uni-hamburg.de
www.knowledge-technology.info

Abstract. Artificial neural networks still fall short of human-level generalization and require a very large number of training examples to succeed. Model architectures that further improve generalization capabilities are therefore still an open research question. We created a multimodal dataset from simulation for measuring the compositional generalization of neural networks in multimodal language learning. The dataset consists of sequences showing a robot arm interacting with objects on a table in a simple 3D environment, with the goal of describing the interaction. Compositional object features, multiple actions, and distracting objects pose challenges to the model. We show that an LSTM-encoder-decoder architecture jointly trained together with a vision-encoder surpasses previous performance and handles multiple visible objects. Visualization of important input dimensions shows that a model that is trained with multiple objects, but not a model trained on just one object, has learnt to ignore irrelevant objects. Furthermore we show that additional modalities in the input improve the overall performance. We conclude that the underlying training data has a significant influence on the model's capability to generalize compositionally.

Keywords: Compositional generalization · Computer Vision · Multimodality · Sequence-to-sequence · Robotics

1 Introduction

Artificial neural networks made great advances in the last decade and are state of the art for natural language processing and computer vision tasks [13]. Neural networks learn to approximate functions in high dimensional space from a set of samples from the target function. The goal is to generalize outside of the known training examples. But neural networks still fall short of human-level generalization and need a lot of training data to approximate their target function well [11]. Compositionality in language learning describes the ability to understand and produce novel combinations from known components [15]. For

* The authors acknowledge support from the German Research Foundation DFG under project CML (TRR 169) and from the BMWK under project SiDiMo.

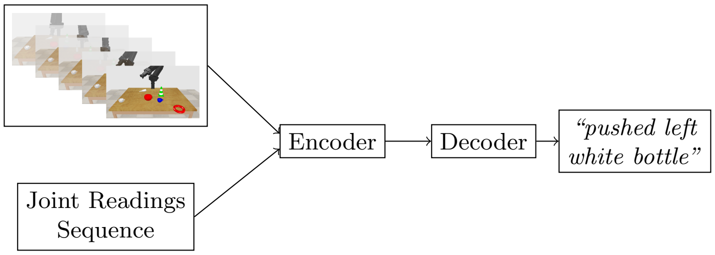


Fig. 1. The task of the model. It is tested on novel action-color-object combinations that are not part of the training data.

example, a neural model should be able to generate a novel word combination on demand, such as "red banana", if it visually perceives such an object for the first time. Recent work showed that neural networks struggle to combine known elements in a new way, even in simple cases [3, 16, 7, 14, 10, 6, 1]. These limitations can be attributed to the binding problem that describes the inability of neural networks to bind information that is distributed throughout the network and form symbol-like entities [3]. Using models that can better generalize compositionally reduces the amount of data required to handle new scenarios. It is therefore an open research question what aspects benefit neural networks' capabilities to generalize outside of the learned data distribution. We show that our neural network architecture is able to generalize compositionally, but the capabilities to do so are strongly impacted by the underlying data distribution. The model maps input video sequences, enriched with sensor data sequences, to short descriptive sentences. Figure 1 gives a brief overview of our problem setting. We created the dataset in a way, that we systematically leave out word combinations to check if our model generalizes to these new combinations.

2 Related Work

Compositional generalization is part of different fields in artificial intelligence research. Different benchmarks and customized architectures have been presented with the goal of understanding limitations and pushing the abilities of current systems.

The SCAN dataset is designed to test compositional generalization on a sequence-to-sequence task [10]. Natural language descriptions are mapped to a sequence of navigation commands. The authors used different recurrent neural network (RNN) architectures to learn the mapping of the input description to the command sequence output. They found that RNNs showed good generalization capabilities in testing when commands were arbitrarily split between train and test set, but they failed in cases that required compositionality. They also show that the generalization problem is related to the problem that RNNs learn embeddings for new verbs which are different to the representation of known verbs. Loula et al. [14] confirm these findings when they investigated other kinds of compositionality on the SCAN dataset, which envolved combining highly familiar words in new ways to create novel meaning.

The grounded SCAN dataset (gSCAN) extends the SCAN dataset by another modality [16]. The task is again to produce an output sequence of commands, but in addition to the description of the action, a two dimensional grid world with an agent and different objects is part of the input. In contrast to reinforcement tasks only the initial world state is part of the input and the complete command sequence has to be generated from that. The objects placed in the world have different sizes, shapes and colors and the input sentence describes what actions the agent should perform. The task descriptions contain relative terms like "small", where the model needs to understand the underlying concept. The authors use two encoders for the different modalities in the input, a bidirectional LSTM for the language encoding and a convolutional neural network for the image of the world state. A decoder LSTM attends to the inputs and produces the output sequence. Ruis et al. [16] report that their model failed on most compositional generalization tests.

Eisermann et al. [1] investigated the effect of the data distribution on the ability of a recurrent neural network-based architecture to generalize and do compositional generalization. For this, they created a multimodal dataset to systematically measure the ability of a model to do compositional generalization. The dataset contained sequences showing interactions of a robotic arm with various objects on a table and a descriptive sentence as label. Their results showed some significant factors of the dataset for the model to generalize well. They showed that more diverse training data with more overlap of attributes improved the generalization performance significantly. Leaving away additional sensory data and only relying on vision data led to worse results not only of the generalization performance but also of the training performance. A noticeable problem of the model was the introduction of a distractor object on the table, which means a second object which is not part of the action and thus not part of the label sentence. In this setting even the accuracy on the training data was poor and significantly worse than with one visible object. This implies that the underlying model architecture was not able to process the visual input in a sufficient way, because it could not separate the features of the two objects, as is evident in the binding problem. This limitation was the motivation for our work, with the goal to overcome this problem and further improve the compositional generalization with a different model architecture.

The EMIL dataset is a quite similar dataset, which consists of recordings of real interactions of a humanoid robot [5]. The robot interacts with objects on a table in front of him in a child-like manner, performing actions which an external teacher describes. The recordings contain image data from the robot's two cameras, auditory data recorded by the robot's microphones and additionally sensorimotor data describing the different joints of the robot. This dataset is comparably small with 240 sequences. Heinrich et al. [6] use different continuous time RNN-based architectures which take multimodal sensory input and map it to language. They report that all models struggle on generalization tasks. They

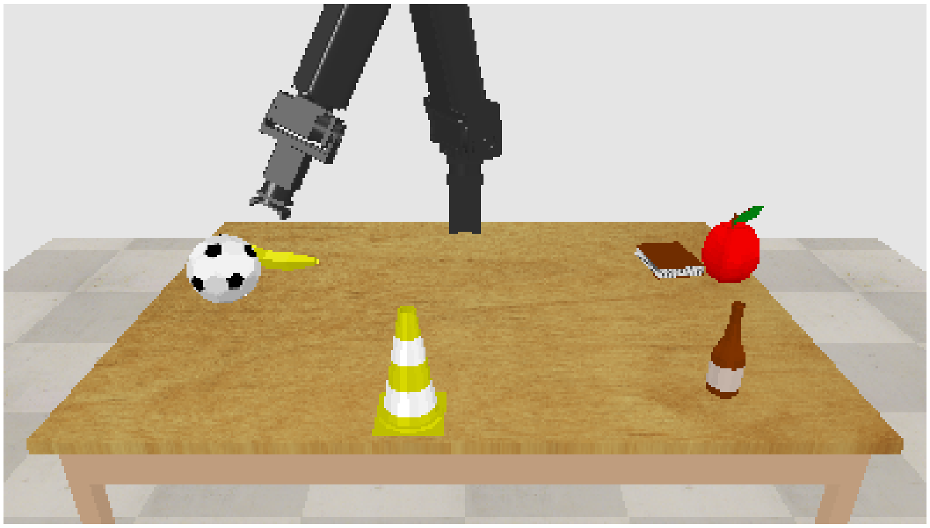


Fig. 2. Example frame with six visible objects. After putting down the banana it is partially occluded.

also find that their models tend to rely on a single modality when the training dataset is smaller, but with a larger training set, the model benefits from the additional modalities.

3 Multimodal Dataset

Based on the work of Eisermann et al. [1] we created a dataset and expanded it to more complex scenes. It consists of sequences of varying length that show video data capturing a robot arm behind a table, which interacts with objects on the table and corresponding sensory data. See figure 2 for an example frame of a sequence. There are between one and six objects on the table depending on experimental condition. The camera capturing the scene is placed in front of the table looking down on it towards the robot arm, also capturing white background and some space to the left and to the right of the table. A scene captures one of four different actions, which the robot arm performs with exactly one object. In addition to the video data, for each frame, the corresponding joint angles of the robot arm are part of the data. The dataset was generated using the robot simulation software CoppeliaSim (www.coppeliarobotics.com). The sequences consist on average of 20 frames, where each frame is the combination of a 224×398 8bit RGB pixel image and six 32bit floating point numbers representing the sensor readings of the six joint positions of the robot arm. Each sequence has a corresponding descriptive sentence of three words, describing the interaction of the robot arm. The four different actions are *"pushed right"*, *"pushed left"*, *"picked up"* and *"put down"*. For the action *"push left"* the robot arm moves to the

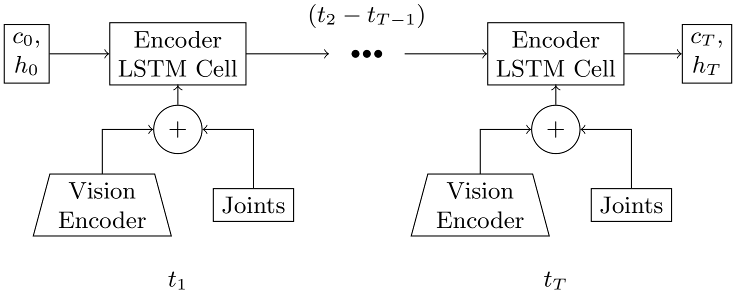


Fig. 3. Encoder architecture

object, places the gripper (front end of the robot arm) to the right of the object, and pushes the object to the left. If the object lays near the left end of the table the object can get pushed off the table which is also partly captured by the camera. *"pushed right"* happens vice versa. The action *"picked up"* captures the robot arm moving to the object placing the gripper on the object, gripping it and moving it up. In the air the capturing of the action *"put down"* begins where the robot arm moves the object in the air to a random location over the table and lays it down. When multiple objects are visible simultaneously on the table the collision of the objects with each other and the gripper are all simulated. It can therefore happen that the path that the gripper pushes an object collides with another object leading to multiple objects being moved. In these cases it is necessary to consider the relative positioning of the objects towards the gripper and the entire sequence to describe it correctly. Another challenging scenario of multiple objects is occlusion, or partial occlusion of objects in parts of the sequence. In this case the model needs to recognize the object in parts where it is visible and compose this information. These difficulties happen at a higher frequency with increasing number of visible objects.

4 Model

The task of our model is to find the correct description given the input sequence. We use an LSTM encoder-decoder architecture similar to the architecture Sutskever et al. used for language translation [19], because it can naturally deal with sequences of varying length. One LSTM is used to encode the input sequence one time step after another into a fixed size vector. This vector is used as the hidden state input to the decoder LSTM which produces the output sentence token by token. Figure 3 illustrates our encoder architecture. At each time step t_i the input to the encoder is the image data and the joint positions. Similar to Eisermann et al. [1] we use a convolutional neural network as vision encoder to preprocess the high dimensional image data and encode it to a lower dimensional feature vector [12, 2, 13, 9]. We modified a ResNet18 network to fit to our image dimensions, since it was designed for the 224×224 pixel images

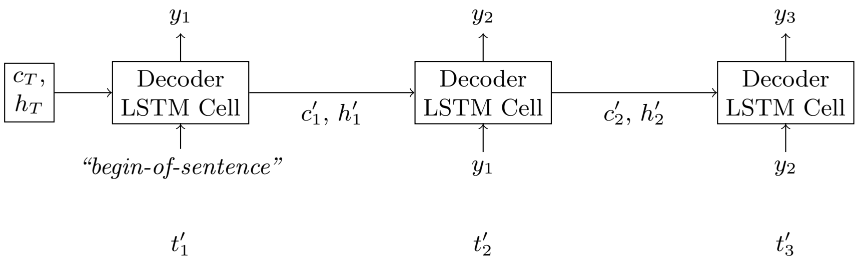


Fig. 4. Decoder architecture

of the ImageNet challenge [4, 17].After the convolutional layers of the ResNet we use a fully connected layer which outputs a vector of dimension d_{image}. The encoded image vector concatenated with the 6-dimensional vector of the joint positions is the input to the encoder LSTM at each time step. At time t_0 the initial cell state and hidden state vectors are zero vectors and the resulting cell state and hidden state after t_i gets passed to the next time step t_{i+1}. After all inputs at time step t_T the resulting cell and hidden state c_T and h_T contain the encoded information of the input sequence and form the input to the decoder.

The decoder takes the cell state and hidden state vectors from the encoder and gets a *"begin-of-sequence"*-token as input for the first output generation. Each token is a 19-dimensional vector, where we use one-hot encoding of all possible words which are 4 actions, 6 colors and 9 objects. The begin-of-sequence vector is an zero vector. The tokens get generated through a linear transformation applied on the LSTM output of dimension d_{hidden} which maps to the output dimension 19. A softmax function transforms the output to a probability distribution over the 19 dimensions. After the first output at t'_1 the generated output y_1 is the input to the decoder at t'_2. The same is applied to t'_3. Note that we do not require an *"end-of-sequence"*-token for the decoder to stop [19]. Since all output sentences have the same length of three tokens we simply take the first three outputs. This change could however be made if varying output lengths were required.

5 Training Setup

For our experiments we created several different training and validation datasets in a systematic way. We altered four parameters for the dataset composition and generated the datasets accordingly. We trained our model the same way on all datasets and evaluated the model on a constant test set and a compositional generalization test set. For comparability of our results we took the same action-color-object combinations for the constant test sets as Eisermann et al. [1] and organized the training and validation datasets in a similar way. The constant test set contains the four different combinations *"pushed right white football"*, *"pushed right yellow banana"*, *"pushed left brown bottle"* and *"pushed left red* *ring"*. All these action-color-object combinations were part of every training and validation set and only varied in the randomised positions on the table of the objects and robot arm. The compositional generalization test set on the other hand contained only action-color-object combinations which were excluded from the training and validation datasets. The scenes contained are *"pushed left white football"*, *"pushed left yellow banana"*, *"pushed right brown bottle"* and *"pushed right red ring"*, which are the opposite pushing directions than in the constant test set. The training and validation datasets were generated according to the following parameters:

V1, V2, V6: The given number denotes the number of simultaneously visible objects. This affects also the constant test and compositional generalization test set, where as many objects are visible as in the corresponding training and validation set.

C1, C6: The number of different colors each object can randomly appear in. **C1** means that each object always appears in the same color. The six possible colors are *red*, *green*, *blue*, *white*, *brown*, and *yellow*.

O4, O9: The number of object types that can show up in a scene. **O4** shows the same objects as in the constant test set, which are a*football*, *banana*, *bottle*, and *ring*. **O9** shows additional five object types, which are not part of the test sets.

X, ¬X: This parameter controls whether the colors of the objects in the test sets are exclusive to them. Exclusive meaning that no other object appears in the same color as the objects in the test set. For example, only the banana would appear in yellow in the training and validation set if the colors were exclusive.

J, ¬J: Specifies whether the positions of the robot arm joints are part of the input sequence. If not, the model needs to generate a descriptive sentence solely relying on the visual input.

Each training set contained 5000 samples and the validation set 2500. We trained the model on each dataset for 20 epochs and evaluated the model after each epoch on the validation dataset. Here we calculated the word-wise accuracy on the validation dataset and saved the model parameter corresponding to the epoch with the highest accuracy, as a kind of early stopping procedure [2]. Word-wise accuracy means the percentage of correctly generated words in the output sentence. At each epoch we trained the model in mini-batches of size 16 with the Adam optimizer [8]. We calculated the loss for each output using the cross-entropy loss function and back-propagate through time, also jointly updating the parameter of the vision encoder. At the start of training we initialised the weights of the ResNet18 convolutional layers with weights pretrained on the ImageNet dataset to speed up the learning process [17]. For our experiments we used a hidden dimension for c_i and h_i of $d_{hidden} = 512$ and the dimension for the encoded images of $d_{image} = 256$. We found these dimensions to perform the best in a prior hyperparameter search.[1]

[1] The source code for the model and the data generation can be found at this link: https://github.com/Casparvolquardsen/Compositional-Generalization-in-Multimodal-Language-Learning

Table 1. Sentence-wise accuracy of our model for the different training datasets in percent. The abbreviation "Comp. Gen." stands for the compositional generalization test set.

		4 Objects (**O4**)		9 Objects (**O9**)	
		Exclusive Colors (**X**)		Color Overlap (¬**X**)	
		1 Color (**C1**)	6 Colors (**C6**)	1 Color (**C1**)	6 Colors (**C6**)
With Joint Readings (**J**)					
1 Visible Object (**V1**)	Training	93.3	99.5	99.9	99.9
	Constant Test	99.9	100.0	99.9	100.0
	Comp. Gen.	**1.25**	**56.7**	**64.4**	**99.4**
2 Visible Object (**V2**)	Training	99.1	99.8	99.0	99.2
	Constant Test	97.3	98.8	96.7	92.6
	Comp. Gen.	**7.0**	**44.6**	**38.6**	**87.6**
6 Visible Object (**V6**)	Training	100.0	99.7	99.8	99.2
	Constant Test	98.6	96.5	95.35	87.0
	Comp. Gen.	**7.2**	**48.1**	**42.4**	**76.5**
Without Joint Readings (¬**J**)					
1 Visible Object (**V1**)	Training	96.7	99.6	99.9	100.0
	Constant Test	100.0	99.9	99.9	100.0
	Comp. Gen.	**0.0**	**54.0**	**35.1**	**99.5**
2 Visible Object (**V2**)	Training	98.9	99.6	99.0	99.4
	Constant Test	98.6	98.8	94.3	88.5
	Comp. Gen.	**5.5**	**40.1**	**33.9**	**78.4**
6 Visible Object (**V6**)	Training	99.9	99.8	99.9	99.4
	Constant Test	97.4	96.5	95.1	80.6
	Comp. Gen.	**7.9**	**42.0**	**34.0**	**68.6**
Num. different sentences		12	44	32	212
Samples per sentence		416	113	156	23

6 Results

Table 1 shows the results of our experiments. We report the sentence-wise accuracy which is the percentage of label sentences which were generated completely correct. This metric is more informative than the word-wise accuracy regarding the compositional generalization capabilities, because the sentences in the compositional generalization set differ only in one word to sentences contained in the training set and so a word-wise accuracy would be misleadingly high when the model does not generalize compositionally. We summarize our findings as follows:

Distractor objects can be handled: Compared to the model of Eisermann et al. [1] our model achieved better training and constant test accuracy for every dataset. On the datasets with joint readings our model achieved on average 43.5% better constant test results [1]. Our model especially overcame the problem of significant worse performance with two visible objects pointed out by Eisermann et al. [1]. Our average constant test performance with two visible objects and joint readings was 96.4% compared to 56.1% [1]. We therefore tested our model also on the more complex scenarios with six visible objects and found only a minor drop in performance with still better test results than the previous model on simpler datasets [1].

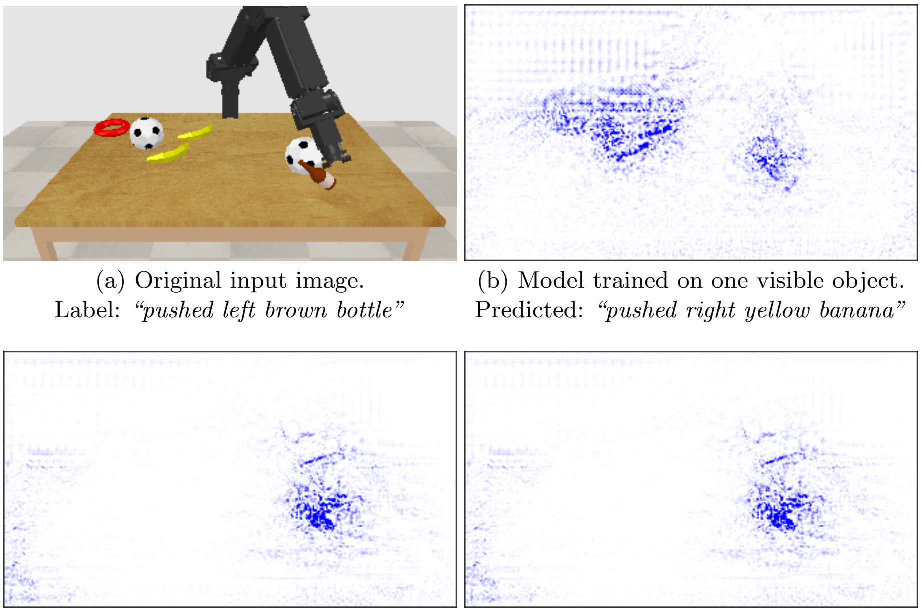
Successful compositional generalization: We achieve over 99% compositional generalization test accuracy for the V1-C6-O9 dataset. Given here one visible object (V1) and the largest diversity in colors (C6) and object types (O9), the compositional generalization performance comes close to the training and constant test performance (see rightmost column in table 1). Our model also surpasses the compositional generalization capabilities of Eisermann et al. [1] in nearly all cases, the only exception being the V1-C6-O4-X-J data set, which is challenging in its small number of object types and exclusive colors.

More diverse training set increases compositional generalization: We find that both showing each object in more different colors (C1 to C6) and increasing the number of different objects that are shown (O4 to O9) benefit the models' capability to generalize compositionally, confirming the findings of Eisermann et al. [1]. For only one visible object (V1) we achieve the highest compositional generalization capabilities with an accuracy of over 99%. For multiple visible objects on the C6-O9 datasets the compositional generalization accuracies are still between 68.6% and 87.6%. A more detailed analysis of the results shows that the datasets V1-C1-O4-X with the lowest sentence-wise compositional generalization performance, have a much higher word-wise accuracy of 66.6%, where in most cases the model correctly named color and object, but not the correct action.

More diverse training set decreases constant test accuracy: The highest constant test accuracies are achieved in the least diverse datasets (C1-O4) where the training set only contains sequences of 12 different sentences, where four of them are inside the constant test set. We found the worst constant test accuracies for the most diverse datasets (C6-O9) where 212 different sentences were part of the training set. Because we use the same number of training samples for each experiment, the number of samples per word combination is lower in these cases and therefore also the number of samples of the sentences contained in the constant test set (see in the bottom two rows of the table 1).

Removing a sensory modality decreases overall performance: We find that leaving away the joint readings as input to the model decreased the overall performance slightly. In these cases the model had to rely solely on the vision input with no additional modality. The joint readings in theory are able on their own to specify which action was performed by the robot arm. We found that leaving the joint readings away decreases the constant test accuracy on average by only 1% and the compositional generalization accuracy by 13%. Therefore we confirm the finding of Eisermann et al. [1], but show a lower influence on the performance of our model.

Models trained on one visible object fail on multiple objects: Evaluating the models also on the constant test sets with a different number of visible objects as in the training set shows that models which were trained on V2 or V6 training datasets also generalize well to samples with 1-6 visible objects. On



(a) Original input image.
Label: *"pushed left brown bottle"*

(b) Model trained on one visible object.
Predicted: *"pushed right yellow banana"*

(c) Model trained on two visible objects.
Predicted: *"pushed left brown bottle"*

(d) Model trained on six visible objects.
Predicted: *"pushed left brown bottle"*

Fig. 5. An example frame input of the constant test set with six visible objects. (b), (c), and (d) show input feature importance from the attribution method integrated gradients for models trained on different datasets, given input (a). The model used for (b) fails to focus and predicts a wrong sentence.

the other hand, models which were trained on the V1 datasets generalize poorly to more visible objects and achieve a sentence-wise accuracy of below 55% in all cases when tested on constant test sets with 2-6 visible objects. We further analysed the functioning of the models using the integrated gradients attribution method [18]. Integrated Gradients is an axiomatic model interpretability algorithm that assigns an importance score to each input feature by approximating the integral of gradients of the model's output with respect to the inputs. We find that models trained on multiple visible objects form their color and object output mostly based on the input pixels near the robot arm gripper and do not consider the rest of the image. Contrarily models trained on only one visible object also consider input pixels at distractor objects for their prediction leading to wrong predictions. Figure 5 shows an example frame and the corresponding visualization of the integrated gradients method for three different models.

7 Discussion

Our model achieves over 99% compositional generalization performance in the best condition, compared to 65.62% of the model by Eisermann et al. [1]. We hypothesize that our encoder-decoder design forces the model to encode the information contained in the sequence in an organized way which benefits compositional generalization. Especially the increasing performance with more diverse training sets supports this hypothesis, because the relatively small hidden vector is less able to store the information for each word combination separately, but needs a common way of encoding the information. Another cause for improvement could be our ResNet vision encoder. Both our higher encoding dimension and the fact that we train it jointly with the rest of the network, including the convolutional kernels, may contribute to the improved performance.

The compositional generalization test set does not contain any unseen color-object combinations. For example, while *"pushed left yellow banana"* is unseen, the yellow banana was part of the training set together with other actions. Future research could extend the dataset to also investigate to what degree models are able to decompose the shape and color attributes of the objects.

Our results show that the model does not separate the features action type, color or object automatically, but only with sufficient variability in the training data. The lacking compositional generalization for the least diverse datasets indicates that input features are not separated compositionally in those cases, so the naming of the action is not only based on the motion of the robot arm, but influenced by the shape and color of the object. Such a phenomenon was also found in other experiments [10, 14, 16, 1] and indicates a need for neural network architectures that use object-centric encoding [3]. It remains for future research to find generic architectures that further improve compositional generalization.

8 Conclusion

In summary, we created a dataset to investigate compositional generalization in multimodal sequences of a 3D environment. We showed a model, which can handle complex scenes with multiple visible objects, and is able to generalize compositionally with sufficient training data. An analysis with the integrated gradients method shows, that a model trained with only one visible object fails to focus if multiple objects are shown. In contrast, a model trained on two or six visible objects generalizes to arbitrary numbers of objects. We found remaining limitations of the model to systematically generalize compositionally, when trained on less diverse data, which confirms previous findings [10, 14, 16, 1], while additional input modalities improve the model's generalization capabilities. The results provide guidance for model architecture design and training data selection that promise good generalization to unseen data.

References

1. Eisermann, A., Lee, J.H., Weber, C., Wermter, S.: Generalization in multimodal language learning from simulation. In: Proceedings of the International Joint Conference on Neural Networks (IJCNN 2021) (Jul 2021)

2. Goodfellow, I., Bengio, Y., Courville, A.: Deep learning. Adaptive Computation and Machine Learning, MIT Press (2016)
3. Greff, K., van Steenkiste, S., Schmidhuber, J.: On the binding problem in artificial neural networks. arXiv 2012.05208 (Dec 2020)
4. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. arXiv 1512.03385 (2015)
5. Heinrich, S., Kerzel, M., Strahl, E., Wermter, S.: Embodied multi-modal interaction in language learning: the EMIL data collection. In: Proceedings of the ICDL-EpiRob Workshop on Active Vision, Attention, and Learning (ICDL-Epirob 2018 AVAL). Tokyo, Japan (Sep 2018)
6. Heinrich, S., Yao, Y., Hinz, T., Liu, Z., Hummel, T., Kerzel, M., Weber, C., Wermter, S.: Crossmodal language grounding in an embodied neurocognitive model. Frontiers in Neurorobotics (Oct 2020)
7. Keysers, D., Schärli, N., Scales, N., Buisman, H., Furrer, D., Kashubin, S., Momchev, N., Sinopalnikov, D., Stafiniak, L., Tihon, T., Tsarkov, D., Wang, X., van Zee, M., Bousquet, O.: Measuring compositional generalization: A comprehensive method on realistic data. arXiv 1912.09713 (2019)
8. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv 1412.6980 (2017)
9. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: Pereira, F., Burges, C.J.C., Bottou, L., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems. vol. 25. Curran Associates, Inc. (2012)
10. Lake, B.M., Baroni, M.: Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. arXiv 1711.00350 (2017)
11. Lake, B.M., Ullman, T.D., Tenenbaum, J.B., Gershman, S.J.: Building machines that learn and think like people. arXiv 1604.00289 (2016)
12. LeCun, Y.: Generalization and network design strategies. Technical Report CRG-TR-89-4, University of Toronto (1989)
13. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**, 436–44 (05 2015)
14. Loula, J., Baroni, M., Lake, B.M.: Rearranging the familiar: Testing compositional generalization in recurrent networks. arXiv 1807.07545 (2018)
15. Montague, R.: Universal Grammar, vol. 36. Blackwell Publishing Ltd (1970)
16. Ruis, L., Andreas, J., Baroni, M., Bouchacourt, D., Lake, B.M.: A benchmark for systematic generalization in grounded language understanding. arXiv 2003.05161 (2020)
17. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: ImageNet large scale visual recognition challenge. arXiv 1409.0575 (2014)
18. Sundararajan, M., Taly, A., Yan, Q.: Axiomatic attribution for deep networks. arXiv 1703.01365 (2017)
19. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. arXiv 1409.3215 (Dec 2014)