Analyzing structural priors in multi-agent communication*

Abhinav Gupta*
MILA
abhinavg@nyu.edu

Agnieszka Słowik*
University of Cambridge
agnieszka.slowik@cl.cam.ac.uk

William L. Hamilton
McGill University
MILA
wlh@cs.mcgill.ca

Mateja Jamnik
University of Cambridge
mateja.jamnik@cl.cam.ac.uk

Sean B. Holden
University of Cambridge
sbh11@cam.ac.uk

Christopher Pal
Polytechnique Montréal
MILA
christopher.pal@polymtl.ca

ABSTRACT

Human language and thought are characterized by the ability to systematically generate a potentially infinite number of complex structures (e.g., sentences) from a finite set of familiar components (e.g., words). Recent works in emergent communication have discussed the propensity of artificial agents to develop a systematically compositional language through playing co-operative referential games. The degree of structure in the input data was found to affect the compositionality of the emerged communication protocols. Thus, we explore various structural priors in multi-agent communication and propose a novel graph referential game. We compare the effect of structural inductive bias (bag-of-words, sequences and graphs) on the emergence of compositional understanding of the input concepts measured by topographic similarity and generalization to unseen combinations of familiar properties. We empirically show that graph neural networks induce a better compositional language prior and a stronger generalization to out-of-domain data. We further perform ablation studies that show the robustness of the emerged protocol in graph referential games.

KEYWORDS

multi-agent communication; graph representation; emergent languages

1 INTRODUCTION AND RELATED WORK

Human communication and reasoning are characterized by the ability to systematically generate a potentially infinite number of complex structures (e.g., sentences) from a finite set of familiar components (e.g., words). For instance, if a person knows the meaning of utterances such as 'red circle' and 'blue square', she can easily understand the utterance 'red square' even if she has not encountered this particular combination of shape and color in the past. This type of generalization capacity is referred to as *compositionality* [1, 3, 22] or *systematic generalization* [2].

The ability of artificial agents to develop a compositional language has been investigated through *emergent communication* studies. Motivated by the assumption that human language derives meaning from its use [27], the agents are left to develop a communication protocol from scratch based on solving a shared task. To this end, the agents learn to communicate in end-to-end virtual environments such as referential games. A referential game (Figure 1) consists of a perceptual input, agents, communication channel (discrete symbols without any predefined meaning) and an action to be rewarded (e.g., distinguishing the target input among distractors). This perspective on learning to communicate mitigates some of the issues observed in supervised training of language models such as sample inefficiency and exploiting superficial statistical signals [3, 11]. Treating communication as an interactive, goal-driven multi-agent learning problem rather than a static supervised learning task is more intuitive and natural.

Communication success in a referential game is known to be insufficient to guarantee emergence of a compositional language [10]. Several factors that influence compostionality in emergent communication games have been investigated of late, such as the degree of structure in the input data using the examples of images and sequences [12], periodical resetting of one of the agents [14], and constraints related to the capacity and bandwidth of the agents [20]. However, graph representation learning has not been explored in emergent communication. Sequential inputs used in existing work imposes artificial order on independent properties (such as shape and color; Figure 1). Relational and modular entities that can be naturally represented as graphs provide a more challenging and realistic grounding for the communication protocol.

In this work, we investigate the effect of varying the representation of the perceptual input and the corresponding representation learning methods on the systematic generalization, task success and compositionality of the emerged language. In the existing work on emergent communication, sequences and bag-of-words are most commonly used as input data [4, 12]. Given the recent resurgence of explicit structural bias in neural networks (e.g., *graph neural networks*), we propose two graph-based referential games of a varying degree of complexity. We evaluate the hypothesis that graph representations encourage compositionality and generalization. The study of the effect of data structures on compositional understanding reflected in the language is an important step towards learning to communicate in more complex and realistic environments.

Our contributions are as follows:

- We propose two graph referential games. In the first game, we focus on learning the hierarchy of *concepts* (e.g., red square), *properties of a concept* (e.g., color) and *property types* (e.g., red) represented using a tree. The second game implements the *graph isomorphism* test for arbitrary graphs in the multi-agent environment with a communication channel.

*The first two authors contributed equally.

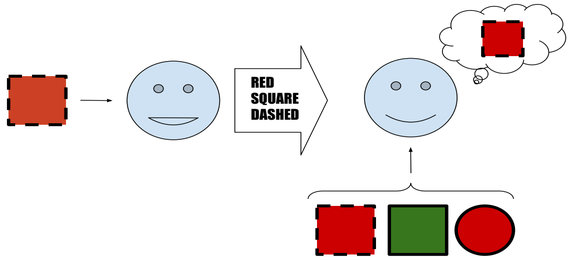

Figure 1: Referential game. The first agent (the *speaker*) must communicate the target object to the second agent (the *listener*). Based on the received message, the listener learns to recognize the target object among a set of distractors. A training sample in this game consists of the target and a set of distractors, all of which come from the same distribution and are sampled without replacement.

- We provide baseline agents for the graph referential games based on Graph Convolutional Networks (GCNs) [9] and GraphSAGE [6] in a graph-to-sequence model. We compare these models with sequence-to-sequence [24] and bag-of-words networks in terms of task success, generalization to unseen combinations of familiar concepts (e.g., if a model trained using the 'red square' and 'blue circle' examples can understand the meaning of a 'red circle') and compositionality of the emerged language. Our results show that graph representation learning methods induce a better compositionality prior and achieve stronger generalization to out-of-domain data.
- We verify that the agents' ability to solve the task is grounded in the communication protocol they develop. Our analysis shows that a change in the message results in a selection of a different object, i.e. the listener relies on the message sent by the speaker to correctly identify the target among distractors. We also investigate whether the agents learn an order invariant representation of objects by shuffling the distractors and the target. We posit that the communication channel is robust with respect to the changes in the ordering of the target and the distractors.

2 ENVIRONMENTAL SETUP

2.1 Multi-Agent Referential Games

Referential games that we study in this paper are variants of the Lewis Signaling Game [13], which has been extensively used in emergent communication.

A Lewis Signaling Game involves two agents of fixed roles: the *speaker* and the *listener*. The speaker has access to some information (the *target*) which the listener cannot directly observe. The speaker then sends a message describing the target and the listener acts upon receiving the message by deducing the target based on the message. Referential games, in particular, are composed of the listener receiving the message and a set of entities consisting of the target and distractors. In each such set, the target and distractors are sampled without replacement from the same distribution.

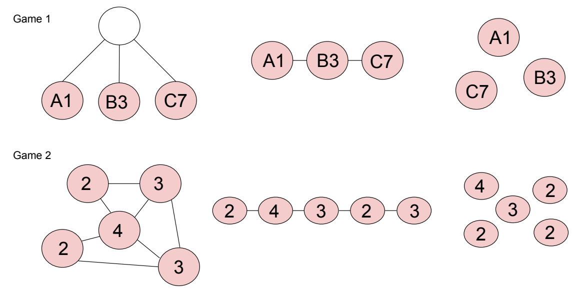

Figure 2: Input representations in Game-1 and Game-2. In Game-1, the hierarchy of an object and its properties is represented as a tree of a fixed topology. Each child node encodes the property value (e.g. property: A, property value: 1) in a concatenation of the one-hot vectors representing the property and the property value, respectively. In baseline representations, we remove the graph topology and use 1) a sequence of a randomly ordered one-hot vectors 2) bag-of-words representation with no structural prior. In Game-2, graph topology varies since the edges are randomly assigned.

Formally, the game is played as follows. The target t_i is observed by the speaker which then transmits a message m to the listener. In turn, the listener selects a response r_k. For each target t_i there is one correct response r_i, i.e. distinguishing the target among the distractors. The agents are rewarded with a payoff u if and only if the listener identifies the target:

$$u = \begin{cases} 1 & \text{if } k = i, \\ 0 & \text{otherwise.} \end{cases}$$

Note that the payoff does not depend on the message sent, and so the agents are rewarded for solving the task and not for the emerged communication protocol.

This framework can be also considered as a co-operative partially-observable Markov game [15]. The target and the distractors can be interpreted as permutations of combinatorial properties such as shape and color (Figure 1). They can be represented as symbolic vectors, images, graphs or any efficient data structure able to encode complex entities.

2.2 Graph Referential Games

We introduce two graph referential games. In Game-1, we use trees to represent a hierarchy of concepts (e.g. red square) composed of different properties (e.g. color, shape). In Game-2, we implement graph isomorphism test for arbitrary graphs. Figure 2 shows the input data to our graph referential games. In each game, we compare the graph representation to a corresponding sequence and bag-of-words encoding in terms of task success, out-of-domain generalization and language compositionality. In each game, a *sample* consists of a target graph and the set of K distractors. We obtain a collection of these samples and create the train, validation and test splits (60%/20%/20%).

Game-1: hierarchy of concepts and properties. In this game, we construct a tree from a vector of *perceptual dimensions* $[p_1, p_2, \ldots, p_n]$ where n corresponds to the number of properties and $p_1, p_2, \ldots, p_n$ denote the number of possible types per property. Each tree has the same number of properties n and they only differ in the property values. Formally, each tree is an instance of a graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ where $\mathcal{V}$ corresponds to the set of all nodes representing unique properties, and a 'central' node such that $|\mathcal{V}| = n + 1$. The set $\mathcal{E}$ comprises of undirected edges that connect two nodes with a relation. $\mathcal{E}$ consists of the edges between the central node and its children, that represent individual properties, such that $|\mathcal{E}| = n$. All the nodes except the central node are represented using node features. The node features consist of a concatenation of the property encoding and the type encoding (represented as one-hot vectors). The central node is encoded as an empty node and no edge features are used. For the purpose of providing a simple graph baseline, we use one concept (the root) and one level of properties (successors of the root). This representation can be easily extended to a deeper hierarchy of properties and sub-properties.

Game-2: graph isomorphism test. We introduce Game-2 to investigate whether the agents can learn graph topology. Since Game-1 comprises trees of a fixed structure that can be directly represented as sequences or bag-of-words, we want to evaluate the graph encoders using arbitrary graphs of a varying structure. Such graphs can be used to represent relations between arbitrary entities, e.g. connections between users of a social media platform. Game-2 defines a more realistic and flexible framework for multi-agent communication on graphs. In Game-2, each graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ is defined over the number of nodes N, $|\mathcal{V}| = N$, and the set of undirected edges $\mathcal{E}$, such that $|\mathcal{E}| \in [N-1, \frac{N(N-1)}{2}]$ (see Figure 2). For a graph of N nodes, the total number of edges is sampled from the $[N-1, \frac{N(N-1)}{2}]$ interval, where $\frac{N(N-1)}{2}$ is the number of edges in a complete graph. In a given instance of the game, N is fixed for all targets and distractors. We create a directed edge (i, j) where $i, j \in \{1, 2, \ldots, N\}$ and $i \neq j$ by sampling the source i and the destination j from $\mathcal{V}$. We then add the corresponding (j, i) edges to the graph to make it undirected. We add a self-loop to each node to include its own features in the node representation aggregated through message passing. We use node degrees converted to one-hot vectors as the initial node features.

In both games, we use two baseline representations— sequences (Seq) and bag-of-words (BoW). These data representations are constructed similarly as in the existing work on emergent communication [4, 10, 12]. A sequence consists of randomly ordered features, which in our games are equivalent to the node features in the graph representations. Bag-of-words representation does not impose any order on the features (see Figure 2).

2.3 Training and Models

In order to propagate the gradients through a non-differentiable communication channel, we train the games using a Gumbel-Softmax trick. The speaker produces a softmax distribution over the vocabulary V, where V refers to the finite set of all distinct words that can be used in the sequence generated by the speaker. Similar to [18, 23], we use the 'straight through' version of Gumbel-Softmax [7, 17] during training to make the message discrete. At test time, we take the `argmax` over the whole vocabulary.

In our graph referential games, the listener receives the discretized message m sent by the speaker along with the set of distractors K and the target graph d^*. The listener then outputs a softmax distribution over the $|K| + 1$ embeddings representing each graph. The speaker f_θ and the listener g_ϕ are parameterized using graph neural networks. We formally define it as follows:

$$m(d^*) = \texttt{Gumbel-Softmax}(f_\theta(d^*))$$

$$o(m, \{K, d^*\}) = g_\phi(m, \{K, d^*\})$$

In order to handle raw graph input, the speaker and the listener are parameterized using a graph encoder. The speaker additionally uses a sequence decoder to generate a message. Following the conventional encoder-decoder architecture, the graph encoder first generates node embeddings for each node, and then it uses them to construct an embedding of the entire graph. The sequence decoder takes the graph embedding as input and generates a message. A graph encoder consists of the node representation learning method (e.g. Graph Convolutional Network (GCN) [9]) and a graph pooling method. Node representations are computed for each node v_i through neighborhood aggregation that follows the general formula of:

$$h_{v_i}^{(l+1)} = \texttt{ReLU}\left(\sum_{j \in N_i} h_{v_j}^{(l)} W^{(l)}\right)$$

where l corresponds to the layer index, h_{v_i} are the features of the node v_i, W refers to the weight matrix, and N_i denotes the neighborhood of the node v_i. In this paper, we use a standard GCN architecture as well as GraphSAGE [6], an extension of GCN which allows modifying the trainable aggregation function beyond a simple convolution. GraphSAGE learns aggregator functions that can induce the embedding of a new node given its features and the neigborhood, without re-training on the entire graph. The GraphSAGE encoders are thus able to learn dynamic graphs. In this paper, we experiment with commonly used 'mean', 'pool' and 'gcn' aggregator types. In order to compute the graph embedding, we use simple graph pooling methods. A graph embedding is obtained through a linear transformation of the node features using a mean or sum. A graph embedding vector in our graph-to-sequence implementation of the speaker corresponds to the context vector in the existing sequence-to-sequence implementations.

Similarly as in sequence-to-sequence architectures, the sequence decoder in graph-to-sequence outputs a probability distribution over the whole vocabulary for a fixed message length which is then discretized to produce the message.

3 EXPERIMENTS & ANALYSIS

We present the comparative analysis of the different data representations presented in this paper. In this paper, we do not aim to propose the best model to encode a given data representation but instead perform a comparative analysis of models with varying levels of structural inductive bias on both graph referential games. In particular, we aim to answer the following questions:

- Which model provides the best inductive bias for generalization? Does the complexity of the game affect generalization across different models?
- How does the compositionality of the learned protocols differ between models? What structural priors are beneficial for agents to learn a compositional protocol?

Furthermore, we perform ablation studies on graph based representations and analyze different graph neural network architectures on the basis of generalization and compositionality. We used the Deep Graph Library [26] when using graphs and EGG [8] for building the framework while the whole codebase was written using PyTorch [19].

3.1 Generalization and Inductive Bias

	Game type	BoW	Seq	Graph
Game-1	[10, 6, 8]	99.2 ± 1.1	97.5 ± 1.5	99.0 ± 0.6
	[10, 6, 8, 8, 10]	98.1 ± 0.5	95.4 ± 2.7	98.8 ± 0.13
Game-2	15 nodes	95.1 ± 0.59	91.3 ± 0.38	95.4 ± 0.57
	20 nodes	94.3 ± 0.85	89.7 ± 1.28	95.3 ± 1.36
	25 nodes	93.8 ± 0.22	89.5 ± 1.3	94.9 ± 0.95

Table 1: Test accuracies on both games. For Game-1, the Game type refers to the perceptual dimensions used while in Game-2, it refers to number of nodes used.

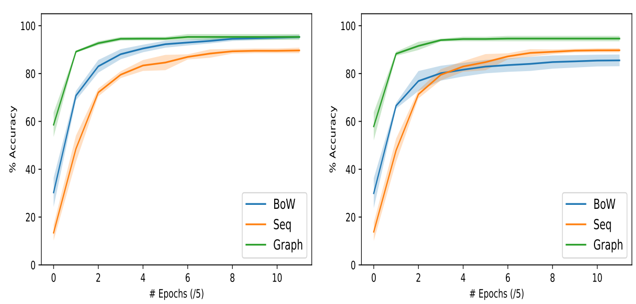

Figure 3: Game-1 (left) and 2 (right) test accuracies. Both plots show the accuracy on the test dataset which is the set of unseen combinations of distractors and target. For Game-1, we used a perceptual dimension of [10, 6, 8, 8, 10] and the speaker sends a fixed length (5) message of vocabulary size of 10. For Game-2, we used 20 nodes and the message consists of 20 words each of vocabulary size 20. Both games use 9 distractors. All runs are averaged over 3 seeds.

For both of our games, we measure generalization using test accuracy on a set of unseen combinations of targets and distractors. While the agents have seen the target with some other combination of distractors in the training set, they haven't seen the 'test' combinations during training. In Table 1 we show the test accuracies for both games where the speaker observes a target and sends a unique code to the listener such that it is able to distinguish the target when seen in combination with other distractors. This generalization to unobserved pairs of (target, distractors) means that both the speaker and the listener learned to 1) extract relevant properties from the data (no matter how they are represented as input) 2) create a unique code-book to refer to these essential properties which are used to identify the target.

When comparing the data representations, we see that a graph based neural network consistently outperforms the bag-of-words and sequence based networks. An interesting point to note here is that the bag-of-words model easily overfit to the training data and thus performed the worst during test time. On the other hand, the graph based model did not give the best training accuracy but performed the best during test time.

3.2 Compositionality and Inductive Bias

We use two measures of compositionality, namely topographic similarity and out-of-domain generalization, and study the effect of varying structural priors on the input representation.

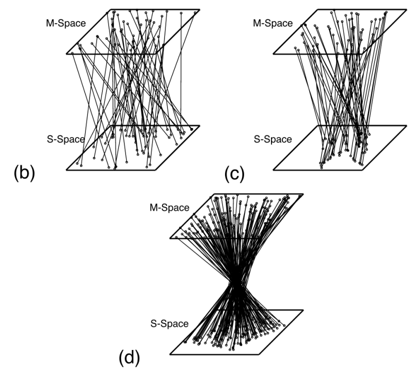

Figure 4: Intuition behind topographic similarity. In our analysis, M-Space corresponds to the input space (e.g. graphs) and S-Space refers to the message space. In the example (b) the emerged language is random: the neighbors surrounding a point in the meaning space suggest nothing about the location of the corresponding message, and so the topographic similarity will be low. Examples (c) and (d) show languages of high topographic similarity, as objects that are close in the meaning space are mapped to similar signals. Image source: [5].

Topographic similarity. We use topographic similarity as one measure for compositionality, following common practice in this domain of referential games [12, 14]. We take the complete dataset and compute the cosine distances between all possible pairs of the input features. For this purpose, we linearize all the input features into a single dimension vector. Specifically, we concatenate the features of all the nodes in the graph in the same order as done in sequences/bag-of-words for a fair comparison.[1] Then we calculate the Levenshtein distances between every pair of message which the speaker produces over the whole dataset. The topographic similarity is then defined as the negative Spearman correlation coefficient ρ between the aforementioned cosine distance and the Levenshtein distance. Its value ranges from -1 to 1.

[1] We also tried shuffling the order of concatenation and the results were similar.

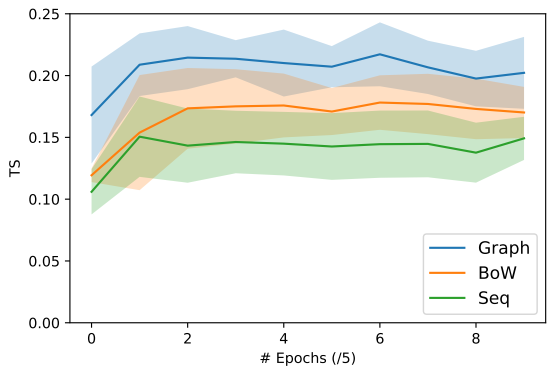

Figure 5: Topographic similarity in Game-1.

In Figure 5 we show the topographic similarity of all three data representations in Game-1. Since for all data representations, we see a positive value of ρ, this implies that there is a direct correlation on how the input features change the messages. Throughout the training, the graph representation are found to be more compositional (high topographic similarity) than bag-of-words and sequences. This aligns perfectly with the hypothesis that is found in previous literature that graph neural networks tend to capture compositional properties better than seq2seq networks [21].

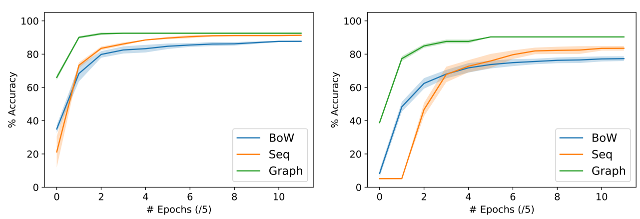

Figure 6: Out-of-domain generalization in Game-2. We show two plots with varying complexity of the game. The left plot is for a comparatively easier task with 9 distractors while the right one is on a harder version with 19 distractors. Both versions use 25 nodes, a vocabulary size of 25 and a message length of 25. Increasing the complexity of the game led the bag-of-words model to overfit to the training data since it performs the worst on generalization. We hypothesize that it created a partial hash-map for the set of inputs and failed to completely learn the combinatorial properties of the input space.

Out-of-domain generalization. We measure compositionality through out-of-domain generalization accuracy in Game-2. In this setup, the speaker observes a target entity which it has never observed before (i.e. an unseen combination of properties) and it is tasked to produce a unique code to be sent to the listener. If the agents learned to speak and understand compositional codes, then the speaker will utter a message which will consist of new symbols that are never seen by the listener during training. The listener then should be able to identify the target when mixed in a pool with other unseen distractors that are also composed of new properties. The only way the speaker can solve this task is if the speaker learned to disentangle the individual properties in the input and to encode it using a sequence of words with non-overlapping vocabulary. The job of the listener is to associate meanings to a subset of these words and relate it to the input properties.[2]

In Figure 6, we observe that graph neural networks perform the best across all models, and the gap between the second best performing model (seq2seqs) increases even more when the game complexity is increased. We attribute this to graph neural networks being able to capture better compositional structure of the input rather than learning a partial hash-map, which we hypothesize is what happens in the case of bag-of-words model. We also show some qualitative examples of the input data and messages transmitted by the speaker in Table 2.

Input data	BoW	Seq	Graph
A2 B4 C6	*[1 4 4]*	*[5 1 3]*	*[7 1 4]*
A2 B4 C5	[1 0 0]	[8 3 2]	[4 7 2]
A2 B2 C6	[6 4 1]	[9 9 1]	[9 4 1]
A5 B4 C6	[8 8 9]	[3 5 2]	[6 6 1]

Table 2: We show some qualitative samples from different models on Game-1. We vary one property while keeping the others fixed to show the corresponding change in the message. We represent the input data as shown in Figure 2. The vocabulary size is 10 with message of fixed length 3 and the perceptual dimensions being $[10, 6, 8]$. We see that in graphs varying one property changes only one symbol in the message thereby implying that they learned the compositional structure of the input. Similar behavior can be observed with bag-of-words model except the last one. The sequences perform comparatively worse than the other two often changing the whole message instead of just one symbol.

3.3 Graph Neural Networks in Emergent Communication

In both games, the two metrics used (topographic similarity and out-of-domain generalization) lead us to infer that graph representation performs better than the other two data representations in terms of learning systematic generalization and compositionality. To our knowledge, this is the first work which uses the benefits of graph neural networks in the domain of emergent communication. Thus we perform further analysis on using various types of graph encoders and pooling methods, and investigate the effect they have on learned communication protocols.

3.3.1 Ablation studies on Graph Neural Networks.

Pooling methods: In order to compute the graph embedding, we experimented with the standard graph pooling methods: mean, sum and max functions. We found that the sum pooling gave a significant boost in performance, and thus we use sum pooling throughout the experiments presented in this paper.

[2]Note that the listener also has to learn to disentangle the input properties. We tried sharing the same encoder for both speaker and the listener but it performed slightly worse than having separate modules.

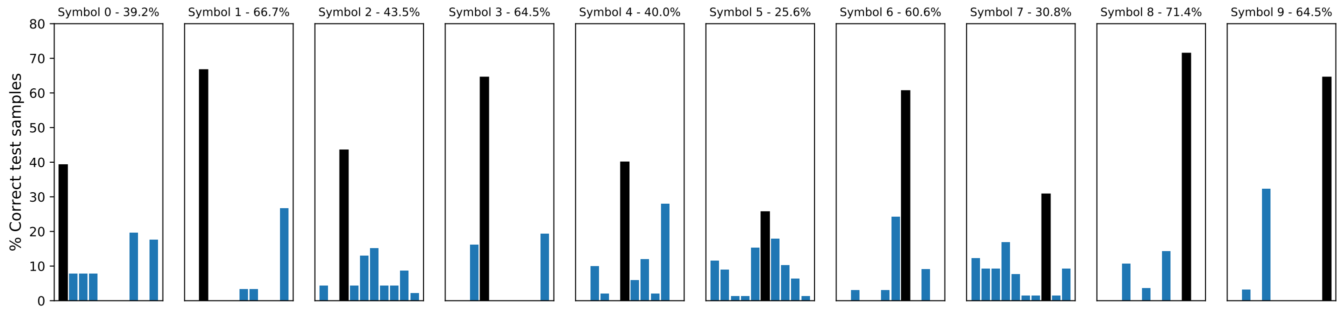

Figure 7: Robustness of the communication protocol. We show that for a symbol other than the one sent by the speaker (position of the black bar), the listener does not correctly identify the target. For more details, refer to §3.3. The above chart is for Game-1 with perceptual dimensions $[10, 6, 8]$ with a message length of size 3 and a vocabulary size of 10. We enumerate over the first property with vocabulary size 10 while keeping the other symbols in the message fixed.

Encoder networks: We also experimented with two popular graph neural networks to compute the graph encoding, namely GraphConv and SAGEConv. We did not find a significant difference in performance between the two models.

Aggregator types: Another axis of variation is the aggregator type used in GraphSAGE and we found that the effect of all types- 'mean', 'pool' and 'gcn' is the same across both games.

3.3.2 Robustness of communication protocols.

In Figure 7, we show the results of a robustness analysis of the communication protocols learned by the speaker/listener using the graph neural network approach. The purpose of this analysis is to see if the listener is able to correctly identify the target even if the message changed. If the agents learned to distinguish the target among distractors through communication, the listener should not be able to recognize the target if it has not received the target encoding from the speaker. For a given message of length $l = 3$, we fixed $l - 1$ i.e. 2 words at some arbitrary positions in the message and varied the third word over the whole vocabulary. For each word i in the vocabulary, we collected the set of test samples D_i where the speaker used the corresponding word i along with the other fixed symbols to represent the target graph, and the listener was able to correctly point to the target graph. Each subplot in Figure 7 represents the distribution of D_i over the whole vocabulary size $|V| = 10$. We observe that in all cases the symbol sent by the speaker (referred by the title in each subplot and the position of the *black* bar) is the one that allows the listener to correctly identify the target graph. We performed this analysis by randomizing the fixed positions over the whole message length. It shows that the listener actually cooperates with the speaker to build robust codes and uses the unique message to identify the target among distractors.

We also analyze the behavior of the listener when presented to the shuffled set of graphs. In each experiment, the position of the target was permuted across all of the possible $|K| + 1$ positions, where K is the number of distractors. We observe that the listener was still able to correctly identify the target based on the message sent from the speaker. We thus posit that the agents learned an order invariant representation of the graphs and not some positional information about the ordering of the graphs in the distractors set.

4 CONCLUSION AND FUTURE WORK

In this paper, we compared the various types of data representations inducing different levels of structural bias through the lens of emergent communication, and proposed a novel graph referential game. We showed that for two structured referential games, using a graph representation performs better than sequences and bag-of-words in terms of compositionality measured by topographic similarity and out-of-domain generalization. We showed that agents parameterized by simple graph neural networks also generalized more effectively to unseen combinations of familiar concepts and types. We also performed ablation studies on different variants of graph neural networks and showed robustness of the communication channel with respect to the position of the target among distractors. We found that the agents made an efficient use of the vocabulary, learned an order-invariant representation of the target graph, and solved the games with varying number of distractors through communication and cooperation. With the recent advancements of using graph based neural networks in natural language processing [25], a future direction could be to use these networks to generate sentences close to natural language [16]. Similarly to the state-of-the-art transformers, one can replace the separate sequence decoder with a fully connected graph network and might observe better performance. Another possible direction can be to use a deeper hierarchy of properties in the graph representations. We believe that more complex and realistic graphs would yield a higher degree of compositional understanding.

REFERENCES

[1] Jacob Andreas. 2019. Measuring Compositionality in Representation Learning. In *ICLR*. https://openreview.net/forum?id=HJz05o0qK7
[2] Dzmitry Bahdanau, Shikhar Murty, Michael Noukhovitch, Thien Huu Nguyen, Harm de Vries, and Aaron Courville. 2019. Systematic Generalization: What Is Required and Can It Be Learned?. In *ICLR*. https://openreview.net/forum?id=HkezXnA9YX
[3] Marco Baroni. 2020. Linguistic generalization and compositionality in modern artificial neural networks. *Philosophical Transactions of the Royal Society B: Biological Sciences* 375 (02 2020), 20190307. https://doi.org/10.1098/rstb.2019.0307
[4] Diane Bouchacourt and Marco Baroni. 2018. How agents see things: On visual representations in an emergent language game. In *EMNLP*. Association for Computational Linguistics, Brussels, Belgium, 981–985. https://doi.org/10.18653/v1/D18-1119

[5] Henry Brighton and Simon Kirby. 2006. Understanding Linguistic Evolution by Visualizing the Emergence of Topographic Mappings. *Artif. Life* 12, 2 (March 2006), 229–242. https://doi.org/10.1162/106454606776073323

[6] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive Representation Learning on Large Graphs. In *NIPS*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.). Curran Associates, Inc., 1024–1034. http://papers.nips.cc/paper/6703-inductive-representation-learning-on-large-graphs.pdf

[7] Eric Jang, Shixiang Gu, and Ben Poole. 2017. Categorical Reparameterization with Gumbel-Softmax. In *ICLR*. https://openreview.net/forum?id=rkE3y85ee

[8] Eugene Kharitonov, Rahma Chaabouni, Diane Bouchacourt, and Marco Baroni. 2019. EGG: a toolkit for research on Emergence of lanGuage in Games. In *EMNLP-IJCNLP*. Association for Computational Linguistics, Hong Kong, China, 55–60. https://doi.org/10.18653/v1/D19-3010

[9] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *ICLR*.

[10] Satwik Kottur, José Moura, Stefan Lee, and Dhruv Batra. 2017. Natural Language Does Not Emerge 'Naturally' in Multi-Agent Dialog. In *EMNLP*. Association for Computational Linguistics, Copenhagen, Denmark, 2962–2967. https://doi.org/10.18653/v1/D17-1321

[11] Brenden M. Lake, Tomer D. Ullman, Joshua B. Tenenbaum, and Samuel J. Gershman. 2017. Building machines that learn and think like people. *Behavioral and Brain Sciences* 40 (2017), e253. https://doi.org/10.1017/S0140525X16001837

[12] Angeliki Lazaridou, Karl Moritz Hermann, Karl Tuyls, and Stephen Clark. 2018. Emergence of Linguistic Communication from Referential Games with Symbolic and Pixel Input. In *ICLR*. https://openreview.net/forum?id=HJGv1Z-AW

[13] David Lewis. 1969. *Convention: A philosophical study*. Harvard University Press.

[14] Fushan Li and Michael Bowling. 2019. Ease-of-Teaching and Language Structure from Emergent Communication. In *NeurIPS*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d Alché-Buc, E. Fox, and R. Garnett (Eds.). Curran Associates, Inc., 15825–15835. http://papers.nips.cc/paper/9714-ease-of-teaching-and-language-structure-from-emergent-communication.pdf

[15] Michael L Littman. 1994. Markov Games as a Framework for Multi-Agent Reinforcement Learning. In *ICML*, Vol. 157. 157–163.

[16] Ryan Lowe*, Abhinav Gupta*, Jakob Foerster, Douwe Kiela, and Joelle Pineau. 2020. On the interaction between supervision and self-play in emergent communication. In *ICLR*. https://openreview.net/forum?id=rJxGLlBtwH

[17] Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. 2017. The Concrete Distribution: A Continuous Relaxation of Discrete Random Variables. In *ICLR*. https://openreview.net/forum?id=S1jE5L5gl

[18] Igor Mordatch and Pieter Abbeel. 2018. Emergence of Grounded Compositional Language in Multi-Agent Populations. In *AAAI Conference on Artificial Intelligence*. https://aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/17007

[19] Adam Paszke, Sam Gross, Francisco Massa, et al. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *NeurIPS*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché Buc, E. Fox, and R. Garnett (Eds.). Curran Associates, Inc., 8024–8035. http://papers.nips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

[20] Cinjon Resnick*, Abhinav Gupta*, Jakob N. Foerster, Andrew M. Dai, and Kyunghyun Cho. 2019. Capacity, Bandwidth, and Compositionality in Emergent Language Learning. *ArXiv* abs/1910.11424 (2019).

[21] Koustuv Sinha, Shagun Sodhani, Jin Dong, Joelle Pineau, and William L. Hamilton. 2019. CLUTRR: A Diagnostic Benchmark for Inductive Reasoning from Text. *EMNLP-IJCNLP* (2019). https://doi.org/10.18653/v1/d19-1458

[22] Kenny Smith, Simon Kirby, and Henry Brighton. 2003. Iterated Learning: A Framework for the Emergence of Language. *Artif. Life* 9, 4 (Sept. 2003), 371–386. https://doi.org/10.1162/106454603322694825

[23] Sainbayar Sukhbaatar, Arthur Szlam, and Rob Fergus. 2016. Learning Multiagent Communication with Backpropagation. In *NeurIPS*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett (Eds.). Curran Associates, Inc., 2244–2252. http://papers.nips.cc/paper/6398-learning-multiagent-communication-with-backpropagation.pdf

[24] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to Sequence Learning with Neural Networks. In *NeurIPS*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger (Eds.). Curran Associates, Inc., 3104–3112. http://papers.nips.cc/paper/5346-sequence-to-sequence-learning-with-neural-networks.pdf

[25] Ashish Vaswani, Noam Shazeer, Niki Parmar, et al. 2017. Attention is All you Need. In *NeurIPS*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.). Curran Associates, Inc., 5998–6008. http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf

[26] Minjie Wang, Lingfan Yu, Da Zheng, et al. 2019. Deep Graph Library: Towards Efficient and Scalable Deep Learning on Graphs. *ICLR Workshop on Representation Learning on Graphs and Manifolds* (2019). https://arxiv.org/abs/1909.01315

[27] Ludwig Wittgenstein. 1953. Philosophical investigations. (1953).