UNIVERSITÄT OSNABRÜCK

Community Size Effects on Language Emergence in a Multi-Agent Game

Bachelor Thesis

Juri S. Morisße

Institute of Cognitive Science

Matriculation Number:	982393
Course of Studies:	Cognitive Science B.Sc.
First Examiner:	Prof. Dr. Elia Bruni
Second Examiner:	M.Sc. Xenia Ohmer
Date:	03.08.2022

Abstract

Language is a crucial tool for cooperation between individuals and it is thus a key interest of researchers who are trying to understand human and artificial intelligence. One approach to study the development and use of language simulates language emergence among agents that play a cooperative game. However, most of these studies make use of only a single pair of agents which is an unrealistic restriction. In this study, I contribute to the little research done on language emergence among communities of agents by investigating the effects community size has on the emerging languages in a multi-agent discrimination game. I find that communities with more than one agent pair achieve lower performance in their cooperative game, show less topographic similarity, and encode less conceptual information in their messages compared to the single agent pairs. Agents from communities pursue a strategy of co-adaptation between agents by generating messages that contain a “signature” of their author. As a consequence, agent communities do not converge towards a common language but give rise to several agent specific languages.

Table of Contents

1. Introduction

Communication is of fundamental importance to intelligence. It enables its users to coordinate, cooperate, and negotiate to achieve common goals. Consequently, understanding the development and use of language is of great relevance to researchers interested in natural or artificial intelligence. Simulating language development with artificial agents is one scientific approach that allows the investigation of both language development and its use. Inspired by Wittgenstein's famous idea that language derives its meaning from its use (Wittgenstein, 1953), many of these language simulations are embedded in a game scenario in which agents must learn to communicate to play a cooperative game. By varying factors like the game setup, interactions between agents, or restrictions on communication, researchers try to identify properties of language development that can be transferred also to natural languages. In addition, insights into unsupervised, task-oriented language learning of multiple agents can benefit the application of artificial agents to real world tasks.

While there have been many studies on language emergence in multi-agent games, the majority are using only a single pair of agents. Considering that the goal is to contribute to the understanding of natural languages or improving communication between artificial agents, this is a limiting factor. After all, both humans and cooperative artificial agents must be able to communicate with more than one partner which in turn is likely to influence the development and use of their languages. There are only a few previous studies that have looked at effects on language emergence in multi-agent games played by more than one agent pair. My study builds on these few studies. Specifically, **I aim to answer the question whether there is a community size effect on how and what languages emerge in a multi-agent communication game**. I test the community size effect on game success achieved by the agents, their convergence towards a common language, the topographic similarity between their input and the messages they produce from it, and the degree to which they use the emerging language for encoding information about high level concepts in their messages. To do so, I train agents to play a version of Lewis' signaling game (1969) that requires the agents to develop communication allowing them to cooperate. By doing this for communities of differing sizes and comparing the emerging languages, possible community size effects can be found.

There are previous studies on language emergence in games in which agents were part of larger communities (Cogswell et al., 2019; Dagan et al., 2021; Dubova & Moskvichev, 2020;

Graesser et al., 2019; Kim & Oh, 2021; Tieleman et al., 2019). However, most of them aim to answer other research questions and only a few also have their focus on investigating the community size effect (Dubova & Moskvichev, 2020; Kim & Oh, 2021; Tieleman et al., 2019). In this work, I build up on these previous studies by using images from a more diverse and complex image dataset and by investigating the community size effect on several metrics. This makes my study the most in-depth investigation into community size effects so far and thus an important addition to the scarce research done in this direction.

My findings reveal that with increasing size, communities become less successful in playing the game. Further, communities with more than one agent pair show lower topographic similarity between their input and message space, and their messages contain less information about high level concepts of the inputs. The analysis of the emerging languages shows that agents from larger communities do not agree on a common language. An exploratory analysis shows that they instead follow a strategy of generating sender specific messages that allows the receivers to co-adapt to several senders at the same time and prevents the community from developing one common language. I discuss the implications of this observation has and problems in preventing co-adaptation with previously suggested techniques. To overcome this, I suggest the development of an effective strategy for preventing co-adaptation in agent communities to be the subject of future research.

In the following, I start by giving a comprehensive overview of research in the area of language emergence in multi-agent games. I continue by giving a description of my experimental setup and providing details on the experiments performed. Afterwards, I present the obtained results and discuss them critically. In the end, I review my study and provide a conclusion.

2. Related Work

The field of simulating language emergence has drawn researchers’ attention for over three decades. Since the beginning, the underlying motivation for this type of research has been to better understand the emergence and development of natural languages and improve communication of artificial systems (Wagner et al., 2003). While the motivation was clear, the employed methods to implement agents and the communication learning process showed a large variety. In one of the earliest works in the field, MacLennan and Burghardt (1993) demonstrated that agents based on finite-state machines can be optimized with an evolutionary algorithm to develop the ability to communicate for cooperation. Other early approaches made use of evolutionary optimization of agents implemented via Recurrent Neural Networks (RNN; Batali, 1994). Later on, Kasai et al. (2008) employed multi-agent reinforcement learning to investigate the effect different signal lengths have on agents' communication success. This was one of the first instances of multi-agent reinforcement learning that did not rely on pre-defined communication protocols, an approach that has become widely employed in recent works.

Using games as an environment in which agents learn to communicate is another characteristic of modern language emergence studies that was also employed in early works (Giles & Jim, 2003, Steels 1998), even before the use of multi-agent reinforcement learning. Some of these early studies saw games as a way to replace the problem of communication learning with the problem of achieving an equilibrium in game success (Hasida et al., 1995). In modern language emergence literature, games are still the established environment in which agents learn to communicate but this choice is rooted in the avant-guard philosophical work of Ludwig Wittgenstein (1953). In this work, Wittgenstein describes language as a tool for cooperation and that the meaning of a word is defined by the way it is used by speakers to achieve a common goal. The first to operationalize Wittgenstein's ideas in the context of experimental linguistics was Lewis (1969). He proposed the signaling game, the first pioneer experiment on language emergence in a game-theoretic setting. The signaling game is a two-player game, where a speaker sends a signal to which the listener reacts. If the listener reacts correctly to the signal, both players receive a positive reward and will adjust future actions accordingly. Variations of the signaling game are used in nearly all modern language emergence studies (Choi et al., 2018; Dagan et al., 2021; Havrylov & Titov, 2017; Lazaridou et al., 2017; Lazaridou et al., 2018; Li & Bowling, 2019; Lowe et al., 2019; Luna et al., 2020).

As pointed out, early work on simulating language emergence has many similarities with modern studies. Until recently, however, computational limitations prevented the study of language emergence in more complex settings. The rise of deep learning to achieve state-of-the-art performance in many domains (LeCun et al., 2015) brought with it a renewed interest in the field. While supervised deep learning became a successful approach for natural language processing tasks like dialogue modelling (Vinyals & Le, 2015) or translation (Bahdanau et al., 2015), deep reinforcement learning became the dominant approach for communication learning in multi-agent games (Foerster et al., 2016a; Foerster et al., 2016b;

Havrylov & Titov, 2017; Jorge et al., 2016; Kajić et al., 2020; Lazaridou et al., 2017; Lazaridou et al., 2018; Luna et al., 2020; Sukhbaatar et al., 2016). Supervised learning was seen to be more suitable for learning statistical relations of communication symbols but deficient in accounting for the functional use of language, which is a key aspect of language emergence in games (Lazaridou et al., 2017). Initially communication symbols were often continuous (Sukhbaatar et al., 2016) or messages consisted only of single symbols (Lazaridou et al., 2017; Foerster et al., 2016a). Researchers quickly moved on to communication via messages made up of discrete symbol strings (Havrylov & Titov, 2017; Lazaridou et al., 2018) because this more closely resembles human languages.

After researchers had demonstrated that modern deep learning agents can successfully learn to communicate about solving a task with natural language-like messages, scientific focus shifted towards analyzing emergent languages and factors influencing them. While earlier work reported that the learned languages encode physical concepts (Jorge et al., 2016) and possess compositional properties (Havrylov & Titov, 2017), this view was later challenged by Kottur et al. (2017). They found that success of multi-agent communication is not in itself an indicator for the learned communication protocol to be compositional or interpretable. This was supported by other findings which showed that agents might simply encode pixel values instead of high-level concepts (Bouchacourt & Baroni, 2018). Both Kottur et al. (2017) as well as Bouchacourt and Baroni (2018) argued that the design of the game environment has a fundamental influence on what the learned language actually encodes. As Kottur et al. (2017) put it “compositional language does not naturally emerge without an explicit need for it” (p. 6).

It became a key interest in the field to understand how such an explicit need for a desired language property like compositionality can be included in a setting to promote the emergence of more natural-like languages. The environmental factors that cause this need are often referred to as pressures. They are implemented into a system to pressure agents towards communicating in a desired manner. An often-employed strategy is to make the environment more similar to the conditions in which humans acquire language. Luna et al. (2020) investigated three such pressures present in human communication: least effort, subjective consistency, and object frequency. Least effort describes the human tendency to communicate as efficiently as possible. Subjective consistency is the ability to recognize objects regardless of their orientation or illumination. Object frequency describes the fact that humans do not encounter objects equally often, but some are more frequent and therefore easier to differentiate. Luna et al. (2020) employed these pressures by introducing an additional loss to punish the use of longer messages and skewing the distribution and composition of input image-pairs. They reported that introducing pressures causes the emerging language to exhibit increases in compositionality and ability to generalize. Other investigated pressures reported to lead to compositional communication include coaxing agents to communicate with messages that are easier to learn for untrained agents (Li & Bowling, 2019), and situating agents into an environment in which they do not just communicate but can also interact in non-communicative ways (Mordatch & Abbeel, 2018; Kajić et al., 2020). However, introducing pressures that represent circumstances in which human language evolved, has not always resulted in more compositionality. Lazaridou et al. (2018) investigated whether compositional language also emerges when the sender agents do not get a disentangled attribute vector as input but instead a vector of raw pixel data. The raw pixel data is closer to the raw visual input humans receive. They report that using disentangled attribute vectors leads to communication protocols with a higher degree of compositionality. My work builds up on this previous research by investigating a different kind of pressure, namely community size.

Previous studies on language emergence have demonstrated that successful communication learning is also possible in settings with agent communities instead of a single pair of agents. As already mentioned, Li and Bowling (2019) investigated pressures promoting easier-to-teach communication protocols. They did so by periodically pairing the sender with a new untrained receiver, which is essentially equal to pairing the sender to a receiver population agent after agent. Other studies have used sender and receiver populations to mimic language transmission between generations (Cogswell et al., 2019; Dagan et al., 2021), or to investigate the effect of social structure within the populations (Graesser et al., 2019; Dubova et al., 2020). While these works used settings with agent populations, their focus was not on the effect community size has on the emerging language.

The community size effect has been investigated in only a few studies. The initial work was done by Tieleman et al. (2019) in which larger community sizes were reported to reduce idiosyncrasy in agent communication and lead to emerging language that better encodes high-level concepts. Other work on the community size effect found that larger communities cause higher symmetry of agent communication (Dubova & Moskvichev, 2020) and increased language variability under stable success rates (Kim & Oh, 2021). My work builds up on the study by Tieleman et al. (2019). Similar to them, I focus only on the community size effect without investigating the effect of any other factors. Furthermore, I too will analyze the emerging language with regards to its ability to encode concept-level information. In addition to the analysis by Tieleman et al. (2019), I will also explore the community size effect on the compositionality of the learned language by looking at topographic similarity. Another key difference is that I make use of deep reinforcement agents that communicate via sequences of discrete symbols to play a discrimination game compared to the traditional autoencoder setup used by Tieleman et al. (2019) in which the latent vectors, ^^, the encodings, resemble fixed length continuous communication that are used by the agents to play a reconstruction game.

3. Experimental Framework

In this study, I investigate the community size effect on language emergence in multi-agent games. For the implementation of the study, I make use of the Emergence of Language in Games (EGG) toolkit which was introduced by Kharitonov et al. (2019)[1]. EGG was specifically developed to help researchers in conducting studies about multi-agent language emergence by providing pre-implemented functions and games that can be individualized and put together to create unique study designs. Below, I present the design of my study by providing details on the multi-agent game, the implementation of the agents, the procedure for forming communities of agents, and the evaluations performed for revealing the community size effect.

3.1. Game

In line with modern studies on language emergence in multi-agent games (Dagan et al., 2021; Havrylov & Titov, 2017; Luna et al., 2020), I let the agents play a discrimination game. The discrimination game is a version of the signaling game proposed by Lewis (1969). It is played by two agents. One of them (the sender) receives an image (the target image) and produces a variable length discrete message that is sent to the other agent (the receiver). The receiver uses the message to discriminate the image the sender saw from a randomly ordered set of images that includes the target image and distractor images. To successfully play the game, the sender must learn to encode information about the target image and the receiver must learn to extract that information from the message. The procedure is based on the formalization by Havrylov and Titov (2017) and consists of three steps:

[1] https://github.com/facebookresearch/EGG

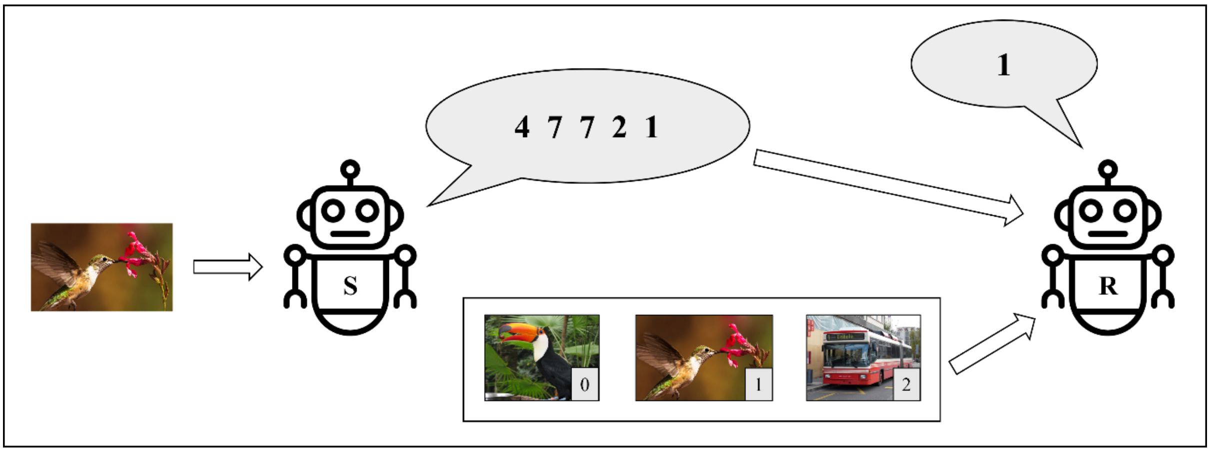

Figure 1: *An example of a successful game run with images from the ILSVRC2012 dataset. The sender agent (S) receives the target image and sends a message to the receiver agent (R). The receiver reads the message and uses it to discriminate the target image from a set of given images.*

1. A target image t and a set of K distractor images $\{d_k\}_{k=1}^{K}$ are sampled from a set of N images $\{i_n\}_{n=1}^{N}$.
2. The sender sees the target image t and produces a message m_t, which is a sequence of discrete symbols from vocabulary V of size $|V|$. The maximum length of m_t is L.
3. The receiver gets the message m_t along with a randomly ordered set of images which contains the distractor images and the target image. The task of the receiver is to identify which image was seen by the sender.

In my setup, the sender produces messages with a maximum length of five. The symbols are integers in the range from 0 to 9. However, 0 serves as a stop symbol which means $|V|$ is effectively 9. Information about the set of images from which target and distractor images are sampled from can be found in section 4.1. A visualization of one exemplary game run is shown in Figure 1.

3.2. Agent Architecture

Agents implemented within EGG consist of two parts: a core architecture and a pre-implemented wrapper function. The wrapper functions are provided by EGG for implementing the communication channel and are applied on the core architectures. Both parts differ between senders and receivers, i.e., each agent can only perform one of the roles. The sender core architecture defines how a sender maps an image to the initial hidden state that is used by its message-producing wrapper function. In my setup, this is done via a single fully connected layer. This layer maps the input of size 512 to an initial hidden state of size 10. Before this initial hidden state is fed to the sender's wrapper function, a tanh activation function is applied. The sender's wrapper function takes this initial hidden state and feeds it into a gated recurrent unit (GRU; Cho et al., 2014). The wrapper unrolls the GRU to produce message symbols until the maximum length is reached or the stop symbol 0 is produced.

The receiver core architecture defines how a receiver discriminates the target image from the distractors with the help of a message representation provided by the receiver's wrapper function. To produce this message representation, the receiver's wrapper processes the sequential message via GRU and the final hidden state of size 10 serves as a message representation that is fed into the receiver's core. In the core, all images are mapped to the same dimension as the message representation, i.e., 10. Afterwards, the dot products between the message representation and all mapped images are computed and stored in a list. The output of the receiver is this list of dot products to which a softmax activation function has been applied. Consequently, the output can be interpreted as the receiver's certainty about which of its received images is the target image. If the index of the maximum in the dot product list is identical with the index of the target image in the randomly ordered set of target and distractor images, the receiver made the correct decision.

3.3. Agent Communities

To investigate the community size on language emergence, I construct several agent communities with differing community sizes C. When speaking of a community with size C, I refer to a setting in which there is one sender community and one receiver community and both contain C agents, so a total number of $2 \times C$ agents. To implement a community, I implement C senders and C receivers with the architecture described in 3.2. It is important to note that all senders share a common architecture but are trained and optimized individually and as such do not share any weights. The same is the case for the receivers. More details on training can be found in section 4.2. I investigate communities of size 1, 2, and 4 and only use setups in which the sender and receiver communities are of the same size.

3.4. Evaluations

Once the agents are trained, I perform evaluations to analyze the effect of their community sizes on the languages they learn. All evaluations are performed using the evaluation data that is described in 4.1.2.4. Since I run training with five different seeds for each community size (see section 4.3), all evaluations are also conducted five times per community size, once per seed. The average across seeds yields a score for the respective community size. In this subsection, I will outline these evaluations.

3.4.1. Game Success

Quantification of the game success is achieved by computing the receiver's accuracy in discriminating the target image from the distractor images. For each sample of the evaluation data, I average the accuracies achieved by all possible pairs of senders and receivers. Averaging over all the samples' mean pair accuracies yields a game success score for a particular community size and seed. Game success scores of all community sizes are compared to find an eventual community size effect on the game success.

3.4.2. Language Convergence

To understand whether agents develop a common language, I compare the messages generated by different senders given the same input. To quantify the similarities of these discrete messages, I use the Hamming Distance which has been used in earlier studies as a measure of language convergence in the case of discrete communication (Kim & Oh, 2021). The Hamming Distance is defined as the number of differences between two strings of symbols. I will normalize the Hamming Distance between messages to the range from 0 to 1. Further, I will deduct this normalized Hamming Distance from 1 to obtain a similarity measure ranging from 0 to 1. Lastly, I compute the mean pairwise similarity between all messages for each sample in the evaluation dataset and average over all the samples' mean message similarities. This average over all samples acts as an overall language convergence score for a particular community size and seed. Comparing the language convergence scores of all community sizes allows to identify a possible community size effect on language convergence.

3.4.3. Topographic Similarity

As pointed out in previous work (Lazaridou et al., 2018; Li & Bowling, 2019), there is no metric to measure compositionality directly. To circumvent this problem, evaluation of compositionality often focuses on analyzing properties of a language that are suggested to be required for compositionality. The characteristics of a language regarding such properties then serve as indicators for compositionality. I will investigate the compositionality of the languages emerging in my game setup by measuring one of the indicators that are most used in earlier studies: topographic similarity. In a compositional language, the meaning of a sentence is defined by the words it contains. When describing an object in a compositional language, it is intuitive to think of composing the description of the object from descriptions of its attributes (e.g., shape, color, size). Consequently, objects that share similar attributes, would lead to similar descriptions. This idea is the motivation behind using topographic similarity as an indicator for a language's compositionality like it has been done in previous studies (Keresztury & Bruni, 2020; Lazaridou et al., 2018; Luna et al., 2020).

Topographic similarity measures the relation between two different spaces. In my study, those are the input space (the images fed into the senders) and the message space. Topographic similarity is defined as the correlation between distances of pairs in the input space and distances of the respective messages in the message space. If the language used by senders is compositional, similar images should be described with similar messages. Therefore, a high topographic similarity would be an indicator of compositionality in the emerged language.

I compute topographic similarity per sender. As representations of the input and message space, I collect 10,000 pairs of samples from the evaluation dataset. The sender's topographic similarity will then be computed as the Pearson correlation between the cosine distances of the sender inputs and the Hamming distances of the messages. Averaging over all the senders in a community yields a topographic similarity score for a particular community size and seed. By comparing the topographic similarity scores of all community sizes, I can identify a possible community size effect on topographic similarity and indirectly on compositionality of the emerged languages.

3.4.4. Concept Encoding

The use of artificial neural networks (ANN) for implementing agents allows for better performance at the cost of interpretability. Determining what information the agents make use of and how they process it is a difficult task. It is telling that ANNs are often referred to as a black box. Consequently, it is by no means an easy task to investigate if the senders use the emerged language to encode high level concepts. A technique that is used to extract information from intermediate layers of an ANN makes use of a second ANN, a diagnostic classifier (Hupkes et al., 2018). This approach is based on the idea that if an ANN conveys information through its layers, it should be possible to extract that information with the help of an additional ANN, the diagnostic classifier. This idea is transferable to my game scenario and the problem of understanding whether agents use an emerged language to encode concepts. If the senders use their messages to describe high level concepts of their input image, then a diagnostic classifier should be able to extract that information and use it to make an inference about the input image. Indeed, diagnostic classifiers have been used in earlier studies to compare the concept encoding abilities of agents trained in different community sizes (Tieleman et al., 2019). However, a crucial difference is that the messages in my game setup are sequential strings of symbols in contrast to the fixed length latent vector used by Tieleman et al. (2019). To enable my diagnostic classifier to properly understand the sequential messages generated by the senders, I use a GRU as the input layer of my diagnostic classifier. Hupkes et al. (2018) have shown that RNNs, like a GRU, are capable of successfully functioning as a diagnostic classifier. In addition to the GRU that functions as an input layer and reads in the messages, my diagnostic classifier has two more layers. A fully connected hidden layer maps the final hidden state of the GRU cell to a latent representation. This latent representation is then mapped by a fully connected output layer to a representation with one element for each possible image class. The element with the highest activation in the output layer is the diagnostic classifier's prediction about the inputs' class. Random subsets of the evaluation data are used to construct a training, validation and test dataset. While the evaluation data differs between seeds and community sizes, the sub-setting procedure is fixed to ensure that no diagnostic classifier has an advantage or disadvantage from it. The subset used for constructing the training dataset contains 60% of the evaluation data's samples. The resulting training dataset consists of the message from the sender that is to be diagnosed and the class of the image that was fed into the sender. The messages are the input to the diagnostic classifier and the image classes labels to evaluate the classification performance. The test and validation dataset are constructed in the same way from the remaining 40% of the evaluation data. The validation dataset is used for finding the right hyperparameter settings. Only once the hyperparameter best for performance on the validation data are found, is the test data used. Training is performed for 30 epochs with a learning rate of 0.001 and a batch size of 256. Cross entropy is the loss function in use and the diagnostic classifiers are optimized via Adam. After training, the diagnostic classifier's accuracy on the test set is recorded. That accuracy is the concept encoding score of the sender for which the diagnostic classifier was trained. Averaging the concept encoding scores of all senders yields a concept encoding score for a particular community size and seed. Comparing the concept encoding scores of all communities allows to identify a possible community size effect on the ability of senders to encode high-level concepts like image classes.

4. Experiments

Within the framework outlined in the previous section, I train communities of differing sizes to play the discrimination game and evaluate the language that they learn in the process. To enable reproduction of my results, I use this section to provide details on the data, optimization, and training procedure used.

4.1. Data

All images used in my game are from the ILSVRC2012 dataset which is based on the ImageNet dataset (Russakovsky et al., 2015). ILSVRC2012 contains over 1.2 million images each belonging to one of 1,000 classes. This means ILSVRC2012 is much larger compared to CIFAR10 or CIFAR100, the other most used natural image datasets in similar studies. On top of the 10 to 100 times more classes present in ILSVRC2012, the images are also of higher resolution and belong to classes that are less distinct. For example, while CIFAR10 has images of the class dog, ILSVRC2012 contains images of 120 different dog breeds that all constitute their own class. For all these reasons, images from ILSVRC2012 are more diverse and complex than images used in previous studies. Below, I will explain the pre-processing performed on the original ILSVRC2012 images and provide details of the different datasets used for training, validation, testing, and evaluation.

4.1.1. Pre-processing

The size of the ILSVRC2012 dataset and its high-resolution images are computationally expensive to process. To reduce the complexity of each game iteration, I do not use the raw images directly. Instead, I produce image embeddings that are used as inputs to the agents. I use the pretrained ResNet18 (He et al., 2016) for embedding the images. The second to last latent layer of ResNet18 serves as the embedding space. This results in an embedding dimensionality of 512 and allows the agents to process images without having a vision module themselves. The embedding process takes place during the generation of the game datasets as described in section 3.1. For simplicity, I refer to these image embeddings as images.

4.1.2. Datasets

I use the ILSVRC2012 images to construct three game datasets: training, validation, and test data. This is done in two steps. First, I perform the pre-processing as described in 3.5.1 and store the embeddings of the target and distractor images along with the ILSVRC2012 class of the target image and the index of the target image. In the second step, I use this data to produce the game datasets. All game datasets contain samples with three variables: sender input, receiver input, and target index. The test dataset contains an additional variable for each sample in which the ILSVRC2012 class of the target image is stored for later analysis of concept encoding (see 4.3). All game datasets are non-overlapping, i.e., any image can be only in one of the game datasets. On top of the four game datasets, I produce a dataset for evaluation. This evaluation dataset stores variables to allow analysis of a community's performance on the test dataset. While the game datasets are identical across all communities and seeds, evaluation datasets share the same variables within a certain community size, but the values are unique to the agents of a particular seed. In the following, I will explain the different datasets in detail.

4.1.2.1. Training Data. The training data consist of 150,000 samples. Each sample contains one target image, two distractor images, and a target index indicating the position of the target image. All images and the target index are randomly sampled. The images are from randomly chosen 60% of the images in the ILSVRC2012 train split that is provided by the Torchvision package. 60% of that train split corresponds to about 720,000 images. The training data is used in each game iteration for training the agents.

4.1.2.2. Validation Data. The validation data contains 30,000 samples. Each sample contains one target image, two distractor images, and a target index. All images and the target index are randomly sampled. The images are from the ILSVRC2012 val split that is provided by the Torchvision package. This val split contains 50,000 images. The validation data is used after each game iteration for validating the agent's performance on untrained data. Hyperparameters are adjusted based on this validation performance to improve performance also on the test data.

4.1.2.3. Test Data. The test data contains 100,000 samples. Each sample contains one target image, two distractor images, the target image's ILSVRC2012 class, and the target index. All images and the target index are randomly sampled. The images are from randomly chosen 40% of the images in the ILSVRC2012 train split that is provided by the Torchvision package. The 60% and 40% subsets of the ILSVRC2012 train split for producing the training and testing data are sampled such that they are non-overlapping. 40% of the train split corresponds to about 480,000 images. The test data is used only after the adjustments of hyperparameter. The agents' performance on the test data is recorded and stored as the evaluation data.

4.1.2.4. Evaluation Data. The evaluation data contains 100,000 entries, one for each sample of the test dataset. Each entry contains the target image from the respective test dataset sample, the ILSVRC2012 class of that target image, the average accuracy of all possible sender-receiver-pairs from the community when confronted with this sample, and the messages produced by all the senders of the community. While training, validation, and test data are fixed across all community sizes and seeds, the evaluation data records the performance of a certain community of agents and is therefore unique to that community. It is used to analyze the language used by that community as described in section 4.

4.2. Training and Hyperparameter

All communities are trained using the training dataset described in section 4.1.2.1. In each training epoch, a randomly chosen sender is paired with a randomly chosen receiver. The sampled pair then plays the game once for each sample in the training dataset and is optimized each time it completes a full batch. Once the pair has completed its training on the whole training dataset, the next training epoch begins in which a new pair is sampled and trained. The number of possible pairs grows exponentially with the community size, i.e., in a run with community size C there are C^2possible pairs. To account for that, the number of training epochs is computed based on a variable denoting the average training epochs per pair. Consequently, the number of total training epochs also grows exponentially with the community size. I set the number of average training epochs per pair to 200. Therefore, for community size 1, 2 and 4, 200, 800 and 3200 training epochs are performed respectively. Training is performed in batches of 128 samples with a learning rate of 0.0001. As loss function, I use cross entropy. The optimization technique used is REINFORCE (Williams, 1992) and as an optimizer I use Adam. To account for the random weight initialization of agents and other random factors in the training procedure, I perform the training with five different seeds for each community size. To visualize the training, I record the accuracy and loss of all pairs on each sample of the training and validation dataset. This evaluation is performed once per epoch and the resulting graphs of accuracy and loss over epochs are plotted in Figure 2. Due to a significant increase in runtime caused by this procedure, I only do this visualization for the first seed. All evaluations are performed on all seeds as described in section 3.4.

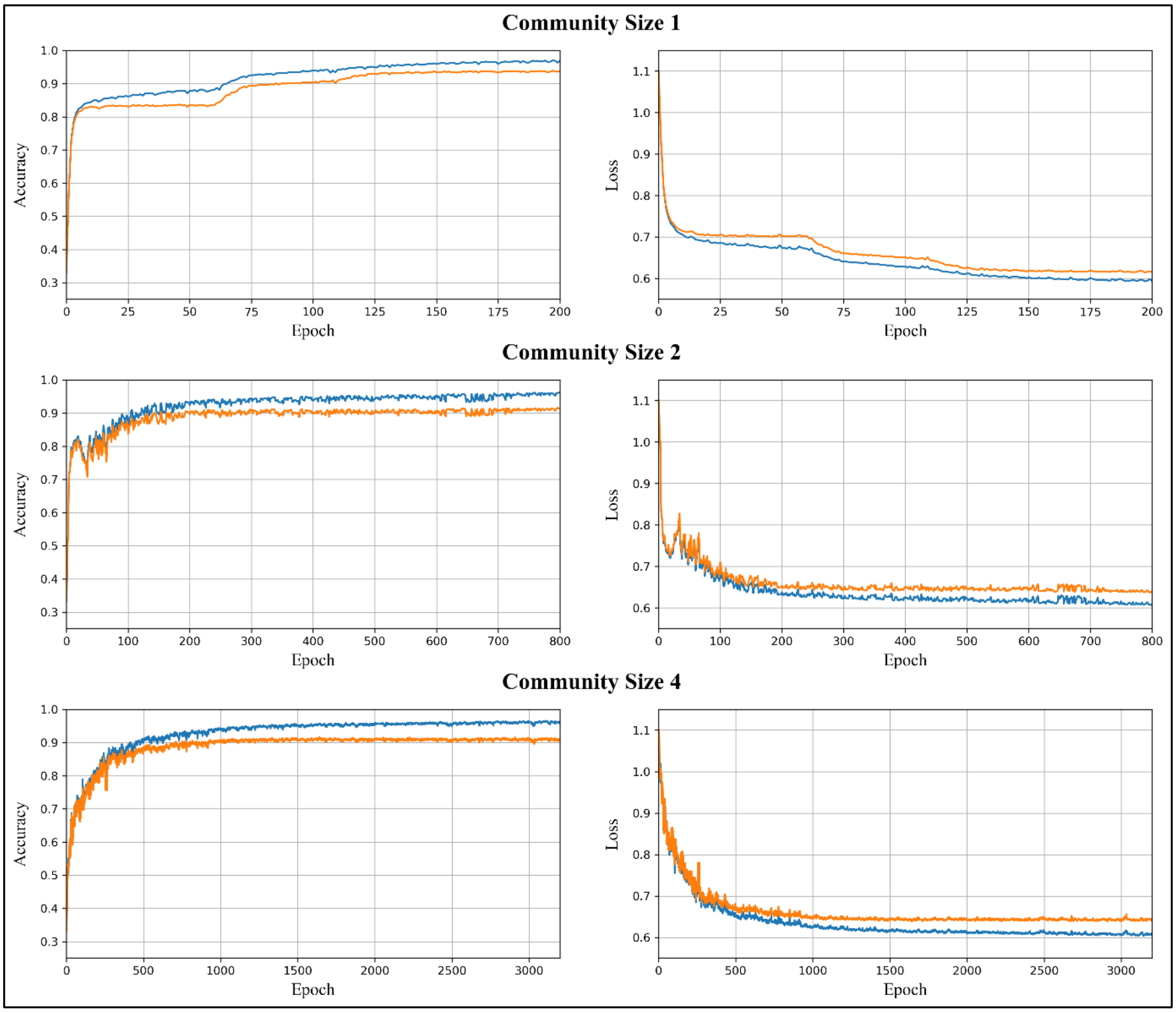

Figure 2: *The development of accuracy and loss during the training procedure of the first seeds of each community size. Accuracy and loss are computed as an average achieved by all possible pairs on all samples of the training (blue) and validation (orange) datasets for each epoch.*

5. Results

In this section, I present the results of the evaluations described in section 3.4. The reported results are averages over five seeds for each community size and are listed in Table 1. To give the reader insights into the variations between the different seeds, Table 1 also includes the standard deviations over seeds for each community size and metric. The individual seed metrics are included in the respective figures along with their average. In addition, I present the correlations between the metrics of individual seeds and their community size. For easier readability, all metrics are rounded to four decimal places.

5.1. Game Success

As can be seen in Figure 3, agents of all community sizes develop a communication

Community Size	Game Success	Language Convergence	Topographic Similarity	Concept Encoding
1	94.15 ± 2.71	1.0000 ± 0.0000	0.2367 ± 0.0109	0.68 ± 0.21
2	91.28 ± 1.06	0.2456 ± 0.1338	0.2310 ± 0.0256	0.43 ± 0.02
4	90.30 ± 0.96	0.2495 ± 0.0906	0.2629 ± 0.0135	0.44 ± 0.02

Table 1: *Results of the evaluations performed for all community sizes. Game Success and Concept Encoding scores are given in %, Language Convergence and Topographic Similarity in scores between the range from 0 to 1. Entries are the averages across all seeds of a community size along with the standard deviation across seeds.*

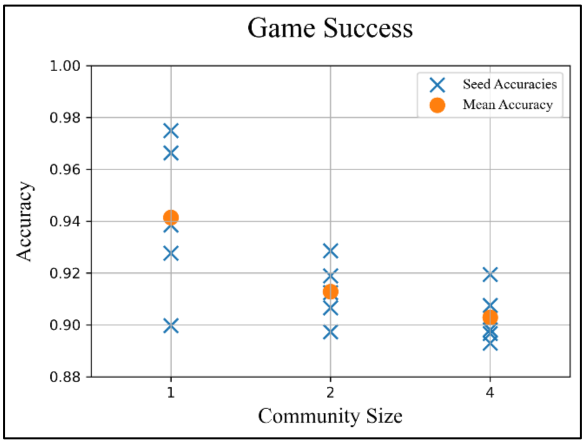

Figure 3: *Game success achieved in all seeds of all community sizes and their respective means. Scores are averages of the accuracies achieved by all pairs of a community on the test data.*

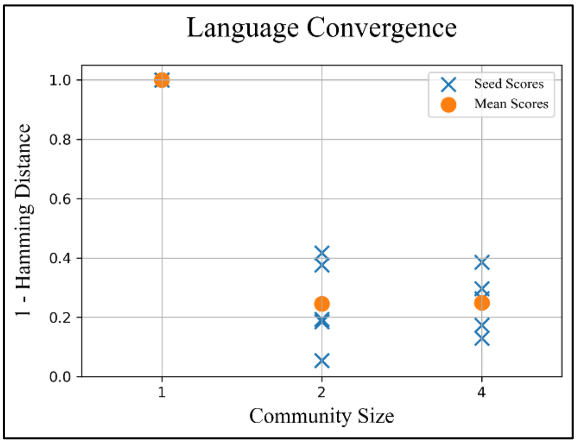

Figure 4: *Language convergence computed as the average message similarity between all messages produced given the same input. Message similarities are computed as 1 – Hamming Distance. The figure includes average message similarity for each individual seeds of a community size and the respective mean.*

protocol that enabled them to play the game with an accuracy above 90%. Communities of size one, i.e., communities with only a single sender-receiver pair, were successful in 94.15% of their game runs. Accuracies over all seeds with the community size one have a standard deviation of 2.71. When the community size is increased to two, pairs achieve an accuracy of 91.28% with a standard deviation of 1.06. In the setting with community size four, pairs achieve an accuracy of 90.3% with a standard deviation of 0.96. The correlation between community size and game success is found to be negative with a coefficient of -0.606 and significant with $p = 0.0166$.

5.2. Language Convergence

To evaluate the degree to which agents of a community agree on a common language, I report convergence scores based on the Hamming distance between messages of different senders when given the same input. A detailed description of the language convergence score is given in section 3.4.2. As there is only a single sender in a community of size one, this setting constitutes the ideal case of language convergence, i.e., the same input always results in the same message. Consequently, the language convergence score is set to its maximum,

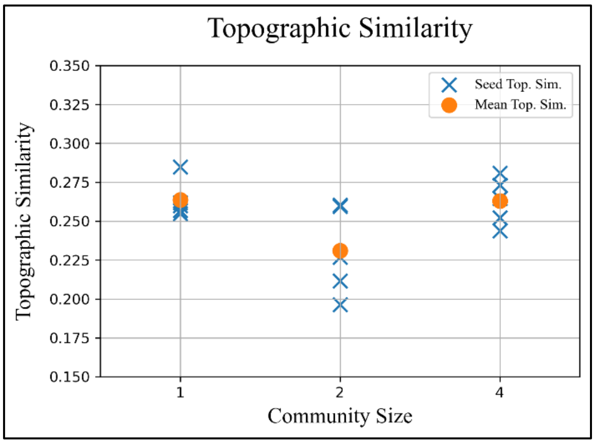

Figure 5: *Topographic similarity between input and message space for all seeds per community size and their respective mean. Reported scores are averages of topographic similarities of all senders in a community.*

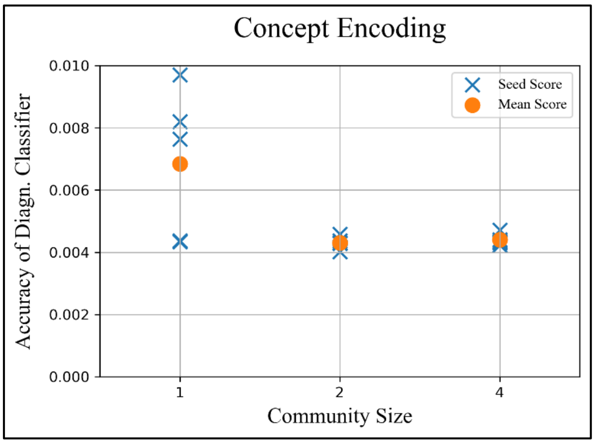

Figure 6: *Concept encoding scores of all seeds of all community sizes and the mean across seeds. The concept encoding score of a seed is the average accuracy of all diagnostic classifiers in predicting the class of the target image from the message produced to describe it by the sender.*

which is 1.0, for all seeds. Because of that, the standard deviation is 0 across all seeds. Communities of size two have a language convergence score of 0.2456 and show a standard deviation of 0.1338 across seeds. Communities of size four have a language convergence score of 0.2495 with a standard deviation 0.0906 across all five seeds. The individual language convergence scores of all seeds as well as the reported averages can be seen in Figure 4. Regarding the correlation between community size and language convergence, a negative coefficient of -0.7282 is found with a significant *p*-value of 0.0021.

5.3. Topographic Similarity

In my study, topographic similarity acts as a proxy for the compositionality of the emerged languages. The observed topographic similarities are plotted in Figure 5. Communities of size one have a topographic similarity of 0.2637 and show a standard deviation of 0.0109 across the five seeds. Communities of size two exhibit a topographic similarity of 0.2310 with a standard deviation of 0.0256. Communities of size four have a topographic similarity of 0.2629 with a standard deviation of 0.0135. While a small positive correlation is found between the community size and the topographic similarity, the coefficient of 0.1101 has a non-significant *p*-value of 0.6961.

5.4. Concept Encoding

Diagnostic classification is used to determine whether agents communicate about high level concepts of the images like the object classes depicted in them. Indeed, diagnostic classifiers for communities of all sizes are able to use messages to classify the ILSVRC2012 class of the respective target image with an accuracy that is well above chance. Since ILSVRC2012 contains images of 1,000 classes, the chance baseline for classifying a message's underlying ILSVRC2012 class is 0.1%. The accuracies achieved by the diagnostic classifiers for all seeds of each community size are plotted along with the community size average in Figure 6. For communities of size one, diagnostic classifiers achieve an accuracy of 0.68% which is almost seven times higher than the chance baseline. Across seeds, accuracies show a standard deviation of 0.21. Diagnostic classifiers for communities of size two exhibit the lowest performance with an accuracy of 0.43% and a standard deviation of 0.02. Based on messages from communities of size four, diagnostic classifiers are successful in 0.44% of the cases. With community size four, accuracy shows a standard deviation of 0.02. Given the seeds of all community sizes, the correlation coefficient between the community size and the accuracy of the diagnostic classifiers is found to be -0.5039 with a non-significant *p*-value of 0.0554.

6. Discussion

In this section, I conduct a critical discussion of the results presented in the previous section. I start with reviewing the results presented in the previous section and whether they reveal any community size effect. Afterwards, I contextualize my findings with those of previous studies. I continue by describing the strategy of co-adaptation that was found to be followed by agents in earlier studies. I show that this is also the case in my experiments and discuss the implications this has. Lastly, I point out the limitations of my study and how future research can contribute to better understanding the results obtained in my study.

6.1. Community Size Effects on Game Success & the Emerging Language

While significant correlations with the community size are found for game success and language convergence, one can see from the respective figures that only game success continuously decreases with increasing community sizes. For language convergence, the decrease from community size one to community size two is followed by a slight increase with community size four. Considering this, the correlation between community size and language convergence should be interpreted critically. It is likely that this correlation is significant only because of the large metric differences between community size one, and two and four. Therefore, the found correlation would likely not hold when one includes more samples from larger community sizes. An interesting observation is that the metrics for language convergence, topographic similarity and concept encoding all follow the same pattern: After an initial decrease from community size one to community size two, they increase with community size four, be it with differing strength. These increases from community size two to community size four take place despite the decrease of overall game success. Due to computational limitations (see section 6.4), I am unfortunately not able to investigate whether this trend between community size two and four would continue for larger community sizes.

6.2. Comparison with Previous Studies

The large number of variables in a game setup like the one employed in my study, makes for a limited comparability between results from different studies. Even when looking only at studies that used a similar game design, there are still many factors that can influence both game success and the emerging language. For example, the agents in the study by Havrylov and Titov (2017) have an accuracy below 90% when playing a very similar game. However, their receiver has a much more difficult task because it is confronted with 127 distractor images compared to only two in my setting. On the other hand, they allow their sender to produce a message with symbols from a vocabulary of size 10,000 while my vocabulary has an effective size of nine. Because of these vast differences in study designs, I compare my findings to two studies of the community size effect in language games that are closest to my approach: Kim and Oh (2021) and Tieleman et al. (2019). While there are still differences to the game setup of my study, both of these studies probe the community size effect in isolation and include experiments with natural images.

Regarding the relation between community size and game success, my results confirm findings by Tieleman et al. (2019) who also report that larger community sizes led to higher errors in their reconstruction game. Kim and Oh (2021) find no such relation and report stable accuracies with increasing community size. Considering that the discrimination game used by Kim and Oh (2021) is closer to my study design, this is surprising. A possible reason might lie in the details of the game mechanics. In the setup by Kim and Oh (2021), agents are not assigned a fixed role. Instead, all agents can act as a sender or receiver and their role is assigned at random in the beginning of a training epoch. It is possible that training agents as both, sender, and receiver, leads to a better language understanding that is more robust towards the challenges caused by increasing community sizes. However, this is purely speculative and game success can be affected by many other variables. It is impossible to determine without further analysis what exactly are the differences that cause our contradicting findings on the community size effect on game success.

Regarding the community size effect on language convergence, my findings support the ones obtained by Kim and Oh (2021): Larger communities lead to larger variety in languages used by senders. Interestingly, the overall language agreement is much lower in my study compared to the one by Kim and Oh (2021). For community sizes larger than one, they record message similarities among senders between 0.75 and 0.95 compared to the values under 0.25 recorded by me. This is despite the fact that we use the same metric for measuring language convergence. I will discuss the low language convergence and a likely reason for it in more detail in the following section. Tieleman et al. (2019) do not analyze language convergence in their setting.

Since neither Kim and Oh (2021) nor Tieleman et al. (2019) investigate the community size effect on topographic similarity, my finding that there is no clear relation constitutes the first observation regarding this effect. Tieleman et al. (2019) report results suggesting that increasing community size improves senders' ability to capture human level concepts in their messages. My findings do not support this result. Indeed, diagnostic classifiers for community size one are about 50% more capable in identifying the class of an image based on its respective message compared to those trained with messages from community size two and four. Again, identifying the exact reasons would require an additional in-depth analysis.

6.3. Co-Adaptation of Agents

Although Tieleman et al. (2019) do not investigate the community size effect on language convergence, they make an interesting observation that might explain why the communities in my setup do not develop a common language to the same degree as in Kim and Oh (2021). Tieleman et al. (2019) observed that senders, especially when confronted with images from more diverse datasets, follow a strategy in which they individualize their messages in a way that allows receivers to identify from which sender a message stems. This allows the receivers to interpret a message differently depending on which sender produced it. When agents follow this strategy, it effectively prevents communities from learning a common language and instead leads to co-adaptation of receivers to all senders. To test whether this co-adaptation occurs also in my setup, I again use diagnostic classifiers (Hupkes et al., 2018), namely sender classifiers. These sender classifiers receive messages as input and try to classify from which sender this message was produced. For each seed of all community sizes

Community Size	Sender Decoding Accuracy
2	95.83 ± 2.99
4	90.62 ± 4.79

Table 2: *Accuracies achieved by the diagnostic sender classifiers in predicting the identities of senders given messages given in %. Entries are averages across all seeds of a community size along with the standard deviation across seeds.*

one diagnostic classifier is trained using half of the messages in the evaluation data. Once trained, the sender classifiers' performance on the remaining half of the messages from the evaluation data is recorded. This procedure is done for each seed of all community sizes larger than one. The architecture of the sender classifiers and the training parameter are identical to those used for the diagnostic classification performed to evaluate concept encoding (see section 3.4.4). The average sender classification accuracies across seeds are listed in Table 2. The high accuracies above 90% indicate that senders do indeed encode their identity in the messages. The fact the agents follow this strategy of co-adaptation is likely the reason for the low language convergence shown by the communities in my setup. In addition, it could explain the differences between the community size effects found in my study and those found by Kim and Oh (2021) and Tieleman et al. (2019) where co-adaptation takes place to a lesser degree or is prevented via an adversarial loss.

If larger communities cope with the task of successfully playing a communication game by developing languages distinct to each sender, then communication learning in communities is basically the same as learning for a single pair with the added task of correctly encoding and decoding sender identity in the messages. To determine whether co-adaptation is indeed influencing the recorded community size effects in that way, a comparison to results from a setting in which co-adaptation is prevented is necessary. Tieleman et al. (2019) suggest preventing co-adaptation by introducing an adversarial loss based on a fixed diagnostic classifier that was trained to distinguish senders. However, this procedure failed to achieve the desired effect in my setup. This is to be expected as a fixed diagnostic classifier can only recognize a specific encoding scheme used by senders to convey their identity. The flexibility of senders to develop a new identity encoding scheme during their training procedure over epochs allows them to enable co-adaptation and at the same time keep the adversarial loss low. Tieleman et al. (2019) do not report the degree to which this method reduces co-adaptation in their setting. Interestingly, agents show less signs of co-adaptation in the study by Kim and Oh (2020) although they do not make use of any measures to prevent it. Considering that they use a dataset of artificial images and natural images from the small CIFAR10 dataset, this might confirm the observation by Tieleman et al. (2019) that agents resort to co-adaptation when confronted with diverse datasets like ILSVRC2012 used in my setup.

6.4. Limitations

The fact that agents of larger communities follow the strategy of co-adaptation limits the interpretability of the community size effects found in my setting. It would be an interesting task for future research to develop a procedure for effectively preventing co-adaptation and probing whether the found community size effects then better align with the results of previous research. My study is further limited by the available computational resources which allowed for the analysis of communities of only limited sizes. Larger sizes would have required unfeasible training times; training a community of size four took me over 16 hours, even without recording the community’s performance between epochs. Including larger communities in the experiment would have allowed for more robust conclusions about community size effects. In addition, including more large communities would allow to probe whether the observed positive trend between community size two and four would have continued. This also could be an interesting objective of future research.

7. Conclusion

In this study, I conducted an in-depth investigation of the community size effect on language emergence in multi-agent communication games. To do so, I trained agent communities of differing sizes to play a cooperative game that requires agents to communicate about images. While similar game setups are widely used in previous studies, my study is the first to make use of images from a dataset as complex as ILSVRC2012. In an analysis of the languages that emerged during training, I found that increasing community size leads to a reduction in game success and the emergence of several agent specific languages rather than one common language. Further, I found that messages produced in communities with more than one agent-pair show less topographic similarity and encode less information about the high-level concepts present in the input images. While my results mostly differ from the findings of previous studies, the large number of variables of such experimental setups make for a very limited comparability across studies. Nonetheless, to better understand emergence of language in my setup, I drew inspiration from observations about co-adaptation of agents made by Tieleman et al. (2019). An exploratory analysis revealed that the agents in my setup indeed also follow a strategy of co-adaptation that prevents the development of one common community language. Considering that positive community size effects may require the development of a common community language, an extended study design in which co-adaptation is prevented would be of great interest. However, a technique for preventing co-adaptation suggested by Tieleman et al. (2019) proved to be unsuccessful in solving this problem in my setup. Consequently, I see it as an important topic of future research to introduce a strategy that effectively prevents co-adaptation and facilitates a community's development of a common language also when using discrete, sequential, and variable length. For now, my study shows that just increasing community sizes does not have any positive effects on the language emerging in a multi-agent discrimination game that uses diverse and natural images. In fact, compared to having only a single agent pair, it leads to the emergence of several sender specific languages that show less topographic similarity, less ability to encode high-level concepts and reduced success in agent communication.

8. Data Availability

The datasets described in section 4.1.2. can be found together with scripts for training, evaluation, and exploratory analysis in the study's GitHub repository[2].

[2] https://github.com/jumorisse/Community-Size-Effect-on-Language-Emergence-in-Multi-Agent-Games

9. References

Bahdanau, D., Cho, K., & Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. *3rd International Conference on Learning Representations, ICLR 2015*. http://arxiv.org/abs/1409.0473

Batali, J. (1994). Innate biases and critical periods: Combining evolution and learning in the acquisition of syntax. *Artificial life IV* (pp. 160-171).

Cho, K., van Merrienboer, B., Gülçehre, Ç., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using rnn encoder-decoder for statistical machine translation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014.* https://doi.org/10.3115/v1/d14-1179

Choi E., Lazaridou, A. & De Freitas N. (2018). Compositional obverter communication learning from raw visual input. *Proceedings of the 6th International Conference on Learning Representations (ICLR).*

Cogswell, M., Lu, J., Lee, S., Parikh, D., & Batra, D. (2019). Emergence of Compositional Language with Deep Generational Transmission. *CoRR*, http://arxiv.org/abs/1904.09067

Dagan, G., Hupkes, D., & Bruni, E. (2021). Co-evolution of language and agents in referential games. *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, EACL 2021.* https://doi.org/10.18653/v1/2021.eacl-main.260

Dubova, M., Moskvichev, A., & Goldstone, R. L. (2020). Reinforcement communication learning in different social network structures. *CoRR.* https://arxiv.org/abs/2007.09820

Dubova, M.A., & Moskvichev, A. (2020). Effects of supervision, population size, and self-play on multi-agent reinforcement learning to communicate. *Proceedings of the ALIFE 2020: The 2020 Conference on Artificial Life.*

Foerster, J. N., Assael, Y. M., de Freitas, N., & Whiteson, S. (2016a). Learning to communicate with deep multi-agent reinforcement learning. *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information*

Processing Systems 2016. https://proceedings.neurips.cc/paper/2016/hash/c7635bfd99248a2cdef8249ef7bfbef4-Abstract.html

Foerster, J. N., Assael, Y. M., de Freitas, N., & Whiteson, S. (2016b). Learning to communicate to solve riddles with deep distributed recurrent q-networks. *CoRR*. http://arxiv.org/abs/1602.02672

Giles, C. L., & Jim, K.-C. (2003). Learning communication for multi-agent systems. *Innovative Concepts for Agent-Based Systems* (pp. 377–390). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-45173-0_29

Graesser, L., Cho, K., & Kiela, D. (2019). Emergent linguistic phenomena in multi-agent communication games. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*.

Hasida, K., Nagao, K., & Miyata, T. (1995). A game-theoretic account of collaboration in communication. *Proceedings of the First International Conference on Multiagent Systems.* The MIT Press.

Havrylov, S., & Titov, I. (2017). Emergence of Language with Multi-agent Games: Learning to Communicate with Sequences of Symbols. *Advances in Neural Information Processing Systems 30 (NIPS 2017).* https://proceedings.neurips.cc/paper/2017/hash/70222949cc0db89ab32c9969754d4758-Abstract.html

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016*. https://doi.org/10.1109/CVPR.2016.90

Hupkes, D., Veldhoen, S., & Zuidema, W. H. (2018). Visualisation and 'diagnostic classifiers' reveal how recurrent and recursive neural networks process hierarchical structure. *Journal of Artificial Intelligence Research., 61,* 907–926. https://doi.org/10.1613/jair.1.11196

Jorge, E., Kågebäck, M., & Gustavsson, E. (2016). Learning to play guess who? And inventing a grounded language as a consequence. *CoRR*. http://arxiv.org/abs/1611.03218

Kajic, I., Aygün, E., & Precup, D. (2020). Learning to cooperate: Emergent communication in multi-agent navigation. *Proceedings of the 42th Annual Meeting of the Cognitive Science Society – Developing a Mind: Learning in Humans, Animals, and Machines, CogSci 2020.* https://cogsci.mindmodeling.org/2020/papers/0459/index.html

Kasai, T., Tenmoto, H. & Kamiya, A. (2008). Learning of communication codes in multi-agent reinforcement learning problem. *2008 IEEE Conference on Soft Computing in Industrial Applications*. https://doi.org/10.1109/SMCIA.2008.5045926

Keresztury, B., & Bruni, E. (2020). Compositional properties of emergent languages in deep learning. *CoRR*. https://arxiv.org/abs/2001.08618

Kharitonov, E., Chaabouni, R., Bouchacourt, D., & Baroni, M. (2019). EGG: A toolkit for research on emergence of language in games. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019.* https://doi.org/10.18653/v1/D19-3010

Kim, J., & Oh, A. (2021). Emergent communication under varying sizes and connectivities. *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021.*

Kottur, S., Moura, J. M. F., Lee, S., & Batra, D. (2017). Natural language does not emerge ‘naturally’ in multi-agent dialog. *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017. https://doi.org/10.18653/v1/d17-1321*

Lazaridou, A., Hermann, K. M., Tuyls, K., & Clark, S. (2018). Emergence of linguistic communication from referential games with symbolic and pixel input. *6th International Conference on Learning Representations, ICLR 2018.* https://openreview.net/forum?id=HJGv1Z-AW

Lazaridou, A., Peysakhovich, A., & Baroni, M. (2017). Multi-Agent cooperation and the emergence of (natural) language. *5th International Conference on Learning Representations, ICLR 2017*. https://openreview.net/forum?id=Hk8N3Sclg

LeCun, Y., Bengio, Y., & Hinton, G. E. (2015). Deep learning. *Nature*, *521*(7553), 436–444. https://doi.org/10.1038/nature14539

Lewis, D. (1969). *Convention: A philosophical study*. Harvard University Press.

Li, F., & Bowling, M. (2019). Ease-of-Teaching and language structure from emergent communication. *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019*. https://proceedings.neurips.cc/paper/2019/hash/b0cf188d74589db9b23d5d277238a929-Abstract.html

Lowe, R., Foerster, J. N., Boureau, Y.-L., Pineau, J., & Dauphin, Y. N. (2019). On the pitfalls of measuring emergent communication. *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS '19*. http://dl.acm.org/citation.cfm?id=3331757

Luna, D. R., Ponti, E. M., Hupkes, D., & Bruni, E. (2020). Internal and external pressures on language emergence: Least effort, object constancy and frequency. *Findings of the Association for Computational Linguistics: EMNLP 2020*. https://doi.org/10.18653/v1/2020.findings-emnlp.397

MacLennan, B. J., & Burghardt, G. M. (1993). Synthetic ethology and the evolution of cooperative communication. *Adaptive Behavior.*, *2*(2), 161–188. https://doi.org/10.1177/105971239300200203

Mordatch, I., & Abbeel, P. (2018). Emergence of grounded compositional language in multi-agent populations. *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18)*. https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/17007

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M. S., Berg, A. C., Fei-Fei, L. (2015). ImageNet large scale visual recognition challenge. *International Journal of Computer Vision., 115*(3), 211–252. https://doi.org/10.1007/s11263-015-0816-y

Steels, L. (1998). Synthesising the origins of language and meaning using co-evolution, self-organisation and level formation. In J. R. Hurford, M. Studdert-Kennedy, & C. Knight (Eds.), *Approaches to the evolution of language* (pp. 384–404). Cambridge University Press.

Sukhbaatar, S., Szlam, A., & Fergus, R. (2016). Learning multiagent communication with backpropagation. *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016.* https://proceedings.neurips.cc/paper/2016/hash/55b1927fdafef39c48e5b73b5d61ea60-Abstract.html

Tieleman, O., Lazaridou, A., Mourad, S., Blundell, C., & Precup, D. (2019). Shaping representations through communication: Community size effect in artificial learning systems. *CoRR*, http://arxiv.org/abs/1912.06208.

Vinyals, O., & Le, Q. V. (2015). A neural conversational model. *CoRR*. http://arxiv.org/abs/1506.05869

Wagner, K., Reggia, J. A., Uriagereka, J., & Wilkinson, G. (2003). Progress in the simulation of emergent communication and language. *Adaptive Behavior, 11*(1), 37–69. https://doi.org/10.1177/10597123030111003

Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning, 8,* 229–256. https://doi.org/10.1007/BF00992696

Wittgenstein, L. (1953). *Philosophical investigations*. Wiley-Blackwell.

10. Appendix

10.1. Declaration of Authorship

I hereby certify that the work presented here is, to the best of my knowledge and belief, original and the result of my own investigations, except as acknowledged, and has not been submitted, either in part or whole, for a degree at this or any other university.

____

Signature

____

City, Date

10.2. Declaration of Equality between Versions

I hereby declare that the PDF version and the printed version are identical and do not differ in any way.

____

Signature

____

City, Date