Universiteit Leiden
The Netherlands

Opleiding Informatica

Learning Signal-Meaning Mappings in an Emergent Communication System using Neural Networks

Thomas Blom

Supervisors:
Tom Kouwenhoven
Tessa Verhoef

BACHELOR THESIS

Leiden Institute of Advanced Computer Science (LIACS)
www.liacs.leidenuniv.nl
20/08/2022

Abstract

Explaining the emergence of human communication poses a problem because how do individuals come to an agreement on the relationship between a symbol and its meaning when this relationship is not innate? Previous work provides some insights on how this emergence occurs in humans with the use of the Embodied Communication Game (ECG). This study focuses on the emergence of communication between humans and machines. We investigate the ability of a neural network with LSTM layers to map the meaning of signals produced by humans playing the ECG. We use the sequences of movements of 23 duo's playing the game, after each game each player reported if communication had been established. The network needs to predict the color signaled by a player using the sequences of movements performed by a player in a round of the game. The accuracy of our model is significantly higher on data from players that established communication compared to players that did not ($t=6.00$, $p=3.35e-07$). The accuracy of our model depends on the percentage of successful rounds played in a game ($r=0.90$, $p=1.95e-17$). Furthermore, we can see a correlation between the model's accuracy and the similarity of the player's and model's color distribution. In general, a more accurate model produces a color distribution similar to the color distribution in the target labels ($r=-0.83$, $p=6.99e-13$). Less accurate models have a bias toward the color with the highest frequency in the target labels $r=-0.69$, $p=9.48e-08$). To conclude, our model is successful in mapping the meaning of signals produced in an emergent symbolic communication system in the ECG.

Contents

1 Introduction

The question of how human communicative behavior has emerged is a difficult question to answer and widely debated. This is because human language is both learned and symbolic (symbols are used to represent concepts)[5]. This poses a problem; how do individuals come to an agreement on what the relationship is between a symbol and its meaning when this relationship is not innate [16]? Furthermore, before the receiver of a communicative signal can even make a connection between a signaled symbol and its meaning, he first must understand that the signal is indeed communicative. The symbolism in human communication consists of body language like gestures and facial expressions, as well as words or vocal moans. The symbol signals the meaning, you can for example explain to someone that you feel angry by telling them. The word "angry" now signals your emotional state to that person, and this person is now able to understand how you feel. A dog can also signal that it is angry but only using non-verbal signals like showing its teeth and growling. The second property of human language is that this symbolism can be learned [5]. These two properties of the human communication system have contributed to the human language that is used today. The communicative processes that interleaved the cognitive abilities and experiences of many individuals over time have resulted in today's shared set of communicative behaviors and conventions [24, 11]. Tomasello [24] argues that the set of cognitive capacities, of which some are unique to humans has resulted in a symbol-based culture. Examples of these are the human ability to recognize that others have consciousness of their own and understand someone's intentions and emotions. These capacities resulted in the ability to learn from others by imitating their actions and to inform others by teaching. This caused cultural artifacts to emerge and be passed along with each new generation. A common approach to come to an understanding of human language evolution and emergence apart from experiments on humans is the use of computer modeling and simulation. The use of agent-based simulation models can potentially help explore the question of how communicative behavior has originated. Quinn [17] developed such a model where the transition from non-communicative to communicative behavior between agents is modeled. In his model robots are equipped with proximity sensors and controlled by a neural network that controls the movement of the agent to perform a coordinated task. The agent has no predefined communication channel, so the behaviors are initially random. However, these random behaviors evolve to become a coordinated signaling system with a leader and follower role. The model of Quinn [17] shows that the emergence of cooperative signals between computational agents without a dedicated communication channel is possible. Scott-Phillips et al. (2009) [20] did a comparable investigation where two individuals needed to communicate their intentions without a pre-defined communication channel. Thus, the researchers could in both cases investigate how communicative behavior emerged de novo. While the human [4][6][7][8][21] and agent-based [10][14][15][22] emergence of communication is well researched, only Steel's Talking Heads experiment investigated the emergence of vocabulary between humans and embodied agents [23]. The agents had no prior categorization of the word and no programmed language, instead, it emerges. By interacting with humans in the real world and each other they construct this by themselves. A novel language-like system emerges through interaction with humans. The agents however are pre-programmed with a basic cognitive architecture, based on plausible biological theories. We propose to revisit the emergence of communication between humans and machines with the use of a literature study on how this can be achieved as well as an experiment using the Embodied Communication Game (ECG) designed by Scott-Phillips et al. (2009) [20]. We use data gathered by Tom Kouwenhoven et al. (2022) [13] of participants playing the ECG. The ECG is interesting since the participants have no pre-defined communication channel and they need to solve the problem of recognizing a signal and mapping its meaning by themselves since the space of possible signals is not defined. However, designing an algorithm that can recognize a signal is out of scope for this project and will only be discussed in a literature study. We focus on defining a communication channel between humans and machines using a neural network. The goal of the network is to learn the mapping between a sequence of different movements and their meaning. Therefore, the research question is; Is it possible for a neural network to map the meaning of signals in an emergent symbolic communication system between two humans in ECG? In the upcoming sections, background information including the necessary algorithms and theory that we use for our experiments are discussed. Furthermore, we discuss related work on the emergence of communication between humans, agents, and humans and embodied agents. Then, the methods we use to perform our experiments as well as their results are described and discussed.

2 Background

This section discusses the necessary research, algorithms, and theory. This background information includes the Embodied Communication Game (ECG), neural networks, RNNs and LSTMs. Elements discussed in this section will be used in our neural network model and the experiments.

2.1 Embodied Communication Game

The Embodied Communication Game (ECG) [20] created by Scott-Phillips et al. (2009) is a simple cooperative game in which two players have a common goal. The two players are separated and thus, unable to communicate using the normal communication channels. Instead, players need to learn to communicate through movements in a 2D game. Each player is represented as a stickman in his own 2x2 box. Within this 2x2 box, the player can move freely, but only to the center of each quadrant. Each quadrant of the box is randomly colored either blue, red, green, or blue. Each player can see both 2x2 boxes and the movements within them but can only see the colors of the quadrants in his own box, see figure 1. The game consists of multiple rounds, after each round the players can see both boxes in color, the location of both players in the box, and their score. Only if both players end up on a quadrant with the same color, do they receive a point. There is always one mutual color between the two boxes, so the players are always able to score a point. The final score of the pair of participants is not the total points received, but instead the longest run of consecutive wins. This prevents a pair of players to perform well just by playing a lot of games and instead focuses on the reliability of the communication between the players. Using this setup, ECG ensures that there is no pre-defined communication channel. While there is only one possible channel to communicate through (movement), this channel is not pre-defined since the communicative behavior still must be embodied within this channel. The participants need to solve the problem of signaling signalhood by themselves since the space of possible signals is not defined. Scott-Phillips et al. found that there is an importance of common ground in language emergence. These assumptions are made since successful pairs typically establish a default color strategy. At some point the pairs will end up on a same-colored quadrant and score a point, moving to this color for the next rounds then becomes the default strategy. When this default color is not available for a player in an upcoming round, particular movements emerge to signal this absence. These movements are then later linked to a different color. This way a mapping emerges of movement sequences to color. This process can be seen in figure 2. Most players report that the emergence of communication in ECG is not dedicated to one player, instead, it follows a process of communication between the two players that form the mapping of signals to color. An important conclusion that Scott-Phillips et al. make is that the systems that emerge between two players do not resemble a system that would have emerged from a single player. The process of emergence fundamentally affects the form of the final communicative system. A system that is created by a single player typically has quite a different form where for example, the number of movements is associated with a color. The findings suggest that the constraint on the embodiment of the communication system shapes its final form.

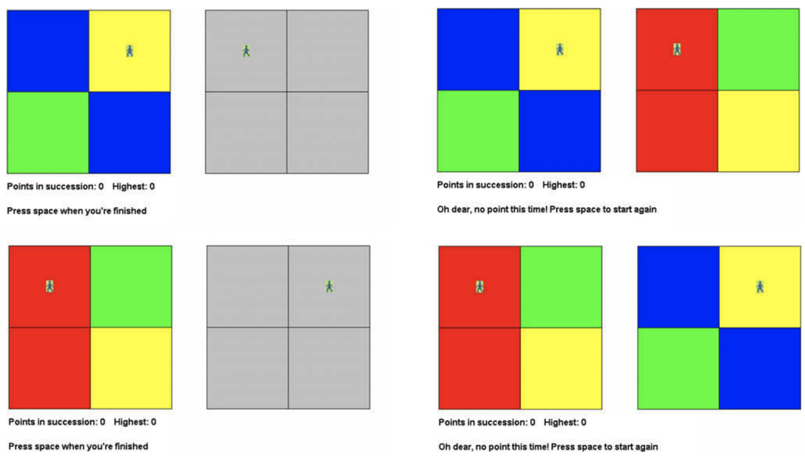


Figure 1: Screen-shots of the two players for one round of the ECG (image from Scott-Phillips et al. [20]). Each row shows the view of one player during a game (left-hand side) and after a round is finished (right-hand side).

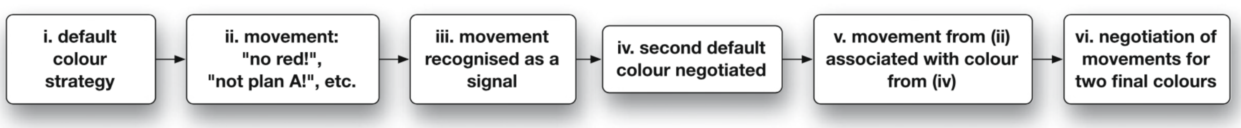


Figure 2: Typical process of communication emergence between two successful players in the ECG. (image from Scott-Phillips et al. [20]).

2.2 Neural Networks

A neural network [18] can be categorized as a machine learning algorithm, in which the computer learns to perform a task by training on data. This training data is usually hand-labeled. For example, if you would want to train an image classifier that can classify images of animals into the correct species. You would train the system on a set of labeled images of animals. During training, the system learns what pixel compositions and patterns correlate to what label. A neural network does this in a way that is modeled after the way neurons work in our brains. It uses simple nodes that are connected through weights. The exact composition of the neural network differs per application, but most are set up in a feed-forward manner and consist of multiple layers. A layer consists of nodes, see figure 3.

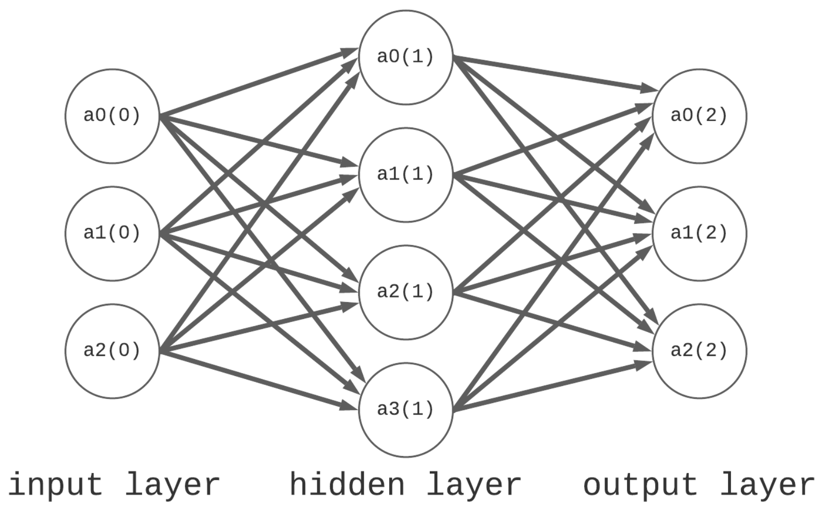


Figure 3: Layer types in a neural network.

A node assigns a weight (w) to each of its inputs, each input is then multiplied by its assigned weight, these products are then added together to a single number as well as a bias (b). This sum is then inserted into an activation function, for example a sigmoid function $\sigma()$ as we can see in equation 1.

$$a0(1) = \sigma(w_{0,0} * a0(0) + w_{0,1} * a1(0) + w_{0,2} * a2(0) + b0(1)) \tag{1}$$

Thus, each neuron can be seen as a function that takes in the outputs of all neurons in the previous layer and calculates a number between 0 and 1 as its output. This function is called the activation function. In the training phase of a neural network, the weights are initialized with random values. The training data is then fed into the input layer, from here it passes through all nodes in the hidden layers and eventually generates an output in the output layer. While training, the weights and biases are adjusted so input data with the same labels create similar outputs. This is done through backpropagation [12], for each training example the output of the network is compared to the desired output. The squares of the differences of each of the components of the output are then added up to calculate a cost. Averaging out the costs for all the training examples gives the total cost of the network. The goal is to minimize this cost by adjusting the weights and biases.

2.3 Recurrent Neural Networks

Feed-forward neural networks don't have any memory for the inputs they received. It only considers the current input when calculating an output since they do not allow information of previous inputs to persist in the network. Recurrent Neural Networks (RNNs) [19] address this issue. Humans also have some form of persistence of information in their memory. When for example reading an article, you understand each word you read based upon the understanding of the previous read words. You keep track of the information you already processed based on their importance to construct an understanding of the words that still need to be read, and eventually the entire article. Recurrent networks work similarly, they consist of multiple recurrent layers, each layer stores information in the weighted connections between the previous and next layers as well as a shared hidden layer. For example, if we would consider a phrase like "Wrapping your head around something". The order in which these words are placed is important for their meaning. Therefore, an RNN needs to keep track of the specific order of them. Figure 4 shows an example of an RNN, the "rolled" visualization shows the entire RNN. The "unrolled" visualization shows all the layers within the RNN. Each layer maps to a single timestep and a single word of that phrase. When the inputs have been "wrapping your head around", each of these words represents a layer. Predicting the next word "something" is aided by the propagation of information in the hidden layers to predict the output.

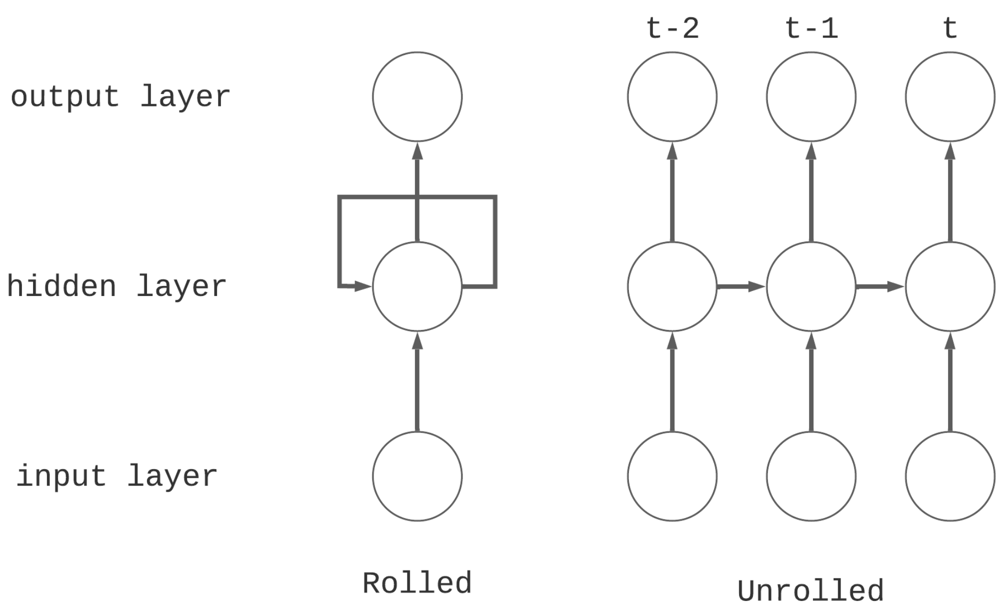


Figure 4: Rolled and unrolled visualisation of an RNN.

2.4 Long Short-Term Memory Networks

Long short-term memory networks (LSTMs) [9] expand upon recurrent neural networks. It is a special kind of RNN that is capable of long-term memory as well as the ability to regulate the addition or removal of information from the network. LSTMs add a state to the RNN, this state is made up of the LSTM cell. The LSTM cell uses three gates that control the flow of information in a sequence in the network. The forget gate controls what information in the state can be forgotten since it is no longer relevant. The input gate controls what information should be added to the state information. The output gate controls what part of the information stored in the state is outputted. How these gates are formed can be seen in figure 5. The managing of information by the gates depends on the importance that is assigned to the information through weights, which are also learned by the algorithm. The benefit of using an LSTM over a normal RNN is that it can capture potential long-distance dependencies in the data due to the introduced gates.

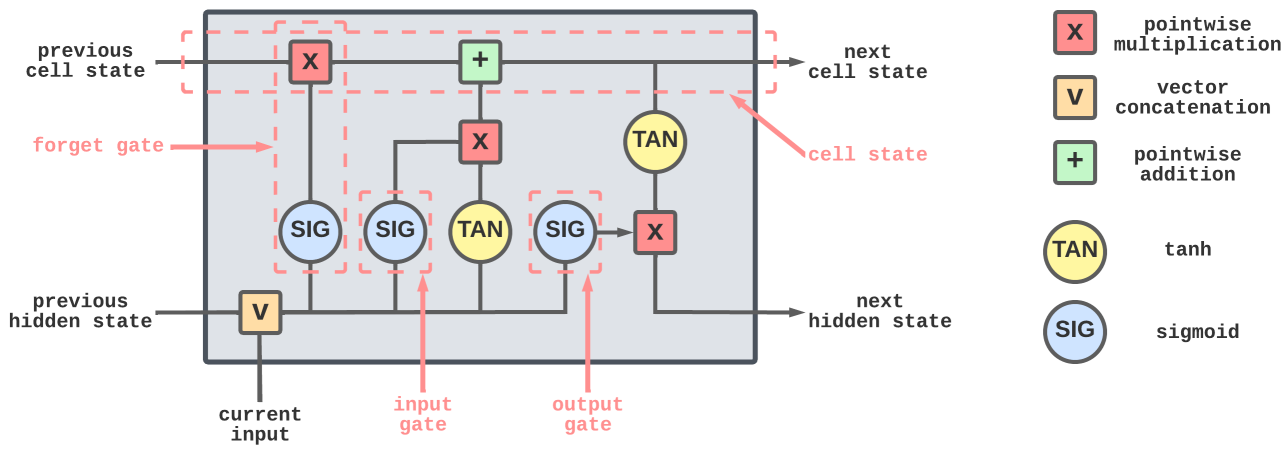


Figure 5: LSTM state cell.

2.5 K-fold cross-validation

A good machine learning model should not only give accurate predictions on the data it is trained on but also on new data and thus avoids overfitting. Overfitting occurs when the model is fitted to the training data too well. This causes the model to have high accuracy on the training dataset but perform poorly on new data. Testing if a model is overfitting can be done by splitting the dataset into two parts: a train set and a test set. The train set is used for training the model, and the test set for validating the model. The model is not overfitting when the loss on the train set is similar to that on the test set. However, this way of validating a model is not very robust. Since the split is done only once and the data in the test and train set can have a big effect on the results. A more robust way of validating the model is by using K-fold cross-validation. In K-fold cross-validation, K is a parameter that represents the number of equal-sized groups the dataset is split into. The value of K also represents the number of evaluation folds. In each fold the following procedure occurs with a different group:

1. Take this group as the test set
2. Take the remaining groups as the train set
3. Train a new model on the train set and evaluate it on the test set
4. Store the evaluation score

After this procedure has been performed on each of the K groups, the evaluation scores are averaged. This averaged score gives a more robust insight into the performance of the model and if it is overfitting.

3 Related Work

In this section, various findings in research into language emergence are discussed. We will discuss research on human-based and agent-based language emergence as well as the emergence of vocabulary between humans and machines in Steel's Talking Heads experiment [23].

3.1 Galantucci's communication game

Galantucci [7] developed a game to investigate how human communication systems emerge in the context of joint human activities. For his research, he did not want to rely on any pre-established language such as a natural language or a designed artificial language. Therefore, the general idea behind his method was the following. In the game, two participants share a two-dimensional virtual environment that consists of four connected rooms. The experimental setup can be seen in figure 6.

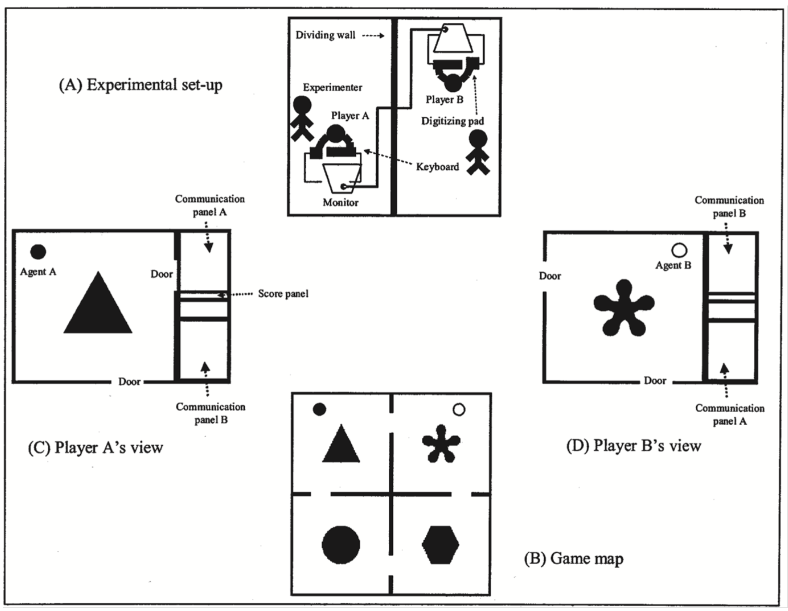


Figure 6: Overview of Galantucci's communication [7] game. (A) Experimental setup. (B) The four connected rooms in the virtual environment, each room is marked with its own icon. (C) Player A's view, showing player A's current location in the virtual environment, its own communication pad, the pad of player B, and the score. (D) Player B's view, showing player B's current location in the virtual environment, its own communication pad, the pad of player A, and the score. (image from Galantucci [7]).

Each of the rooms is connected to two other rooms and marked by a different icon. Participants can freely move in and between the rooms but can only see the room in which they are located and can see the other player if they are in the same room. The only way for the two participants to communicate with each other is by using a stylus on a small digitizing pad. Both players can see what was being written on these pads, however, writing on the pad is subjected to a constant downwards drift as well as a quick fading of what is written. This mostly prevents the participants from the use of common graphical symbols. The common goal of the two participants for each round is to end up in the same room while just changing rooms once. Thus, the participants need to signal to which room to go. The method showed that such communication systems emerged quickly and reliably. The participants use the writing pad to create a sign system for signaling each room. These sign systems can, however, depending on the pair, originate from a very different mapping of environment properties to the properties of the communicative signals. For example, some pairs used a numeration-based system while others used an icon or map-based system. While Galantucci's experimental setup is comparable to the ECG created by Scott-Phillips et al. [20] in many aspects, one key difference is that in the ECG there is no dedicated pre-defined communication channel. In Galantucci's experiment, there also emerges a new communication system between participants since they need to signal their intent. However, in the ECG, the channel through which this communication has to emerge needs to be discovered and defined by the participants themselves. While in Galantucci's experiment the communication channel is pre-defined by the researcher since there is a writing pad.

3.2 The Talking Heads experiment

As mentioned in the introduction, the Talking Heads (TH) experiment investigates the emergence of vocabulary between humans and embodied agents [23]. It was the first large-scale experiment where embodied agents created a shared vocabulary. The agents' shared environment consisted of a whiteboard in which various colored shapes were placed, as can be seen in figure 7. The agents play the "guessing game", one of the agents perceives an object in the scene, gives meaning to it, and verbalizes it into speech. The other agent hears these utterances and conceptualizes them into meaning, then picks out an object in its scene. The goal is to pick out the correct object. The agents could teleport through the internet to various physical sites around the world. Besides the agent's interaction with each other, humans could also interact with them, influencing the evolving vocabulary and understanding of the agents. The agents are pre-programmed with a basic cognitive architecture based on plausible biological theories. However, agents had no prior categorization of the world and no pre-programmed language. Instead, it emerges. In the experiments, the agents were able to learn all of the categories and linguistic structures in a usage-based way, by interacting with humans and each other. This shows that computers can learn linguistic meaning if this is constructed from the ground up. The agents can map the meaning of an object and even verbalize it. However, the agents are pre-programmed with basic cognitive architecture and the communication channel is pre-defined. While in the ECG there is no pre-defined communication channel, the Talking Heads experiment shows that it is possible for computers to map human signals to meaning and signal information by themselves. Furthermore, the experiment shows that it is possible for a novel communication system to emerge between humans and machines.

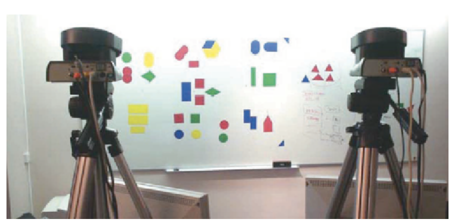


Figure 7: Talking Heads setup. (image from Steels [23])

3.3 Evolving communication with simulated Khepera robots

Similar to the experiment of Scott-Phillips et al. (2009) using the ECG, Quinn [17] presents a model to investigate how communicative behavior can evolve from initially non-communicative behavior that also ensures there are no dedicated pre-defined communication channels. However, while the experiment of Scott-Phillips et al. (2009) is performed on humans, Quinn's model uses computer agents (simulated Khepera robots). These agents are equipped with distance sensors and motorized wheels as can be seen in figure 8. Thus, they do not have a dedicated communication channel.

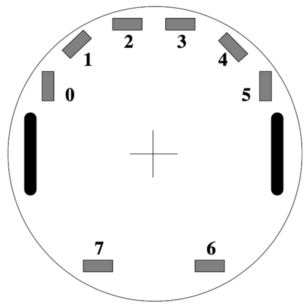


Figure 8: Khepera robot, consisting of 8 IR-sensors (numbered 0-7) and 2 driven wheels. (image from Quinn [17])

The agents are controlled by a neural network that turns the sensory input of the distance sensors into motor outputs. With these motor outputs, the agent is able to move through the environment. The agents are given a task in pairs. Their task is to move at least 25cm from their starting position within the given ten seconds. However, the agents are not allowed to leave each other's sensory range, nor collide with one other. The agents' performance is evaluated each round in pairs, thus the performance of the pair determines the performance of each agent. After each round, the agents received the same score. However, the fitness of an agent was determined by the average score received throughout its rounds played. Furthermore, each agent forms its own population. In total there are 180 populations with initially random genotypes, a generational evolutionary algorithm is used to evolve each population for a total of 2000 generations. Each pair is placed in an empty environment such that they are in sensor range of each other. In successful pairs, it is observed that in the early stages of the evolution a behavior emerged, which was still non-communicative. This non-communicative behavior forms the basis and later evolves toward communicative behavior in which leader and follower roles are allocated. However, this role allocation is not set in the agents' genotype. Instead, the allocation of roles happens during a round, and communicative signals are exchanged to coordinate this allocation. This interaction occurs due to a timing difference in when the agents would get aligned with each other as can be seen in figure 9. The agent that gets aligned first will oscillate, when the second agent also gets aligned it will interpret this oscillation as a signal and adopt the leader role. Without this signal, the agent would start oscillating itself. After both agents are aligned the leader will start moving backward while being followed by the other agent. The model has not set out to solve any particular hypothesis, it is only intended as a proof of concept. However, it shows that agent-based models can evolve communication without dedicated communication channels. Thus, this shows it might be possible for two simulated agents to evolve behavioral sequences that function as signal and response in the ECG as well. Since in the ECG there are also no pre-defined communication channels. If interaction with humans would be introduced in the evolution of behavior, it might also be possible to create a common communication channel between humans and computers. This would be similar to the Talking Heads experiment [23] but without the pre-defined communication channel and pre-programmed basic cognitive architecture.

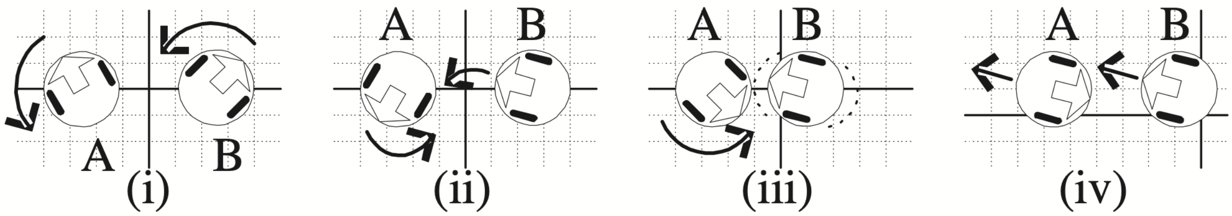


Figure 9: An example interaction between two successful Khepera agents. (i) Agents A and B are moving without being aligned (ii) Agent B aligns with agent A and stops. (iii) Agent B starts oscillating at this position. (iv) Agent A aligns with B and reverses, and agent B follows. (image from Quinn [17])

4 Methods

In this chapter, we describe the implementation of our neural network, the data, and the methods we use for our experiments.

4.1 Data

The current study investigates the possibility of learning Signal-Meaning of different movement sequences performed by a participant in the ECG using a neural network. The dataset we use is gathered by Tom Kouwenhoven et al. (2022) [13]. In their research into the relationship between a participant's Personal Need for Structure (PNS) and the emergence of a communicative system while playing the ECG is investigated. The dataset contains information on the games played by 46 participants (36 females, 10 males; Mage = 22.39, SDage = 3.52). The experiment took place in a similar setup used by Scott-Phillips et al. (2009) [20]. For each game, two participants took seat in different rooms. The participants played the ECG together in separate rooms using a web application on two connected computers. They were informed about how the game mechanics work and that the goal is to score as many consecutive points as possible in 40 minutes. After 40 minutes the game stopped, and the participants had to report if they thought any communication had emerged. The data we use to train our neural network consists of the keyboard inputs performed by the participants. The participants used the arrow keys to move between the quadrants and finalized their movements with the spacebar. Furthermore, we use the participants' different locations on the 2x2 grid for each round, and the color they ended up on.

4.2 Implementation

For the implementation of our neural network we use LSTM layers, since unlike a feed forward layer, an LSTM layer can process entire sequences of data. However, a standard recurrent layer is also capable of processing sequences of data. The benefit of using a LSTM is that it can capture potential long-distance dependencies in the data due to the introduced gates as mentioned in the background section. This can cause the performance of the neural network to increase and convergence to go faster [3]. To implement our neural network, there are several platforms available. We use TensorFlow [1] which is an open source and easy-to-use library that can be used to create machine learning models. Keras [2] is also a python based library and can be integrated into TensorFlow. Keras provides a more user-friendly interface and expands TensorFlows' functionality. Keras which is a high-level API has become fully integrated into the low-level TensorFlow API and is no longer a separate library. Therefore, we use both TensorFlow and Keras for creating our model. For every participant in the dataset, a new model is trained using the input data or set of samples. Each sample represents a round in the 40-minute game the participant has played. A sample consists of the sequence of actions and corresponding locations of the player in that round. There are 5 actions the player can perform; "left", "right", "up", "down", and "finish" which finalized the round. Furthermore, there are 4 different locations the player can move to, these are the centers of each quadrant. The target data or set of labels that are used to train the model consists of the color the player ends up on. The goal of the model is to predict the color on which the player ends up using the sequence of actions and locations the player used to get there. We chose to use this data since this is the same information a human player would get. Each player can see the movements, locations, and end color of the other player. Our model will act as a player guessing the color that a human player is signaling with its movements, these movements are the input it is receiving. The model should learn the mapping between the movements and the color the player tries to signal with those movements. However, not all pairs of participants were successful in establishing communication. Therefore, we do not expect the model to be accurate for all participants since they may not have tried to signal any color with their movements. Furthermore, if communication has been established between a pair, not all rounds of a player in that game contain a signal mapping to a color. For example, when a player is only receiving information from the other player, and then moving to the signaled color. Then the sequences of movements of the receiving player won't contain any information about the chosen color. Furthermore, in the first rounds of the game, when communication is not yet established, there may also not be any information contained in the movements.

4.3 Pre-processing of the data

As mentioned above, the input of the model consists of the movement and location sequences of the player. With this input, the model needs to predict the color that the player has ended up on. However, this data needs an embedding before the model can process it. Therefore, we use multi-hot encoding for encoding the input and one-hot encoding for the output. For each action in the sequence the move is encoded with 5 bits and the location with 4 bits, both using one-hot encoding. These one-hot encodings are concatenated to form a 9-bit multi-hot encoding of the action. The label or target data for each color is encoded with 4 bits using one-hot encoding. The specific encoding for each movement and location can be found in table 1.

Movement\Index:	0	1	2	3	4
Left	1	0	0	0	0
Right	0	1	0	0	0
Up	0	0	1	0	0
Down	0	0	0	1	0
Finish	0	0	0	0	1

Location\Index:	5	6	7	8
Top left	1	0	0	0
Top right	0	1	0	0
Bottom left	0	0	1	0
Bottom right	0	0	0	1

Table 1: The one-hot encodings for movement and location.

Each sample contains a sequence of states; however, the length of this sequence is not the same for each round. Some rounds of a game contain only the "finish" move while others contain many consecutive states. However, the model expects each sample to have the same number of sequences. Therefore, for each game, we take the length of the longest sequence and then pad all samples with value '9' to have this length. The model is instructed to "ignore" these padded values.

4.4 The Model

We use the sequential Keras model, which is a straightforward to use model that uses a list or stack of single-input, single-output layers. The loss function we use is the Categorical Cross Entropy Loss Function from Keras. We use Categorical Cross Entropy since our model has to perform multi-class classification. A representation of the model can be seen in table 2.

Layer	Input Shape	Output shape	Activation
Masking	(N,9)	(N,9)	-
LSTM	(N,9)	(N,16)	ReLu
Dropout	(N,16)	(N,16)	-
LSTM	(N,16)	(N,32)	ReLu
Dropout	(N,32)	(N,32)	-
TimeDistributed(Dense)	(N,32)	(N,4)	Tanh
Flatten	(N,4)	(NX4)	-
Dropout	(NX4)	(NX4)	-
Dense	(NX4)	(64)	ReLu
Dropout	(64)	(64)	-
Dense	(64)	(32)	ReLu
Dense	(32)	(4)	SoftMax

Table 2: The input and output shapes for the layers in our model.

Masking Layer:
The masking layer takes the input sequence of the model and masks all padded sequences containing 9 bits of value 9. This ensures that the following layers "ignore" the padded sequences.

LSTM layers:
The LSTM layers are used to learn long term dependencies in a sequence. The layer proccesses the sequence in chronological order. For each input, A new LSTM cell is created. The hidden state of this LSTM cell depends on the hidden state of the previous LSTM cell and the input. The LSTM layer outputs the hidden state of all LSTM cell's.

Time Distributed layer:
The Time Distributed layer takes the input sequence of hidden states as its input. It applies a dense layer to each of the unit sets in the sequence. The previous LSTM layer outputs a set of hidden states for each element in the sequence. The dense layer cosists of 4 units that are fully connected to the set of hidden states.

Flatten layer:
The flatten layer takes the rows of the two-dimensional shaped input and concatenates them to form a one-dimensional shaped output. This is done as preparation for the dense layers since they only allow for one-dimensional inputs.

Dense layers:
In total there are three dense layers, the first two were added to extract more detail from the LSTM layers. While the last layer gives a probability for each of the four colors. Before this final dense layer, only two dense layers were used since adding more did not improve the model's performance and only increased complexity.The first dense layer consists of 64 units with RELU activation functions that are fully connected to the 1D output of the flattened layer. The second dense layer consists of 32 units with RELU activation functions that are fully connected to the 64 units of the first dense layer.The final dense layer has a unit size of 4 with SoftMax activation. This layer gives a probability for each of the 4 colors, ranging from 0 to 1. The 4 probabilities together sum up to 1. The model predicts the color the player ended up on by choosing the color with the highest probability, given the input sequence.

4.4.1 Fitting the model

The model.fit function of Keras fits the model with the train data for 20 epochs. For each epoch, the entire train data is fed into the model, and the loss and accuracy are calculated. The weights are then adjusted to decrease the loss. Additionally, the model.fit function also calculates the loss and accuracy on the test set. This is not used for training the model, but for validating it. The loss and accuracy of the train set should be similar between the train and test set. Otherwise, the model is over- or under-fitting on the train set.

5 Experiments and Results

In our experiments, we try to demonstrate if our model can map the meaning of signals produced by participants in the ECG. We create and train a new model for each of the 46 participants to collect and compare their results. In this section, we describe what our experimental setup is and what results were obtained. The accuracy, loss, and historical data that is produced by our model is averaged for each player using 4-fold cross-validation. The samples of each player are first shuffled and then split into four equal-sized groups. Each group is used once for validation, the remaining three are then used for training. Thus, the model is trained and validated 4 times per player, generating 4 results. These results are then averaged for each player to be used in our experiments.

5.1 Experiment 1

With our second experiment, we investigate the ability of our neural network to learn signal-meaning mapping. Our neural network had to predict the signaled color using a sequence of movements. We discuss the average performance of the model over all players and the difference in the performance of the model on players that reported successful communication versus players that were unsuccessful. The model should perform worse on sequences of a pair that is unsuccessful in building a novel system of signals since these sequences will probably not contain any communicative signals the model can interpret. We expect that the neural network is more accurate in mapping the meaning of actions from participants that reported they were successful in creating a communication system in the ECG game. Since in an established communication system, the players can communicate with each other about what the end color should be. Thus, the action sequences in such a game should map to a specific color. However, we do not expect the model to be completely accurate for all of the sequences in an established communication system for 2 reasons:

1. No complete information

The model is trained on the action sequences of only one player in a two-player communicative game, the information the model receives is not complete. The sequences of actions in each round contain the communication of one player interacting with another player. Therefore, the communication is not complete. For example, in a round where player 1 "leads" the action sequences of this player signal a specific color. The other player may just react by moving to this color and finalizing the game. This player is following the instructions of the other player and thus not creating any signaling movements itself. The model does not have a complete overview of the interaction and may thus misinterpret an action sequence.

2. Shuffled dataset

The model does not learn the sequences in the same order the player created them in. Instead, the order of the sequences in the dataset is shuffled before training and testing. Therefore, there may be action sequences from rounds early in the game. While after the 40-minute game the participant reports that successful communication has been established with the other player, this may not be true in the early rounds of the game. In these early rounds, communication is not yet established. Thus, the sequences in these rounds may not contain any signaling information. Shuffling the data is common practice since it prevents the model from learning a pattern from the order in which the game is played instead of basing the prediction on solely a round's sequence.

5.1.1 Result

The model has a mean accuracy of 60.07% with a standard deviation of 21.82% (M=60.07%, SD=21.82%). The boxplot in figure 10 shows that the model has higher median accuracy on players that reported they were successful (72%) compared to the players that reported there was no communication or were not sure (28%). The mean accuracy of the model on the players that reported unsuccessful communication or were not sure is 30.39% with a standard deviation of 6.14% (M=30.39%, SD=6.14%). The mean accuracy of the model on players that reported they were successful at establishing communication is 67.29% with a standard deviation of 17.78% (M=67.29%, SD=17.78%). The accuracy of our model is significantly higher on data from players that established communication compared to players that did not (t=6.00, p=3.35e-07). This shows that the model is better at predicting the end color with sequences of players that reported they were successful. However, there is a wide range of accuracies for successful players.

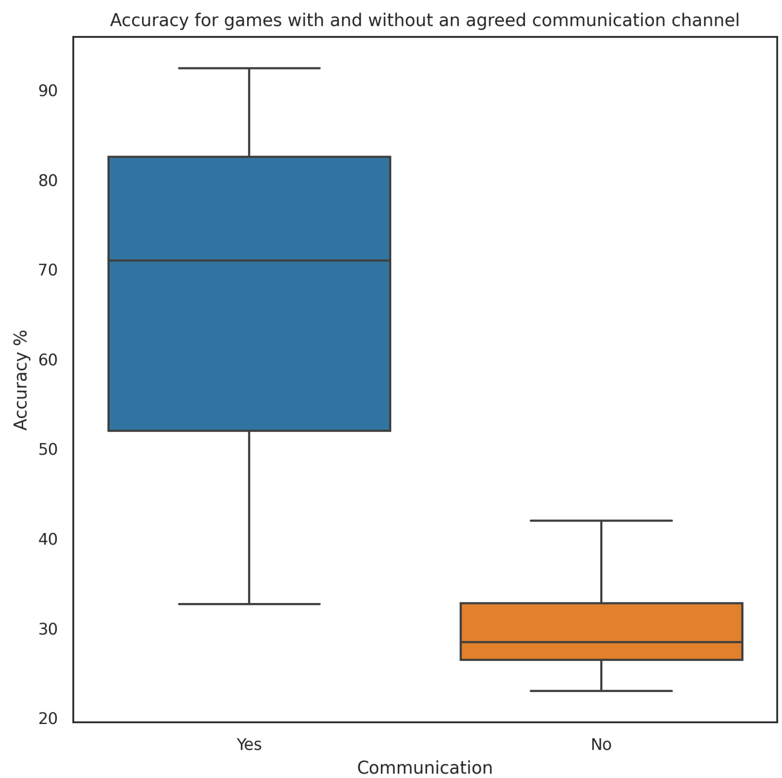


Figure 10: A boxplot visualizing the accuracy for players reporting successful and unsuccessful communication .

5.2 Experiment 2

With our second experiment, we investigate why the distribution of the model's accuracy is so wide for the successful players. We try to understand why for some successful players the model is more than 90 percent accurate while for others it can be less than 30 percent accurate. We investigate three explanations for this wide range of accuracy.

5.2.1 Experiment 2.1

An explanation for the models' low accuracy on some successful players can be that the model is not generalizable. The architecture of the model and its hyperparameters might be unsuited for application on some players. This could cause the model to overfit on the train data, causing low accuracy on the test data. If the validation loss is much higher than the training loss, the model is overfitting. We visualize for each player if such overfitting occurs by plotting the validation and train loss during training averaged over all folds. We also visualize this by plotting the validation and train accuracy during training averaged over all players.

5.2.2 Result Experiment 2.1

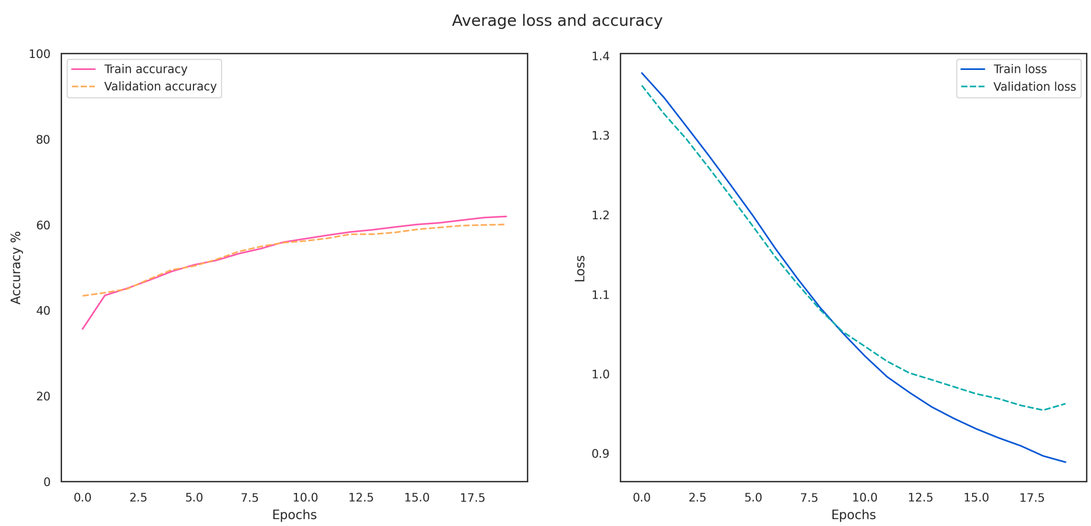


Figure 11: average accuracy (left) and validation(right) plot of our model during training on the data of 46 the participants.

The line plots in figure 11 show no significant overfitting on the averaged data. However, we also produce two line plots for every player, these results of our experiment are located in appendix A. From these plots, we can see that the model overfits on data from players that reported no communication was established or were not sure. For this data, the validation loss starts increasing early in training while the loss on the training set keeps decreasing. This indicates overfitting, which we expected. Since the data contains almost no information or patterns, the model starts learning the data, resulting in overfitting. In general, the model does not overfit on data from players that reported communication was successful. For this data, the validation loss decreases in a similar manner as the training loss. However, there are some exceptions. While to a lesser extent as compared to players that reported no communication was established or were not sure. For some successful players, the validation loss increases slightly or stagnates, while the loss on the training set keeps decreasing. The general pattern for such successful players is that the accuracy is low and its increase stagnates early in training, causing a horizontal line in the accuracy plot. Why this happens is not obvious, but since the model overfits on this data almost similairly to data from non-successful players, it might indicate that this data also contains almost no information or patterns. Causing the model to learn the data, resulting in overfitting. This data probably causes the wide range of accuracy for the successful players. We further investigate this in the next experiments.

5.2.3 Experiment 2.2

A second explanation for the wide range of accuracy for the successful players could be that one player in a pair is a leader while the other is a follower. With such a role allocation we expect the action sequences of the leader to contain much more information and thus signal a color better. While the other player is following and thus probably not creating many signaling movements itself. If the model has a low accuracy on a successful player we will compare it with the model's accuracy on the paired player. If the models' accuracy is much higher for the paired player, this might indicate such a leader-follower role allocation between the players. This could explain why the model's accuracy for some successful players is so low. We will investigate this explanation by comparing the model's accuracy scores for each pair. If there is a significant difference in accuracy between a pair we will investigate further by looking at the average number of movements of each player. Under the assumption that a player with a lead role will have a higher average number of movements, since this player has to signal, and the follower only has to follow. Finally, we will compare the participants' descriptions of how communication emerged to see if any role allocation is described.

5.2.4 Result Experiment 2.2

While comparing the players in each successful pair we can only find one pair with a significant difference in model accuracy. The model had an accuracy for player 1 of 41% while having an accuracy of 95% for player 2. By looking at the average number of movements of each player we can see that player 1 has on average 3,03 moves per round while player 2 has 13,19. Under the assumption that a player with a leading role has a much higher average number of movements compared to the follower, it could be that there is an allocation of leader and follower roles in this pair. Finally, we will compare the participants' descriptions of how communication emerged to see if any role allocation is described. Player 2 tries to signal the end color from the beginning of the game using movements. Later in the game (according to player 2 after about 30 minutes), player 1 realized that player 2 was trying to signal color with its movements. After communication had emerged between the pair, no clear role allocation is described. While both players report that communication has been established, this was late in the game. Player 2 was signaling a color with its movements from the start of the game while player 1 did only after communication was established. This late establishment of communication can explain the low accuracy for player

1 since only a small portion of its rounds may contain a signal for a color. Thus, the model's prediction will be wrong for most rounds where the movements don't signal a color.

Table 3: Description of how communication emerged, model's accuracy, and average number of moves for participant 1 and 2 of game 2.

Player	**Accuracy**	**Average number of moves**	**Discription of communication**
1	41%	3.03	At first not really any. I made a hierarchy in which I would choose the colours and that is how I realised that the other person used the code of moving the curser into all corners when talking about green. So we would always first test if the person had green. Then I realised that when they moved the courser side to side that meant blue, up and down on the left meant red and up and down on the right meant yellow. My partner was an absolute genius, coming up with this system and after my slow brain caught up the excersise became fun.
2	95%	13,19	From round three onwards tried to communicate green (clockwise round with dot), blue (bottom left-rigth-left-right-etc), red (left side up-down-up-down-etc), and yellow (right side up-down-up-down-etc). In the end there was communication (after like 30 min). In the end we also had winning streak due to communication.

5.2.5 Experiment 2.3

The phenomenon found in the previous experiment can also explain the models' wide range of accuracy. Players only report if communication was established but not when. If communication is established later in the game, there is a higher number of samples in the dataset that don't contain any signals. This causes more noise when training the model, but also reduces the accuracy when testing the model. Our final experiment will investigate this hypothesis by plotting the players' percentage of successful games against the accuracy of the model on that player. If a player has a higher percentage of successful games, we expect that a higher percentage of sequences of this player contain signal information. Thus, our model will be trained on data with less noise and the test data contains less noise which increases the accuracy when testing the model. Therefore, we expect our model to have a higher accuracy on games with a higher percentage of successful rounds.

5.2.6 Result Experiment 2.3

Each data point in the regression plot in figure 12 represents a player in the ECG game. We can see a correlation between the accuracy of the model on a player and the percentage of successful rounds played by that player. Furthermore, we can see an outlier in the top-left corner of the plot. This data point represents player 2 from the pair we discussed in experiment 2.2.

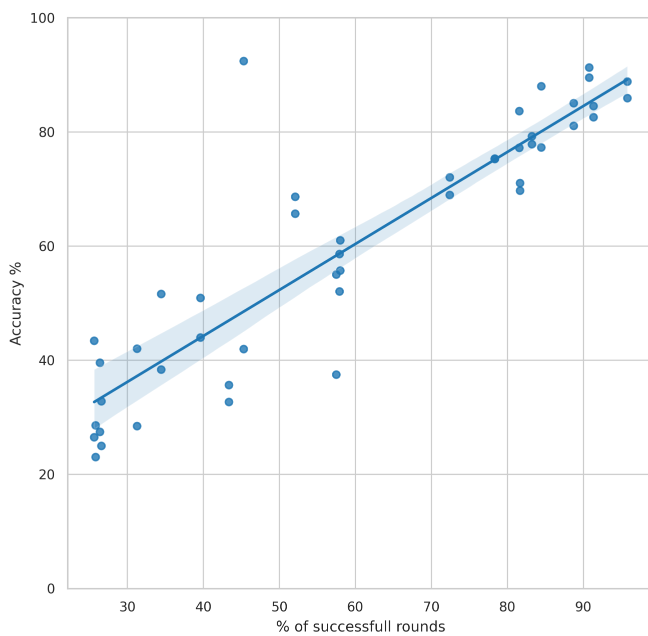


Figure 12: Regression plot of the accuracy of our model on a player and the percentage of successful rounds played by that player (r=0.90, p=1.95e-17).

5.3 Experiment 3

In the research of Scott-Phillips et al. (2009) [20] into the emergence of communication in the ECG, the general procedure of establishing communication was described. In general, a pair would start with a standard color to go to and then agree on signals for other colors when this standard color is not available. If the participants in Tom Kouwenhoven et al. (2022) [13] research also follow such a path in the communication emergence, we expect the frequency of the player ending up on the standard color to be higher than all other colors. Because this would be the players' "first choice", only if the color is not available another color would be chosen. In this experiment, we investigate if our model learns the same distribution of colors for its predictions.

We will make a barplot for each player containing the color distribution of all target labels. We compare this distribution with the distribution of the models' color predictions on the test set over all folds. For an accurate model, we expect the color distribution of the predictions to be similar to the distribution of colors in the target labels. Because, if the model is accurate, the predictions of the target values are accurate, and thus the distributions should be similar. We make a regression plot to visualize if there is a correlation. We measure the similarity of the distributions for each player by calculating the frequency for each color in the target labels and the test predictions. We then calculate the absolute difference for each color in terms of percentage in the test predictions compared to the target labels. For example, if red has a frequency of 100 in the target labels and 150 in the test predictions, there is a 50 percent difference for red. We calculated the distribution difference by summing the differences for each color, the higher this difference, the lower the distribution similarity. Inaccurate models are not that good at predicting the target value with the action sequences. These models have not "learned" the mapping between a signal and its meaning. Therefore, we expect these models to be biased towards predicting the color with the highest frequency in the train set. Because this maximizes the accuracy since it has the highest probability of occurring. Therefore, the color prediction distribution on the test set should be biased towards this color. We make a second regression plot to visualize if there is a correlation between the accuracy and the bias. The bias is calculated for each player by taking the color with the highest frequency in the target labels. We then compare the percentual difference in the frequency of this color in the test distribution compared to the frequency of this color in the target distribution.

5.3.1 Result Experiment 3

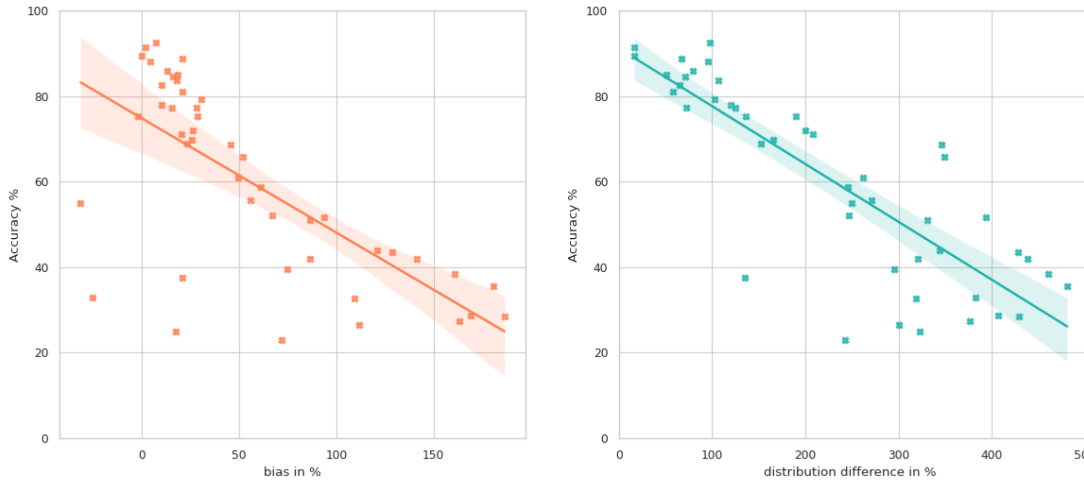


Figure 13: Regression plot of the accuracy of our model on a player and the model's bias on the left(r=-0.69, p=9.48e-08) and distribution difference on the right (r=-0.83, p=6.99e-13).

Each data point in the regression plots in figure 13 represents a player in the ECG game. We can see a correlation between the accuracy of the model on a player and the similarity of the color distributions. In general, a more accurate model produces a color distribution in its predictions similar to the color distribution in the target labels. Furthermore, we can see that less accurate models have a significant bias toward the color with the highest frequency in the target labels. It is possible for the bias to be negative since the color with the highest frequency in the target distribution can have a lower frequency in the test distribution. Since we produce a bar plot for every game, these results of our experiment are located in appendix B.

6 Discussion

In this paper, we describe research into the emergence of communication between humans, computers, and both. We further investigate the ability of a neural network with LSTM layers to map the meaning of signals produced by participants in the Embodied Communication Game from Scott-Phillips et al. (2009) [20]. For this, we use data gathered by Tom Kouwenhoven et al. [13] that replicated the findings of Scott Phillips et al. From our experiments we can find that our network has a significantly higher mean accuracy on data from players that established communication compared to players that did not (t=6.00, p=3.35e-07). This shows that the model is better at predicting the end color with sequences of players that reported they were successful. The models' wide range of accuracies on players that report communication was established suggests that there is a gradient in the quality of this data. This gradient can come from players reporting communication was established while there is no complete mapping from signal to color. For example, some successful players reported only an agreement on a standard color to move to and a secondary color if the standard color was unavailable while having no mapping for the other colors. Another explanation for the gradient is that while players reported that communication was established, they did not report when. If communication is established later in the game, there is a higher number of samples in the dataset from losing rounds. If a pair establishes communication when most of the game has been played the data from the entire game contains a lot of noise from rounds where no communication was present. We therefore measure how successful a pair was in establishing communication by calculating the percentage of successful rounds in their game. We can see a correlation between the accuracy of our model on a player and the percentage of successful rounds played by that player. However, since the sequences in the dataset are shuffled before splitting them into a train and validation set. While validating the model this results in the prediction of sequences that can be noise from early rounds. These sequences do not contain a signal to color mapping and thus even if the model can separate signals from noise is likely to make an incorrect prediction. We have to take into account the decrease in model accuracy when the noise increases. Thus, the strong correlation and the high accuracy for players with a high percentage of successful rounds show that the neural network performs well in mapping the meaning of signals in an emergent symbolic communication system in the ECG.

7 Conclusion

To conclude, our neural network can map the meaning of signals produced in an emergent symbolic communication system in the ECG. However, the accuracy of our model depends on the amount of noise in the data. If no communication has been established between two players, the data from each player in this game likely only contains noise instead of signals. And even if players report successful communication if a pair establishes communication when most of the game has been played the data from the entire game can still contain noise from rounds where no communication was present. We can see a strong correlation between the accuracy of our model and the percentage of successful rounds. The percentage of successful rounds also indicates the rate of samples in the dataset that don't contain any signals. This causes more noise when training the model, but also reduces the accuracy when testing the model. Since this noisy data can also be found in the test set due to shuffling, we have to take into account a decrease in model accuracy when the noise increases. Thus, the strong correlation and the high accuracy for players with a high percentage of successful rounds show that the neural network performs well in mapping the meaning of signals in an emergent symbolic communication system in the ECG. Furthermore, we can see a correlation between the model's accuracy and the similarity of the player's and model's color distribution. In general, a more accurate model produces a color distribution similar to the color distribution in the target labels. Less accurate models have a bias toward the color with the highest frequency in the target labels. Relating our findings in the literature and our experiment back to the question of whether a machine could learn to play the game of the ECG with a human. We can conclude from our experiments that a machine can learn the signal-meaning mappings of players in the ECG. And from Steels Talking Heads experiment that machines can recognize human signals and map their meaning if this system is constructed together with humans from the ground up. However, one problem with combing these two findings and concluding that a machine could learn to play the game of the ECG with a human is that the agents in the Talking Heads Experiment are pre-programmed with basic cognitive architecture and have a pre-defined communication channel.

8 Future Work

From our findings in literature and experiments, we can not conclude if a machine can play the Embodied Communication Game with a human. In future work, this possibility could be further investigated. Inspiration of how this can be achieved can be found by looking at the successful setup of the Talking Heads experiment and Khepera agents. The Talking Heads experiment shows the possibility for a novel communication system between humans and machines to emerge if it is built from the ground up. A first step should obviously be to actually enable our model to play together with a human player. The model should receive its own 2x2 with random colors and be able to perform sequences of movements with a human player. The model then needs to learn how to actually play the ECG. This is a difficult task since emerging a novel communication system depends on an interaction between the two players. Players need to recognize communicative signals and cooperate through dialogue to establish their goals. This could maybe be established by combining methods from the talking heads experiment and Khepera agents. By not only training the model by playing with a human, but the model should also train with other agents. The Khepera robots show that an agent can adopt a leader or follower role, depending on an incoming signal. This could be useful in an ECG agent as well since interaction and dialogue happen through turn-taking. The agent could understand that a certain incoming signal means to follow the other player. For example, if the other player moves from left to right, the model can predict to what color this maps and the agent can then check if this color is available. If it is the agent would move to this color. If it is not, this signals the agent would take on the leading role by performing a movement sequence of its own to convince the human player to move towards a certain color. It would be interesting to investigate the precise problems and pitfalls in designing an agent that plays the ECG with a human player.

References

[1] Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)* (2016), pp. 265–283.

[2] Chollet, F., et al. Keras, 2015, Available at: https://github.com/fchollet/keras.

[3] Chung, J., Gulcehre, C., Cho, K., and Bengio, Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555* (2014).

[4] De Ruiter, J. P., Noordzij, M., Newman-Norlund, S., Hagoort, P., and Toni, I. On the origin of intentions. *Attention & Performance XXII* (2007), 593–610.

[5] Deacon, T. *The symbolic species: The co-evolution of language and the human brain*. W.W. Norton Co., 1997.

[6] Fay, N., Garrod, S., MacLeod, T., Lee, J., and Oberlander, J. Design, adaptation and convention: The emergence of higher order graphical representations. In *Proceedings of the Annual Meeting of the Cognitive Science Society* (2004), vol. 26.

[7] Galantucci, B. An experimental study of the emergence of human communication systems. *Cognitive science 29*, 5 (2005), 737–767.

[8] Healey, P. G., Swoboda, N., Umata, I., and King, J. Graphical language games: Interactional constraints on representational form. *Cognitive science 31*, 2 (2007), 285–309.

[9] Hochreiter, S., and Schmidhuber, J. Long short-term memory. *Neural computation 9*, 8 (1997), 1735–1780.

[10] Hurford, J. R. Biological evolution of the saussurean sign as a component of the language acquisition device. *Lingua 77*, 2 (1989), 187–222.

[11] Hutchins, E. *Cognition in the Wild*. MIT press, 1995.

[12] Kelley, H. J. Gradient theory of optimal flight paths. *Ars Journal 30*, 10 (1960), 947–954.

[13] Kouwenhoven, T., de Kleijn, R., Raaijmakers, S., and Verhoef, T. Need for structure and the emergence of communication. In *Proceedings of the Annual Meeting of the Cognitive Science Society* (2022), vol. 44.

[14] Noble, J. Co-operation, competition and the evolution of. *The evolutionary emergence of language: social function and the origins of linguistic form* (2000), 40.

[15] Nowak, M. A., and Krakauer, D. C. The evolution of language. *Proceedings of the National Academy of Sciences 96*, 14 (1999), 8028–8033.

[16] Oliphant, M. *Learned systems of arbitrary reference: The foundation of human linguistic uniqueness. In E. Briscoe (Ed.), Linguistic evolution through language acquisition: Formal and computational models*. Cambridge University Press, 2002.

[17] Quinn, M. Evolving communication without dedicated communication channels. In *European Conference on Artificial Life* (2001), Springer, pp. 357–366.

[18] Rosenblatt, F. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review 65*, 6 (1958), 386.

[19] Rumelhart, D. E., Hinton, G. E., and Williams, R. J. Learning internal representations by error propagation. Tech. rep., California Univ San Diego La Jolla Inst for Cognitive Science, 1985.

[20] Scott-Phillips, T. C., Kirby, S., and Ritchie, G. R. Signalling signalhood and the emergence of communication. *Cognition 113*, 2 (2009), 226–233.

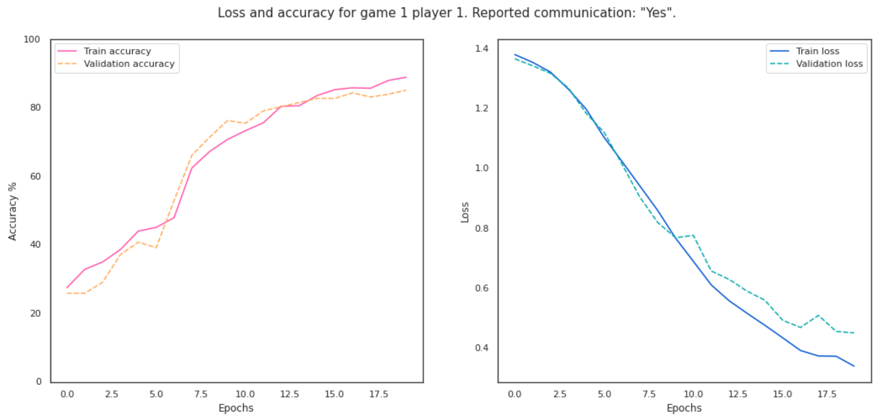
[21] Selten, R., and Warglien, M. The emergence of simple languages in an experimental coordination game. *Proceedings of the National Academy of Sciences 104*, 18 (2007), 7361–7366.

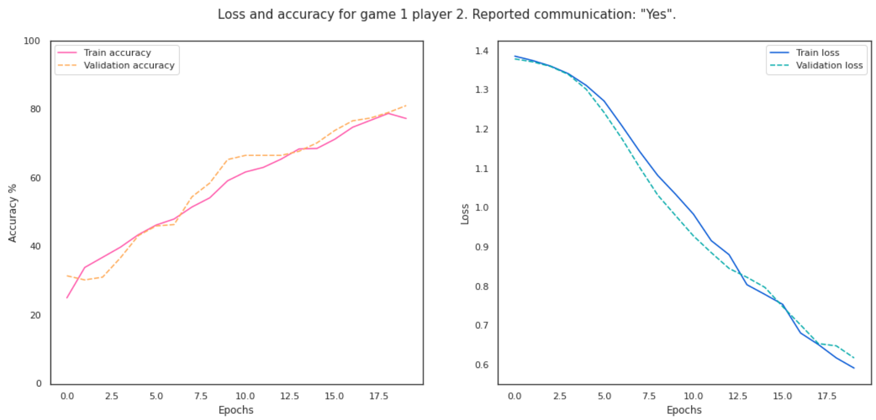
[22] Smith, K. The evolution of vocabulary. *Journal of theoretical biology 228*, 1 (2004), 127–142.

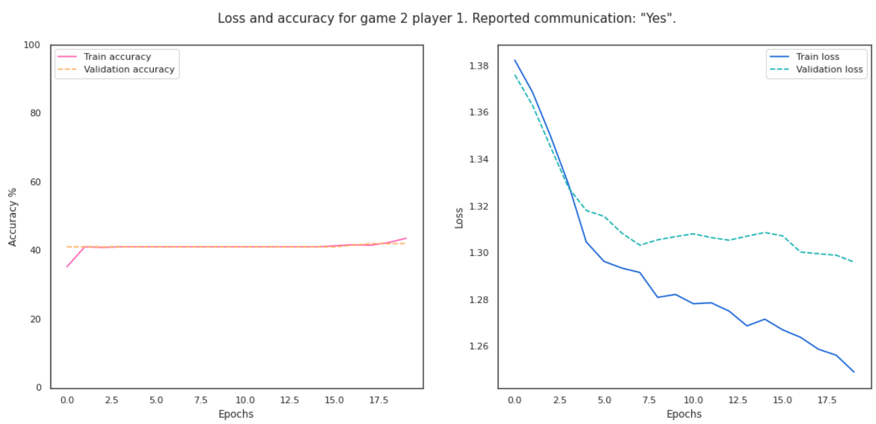
[23] Steels, L. *The Talking Heads Experiment.* Laboratorium, Antwerpen, 1999. Laboratorium, Antwerpen.

[24] Tomasello, M. *The cultural origins of human cognition.* Harvard university press, 2009.

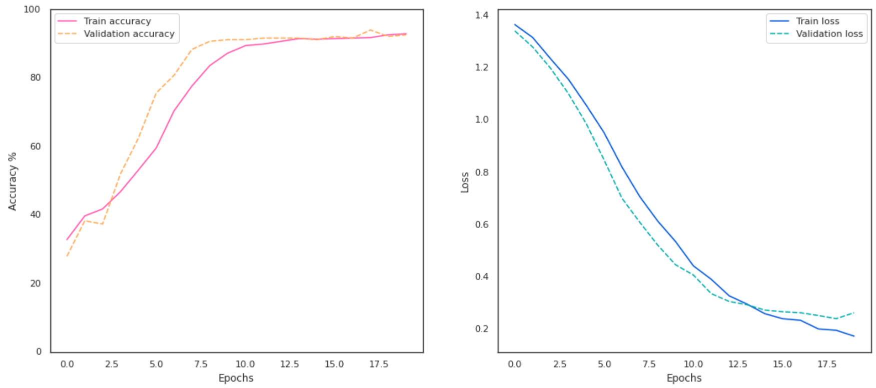
A Results experiment 2.1



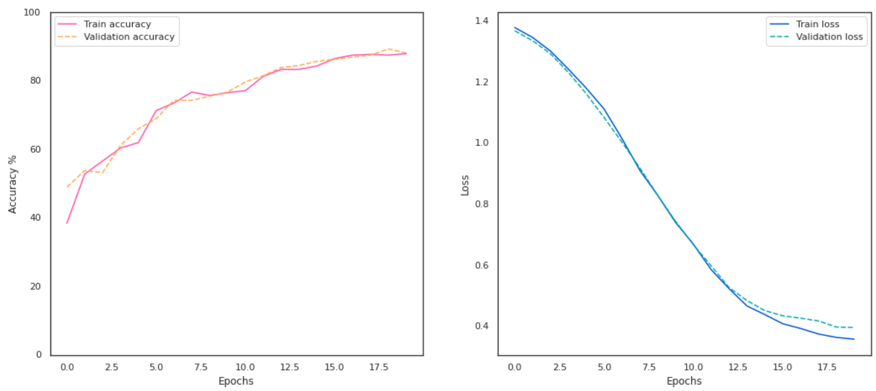




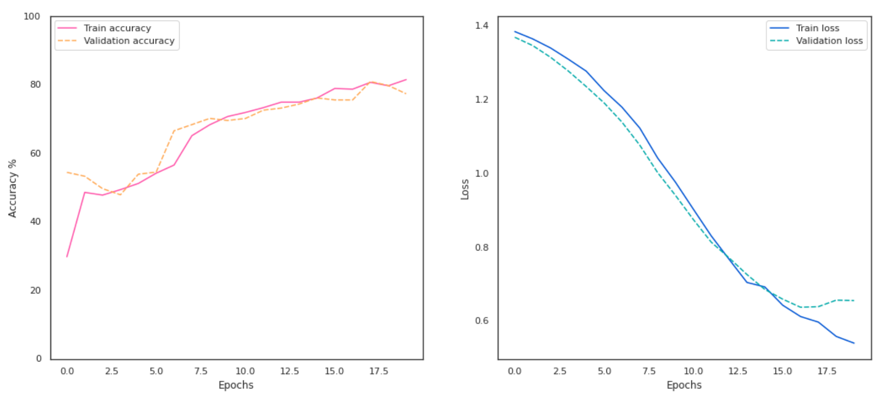
Loss and accuracy for game 2 player 2. Reported communication: "Yes".



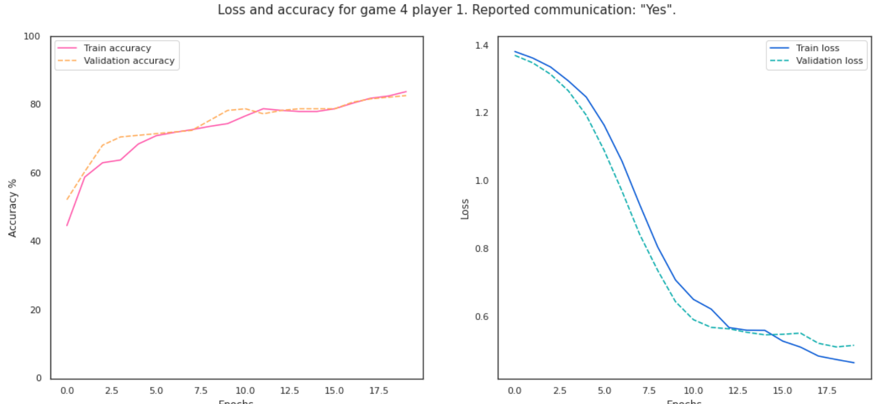
Loss and accuracy for game 3 player 1. Reported communication: "Yes".



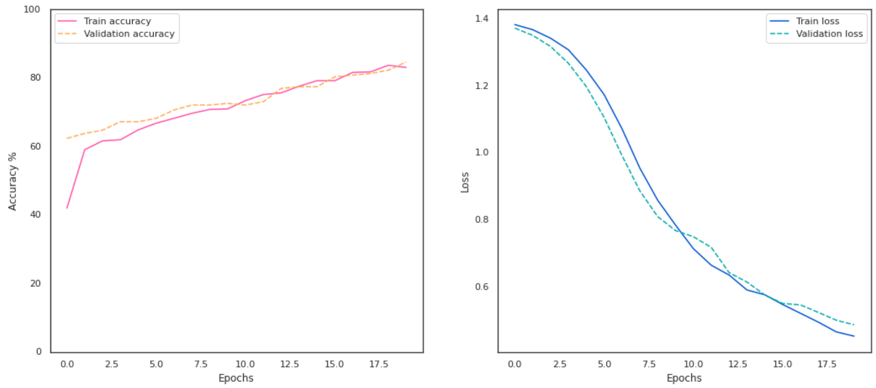
Loss and accuracy for game 3 player 2. Reported communication: "Yes".



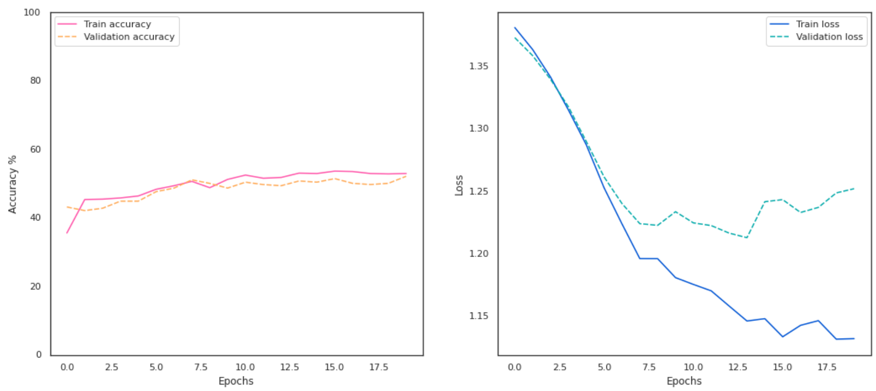
Loss and accuracy for game 4 player 1. Reported communication: "Yes".



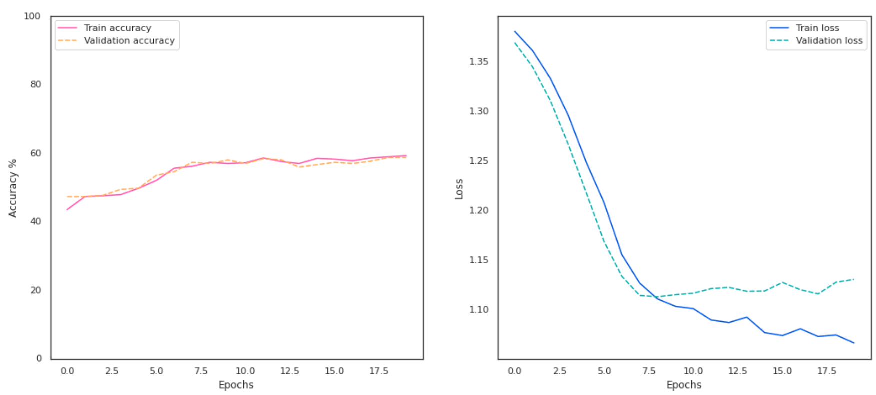
Loss and accuracy for game 4 player 2. Reported communication: "Yes".



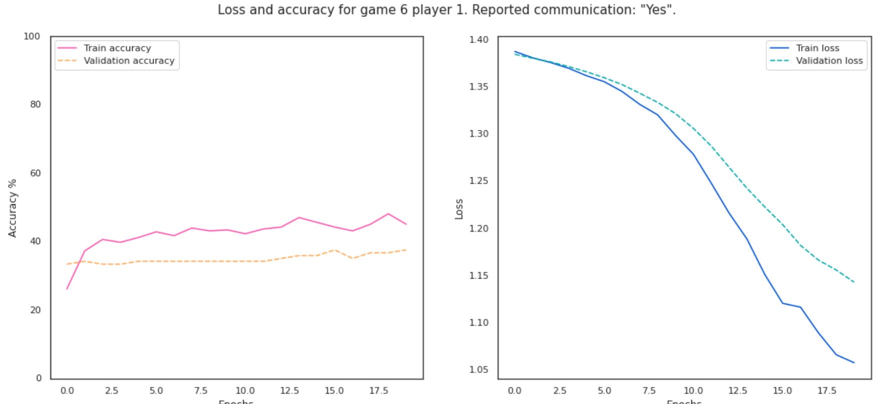
Loss and accuracy for game 5 player 1. Reported communication: "Yes".



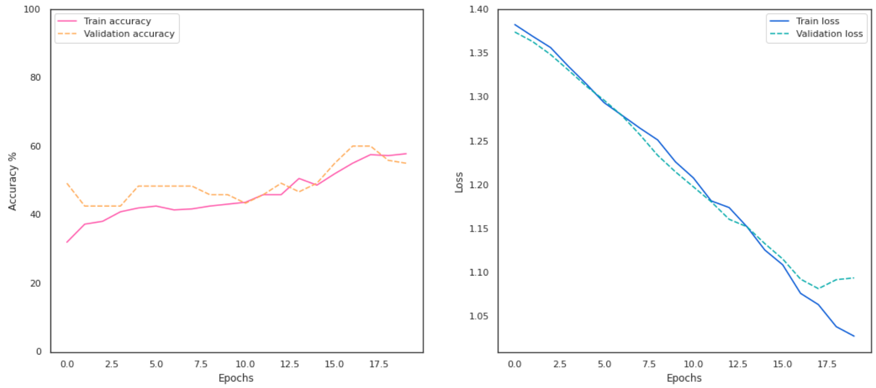
Loss and accuracy for game 5 player 2. Reported communication: "Yes".



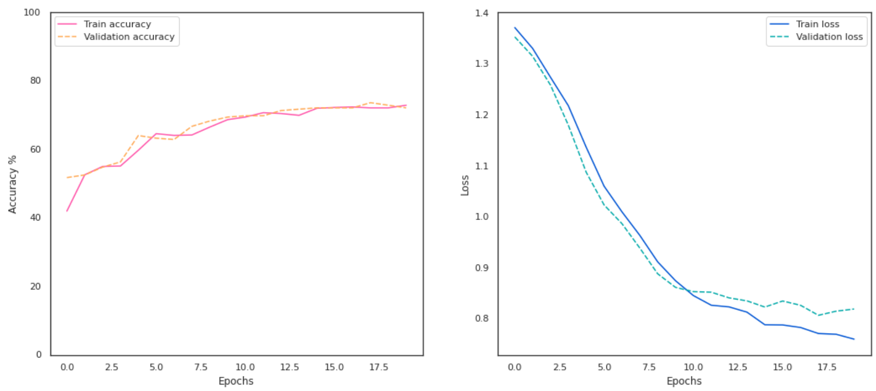
Loss and accuracy for game 6 player 1. Reported communication: "Yes".



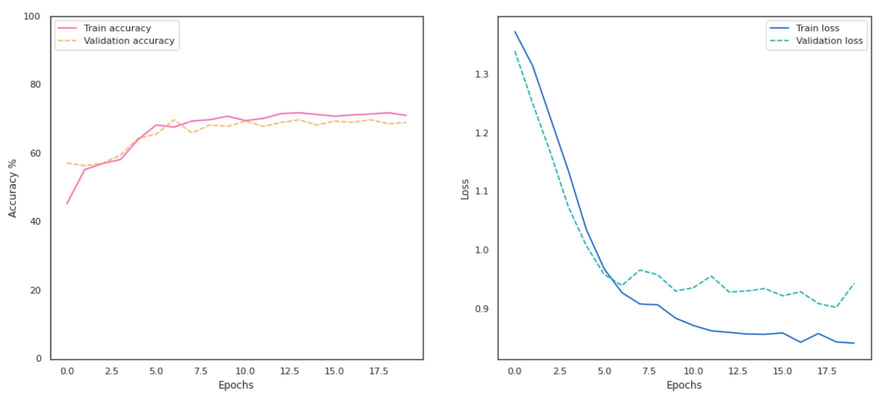
Loss and accuracy for game 6 player 2. Reported communication: "Yes".



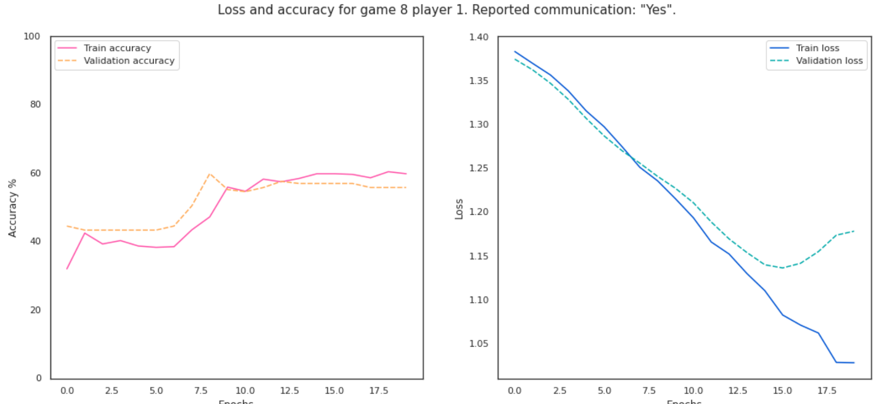
Loss and accuracy for game 7 player 1. Reported communication: "Yes".



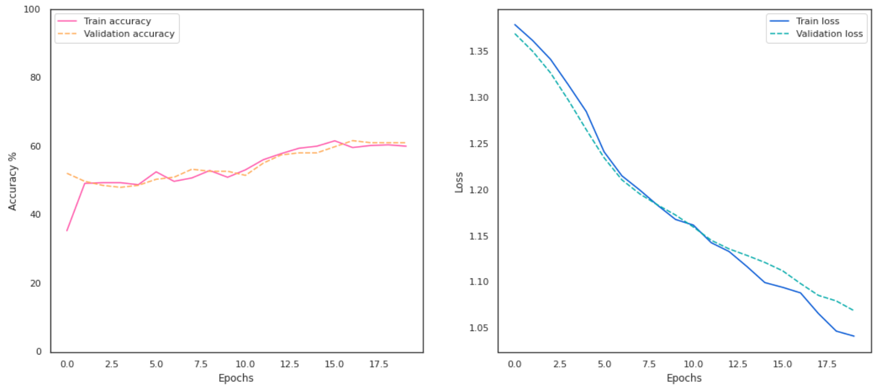
Loss and accuracy for game 7 player 2. Reported communication: "Yes".



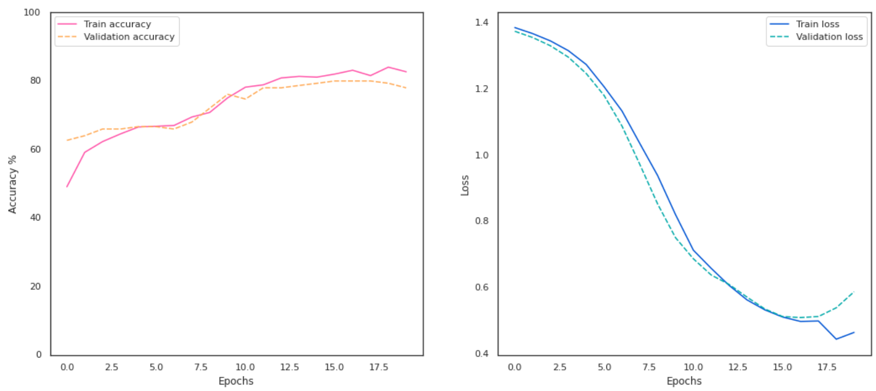
Loss and accuracy for game 8 player 1. Reported communication: "Yes".



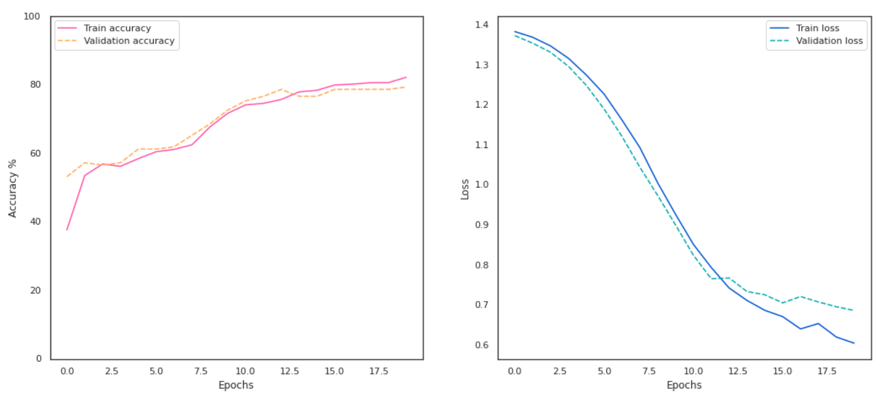
Loss and accuracy for game 8 player 2. Reported communication: "Yes".



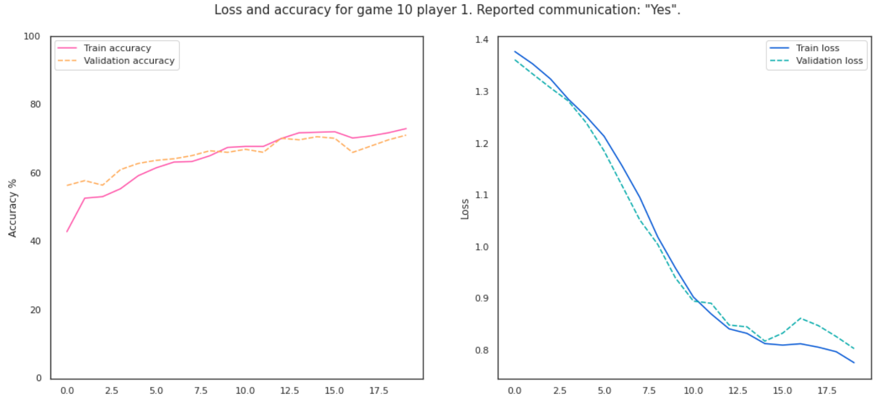
Loss and accuracy for game 9 player 1. Reported communication: "Yes".



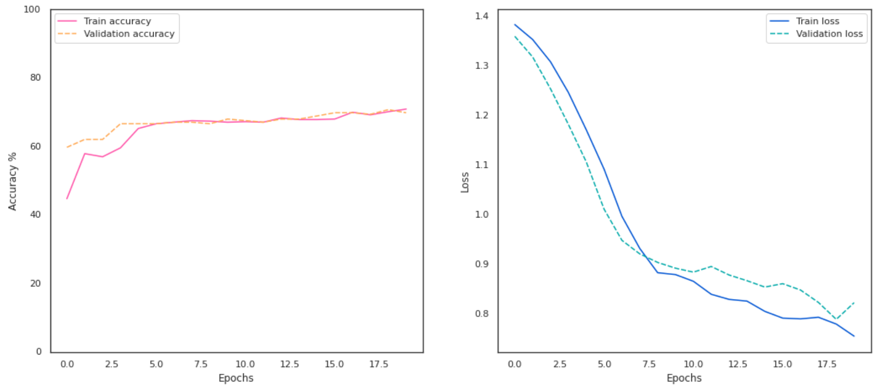
Loss and accuracy for game 9 player 2. Reported communication: "Yes".



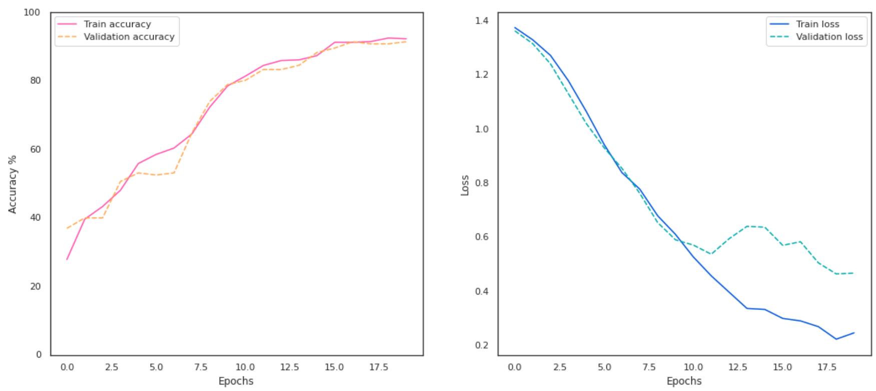
Loss and accuracy for game 10 player 1. Reported communication: "Yes".



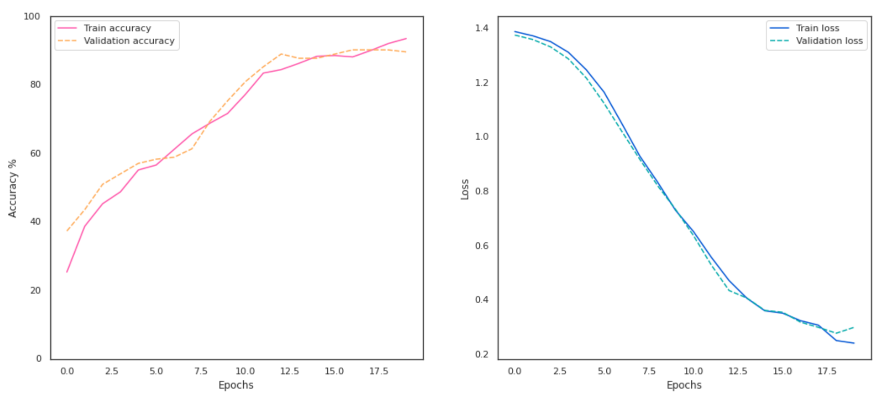
Loss and accuracy for game 10 player 2. Reported communication: "Yes".



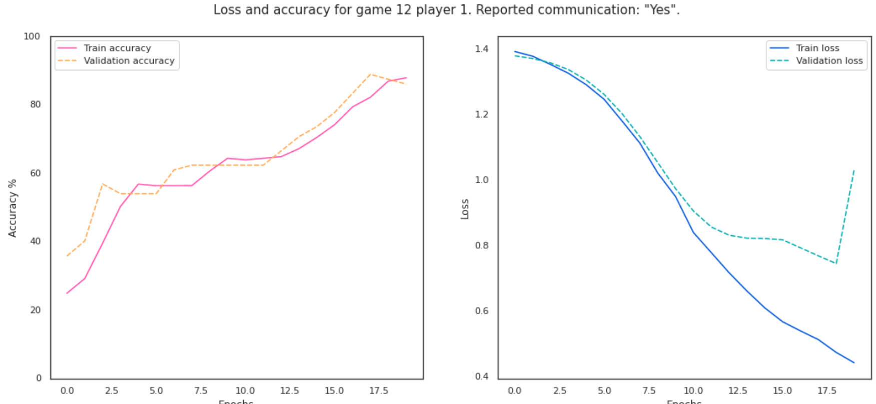
Loss and accuracy for game 11 player 1. Reported communication: "Yes".



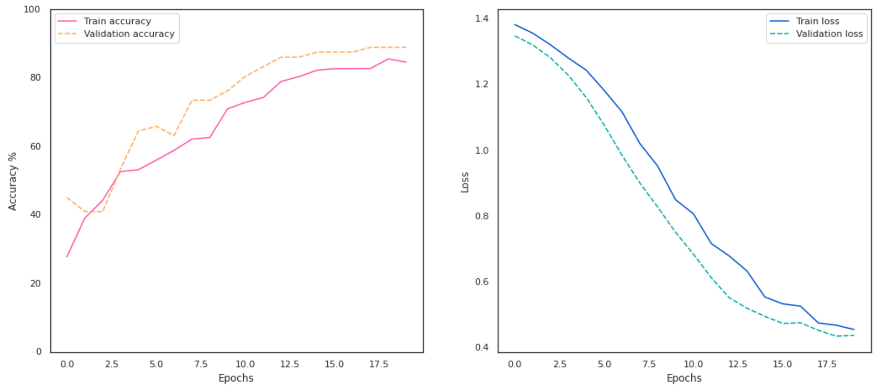
Loss and accuracy for game 11 player 2. Reported communication: "Yes".



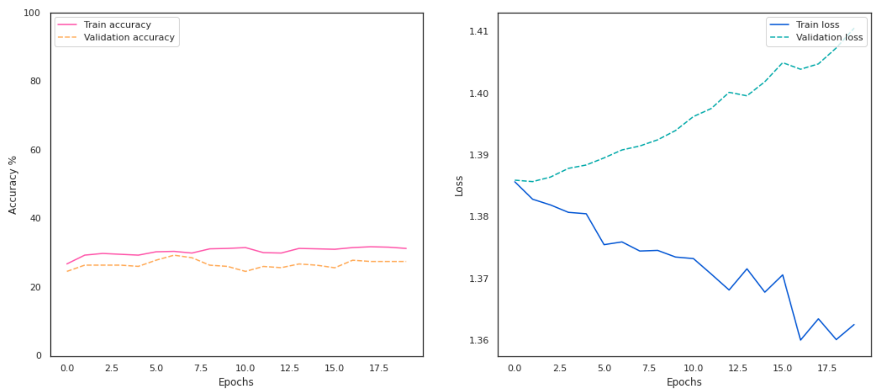
Loss and accuracy for game 12 player 1. Reported communication: "Yes".



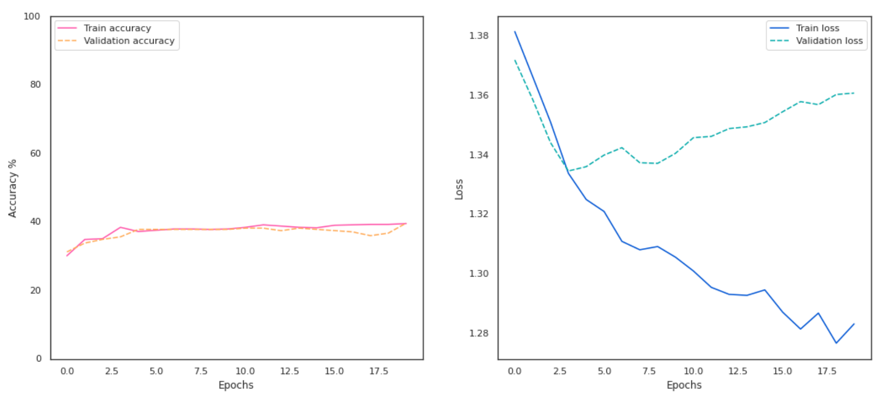
Loss and accuracy for game 12 player 2. Reported communication: "Yes".



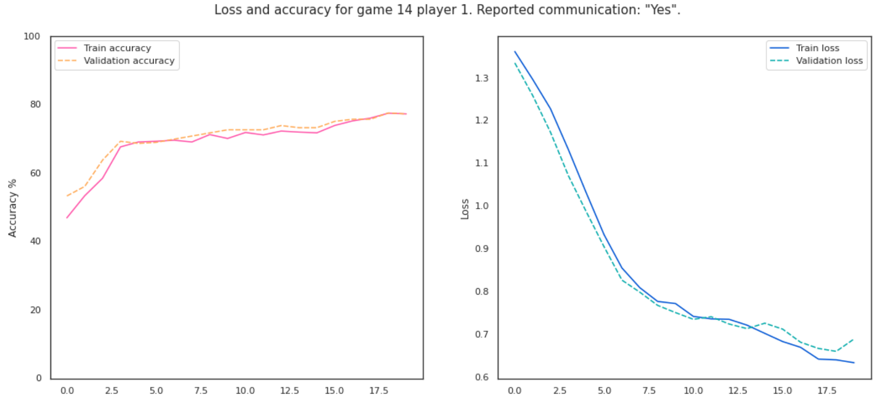
Loss and accuracy for game 13 player 1. Reported communication: "Not sure".



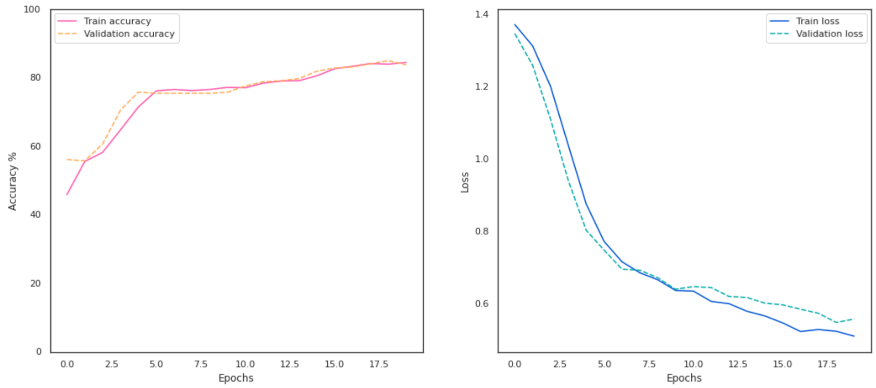
Loss and accuracy for game 13 player 2. Reported communication: "Not sure".



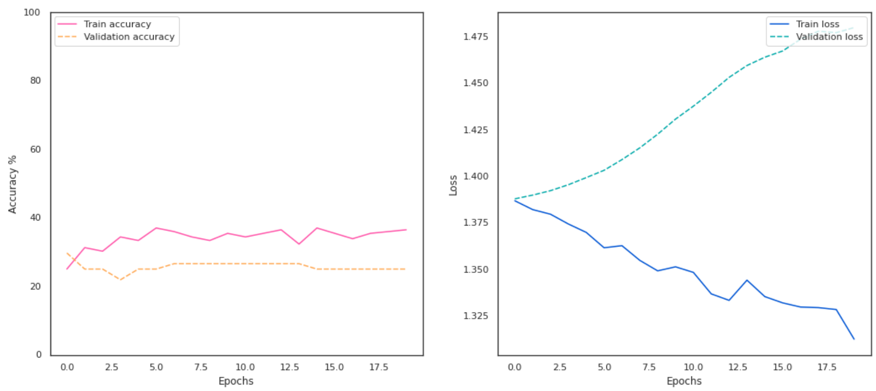
Loss and accuracy for game 14 player 1. Reported communication: "Yes".



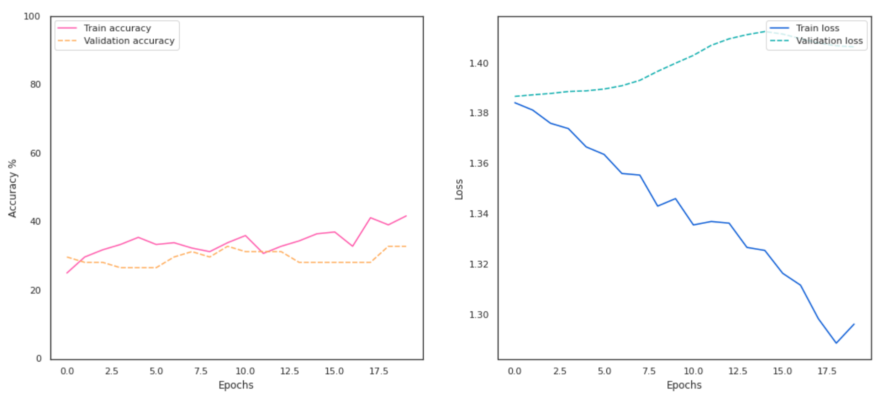
Loss and accuracy for game 14 player 2. Reported communication: "Yes".



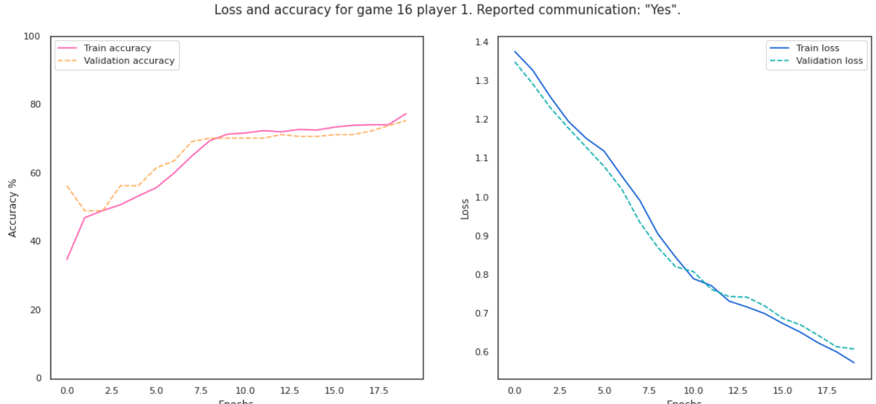
Loss and accuracy for game 15 player 1. Reported communication: "Not sure".



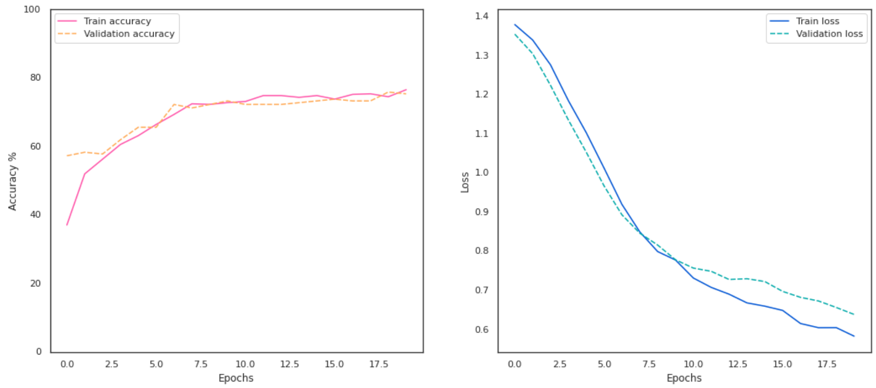
Loss and accuracy for game 15 player 2. Reported communication: "Not sure".



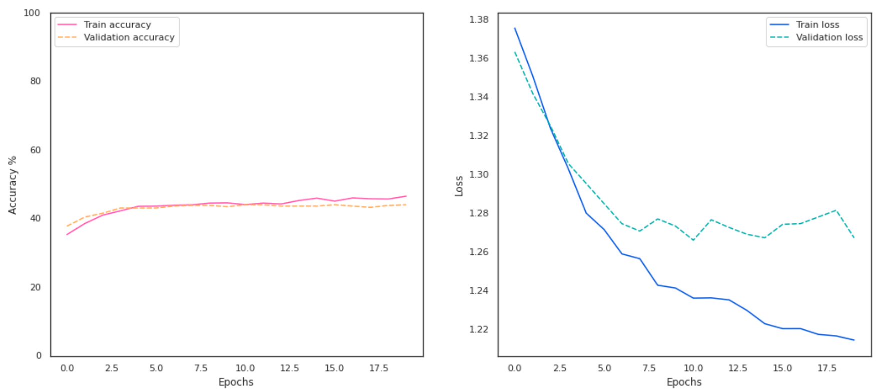
Loss and accuracy for game 16 player 1. Reported communication: "Yes".



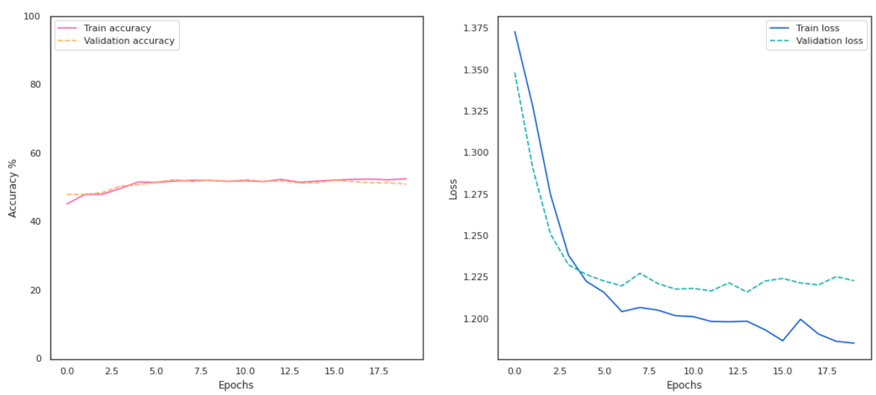
Loss and accuracy for game 16 player 2. Reported communication: "Yes".



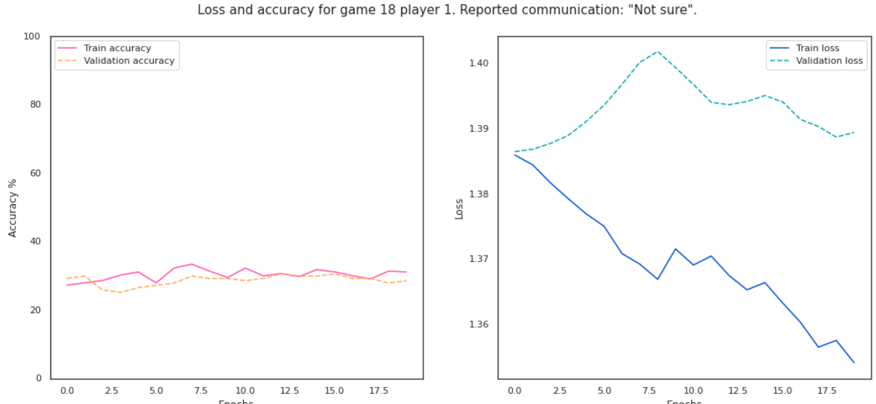
Loss and accuracy for game 17 player 1. Reported communication: "Yes".



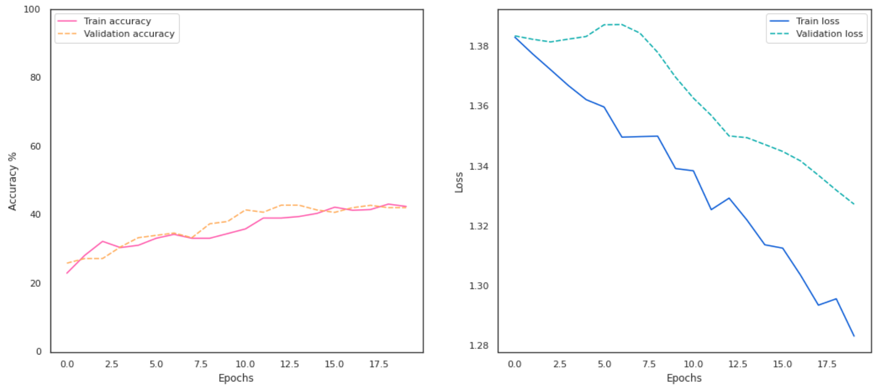
Loss and accuracy for game 17 player 2. Reported communication: "Yes".



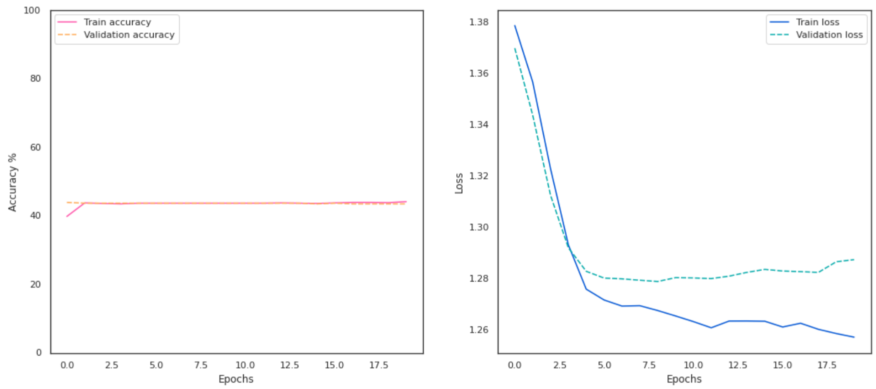
Loss and accuracy for game 18 player 1. Reported communication: "Not sure".



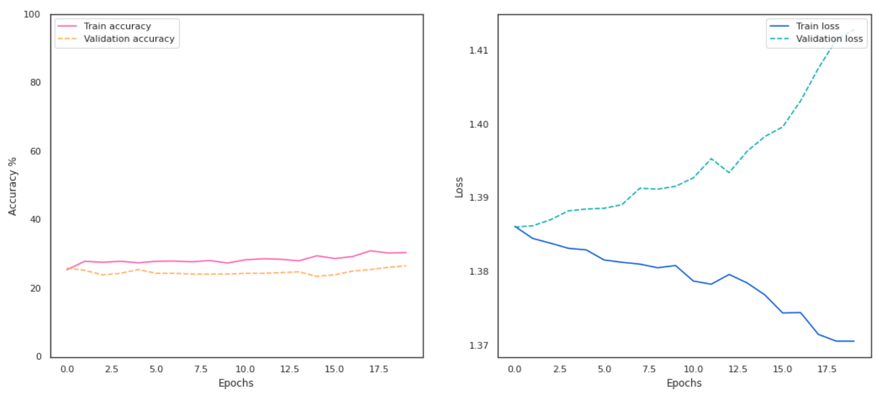
Loss and accuracy for game 18 player 2. Reported communication: "No".



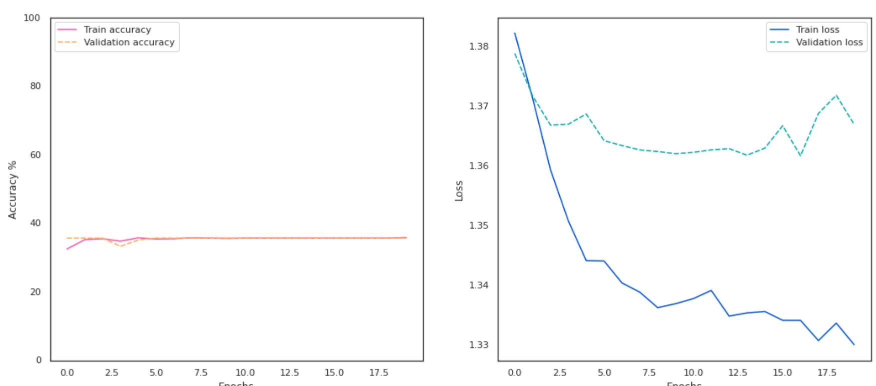
Loss and accuracy for game 19 player 1. Reported communication: "Yes".



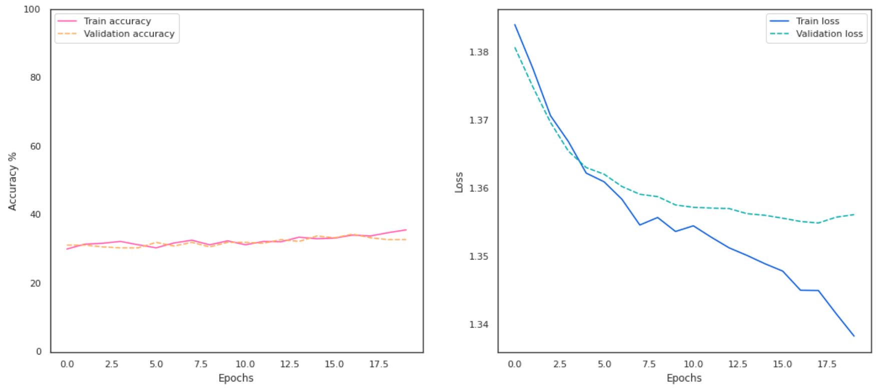
Loss and accuracy for game 19 player 2. Reported communication: "No".



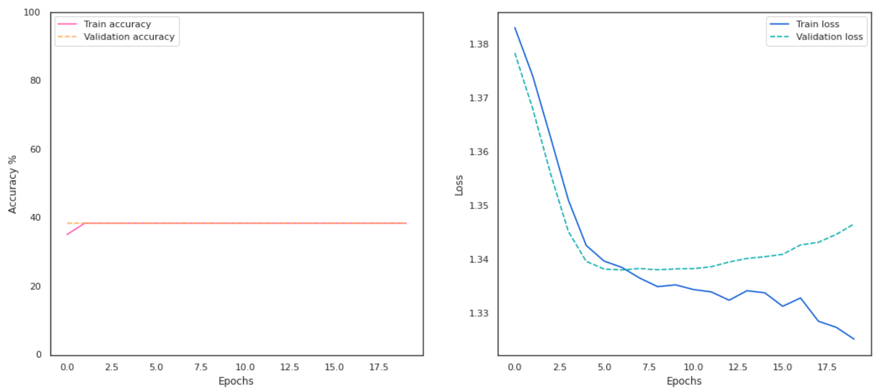
Loss and accuracy for game 20 player 1. Reported communication: "Yes".



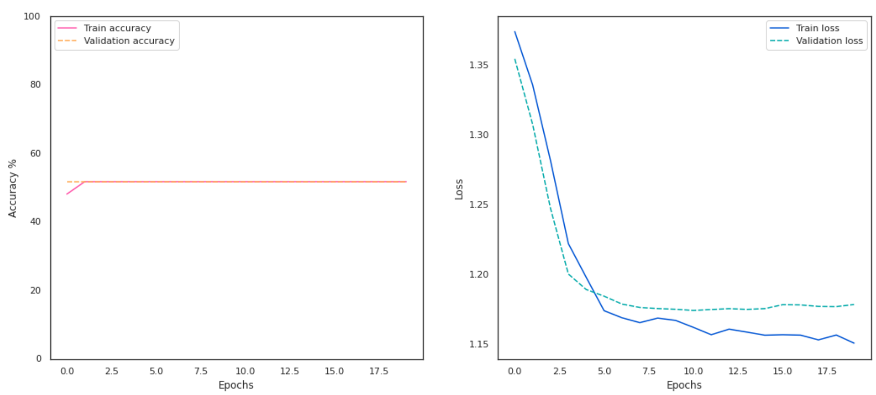
Loss and accuracy for game 20 player 2. Reported communication: "Yes".



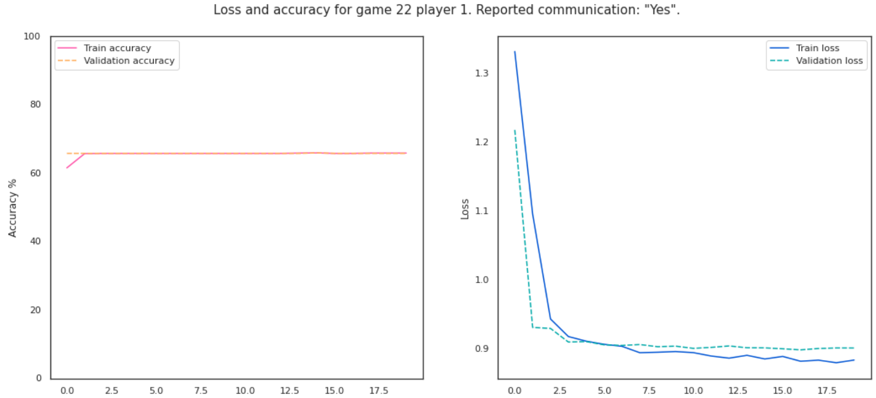
Loss and accuracy for game 21 player 1. Reported communication: "Yes".



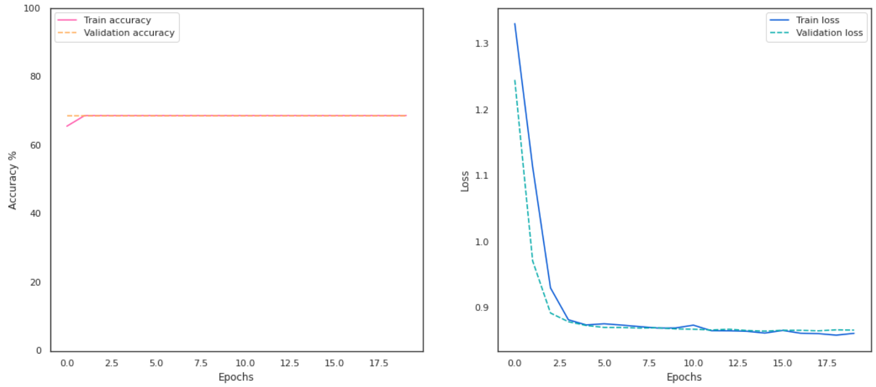
Loss and accuracy for game 21 player 2. Reported communication: "Yes".



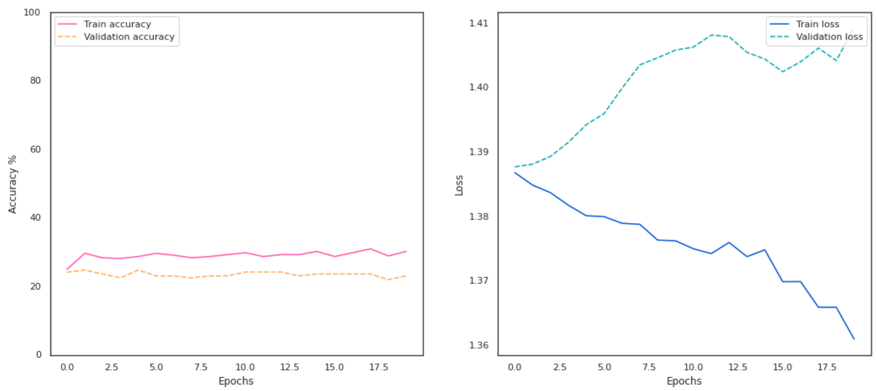
Loss and accuracy for game 22 player 1. Reported communication: "Yes".



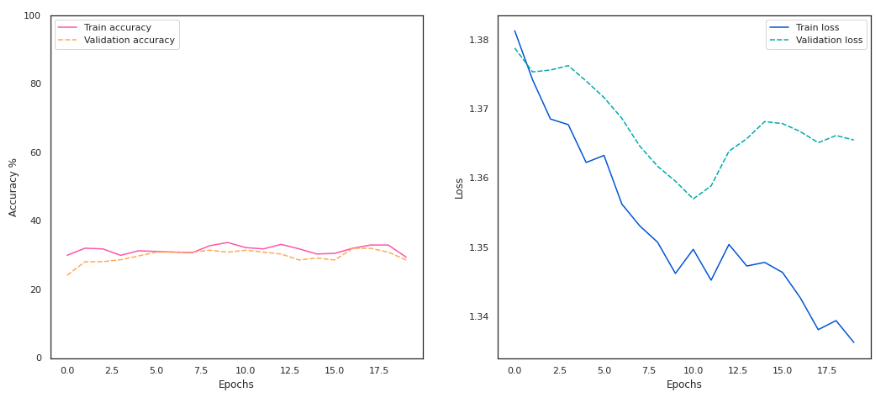
Loss and accuracy for game 22 player 2. Reported communication: "Yes".



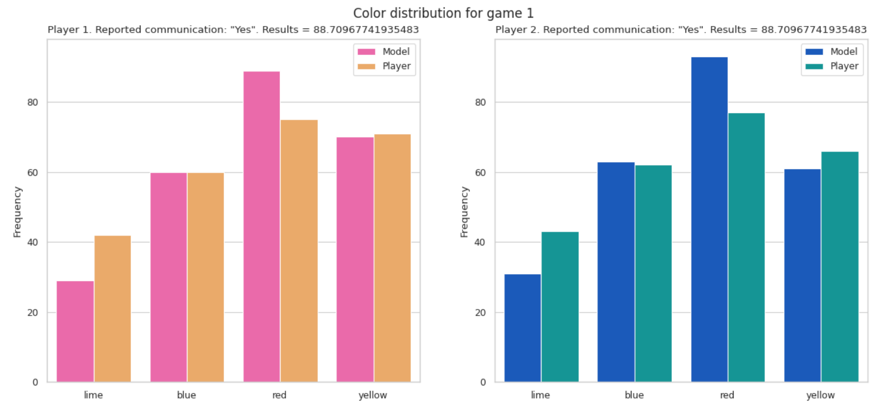
Loss and accuracy for game 23 player 1. Reported communication: "Not sure".

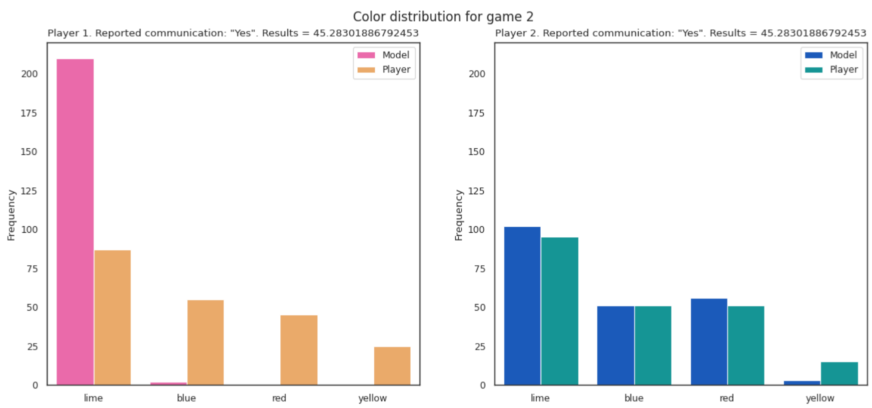


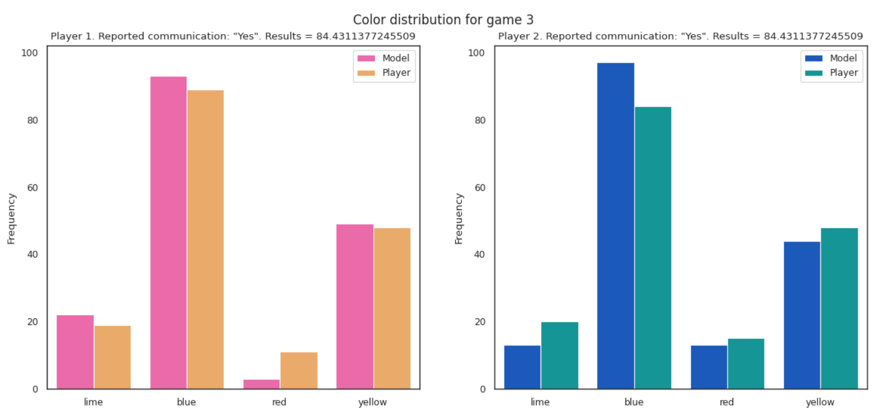
Loss and accuracy for game 23 player 2. Reported communication: "No".

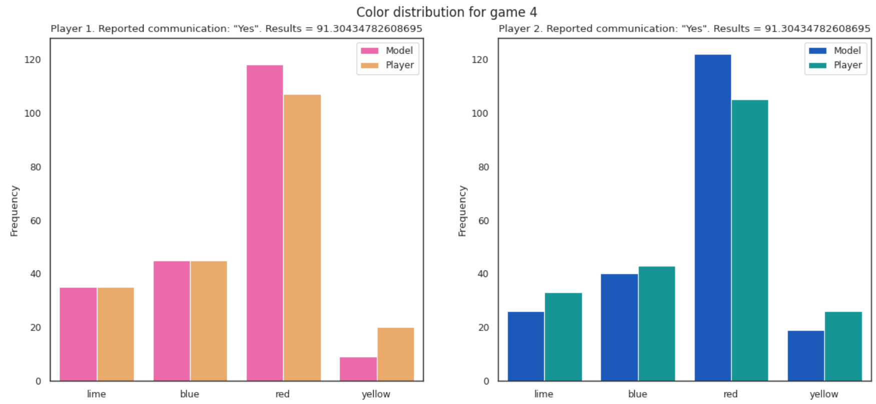


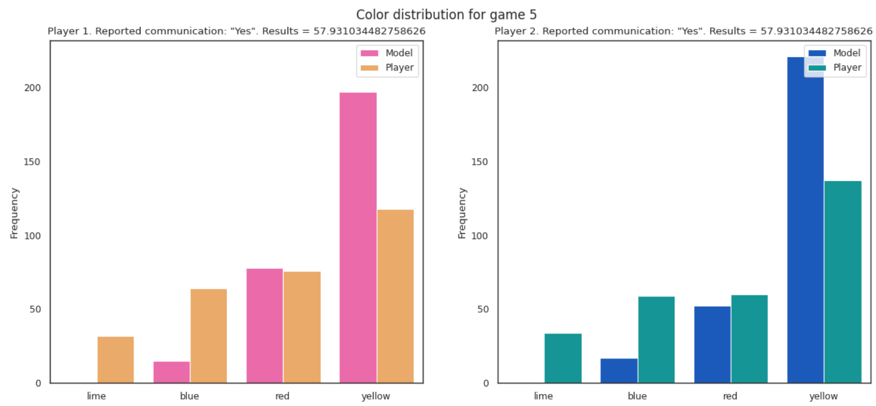
B Results experiment 3

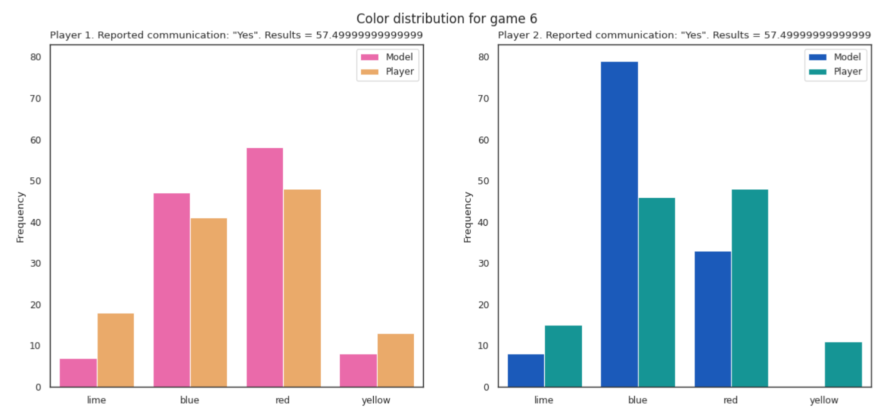


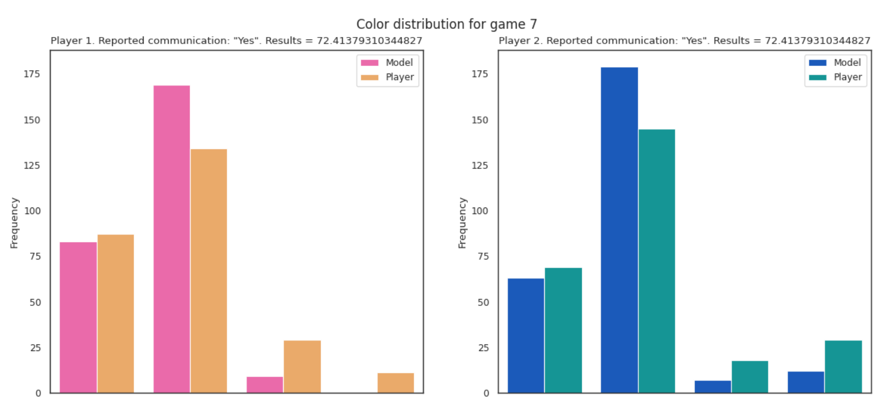


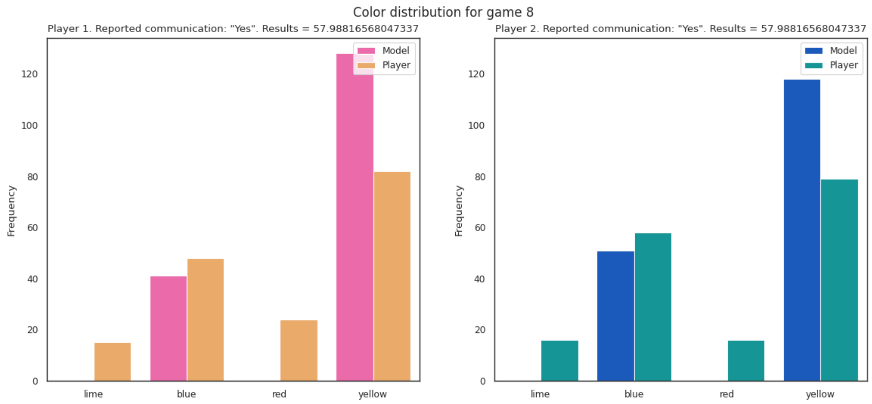


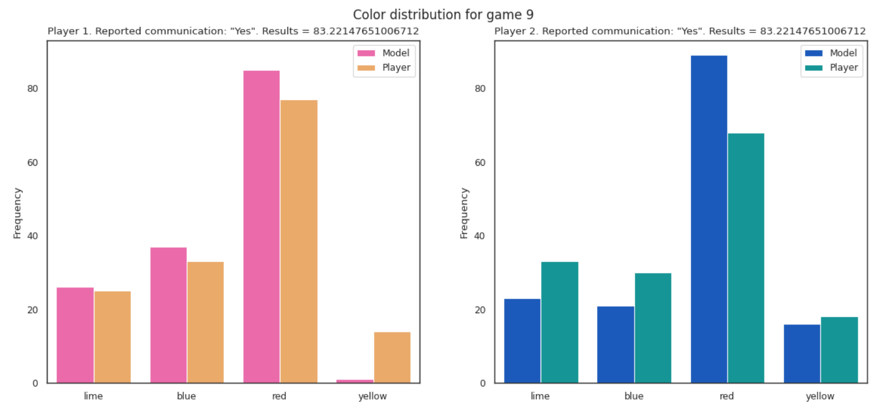

Color distribution for game 4
Player 1. Reported communication: "Yes". Results = 91.30434782608695
Player 2. Reported communication: "Yes". Results = 91.30434782608695
Frequency
Model
Player
lime
blue
red
yellow

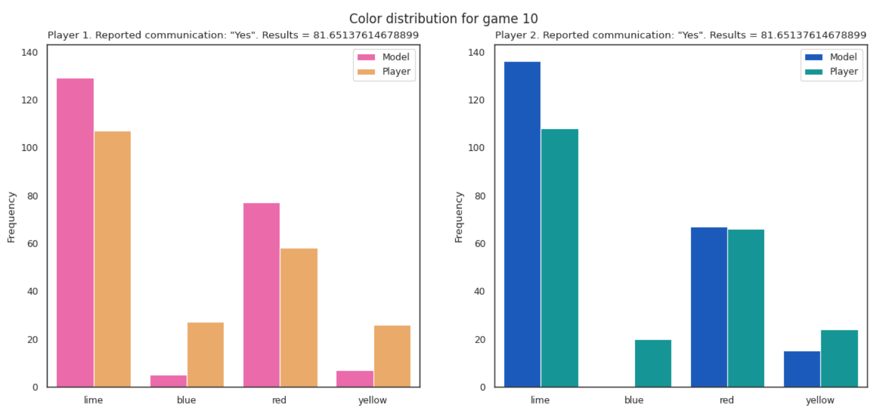

Color distribution for game 5
Player 1. Reported communication: "Yes". Results = 57.931034482758626
Player 2. Reported communication: "Yes". Results = 57.931034482758626
Frequency
Model
Player
lime
blue
red
yellow

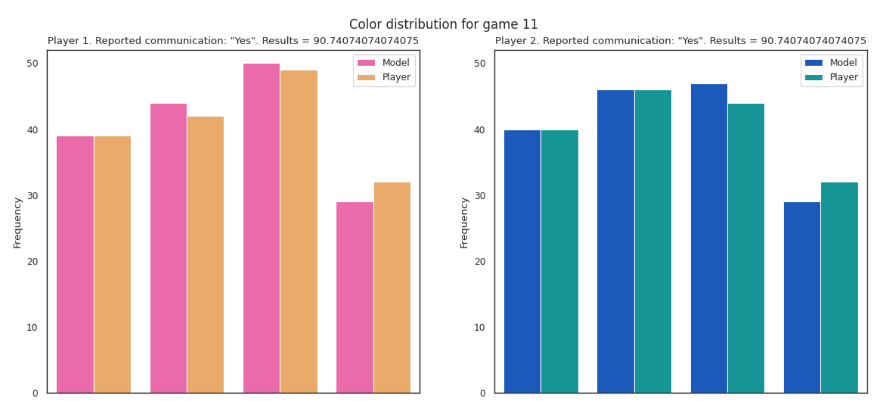

Color distribution for game 6
Player 1. Reported communication: "Yes". Results = 57.49999999999999
Player 2. Reported communication: "Yes". Results = 57.49999999999999
Frequency
Model
Player
lime
blue
red
yellow

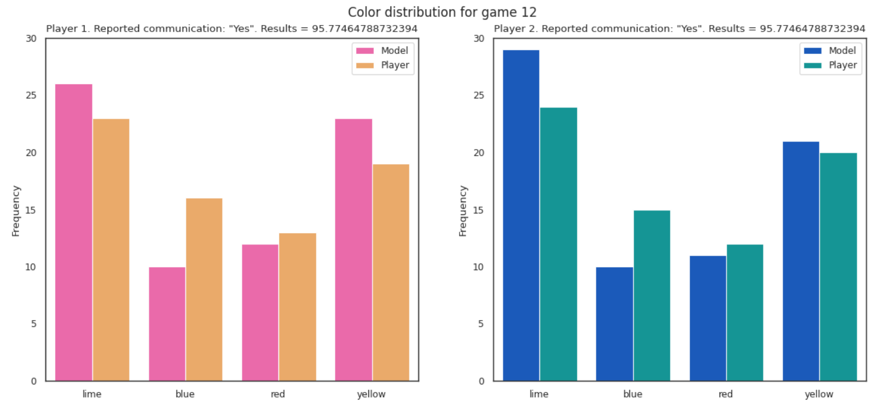

Color distribution for game 7
Player 1. Reported communication: "Yes". Results = 72.41379310344827
Player 2. Reported communication: "Yes". Results = 72.41379310344827
Frequency
Model
Player
lime
blue
red
yellow

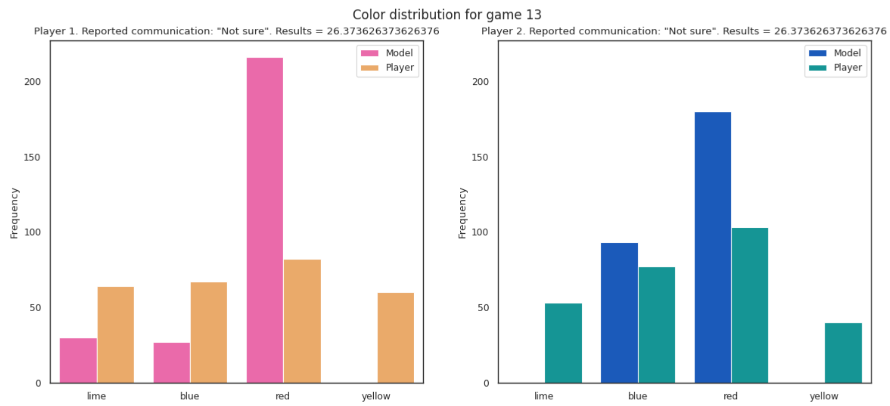

Color distribution for game 8
Player 1. Reported communication: "Yes". Results = 57.98816568047337
Player 2. Reported communication: "Yes". Results = 57.98816568047337
Model
Player
Frequency
lime
blue
red
yellow

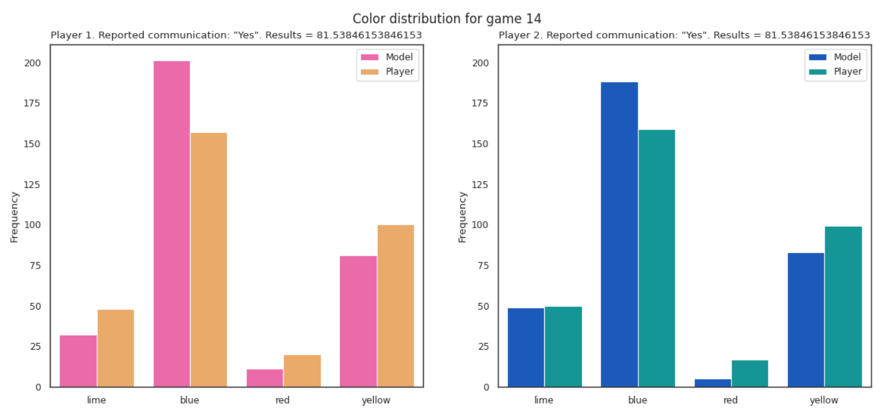

Color distribution for game 9
Player 1. Reported communication: "Yes". Results = 83.22147651006712
Player 2. Reported communication: "Yes". Results = 83.22147651006712
Model
Player
Frequency
lime
blue
red
yellow

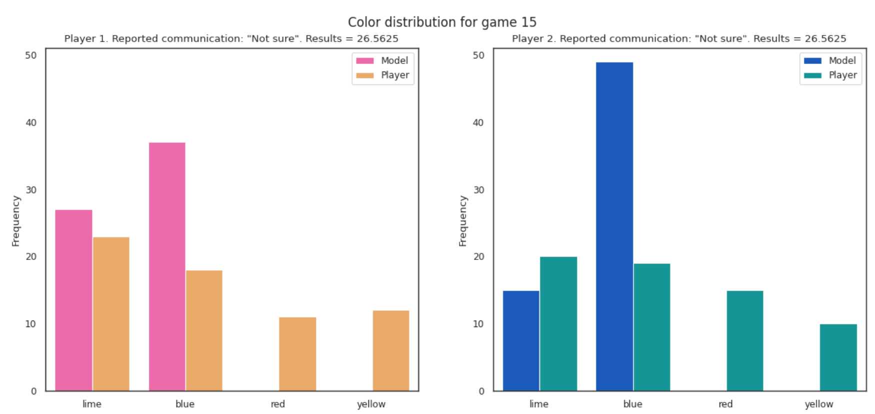

Color distribution for game 10
Player 1. Reported communication: "Yes". Results = 81.65137614678899
Player 2. Reported communication: "Yes". Results = 81.65137614678899
Model
Player
Frequency
lime
blue
red
yellow

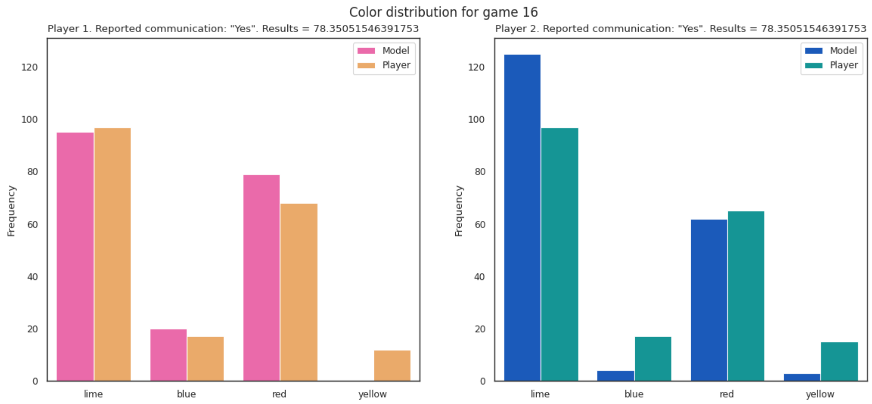

Color distribution for game 11
Player 1. Reported communication: "Yes". Results = 90.74074074074075
Player 2. Reported communication: "Yes". Results = 90.74074074074075
Model
Player
Frequency
lime
blue
red
yellow

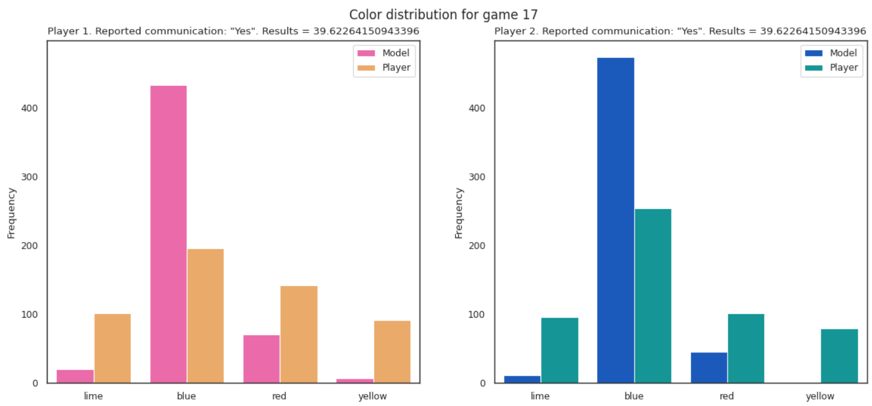

Color distribution for game 12
Player 1. Reported communication: "Yes". Results = 95.77464788732394
Player 2. Reported communication: "Yes". Results = 95.77464788732394
Model
Player
Frequency
lime
blue
red
yellow

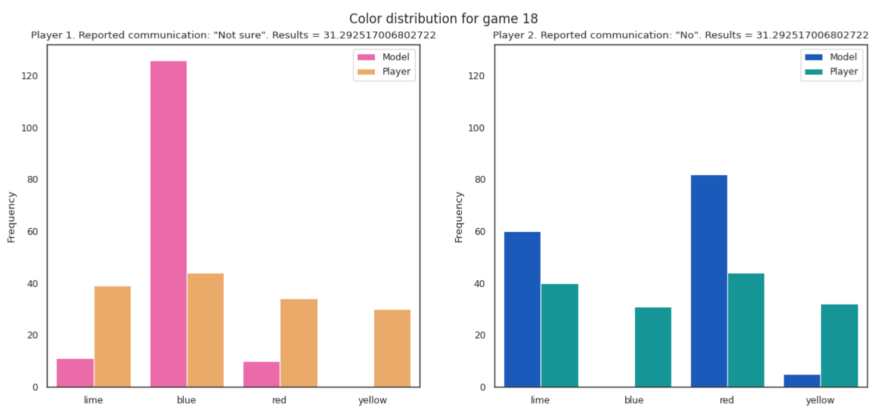

Color distribution for game 13
Player 1. Reported communication: "Not sure". Results = 26.373626373626376
Player 2. Reported communication: "Not sure". Results = 26.373626373626376
Model
Player
Frequency
lime
blue
red
yellow

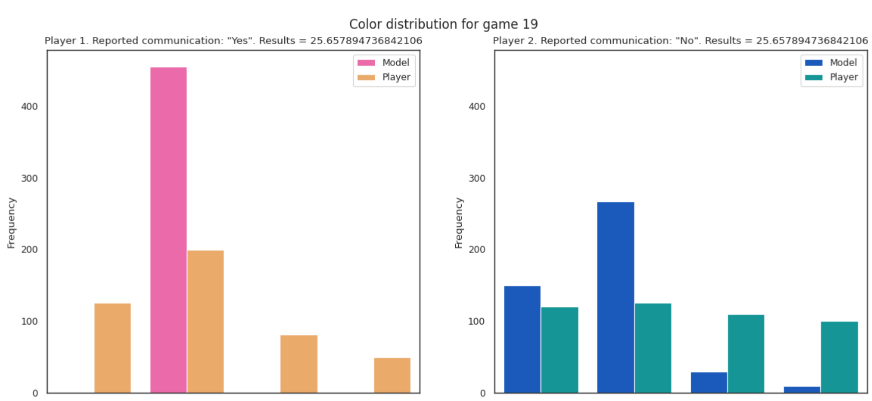

Color distribution for game 14
Player 1. Reported communication: "Yes". Results = 81.53846153846153
Player 2. Reported communication: "Yes". Results = 81.53846153846153
Model
Player
Frequency
lime
blue
red
yellow

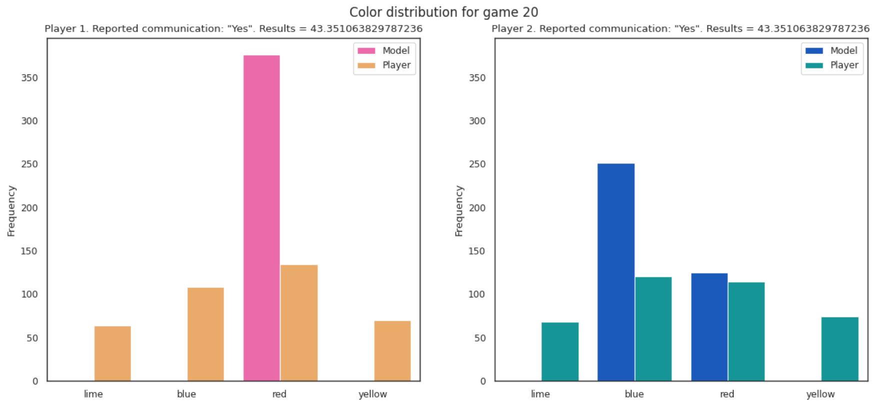

Color distribution for game 15
Player 1. Reported communication: "Not sure". Results = 26.5625
Player 2. Reported communication: "Not sure". Results = 26.5625
Model
Player
Frequency
lime
blue
red
yellow

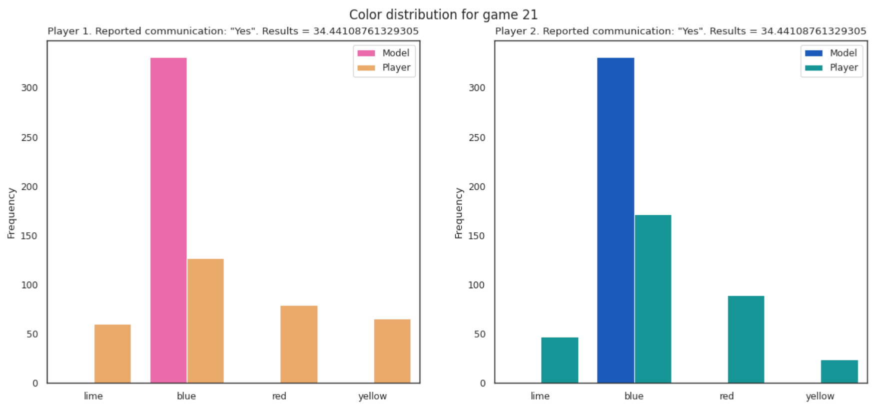

Color distribution for game 16
Player 1. Reported communication: "Yes". Results = 78.35051546391753
Player 2. Reported communication: "Yes". Results = 78.35051546391753
Model
Player
Frequency
lime
blue
red
yellow

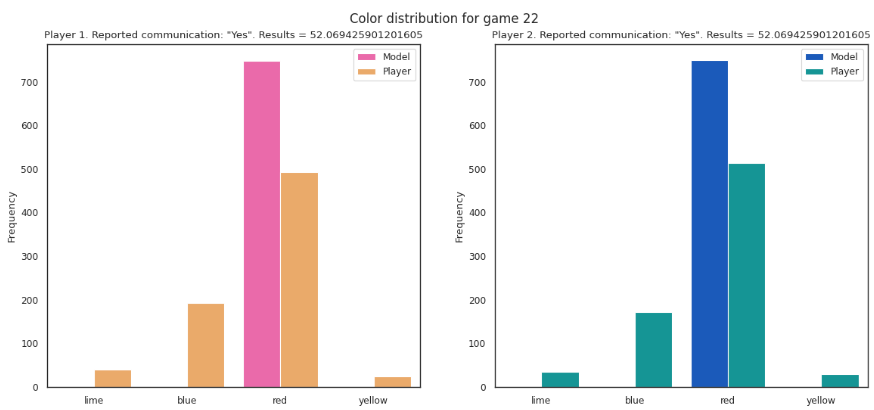

Color distribution for game 17
Player 1. Reported communication: "Yes". Results = 39.62264150943396
Player 2. Reported communication: "Yes". Results = 39.62264150943396
Model
Player
Frequency
lime
blue
red
yellow

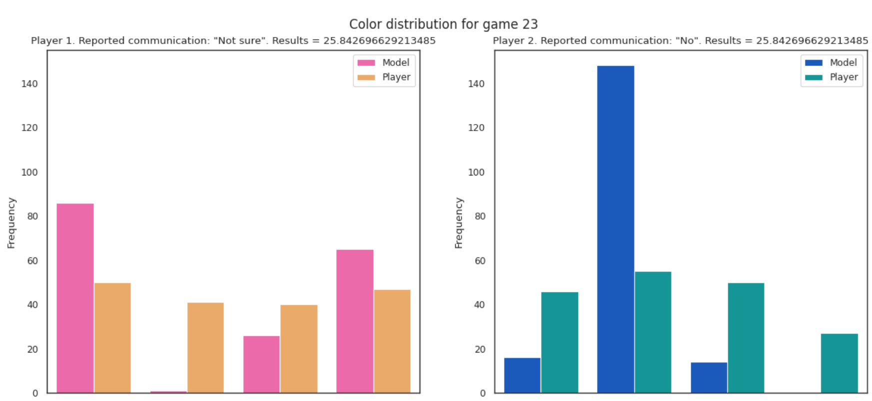

Color distribution for game 18
Player 1. Reported communication: "Not sure". Results = 31.292517006802722
Player 2. Reported communication: "No". Results = 31.292517006802722
Model
Player
Frequency
lime
blue
red
yellow


Color distribution for game 19
Player 1. Reported communication: "Yes". Results = 25.657894736842106
Player 2. Reported communication: "No". Results = 25.657894736842106
Model
Player
Frequency
lime
blue
red
yellow


Color distribution for game 20
Player 1. Reported communication: "Yes". Results = 43.351063829787236
Player 2. Reported communication: "Yes". Results = 43.351063829787236
Model
Player
Frequency
lime
blue
red
yellow


Color distribution for game 21
Player 1. Reported communication: "Yes". Results = 34.44108761329305
Player 2. Reported communication: "Yes". Results = 34.44108761329305
Model
Player
Frequency
lime
blue
red
yellow


Color distribution for game 22
Player 1. Reported communication: "Yes". Results = 52.069425901201605
Player 2. Reported communication: "Yes". Results = 52.069425901201605
Model
Player
Frequency
lime
blue
red
yellow


Color distribution for game 23
Player 1. Reported communication: "Not sure". Results = 25.842696629213485
Player 2. Reported communication: "No". Results = 25.842696629213485
Model
Player
Frequency
lime
blue
red
yellow