Learning visual representations through communication

Fenil Doshi and **Rushikesh Dudhat** and **Haowen Yu** and **Vengal Rao Guttha**
College of Information and Computer Sciences
University of Massachusetts Amherst

Abstract

Inspired by human language, cutting-edge research in emergent communication explores the ability of agents to communicate with one another using discrete symbols, which allow for greater interpretability. Emergent communication has also been shown to lead agents to develop high-quality representations of their input, acting as a self-supervised feature learning algorithm. This research demonstrates empirically that agents can be trained to use a single symbol as an effective communication medium, however one would expect multiple symbols to allow for greater expressivity as well as compositional generalization. In this project, we explore multiple symbol fixed-length communication in the context of a self-supervised vision framework. We develop a novel fixed-length communication mechanism, as well as evaluation metrics for the quality and interpretability of the emergent language between the agents. We also externally incorporate structure in the communication channel and show how it affects the game accuracy. After training on ImageNet data, we observe that communicating multiple symbols increases the game accuracy to 97.34 % from 82.2 % as was in the case of communicating a single symbol (baseline).

1 Introduction

In this project, we consider the phenomenon of *emergent language* between two agents. *Emergent language* is interesting because learning the language happens through interaction rather than static and passive "learning" like language modeling. This is a more realistic approach to developing a language ability and might grasp all the functional aspects of language that are otherwise not captured in the learning dynamics. There are different ways the communication can be achieved. However,we restrict ourselves to discrete communication, i.e. symbols (one-hot vectors from some "vocab" V with size $|V|$ for every position), which can facilitate interpretation of the emergent communication strategies. As noted by Dessì et al. (2021), the main reason for choosing discrete symbols is it's intuitive similarity with human languages where in humans use a set of fixed symbols (i.e. words) in their communication so they can be easily decoded. So our hope is using discrete symbols we can understand the language that the agents develop when solving a task at hand.

1.1 Task Description

Our task is to explore the phenomenon of emergent communication with multiple symbols. The long-term goal of this line of research is to explore autonomous agent communication, with an eventual end goal of getting to interpretable dialog between agents. To this end, we will implement a guessing game between the SENDER and the RECEIVER, using some unlabelled dataset $\mathbf{X}$.

A SENDER receives an item from $\mathbf{X}$, called the `target`, and produces an N-tuple of symbols from some "vocab" V with size $|V|$.

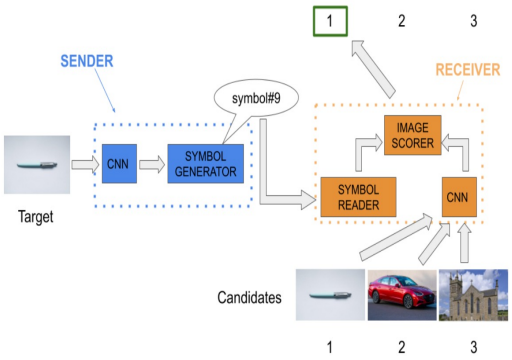

Figure 1: Referential Communication Game between two agents - **SENDER** and **RECEIVER** (Figure: Dessì et al. (2021), Images: `https://unsplash.com`)

The RECEIVER takes the N symbols as input, and a set of random objects from X which includes the `target` and must identify which of the items is the target. Learning effective representations and compactly communicating the message from SENDER to RECEIVER is necessary for the successful completion of the task.

We experiment on different datasets (CIFAR10, CIFAR100 (Krizhevsky, 2009), ImageNet (Russakovsky et al., 2015)) and use a similar architecture as Dessì et al. (2021). The input dataset $\mathbf{X}$ will contain images. The SENDER architecture will encode `target` into a feature vector using a convolutional neural network. A one-layer feed-forward network will map this into a $|V|$-dimensional vector and then extract multiple symbols from this feature vector (more details about the process in Section 1.3).

1.2 Motivation and Limitations of Existing Work

The primary motivation for this work is to make progress toward interpretable dialog between autonomous agents. This is inspired by human language, where we use multiple words to express our intent or describe what we see. Humans use compositional structure in language as a form of compression to express rich concepts with a compact vocabulary. For example, if a person understands the meaning of 'red', 'brown', 'black', 'dog,' and 'cat', they can make sense of 'red dog', 'black cat', 'brown dog', and so forth, without having to use separate symbols for each entity such as 'red_dog' and 'black_cat'. This is a form of compositional generalization, which existing neural networks struggle with (Li et al., 2019; Kim and Linzen, 2020). Allowing agents to leverage such a powerful paradigm should greatly increase the richness of the emergent communication.

As mentioned previously, our proposed task and architecture is based on Dessì et al. (2021), which utilizes a single symbol for agent communication. This work demonstrates agents can communicate effectively using a single symbol in a communication channel, and moreover the communication channel is interpretable. However, the limitation of using a single symbol and the richness which may be found through compositional generalization when using multiple symbols is an exciting avenue of research. As a first step in this direction, we would like to explore a fixed-length communication strategy wherein SENDER and RECEIVER exchange a fixed number of symbols. We want to understand if this method can develop a communication channel between agents which is interpretable like Dessì et al. (2021) in the case of single symbol communication.

1.3 Proposed Approach

We explore different ways of selecting these discrete symbols both with and without incorporating structure using independent fully connected network layers. Refer to Section 3.2 for more details about the architecture.

We use the metrics from Dessì et al. (2021) as the starting point to check the quality of the generated symbol. The details of these evaluation metrics are provided in Appendix F. For qualititive evaluation of learned features we used VISSL (Goyal et al., 2021) to benchmark the performance of trained backbone on downstream task of linear image classification.

2 Related Work

Our proposed work involves communication between two networks in order to understand the emergent communication strategy. Interpreting the communication strategy between the two networks also helps us to contribute to the larger goal of explainable AI (Xie et al., 2020) by decoding the language to understand the emergent communication developed between SENDER and RECEIVER networks in order to efficiently solve a given task.

Emergent communication between deep neural networks has been widely studied in the past. Lazaridou and Baroni (2020) provide a brief overview of recent communication strategies. Havrylov and Titov (2017) also did a similar experiment to ours where the agents played a referential game and allowed the communication to be a sequence of symbols using an LSTM network. Lazaridou et al. (2017) worked on referential game setup where there is a single distractor and is seen by both, the SENDER and the RECEIVER. In both of these networks, the image feature extraction network relied on pretrained convolutional network and hence, the systems relied on huge amounts of annotated data required for pretraining.

One of the major advantages of referential game setup is the visual representations that are induced as a by-product of the emergent communication. These representations are produced without requir-

ing any manual annotations. Interestingly, there is a very active independent line of research that focuses on how to induce good representations without supervision. Doersch et al. (2015) uses image-patch prediction as a pretraining objective to get better visual representations. Caron et al. (2019) employ hierarchical clustering for the self-supervision on large-scale uncurated data. Our work also benefits from leveraging a contrastive loss objective, and the idea of learning richer representations by using multiple views of the same image. Chen et al. (2020) uses data augmentation techniques for creating multiple views of the target image and trains with larger batch sizes, and demonstrates that applying a non-linear transformation before the contrastive loss benefits the model.

We build upon the previous work of Dessì et al. (2021) where the authors implemented an end-to-end discrimination game from scratch between SENDER and RECEIVER where the communication involved only a single discrete symbol. The authors observed that the visual representations that resulted from this unsupervised setting were of comparable quality to the self-supervised learning model SimCLR (Chen et al., 2020) and provided competitive accuracy when evaluated on downstream tasks. Our work furthers their research by relaxing the unit length constraint on the communication channel and replacing it with a fixed length symbol signal. Following Dessì et al. (2021), we use visual feature induction as proxy to test the performance of the system because visual features will give better downstream performance if the emergent language is highlighting genuine semantic aspects of the scene.

3 Architecture

We explain the fully connected network (FCN) (with and without structure) for ensuing effective multi-symbol discrete communication in detail. Our games have three major components, namely a SENDER Agent, COMMUNICATOR, and RECEIVER Agent as displayed in Figure 2.

3.1 SENDER Agent

The role of SENDER is to take the input target image $\mathcal{I}$ and output an encoding of the image f_I that needs to be sent to the COMMUNICATOR.

We use a ResNet-50 architecture from scratch for the SENDER for all our experiments.

3.2 COMMUNICATOR

The COMMUNICATOR component is the key component for all our experiments. We modify the network in the COMMUNICATOR as shown in Figure 2. Previous work (Dessì et al., 2021) used the COMMUNICATOR to transmit a single symbol message (one-hot vector). We extend the capabilities of the COMMUNICATOR by transmitting multiple symbols.

All of these networks use Gumbel-Softmax reparameterization in order to obtain a discrete message (one-hot like) from continuous feature vectors during training time (Jang et al., 2016; Maddison et al., 2016). This keeps the model differentiable and facilitates end-to-end training of all the networks.

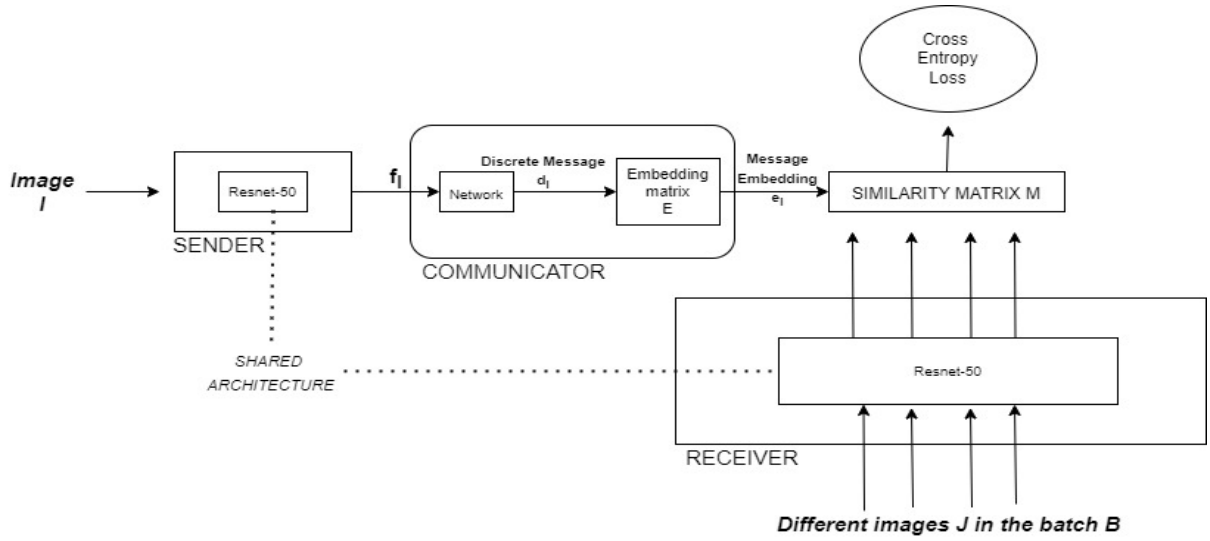

Figure 2: Architecture of the model.

During test-time, we use the $argmax$ function instead of Gumbel-Softmax since differentiability is not necessary.

3.2.1 Baseline: Single symbol communication

The COMMUNICATOR receives the feature vector f_I from the SENDER (output of ResNet50). This feature vector is then directly passed through a Gumbel-Softmax layer during training to get a one-hot-like sample d_I. During test time, we just take argmax of the feature vector and hence it contains only a single symbol. $d_I \in \mathbb{R}^{|V|}$ where $|V|$ is the vocabulary size of the model.

The output d_I is then passed through embedding matrix to get the continuous vector e_I corresponding to the transmitted symbol d_I.

$$d_I = \text{GumbelSoftmax}(f_I)$$

$$e_I = \text{Embedding}(f_I)$$

The message space here (total number of distinct messages that can be transmitted from SENDER here) is $|V|$ (growing linearly as $|V|$ increases). Finally, the vector e_I is then sent over to calculate the similarity and loss.

3.2.2 Fully connected network for multiple symbols

This is similar to baseline model but instead of a single symbol using one embedding matrix and one fully connected layer, we transmit multiple symbols using multiple embedding matrices and multiple fully connected layers as shown in Figure 3 . Let N be the total number of symbols transmitted. Now, we have N different fully connected and embedding layers.

$$g_i = \text{FCN}_i(f_I) \text{ where } i \in [1, N]$$

$$d_i = \text{GumbelSoftmax}(g_i) \text{ where } i \in [1, N]$$

$$e_i = \text{Embedding}_i(d_i) \text{ where } i \in [1, N]$$

$$d = \text{concat}(d_1, \cdots, d_i, \cdots, d_N)$$

$$e = \text{concat}(e_1, \cdots, e_i, \cdots, e_N)$$

The final discrete message is represented by d and the corresponding final embedding vector is e. The message space now is $|V|^N$ (exponential in $|V|$). This comes at a cost of increasing the parameters by a constant multiplicative factor of N.

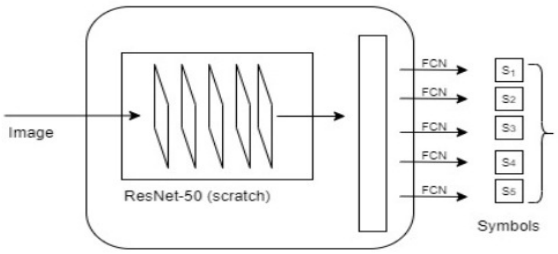

Figure 3: SENDER Architecture using Fully Connected Layers

3.2.3 Fully connected network for multiple symbols using Structured Communication

In the previous fully connected network that generated symbols, all the symbols were generated from the last layer of the ResNet-50. In the previous method, we have no control over the communicated symbols. Here, we try to incorporate structure externally into the communicated symbols.

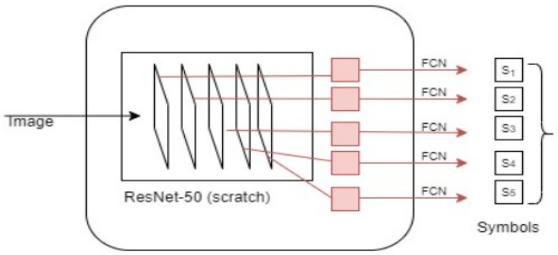

Figure 4: SENDER Architecture using Fully Connected Layers with Structured Communications

We enforce the structure by making the symbols rely on different layers of ResNet instead of just the last layer as shown in Figure 4. This ensures that symbols are generated from looking at different levels of information from the image. Zeiler and Fergus (2013) demonstrates the hierarchical nature of features in a trained convolutional neural network. Earlier layers learn low-level information about local regions in an image like edges, corners, shades, etc. whereas latter layers build upon the features of earlier layers and distinguish more complex invariances like textures, shapes, etc.

Motivated by this, we use different layers of the convolutional architecture ResNet to communicate different symbols to the receiver. This would ensure that symbols in the earlier position communicate about low-level information from local-regions in the image whereas symbols at latter positions communicate high level information about the entire global image. We place multiple exit points

(depending upon the number of symbols N transmitted) across equally spaced different layers of ResNet. If the layer is a convolutional layer, we flatten the output and call the feature vector as f_i for i^{th} exit point from the ResNet. We train a seperate FCN layer for every position that takes in input as the vector f_i from the corresponding exit point. Every layer learns to extract the necessary message from different ResNet layers (learning from fine-grained to coarse information)

Now the input g_i to Gumbel Softmax becomes:

$$g_i = \text{FCN}_i(f_i) \text{ where } i \in [1, N]$$

Rest of the architecture details about embedding and concatenation remains the same.

For both of the structured and unstructured models, we also experiment with a Shared-Vocabulary setting where weights of the embedding matrices (symbol vectors) are shared across all the positions (similar to human language). Refer to Section 4.3 for more details and results.

3.3 RECEIVER Agent

The architecture of the RECEIVER is similar to that of the SENDER. It takes in an image $\mathcal{J}$ and outputs a feature vector f_J that is used in order to calculate the loss. Similar to SENDER, FCN layer sits at the top of the ResNet head and the output of the layers are then concatenated. Unlike COMMUNICATOR, there are no Gumbel Softmax operations and no Embedding layers.

We use an untrained ResNet-50 architecture for all the experiments, and share the weights for SENDER and RECEIVER networks. We also experimented with not sharing weights and did not notice any significant difference between the two settings.

The ResNet architecture plays the common role of feature extractor from the images that is shared between the SENDER and the RECEIVER. Most of the joint learning for the task happens in these FCN layers and Embedding layer (in the COMMUNICATOR)

For the Structured communication model, the FCN layer in the RECEIVER, does not just sit at the output of ResNet but sits on top of every exit-point i in the ResNet and then the output of all FCN layers is concatenated to obtain the feature vector f_J

3.4 Loss

For all our experiments, we use cross-entropy loss over the model predictions and the ground truth. The ground truth label is simply the index i of the target image that the RECEIVER received among other images in the batch B. We treat all the other samples in the batch as distractors. This facilitates parallel computation and is more efficient because we do not compute f_I and f_J seperately. In order to make the prediction, the loss calculates the cosine similarity between feature vectors e_I (embedding vector of the discrete message that the COMMUNICATOR output after receiving the image $\mathcal{I}$) and f_J for all pairs of images $\mathcal{I}$ and $\mathcal{J}$ in the batch.
For input image $\mathcal{I}$, the prediction is -

$$j = \arg\max_{J \in B} \left(\frac{e_I . f_J}{||e_I||_2 \, ||f_J||_2} \right)$$

Here, B indicates the batch of images where one of the images in the batch (say, at index i) is an augmented view of input target image $\mathcal{I}$ that was sent to the SENDER. It is a correct prediction if $j = i$ or incorrect prediction, otherwise. The loss thus tries to maximize the cosine similarity if the image $\mathcal{I}$ and $\mathcal{J}$ are different views of the same image (augmented pairs) and penalizes the loss if they are two different images. We use a cross-entropy loss over the predictions and then train the entire model end-to-end from scratch.

We provide results in the below Section 4 after training the models with and without data augmentations. We use random perturbations, gaussian blurring, random cropping and resizing for data augmentation pipeline, motivated from (Dessì et al., 2021).

We also try an alternate architecture that relies on Recurrent Neural networks in the COMMUNICATOR. Further details about RNN are mentioned at Appendix D and E

4 Results

We show the results of our experiments on different datasets (CIFAR10, CIFAR100 (Krizhevsky, 2009), ImageNet (Russakovsky et al., 2015)) and evaluate the ensued communication protocol and the induced feature representation.

4.1 Results on CIFAR-10 and CIFAR-100

We have trained our FCN architecture and baseline using shared ResNet-50 in SENDER and RECEIVER on CIFAR-10 with and without data augmentation for 25 epochs and then tested the game accuracy on test set, below are the results in comparison with baseline, FCN architecture with number of symbols exchanged (nos) in communication are 2 and 3.

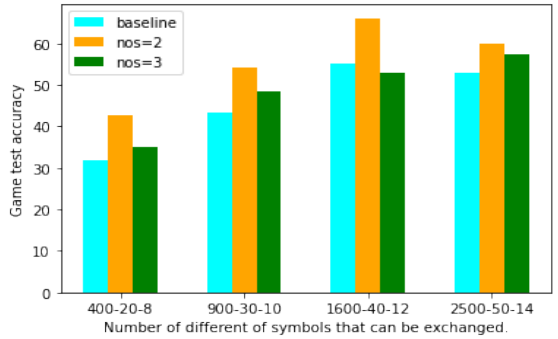

Figure 5: FCN vs baseline on CIFAR-10 without data augmentation

Table 1: Results without data augmentation.

Architecture	Dataset	Vocab size	NOS	Accuracy
Baseline	CIFAR-10	400	1	31.7
FCN	CIFAR-10	20	2	42.84
Baseline	CIFAR-10	1600	1	55
FCN	CIFAR-10	40	2	66.06
FCN	CIFAR-100	40	2	63.26

Table 2: Results with data augmentation.

Architecture	Dataset	Vocab size	NOS	Accuracy
Baseline	CIFAR-10	400	1	5.16
FCN	CIFAR-10	20	2	41.3
FCN	CIFAR-10	40	2	59.45
FCN	CIFAR-100	40	2	60.53

We have continued our analysis with FCN on CIFAR-10 using data augmentations. We have used random crop, random apply, random grayscale, random horizontal flip techniques to generate the augmented dataset. Table 1 and Table 2 show the results of observed game accuracy with and without data augmentation. We can notice that we are using vocabulary of different size when the number of symbols exchanged are 1 and 2. The rational behind choosing these numbers is because when there are 2 symbols then effective we have $|V|^2$ size of ways information can be exchanged between SENDER and RECEIVER. So for fair comparison we made sure this space matches for in both 1-symbol and 2-symbol game.

Table 3: Structured communication on CIFAR.

Architecture	Data Aug?	Vocab size	NOS	Accuracy
FCN	False	10	5	81.81
Structured FCN	False	10	5	83.68
FCN	True	10	5	80.87
Structured FCN	True	10	5	85.82

We have also performed same set of experiments on our model and baseline using CIFAR-100 and found that similar trends are holding which we have provided in Table 1 and Table 2

We then did experiments using Structured FCN architecture and recorded results shown in Table 3. We can observe that using explicit structure seems to helping the communication and resulting in the improved game accuracy compared to FCN which is unstructured.

Our takeaway from all the above experiments on CIFAR-10 and CIFAR-100 is that when we have multiple symbols being exchanged during SENDER and RECEIVER communication, we notice that there is an improvement in game accuracy compared to single symbol game. Moreover, adding explicit structure in the communication is performing better than simple communication.

4.2 Gaussian Test

We wanted to check if our SENDER and RECEIVER are sending low level pixel information. So we have used Gaussian Test for validating this. In this test, we have generated a image dataset containing 50k images which are constructed by drawing from the Gaussian space. We then used the models which are trained in Section 4.1 to evaluate on this generated dataset. Below shows the corresponding results for this test.

Table 4: Results of Gaussian test.

Architecture	Data Aug?	Vocab size	NOS	Accuracy
Baseline	False	1600	1	0.39
Baseline	True	1600	1	0.39
FCN	False	40	2	0.39
FCN	True	40	2	2.5

For all the above experimentation we have used a batch size of 256. So theoretically, if there is

a random model then the probability of correctly identifying the correct image from a batch containing distractors is 1/(batch size). In our case it should be 0.39%. We can observe from Table 4 that we have close to expected percentage. This indicates that our SENDER and RECEIVER are not sending pixel level information.

4.3 Effect of shared vocabulary setting on the Communication

We have come up with a shared vocabulary setting which can be used to control the capacity of the model. In this setting we are sharing the embeddings of the symbols over different timesteps. For example this can be thought as Symbol 1 produced at the first time step or the third time step has same meaning (similar to human languages).

Table 5: Results of shared vocabulary on CIFAR with NOS=5, Vocab size = 10.

Architecture	Data Aug?	Accuracy
FCN	False	95.46
Structured FCN	False	92.64
FCN	True	88.04
Structured FCN	True	88.96

We can compare the results from Table 3 and Table 5. We observed that as we used the shared vocabulary setting there is a significant improvement in the performance of game, keeping all other parameters like NOS and Vocab size same. We think that since CIFAR dataset is small but our model (without shared vocabulary) has huge capacity, it might be slightly overfitting to train set which effects it's generalization capability to test set. However, when we use shared vocab setting since we are reducing the capacity of the model it might be able to generalize well compared to other.

4.4 Results on ImageNet

We also run the experiments on ImageNet (Russakovsky et al., 2015) data containing 1.2 million images. We used FCN architecture with NOS (Number of symbols) being 2 and total vocabulary size 2048 and the results are showed in 6.

We can see that there is a huge improvement in the game accuracy. Thus validating our hypothesis that adding more symbols improves the communication compared to single symbols. Given the limited time and compute resources we weren't able to do the similar experiments using structured FCN on Imagenet (which we plan to do as mentioned in future work).

Table 6: ImageNet results.

Architecture	Vocab size	NOS	Accuracy
Baseline (No augmentation)	2048	1	92.2
Baseline (with augmentation)	2048	1	82.2
FCN (No augmentation)	2048	2	99.69
FCN (with augmentation)	2048	2	97.34

4.5 Unsupervised benchmark evaluation using VISSL

VISSL is a computer vision library to design new self-supervised tasks and evaluate the learned representations. Using VISSL, we evaluated SENDER's CNN layers feature representation quality on the downstream task of **Linear Image classification**. We first extracted ResNet50 layers from the trained SENDER and attached an MLP head to it. Keeping the weights of ResNet50 layers fixed we trained MLP head using given default benchmark hyperparameters and measured the top 1% accuracy.

Table 7: Linear Image Classification benchmark evaluation with VISSL.

Model	Vocab size	NOS	Accuracy
ResNet50_S	–	–	48.36
Baseline	1600	1	39.816
FCN	40	2	41.872
Structured FCN	10	5	59.3

Table 8: VISSL accuracy for FCN models trained on CIFAR-100 with structure and shared vocabulary.

Vocab Size	NOS	Data Aug	Shared	Accuracy
10	5	No	No	18.3
10	5	No	Yes	15.8
10	5	Yes	No	36.2
10	5	Yes	Yes	33.9

In the table 7 the first row shows the quality of trained features of ResNet-50 trained with supervision from scratch on the CIFAR-10 dataset. We trained for with a batch size of 128, weight decay of 1e-4, learning rate of 0.1, and momentum of 0.9, as mentioned in original paper by (He et al., 2015). We only trained for 800 epochs and speculate that the supervised ResNet50 VISSL accuracy could be slightly higher for more epochs. The second and third columns show the benchmark accuracy of

the ResNet-50 backbone trained with the discrimination game with single (baseline) and multiple (FCN) symbols. We observe that learned representation quality is better with multiple symbols. The fourth row shows the accuracy of structured communication. We found that the accuracy with structured communication is significantly better than the baseline and FCN approaches, implying that adding explicit structure in communication improves the quality of learned features.

We perform similar experiments for CIFAR100 data and show the results in table 8. We confirm our assumption that the structured communication in general helps to improve the quality of learned visual features.

4.6 Inter-Class Qualitative Analysis of Frequently Communicated Symbols

To better understand the nature of the learned language, we examined the most frequently communicated symbol for each of the class labels in the CIFAR-10 dataset. Particularly we plot the frequency of each symbol in the vocabulary used to communicate images with the same labels for each position in message. We have mentioned some plots in Appendix A. Table 9 shows our analysis for 2-symbol communication models trained with and without data augmentations.

Table 9: Most Frequently communicated Symbols for models trained on CIFAR-10 with 2 symbol communication.

Super Class	Class Label	W/ Data Aug	W/o Data Aug
Animals	Cat	[#6 , **#5**]	[* , **#0**]
Animals	Dog	[#1 , **#5**]	[* , **#0**]
Animals	Deer	[#2 , **#5**]	[* , **#0**]
Animals	Horse	[#2 , **#5**]	[* , **#0**]
Automobiles	Automobiles	[#8 , **#7**]	[* , **#0**]
Automobiles	Truck	[#6 , **#7**]	[* , **#0**]

For the models with data augmentation, we observed that similar symbols are communicated for classes with a similar semantic structure. For instance, **symbol#5** was used in position#2 for all 'animal' labels namely cat, dog, deer and horse, indicating that communication carries semantic information about the structure of four legged terrestrial animals. We observe similar characteristics for "automobiles" and "trucks", suggesting that they carry semantic information about the structure. In contrast, for the model without data augmentation, **symbol#0** is being communicated for all the labels at position #2.

Thus we can infer that with data augmentation, the quality of communication is better, the symbols are more disentangled and carry high-level semantic meaning. This also suggests that with data augmentation, the developed language is more interpretable.

We also give details about the Analysis of communicated symbols in the Appendix section B and C.

5 Conclusion and Future Work

Below are the set of conclusions, we draw from all the experiments that we did and analysis on the quality of communication channel.

- Adding multiple symbols to communication channels helps SENDER and RECEIVER communicate more effectively compared to our baseline (which uses single symbol).
- Enforcing explicit structure in the communication seems to improve performance of the model. This seems a promising direction which can be explored further.
- The quality of visual embeddings learned in the process of self-supervised training are on par with those of supervised setup.
- Data augmentation helps the model generalize well which results in generating more interpretable communication patterns.

In the future, one can explore the game accuracy for Structured FCN trained on ImageNet data and see how it affects the downstream performance using VISSL and compares to other baseline and unstructured approaches. Moreover, we only demonstrate the effectiveness of the communication strategy using Qualitative analysis. An interesting possible direction is to develop quantitative metrics that can evaluate how interpretable the communicated symbols are.

6 Acknowledgements

We would like to thank our industry mentors **Roberto Dessi** and **Marco Baroni** for proposing an innovative project to work on and providing the background and related help needed in laying a direction to the work. We also want to thank our postdoc mentor **Michael Borakto** for providing all the inputs and suggestions.

References

Mathilde Caron, Piotr Bojanowski, Julien Mairal, and Armand Joulin. 2019. Unsupervised pre-training of image features on non-curated data.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations.

Li Deng. 2012. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142.

Roberto Dessì, Eugene Kharitonov, and Marco Baroni. 2021. Interpretable agent communication from scratch (with a generic visual processor emerging on the side).

Carl Doersch, Abhinav Gupta, and Alexei A. Efros. 2015. Unsupervised visual representation learning by context prediction. *CoRR*, abs/1505.05192.

Priya Goyal, Quentin Duval, Jeremy Reizenstein, Matthew Leavitt, Min Xu, Benjamin Lefaudeux, Mannat Singh, Vinicius Reis, Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Ishan Misra. 2021. Vissl. `https://github.com/facebookresearch/vissl`.

Serhii Havrylov and Ivan Titov. 2017. Emergence of language with multi-agent games: Learning to communicate with sequences of symbols.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep residual learning for image recognition.

Eric Jang, Shixiang Gu, and Ben Poole. 2016. Categorical reparameterization with gumbel-softmax.

Najoung Kim and Tal Linzen. 2020. Cogs: A compositional generalization challenge based on semantic interpretation.

Alex Krizhevsky. 2009. Learning multiple layers of features from tiny images. Technical report.

Angeliki Lazaridou and Marco Baroni. 2020. Emergent multi-agent communication in the deep learning era.

Angeliki Lazaridou, Alexander Peysakhovich, and Marco Baroni. 2017. Multi-agent cooperation and the emergence of (natural) language.

Yuanpeng Li, Liang Zhao, Jianyu Wang, and Joel Hestness. 2019. Compositional generalization for primitive substitutions. *arXiv preprint arXiv:1910.02612*.

Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. 2016. The concrete distribution: A continuous relaxation of discrete random variables.

Jason Tyler Rolfe. 2016. Discrete variational autoencoders.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Li Fei-Fei. 2015. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115:211–252.

Ning Xie, Gabrielle Ras, Marcel van Gerven, and Derek Doran. 2020. Explainable deep learning: A field guide for the uninitiated. *CoRR*, abs/2004.14545.

Matthew D Zeiler and Rob Fergus. 2013. Visualizing and understanding convolutional networks.

Appendix A Intra-Class Symbol Statistics

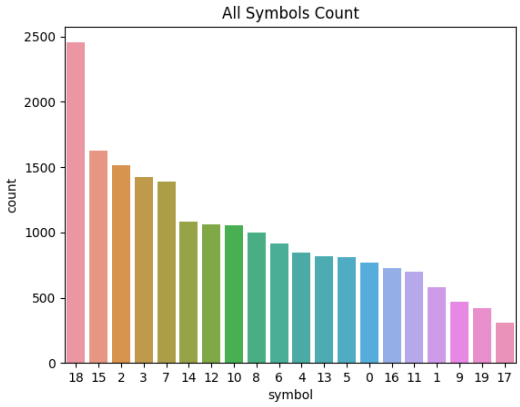

Figure 6: Number of all symbols used to describe label 0.

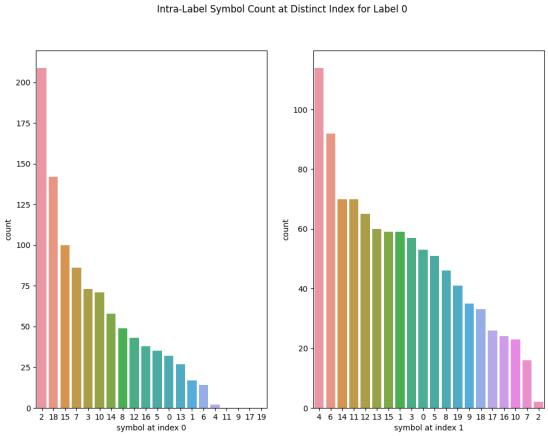

Figure 7: Number of all symbols at two different locations in the message vector. Sorted in descending order.

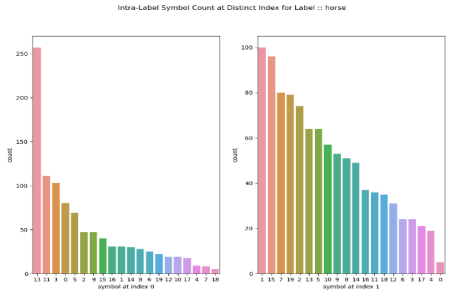

Figure 8: Symbol statistics for horse image.

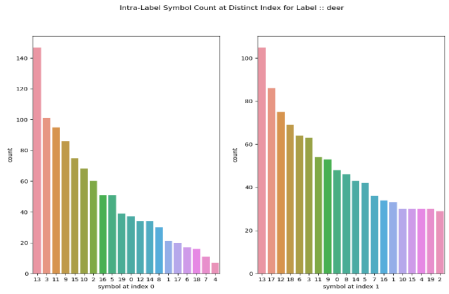

Figure 9: Symbol statistics for deer image.

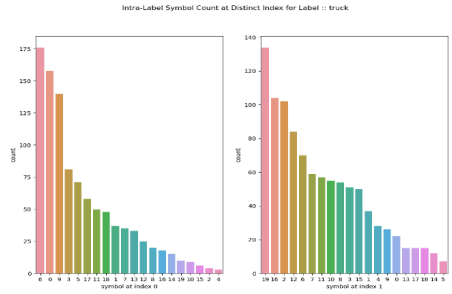

Figure 10: Symbol statistics for truck image.

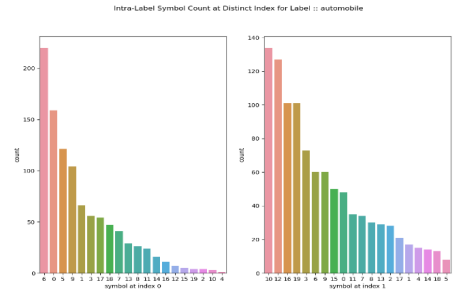

Figure 11: Symbol statistics for automobile image.

Appendix B Inter-Class Embedding and Symbol Evaluation

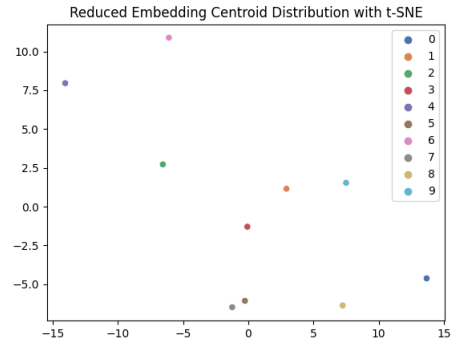

Figure 12: Embedding centroid visualization after t-SNE dimensionality reduction. For each class, the centroid is the geometric center of reduced embeddings.

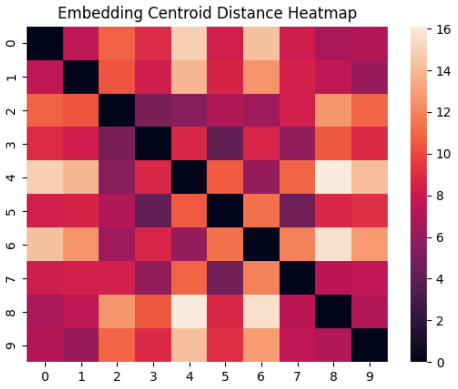

Figure 13: Embedding centroid distance heatmap. The embeddings are from the original sender resnet output. The distance metric is Euclidean distance.

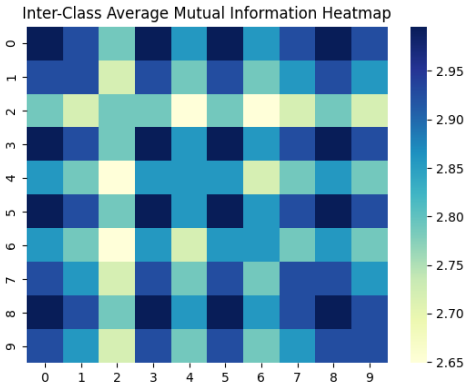

Figure 14: Mutual information heatmap between classes. We use distribution in Figure 7 to compute mutual information between classes.

Appendix C Symbol Analysis

In appendix A and B, we have created some plots to give intuition about the communicated symbols. The plots come from a FCN-CIFAR10 experiment with vocabulary size set to 20 and message length set to 2. There are two sorts of plots: intra-class and inter-class. The reason to create these plots is for preliminary analysis about how the SENDER can describe a single class, and different classes. There are some patterns may be noticed:

- For a single class, the message has some different symbol distributions on distinct locations. The symbol 2 dominates the first location and symbol 4 dominates the second. And in the first location the symbol 4 rarely appears. There seems to be some disentanglement between distinct locations in the multi-symbol message vector.
- Some classes sharing similar embedding patterns also share similar mutual information patterns with their intra-class symbol distributions. For example, class 4 and 6 share similar patterns in the 2 heatmaps, and they are closest to each other in t-SNE embedding space. This indicates some connection between embedding space and symbol space of sender.

Appendix D Recurrent neural network for multi-symbol communication

The architecture of fixed length multi-symbol communication using a recurrent neural network is shown in Figure 15. The SENDER takes an image and produces the feature vector f_I. This feature vector f_I is then passed to COMMUNICATOR by projecting the feature vector to the hidden space of the RNN. In COMMUNICATOR, there are sequence of RNN cells, each cell takes responsibility of generating a symbol at the given time step by taking the hidden dimension vector at the previous time step and output from previous RNN cell as input and produces the symbol at the current time step. This process is repeated N times to produce a discrete message d which is concatenation of N symbols. The message d is then passed to RECEIVER, which unrolls the symbols and then uses another RNN network to get a final embedding e representing the message d. The produced final embedding e is then used in the loss to find the true image from the set of distractors.

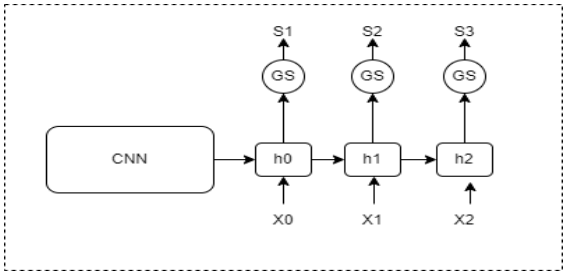

Figure 15: SENDER Architecture using RNN Cells. GS represents Gumbel-Softmax

We have tried applying RNN while training the game but then noticed that the training loss didn't decrease as the training progressed. We have did a lot of debugging to figure out the reasons but we aren't successful in doing so. So, keeping time

constraint in mind we have worked on the other approaches i.e FCN and Structured FCN.

Appendix E MNIST Reconstruction

In order to check for the efficacy of multi-symbol communication game, we first check the results on a relatively small and easier to analyze data of MNIST (Deng, 2012). Here, instead of a discrimination game, we make the agents play a reconstruction game and provide the results and evaluations on our RNN model.

Details of Reconstruction Game: In reconstruction games, the task of the RECEIVER changes from discrimination to reconstruction. The SENDER agent and Communicator stay the same, but the RECEIVER agent takes in the message from Communicator and tries to reconstruct the target image I that was input to the the SENDER. The game thus becomes very similar to learning discrete latent representations with variational autoencoders. (Rolfe, 2016).

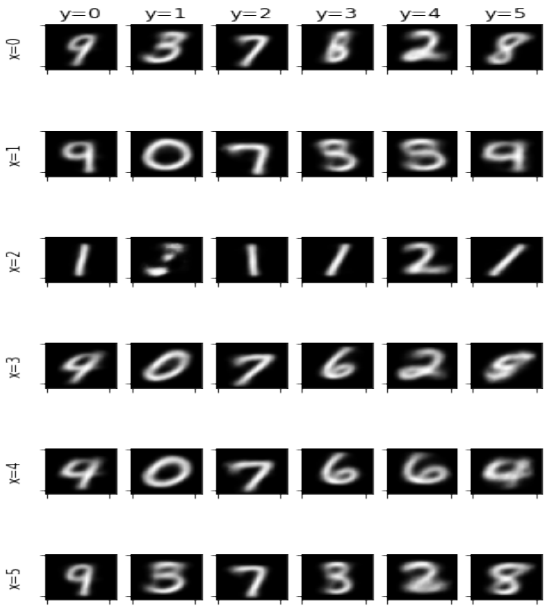

Figure 16: Receiver generated Image for given symbols

In this experiment, we have trained a fixed 2 symbol communication game using the RNN architecture. As mentioned the RECEIVER here takes the symbols that are generated by SENDER and tries to reconstruct the image. Below figure shows on y-axis symbol 1 and on x-axis symbol 2 which are given as input to RECEIVER after training the network for 15 epochs. There are a total of 6 symbols (vocabulary size of the game). We can observe that the RECEIVER component is able to reconstruct the images from the discrete symbols.

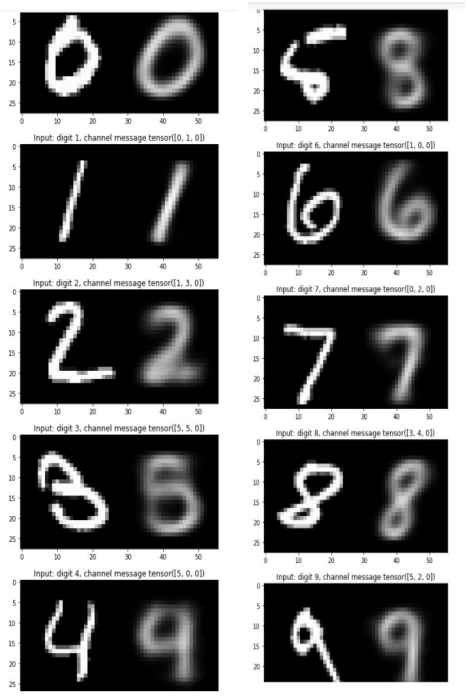

Figure 17: Reconstructions of MNIST input

Appendix F Evaluation Metrics for learned protocol

Since, (Dessì et al., 2021) is our primary baseline we will be using similar evaluation metrics in our work. We also plan on experimenting with newer metrics for evaluating the interpretability and disentanglement of the multi-symbol protocol.

Gaussian blob sanity check: Communication symbols may learn to refer to pixel-level aspects of the image rather than capturing semantic information, implying that they developed an opaque communication protocol. To avoid this, we will freeze the trained models and let them play communication game with random images with pixel values drawn from Gaussian distribution $\mathcal{N}(0, 1)$. The sane communication protocol would have a very low accuracy.

Normalized Mutual Information(nMI): Since we are training with the ImageNet dataset we will calculate nMI between the ground truths of images and the symbols generated by the trained Sender. The nMI of two variables is obtained by dividing their MI by their average entropy, ranging between 0 and 1.

Similarity Measure: We will calculate the similarity between two categories based on their short-

est path in WordNet is-a taxonomy. Precisely, we will calculate the average shortest path between the ground truth categories of images sharing the same symbols. The intuition behind this score is to check the Sender using the same set of symbols for dissimilar ground truth categories(truck and dog).