Interpretable agent communication from scratch (with a generic visual processor emerging on the side)

Anonymous Author(s)
Affiliation
Address
email

Abstract

As deep networks begin to be deployed as autonomous agents, the issue of how they can communicate with each other becomes important. Here, we train two deep nets from scratch to perform realistic referent identification through unsupervised emergent communication. We show that the largely interpretable emergent protocol allows the nets to successfully communicate even about object types they did not see at training time. The visual representations induced as a by-product of our training regime, moreover, show comparable quality, when re-used as generic visual features, to a recent self-supervised learning model. Our results provide concrete evidence of the viability of (interpretable) emergent deep net communication in a more realistic scenario than previously considered, as well as establishing an intriguing link between this field and self-supervised visual learning.

1 Introduction

As deep networks become more effective at solving specialized tasks, there has been interest in letting them develop a language-like communication protocol so that they can flexibly interact to address joint tasks [1]. One line of work within this tradition has focused on what is arguably the most basic function of language, namely to point out, or *refer to*, objects through discrete symbols. Such ability would for example allow deep-net-controlled agents, such as self-driving cars, to inform each other about the presence and nature of potentially dangerous objects, besides being a basic requirement to support more advanced capabilities (e.g., denoting relations between objects).

While discreteness is not a necessary prerequisite for agent communication [2, 3], practical and ethical problems might arise if communication is incomprehensible to humans. A discrete code analogous to language is certainly easier to decode for us, helping us to understand the agents' decisions, and ultimately contributing to the larger goal of explainable AI [4].

In this paper, we study emergent discrete referential communication between two deep network agents that are trained from scratch on the task. We observe that the referential discrimination task played by the networks is closely related to pretext contrastive objectives used in self-supervised visual representation learning [5–8]. We exploit this insight to develop a robust end-to-end variant of a communication game. Our experiments confirm that, in our setup: i) the nets develop a set of discrete symbols allowing them to successfully discriminate objects in natural images, including novel ones that were not shown during training; ii) these symbols denote interpretable categories, so that their emergence constitutes a form of fully unsupervised image annotation; iii) the visual features induced as a by-product can be used as high-quality general-purpose representations, whose performance in various object classification tasks is not lagging much behind that of features induced by a popular self-supervised representation method specifically designed for this task.

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

2 Background

2.1 Deep net emergent communication

There has recently been interest in letting deep nets communicate through learned protocols. This line of work has addressed various challenges, such as communication in a dynamic environment or how to interface the emergent protocol with natural language (see [1] for a recent survey). Probably the most widely studied aspect of emergent communication is the ability of deep net agents to use the protocol to refer to objects in their environment [e.g., 9–16].

The typical setup is that of a referential, or discriminative, communication game. In the simplest scenario, which we adopt here, an agent, the Sender, sees one input (the target) and it sends a discrete symbol to another agent, the Receiver, that sees an array of items including the target, and has to point to the latter for communication to be deemed successful. Importantly, task success is the only training objective; the communication protocol emerges purely as a by-product of game-playing, without any direct supervision on the symbol-transmission channel.

In one of the earliest papers in this line of research, Lazaridou et al. [9] used images from ImageNet [17] as input to the discrimination game; Havrylov and Titov [11] used MSCOCO [18]; and Evtimova et al. [12] used animal images from Flickr. While they used natural images, all these studies were limited to small sets of carefully selected object categories. Moreover, in all these works, the agents processed images with convolutional networks pretrained on supervised object recognition. While this sped up learning, it also meant that all the proposed systems *de facto* relied on the large amount of human annotated data used for CNN training. Lazaridou et al. [13] and Choi et al. [14] dispensed with pre-trained CNNs, but they used synthetically generated geometric shapes as inputs.

Results on the interpretability of symbols in games with realistic inputs have generally been mixed. Indeed, Bouchacourt and Baroni [19] showed that, after training Lazaridou et al. [9]'s networks on real pictures, the networks could use the learned protocol to successfully communicate about blobs of Gaussian noise, suggesting that their code (also) denoted low-level image features, differently from the general semantic categories that words in human language refer to. In part for this reason, recent work tends to focus on controlled symbolic inputs, where it is easier to detect degenerate communication strategies, rather than attempting to learn communication "in the wild" [e.g., 10, 15, 16].

2.2 Self-supervised representation learning

Self-supervised learning of general-purpose visual features has received much attention in recent years. The main idea is to train a visual network on a pretext task that does not require manual annotation. After convergence, the net is used to extract high-quality features from images, to be applied to various "downstream" tasks of interest. This is often done by training a simple classifier on top of the frozen trained architecture [20–23].

Early models used image patch prediction as the proxy task [8, 24]. Recent work has instead focused on an instance-level contrastive discrimination objective [5–7, 25]. Two symmetric networks encode different views of the same input images obtained through a stochastic data augmentation pipeline. Optimization is done with variants of the InfoNCE loss [24, 26], that tries to maximize similarity among representations of the same image while minimizing similarity of different ones.

Interestingly, the contrastive pretext task is very close to the one of identifying a target image among distractors, as in the standard referent discrimination game of emergent communication. The influential SimCLR model proposed by Chen et al. [5] is particularly similar to our setup. It uses two twin networks with a shared convolutional module optimizing the (dis)similarity of sets of target/distractor images. The main conceptual differences are that there is no discrete bottleneck imposed on "communication" between the networks, and there is no asymmetry, so that both networks act simultaneously as Sender and Receiver (both networks produce a continuous "message" that must be as discriminative for the other network as possible).

We rely here on the connection with self-supervised learning in two ways. First, we import the idea of data augmentation from this literature into the communication game, showing how it helps in evolving a more semantically interpretable protocol. Second, we evaluate the discrimination game as a self-supervised feature extraction method. We find that the visual features induced by the CNNs embedded in our agents are virtually as good as those induced by SimCLR, while the emergent

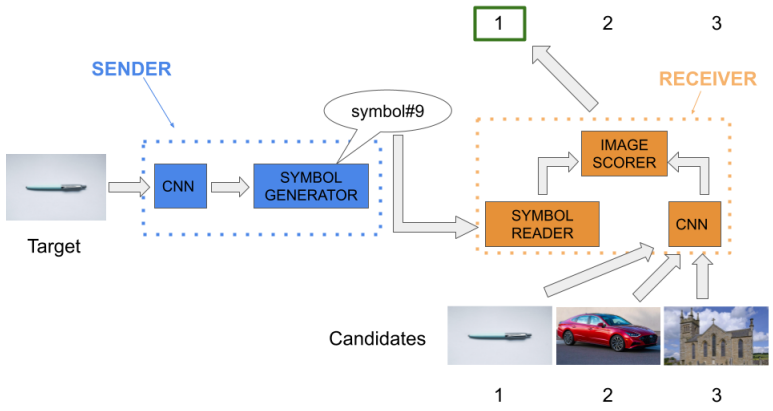

Figure 1: Game setup and agent architectures. Image sources: `https://unsplash.com`

protocol is better for communication than the one obtained by adapting SimCLR to the discrete communication setup.

3 Setup

3.1 The discrimination game

A Sender network receives as input a *target* picture, and it produces as output one of $|V|$ *symbols*. A Receiver network receives in input this symbol, as well as a list of n pictures, one of them (randomly placed in the ith position of the list) being the same target presented to the Sender. Receiver produces a probability distribution of cardinality n, interpreted as its guess over the position of the target. The guess is correct iff Sender concentrates the largest probability mass on the ith position, corresponding to the target slot.

Agent architecture Agent architecture and game flow are schematically shown in Fig. 1. Sender reads the target image through a convolutional module, followed by a one-layer network mapping the output of the CNN onto $|V|$ dimensions and applying batch normalization [27], to obtain vector $\mathbf{v}$. Following common practice when optimizing through discrete bottlenecks, we then compute the Gumbel-Softmax continuous relaxation [28, 29], which was shown to also be effective in the emergent communication setup [11]. At train time, Sender produces an approximation to a one-hot symbol vector with each component given by $m_i = \frac{\exp[(s_i+v_i)/\tau]}{\sum_j \exp[(s_j+v_j)/\tau]}$, where s_i is a random sample from Gumbel(0,1) and v_i a dimension of $\mathbf{v}$. The approximation is controlled by temperature parameter τ: as τ approaches 0, the approximation approaches a one-hot vector, and as τ approaches $+\infty$, the relaxation becomes closer to uniform. Importantly, at test time the Sender's output is generated by directly argmax-ing $\mathbf{v}$, so that it is a discrete one-hot vector indexing one of $|V|$ possible symbols.

Receiver passes each input image through its visual module (a CNN architecture), followed by a two-layer MLP with batch normalization and ReLU after the first layer [30]. It then computes temperature-weighted cosine scores for the linearly embedded symbol compared to each image representation. The resulting vector of cross-modal (symbol-image) similarities is transformed into a probability distribution over which image is the likely target by applying the softmax operation.

For both Sender and Receiver, we use ResNet-50 [31] as visual module. As they are different agents, that could (in future experiments) have very different architectures and interact with further agents, the most natural assumption is that each of them does visual processing with its own CNN. We consider however also a setup in which the CNN module is shared (closer to earlier emergent-communication work, where the agents relied on the same pre-trained CNN).

Optimization Optimization is performed end-to-end and the error signal, backpropagated through Receiver and Sender, is computed using the cross-entropy cost function by comparing the Receiver's output with a one-hot vector representing the position of the target in the image list.

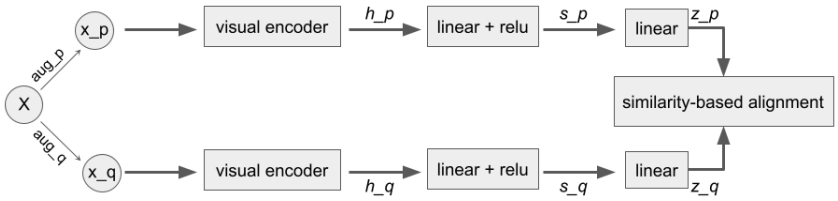

Figure 2: The SimCLR architecture. *aug_p* and *aug_q* are outputs of the stochastic augmentation pipeline used to generate two views of the same image.

SimCLR as a comparison model Given the similarity between the referential communication game and contrastive self-supervised learning in SimCLR [5], we use the latter as a comparison point for our approach. Fig. 2 schematically shows the SimCLR architecture. The crucial differences between SimCLR and our communication game are the following. i) In SimCLR, the agents are parameterized by the same network, that is, the visual encoder and transformation modules in the two branches of Fig. 2 are instances of the same net. ii) The setup is fully symmetric. Like our Receiver, both agents get a set of images in input, and, like our Sender, both agents can be seen as producing "messages" representing each input image. iii) Instead of (a probability distribution over) symbols, the exchanged information takes the form of continuous vectors (s in the figure). iv) The loss is based on directly comparing embeddings of these continuous vectors (z in the figure), maximizing the similarity between pairs representing the same images (positive examples in contrastive-loss terminology) and minimizing that of pairs representing different images (negative examples). This differs from our loss, that maximizes the similarity of the Receiver embedding of the Sender-produced discrete symbol with its own representation of the target image, while minimizing the similarity of the symbol embedding with its representation of the distractors.

Data augmentation In the original discrimination game proposed in [9], the agents are shown exactly the same target image. In self-supervised learning, on the other hand, it is common practice to "augment" images in different ways, e.g., by applying different croppings or color perturbations [5, 32, 33]. In standard contrastive learning frameworks, where all the weights are shared and there is no communication bottleneck, it is necessary to create these different views, or else the system would trivially succeed at the pretext contrastive task without any actual feature learning. We conjecture that data augmentation, while not strictly needed, might also be beneficial in the communication game setup: presenting different views of the target to Sender and Receiver should make it harder for them to adopt degenerate strategies based on low-level image information [19].[1] We follow the same data augmentation pipeline as [5], stochastically applying crop-and-resize, color perturbation, and random Gaussian blurring to every image.

Implementation details All hidden and output layers are set to dimensionality 2048.[2] Note that this implies $|V| = 2048$, more than double the categories in the dataset we use to train the model (see Section 3.2 below), to avoid implicit supervision on optimal symbol count. We fix Gumbel-Softmax temperature at 5.0, and Receiver cosine temperature at 0.1. The latter value is also used for the equivalent τ parameter in the NTXent-loss of our SimCLR implementation.

We train with mixed precision [34] for 100 epochs, with a batch of size $16 \times 128 = 2048$, divided across 16 GPUs. Rather than sampling distractors from the entire dataset, we take them from the current device's batch, thus playing the communication game with 127 distractor images in all reported experiments. We do not share distractors (negative samples) across devices. As in SimCLR, we use the LARS optimizer [35] with linear scaling [36], resulting in an initial learning rate of 2.4.

[1] Lazaridou and colleagues [9] also considered a variant of the game in which the agents see different pictures of the same category (e.g., the shared target is *dog*, but the agents get different dog pictures). This version of the game is however severely limited by the requirement of manual category annotation. Lazaridou et al. [13] also provide different images to Sender and Receiver, by feeding them different viewpoints of the same synthetically generated objects: again, a strategy that will not scale up to natural images.

[2] This is the same size used in the original SimCLR paper, except for the nonlinear projection head. For the latter, a number of sizes were tested, and the authors report that they do not impact final performance. We use 2048 for direct comparability with our setting.

We apply a cosine decay schedule without warmup nor restart [37]. Compute requirements are reported in Appendix A.1. All models are implemented with the EGG toolkit [38].[3]

3.2 Data

Data for model training Targets and distractors are sampled from the ILSVRC-2012 training set [39], containing 1.3M natural images from 1K distinct categories.

Referential game testing We use two image sources for testing. First, we use the ILSVRC-2012 validation set, containing around 50K images from the same categories as the training data. Second, in order to probe if the emergent protocol possesses a language-like ability to seamlessly refer to new things, we introduce a new "out-of-distribution" dataset (henceforth, the OOD set). To build the latter, we relied on the whole ImageNet database [17], exploiting its WordNet-derived hierarchy [40]. In particular, we randomly picked (and manually sanity-checked) 80 categories that were neither in ILSVRC-2012 nor hypernyms or hyponyms of ILSVR-2012 categories (e.g., since *hamster* is in ILSVR-2012, we avoided both *rodent* and *golden hamster*). We also attempted to sample categories of comparable degree of generality to those in ILSVRC-2012. For each of the categories chosen according to these criteria, we randomly sampled 128 images from ImageNet. Examples of included categories are *eucalyptus*, *amoeba*, and *drawer*.[4]

Linear evaluation of visual features on downstream tasks Following standard practice in self-supervised learning [e.g., 5, 20, 41], we evaluate the visual features induced by the CNN components of our models by training a linear object classifier on top of them. We use four common data-sets: ILSVRC-2012, Places205 [42], iNaturalist2018 and VOC07.[5] Evaluation is carried out with the VISSL toolkit [43],[6] adopting the hyperparameters in its configuration files without changes.

4 Experiments

4.1 Referential communication accuracy

We start by analyzing how well our models learn to refer to object-depicting images through a learned protocol.[7] While some models use data augmentation at training time, we do not apply this transformation when testing the learned communication pipeline.

As a strong baseline, we let the trained SimCLR model play the referential game by argmax-ing its s layer (see Fig. 2 above) into a discrete "symbol" (SimCLR$_{disc}$).[8]

Accuracy is given by the proportion of times in which a system assigns the largest symbol-embedding/image-representation similarity to the target compared to 127 distractors (chance $\approx 0.8\%$).

Results are in Table 1. The ILSVRC-val column shows that all models can play the game well above chance, when tested on new images of the same categories encountered during training. The next column (OOD set) shows that the models also play the game well above chance with input images from new categories, although mostly with a drop in performance. All variants of our model are considerably more robust than the SimCLR baseline (which, however, does remarkably well, out-of-the-box, at this discrete communication game).

[3] `https://github.com/facebookresearch/egg`. Our code is included as supplementary material, and it will be open-sourced upon paper acceptance under the same MIT license used by EGG.

[4] We uploaded a supplementary file with paths to the ImageNet images in the OOD set and the corresponding categories. We will make it publicly available upon paper acceptance.

[5] `http://places.csail.mit.edu/index.html`, `https://www.kaggle.com/c/inaturalist-2018`, `http://host.robots.ox.ac.uk/pascal/VOC/voc2007/`

[6] `https://vissl.ai/`

[7] Appendix A.2 reports this and all following experiments repeated with 5 distinct initializations of our most representative model (+*augmentations -shared*). It shows that variance across runs is negligible. We did not repeat the check for the remaining models due to time and resource constraints (see Appendix A.1).

[8] When looking at SimCLR as a communication game, s constitutes the discrete symbol layer, with z functioning as symbol embedding layer. Discretizing h produced worse game-playing performance in any case.

	ILSVRC-val	OOD set	Gaussian Blobs
SimCLR$_{disc}$	56.9%	47.4%	0.8%
Communication Game			
-augmentations -shared	91.2%	90.8%	43.4%
-augmentations +shared	92.8%	92.7%	84.7%
+augmentations -shared	81.5%	72.0%	0.8%
+augmentations +shared	82.2%	73.7%	0.8%

Table 1: Game-playing accuracy. $\pm$ *augmentations* marks whether the game was trained with data augmentations or not. $\pm$ *shared* indicates whether there was CNN weight sharing or not between Sender and Receiver.

In summary, neural networks trained from scratch are able to communicate quite accurately through a discrete channel even in the challenging setup in which the target referent is mixed with more than one hundred distractors, and when it belongs to new categories not seen at training time.

Looking at model variants, sharing CNN weights or not makes little difference (an encouraging first step towards communication between widely differing agents, that will obviously not be able to share weights). On the other hand, data augmentations harm performance.

However, it turns out that the better performance of the non-augmented models is due to an opaque communication strategy in which the agents are evidently referring to low-level aspects of images (perhaps, specific pixel intensity levels?), and not to the high-level semantic information they contain (ideally, object categories). To show this, we replicated the sanity check from [19]. We freeze the trained models and let them play the communication game with blobs of Gaussian noise as targets and distractors. We use 384 batches of 128 224x224-sized random images whose pixels are drawn from the standard Gaussian distribution $\mathcal{N}(0,1)$, for a total of 49152 items (a size comparable to that of ILSVRC-val). Results are in the last column of Table 1.

The game-based models with augmentations and SimCLR fully pass the sanity check, with performance exactly at the 0.8% chance level. The *-augmentation* models, on the other hand, are able to use the symbols they learned from sane input to communicate about the Gaussian blobs, showing that they developed a deeply opaque protocol. The *-agumentations +shared* model, in particular, is hardly affected by the switch to noise data. This is the closest setup to previous emergent communication studies, in which no data augmentation is applied and the agents share the same pre-trained CNN. It is no surprise, then, that protocol interpretability results are often mixed.

4.2 Protocol analysis: emergent communication as unsupervised image annotation

The Gaussian blob test suggests that the *+augmentations* models do not fall into the trap of a degenerate low-level protocol. However, it is not sufficient to conclude that they learned to associate symbols with human-meaningful referents. To test whether this is the case, we exploit the fact that, as we are working with ImageNet data, we have labels denoting the objects depicted in the images. We use this information in two ways. We compute the normalized mutual information (*nMI*) between the ground-truth labels of target images and the symbols produced for the same images by the trained Sender. The nMI of two variables is obtained dividing their MI by their average entropy, and it ranges between 0 and 1. We also compute a normalized similarity measure based on the shortest path between two categories in the WordNet is-a taxonomy. Our *WNsim* score is the average shortest-path similarity of the ground-truth categories of all target pairs that share the same Sender symbol, and it also ranges between 0 and 1. WNSim is more nuanced than nMI, as it will penalize less a Sender using the same symbol for similar categories, such as cats and dogs, than one using the same symbol for dissimilar ones, such as cats and skyscrapers. WNsim is computed with NLTK [44].

We again take SimCLR$_{disc}$ as a comparison point, where a "symbol" is simply the dimension with the largest value on a certain layer. To give this approach its best chance, we evaluated its h, s and z layers (see Fig. 2), and report statistics for h (CNN output), as it produced the best overall scores across data sets. We also run k-means clustering on the h layer (SimCLR$_{kmeans}$). We tried

	ILSVRC-val			OOD set		
	| P |	nMI	WNsim	| P |	nMI	WNsim
SimCLR$_{disc}$	1489	0.49	0.11	1069	0.46	0.19
SimCLR$_{kmeans}$	2035	0.59	0.18	1519	0.54	0.30
Communication Game						
-augmentations -shared	2044	0.50	0.08	1921	0.45	0.11
-augmentations +shared	2048	*NS*	*NS*	2025	*NS*	*NS*
+augmentations -shared	2042	0.58	0.18	1752	0.53	0.32
+augmentations +shared	2046	0.56	0.15	1765	0.51	0.25

Table 2: Protocol analysis. |P| is the observed protocol size, that is, the number of distinct symbols actually used at test time. We mark as NS the cases where the obtained scores were not significantly different from chance according to a permutation test with $\alpha = 0.01$.

clustering with $k = 1000$ (ground-truth class cardinality) and $k = 2048$ (same as vocabulary size of our models).[9]. We report the significantly better results we obtained with the second choice.

Looking at Table 2, we first observe that, consistent with the Gaussian blob sanity check, there is no significant sign of symbol-category association for the *-augmentations +shared* protocol. All other models show some degree of symbol interpretability (with significantly above-chance nMI and WNsim scores). Even when there is no data augmentation during training, using different visual modules (*-augmentations -shared*) leads to some protocol interpretability, coherently with the fact that this configuration was less able than its *+shared* counterpart to communicate about noise. We were moreover surprised to find that simply argmaxing the SimCLR visual feature layer produces meaningful "symbols", which suggests that information might be more sparsely encoded by this model than one could naively assume.

Importantly, the game protocols derived with data augmentation have particularly high nMI and WNsim scores. Impressively, the scores achieved by our models, and the *+augmentations -shared* setup in particular, are very close to those obtained by clustering SimCLR visual features. Recall that, unlike our models, whose protocol independently emerges during discriminative game training, SimCLR$_{kmeans}$ runs a clustering algorithm on top of the representations produced by the SimCLR visual encoder with the express goal to discretize them into coherent sets, thus constituting a hard competitor to reach.

Even more than the quantitative results, a sense of how good our symbols are as unsupervised image labels is given by qualitative inspection images sharing the same assigned of symbol. Fig. 3 shows a random set of such images for the 9 symbols most frequently produced by the *+augmentations -shared* Sender in ILSVRC-val, without any hand-picking.[10] Some symbols denote intuitive categories, although, interestingly, ones that do not correspond to specific English words (birds on branches, dogs indoors...). Other sets are harder to characterize, but they still share a clear high-level "family resemblance" (Symbol 2: objects that glow in the dark; Symbol 3: human artifacts with simple flat shapes, etc). Frequency imbalance in input categories, together with the fact that the agents are allowed to use a large number of symbols, leads to partially overlapping categories (Symbol 9 might denote living things in the grass, whereas Symbol 4 seems to specifically refer to mammals in the grass).

The fact that symbols do not exactly denote ILSVRC categories plays to the agents' advantage when they must communicate about OOD set images. While this set, by construction, does not contain ILSVRC categories, as Figure 4 shows, it still contains birds on tree, glowing objects and artifacts with flat shapes.

[9]Cluster centroids were estimated on a random 10% of the training set. Cluster assignments of the test data were used as symbols in the protocol analysis for SimCLR$_{kmeans}$

[10]To preserve privacy, we excluded 25% of symbol-5 images before sampling, as they depicted people. Consistent with this symbol's "theme", the latter mostly show people in dark backgrounds.

Figure 3: Randomly selected ILSVRC-val images triggering the +*augmentations* -*shared* Sender to produce its 9 most frequent symbols.

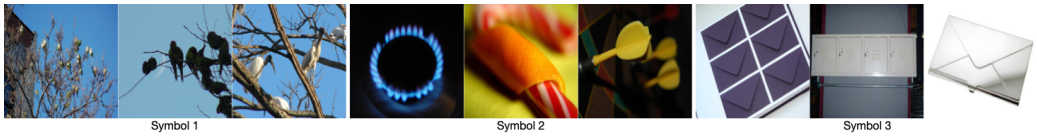

Figure 4: Randomly selected OOD set images triggering the +*augmentations* -*shared* Sender to produce the 3 most frequent ILSVRC-val symbols (cf. Fig. 3).

4.3 Downstream object classification: emergent communication as self-supervised visual feature learning

Finally, we evaluate how the features produced by the Sender CNN trained on the communication game work as out-of-the-box visual representations. We follow the standard protocol for training a linear classifier on the output of the frozen CNN trunk with various object classification datasets (see 3.2 above). We focus on Sender because the features produced by the two agent networks are always highly correlated.[11] We further exclude the *-augmentations* models, given that, having learned a degenerated strategy, they extremely poor performance on ILSVRC-val (below 5% accuracy). We thus decided not to test them on other downstream tasks, as this is a lengthy and resource-intensive procedure (see Appendix A.1).

As a reasonable upper bound, Table 3 reports the fully-supervised object classification results from [21]. As a more direct point of comparison, we also report the performance of our SimCLR implementation.[12] As the table shows, the features developed as by-product of communication game

[11] Across setups and data sets, the Sender/Receiver correlation between all pairwise visual representation similarities was never below 0.96.

[12] It is difficult to compare our SimCLR ILSVRC-val performance precisely to that reported in the original paper since, coherently with the communication game setup, we use a per-GPU batch size of 128 without sharing negatives across GPUs. By looking at the leftmost bars of Fig. 9 in [5], we note that the performance we report is within the range of their results for the same number of training epochs (100).

	ILSVRC-val	Places205	iNaturalist2018	VOC07
Supervised	76.5%	53.2%	46.7%	87.5%
SimCLR	60.6%	49.0%	31.8%	78.7%
Communication Game				
+augmentations -shared	59.0%	47.9%	30.8%	77.0%
+augmentations +shared	60.2%	49.1%	31.3%	78.8%

Table 3: Linear evaluation on object classification. Reported scores are mAP for VOC07, top-1 accuracy elsewhere. Supervised results are from [21].

playing are consistently of comparable quality to those of SimCLR, a method developed specifically for visual feature learning. This is an extremely promising first step towards employing emergent communication as a form of self supervision. Many ideas from the self-supervised literature (e.g., new data augmentation pipelines, the use of memory banks for distractor sampling or variants of the similarity-based pretext task) could straightforwardly be integrated into our setup, hopefully leading to the emergence of even better visual features and, perhaps, an even more transparent protocol.

5 Conclusion

Deep agent coordination through communication has recently attracted considerable interest. Referential games are a natural environment to test the agents' emergent communication strategies. Past approaches, however, relied on relatively small image pools processed with pretrained visual networks, or on artificial input. We showed instead that deep agents can learn to refer to a high number of categories depicted in large-scale image datasets, while communicating through a discrete channel and developing their visual processing modules from scratch. Performance on referential games with two distinct test sets (one with categories not presented at training), along with protocol analysis, shows that the agents' protocol is effective and interpretable.

A key ingredient to success was input data augmentation. We borrowed this idea from recent approaches to self-supervised visual learning. We believe that the connection between emergent communication and this field can be mutually beneficial. In this work we showed how the agents' visual networks emerging from discriminative game playing already produce high-quality visual features. On the other hand, imposing a discrete bottleneck on network communication can be seen as an augmentation operation in the latent space of an existing contrastive learning system. Further integration with self-supervised learning methods should be explored in the future.

In our experiment, agents communicate through a single symbol, but the true expressive power of human language comes from the infinite combinatorial possibilities offered by composing sequences of discrete units [45]. Allowing messages of greater length and probing whether this results in the development of a compositional code should be a priority for future work. Additionally, while in our experiments distractors are selected at random, this is obviously not the case in real-life referential settings (dogs will tend to occur near other dogs or humans, rather than between a whale and a space shuttle). Dealing with realistic category co-occurence is thus another important future direction. Finally, although we provided quantitative and qualitative evidence that the agents' protocol is reasonably transparent, the extent to which the achieved degree of interpretability is good enough for human-in-the-loop scenarios remains to be experimentally investigated.

Much recent work in the field has moved towards a theoretically-oriented understanding of deep agent communication in symbolic and artificial setups. We went back instead to the original motivation behind the study of emergent language, as a path towards the development of autonomous AIs that can interact with each other in a realistic environment. While we are still far from real-life-deployable and fully interpretable machine-machine interaction, we believe that our work represents an important early step towards this goal.

References

[1] Angeliki Lazaridou and Marco Baroni. Emergent multi-agent communication in the deep learning era. `https://arxiv.org/abs/2006.02419`, 2020.

[2] Sainbayar Sukhbaatar, Arthur Szlam, and Rob Fergus. Learning multiagent communication with backpropagation. In *Proceedings of NIPS*, pages 2244–2252, Barcelona, Spain, 2016.

[3] Nur Geffen Lan, Emmanuel Chemla, and Shane Steinert-Threlkeld. On the Spontaneous Emergence of Discrete and Compositional Signals. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4794–4800, Online, July 2020. Association for Computational Linguistics.

[4] Ning Xie, Gabrielle Ras, Marcel van Gerven, and Derek Doran. Explainable deep learning: A field guide for the uninitiated. `https://arxiv.org/abs/2004.14545`, 2020.

[5] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607. PMLR, 13–18 Jul 2020.

[6] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[7] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, Bilal Piot, koray kavukcuoglu, Remi Munos, and Michal Valko. Bootstrap your own latent - a new approach to self-supervised learning. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 21271–21284. Curran Associates, Inc., 2020.

[8] Carl Doersch, Abhinav Gupta, and Alexei A. Efros. Unsupervised visual representation learning by context prediction. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1422–1430, 2015.

[9] Angeliki Lazaridou, Alexander Peysakhovich, and Marco Baroni. Multi-agent cooperation and the emergence of (natural) language. In *Proceedings of ICLR Conference Track*, Toulon, France, 2017. Published online: `https://openreview.net/group?id=ICLR.cc/2017/conference`.

[10] Satwik Kottur, José Moura, Stefan Lee, and Dhruv Batra. Natural language does not emerge 'naturally' in multi-agent dialog. In *Proceedings of EMNLP*, pages 2962–2967, Copenhagen, Denmark, 2017.

[11] Serhii Havrylov and Ivan Titov. Emergence of language with multi-agent games: Learning to communicate with sequences of symbols. In *Proceedings of NIPS*, pages 2149–2159, Long Beach, CA, 2017.

[12] Katrina Evtimova, Andrew Drozdov, Douwe Kiela, and Kyunghyun Cho. Emergent communication in a multi-modal, multi-step referential game. In *Proceedings of ICLR Conference Track*, Vancouver, Canada, 2018. Published online: `https://openreview.net/group?id=ICLR.cc/2018/Conference`.

[13] Angeliki Lazaridou, Karl Moritz Hermann, Karl Tuyls, and Stephen Clark. Emergence of linguistic communication from referential games with symbolic and pixel input. In *Proceedings of ICLR Conference Track*, Vancouver, Canada, 2018. Published online: `https://openreview.net/group?id=ICLR.cc/2018/Conference`.

[14] Edward Choi, Angeliki Lazaridou, and Nando de Freitas. Compositional obverter communication learning from raw visual input. In *Proceedings of ICLR Conference Track*, Vancouver, Canada, 2018. Published online: `https://openreview.net/group?id=ICLR.cc/2018/Conference`.

[15] Rahma Chaabouni, Eugene Kharitonov, Emmanuel Dupoux, and Marco Baroni. Anti-efficient encoding in emergent communication. In *Proceedings of NeurIPS*, Vancouver, Canada, 2019. Published online: `https://papers.nips.cc/paper/2019`.

[16] Rahma Chaabouni, Eugene Kharitonov, Diane Bouchacourt, Emmanuel Dupoux, and Marco Baroni. Compositionality and generalization in emergent languages. In *Proceedings of ACL*, pages 4427–4442, virtual conference, 2020.

[17] Jia Deng, Wei Dong, Richard Socher, Lia-Ji Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of CVPR*, pages 248–255, Miami Beach, FL, 2009.

[18] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common objects in context. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 740–755, Cham, 2014. Springer International Publishing.

[19] Diane Bouchacourt and Marco Baroni. How agents see things: On visual representations in an emergent language game. In *Proceedings of EMNLP*, pages 981–985, Brussels, Belgium, 2018.

[20] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. *arXiv preprint arXiv:2104.14294*, 2021.

[21] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 9912–9924. Curran Associates, Inc., 2020.

[22] Debidatta Dwibedi, Yusuf Aytar, Jonathan Tompson, Pierre Sermanet, and Andrew Zisserman. With a little help from my friends: Nearest-neighbor contrastive learning of visual representations. *CoRR*, abs/2104.14548, 2021.

[23] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. *CoRR*, abs/2103.03230, 2021.

[24] Aäron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *CoRR*, abs/1807.03748, 2018.

[25] Xinlei Chen and Kaiming He. Exploring simple Siamese representation learning. *CoRR*, abs/2011.10566, 2020.

[26] Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In Yee Whye Teh and Mike Titterington, editors, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 297–304, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010. PMLR.

[27] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 448–456, Lille, France, 07–09 Jul 2015. PMLR.

[28] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with Gumbel-Softmax. In *Proceedings of ICLR Conference Track*, Toulon, France, 2017. Published online: `https://openreview.net/group?id=ICLR.cc/2017/conference`.

[29] Chris J Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. *arXiv preprint arXiv:1611.00712*, 2016.

[30] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In *Proceedings of AISTATS*, pages 315–323, Fort Lauderdale, FL, 2011.

[31] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[32] Philip Bachman, Devon Hjelm, and William Buchwalter. Learning representations by maximizing mutual information across views. In *Proceedings of NeurIPS*, Vancouver, Canada, 2019. Published online: `https://papers.nips.cc/paper/2019`.

[33] Mang Ye, Xu Zhang, Pong Yuen, and Shih-Fu Chang. Unsupervised embedding learning via invariant and spreading instance feature. In *Proceedings of CVPR*, pages 6210–6219, Long Beach, CA, 2019.

[34] Paulius Micikevicius, Sharan Narang, Jonah Alben, Gregory F. Diamos, Erich Elsen, David García, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, Ganesh Venkatesh, and Hao Wu. Mixed precision training. *CoRR*, abs/1710.03740, 2017.

[35] Yang You, Igor Gitman, and Boris Ginsburg. Scaling SGD batch size to 32k for imagenet training. *CoRR*, abs/1708.03888, 2017.

[36] Priya Goyal, Piotr Dollár, Ross B. Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch SGD: training imagenet in 1 hour. *CoRR*, abs/1706.02677, 2017.

[37] Ilya Loshchilov and Frank Hutter. SGDR: stochastic gradient descent with restarts. *CoRR*, abs/1608.03983, 2016.

[38] Eugene Kharitonov, Rahma Chaabouni, Diane Bouchacourt, and Marco Baroni. EGG: a toolkit for research on Emergence of lanGuage in Games. In *EMNLP-IJCNLP: System Demonstrations*, 2019.

[39] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[40] Christiane Fellbaum, editor. *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, MA, 1998.

[41] Junnan Li, Pan Zhou, Caiming Xiong, and Steven Hoi. Prototypical contrastive learning of unsupervised representations. In *Proceedings of ICLR*, Online conference., 2021. Proceedings at: `https://openreview.net/group?id=ICLR.cc/2021/Conference`.

[42] Bolei Zhou, Agata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. Learning deep features for scene recognition using Places database. In *Proceedings of NIPS*, Montreal, Canada, 2014. Published online: `https://papers.nips.cc/paper/2014`.

[43] 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. Vissl. `https://github.com/facebookresearch/vissl`, 2021.

[44] Steven Bird, Ewan Klein, and Edward Loper. *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.", 2009.

[45] Robert Berwick and Noam Chomsky. *Why Only Us: Language and Evolution*. MIT Press, Cambridge, MA, 2016.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See the Conclusion section.
   (c) Did you discuss any potential negative societal impacts of your work? [No] We believe that our work, by insisting on an interpretable channel for machine-machine communication, should rather have a positive impact in terms of explainable AI. We briefly mention this in the Introduction.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...

(a) Did you state the full set of assumptions of all theoretical results? [N/A]
(b) Did you include complete proofs of all theoretical results? [N/A]

3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] Code and data that are not already publicly available are included as supplemental material, and will be made publicly available upon acceptance.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] We specified this information, where we did not use standard datasets coming with pre-determined data splits, or default configuration files from public toolkits . For the Gaussian blob sanity check, the exact data we generated can be reproduced by using the seed specified in the script we include in the supplemental materials.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] As model training is resource- and time-consuming, we report variance over multiple runs for our main model, but not for all the other variants (see Appendix A.2). We also perform permutation tests to ascertain the statistical significance of the reported results in Section 4.2.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Appendix A.1.

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] We provide bibliographic references and/or URLs for the relevant resources.
   (b) Did you mention the license of the assets? [No] We provide instead links or references that will allow the interested reader to find out about the licenses of the various assets (all widely used in the field).
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] The OOD set that we include as supplementary material and that we will also release (as a list of ImageNet paths) upon publication.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A] We are using existing assets in compliance with their respective licenses, that do not require direct consent by the creators. We are releasing a list of paths to reconstruct our OOD set from ImageNet, but we leave it to the interested reader to obtain permission to use the relevant ImageNet images.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [Yes] We explicitly mention (in Footnote 11) that we removed pictures depicting human beings.

5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]