Learning to Draw: Emergent Communication through Sketching

Anonymous Author(s)
Affiliation
Address
email

Abstract

Evidence that visual communication preceded written language and provided a basis for it goes back to prehistory, in forms such as cave and rock paintings depicting traces of our distant ancestors. Emergent communication research has sought to explore how agents can learn to communicate in order to collaboratively solve tasks. Existing research has focused on language, with a learned communication channel transmitting sequences of discrete tokens between the agents. In this work, we explore a visual communication channel between agents that are allowed to draw with simple strokes. Our agents are parameterised by deep neural networks, and the drawing procedure is differentiable, allowing for end-to-end training. In the framework of a referential communication game, we demonstrate that agents can not only successfully learn to communicate by drawing, but with appropriate inductive biases, can do so in a fashion that humans can interpret. We hope to encourage future research to consider visual communication as a more flexible and directly interpretable alternative of training collaborative agents.

1 Introduction

Imagine you and a friend are playing a game where you have to get your friend to guess an object in the room by you sketching the object. No other communication is allowed beyond the sketched image. This is an example of a *referential communication game*. To play this game you need to have learned how to draw in a way that your friend can understand. This paper explores how artificial agents parameterised by neural networks can learn to play similar drawing games.

Spurred by innovations in artificial neural networks, deep and reinforcement learning techniques, recent work in multi-agent emergent communication [3, 12, 14, 22, 29] pursues interactions in the form of gameplay between agents to induce human-like communication. Artificial communicating agents can collaborate to solve various tasks: image referential games with realistic visual input [14, 21, 22], negotiation [1], navigation of virtual environments [6, 15] or reconstruction of missing input [3, 12]. The key to achieving the shared goal in many of these games is collaboration, and implicitly, communication. To date, studies on communication emergence in multi-agent games have focused on exploring a language-based communication channel, with messages represented by discrete tokens or sequences of tokens [5, 12, 14, 18, 21, 22, 27]. However, these communication protocols can be difficult to interpret for a human observer [2, 19, 25]. In this work, we propose a more direct and potentially self-explainable means of transmitting knowledge: *sketching*.

Gelb [11] discusses the origins of writing, which is nowadays a common means of communication although this has not always been the case. Evidence suggests pre- and early-humans were able to communicate by drawing long before developing the various stages of written language. Drawings such as petrograms and petroglyphs exist from the oldest palaeolithic times and may have been used to record past experiences, events, beliefs or simply the relation with other beings. These

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

pictorial characters which are merely impressions of real objects or beings stand at the basis of all writing [11]. This leads us to question if drawing is a more natural way of *starting* to study emergent communication and if it could lead to better written communication later on.

Concretely, we propose a visual communication channel in the context of image-based referential games. We leverage recent advances in differentiable sketching that enables us to construct an agent that *can learn to communicate intent through drawing*. Through a range of experiments we show that:

- Agents can successfully communicate about real-world images through a sketching game. However, training with a loss that tries to maximise gameplay alone does not lead to human decipherable sketches, irrespective of any visual system preconditioning;
- Introducing a perceptual loss improves human interpretability of the communication protocol, at little to no cost in the gameplay success;
- Changes to the game objective, such as playing an object-oriented game, can steer the emergent communication protocol towards a more pictographic or symbolic form of expression;
- Inducing a shape-bias into the agents' visual system leads to more explainable drawings.

2 Communication between agents

Communication emerges when two or more participants are involved, share a goal, task or incentive which can be achieved only by transfer of information and so is beneficial for all parties involved. Studies on language origins [28, 32] consider cooperation to be a key prerequisite to language evolution as it implies multiple agents having to self-organise and adapt to the same convention. Studies on the emergence of communication in cooperative multi-agent environments from recent years have focused on (natural) language learning [21, 22] and its inherent properties such as compositionality and expressivity [12, 13, 29].

A number of works specifically relate to the overarching ideas of gameplay and learning in this paper. For example, Foerster et al. [9] proposed a framework for differentiable training of communicating agents which was later used by Jorge et al. [16] to solve image search tasks with two interacting agents communicating with atomic symbols. Lazaridou et al. [21] proposed an image-based referential game in which the agents again communicated using atomic symbols, and were trained using policy gradients. Havrylov and Titov [14] and Mordatch and Abbeel [27] both demonstrated that it was possible to use differentiable relaxations to train agents that communicated with sequences of symbols. In the former case, the agents played the referential game that we adopt for our experiments.

One of the long-term goals of this research in language emergence is to develop interactive machines that can productively communicate with humans. As such we should ensure that whatever language artificial agents develop, it is one that human agents can understand. In our work, we take inspiration from the process and evolution of writing. Written language has undergone many transitions from early times to reach the forms we now know: from pictures and drawings to word-syllabic, syllabic and, finally, alphabetic systems. In the beginning, our early ancestors did not know how to communicate in writing. Instead, they began drawing and painting pictures of their life, representing people and things they knew about [11]. Studies on the communication systems developed in primitive societies compare ancient drawings to the very early sketches drawn by children and talk about their tendency of concretely identifying certain things or events in their surrounding world [11, 17]. Psychological and behavioural studies have shown that children try to communicate to the world through the images they create even when they cannot associate them with words [7].

3 A model for learning to communicate by drawing

We present a model consisting of two agents, the sender and the receiver in which the sender learns to draw by playing a game with the receiver. The overall architecture of the agents in the context of the game they are learning to play is shown in Figure 1.

3.1 The Game Environment

Our experimental setup builds upon the image referential game previously explored in studies of emergent communication [14, 21, 22] that derives from Lewis's signalling game [23]. We

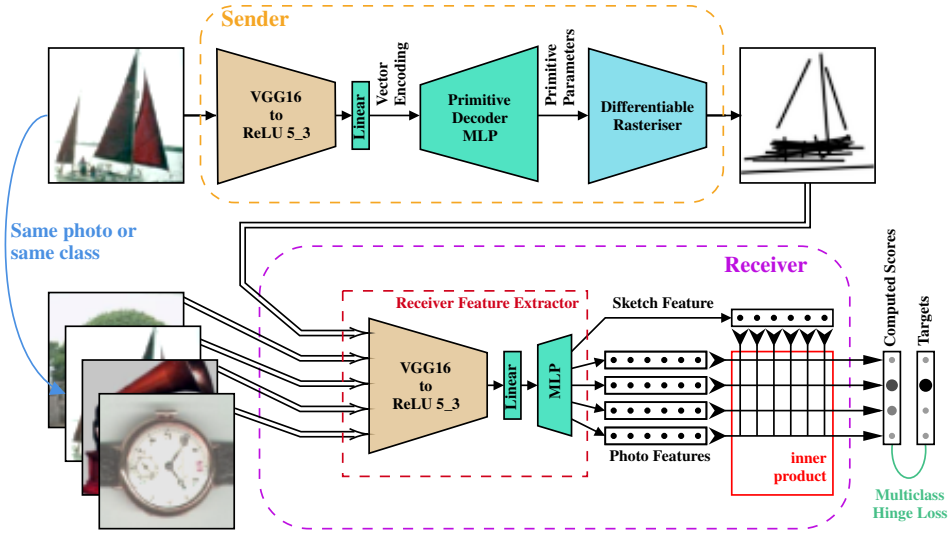


Figure 1: **Overview of the agent architecture and game setup.** A 'sender' agent is presented with an image and sketches its content through a learnable drawing procedure. The 'receiver' agent is presented with the sketch and a collection of photographs, and has to learn to correctly associate the sketch with the corresponding photograph by predicting scores which are compared to a one-hot target. Both agents are parameterised by neural networks trained end-to-end using gradient methods.

implemented several variants of Havrylov and Titov [14]'s image guessing game. The overall setting of these games is formulated as follows:

1. Two target photographs, $\mathbf{P}_s$ and $\mathbf{P}_r$, and set of K distractor photographs, $\{\mathbf{P}_d^{(k)}\}_{k=1}^K$, are selected.
2. There are two agents: a sender and a receiver.
3. After being presented the $\mathbf{P}_s$ target image, the sender has to formulate a message conveying information about that image.
4. Given the message and the set of photographs, $\{\mathbf{P}_d^{(k)}\}_{k=1}^K \cup \{\mathbf{P}_r\}$, consisting of all the distractors and the target $\mathbf{P}_r$, the receiver has to identify the target correctly.

The specifics of how the photographs are selected (step 1 above) depends on the game variant as described below. Success in these games is measured by the binary ability of the receiver to correctly guess the correct image or not; as such, the measure of *communication rate* is used to assess averaged performance over many games using independent images to those used during training. Unlike Havrylov and Titov [14]'s game in which the sender helps the receiver identify the correct image by sending a message constructed as a sequence of tokens drawn from a predefined vocabulary, we propose using a directly interpretable means of communication: *sketching the target photograph*.

Original game variant. In Havrylov and Titov [14]'s variant of the game there is a pool of photos from which the distractors and target $\mathbf{P}_s$ are drawn randomly without replacement. The target $\mathbf{P}_r$ is set to be equal to $\mathbf{P}_s$. In our *original* variant experiments the number of distractors, K, is set to 99.

Object-oriented game variants. In addition to the original setup, we explored two slightly different and potentially harder game configurations which were intended to induce the agents to draw sketches that would be more representative to the object class they belong to rather than to the specific instance of the class. These setups use labelled datasets where each image belongs to a class based on its contents. In the first of these variants (we refer to this as *OO-game same*), the target $\mathbf{P}_r$ is set to be equal to $\mathbf{P}_s$, and the distractors and target are sampled such that their class labels are disjoint (that is every photo provided to the receiver has a different class). The second setup (*OO-game different*) is similar to the first, but the target $\mathbf{P}_r$ is chosen to be a different photograph with the same class label as target $\mathbf{P}_s$. The intention behind these games is to explore a universally interpretable depiction of the different object classes, which does not focus on individual details but rather conveys the concept.

3.2 Agents' Architectures

Both agents act on visual inputs. The agents are parameterised by deep neural networks and are trained using standard gradient techniques (Section 3.3).

The agent's early visual system. We choose to model the early visual systems of both agents with the head part of the VGG16 CNN architecture [31] through to the ReLU activation at the end of the last convolutional layer (commonly referred to as the `ReLU5_3` layer) before the final max-pooling and fully connected layers. In all experiments, we utilise pretrained weights and freeze this part of the model during training. We justify this choice on the basis that it provides the agents with an initial grounding in understanding the statistics of the visual world, and ensures that the visual system cannot collapse and remains universal. The weights are the standard `torchvision` ImageNet weights, except in the cases where we explore the effect of shape bias (see Section 4.5). As these pretrained weights were learned with images that were normalised according to the ImageNet statistics, all inputs to the VGG16 backbone (including sketches) are normalised accordingly. The output feature maps of this convolutional backbone are flattened and are linearly projected to a fixed dimensional vector encoding (64-dimensions unless otherwise specified). Because the datasets used in gameplay have different resolutions, the number of weights in the learned projection varies.

3.2.1 Sender Agent

The goal of the sender is to produce a sketch from the input photograph. For experiments in Section 4, we restrict the production of sketches to be a drawing composed of 20 black, constant width, straight lines on a white canvas of the same size as the input images. Experiments with fewer lines can be found in Appendix A. It is of course possible to have a much more flexible definition of a sketch and incorporate many different modelling assumptions. We choose to leave such exploration for future work and focus on the key question of whether we can actually achieve successful (and potentially interpretable) communication with our simplified but not unrealistic setup.

Given an input image, the agent's early visual system produces a vector encoding which is then processed by a three-layer multilayer perceptron (MLP) that learns to decode the primitive parameters used to draw the sketch. This MLP has ReLU activations on the first two layers and $\tanh$ activation on the final layer. Unless otherwise specified, the first two layers have 64 and 256 neurons respectively. The output layer produces four values for each line that will be drawn; the values are the start and end coordinates of each line stroke in an image canvas with the origin at the centre and edges at ± 1.

To produce a sketch image from the line parameters output by the MLP, we utilise the differentiable rasterisation approach introduced by Mihai and Hare [26]. At a high level, this approach works by computing the distance transform on a pixel grid for each primitive being rendered. A relaxed approximation of a rasterisation function is applied to the distance transform to compute a raster image of the specific primitive. Finally, a differentiable composition function is applied to compose the individual rasters into a single image. More specifically, the squared Euclidean Distance Transform is computed, $\mathbf{D}^2_{\mathrm{seg}}(\boldsymbol{s}, \boldsymbol{e})$ over all pixels in the image, for each line segment starting at coordinate $\boldsymbol{s}$ and ending at $\boldsymbol{e}$. These squared distance transforms are simply images in which the value of each pixel is replaced with the closest squared distance to the line (computed when the pixels are mapped to the same coordinate system as the line — so the top left of the image is $(-1, -1)$ and bottom-right is $(1, 1)$). Using the subscript i to refer to the i-th line in the sketch, each $\mathbf{D}^2_{\mathrm{seg}}(\boldsymbol{s_i}, \boldsymbol{e_i})$ is rasterised as

$$\mathbf{R}_i = \exp\left(-\frac{\mathbf{D}^2_{\mathrm{seg}}(\boldsymbol{s_i}, \boldsymbol{e_i}))}{\sigma^2}\right) , \tag{1}$$

where σ^2 is a hyperparameter that controls how far gradients flow in the image, as well as the visible thickness of the line ($\sigma^2 = 5 \times 10^{-4}$ for all experiments in this paper). We adopt the soft-or composition function [26] to compose the individual line rasters into a single image, but incorporate an inversion so that a sketch image, $\mathbf{S}$, with a white canvas (pixels with value 1) and black lines (pixels 0 valued) is produced,

$$\mathbf{S} = \prod_{i=1}^{n} (\mathbf{1} - \mathbf{R}_i) , \tag{2}$$

where n is the number of lines. Finally, because the backbone CNNs work with three-band colour images, we replicate the greyscale sketch image three times across the channel dimension.

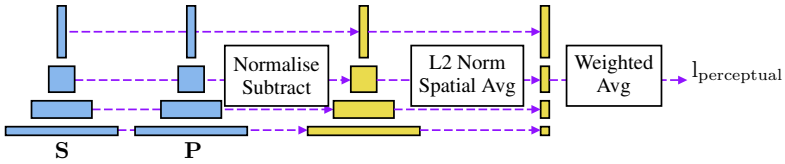


Figure 2: **Computing a 'perceptual' loss with the early visual system.** Features are extracted from the sketch **S** and corresponding photograph **P** from different layers of the backbone. The features are normalised over channels and subtracted. We take the sum of the squared differences over channels and average spatially. Finally, we compute a weighted average across layers.

3.2.2 Receiver Agent

The receiver agent is given a set of photographs and a sketch image, and is responsible for predicting which photograph matches the sketch under the rules of the specific game being played. The receiver's visual system is coupled with a two-layer MLP with a ReLU nonlinearity on the first layer (the latter layer has no activation function). Unless otherwise specified, all experiments use 64 neurons in the first layer and 64 in the final layer. The sketch image and each photograph are passed through the visual system and MLP independently to produce a feature vector representation of the respective input. A score vector $\boldsymbol{x}$ is produced for the photographs by computing the scalar product of the sketch feature with the feature of each respective photograph. This score vector is un-normalised but could be viewed as a probability distribution by passing it through a softmax. The photograph with the highest score is the one predicted.

3.3 Training details

By incorporating a loss between the predicted scores of the receiver agent and the known correct target photograph, it is possible to propagate gradients back through both the receiver and sender agents. As such, we can train the agents to play the different game settings. For the loss function, we follow Havrylov and Titov [14] and choose to use Weston and Watkins [34]'s multi-class generalisation of hinge loss (aka multi margin loss),

$$\mathrm{l}_{\mathrm{game}}(\boldsymbol{x}, y) = \sum_{j \neq y} \max(0, 1 - \boldsymbol{x}_y + \boldsymbol{x}_j) \ , \tag{3}$$

where $\boldsymbol{x}$ is the score vector produced by the receiver, and y is the true index of the target, and the subscripts indicate indexing into the vector. The rationale for this choice is that the (soft) margin constraint should help force the distractor photographs' features to be more dissimilar to the sketch feature. Tests using cross-entropy also indicated that it could work well as an alternative, however.

Optimisation of the parameters of both agents is performed using the Adam optimiser with an initial learning rate of 1×10^{-4} for all experiments. For efficiency, we train the model with batches of games where the sender is given multiple images which are converted to sketches and passed to the receiver which reuses the same set of photographs for each sketch in the batch (with each sketch targeting a different receiver photograph). The order of the targets with respect to the input image's sketches is shuffled every batch. Batch size is $K + 1$, where K is the number of distractors, for all experiments. Unless otherwise stated, training was performed for 250 epochs. A mixture of Nvidia GTX1080s, RTX2080s and an RTX-Titan was used for training the models with higher resolution images which required more memory. Training time varied from around 488 games/second (10 secs/epoch) for games using STL10 to around 175 games/second (around 5 mins/epoch) for Caltech-101 experiments with 128px images.

3.4 Making the sender agent's sketches more perceptually relevant

Perception of drawings has a long history of study in neuroscience [see e.g. 30, for an overview]. In order to induce the sender to produce sketches that are more interpretable, we explore the idea of using an additional loss function between the differences in feature maps of the backbone CNN from the produced sketch and the input image. Such a loss has a direct grounding in biology, where it has been observed through human brain imaging studies that sketches and photographs of the same scene result in similar activations of neuron populations in area V4 of the visual cortex, as

well as other areas related to higher-order visual cognition [33]. At the same time, it has also been demonstrated that differences in feature maps from pre-trained CNN architectures can be good proxies for approximating human notions of perceptual similarity between pairs of images [36].

Inspired by Zhang et al. [36] we formulate a loss based on the normalised differences between feature maps of the backbone network from the application of the network to both the input photograph and the corresponding sketch. Unlike Zhang et al. we choose not to learn weightings for each feature map channel individually, but rather we consider all feature maps produced by a layer of the backbone to be weighted equally. Learning individual channel weighting would be an interesting direction for future research, but is challenging because we would want to avoid the network learning zero weights for each channel, where the perceptual loss is basically ignored.

Figure 2 illustrates our perceptual loss formulation; note that unlike Zhang et al. [36] the final averaging operation does incorporate a (per-layer) weighting, $\boldsymbol{w}_l$, which we explore the effect of in Section 4.2. More formally, denoting the sketch as $\mathbf{S}$ and corresponding photo as $\mathbf{P}$, we extract $L = 5$ feature maps, $\hat{\mathbf{S}}^{(l)}, \hat{\mathbf{P}}^{(l)} \in \mathbb{R}^{H_l \times W_l \times C_l}$, for the l-th layer from the backbone VGG16 network and unit normalise each across the channel dimension. The loss is thus defined as,

$$\mathrm{l}_{\text{perceptual}}(\mathbf{S}, \mathbf{P}, \boldsymbol{w}) = \sum_l \frac{\boldsymbol{w}_l}{H_l W_l} \sum_{h,w} \left\| \hat{\mathbf{S}}^{(l)}_{hw} - \hat{\mathbf{P}}^{(l)}_{hw} \right\|_2^2 . \tag{4}$$

To extract the feature maps we choose to use the outputs of the VGG16 layers immediately before the max-pooling layers (`relu1_2`, `relu2_2`, `relu3_3`, `relu4_3` and `relu5_3`). During training, this perceptual loss is added to the game loss (l_{game}).

4 Experiments

We next present a series of experiments where we attempt to answer if it is possible that our agents learn to successfully communicate, and what factors affect human interpretation of the drawings.

4.1 Can agents communicate by learning to draw?

We explore the game setups described in Section 3.1 and train our agents to play the games using 96×96 photographs from the STL-10 dataset [4]. For the *original* game we use 99 distractors. For the object-oriented games, due to the dataset only having 10 classes, we are limited to 9 distractors.

In Table 1, we show quantitative and qualitative results of the visual communication game played under the three different configurations. The results demonstrate that it is possible for agents to successfully play this type of image referential game by learning to draw. One can observe that although agents achieve a high communication success rate, using only the l_{game} loss leads to the emergence of a communication protocol that is indecipherable to a human. However, the addition of the perceptual loss, motivated in Section 3.4, significantly improves the interpretability of the communication channel at almost no cost in the actual communication success rate.

One interesting observation is that although the sketches for some of the classes have greatly improved when incorporating the perceptual loss, for photographs of animals or birds, the sketches are not particularly representative of the class instance or distinguishable for the human eye. In the following sections we explore the model to try to better understand what factors affect the drawings that are produced.

4.2 What effect does weighting the perceptual loss have on the sketches?

Next, we explore the effect of manually weighting the perceptual loss. More precisely, we look at what happens when the perceptual loss is applied to the features maps from just one layer of the backbone network. As previously mentioned in Section 3.2, the feature maps are extracted using a VGG16 CNN up to `ReLU5_3` layer. For example, we can discard all feature maps except those from the first layer by weighting the perceptual loss by $[1, 0, 0, 0, 0]$. The effect of the different weights, which allow only one block of feature maps to be used for drawing the sketch, is illustrated in Table 2. We apply these constraints in two setups, the original and the *OO-game different*. In both cases, the drawings are unrecognisable if the perceptual loss takes into account only the first or the second block of feature maps. Blocks 3 through 5 seem to provide increasing structure under both game setups. It is

Table 1: **Communication success rate and example sketches produced by the agents in order to achieve the game objective in various setups and with different losses.** Sample input images seen by the sender (the left column) are described as the sketches in the second and third column. Although successful communication seems to be achieved in all setups, the addition of the perceptual loss significantly improves human interpretability of the drawings. Examples are from STL-10.

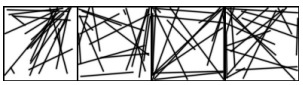
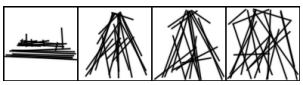
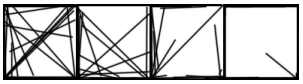
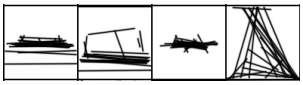
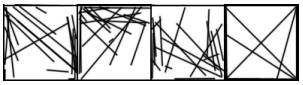
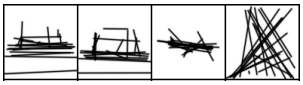
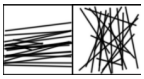
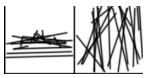
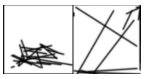
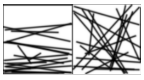
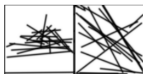
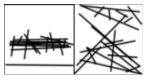
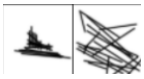
	l_{game}	$l_{game} + l_{perceptual}$
Original game	69.0 %	66.0 %
		
OO-game same	96.3%	93.0%
		
OO-game different	87.0%	86.3%
		

Table 2: **The effect of weighting the perceptual loss such that only the feature maps from one backbone layer are used.** The features extracted in the last three layers of the visual system seem to capture information that leads to sketches which resemble to an extent the corresponding photograph.

Loss weights	$[1, 0, 0, 0, 0]$	$[0, 1, 0, 0, 0]$	$[0, 0, 1, 0, 0]$	$[0, 0, 0, 1, 0]$	$[0, 0, 0, 0, 1]$
Orig. game	54.0%	64.0%	77.0%	74.0%	67.0%
					
OO-game diff	85.0%	87.5%	86.3%	88.7%	90.5%
					

worth noticing that, similar to the results shown in Section 4.1, the communication success rate in the original setup is always lower than that from the *OO-game different* setup. Overall, the information provided by individual layers in the visual extractor network is enough for the agents to develop a visual communication strategy that can be used to play the game. For humans, however, the later layers contribute the most to the emergence of a communication protocol that we can understand.

4.3 Does the *OO-game* influence the sketches to be more recognisable as the type of object?

Comparing the qualitative results of different game formats from Table 1, we notice that agents develop distinct strategies for representing the target photograph under different conditions. If there is more variability in the sketches that correspond to photographs from the same class in the original game setup, and a bit less in the *OO-game same*, the sketches become almost like symbols representing all the photographs from one class when playing *OO-game different*. In other words, the object-oriented games influence the sketches to be more recognisable as the type of object, than the specific instance of the class. For a better illustration of this, please see Appendix B.

4.4 How does the model's capacity influence the visual communication channel?

Regarding the model's architecture, we look into how drawings are influenced by the width of the model. In this experiment (results shown in Table 3), we compare the baseline model architecture detailed in Section 3.2 with a wider variant that has the following changes: the sender encodes the target photograph to a 1024-dimensional vector (baseline model encodes to 64-dimensional vector),

Table 3: **The effect of the model's capacity on its sketches.** The wide model's sender encodes the photo into a 1024-dimensional vector (baseline 64), and the receiver's MLP linear layers have 1024 neurons each versus 64. Examples from training on Caltech-101 in the *OO-game different* setting.

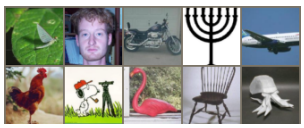
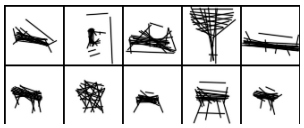
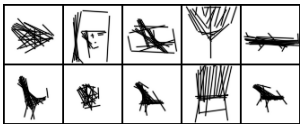
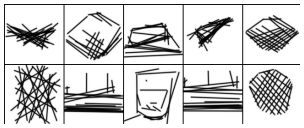
	Baseline	Wide
	39.0%	47.0%
		

Table 4: **The effect on the communication protocol of using a VGG16 feature extractor network pretrained on datasets that have texture (ImageNet) or shape (Stylized-ImageNet [10]) bias.** Examples are from agents trained using the *original* game with Caltech-101 data. The shape-biased sketches are better at capturing the overall object form, particularly for things like faces.

	ImageNet weights	Stylized-ImageNet weights
	78.6 %	80%
		

the receiver's MLP capacity is also increased from 64 to 1024 in both layers. We present results for the *OO-game different* setup played with 128×128 Caltech-101 images [8]. The increased number of classes in Caltech-101 may explain the drop in the communication rate, which compared to the same model played under the original game setup (see the ImageNet-pretrained model in Table 4), has almost halved. As one might expect, the wider model allows for more details to be captured, and, hence, conveyed in the sketches. Unlike the baseline model which, in this object-oriented setup, develops a communication system that is more representative to the class than to the instance (as discussed in Section 4.3), the wider model starts to draw distinctive representations for objects of the same type. More sketches can be found in Appendix C where one can observe the difference between all images with chairs, for example.

4.5 How does the texture/shape bias of the visual system alter communication?

Next, we show that a texture or shape bias of the visual system influences visual communication. This experiment was run under the original game setup with 128×128 Caltech-101 images [8]. The results shown in Table 4 suggest that inducing a "shape bias" into the model does not significantly improve the agent's performance in playing the game, but produces more meaningful drawings. By using the VGG16 weights pretrained on Stylized-ImageNet [10], the communication protocol also becomes more faithful to the actual shape of the objects. A shape-based sketch is much more interpretable to humans, as it has been known for a long time that shape is the most important cue for human object recognition [20]. Further results from this experiment can be found in Appendix D.

4.6 Do the models learn to pick out salient features?

From the results we have presented so far, it is evident that, particularly with the perceptual loss, the sender agent is able to broadly draw pictures of particular classes of object. The high communication rates in the *original* game setting would also suggest that the drawings can capture something *specific* about the target images that allow them to be identified amongst the distractors. To further analyse what is being captured by the models we train the agents in the original game setting (using both normal and stylized backbone weights) with images from the CelebA dataset [24], which we take the maximal square centre-crop and resize to 112px. As this dataset contains only images of faces, messages between the agents will have to capture much more subtle information to distinguish the target from the distractors. Results are shown in Figure 3; the communication rate is near perfect

Figure 3: **Sketches from *original* variant games using the CelebA dataset with perceptual loss and different biases from backbone weights.** Both the texture-biased (ImageNet) and shape-biased (Stylized-ImageNet) settings exhibit near-perfect communication success, but the shape-biased sketches are considerably more interpretable and show visual variations correlated with the photos.

for both models, but the difference between the texture-biased and shape-biased models is striking. There is subtle variation in the texture biased model's sketches which broadly seems to capture the head pose, but the overall structure of the sketches is similar. In the shape-biased model head pose is evident from the sketches, but so are other salient features like hairstyle and (see Appendix E) head-wear and glasses.

5 Conclusions and Future Work

We have demonstrated that it is possible to develop and study an emergent communication system between agents where the communication channel is visual. Further, we have shown that a simple addition to the loss function (that is motivated by biological observations) can be used to produce messages between the agents that are directly interpretable by humans.

The immediate next steps in this line of work are quite clear. It is evident from our experiments that the incorporation of the perceptual loss dramatically helps produce more interpretable images. One big question to explore in the future is to what extent this is influenced by the original training biases of the backbone network — are these drawings produced as a result of the original labels of the ImageNet training data, or are they in some way more generic than that? We plan to address this by exploring what happens if the weights of the backbone are replaced with ones learned through a self-supervised learning approach like Barlow twins [35]. We would also like to explore what happens if the agents' visual systems had independent weights.

Going further, as previously mentioned, learning a perceptual loss would be a good direction to explore, but perhaps this should also be coupled with a top-down attention mechanism based on the latent representation of the input. An open question from doing this would be to ask if this allows for a richer variation in drawing, and for features to be exaggerated as in the case of a caricature. Such an extension could also be coupled with a much richer approach to drawing, with variable numbers of strokes, which are not necessarily constrained to being straight lines. Coupling feedback or attention into the drawing mechanism itself could also prove to be a worthy endeavour.

We hope that this work lays the groundwork for more study in this space. Fundamentally our desire is that it provides the foundations for exploring how different types of drawing and communication — from primitive drawings through to pictograms, to ideograms and ultimately to writing — emerges between artificial agents under differing environmental and internal constraints and pressures. Unlike other work that 'generates' images, we explicitly focus on learning to capture *intent* in our drawings. We recognise however that our work may have broader implications beyond just understanding how communication evolves. Could for example in the future we see a sketching agent replace a trained illustrator? The creation of messages for communication inherently involves elements of individual creative expression and adaption to the emotive environment of both the sender and receiver of the message. Our current models are clearly incapable of this, but such innovations will happen in the future. When they do we need to be prepared for the surrounding ethical debate and discussions about what constitutes 'art'.

References

[1] Kris Cao, Angeliki Lazaridou, Marc Lanctot, Joel Z Leibo, Karl Tuyls, and Stephen Clark. Emergent communication through negotiation. In *International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=Hk6WhagRW`.

[2] Rahma Chaabouni, Eugene Kharitonov, Emmanuel Dupoux, and Marco Baroni. Anti-efficient encoding in emergent communication. *CoRR*, abs/1905.12561, 2019. URL `http://arxiv.org/abs/1905.12561`.

[3] Rahma Chaabouni, Eugene Kharitonov, Diane Bouchacourt, Emmanuel Dupoux, and Marco Baroni. Compositionality and generalization in emergent languages. *arXiv preprint arXiv:2004.09124*, 2020.

[4] Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 215–223. JMLR Workshop and Conference Proceedings, 2011.

[5] Abhishek Das, Satwik Kottur, José MF Moura, Stefan Lee, and Dhruv Batra. Learning cooperative visual dialog agents with deep reinforcement learning. In *Proceedings of the IEEE international conference on computer vision*, pages 2951–2960, 2017.

[6] Abhishek Das, Théophile Gervet, Joshua Romoff, Dhruv Batra, Devi Parikh, Mike Rabbat, and Joelle Pineau. Tarmac: Targeted multi-agent communication. In *International Conference on Machine Learning*, pages 1538–1546. PMLR, 2019.

[7] Masoumeh Farokhi and Masoud Hashemi. The analysis of children's drawings: social, emotional, physical, and psychological aspects. *Procedia-Social and Behavioral Sciences*, 30: 2219–2224, 2011.

[8] Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *2004 conference on computer vision and pattern recognition workshop*, pages 178–178. IEEE, 2004.

[9] Jakob N. Foerster, Yannis M. Assael, Nando de Freitas, and Shimon Whiteson. Learning to communicate with deep multi-agent reinforcement learning. *CoRR*, abs/1605.06676, 2016. URL `http://arxiv.org/abs/1605.06676`.

[10] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. *arXiv preprint arXiv:1811.12231*, 2018.

[11] Ignace J Gelb. *A study of writing*. University of Chicago Press, 1963.

[12] Shangmin Guo. Emergence of numeric concepts in multi-agent autonomous communication. *arXiv preprint arXiv:1911.01098*, 2019.

[13] Shangmin Guo, Yi Ren, Agnieszka Słowik, and Kory Mathewson. Inductive bias and language expressivity in emergent communication. *arXiv preprint arXiv:2012.02875*, 2020.

[14] Serhii Havrylov and Ivan Titov. Emergence of language with multi-agent games: Learning to communicate with sequences of symbols. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 2149–2159. Curran Associates, Inc., 2017.

[15] Natasha Jaques, Angeliki Lazaridou, Edward Hughes, Caglar Gulcehre, Pedro Ortega, DJ Strouse, Joel Z Leibo, and Nando De Freitas. Social influence as intrinsic motivation for multi-agent deep reinforcement learning. In *International Conference on Machine Learning*, pages 3040–3049. PMLR, 2019.

[16] Emilio Jorge, Mikael Kågebäck, and Emil Gustavsson. Learning to play guess who? and inventing a grounded language as a consequence. *CoRR*, abs/1611.03218, 2016. URL `http://arxiv.org/abs/1611.03218`.

[17] Rhoda Kellogg. *Analyzing children's art*. McGraw-Hill Humanities, Social Sciences & World Languages, 1969.

[18] Eugene Kharitonov, Rahma Chaabouni, Diane Bouchacourt, and Marco Baroni. Entropy minimization in emergent languages. In *International Conference on Machine Learning*, pages 5220–5230. PMLR, 2020.

[19] Satwik Kottur, José M. F. Moura, Stefan Lee, and Dhruv Batra. Kotturmlb17. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 2962–2967, 2017.

[20] Barbara Landau, Linda B Smith, and Susan S Jones. The importance of shape in early lexical learning. *Cognitive development*, 3(3):299–321, 1988.

[21] Angeliki Lazaridou, Alexander Peysakhovich, and Marco Baroni. Multi-agent cooperation and the emergence of (natural) language. In *International Conference on Learning Representations*, 2017.

[22] Angeliki Lazaridou, Karl Moritz Hermann, Karl Tuyls, and Stephen Clark. Emergence of linguistic communication from referential games with symbolic and pixel input. In *International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=HJGv1Z-AW`.

[23] David K. Lewis. *Convention: A Philosophical Study*. Wiley-Blackwell, 1969.

[24] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

[25] Ryan Lowe, Jakob Foerster, Y-Lan Boureau, Joelle Pineau, and Yann Dauphin. On the pitfalls of measuring emergent communication. *arXiv preprint arXiv:1903.05168*, 2019.

[26] Daniela Mihai and Jonathon S. Hare. Differentiable drawing and sketching. *CoRR*, abs/2103.16194, 2021. URL `https://arxiv.org/abs/2103.16194`.

[27] Igor Mordatch and Pieter Abbeel. Emergence of grounded compositional language in multi-agent populations. *CoRR*, abs/1703.04908, 2017. URL `http://arxiv.org/abs/1703.04908`.

[28] Martin A Nowak and David C Krakauer. The evolution of language. *Proceedings of the National Academy of Sciences*, 96(14):8028–8033, 1999.

[29] Yi Ren, Shangmin Guo, Matthieu Labeau, Shay B Cohen, and Simon Kirby. Compositional languages emerge in a neural iterated learning model. *arXiv preprint arXiv:2002.01365*, 2020.

[30] Bilge Sayim and Patrick Cavanagh. What line drawings reveal about the visual brain. *Frontiers in Human Neuroscience*, 5:118, 2011. ISSN 1662-5161. doi: 10.3389/fnhum.2011.00118. URL `https://www.frontiersin.org/article/10.3389/fnhum.2011.00118`.

[31] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.

[32] Luc Steels. The synthetic modeling of language origins. *Evolution of communication*, 1(1): 1–34, 1997.

[33] Dirk B. Walther, Barry Chai, Eamon Caddigan, Diane M. Beck, and Li Fei-Fei. Simple line drawings suffice for functional mri decoding of natural scene categories. *Proceedings of the National Academy of Sciences*, 108(23):9661–9666, 2011. ISSN 0027-8424. doi: 10.1073/pnas.1015666108. URL `https://www.pnas.org/content/108/23/9661`.

[34] Jason Weston and Christopher Watkins. Support vector machines for multi-class pattern recognition. pages 219–224, 01 1999.

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

[36] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018.

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] We believe this to be true.
   (b) Did you describe the limitations of your work? [Yes] We covered existing limitations and provided ideas to overcome them such as in Section 3.2.1 or Section 5.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] Please see Section 5.
   (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] We have provided code in the supplementary material upload and will provide the public github URL in the paper if it is accepted.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] Please refer to Section 3.3.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No] The results we focus on are qualitative (sketch interpretability), and for the quantitative results (game-play success), the agents always converge to the reported scores.
   (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] Please see Section 3.3.
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] See Section 3
   (b) Did you mention the license of the assets? [Yes] Where applicable; for example our code (see question 3a) contains the 3-clause BSD license as part of the bundle.
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] See answer to question 3a, above, on the code included in the supplementary material
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
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]