PatchGame: Learning to Signal Mid-level Patches in Referential Games

Anonymous Author(s)
Affiliation
Address
email

Abstract

We study a referential game (a type of signaling game) where two agents communicate with each other via a discrete bottleneck to achieve a common goal. In our referential game, the goal of the speaker is to compose a message or a symbolic representation of "important" image patches, while the task for the listener is to match the speaker's message to a different view of the same image. We show that it is indeed possible for the two agents to develop a communication protocol without explicit supervision. We further investigate the developed protocol and show the applications in speeding up recent Vision Transformers by using only important patches, and as pre-training for downstream recognition tasks (e.g., classification).

1 Introduction

The ability to communicate using language is a signature characteristic of intelligence [55]. Language provides a structured platform for agents to not only collaborate with each other and accomplish certain goals, but also to represent and store information in a compressible manner. Most importantly, language allows us to build infinitely many new concepts by the composition of the known concepts. These qualities are shared by both the natural languages used in human-human communication and programming languages used in human-machine communication. The study of the evolution of language can hence give us insights into intelligent machines that can communicate [49].

Our goal in this paper is to develop and understand an emergent language, *i.e.*, a language that emerges when two neural network agents try to communicate with each other. Clark [14] argued that supervised approaches that consist of a single agent learning statistical relationships among symbols don't capture the functional relationships between the symbols *i.e.*, the use of symbols leading to an action or an outcome. Krishna et al. [40] argued the same viewpoint in the context of images. We, therefore, resort to the recent works in emergent language [5, 27, 39, 42, 43, 69] which show that a communication protocol can be developed or learned by two or more cooperative agents trying to solve a task. The choice of the task is quintessential since the language derives meaning from its use [73]. We choose a task where two agents, a speaker, and a listener, play a *referential game*, a type of signaling game first proposed by Lewis [45]. The speaker agent receives a target image and sends a message to the listener. The message consists of discrete symbols or words, capturing different parts of the image. The listener receives another view of the target image, and one or more distractor images. The goal of the speaker and the listener agents is to maximize the agreement between the message and the target image. Fig. 1 illustrates the overview of the proposed referential game.

In computer vision, a large number of attempts have been made to represent images as visual words [35], with a focus on low-level feature descriptors such as SIFT [46], SURF [6], *etc.*. More recent works in deep learning have attempted to describe the entire image with a fixed number of discrete symbols [53, 54, 58], however, we postulate that large images contain a lot of redundant information and a good visual representation should focus on only the "interesting" parts of the

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

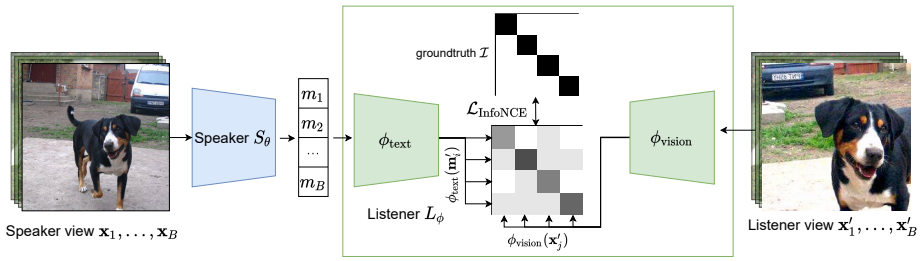


Figure 1: Overview of the Referential Game. We generate two random views of every image in the given batch of images. The speaker takes one of the views as the input and generates a sequence of symbols (or message). The listener takes the message sent by the speaker and the second view of the image, and projects both into an embedding space. Both the speaker and listener agents learn by minimizing the constrastive loss (see Eq. 3) between between the views in this embedding space.

image. To discover what constitutes the interesting part of the image, we take inspiration from the works on **mid-level patches** [18, 34, 65], the patches in the image that are both *representative* and *discriminative* [26, 65]. This means they can be discovered in a large number of images (and hence representative), but simultaneously they should also be discriminative enough to set an image apart from the other images in the dataset. Hence, the speaker agent in our paper focuses on computing a symbolic representation in terms of these mid-level patches, as opposed to the entire image.

To summarize, we propose PatchGame, a referential game formulation where given an image, the speaker sends discrete symbolic signals in terms of mid-level patches, and the listener embeds these symbols to match them with another view of the same image in the presence of distractors. Compared to variants of this referential game setup explored in previous works [21, 27, 42], we make the following key changes:

- Agents in prior works have access to a pre-trained network, such as AlexNet [41] or VGG [64], for extracting features from images. Trained on labeled data, these features already encode the image semantics such as the class and locations of the objects in the image. Instead, in this work, the agents *do not* have access to any class labels or pre-trained representations during the training. Our agents rely on training on a large scale image dataset, and invariance introduced by various image augmentations, to learn the language in an unsupervised (or self-supervised) way.
- We propose a novel patch-based architecture for the speaker agent, which comprises of two modules: (1) PatchSymbol, a multi-layered perceptron (MLP) that operates at the patch-level and converts a given image patch into a sequence of discrete symbols, and (2) PatchRank, a ConvNet that looks at the complete image and ranks the importance of patches in a differentiable manner.
- We introduce a novel transformer-based architecture for the listener agent, consisting of two modules: (1) a language module that projects the message received from the speaker to a latent space, and (2) a vision module that projects the image into the latent space. We use a contrastive loss in this latent space to train both the speaker and the listener agents simultaneously.
- We propose new protocols to evaluate each of the speaker and listeners' modules, beyond the success of the communication task.

We assess the success of PatchGame via qualitative and quantitative evaluations of each of the proposed component, and by demonstrating some practical applications. First, we show that the speaker's PatchRank model does indicate important patches in the image. We use the top patches indicated by this model to classify ImageNet [16] images using a pre-trained Vision Transformer [19] and show that we can retain over 60% top-1 accuracy with just half of the image patches. Second, the listener's vision model (ResNet-18) can achieve upto 30.3% Top-1 accuracy just by using k-NN ($k = 20$) classification. This outperforms other state-of-the-art unsupervised approaches [26, 58] that learn discrete representations of images by 9%. Finally, we also analyze the symbols learned by our model and the impact of choosing several hyperparameters used in our experiments.

2 Related Work

Referential games. Prior to the advent of deep learning, significant research in the field of emergent communications has shown that a communication protocol can be developed or learned by agents

by playing a language game [2–5, 36, 39, 69, 72]. However, the agents employed in these works were typically located in a synthetic world and made several assumptions about the world such as the availability of disentangled representations of objects with discrete properties. More recent works [15, 22, 33, 42–44, 68, 70] have employed deep learning methods to develop a discrete language for communication between the agents. Lazaridou et al. [42] used neural network agents represented by a MLP to communicate concepts about real-world pictures. They used a fixed-sized message composed of a large vocabulary for their communication. Bouchacourt and Baroni [9], Evtimova et al. [21], Havrylov and Titov [27] relax this assumption and allow communication via variable-length sequences. Havrylov and Titov [27] allows the speaker agent to use an LSTM [30] to construct a variable-length message. Havrylov and Titov [27], Lazaridou et al. [43] show that even when we allow agents to use variable-length sequences to represent a message, they tend to utilize the maximum possible sequence to achieve the best performance (in terms of communication success).

The idea of using a Gumbel-softmax distribution [31, 48] with the straight-through trick [7] for learning a language in multi-agent environment was concurrently proposed by Mordatch and Abbeel [51] and Havrylov and Titov [27]. They show that we can achieve a more stable and faster training by using this technique as compared to reinforcement learning used in several other approaches.

Evaluating emergent communication. Evaluating the emergent language learned via these games turns out to be an equally challenging research problem, if not harder. Existing approaches use the successful completion of the task or the correlation between learned language and semantic labels as evaluation metrics. Lowe et al. [47] and Keresztury and Bruni [37] show that simple task success might not be a good or sufficient metric for evaluating the success of a game. They discuss heuristics and advocate measuring both positive signaling and positive listening independently to evaluate agents' communication. Andreas [1] provides a way of evaluating compositional structure in learned representations.

Discrete Autoencoders. In parallel to the works on emergent communication, there is a large body of research on learning discrete representations of images using some form of autoencoding or reconstruction [20, 25, 26, 52, 53, 57, 58, 60] without labels. The focus of VQ-VAE [53] and VQ-VAE-2 [58] is to learn a discrete bottleneck using vector quantization. Once we can represent any image with these discrete symbols, a powerful generative model such as PixelCNN [53, 61], or transformer [56, 71] is learned on top of these symbols to sample new images. PatchVAE [26] achieves the same using Gumbel-Softmax and imposes an additional structure in the bottleneck of VAEs. We argue that because of mismatch in the objectives of reconstruction and visual recognition tasks, each of these models trained using reconstruction-based losses do not capture meaningful representations in the symbols.

Self-supervised learning in vision. Self-supervised learning (SSL) methods, such as [10–13, 23, 29, 50, 76] have shown impressive results in recent years on downstream tasks of classification and object detection. Even though the bottleneck in these methods is continuous (and not discrete symbols), these methods have been shown to capture semantic and spatial information of the contents of the image. Unlike SSL methods, neural networks representing the two agents in our case, do not share any weights. Also, note that the continuous nature of representations learned by SSL techniques is fundamentally different from the symbolic representations used in language. And indeed, we show that a k-Nearest Neighbor classifier obtained from the continuous representations learned by SSL methods can perform better than the one obtained using Bag of Words (or symbols). However, to the best of our knowledge, our work is one of the first attempts to make representations learned in a self-supervised way more communication- or language-oriented.

3 PatchGame

We first introduce various notations and the referential game played by the agents in our work. We provide further details of architectures of the different neural network agents, as well as the loss function. We also highlight the important differences between this work and prior literature. Code and pre-trained models to reproduce the results are provided.

3.1 Referential Game Setup

Fig. 1 shows the overview of our referential game setup. Given a dataset of N images $\{\mathbf{x}_i\}_{i=1}^{N}$, we formulate a referential game [45] played between two agents, a speaker S_θ and a listener L_ϕ as

follows: As in the setting of Grill et al. [23], we generate two "random views" for every image. A random view is generated by taking a 224×224 crop from a randomly resized image and adding one or more of these augmentations - color jitter, horizontal flip, Gaussian blur and/or solarization. This prevents the neural networks from learning a trivial solution and encourages the emergent language to capture invariances induced by the augmentations. Given a batch of B images, we refer to the two views as $\{\mathbf{x}_i\}_{i=1}^{B}$ and $\{\mathbf{x}'_i\}_{i=1}^{B}$. In each iteration during training, one set of views is presented to the speaker agent and another set of the views is shown to the listener agent.

The speaker S_θ encodes each image $\mathbf{x}_i$ independently into a variable length message $\mathbf{m}_i$. Each message $\mathbf{m}$ is represented by a sequence of one-hot encoded symbols with a maximum possible length L and a fixed size vocabulary V. The space of all possible messages sent by the speaker is of the order $|V|^L$. The input to the listener L_ϕ is the batch of messages $\{\mathbf{m}_i\}_{i=1}^{B}$ from the speaker, and the second set of random views of the batch of images. The listener consists of a language module ϕ_{text} to encode embed messages and a vision module ϕ_{vision} to encode embed images. The goal of the listener is to match each message to its corresponding image.

Specifically, for a batch of B (message, image) pairs, S_θ and L_ϕ are jointly trained to maximize the cosine similarities of B actual pairs while minimizing the similarity of $(B^2 - B)$ incorrect pairs. For a target message $\mathbf{m}_j$, the image $\mathbf{x}'_j$ (augmented view of $\mathbf{x}_j$) acts as the target image while all the other $(B-1)$ images act as distractors. And vice versa, for the image $\mathbf{x}'_k$, the message $\mathbf{m}_k$ (encoded by the speaker $\mathrm{S}_\theta(\mathbf{x}_k)$) acts as the target message while all the other $(B-1)$ messages act as distractors. We use the following symmetric and contrastive loss function, also sometimes referred to as InfoNCE loss in previous metric-learning works [54, 67].

$$\mathcal{L}_{\text{text}} = -\sum_{j=1}^{B} \log \frac{\exp(\phi_{\text{text}}(\mathbf{m}_j) \cdot \phi_{\text{vision}}(\mathbf{x}'_j)/\tau)}{\sum_{k=1}^{B} \exp(\phi_{\text{text}}(\mathbf{m}_j) \cdot \phi_{\text{vision}}(\mathbf{x}'_k)/\tau)} \tag{1}$$

$$\mathcal{L}_{\text{vision}} = -\sum_{k=1}^{B} \log \frac{\exp(\phi_{\text{text}}(\mathbf{m}_k) \cdot \phi_{\text{vision}}(\mathbf{x}'_k)/\tau)}{\sum_{j=1}^{B} \exp(\phi_{\text{text}}(\mathbf{m}_j) \cdot \phi_{\text{vision}}(\mathbf{x}'_k)/\tau)} \tag{2}$$

$$\mathcal{L} = (\mathcal{L}_{\text{text}} + \mathcal{L}_{\text{vision}})/2 \tag{3}$$

where τ is a constant temperature hyperparameter.

The game setting used in our work is inspired from Lazaridou et al. [42] and Havrylov and Titov [27], but there are important differences. Both our speaker S_θ and listener L_ϕ agents are trained from scratch. This makes the game setting more challenging, since agents cannot use the pre-trained models which have been shown to encode semantic and/or syntactic information present in natural language or images. Our training paradigm, where we show different views of the same image to the speaker and listener, is inspired by the recent success of self-supervised learning in computer vision. Empirically, we observe that this leads to a more stable training and prevents the neural networks from learning degenerate solutions. However, in contrast with such self-supervised approaches, our goal is to learn a discrete emergent language as opposed to continuous semantic representations. We discuss the differences in the architecture of the two agents in the following sections.

3.2 Speaker agent architecture

A desirable characteristic of the speaker agent is that it should be able to encode "important" components of images with a variable length sequence of discrete symbols. Previous works [9, 21, 27] have achieved this by first converting the image into a continuous deterministic latent vector and then using an LSTM network [30] to generate a sequence of hidden states, and sample from this sequence of hidden state until a special end of sequence token (or maximum length) is reached. As observed by [27, 43], in order to achieve the minimum loss, the model ends up always using the maximum allowable length. In our experiments as well, we observed that having an LSTM makes the training slow and does not achieve the objective of encoding images in variable length sequences. We propose leverage two separate modules in the speaker agent S_θ to circumvent this problem - the first module called PatchSymbol (θ_{symb}) is a 2-layer MLP that computes patch-level embeddings for the image, the second module called PatchRank (θ_{rank}) is a small ConvNet that computes rank or importance of each patch in the image.

PatchSymbol, θ_{symb}. The idea of encoding an image at the patch level is inspired by the works on discovering mid-level patches [18, 34, 65] in images. We use a simple 2-hidden layer

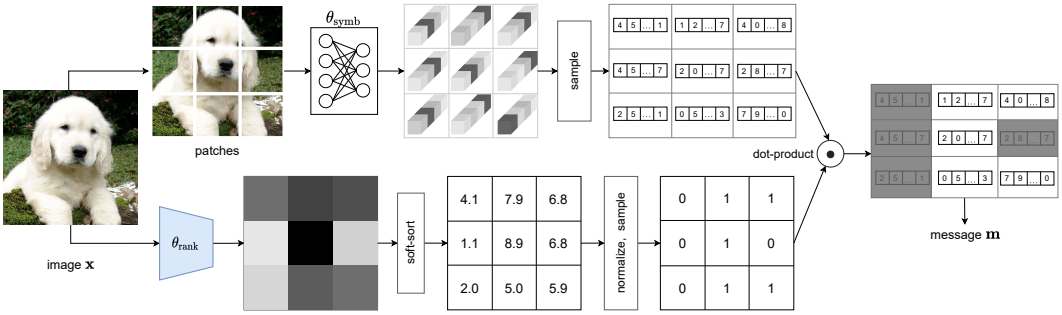


Figure 2: Speaker agent architecture. The upper branch represents the PatchSymbol, θ_{symb} module and the lower branch represents PatchRank, θ_{rank} module. Each of the modules take the raw image as the input. PatchSymbol computes a message for each patch of pre-defined size, using a fixed size vocabulary and message length. PatchRank uses a ConvNet to computes weights or the importance of each patch. Note that the symbols for a patch are context independent, however the importance of a patch depends on the context. The speaker agent combines the output of each of the models and sends a variable length message to the listener.

MLP, to encode each $\mathbb{R}^{C \times S^2}$ dimensional image patch $\mathbf{x}_{\text{patch}}$ to l vectors of log of probabilities $[\log p_1^1, \ldots, \log p_V^1], \ldots, [\log p_1^l, \ldots, \log p_V^l]$. Here C is the number of (color) channels in the input image or patch, S is the spatial dimension of a square patch, V is the size of the vocabulary used to represent a single symbol, and l is the number of symbols used to encode each patch. Hence an image of size $\mathbb{R}^{C \times H \times W}$ can be encoded using $K = \frac{HW}{S^2}$ patches, each consisting of l symbols. The vectors of log probabilities allow us to sample from a categorical distribution of V categories, with a continuous relaxation by using the Gumbel-softmax trick [31, 48]. For a given vector $[\log p_1^j, \ldots, \log p_V^j]$, we draw i.i.d samples g_i^j from the $\mathsf{Gumbel}(0, 1)$ distribution [24] and get a differentiable approximation of $\arg\max$ as follows:

$$\big[[\log p_1^1, \ldots, \log p_V^1], \ldots, [\log p_1^l, \ldots, \log p_V^l]\big] = \text{MLP}(\mathbf{x}_{\text{patch}}) \tag{4}$$

$$y^j = \mathsf{one_hot}(\arg\max_i [\log p_i^j]) \sim \left\{ \frac{\exp((\log p_i^j + g_i^j)/\tau_s))}{\sum_{k=1}^{V} \exp((\log p_k^j + g_k^j)/\tau_s)} \right\}_{i=1}^{V}, \forall j \tag{5}$$

$$\theta_{\text{symb}}(\mathbf{x}_{\text{patch}}) = \mathbf{y} = \{y^j\}_{j=1}^{l} \tag{6}$$

where τ_s controls how close the approximation is to $\arg\max$. The final output of the θ_{symb} network for the entire image $\mathbf{x}$ is L one-hot encoded $V-$dimensional symbols, where $L = l \times \frac{HW}{S^2}$. In all our experiments, we fix $V = 128$ and $l = 1$.

PatchRank, θ_{rank}. An image might have a lot of redundant patches encoded using the same symbols. The goal of the θ_{rank} network is to give an importance score to each patch. Since importance of a patch depends on the context and not the patch alone, we use a small ResNet-9 [28] to compute an importance weight for each of the $K = \frac{HW}{S^2}$ patches. One possible way to use these importance weights is to simply normalize them between $(0, 1)$ and repeat the Gumbel-Softmax trick to sample important patches. The listener network L_ϕ would see only the message consisting of "important" patches. However, we empirically observed that a simple min-max or L2-normalization allows the network to assign high weights to each patch and effectively send the entire sequence of length L to the listener. Instead, we propose to use a *differentiable ranking* algorithm by Blondel et al. [8] to convert the importance weights to soft-ranks $\{1, \ldots, K\}$ in $O(K \log K)$ time. This method works by constructing differentiable operators as projections onto the convex hull of permutations. Once we have the vector of soft-ranks $\mathbf{r} \in \mathbb{R}^K$, we normalize the ranks and sample binary values again using a special case of the Gumbel-softmax trick for Bernoulli distributions [31, 48] as

$$\mathbf{r}(\mathbf{x}) = \arg\max_{\pi \in \Sigma} \langle \text{CNN}(\mathbf{x}), \rho_\pi \rangle \tag{7}$$

$$\theta_{\text{rank}}(\mathbf{x}) \sim \text{Bern}\left(\frac{1}{K}\mathbf{r}(\mathbf{x})\right) \tag{8}$$

where $\text{CNN}(\mathbf{x})$ are the importance weights obtained by applying a ResNet-9 to the image $\mathbf{x}$, Σ is the set of all $K!$ permutations, and ρ_π are the ranks corresponding to the permutation $\pi \in \Sigma$. We refer the reader to [8] for a detailed description of the soft-sort algorithm. Therefore, the final symbols

encoded by the speaker agent, S_θ, is given by:

$$\mathrm{S}_\theta(\mathbf{x}) = \mathbf{m} = \theta_{\text{symb}}(\mathbf{x}) \cdot \theta_{\text{rank}}(\mathbf{x}) \tag{9}$$

3.3 Listener agent architecture

As discussed in §3.1, the listener agent L_ϕ consists of a language module ϕ_{text} and a vision module ϕ_{vision}. We implement ϕ_{text} using a small transformer encoder [71]. We prepend a $\langle$CLS$\rangle$ token at the beginning of each message sequence received from the speaker [17], and use the final embedding of $\langle$CLS$\rangle$ to compute the loss described in Eq. 3. We implement ϕ_{vision} using a small vision transformer [19], and follow a similar procedure as in the text module to obtain the final image embedding. Both the text and vision modules use a similar transformer encoder architecture (no weight sharing) with 192 hidden size, 12 layers and 3 attention heads. Following [11, 13, 76], we add a high dimensional projection at the end of the last layer before computing the loss function.

3.4 Training

Each of the weights of the speaker and listener agents $\{\theta_{\text{symb}}, \theta_{\text{rank}}, \phi_{\text{vision}}, \phi_{\text{text}}\}$ are optimized jointly during the training. We use a 2-layer MLP for θ_{symb}, ResNet-9 [28] for θ_{rank}, a ResNet-18 [28] for ϕ_{vision}, and a small transformer encoder (hidden size $= 192$, 3 heads, 12 layers) for ϕ_{text}. All the experiments are conducted on the training set of ImageNet [16], which has approximately 1.28 million images from 1000 classes. We create a training and validation split from the training set by leaving aside 5% of the images for validation. After obtaining the final set of hyper-parameters, we retrain on the entire training set for 100 epochs. We use Stochastic Gradient Descent (SGD) with momentum and cosine learning rate scheduling. In order to train the stochastic component of the Speaker, we use the straight-through trick [7]. We reiterate that the speaker and listener do not share weights, and the only supervision used is the InfoNCE loss defined in Eq. 3. Please refer to the appendix and code attached in the supplementary material for more details.

4 Experiments

We evaluate the success of communication in the referential game and impact of various hyper-parameters on the success. Also, following the work of Lowe et al. [47], we evaluate the emergent communication in two primary ways. In section 4.1, we measure *positive signaling*, which means that S_θ sends messages relevant to the observation. In section 4.2, we measure *positive listening*, which indicates that the messages are influencing the L_ϕ agent's behavior.

4.1 Positive Signaling - Visualizing Patch Ranks from θ_{rank}

We first visualize the output of PatchRank module as a heatmap overlayed on the original image in Fig. 3. Most important patches are colored towards 'red' and the least important ones are colored towards 'blue'. The figure shows that our PatchRank can capture important and discriminative parts of the images. In case of images of various animals and plants, the model assigns the highest important to discriminative body parts. For the inanimate objects such as abacus or revolver the model is able to distinguish between the foreground and the background. Note that although approaches such as GradCam [62] can provide pixel-level importance heatmaps, they require extensive supervision. Our method on the other hand is self-supervised. We also shows some of the failure cases when discriminative patches in the image cover majority of the image on the rightmost 2 columns of Fig. 3.

4.2 Positive Signaling - Image Classification with subset of patches provided by θ_{rank}

Recently proposed Vision Transformers (ViT) by Dosovitskiy et al. [19] have gained popularity because of their simplicity and performance. These models treat an image as a sequence of $N \times N$ patches, use an MLP to convert the patch into an embedding, and finally use these set of patches to perform classification. We first note that both the inference time and memory consumption of ViT depend largely on the length of sequence, because of the $O(N^2)$ self-attention operation. Secondly, the performance of the Vision Transformers drop if instead of using all patches, we only use a subset of patches during inference. This provides us a simple way to evaluate the θ_{rank} module. We artificially constrain the number of patches available to ViT during inference. We measure the Top-1 Accuracy of ViT using only k allowed patches. The selection of the patches is done by the θ_{rank} model.

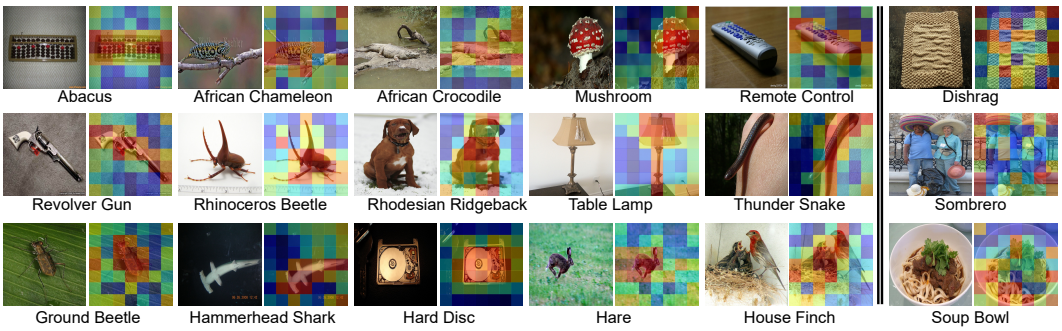
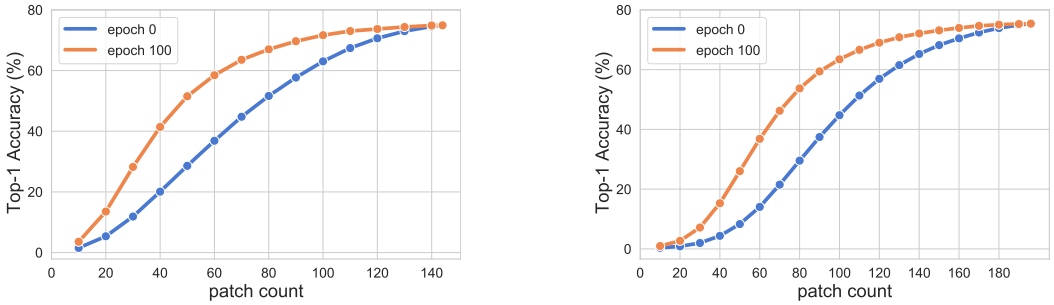


Figure 3: Heat maps based on Patch Importance. Red represents the most important patches and Blue the least important. Labels are listed for better understanding and are not used during training. We have used 224×224 images and 32×32 patches to get 49 non-overlapping patches per image. Model successfully separates the primary object in an image - the "device" in the "Abacus" image, "rabbit" in the "hare" image etc. *(as shown by higher concentration of yellow-to–red patches on these areas).* Some failure cases are shown as well on the right.



(a) 32×32 patches **(b)** 16×16 patches

Figure 4: Impact of number of patches used during evaluation in a pre-trained ViT [19]. The performance of a ViT drops if we provide fewer patches during inference. However a PatchRank model (which is a small ResNet-9) can provide important patches to ViT during inference with minimal loss in Top-1 accuracy.

We consider two different pre-trained ViT [19] models, one trained using 32×32 patches, on the images of size 384×384 (so the number of patches in an image is 144), while the second model is trained using 16×16 patches on the images of size 224×224 (196 patches per image). In Fig. 4a and 4b, we show the Top-1 accuracy obtained by the pre-trained ViT models using important patches predicted by θ_{rank} at different values of k. At the 0th epoch (with random weights), the performance of ViT drops almost linearly as we lower the patch count k. At the 100th epoch, at the end of the training, we observe that performance of the ViT does not drop as drastically.

4.3 Positive Signaling - Visualizing θ_{symb} symbols

As mentioned earlier, we are using a vocabulary $V = 128$ and patch size $S = 32$ in our base model. This means θ_{symb} has to map each 32×32 patch to one of the 128 available symbols. A natural way to analyze what the symbols are encoding to see is to visualize their corresponding patches. While $V = 128$ is far too small a vocabulary to describe all possible patches, we observe some interesting patterns in Fig. 5. Many of the symbols seem to adhere to specific concepts repeatedly. We observed that symbols have a lesser preference for color, but more preference for texture and shape. We discovered several symbols corresponding to textures such as grass, branches, and wood. We also noticed many symbols firing for the patches corresponding to text, eyes, and even faces. There can be multiple symbols representing single concept, *e.g.*, both symbol 91 and 123 both fire in case of eyes.

4.4 Positive Signaling - Visual Bag of Words Classification with θ_{symb}

Representing the image as visual bag of words is a well-known technique in image retrieval works [66]. Since our task is to represent image with words as well, we devise the following way to evaluate the PatchSymbol module or θ_{symb}. For each image in the ImageNet training and validation set, we

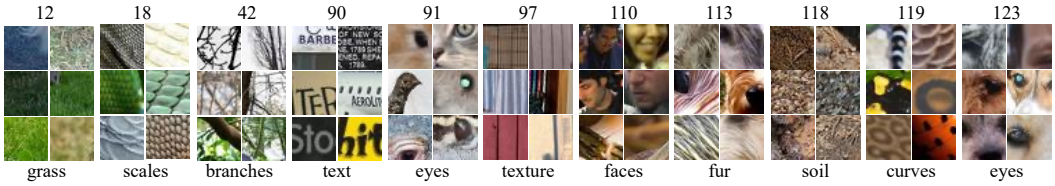


Figure 5: Visualizing patches corresponding to various symbols. The number in the top row corresponds to 1 of 128 vocabulary ids. 6 representative patches corresponding to each patch are shown. The bottom row corresponds to our interpretation of what concept that symbol might be capturing.

Table 1: Performance of Speaker's PatchSymbol module θ_{symb} - Downstream classification accuracy for ImageNet dataset using visual BoW method with Complement Naive Bayes

Method	Top-1 (%)	Top-5(%)
VQ-VAE2 [53]	0.88	5.02
PatchVAE [25] ($S = 32$)	1.02	4.48
PatchVAE [25] ($S = 16$)	1.00	5.04
Ours ($S = 32$)	**1.28**	**6.10**
Ours ($S = 16$)	1.20	5.94

Table 2: Performance of Listener's Vision module ϕ_{vision} - Downstream classification accuracy for ImageNet dataset using k-NN (k=20)

Method	Top-1 (%)	Top-5 (%)
MoCo-v2 [29] (R18)	36.8	60.3
VQ-VAE2 [54]	17.2	30.5
PatchVAE [25] (R18, $S = 32$)	21.3	36.2
PatchVAE [25] (R18, $S = 16$)	16.4	28.5
Ours (R18, $S = 32$)	**30.3**	**49.9**
Ours (R18, $S = 16$)	27.6	46.2

compute the message or the list of symbols generated by θ_{symb} (and filtered according to θ_{rank}). We then compute a feature for each image using the weighted frequency of symbols in the image or *tf-idf* [32]. We then train a simple complement naive bayes classifier [59] on top of these feature vectors. For benchmarking, we choose two prior works that encode images to symbols and have shown strong results in generative modeling. **VQ-VAE-2** or Vector Quantized Variational AutoEncoder was initially proposed by Oord et al. [53] and later improved by Razavi et al. [58] for both conditional and unconditional large scale image generation. We retrain the model for 100 epochs at 256×256 resolution used in the paper using [63]'s repository. **PatchVAE** by Gupta et al. [26] proposes a structured variant of Variational AutoEncoders [38]. We use the authors's code and retrain their models for a 7×7 bottleneck (corresponding to $S = 32$ patch size), and a 14×14 bottleneck (corresponding to $S = 16$ patch size), each for 100 epochs.

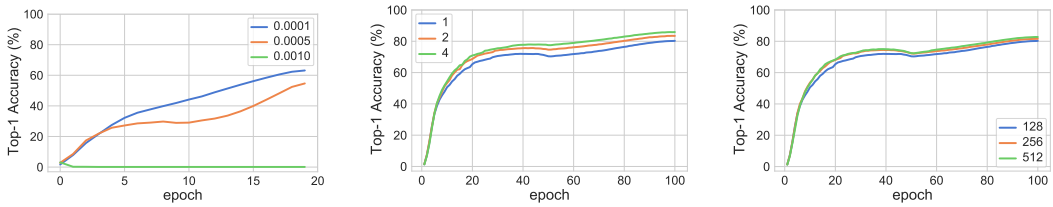
Table 1 shows the Top-1 and Top-5 % classification accuracies on ImageNet. The patch symbols generated with our method outperform those generated by VQ-VAE-2 and PatchVAE in the downstream classification task by factors of 45% and 25% respectively. Note that classification accuracy is far below the ones that can be obtained using standard techniques, however the goal of this evaluation is to demonstrate that the symbols do capture meaningful information.

4.5 Positive Listening

Next, we evaluate the vision module, or ϕ_{vision} of the listener. We follow the protocol employed by various approaches in self-supervised learning literature. We consider the features obtained at the final pooling layer of the ϕ_{vision} (which is a ResNet-18 in our case). Next, we run a k-NN classification with $k = 20$ on the validation dataset. Table 2 shows the Top-1 and Top-5 % accuracy obtained on ImageNet using the listener's vision module and the baselines approaches. Although our method outperforms VQ-VAE-2 and PatchVAE (methods that learn a discrete image representation) we observe that there is still a gap between representations learned by these models as compared to the representations learned by continuous latent models such as MoCo [29]. Note that, because of the resource constraints, all results reported in the table are obtained by training ResNet-18 for only 100 epochs. The results for both MoCo-v2 and our approach continue to improve if we continue the training beyond 100 epochs (as also noted by He et al. [29]).

4.6 Ablation study

A communication iteration is successful if the receiver is able to match the message sent by the speaker to the correct target image. In our experiments, we use an effective batch size is 512 (split over 4 GPUs), so the chance accuracy of success of the speaker is 0.19%. We measure the Top-1 accuracy for each image of the batch and average it over validation data at the end of epoch. In Fig. 6a,



(a) Learning rate **(b)** Length of a single patch **(c)** Vocabulary

Figure 6: Impact of various hyperparameters on the success of communication. Top-1 accuracy denotes the percentage of messages in a batch that model was able to match to the corresponding image. Having a higher message length and vocabulary helps the model to learn faster.

we observe that too high or too low learning rates can be detrimental to the success of communication. We fix the learning rate at 0.0001 which shows the best validation performance empirically. We train our models at different vocabulary sizes, and different message lengths for 100 epochs. From Fig. 6b and 6c, we observe that having either a large vocabulary or larger message length allows the model to reach high accuracy faster. This is intuitive since the larger the space spanned by the messages is, the easier it is for receiver to distinguish between the images. A large message length increases the possible number of messages exponentially, at the cost of much larger computation cost since self-attention [17, 71] is an $O(N^2)$ operation. A large vocabulary also increases the both the span of messages and computation cost linearly.

5 Discussion

Summary. In this work, we have shown that two cooperative neural network agents can develop a variable length communication protocol to solve a given task on a large scale image dataset with only self-supervision. To the best of our knowledge, this is the first work to develop an emergent communication via mid-level patches without using any pre-trained models or external labels. We have introduced a novel mid-level-patch based architecture for the speaker agent, in order to represent only the "interesting" parts of image with discrete symbols. The listener agent learns to align the messages and images using a language and image encoders to a common embedding space. The speaker and listener agents are trained end-to-end jointly to capture invariances induced by various data augmentations in order to solve a constrastive task. We propose a number of quantitative and qualitative measures to analyse and evaluate the emerged language. We also show two major applications of the developed approach - (1) it can be used to extract representative and discriminative parts of the image (2) the pre-trained network can be used for transfer learning in image classification tasks.

Reproducibility. Code and pre-trained models to reproduce the results are provided.

Limitations and Future Work. There are a few limitations of our approach. Firstly, one of the applications discussed in Section 4.1 is using fewer patches for inference in ViT. Although, using fewer patches reduces the memory cost of ViT, the overhead of using another neural network for predicting the importance patches means we our gain is minimal. In the future, we would like to explore even faster architectures to have a bigger impact on classification speed. Second, our method only works with fixed-size square patches. Discovering arbitrary sized mid-level patches is a challenging task that we would like to address in future work. Third, the language emerged from our current model is not grounded in natural language and requires human intervention for interpretation. Going forward, we would like to ground this emerged language in the natural language.

Broader Impact. Like most self-supervised approaches, our approach is data hungry and trained using a large amount of unlabeled data collected from Internet. While the data collection and labeling has been discussed in the original paper [16], the community has focused towards imbalance and privacy violation in existing image datasets only recently [74]. A recent study [75] shows that 997 out of 1000 ImageNet categories are not 'people' categories. To compare against previous methods and allow future benchmarking, we provide results on the 2012 version of the dataset with 1.28 million images in training set and 50000 images in the validation set.

References

[1] Jacob Andreas. Measuring compositionality in representation learning. *arXiv preprint arXiv:1902.07181*, 2019.

[2] Jacob Andreas and Dan Klein. Analogs of linguistic structure in deep representations. *arXiv preprint arXiv:1707.08139*, 2017.

[3] Jacob Andreas, Dan Klein, and Sergey Levine. Learning with latent language. *arXiv preprint arXiv:1711.00482*, 2017.

[4] Andrea Baronchelli, Maddalena Felici, Vittorio Loreto, Emanuele Caglioti, and Luc Steels. Sharp transition towards shared vocabularies in multi-agent systems. *Journal of Statistical Mechanics: Theory and Experiment*, 2006(06):P06014, 2006.

[5] John Batali. Computational simulations of the emergence of grammar. *Approach to the Evolution of Language*, pages 405–426, 1998.

[6] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *European conference on computer vision*, pages 404–417. Springer, 2006.

[7] Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013.

[8] Mathieu Blondel, Olivier Teboul, Quentin Berthet, and Josip Djolonga. Fast differentiable sorting and ranking. In *International Conference on Machine Learning*, pages 950–959. PMLR, 2020.

[9] Diane Bouchacourt and Marco Baroni. Miss tools and mr fruit: Emergent communication in agents learning about object affordances. *arXiv preprint arXiv:1905.11871*, 2019.

[10] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *arXiv preprint arXiv:2006.09882*, 2020.

[11] 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.

[12] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[13] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. *arXiv preprint arXiv:2011.10566*, 2020.

[14] Herbert H Clark. *Using language*. Cambridge university press, 1996.

[15] 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.

[16] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[17] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[18] Carl Doersch, Saurabh Singh, Abhinav Gupta, Josef Sivic, and Alexei Efros. What makes paris look like paris? *ACM Transactions on Graphics*, 31(4), 2012.

[19] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[20] Patrick Esser, Robin Rombach, and Björn Ommer. Taming transformers for high-resolution image synthesis. *arXiv preprint arXiv:2012.09841*, 2020.

[21] Katrina Evtimova, Andrew Drozdov, Douwe Kiela, and Kyunghyun Cho. Emergent communication in a multi-modal, multi-step referential game. *arXiv preprint arXiv:1705.10369*, 2017.

[22] Jakob N Foerster, Yannis M Assael, Nando De Freitas, and Shimon Whiteson. Learning to communicate with deep multi-agent reinforcement learning. *arXiv preprint arXiv:1605.06676*, 2016.

[23] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv preprint arXiv:2006.07733*, 2020.

[24] Emil Julius Gumbel. *Statistical theory of extreme values and some practical applications: a series of lectures*, volume 33. US Government Printing Office, 1954.

[25] Kamal Gupta, Alessandro Achille, Justin Lazarow, Larry Davis, Vijay Mahadevan, and Abhinav Shrivastava. Layout generation and completion with self-attention. *arXiv preprint arXiv:2006.14615*, 2020.

[26] Kamal Gupta, Saurabh Singh, and Abhinav Shrivastava. Patchvae: Learning local latent codes for recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4746–4755, 2020.

[27] Serhii Havrylov and Ivan Titov. Emergence of language with multi-agent games: Learning to communicate with sequences of symbols. *arXiv preprint arXiv:1705.11192*, 2017.

[28] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[29] 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*, pages 9729–9738, 2020.

[30] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[31] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

[32] Karen Sparck Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 1972.

[33] Emilio Jorge, Mikael Kågebäck, Fredrik D Johansson, and Emil Gustavsson. Learning to play guess who? and inventing a grounded language as a consequence. *arXiv preprint arXiv:1611.03218*, 2016.

[34] Mayank Juneja, Andrea Vedaldi, CV Jawahar, and Andrew Zisserman. Blocks that shout: Distinctive parts for scene classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 923–930, 2013.

[35] Frederic Jurie and Bill Triggs. Creating efficient codebooks for visual recognition. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 1, pages 604–610. IEEE, 2005.

[36] Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. Referitgame: Referring to objects in photographs of natural scenes. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 787–798, 2014.

[37] Bence Keresztury and Elia Bruni. Compositional properties of emergent languages in deep learning. *arXiv preprint arXiv:2001.08618*, 2020.

[38] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[39] Simon Kirby. Natural language from artificial life. *Artificial life*, 8(2):185–215, 2002.

[40] Ranjay Krishna, Ines Chami, Michael Bernstein, and Li Fei-Fei. Referring relationships. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6867–6876, 2018.

[41] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.

[42] Angeliki Lazaridou, Alexander Peysakhovich, and Marco Baroni. Multi-agent cooperation and the emergence of (natural) language. *arXiv preprint arXiv:1612.07182*, 2016.

[43] Angeliki Lazaridou, Karl Moritz Hermann, Karl Tuyls, and Stephen Clark. Emergence of linguistic communication from referential games with symbolic and pixel input. *arXiv preprint arXiv:1804.03984*, 2018.

[44] Jason Lee, Kyunghyun Cho, Jason Weston, and Douwe Kiela. Emergent translation in multi-agent communication. *arXiv preprint arXiv:1710.06922*, 2017.

[45] David Lewis. *Convention: A philosophical study*. John Wiley & Sons, 1969.

[46] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[47] 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.

[48] 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.

[49] Tomas Mikolov, Armand Joulin, and Marco Baroni. A roadmap towards machine intelligence. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 29–61. Springer, 2016.

[50] Ishan Misra and Laurens van der Maaten. Self-supervised learning of pretext-invariant representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6707–6717, 2020.

[51] Igor Mordatch and Pieter Abbeel. Emergence of grounded compositional language in multi-agent populations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[52] Charlie Nash, Jacob Menick, Sander Dieleman, and Peter W Battaglia. Generating images with sparse representations. *arXiv preprint arXiv:2103.03841*, 2021.

[53] Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. *arXiv preprint arXiv:1711.00937*, 2017.

[54] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[55] Steven Pinker. *The language instinct: How the mind creates language*. Penguin UK, 2003.

[56] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. 2018.

[57] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. *arXiv preprint arXiv:2102.12092*, 2021.

[58] Ali Razavi, Aaron van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. *arXiv preprint arXiv:1906.00446*, 2019.

[59] Jason D Rennie, Lawrence Shih, Jaime Teevan, and David R Karger. Tackling the poor assumptions of naive bayes text classifiers. In *Proceedings of the 20th international conference on machine learning (ICML-03)*, pages 616–623, 2003.

[60] Jason Tyler Rolfe. Discrete variational autoencoders. *arXiv preprint arXiv:1609.02200*, 2016.

[61] Tim Salimans, Andrej Karpathy, Xi Chen, and Diederik P Kingma. Pixelcnn++: Improving the pixelcnn with discretized logistic mixture likelihood and other modifications. *arXiv preprint arXiv:1701.05517*, 2017.

[62] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.

[63] Kim Seonghyeon. vq-vae-2-pytorch. https://github.com/rosinality/vq-vae-2-pytorch, 2019.

[64] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[65] Saurabh Singh, Abhinav Gupta, and Alexei A Efros. Unsupervised discovery of mid-level discriminative patches. In *European Conference on Computer Vision*, pages 73–86. Springer, 2012.

[66] Josef Sivic and Andrew Zisserman. Video google: A text retrieval approach to object matching in videos. In *IEEE International Conference on Computer Vision*, 2003.

[67] Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pages 1857–1865, 2016.

[68] Matthew Spike, Kevin Stadler, Simon Kirby, and Kenny Smith. Minimal requirements for the emergence of learned signaling. *Cognitive science*, 41(3):623–658, 2017.

[69] Luc Steels. What triggers the emergence of grammar? *Society for the Study of Artificial Intelligence and Simulation of Behaviour*, 2005.

[70] Sainbayar Sukhbaatar, Arthur Szlam, and Rob Fergus. Learning multiagent communication with backpropagation. *arXiv preprint arXiv:1605.07736*, 2016.

[71] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017.

[72] Kyle Wagner, James A Reggia, Juan Uriagereka, and Gerald S Wilkinson. Progress in the simulation of emergent communication and language. *Adaptive Behavior*, 11(1):37–69, 2003.

[73] Ludwig Wittgenstein. *Philosophical investigations*. John Wiley & Sons, 1953.

[74] Kaiyu Yang, Klint Qinami, Li Fei-Fei, Jia Deng, and Olga Russakovsky. Towards fairer datasets: Filtering and balancing the distribution of the people subtree in the imagenet hierarchy. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pages 547–558, 2020.

[75] Kaiyu Yang, Jacqueline Yau, Li Fei-Fei, Jia Deng, and Olga Russakovsky. A study of face obfuscation in imagenet. *arXiv preprint arXiv:2103.06191*, 2021.

[76] 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.

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] See Sec. 4.
   (b) Did you describe the limitations of your work? [Yes] See Fig. 3 and Limitations paragraph in Sec. 5.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See the Broader Impact paragraph in Sec. 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] The code is included, and the data can be obtained from the providers using the link provided in the code.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See the Sec. 4.6.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No] Given the resource constraints and the practice followed in various self-supervised works, we are unable to provide the error bars for all possible experiments at the moment. If deemed necessary by the reviewers, we will include them in the final version.
   (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 the Sec. 4.6.
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 use ImageNet dataset and cite it multiple times [16].
   (b) Did you mention the license of the assets? [Yes] The license for our code is included in the code provided in the supplementary material. The license for ImageNet dataset is mentioned in our code and provided on the provider's website.
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] The code is included in the supplementary material.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes] See the Broader Impact paragraph in Sec. 5.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [Yes] See the Broader Impact paragraph in Sec. 5.
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]