Information Minimization In Emergent Languages

Eugene Kharitonov  
Facebook AI Research  
kharitonov@fb.com

Rahma Chaabouni  
Facebook AI Research / LSCP  
rchaabouni@fb.com

Diane Bouchacourt  
Facebook AI Research  
dianeb@fb.com

Marco Baroni  
Facebook AI Research / ICREA  
mbaroni@fb.com

Abstract

There is a growing interest in studying the languages emerging when neural agents are jointly trained to solve tasks that require communication through discrete messages. We investigate here the information-theoretic complexity of such languages, focusing on the most basic two-agent, one-symbol, one-exchange setup. We find that, under common training procedures, the emergent languages are subject to an information minimization pressure: The mutual information between the communicating agent’s inputs and the messages is close to the minimum that still allows the task to be solved. After verifying this information minimization property, we perform experiments showing that a stronger discrete-channel-driven information minimization pressure leads to increased robustness to overfitting and to adversarial attacks. We conclude by discussing the implications of our findings for the studies of artificial and natural language emergence, and for representation learning.

1 Introduction

The ability to communicate in a human-like, discrete language is considered as an important stepping stone towards a general AI [Mikolov et al., 2016, Chevalier-Boisvert et al., 2019]. This vision revived research interest in artificial neural agents communicating with each other through discrete messages to jointly solve a task [Lazaridou et al., 2016, Kottur et al., 2017, Havrylov and Titov, 2017, Choi et al., 2018, Lazaridou et al., 2018]. These simulations might also help to shed light on how human language itself evolved [Kirby, 2002, Hurford, 2014, Graesser et al., 2019].

We study here the communication biases that arise in this setup. Specifically, we are interested in the properties of the emergent protocols (languages), namely their complexity and redundancy w.r.t. the task being solved. We use simple signaling games [Lewis, 1969], as introduced to the current language emergence literature by [Lazaridou et al., 2016] and adopted in several later studies [Havrylov and Titov, 2017, Lazaridou et al., 2018, Bouchacourt and Baroni, 2018]. There are two agents, Sender and Receiver, provided with individual inputs at the beginning of each episode. Sender sends a single message to Receiver, and Receiver has to perform an action based on its input and the received message. Importantly, there is no direct supervision on the message protocol. We consider agents that are deterministic functions of their inputs (after training).

As an example, consider an image classification task where an image is shown to Sender, and Receiver has to chose its class out of $n$ equiprobable classes, solely based on the message from Sender. Would Sender’s messages encode the target class? In this case, the protocol would be “simple” and only transmit $\log_2 n$ bits. Alternatively, Sender could try to describe the full image, and let Receiver decide

1That is, at each time, the communicating agent must produce one symbol from a fixed vocabulary.
We study two signaling games. In Guess Number, the agents are trained to recover an integer-vector representing a random 8-bit integer \( z \). On its class. The emergent protocol would be more “complex”, encoding more information about the image.

We find that, once the agents are successfully trained to jointly solve a task, the emergent protocol tends to minimize mutual information between Sender input and messages or, equivalently in our setup, the entropy of the messages. In other words, the agents consistently approximate the simplest successful protocol. In the example above, the protocol will transmit \( \approx \log_2 n \) bits.

More precisely, since the agents are deterministic, we can connect entropy and mutual information between Sender’s input \( i_s \), messages \( m = S(i_s) \), and Receiver’s output (the chosen action) \( o = R(m) \) by using the standard Data Processing Inequalities [Cover and Thomas, 2012]:

\[
H(i_s) \geq H(m) \geq H(o) \quad H(i_s) \geq I(i_s; m) \geq I(i_s; o)
\] (1)

Once an agent pair is successfully trained to solve the task, our empirical measurements indicate that the messages \( m \) in the emergent protocol tend to approach the lower bounds \( H(o) \) and \( I(i_s; o) \):

\[
H(m) \approx H(o) \quad I(i_s; m) \approx I(i_s; o)
\] (2)

This effect holds even if there is a large gap to the upper-bound \( H(i_s) \). Moreover, by controlling the amount of redundant information that is provided to Receiver on the side, without changing other parameters, we observe that the emergent protocol becomes simpler (lower \( H(m) \) and \( I(i_s; m) \)) as the amount of information directly provided to the Receiver grows. In other words, the emergent protocol ‘adapts’ to minimize the information that passes through it.

Our paper is split in two parts. First, we study the complexity of the protocols emerging in two setups, in function of various factors. Next, having empirically established the aforementioned information minimization property, we show that neural networks that rely on a discrete communication channel exhibit desirable properties shared with architectures incorporating the Information Bottleneck [Tishby et al., 1999] principle: robustness to overfitting and against adversarial noise. In the conclusion, we outline the implications of our findings for the study of discrete-channel architectures in AI as well as in the investigation of the origin and usage of human language.

## 2 Methodology

### 2.1 Games

We study two signaling games. In Guess Number, the agents are trained to recover an integer-vector representing a random 8-bit integer \( z \), 0 ≤ \( z \) ≤ 255 by sampling its 8 bits independently from the uniform Bernoulli distribution. All bits are revealed to Sender as a 8-dimensional binary vector \( i_s \). The last \( k \) bits are revealed to Receiver (0 ≤ \( k \) ≤ 8) as its input \( i_r \). Sender sends a single-symbol message \( m \) to Receiver. In turn, Receiver outputs a vector \( o \) that recovers all the bits of \( z \) and should be equal to \( i_s \). By changing \( k \), we control the minimal information that Sender has to send Receiver so that it can answer perfectly: \( I_{\text{min}}(m; i_s) = 8 - k \) (similarly, if the task is solved, \( H(m) \geq 8 - k \)).

In this game, Sender has a linear layer that maps the input vector \( i_s \) to a hidden representation of size 10, followed by a leaky ReLU activation. Next is a linear layer followed by a softmax over the vocabulary. Receiver linearly maps both its input \( i_r \) and the message to 10-dimensional vectors, concatenates them, applies a fully connected layer with output size 20, followed by a leaky ReLU. Finally, another linear layer and a sigmoid nonlinearity are applied. When trained with REINFORCE and Stochastic Computation graph approach (see Section 2.2), we increased the sizes of hidden layers 3x, as this led to a more robust convergence. When trained with Gumbel-Softmax relaxation, the hidden sizes are not changed.

**Image Classification** In this game, the agents are jointly trained to classify 28x28 MNIST images. As input \( i_s \), Sender receives the entire image. Receiver also receives 28x28 images as \( i_r \), but only...
We use the running mean baseline $b$, where $\tau$. The expectations are estimated by sampling $m$.

Training to communicate with discrete messages is non-trivial, as backpropagation through communicated messages is not possible. Current language emergence work mostly uses Gumbel-Softmax relaxation (e.g. Havrylov and Titov, 2017) or REINFORCE (e.g. Lazaridou et al., 2016) to get gradient estimates. We also explore the Stochastic Graph optimization approach. We plug the obtained gradient estimates into the Adam optimizer (Kingma and Ba, 2014).

**Gumbel-Softmax relaxation** The core observation is that samples from the Gumbel-Softmax [Madison et al., 2016, Jang et al., 2016] distribution (a) are reparameterizable, hence allow gradient-based training, and (b) approximate samples from the corresponding Categorical distribution. To get a sample that approximates an $n$-dimensional Categorical distribution with probabilities $p_i$, we draw $n$ i.i.d. samples $g_i$ from Gumbel(0,1) and use them to calculate a vector $y$ with components:

$$y_i = \frac{\exp[(g_i + \log p_i)/\tau]}{\sum_j \exp[(g_j + \log p_j)/\tau]}$$

where $\tau$ is the temperature hyperparameter. As $\tau$ tends to 0, the samples $y_i$ get closer to one-hot samples; as $\tau \to +\infty$, the components $y_i$ become uniform. During training, we use these relaxed samples as messages from Sender, making the entire Sender/Receiver setup differentiable.

**REINFORCE** by Williams [1992] is a standard reinforcement learning algorithm. In our setup, it estimates the gradient of the expectation of the loss $\mathcal{L}(\theta)$ w.r.t. the parameter vector $\theta$ as follows:

$$E_{i \sim S, m \sim \mathcal{S}(i), o \sim \mathcal{R}(m, i)}[\mathcal{L}(\theta; l) - b] \log \mathcal{P}_\theta(m; i, o)$$

The expectations are estimated by sampling $m$ from Sender and, after that, sampling $o$ from Receiver. We use the running mean baseline $b$ [Williams, 1992, Greensmith et al., 2004] as a control variate. Another common trick we adopt is to add an entropy regularization term [Williams and Peng, 1991, Mnih et al., 2016] that favors higher entropy. We impose entropy regularization on the outputs of the agents with coefficients $\lambda_s$ (Sender) and $\lambda_r$ (Receiver).

**Stochastic computational graph estimator** In our setup, the gradient estimate approach of Schulman et al. [2015] reduces to computing the gradient of the following surrogate function:

$$E_{i \sim S, m \sim \mathcal{S}(i)}[\mathcal{L}(\theta; l) + \text{stop gradient} \mathcal{L}(\theta; l) - b] \log \mathcal{P}_\theta(m; i, s)$$

In this case, we do not sample actions of Receiver and gradients of its parameters are obtained with standard backpropagation (the first term in Eq. 5). Sender’s messages are sampled, and its gradient are calculated akin to REINFORCE (the second term in Eq. 5). We apply entropy regularization on Sender’s output (with coefficient $\lambda_s$) and use the mean baseline $b$.

### 2.3 Experimental protocol

At test-time, we select Sender’s message symbol greedily, hence the messages are discrete and Sender represents a (deterministic) function $S$ of its input $i_s$, $m = S(i)$. Consequently, message entropy
and mutual information between the messages \( m \) and Sender’s inputs \( i_s \) are equal:

\[
I(i_s; m) = H(m) - H(m|i_s) = H(m)
\]  

Calculating the entropy of the distribution of discrete messages is straightforward. In Guess Number, we enumerate all 256 possible values of \( z \) as inputs, save the messages from Sender and calculate entropy \( H(m) \). As MNIST contains a test set, we use it for the Image Classification analysis. Due to a higher computational complexity, we use the first 16k test images.

We run the training procedure for each task and agent hyper-parameter combination. We select the runs that achieved a high level of task-specific performance (training accuracy above 0.99 for Guess Number and validation accuracy above 0.98 for the harder MNIST classification task). Next, for each selected run, we calculate message entropy as outlined above. We thus study the typical behavior of the agents provided they succeeded at the game.

3 Experiments

3.1 Information minimization property

**Guess Number** In Figure 1, the horizontal axes span the number of bits (binary digits) of \( z \) that Receiver lacks, \( 8 - k \). The vertical axis reports information content of the protocol, measured by mutual information between input and messages, \( I(i_s; m) = H(m) \) (Eq. 6). Each integer on the horizontal axis corresponds to a game configuration, and for each such configuration we aggregate multiple (successful) runs with different hyperparameters and random seeds. The Lower bound lines indicate the minimal amount of bits \( I_{\text{min}}(i_s; m) \) Sender has to send in a particular configuration for the task to be solvable, \( 8 - k \). The upper bound (not shown) is equal to \( H(i_s) = 8 \) bits.

Firstly, consider the configurations where the Receiver’s input is insufficient to answer correctly in all cases (at least one binary digit hidden, \( k \leq 7 \)). From Figure 1a we observe that the transmitted information is strictly monotonically increasing with the number of binary digits hidden from Receiver. Thus, even if Sender sees the very same input in all configurations, a more nuanced protocol only develops when it is necessary. Moreover, for every configuration with \( k \leq 7 \), the
I(\(i_s; m\)), bits

All runs

(a) Successful runs pooled together.

(b) Successful runs grouped by temperature.

Figure 2: Image Classification: mutual information between Sender’s input \(i_s\) and protocol messages \(m\). Shaded regions mark standard deviation.

information transmitted by the protocol \(I(\(i_s; m\))\) remains close to the lower bound. This information minimization property holds for all the considered training approaches across all configurations.

Consider next the configuration where Receiver is getting the whole integer \(z\) as its input (\(k = 8\), leftmost configuration in Figure 1, corresponding to 0 on x axis). Based on the observations above, one would expect that the protocol would transmit nearly zero information in this case (as no information needs to be transmitted). However, the measurements indicate that the protocol is encoding considerably more information. It turns out that this information is entirely ignored by Receiver. To demonstrate this, we fed all possible distinct inputs to Sender, obtained the corresponding messages, and shuffled them to destroy any information about the inputs they might carry. The shuffled messages were then passed to Receiver alongside with its own (un-shuffled) inputs. The overall performance was not affected by this manipulation, confirming the hypothesis that Receiver ignores messages. We conclude that in this case there is no apparent information minimization pressure on Sender simply because there is no communication. This experiment is reported in Supplementary.

We further consider the effect of various hyperparameters. In Figure 1b, we split the results obtained with Gumbel-Softmax by relaxation temperature. As discussed in Section 2.2, lower temperatures more closely approximate discrete communication, hence providing a convenient control of the level of discreteness imposed at training time (recall that at test time we select the symbol greedily from the softmax layer of Sender in all cases). The figure shows that lower temperatures consistently lead to lower \(I_{GS}(\(i_s; m\))\) values. This implies that, as we increase the “level of discreteness” at training-time, we get stronger information minimization pressures.

Similarly, in Figures 1c & 1d, we report mutual information when training with Stochastic Graph optimization and REINFORCE across degrees of entropy regularization. We report curves corresponding to \(\lambda\) values which converged in more than three configurations. For REINFORCE, we see that there is a weak tendency for a higher \(\lambda\) to cause a higher information content of the protocol (only violated by \(\lambda = 0.5\)).

Image Classification As the models are more complex, we only had consistent success when training with Gumbel-Softmax. In contrast to Guess Number, it is hard to estimate the lower bound on the information Sender has to convey to allow success in all configurations. The upper bound is 10 bits (bounded by the vocabulary size). The entropy of the labels \(H(I)\) is 3.3 bits, same as \(I(\(i_s; I\))\).

In Figure 2a, we aggregate all successful runs. As before, we observe that the information encoded by the protocol only grows when the Receiver’s own input becomes less informative. In the configuration where Receiver has no input, message entropy is not significantly higher than 3.3 bits, compatible with the “simplest protocol” scenario exemplified in Section 1, where Sender is directly passing image labels. In Figure 2b, we split the runs by temperature. As in Guess Number, lower temperatures increase information minimization pressure.

Summarizing, when communicating through a discrete channel, there is consistent pressure for the emergent protocol to encode as little information about the input as necessary. This holds across games, training methods and hyperparameters. Moreover, when training with Gumbel-Softmax, temperature controls the strength of the minimization pressure, confirming the relation between the latter and discreteness.
3.2 Robustness of discrete channel

Some recent approaches, inspired by the Information Bottleneck of [Tishby et al., 1999], try to control the amount of the information about the input that is stored in a representation [Achille and Soatto, 2018]. Experiment shows that this kind of regularization is beneficial, leading to robustness to overfitting [Fischer, 2019] and to adversarial attacks [Fischer, 2019; Alemi et al., 2016]. We demonstrate here that our relaxed emergent discrete protocols also possess these properties.

We focus on the Image Classification game. We use the same architecture as above, but Receiver has no input apart from the messages and, consequently, no “vision” module. The agents are trained with Gumbel-Softmax relaxation. However, for practicality, we do not switch to fully discrete communication at test time. We only remove the noise at test-time, effectively reducing Sender’s output to softmax with temperature. We refer to this architecture as GS.

We consider two baseline architectures without relaxed discrete channel. In Linear, the fully connected output layer of Sender is directly connected to the linear embedding input of Receiver. Softmax (SM) places a softmax activation (with temperature) after the Sender’s output layer and passes the result to Receiver. At test time, SM coincides with GS with the same temperature.

Learning in presence of random labels Following [Zhang et al., 2016], we study how successful the agents are in learning to classify MNIST images in presence of randomly-shuffled training examples (the test set is untouched). We vary temperature and amount of training examples with shuffled labels.
We interpret the results as follows. For “successful” overfitting, the agents need to coordinate 
(a) Robustness vs. temp. \( \tau \).

(b) Comparison to the baselines.

Figure 4: Robustness to adversarial examples: higher accuracy given fixed \( \epsilon \) implies more robustness.

labels. We use temperatures \( \tau = 1.0 \) and \( \tau = 10.0 \), having checked that the agents trained with these 
temperatures reach a test accuracy of 0.98 when trained on the original training set.

In Figure 3a we report training accuracy when all labels are shuffled. Linear and SM with \( \tau = 1.0 \) fit 
the random labels almost perfectly within the first 150 epochs. With \( \tau = 10.0 \), GS and SM achieve 
the accuracy of 0.8 in 150 epochs. When GS with \( \tau = 1.0 \) is considered, the agents only start to 

improve over random guessing after 150 epochs, and accuracy is well below 0.2 after 200 epochs. 
As expected, test set performance is at chance level (Figure 3b). In the next experiment, we shuffle 
labels for a randomly selected half of the training instances. Train and test accuracies are shown 
in Figures 3c and 3d respectively. All models initially fit the true-label examples (train accuracy 
\( \approx 0.5 \), test accuracy \( \approx 0.97 \) ). With more training, the baselines (e.g. Linear) and GS with \( \tau = 10.0 \) 
start fitting the randomly labeled examples, too: train accuracy grows, while test accuracy falls. In 
contrast, GS with \( \tau = 1.0 \) does not fit random labels in 200 epochs, and its test accuracy stays high.

We interpret the results as follows. For “successful” overfitting, the agents need to coordinate 
label memorization. This requires passing large amounts of information through the channel. With 
low temperature (more closely approximating a discrete channel), this is hard, due to the stronger 
information minimization pressure. To support the hypothesis, we ran an experiment where one fully 
connected layer of size 400x400 is either added to Sender (just before the channel) or to Receiver 
(just after the channel). We predict that, with the higher \( \tau \) (less pressure), the training curves will be 
very close, as in both cases the capacity can be used for memorization equally easy. With lower \( \tau \) 
(more pressure), the curves would be more distant. Figures 3e & 3f borne out the prediction. Finally, 
on comparing GS and SM in the above experiments, we conclude that the training-time discretization 
noise in GS is instrumental for the information minimization behavior that we observe.

Adversarial attacks To study the robustness of our architectures to undirected adversarial attacks, 
we train them with different random seeds and implement white-box attacks on the trained models, 
keeping temperature \( \tau \) and the allowed perturbation norm, \( \epsilon \). We use the standard Fast Gradient Sign 
Method (FGSM) of [Goodfellow et al., 2014]. The original image \( i_s \) is perturbed to \( i_s^* \) along the 
direction that maximizes the loss of Receiver’s output \( o = R(\hat{s}i_s) \) w.r.t. ground-truth class \( l \):

\[
i_{s}^{*} = \text{clip} [i_{s} + \epsilon \cdot \text{sign} \left( \nabla_{i_{s}} \mathcal{L}(o, I) \right), 0, 1]
\]  

(7)

where \( \epsilon \) controls the \( L_{\infty} \) norm of the perturbation. Under an attack with a fixed \( \epsilon \), a more robust 
method would have a smaller accuracy drop. To avoid the numerical stability issues akin to those 
reported by [Carlini and Wagner, 2016], all computations are done in 64-bit floats. Figure 4a shows 
that, as the relaxation temperature decreases, the accuracy drop also decreases. The highest robustness 
is achieved with \( \tau = 0.1 \). Comparison with the baselines (Figure 4b) confirms that relaxed discrete 
training with \( \tau = 0.1 \) improves robustness. However, this robustness comes at the cost of harder 
training: 2 out of 5 random seeds did not reach a high performance level (0.98) after 250 epochs.

4 Related Work

Recent literature on emergent language analysis concentrates on making sense of individual utterances, 
linking them to higher-level concepts (e.g., ImageNet categories) and studying properties such as 
their compositionality [e.g., Lazaridou et al., 2016, Kottur et al., 2017, Havrylov and Titov, 2017].
Lazaridou et al. [2018], Bouchacourt and Baroni [2018]. Our work is rooted in this line of research; but, in contrast, we focus on the information-theoretic properties of the learned protocols. Evtimova et al. [2018] uses information theory to study the progress of the protocol during learning.

Discrete latent representations are studied in many places [e.g., Jang et al. [2016], van den Oord et al. [2017], Rolfe [2016]]. However, these works focus on ways to learn discrete representations, rather than analyzing the properties of representations that are independently emerging on the side. Other studies, inspired by the Informational Bottleneck method of Tishby et al. [1999], control the complexity of neural-network-induced representations by regulating their information content [Achille and Soatto [2018], Fischer [2019], Alemi et al. [2016], Strouse and Schwab [2017]]. While they externally impose an information bottleneck, we observe that information minimization is an intrinsic feature in learning to communicate with a discrete channel.

Finally, Tieleman et al. [2018] show that communication-based learning helps to build robust representations. They use community-based autoencoders with a continuous latent space, where random pairs of encoders and decoders ‘talk’ to each other.

5 Discussion and future work

We have shown in controlled experiments that, when two neural networks learn to solve a task jointly through a discrete communication code, the latter stays close to the lower bound on the amount of information that needs to be passed through for task success. We further presented empirical evidence that this information minimization property is beneficial, as the communication protocol naturally acts as a useful bottleneck. In particular, imposing progressively more discreteness on the communication channel leads to robustness to noise and to adversarial attacks.

The practical motivation that is usually given for discrete communication protocols is that we eventually want intelligent machines to be able to communicate with humans, and the latter use a discrete protocol. Indeed, discrete messages are not required in multi-agent scenarios where no human in the loop is foreseen [Sukhbaatar et al. [2016]]. Our experiments suggest that, long before agent communication reaches the level of complexity that would allow AIs to have conversations with people, there are independent reasons to encourage discreteness, as it provides a source of robustness in a noisy world. Future applied work should test, in more general settings, discrete communication as a form of representation learning. On the other hand, if the goal is to develop an advanced communication protocol to mimic human language, it is important to design complex tasks that require a large amount of information to be exchanged, as the agents will converge to the simplest protocol they can get away with.

The regularization properties of the emergent discrete channel could also have shaped human language evolution. The discrete nature of the latter is often traced back to the fact that it allows us to produce an infinite number of expressions by combining a finite set of primitives [e.g., Berwick and Chomsky [2016]]. However, it is far from clear that the need to communicate an infinite number of concepts could have provided the initial pressure to develop a discrete code. More probably, once such code independently emerged, it made it possible to develop an infinitely expressive language [Bickerton [2014], Collier et al. [2014]]. Our work suggests that discrete coding is advantageous already when communication is about a limited number of concepts and just one symbol is transmitted at a time. We would like next to compare performance of human subjects equipped with novel continuous vs. discrete non-linguistic communication protocols, adopting the methods of experimental semiotics [Galantucci [2009]]. We expect discrete protocols to favor generalization and robustness.

Finally, information minimization trends have also been observed in many natural language phenomena, ranging from an organization of color terms that minimizes average bit transmission to the avoidance of redundant coding of syntactic information [Gibson et al. [2019]]. Many have sought a functional explanation for this tendency, as one motivated by the need to minimize speakers’ effort [e.g., Zipf [1949], Futrell et al. [2015], Fedzechkina et al. [2017], Mahowald et al. [2018]]. We discovered the same trend in neural network agents that solve a task jointly through a discrete communication channel. One could see this as a challenge to functional explanations of analogous phenomena in language. Perhaps, information minimization is a general property of emerging discrete-channel communication, due to some yet-to-be understood mathematical properties of the setup. Alternatively, information minimization in neural networks might also be due to least-effort factors. We will pursue this topic in future studies.
References

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.

Maxime Chevailler-Boisvert, Dzmitry Bahdanau, Salem Lahlou, Lucas Willems, Chitwan Saharia, Thien Huu Nguyen, and Yoshua Bengio. BabyAI: A platform to study the sample efficiency of grounded language learning. In ICLR, 2019.

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

Satwik Kottur, José MF Moura, Stefan Lee, and Dhruv Batra. Natural language does not emerge ‘naturally’ in multi-agent dialog. arXiv preprint arXiv:1706.08502, 2017.

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

Edward Choi, Angeliki Lazaridou, and Nando de Freitas. Compositional obverter communication learning from raw visual input. arXiv preprint arXiv:1804.02341, 2018.

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.

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

James Hurford. The Origins of Language. Oxford University Press, Oxford, UK, 2014.

Laura Graesser, Kyunghyun Cho, and Douwe Kiela. Emergent linguistic phenomena in multi-agent communication games. https://arxiv.org/abs/1901.08706, 2019.

David Lewis. Convention harvard university press. Cambridge, MA, 1969.

Diane Bouchacourt and Marco Baroni. How agents see things: On visual representations in an emergent language game. In EMNLP, 2018.

Thomas M Cover and Joy A Thomas. Elements of Information Theory. John Wiley & Sons, 2012.

N. Tishby, F. Pereira, and W. Bialek. The information bottleneck method. In Proceedings of the 37th Annual Allerton Conference on Communication, Control and Computing. University of Illinois Press, 1999.

Yann LeCun, Léon Bottou, Yoshua Bengio, Patrick Haffner, et al. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11):2278–2324, 1998.

Yann LeCun, Bernhard E Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne E Hubbard, and Lawrence D Jackel. Handwritten digit recognition with a back-propagation network. In NIPS, 1990.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014.

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.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with Gumbel-Softmax. arXiv preprint arXiv:1611.01144, 2016.

Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. Machine learning, 8(3-4):229–256, 1992.

Evan Greensmith, Peter L Bartlett, and Jonathan Baxter. Variance reduction techniques for gradient estimates in reinforcement learning. JMLR, 5(Nov):1471–1530, 2004.

Ronald J Williams and Jing Peng. Function optimization using connectionist reinforcement learning algorithms. Connection Science, 3(3):241–268, 1991.

Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In ICML, 2016.

John Schulman, Nicolas Heess, Theophane Weber, and Pieter Abbeel. Gradient estimation using stochastic computation graphs. In NIPS, 2015.
Alessandro Achille and Stefano Soatto. Information dropout: Learning optimal representations through noisy computation. *IEEE TPAMI*, 40(12):2897–2905, 2018.

Ian Fischer. The conditional entropy bottleneck, 2019. URL https://openreview.net/forum?id=\texttt{rkVOXhAqY7}

Alexander A Alemi, Ian Fischer, Joshua V Dillon, and Kevin Murphy. Deep variational information bottleneck. *arXiv preprint arXiv:1612.00410*, 2016.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*, 2016.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

Nicholas Carlini and David Wagner. Defensive distillation is not robust to adversarial examples. *arXiv preprint arXiv:1607.04311*, 2016.

Katrina Evtimova, Andrew Drozdov, Douwe Kiela, and Kyunghyun Cho. Emergent communication in a multi-modal, multi-step referential game. In *ICLR*, 2018.

Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *NIPS*, 2017.

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

DJ Strouse and David J Schwab. The deterministic information bottleneck. *Neural computation*, 29(6):1611–1630, 2017.

Olivier Tieuleman, Angeliki Lazaridou, Shibl Mourad, Charles Blundell, and Doina Precup. Shaping representations through communication. 2018.

Sainbayar Sukhbaatar, Arthur Szlam, and Rob Fergus. Learning multiagent communication with backpropagation. In *NIPS*. 2016.

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

Derek Bickerton. *More than Nature Needs: Language, Mind, and Evolution*. Harvard University Press, Cambridge, MA, 2014.

Katie Collier, Balthasar Bickel, Carel van Schaik, Marta Manser, and Simon Townsend. Language evolution: Syntax before phonology? *Proceedings of the Royal Society B: Biological Sciences*, 281(1788):1–7, 2014.

Bruno Galantucci. Experimental semiotics: A new approach for studying communication as a form of joint action. *Topics in Cognitive Science*, 1(2):393–410, 2009.

Edward Gibson, Richard Futrell Steven Piantadosi, Isabelle Dautriche, Kyle Mahowald, Leon Bergen, and Roger Levy. How efficiency shapes human language. *Trends in Cognitive Science*, 2019. In press.

George Zipf. *Human Behavior and the Principle of Least Effort*. Addison-Wesley, Boston, MA, 1949.

Richard Futrell, Kyle Mahowald, and Edward Gibson. Large-scale evidence of dependency length minimization in 37 languages. *Proceedings of the National Academy of Sciences*, 112(33):10336–10341, 2015.

Maryia Fedzechkina, Elissa Newport, and Florian Jaeger. Balancing effort and information transmission during language acquisition: Evidence from word order and case marking. *Cognitive Science*, 41:416–446, 2017.

Kyle Mahowald, Isabelle Dautriche, Edward Gibson, and Steven Piantadosi. Word forms are structured for efficient use. *Cognitive Science*, 42:3116–3134, 2018.
6 Supplementary

6.1 Hyperparameters

In our experiments, we used the following hyperparameter grids.

**Guess Number (Gumbel-Softmax)** Vocab. size: [256, 1024, 4096]; temperature, \( \tau \): [0.5, 0.75, 1.0, 1.25, 1.5]; learning rate: [0.001, 0.0001]; max. number of epochs: 250; random seeds: [0, 1, 2, 3]; batch size: 8; early stopping thr.: 0.99; bits shown to Receiver: [0, 1, 2, 3, 4, 5, 6, 7, 8].

**Guess Number (REINFORCE)** Vocab. size: [256, 1024, 4096]; Sender entropy regularization coef., \( \lambda_s \): [0.01, 0.05, 0.025, 0.1, 0.5, 1.0]; Receiver entropy regularization coef., \( \lambda_r \): [0.01, 0.1, 0.5, 1.0]; learning rate: [0.0001, 0.001]; max. number of epochs: 1000; random seeds: [0, 1, 2, 3]; batch size: 2048; early stopping thr.: 0.99; bits shown to Receiver: [0, 1, 2, 3, 4, 5, 6, 7, 8].

**Guess Number (Stochastic Computation Graph approach)** Vocab. size: [256, 1024, 4096]; Sender entropy regularization coef., \( \lambda_s \): [0.01, 0.025, 0.05, 0.075, 0.1, 0.25]; learning rate: [0.0001, 0.001]; max. number of epochs: 1000; random seeds: [0, 1, 2, 3]; batch size: 2048; early stopping thr.: 0.99; bits shown to Receiver: [0, 1, 2, 3, 4, 5, 6, 7, 8].

**Image Classification experiments** Vocab. size: [16, 64, 256, 1024, 4096]; temperature, \( \tau \): [0.25, 0.5, 0.75, 1.0, 1.25, 1.5, 2.0]; learning rate: [0.01, 0.001, 0.0001], max. number of epochs: 100; random seeds: [0, 1, 2, 3]; batch size: 32; early stopping thr.: 0.98; rows shown to Receiver: [0, 4, 8, 12, 16, 20, 24, 28].

**Fitting random labels experiments** Vocab. size: 1024; temperature, \( \tau \): [1.0, 10.0]; learning rate: 1e-4, max. number of epochs: 200; random seeds: [0, 1, 2, 3, 4]; batch size: 32; early stopping thr.: \( \infty \); prob. of label corruption: [0.0, 0.5, 1.0].

**Adversarial attack experiments** Vocab. size: 1024; temperature, \( \tau \): [0.1, 1.0, 10.0]; learning rate: 1e-4, max. number of epochs: 200; random seeds: [0, 1, 2, 3, 4]; batch size: 32; early stopping thr.: 0.98.

6.2 Does vocabulary size affect the results?

We repeat the same experiments as in Section 3 of the main text while varying vocabulary size. Note that, to make Guess Number solvable across each configuration, the vocabulary has to contain at least 256 symbols. Similarly, for Image Classification, vocabulary size must be of at least 10. We tried vocabulary sizes of 256, 1024, 4096 for Guess Number, and 16, 64, 256, 1024, 4096 for Image Classification. The results are reported in Figures 5 (Guess Number) and 6 (Image Classification). We observe that there is little qualitative variation over vocabulary size, hence the conclusions we had in Section 3 are robust to variations of this parameter.

6.3 Communication analysis

6.3.1 How much does Receiver rely on messages?

We supplement the experiments of Section 3 of the main text by studying the degree to which Receiver relies on messages. In particular, we show that: (a) when Receiver has the full input \( (i_s = i_r) \), it ignores the messages (b) in Image Classification, a higher relaxation temperature not only results in more informative messages, but also into the Receiver more strongly depending on them.

We measure the degree to which Receiver relies on the messages from Sender by constructing a setup where we break communication, but still let Receiver rely on its own input. More precisely, we first enumerate all test inputs for Sender \( i_s \) and Receiver \( i_r \). We obtain messages that correspond to Sender’s inputs, and shuffle them. Next, we feed the shuffled messages alongside Receiver’s own (unshuffled) inputs and compute accuracy, as a measure of Receiver’s dependence on the messages. This procedure preserves the marginal distribution of messages Receiver receives, but destroys all the information Sender transmits.

**Guess Number** It is clear that, without the messages, Receiver with \( k \) bits of input can only reach accuracy of \( 2^{k-k} \). In Figure 7a, we report results aggregated by training method. Receiver is extremely close to the accuracy's higher bound in all configurations. Moreover, when Receiver gets the entire input, the accuracy drop after the shuffling is tiny, hence its reliance on the message is minimal.

**Image Classification** In contrast to Guess Number, it is hard to find an upper bound on performance of Receiver without messages analytically. To get an estimate, we train a Sender/Receiver pair where Sender has all its input zeroed. In Figure 7b, we report the results of the shuffling experiments, split by the relaxation temperature \( \tau \) used during training. We first observe that, again, when Receiver has the entire image as its input, its reliance on messages is small. Further, for all configurations, we see that as the relaxation temperatures are getting lower, so does Receiver’s reliance on the messages.
6.3.2 Visual prototypes of Image Classification messages

To visualize the emerging representations associated to messages, we run the following experiment. For each ground-truth class, we iterate over test set images and find the message which is most likely associated with the class.\footnote{In all cases, the distribution of messages for a fixed class is extremely peaky.} Next, for each of these messages, we generate an image that maximizes the probability that it will be the one being sent. For that, we start with a random image and optimize its pixels while the model is fixed.\footnote{We use Adam [Kingma and Ba, 2014], learning rate 1e-3, 5000 steps.}

Figure 5: Guess Number: Mutual information between Sender’s input $i_s$ and protocol messages $m$, depending on vocabulary size, training method, and relaxation temperature $\tau$ (when trained with Gumbel-Softmax) or Sender’s entropy regularization coefficient $\lambda_s$. Shaded regions mark standard deviation.

\begin{figure}
\centering
\includegraphics[width=\textwidth]{figure5.pdf}
\caption{Guess Number: Mutual information between Sender’s input $i_s$ and protocol messages $m$, depending on vocabulary size, training method, and relaxation temperature $\tau$ (when trained with Gumbel-Softmax) or Sender’s entropy regularization coefficient $\lambda_s$. Shaded regions mark standard deviation.}
\end{figure}
Figure 6: Image Classification: mutual information between Sender’s input $i_s$ and protocol messages $m$ across vocabulary sizes. Successful runs are pooled together. Shaded regions mark standard deviation.

Figure 7: Receiver’s dependence on messages, measured as performance drop under the message intervention.

Figure 8: Visualizing Image Classification. Top (bottom) row corresponds to Sender trained with $\tau = 0.1$ ($\tau = 10.0$). See Section [8](#) for details.