Trading off Utility, Informativeness, and Complexity in Emergent Communication

Mycal Tucker
M.I.T.
mycal@mit.edu

Roger Levy
M.I.T.
rplevy@mit.edu

Julie Shah
M.I.T.
julie_a_shah@csail.mit.edu

Noga Zaslavsky
M.I.T.
nogazs@mit.edu

Abstract

Emergent communication (EC) research often focuses on optimizing task-specific utility as a driver for communication. However, there is increasing evidence that human languages are shaped by task-general communicative constraints and evolve under pressure to optimize the Information Bottleneck (IB) tradeoff between the informativeness and complexity of the lexicon. Here, we integrate these two approaches by trading off utility, informativeness, and complexity in EC. To this end, we propose Vector-Quantized Variational Information Bottleneck (VQ-VIB), a method for training neural agents to encode inputs into discrete signals embedded in a continuous space. We evaluate our approach in multi-agent reinforcement learning settings and in color reference games and show that: (1) VQ-VIB agents can continuously adapt to changing communicative needs and, in the color domain, align with human languages; (2) the emergent VQ-VIB embedding spaces capture semantically meaningful structures; and (3) encouraging informativeness leads to faster convergence rates and improved utility, both in VQ-VIB and in prior neural architectures for symbolic EC, with VQ-VIB achieving higher utility for any given complexity. This work offers a new framework for EC that is grounded in information-theoretic principles that are believed to characterize human language evolution and that may facilitate human-agent interaction.

1 Introduction

Good communication is a critical component of successful teams of humans and artificial agents, but differing notions of "good" makes training agents to develop such communication challenging. One view of communication focuses on *utility* by framing languages as successful to the extent they enable high task performance. Emergent communication (EC) literature, wherein agents learn to communicate while optimizing the utility associated with a task-specific goal, such as reaching a target landmark, emphasizes this view [1–6]. However, training via utility alone may lead to undesirable properties of the EC system, such as violations of linguistic universals [7], poor human interpretability [8], slow convergence rates [9], or high sensitivity to environment design [10]. This implies that some constraints may be needed to guide agents toward human-like communication.

Another view of communication, often considered in the cognitive science literature, focuses on task-agnostic communicative objectives as major forces shaping human languages [11, 12]. Most relevant to our work is a recent line of work [13–17] that argues that languages evolve under pressure to efficiently compress meanings into communicative signals by optimizing the Information Bottleneck (IB) principle [18]. In this context, IB can be interpreted as a tradeoff between the

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

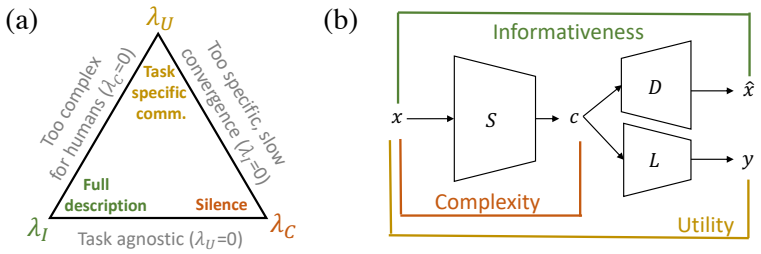


Figure 1: (a) In our theoretical framework, agents evolve to communicate with each other by optimizing a tradeoff between utility, informativeness, and complexity, weighted by λ_U, λ_I, and λ_C respectively. (b) Simplified communication model and agent architecture. A speaker, S, maps an input, x, to a signal, c. A listener, L, takes an action, y, and reconstructs x using a decoder D.

complexity and informativeness of the lexicon [13]. Intuitively, complexity corresponds to the number of bits allocated for communication, and informativeness corresponds to how well a listener can understand a speaker regardless of any task-specific goal (e.g., humans understand the term "blue" in "avoid the blue box" and "find a blue cup"). This information-theoretic framework for grounded semantic categories has been gaining broad empirical support across hundreds of languages and multiple semantic domains [13–16], and has been directly tied to language change over time [17]. However, the process of inducing such of IB-efficient communication systems among artificial agents is less clear. While it has been shown that such systems may emerge in a color reference game via utility maximization [10], language variation in that study is implicitly controlled by environmental factors, leaving open the question of how efficient communication may evolve in a given environment and agent population, and in the context of tasks other than reference games.

In this work, we integrate these two views of communication by training teams of interactive artificial agents to optimize a tradeoff between maximizing utility, maximizing informativeness, and minimizing complexity, as depicted in Figure 1. We hypothesize that taking into account complexity and informativeness, which are cognitively-motivated task-general communicative constraints, in addition to task-specific utility, may help guide artificial agents toward communication systems that can be more naturally aligned with and adapted to human languages. To this end, building on prior work [19–21], we propose Vector-Quantized Variational Information Bottleneck (VQ-VIB), a method for training neural agents to encode inputs into signals, while maintaining a discrete (symbolic) signal representation that is embedded in a continuous space, inspired by word embeddings [22, 23]. We evaluate our approach in two multi-agent reinforcement learning environments and in two variants of a color reference game. We show that: (1) by annealing the complexity weight, VQ-VIB agents can continuously adapt to changing communicative pressures and, in the color domain, align with human languages without changing the agents' architecture or the structure of the environment; (2) the emergent VQ-VIB embedding spaces capture semantically meaningful structures, which are grounded in the agents' embodiment; and (3) encouraging agents to be informative generally leads to faster convergence rates and improved utility, with VQ-VIB achieving higher utility for a given complexity compared to prior neural architectures that support symbolic communication. Taken together, this work offers a new information-theoretic framework for EC that is grounded in general optimization principles that are believed to characterize human language evolution, and that has the potential of improving human-agent interaction.

2 Related Work

In this work, we train neural agents in cooperative settings, focusing on the effects of utility, informativeness, and complexity in three EC environments. Below we review prior work that considers related measures and test environments in EC, as well as in human languages.

Communicative constraints in emergent communication While traditional EC literature focuses on task utility, some studies also consider communicative constraints that are closely related to our measures of complexity and informativeness (defined in Section 3). For example, numerous works consider discretization, environmental noise, or penalty on large vocabulary sizes as proxies for reducing complexity [21, 24, 25]. Wang et al. [26] use an information bottleneck in a centralized

message processor to explicitly limit complexity. Our notion of complexity is grounded in rate-distortion theory [27], capturing the degree of semantic compression seen across languages [13]. Other notions of linguistic complexity often focus on grammar, word forms, or compositionality, and offer promising directions for future work [28–31]. Additionally, some works consider auxiliary reconstruction losses designed to create "meaningful" communication and improve convergence rates [9, 32]. In particular, Lin et al. [33] use an autoencoding loss which is closely related our notion of informativeness. However, because they do not penalize complexity, they note that their method "may fail... where agents need to be more selective of the information communicated." Lastly, several studies consider reinforcement learning settings with informational constraints similar to ours [e.g., 34–38], however not in the context of EC as we do here.

Emergent communication environments and color naming We explore particle-world environments that agents must navigate, as in [3, 21], as well as reference games (related to Lewis signalling games [39]), both of which are popular types of environments for evaluating EC. While reference games may be played with varied inputs, we focus here on the color domain for several reasons. First, it has been a central test-case for cognitive theories of semantic categorization and language evolution [1, 40, 41]. Second, it enables comparison with the World Color Survey [WCS; 42], a dataset of color naming systems from 110 languages of non-industrialized societies. Third, it has been a focus of numerous studies in EC and machine learning more generally [1, 43–45, 10]. In particular, Chaabouni et al. [10] use the IB framework from [13] to evaluate the emergent color signaling systems in artificial agents, and found similarities to human languages. However, to capture the diversity observed across languages, they vary characteristics of the training environment by selecting dissimilar distractor colors, or by adding noise to the agents' communication. Rather than focusing on environment-based factors, we adopt a principled approach by directly optimizing theoretically- and cognitively-motivated terms of complexity and informativeness.

3 Information-Theoretic Emergent Communication

Here, we propose an information-theoretic EC framework that integrates task-general communicative objectives with task-specific utility maximization. We begin with reviewing the theoretical framework on which we build (Section 3.1.1) and the set of variational inference tools that we use in order to extend that framework to EC settings (Sections 3.1.2 and 3.1.3). We then define our integrative objective function for training EC agents (Section 3.2) and present Vector-Quantized Variational Information Bottleneck (VQ-VIB, Section 3.3), a method for training agents with respect to our objective function while maintaining symbolic representations that can be used for communication. The experimental setup and results for evaluating this new method are presented in Sections 4 and 5.

3.1 Preliminaries

3.1.1 The Information Bottleneck framework for grounded semantic systems

We build on and extend Zaslavsky et al. [13]'s information-theoretic framework for grounded semantic systems. In this framework, speakers and listeners jointly optimize the IB tradeoff [18] between the complexity and informativeness of their communication system. A speaker is characterized as a probabilistic encoder $S(c|x)$ that, given an input, $x \sim p(x)$, mentally represents it as m_x and based on that generates a communication signal, c. m_x is the speaker's belief state, which is grounded in the agents' representation of the world and captures any uncertainly the speaker may have about the input. More formally, it is defined by a probability distribution over a set of possible inputs or relevant features that characterize the input space $\mathcal{U}$, i.e, $m_x(u) = p(u|m_x)$. For example, x may be a target color and m_x a Gaussian distribution over the CIELAB perceptual color space (Figure 3c), capturing the speaker's perceptual noise. Given c, the listener constructs $\hat{m}_c$ to infer the speaker's belief state.

Complexity is measured by the mutual information between the speaker's inputs and signals, $I_S(x, c)$, which, as in rate-distortion theory [27, 46], is roughly the number of bits that are needed for communication [18]. Informativeness corresponds to the similarity between the speaker's and listener's belief states. Because these states are defined by probability distributions, maximizing informativeness amounts to minimizing the expected Kullback-Leibler (KL) divergence $\mathbb{E}_S[D[m_x \| \hat{m}_c]]$. Taken together, in this view, for any given tradeoff parameter $\beta \geq 0$ the speaker and listener should jointly

$$\underset{S(c|x), \hat{m}_c}{\text{minimize}} \quad I_S(x, c) - \beta \mathbb{E}_S[D[m_x \| \hat{m}_c]] \,, \tag{1}$$

which is equivalent to optimizing the IB principle [see 13, 47, for derivation]. For $\beta \leq 1$, the optimal systems are non-informative and minimally complex ($I_S(x, c) = 0$), yielding no communication. Gradually annealing β yields optimal systems with greater complexity, and as $\beta \to \infty$, the optimal systems become maximally informative but also maximally complex. Zaslavsky et al. [13] and colleagues have shown that human languages achieve near-optimal IB tradeoffs across multiple semantic domains, and that the continuous evolution of the optimal IB systems via annealing the tradeoff parameter β [48] captures important aspects of language change over time [13, 17].

3.1.2 Variational Information Bottleneck

Variational Information Bottleneck [VIB, 19] is a variational approximation method for IB with continuous representations, using a neural network parameterization and the reparameterization trick [49]. In its typical form, a neural encoder $q_\theta(z|x)$ maps an input x to parameters of a d-dimensional Gaussian distribution, $\mu(x)$ and $\Sigma(x)$, from which a latent representation, $z \in \mathbb{R}^d$, is sampled. Given z, VIB reconstructs a target feature u using $q_\phi(u|z)$, parameterized by a soft-max function. VIB optimizes a variational bound on the IB objective function, based on the fact that

$$I_{q_\theta}(x; z) \leq \mathbb{E}\left[D[q_\theta(z|x) \| r(z)]\right] \tag{2}$$

for any distribution $r(z)$, which is often taken to be $\mathcal{N}(0, I_d)$. VIB also bounds the IB informativeness term. However, because we consider here a different instantiation of IB, based on unsupervised rather than supervised learning, we use a different informativeness bound, as explained in Section 3.3.

3.1.3 Vector-Quantized Variational Autoencoder

While VIB learns continuous representations, Vector-Quantized Variational Autoencoder [VQ-VAE, 20] learns discrete, but potentially highly complex, representations in an unsupervised manner. More specifically, VQ-VAE defines a latent embedding space of K embedding vectors $\zeta_i \in \mathbb{R}^d, i = 1, \ldots, K$. An encoder maps an input x to a discrete variable $k \in \{1, \ldots, K\}$ by first passing x through a deterministic network to obtain a latent representation $z(x) \in \mathbb{R}^d$, and then discretizing it to the index of the nearest embedding vector, i.e., $k = \text{argmin}_i \|z(x) - \zeta_i\|_2$. The decoder receives the corresponding embedding vector, $\zeta_k(x)$, and passes it through another deterministic network to reconstruct the input. The set of trainable parameters Θ consists of the latent embedding space, $\{\zeta_1, \ldots, \zeta_K\}$, and the parameters of the encoder and decoder networks. Because gradients cannot be passed through the `argmin` operator, a straight-through estimator [50] method is used, where the gradients from the decoder's input, $\zeta_k(x)$, are copied to the encoder's output $z(x)$. Similar to VAE [49], VQ-VAE's training loss, shown in Equation (3), uses the evidence lower bound (ELBO) on the log-likelihood of the model, $\log p(x; \Theta)$. However, because the encoder is deterministic and the prior over k is taken to be uniform, the ELBO is reduced to the reconstruction loss (first term in Equation (3)). VQ-VAE adds to it two VQ terms, which are designed for learning the embedding space, yielding the overall training loss

$$L_{\text{VQ-VAE}} = \log p(x|\zeta_k(x); \Theta) + \|\texttt{sg}[z(x)] - \zeta_k(x)\|_2^2 + \alpha \|z(x) - \texttt{sg}[\zeta_k(x)]\|_2^2 \,, \tag{3}$$

where `sg` stands for the `stopgradient` operator, and following [20], $\alpha = 0.25$.

3.2 Integrating utility, informativeness, and complexity

In traditional EC settings, agents are trained to maximize the expected reward, or utility [51]. In a simplified setting, we assume that a speaker, S maps from inputs, x, to communication c, while a listener, L, maps from c to an output y. Utility-based approaches to EC aim to find S, L that maximize the expected $U(x, y)$. Similar utility-based framings apply to more complex settings with discounted rewards, more agents, or agents that can communicate and act.

One may integrate utility with the IB framework into an idealized objective function we seek to maximize, shown in Equation 4. This objective function is the weighted combination of three terms, wherein we seek to maximize a tradeoff between high utility, high informativeness (low KL divergence between m_x and $\hat{m}_x$), and low complexity. Intuitively, these three scalar weights form a simplex (depicted in Figure 1) that describe a range of communication from uninformative communication (minimizing complexity) to task-optimized communication that does not generalize (maximizing utility) to complete descriptions (maximizing informativeness).

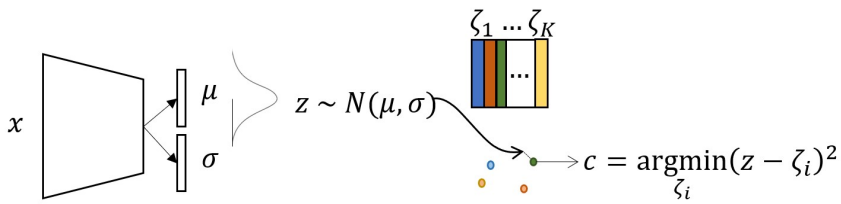


Figure 2: The VQ-VIB encoder architecture. The speaker encodes an input, x, as parameters to a Gaussian distribution, from which a vector is sampled and then discretized by mapping to the nearest encoding, to produce a communication vector, c.

$$\text{maximize} \quad \lambda_U U(x, y) - \lambda_I \mathop{\mathbb{E}}_{x \sim p(x), c \sim S(c|x)} [D[m_x \| \hat{m}_c]] - \lambda_C I_S(x, c) \tag{4}$$

Directly training to maximize this objective is intractable in larger domains, but variational methods, including one we propose in the next section, support useful approximations.

3.3 Vector-Quantized Variational Information Bottleneck

Our proposed method, Vector Quantized Variational Information Bottleneck (VQ-VIB), learns information-constrained discrete communication embedded in a continuous space, reminiscent of the notion of word embeddings [22, 23]. This method combines prior work on Variational Information Bottleneck [VIB, 19] with discrete representation learning via Vector-Quantized Variational Autoencoders [VQ-VAE, 20]. It is also closely related to prototype-based EC [21], which aims to link EC with word embeddings, although using a different objective function and architecture than VQ-VIB.

3.3.1 VQ-VIB encoder (speaker) architecture

Here, we combine VIB and VQ-VAE for a novel method of generating discrete, information-bounded representations, as depicted in Figure 2. Similar to VQ-VAE, VQ-VIB maintains a trainable codebook of K embedding vectors in $\mathbb{R}^d$, $\mathcal{C} = \{\zeta_1, \dots, \zeta_K\}$, which in our case will be used for communication. Given an input x, a neural encoder $q_\theta(z|x)$ is used to sample a latent representation $z \in \mathbb{R}^d$, followed by a quantization layer that discretizes the sample by mapping to the nearest quantized vector $\zeta \in \mathcal{C}$. This generates a communication vector $c = \text{argmin}_{\zeta_i}(z - \zeta_i)^2$. Similar to VIB, we define $q_\theta(z|x)$ by mapping x to parameters of a Gaussian distribution over the latent space. Overall, the VQ-VIB encoder, or speaker agent, is given by

$$S_\theta(c|x) = \mathbb{P}(c = \underset{\zeta_i \in \mathcal{C}}{\text{argmin}}(z - \zeta_i)^2 | x, \mathcal{C}, \theta) \tag{5}$$

We train VQ-VIB agents using Equation 6. We assume access to some utility-based loss function such as the REINFORCE training loss [52]. Informativeness is computed via the negative mean squared error between the speaker's input and a decoder's output, given communication. (MSE is related to the KL divergence between two equal-variance Gaussians). Communication complexity is bounded via the KL divergence between the speaker's distribution for z and a unit Gaussian. (Bounding $I_S(x, z)$ implicitly bounds $I_S(x, c)$ via the data processing inequality). Lastly, we reuse vector quantization losses from Equation 3. In all experiments, we set $\lambda_{\text{VQ}} = 1; \alpha = 0.25$.

$$\begin{aligned} L_{\text{VQ-VIB}} = \lambda_U L_U(x, y) + \lambda_I \mathbb{E}[(m_x - \hat{m}_x)^2] + \lambda_C \mathbb{E}[D[q_\theta(z|x) \| r(z)]] + \\ \lambda_{\text{VQ}}(\| \texttt{sg}[z(x)] - c \|_2^2 + \alpha \| z(x) - \texttt{sg}[c] \|_2^2) \end{aligned} \tag{6}$$

Thus, VQ-VIB may be thought of as generating a complexity-bounded continuous representation followed by a vector quantization process. We also considered a variant of a VQ-VIB speaker, called *VQ-VIB After*, which quantized representations and then sampled around them, but it does not perform as well as our main VQ-VIB speaker. We leave the discussion of VQ-VIB After to Appendix A.

3.4 Decoder (listener) architecture

Regardless of the speaker architecture, we trained agents in the overall network architecture depicted in Figure 1 b. A decoder, $D(\hat{x}|c)$, reconstructed the speaker's input, given c, and a listener output actions or decisions y based on communication (and any additional observations specific to the listener: $L(y|c, x_L)$. Using a separate decoder, as introduced by Lin et al. [33], allows backpropagation through the decoder and speaker even when gradients cannot flow from the listener to the speaker.

3.4.1 Complexity equivalence classes

For a given communication complexity level, there is a set of IB-optimal systems with different effective lexicon sizes $k = |\{c \in \mathcal{C} : \sum_x p(x)S(c|x) > 0\}|$ [48]. We wish to bias agents within a complexity class towards the canonical system, with the minimal k, by regularizing the entropy of the speaker's communication vectors. Regularizing the entropy in VQ-VIB is problematic given the non-differentiablity of the `argmin` operation used in discretization. Instead, we use the entropy of an approximated categorical distribution based on the euclidean distance to each quantized vector, $p_{\text{cat}}(\zeta_i|z) \propto e^{-(z-\zeta_i)^2}$. Because this entropy term is only used to break the symmetry between equally optimal systems, we use a small regularizing weight (0.05 in our experiments, which is orders of magnitude smaller than the weights we use for the other terms in our objective function).

4 Experimental Setup

In designing our experiments, we relied upon the following intuitions (confirmed in findings) about the roles of utility, informativeness, and complexity in emergent communication. Increasing informativeness should cause agents to converge to successful communication systems more quickly than without such training. However, increasing informativeness will also increase complexity, so limiting complexity will be necessary to induce human-like discrete communication systems.

Trials and baselines We compared VQ-VIB with four types of neural architectures: continuous (e.g., Sukhbaatar et al. [53]), onehot (e.g., Lowe et al. [3]), prototype-based [Proto., 21], and VQ-VIB After (defined in Appendix A). All speaker architectures were trained with the same decoder and listener architectures, and in all domains agents were trained from scratch using 5 random seeds. We strengthened the baseline architectures relative to prior art to support variational bounds on complexity. For onehot agents, we bounded complexity via the KL divergence between the speaker's categorical distribution and from a uniform distribution. Typical Proto. agents deterministically map from inputs to a onehot vector that is then multiplied by a learnable "embedding" matrix; in our variational method, we then sampled around such embeddings with variance also output by the speaker (much like a VAE sampling according to a μ and σ). These changes strictly improved baselines, as with no complexity regularization, agents could learn deterministic policies. See Appendix B for full details of the architectures, hyperparameters, and environments.

Particle worlds We train agents in two particle world environments adapted from Tucker et al. [21], 9 points and Uniform, shown in Figure 3. Both environments consist of two-dimensional worlds in which a speaker observes the exact target location that a listener needs to reach but cannot directly observe. In the 9 points environment, there were 9 landmarks at fixed locations, one of which was chosen at random as the target for each episode. In the Uniform environment, two landmarks were spawned in locations drawn uniformly at random in the world; one landmark was specified as the target. Agents received identical reward equal to the negative distance from the listener to the target and were trained using Multi-Agent Deep Deterministic Policy Gradient [3], which is a standard policy-gradient method for training agents without backpropagating through one agent into another. The listener could not observe the target location, and the speaker could only communicate at the first timestep by broadcasting a vector that was observable at all future timesteps; thus, an optimal team policy would consist of the speaker communicating about the target at the first timestep, and the listener navigating to the target based on the message. The two environments differ in their complexity: in the 9 points environment, the speaker had a finite amount of information about which to communicate, whereas in the Uniform environment, the location could be any continuous value.

Color reference games In our two color reference game experiments, we trained agents by differentiating through both the speaker and listener, as depicted in Appendix B, Figure 8. In the

(a) MARL: 9 points (b) MARL: Uniform (c) Reference game: 330 WCS colors

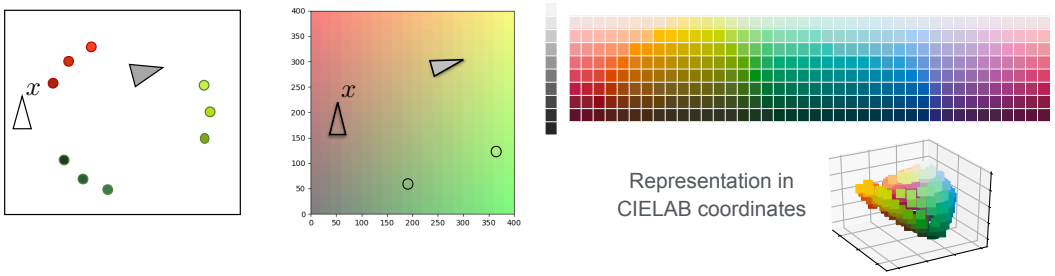


Figure 3: The three test environments. In the MARL experiments (a-b) a speaker (clear triangle) communicates a target location x to a listener (gray triangle), which needs to navigate to that location (adapted from [21]). In the color environment (c), a speaker observes a target color from the WCS palette, represented in the CIELAB perceptual color space, and communicates it to the listener, who then needs to identify the target color.

"Basic" reference game, a speaker observed a target color and output a communication vector. A listener observed the speaker's communication and the target and distractor colors and predicted which of the colors was the target. Colors were represented by their 3D CIELAB coordinates (illustrated in Figure 8 in Appendix B), and speaker observations were corrupted by zero-mean Gaussian noise with a standard deviation of 8, as motivated by human perception [13]. Both colors were sampled from the set of 330 World Color Survey (WCS) colors according to a human-based prior from [13].[1] Only the speaker agent observed which color was the target, so the agents had to learn to convey information from the speaker to the listener. Agents were trained via binary crossentropy loss computed on the listener's output. Additional experiments, wherein agents were trained with REINFORCE, achieved slightly worse results (corroborating Chaabouni et al. [10]'s findings) and are included in Appendix D [52].

In addition to the Basic setting, we also used a "Common Ground" setup which differed only by letting the speaker also observe the distractor color. Thus, while the Basic scenario resembles the WCS setting, in which participants were asked to name individual colors, the Common Ground scenario was inspired by work in pragmatics and Rational Speech Act, wherein humans communicate less information than fully describes a scene, but enough to induce listeners to behave in the desired manner [44, 54]. We note that, unlike Chaabouni et al. [10], we do not change the subset of sampled distractor colors to induce different complexity; our framework results in more stable training and allows us to directly vary communication complexity within the same environment.

5 Experiment Results

Our results indicated that an informativeness loss improved convergence, annealing λ_C generated a suite of agents of different complexities that more closely approximated human languages, and VQ-VIB agents achieved greater utility, for the same complexity, than onehot agents. [2]

5.1 Particle Worlds

In the particle worlds, increasingly penalizing complexity led agents to lower-utility communication, although architecture choices and training to increase informativeness could increase reward. First, we conducted experiments in which we did not penalize complexity, but varied λ_I across trials. Across all architectures, higher λ_I increased the mean team reward and improved convergence rates, as shown in the learning curves of VQ-VIB in Figure 5 a. We note that successfully training agents with a mix of informativeness and utility loss represents an improvement over Lin et al. [33], who struggled to train agents with a combination of losses.

[1] The WCS data are available at `www1.icsi.berkeley.edu/wcs` and the human-based color prior is available at `https://github.com/nogazs/ib-color-naming`.

[2] Code is available at `https://github.com/mycal-tucker/vqvib_neurips2022`

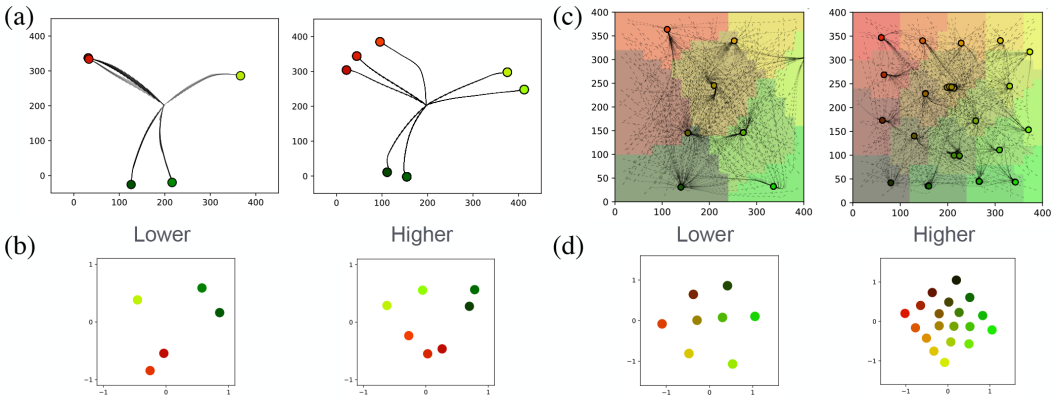


Figure 4: VQ-VIB solutions for the 9 points (a-b) and uniform (c-d) environments, with lower (left panels) and higher (right panels) complexity. Each communication vector is associated with the mean color of the positions it can refer to, based on the speaker's encoder and the colors in Figure 3. **(a, c)** Listener's trajectories (black curves) and final positions (dots) in response to a speaker's communication vector, indicated by the color of the of the final dot. Background colors in (b) show the modal communication vector for each target position in the continuous space. **(b, d)** 2D PCA of the emergent VQ-VIB communication vectors, corresponding to the systems in (a) and (c).

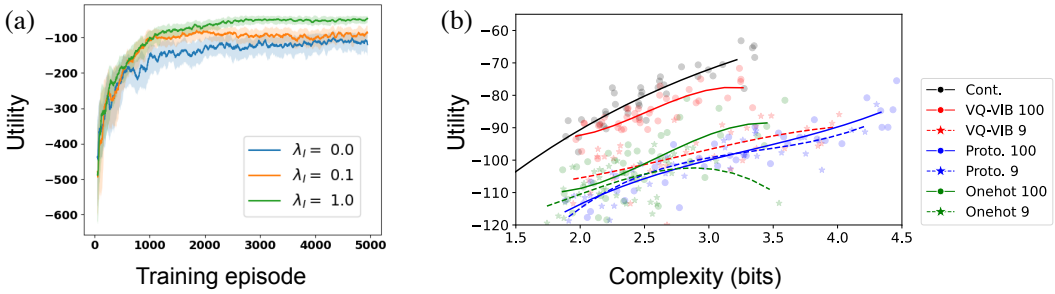


Figure 5: Uniform environment results. (a) Increasing λ_I improved convergence rates and utility during training (shown here for VQ-VIB). (b) For fixed $\lambda_I = 1.0$, annealing λ_C generated a spectrum of agents. VQ-VIB agents outperform agents based on the onehot and Proto architectures, and nearly reaches the performance of the continuous communication architecture with only 100 signals.

Next, for a fixed $\lambda_I = 1.0$ (to improve convergence), we trained agents while annealing λ_C; this smoothly decreased the complexity of communication. Results for the Uniform environment aligned with our expectations (Figure 5 b). Each curve indicates the cubic best fit to the data generated across 5 random seeds for each configuration (discarding outliers for reward less than -120), with complexity and reward measured at each increment of 1,000 training episodes from 3,000 to 10,000. As complexity decreased (moving right to left in the plot), reward decreased as well. Visualizations of agent behavior confirmed that low-complexity VQ-VIB agents communicated via increasingly coarse discretizations of the space and are included in Figure 4, and animations of the evolution of such discretizations as complexity changes are available in our codebase. The plots align with notions of how humans might communicate in natural language about a continuous space via coarse discretizations like "north" or "south" rather than hundreds of specific words for any possible location.

Lastly, these results suggest that neural architecture choice was important. In both 9 points and Uniform, the VQ-VIB agents tended to achieve higher reward, for the same complexity, as onehot or Proto agents. (Continuous communication reflected an optimal upper bound but violated our desire to learn discrete communication.) Furthermore, adding more encodings (9 discrete encodings vs. 100) significantly increased VQ-VIB performance, while changing onehot agents to communicate in 9 or 100 dimensions had a smaller effect. Parametrizing VQ-VIB or onehot agents with even more tokens (200 or 500) did not improve utility beyond 100 tokens, suggesting that greater parametrization will not decrease the gap. This section focuses on VQ-VIB in the Uniform environment; full results are in

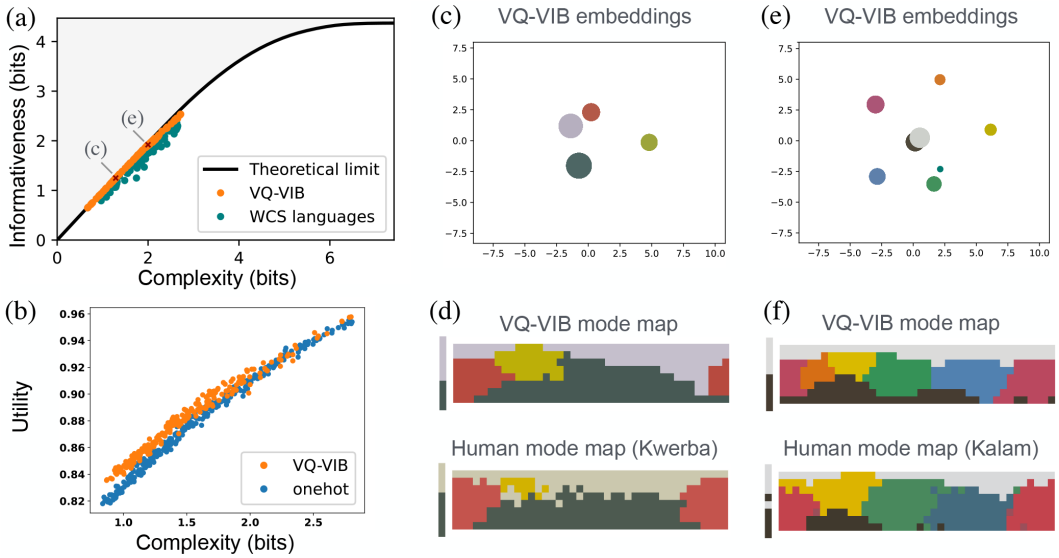


Figure 6: Color reference game results (basic setting). **(a)** VQ-VIB agents (orange dots) reach the theoretical bound from [13] (black curve) and capture the variation observed across the WCS languages (teal dots). **(b)** VQ-VIB achieves greater utility, for the same complexity, than onehot communication. **(c-f)** The emergent VQ-VIB color naming systems and closest human systems in the WCS data, for agents with lower (c-d) and higher (e-f) complexity, as shown in (a). Each communication vector is associated with the mean color it can refer to. **(c, e)** 2D PCA of the VQ-VIB communication vectors, where the size of each vector reflects its relative category size. **(d, f)** Mode maps, where each color chip from Figure 3c is colored according to its modal communication vector.

Appendix C. We observed similar benefits from high λ_I, similar complexity-reward tradeoffs, and similar advantages of VQ-VIB over other architectures in the 9 points environment.

5.2 Color reference games

In the Basic color reference game, an informativeness training loss appeared necessary for onehot, while a complexity-based loss was useful for generating a spectrum of communication conventions. First, we trained agents with a range of λ_I values while leaving $\lambda_C = 0$. We briefly report two main findings: 1) for a given λ_I, agents converged to a small range of complexity and informativeness and 2) VQ-VIB agents typically learned more complex and informative communication than onehot agents did, although increasing λ_I shrank the gap. These experiments are similar to those conducted by Chaabouni et al. [10], who did not train with informativeness or complexity losses. Full results for these experiments are included in Appendix D.

Second, we tested agents for $\lambda_I = 1.0$ while linearly annealing λ_C to penalize complexity. As λ_C increased, the communication complexity decreased, as shown in Figure 6 (left). Over 5 random seeds, the points in the plot, generated at each training epoch, closely approximated the IB upper bound for maximum informativeness for a given complexity. (A zoomed-in version of similar analysis, conducted with 1024 possible tokens, is included in Figure 13.) Inspection of the learned color naming schemes at different complexity levels revealed important trends (Figure 6 right). As complexity decreased, the number of communication clusters emitted by VQ-VIB agents decreased (visualized via 2-dimensional principle component analysis). Clustering in the communication space was reflected in the color space (e.g., "green" and "blue" merging). More generally, for any two randomly selected pairs of colors, the pair of colors that was closer together in CIELAB space resulted in vectors that were closer together in the communication space 70% of the time. This shows a fundamental connection between the form and meaning of communication used by VQ-VIB agents, which onehot-based communication (with all tokens equidistant) is architecturally unable to support.

Lastly, we compared agent color naming schemes to each other and to human language data. For each of the 110 languages in the WCS Dataset [42], we found the best-matching agent as determined

Table 1: Basic reference game results for annealed and fixed models. Human gNID is the average dissimilarity (±STD) between human color naming systems and nearest agent system.. Agent inefficiency is the average deviation from optimality (±STD) of the agents's systems, measured with respect to the theoretical bound from [13]. For both measures, lower values are better. Annealing leads to better alignment with human languages and to agents that communicate more efficiently.

	VQ-VIB Fixed	Onehot Fixed	VQ-VIB Annealed	Onehot Annealed
Human gNID	0.29 (0.03)	0.38 (0.01)	**0.16** (0.01)	**0.16** (0.00)
Agent inefficiency	0.08 (0.01)	0.08 (0.00)	**0.03** (0.00)	0.05 (0.00)

by minimizing gNID, a measure of dissimilarity of naming distributions introduced by Zaslavsky et al. [13]. Example nearest languages are shown in Figure 6. Despite having no supervised data, the artificial agents learned similar color naming schemes to humans, but only when agents changed complexity to match human languages' (shown via the gNID metrics in Table 1). This highlights the importance of controlling complexity in emergent communication: only by changing complexity could we generate naming systems similar to human communication. Thus, methods like Lin et al. [33] that encourage informativeness without penalizing complexity would be unable to recreate our results, and Chaabouni et al. [10] would have to change the environment to indirectly control complexity. Finally, when comparing VQ-VIB and onehot naming schemes, we found that agents were similarly efficient (deviation from optimality is calculated as ϵ in Table 1), but that VQ-VIB achieved higher utility, for the same complexity, as shown in Figure 6 b (corroborating results from particle world experiments). Varying hyperparameters such as the number of tokens or training scheme did not change results, as examined in Appendix D.

Together, these results indicate that annealing complexity is a critical component in recreating human-like color naming systems. Furthermore, VQ-VIB matched the form and meaning of communication by merging clusters in the communication space that represented nearby colors, and outperformed onehot communication by achieving higher utility for the same complexity.

Common ground color reference game We tested agents in the Common Ground color reference game to evaluate contextualized communication (results in Appendix D). The same trends as in the Basic experiments held: an informativeness loss improved convergence and, while both VQ-VIB and onehot were nearly optimally efficient, VQ-VIB achieved greater utility than onehot did.

6 Contributions

In this work, we explore the interplay of utility, informativeness, and complexity in emergent communication. To the best of our knowledge, we are the first to directly optimize agents to communicate according to a weighted combination of all three terms. We found that training agents while rewarding informativeness led to faster communication convergence, and penalizing complexity created more human-like communication. Further, we found that our novel neural architecture, VQ-VIB, achieved greater utility for the same complexity than other discrete methods. VQ-VIB combines the strengths of the methods it builds upon: it has a symbolic structure, defined by quantized vectors, but it also explicitly uses an IB encoder to sample in a continuous space. Natural extensions of our work include incorporation of new, cognitively-inspired terms to guide communication, or more complex domains, which could better illustrate advantages of VQ-VIB.

Our work represents a step towards reconciling principles from cognitive science with emergent communication literature, which we hope better supports human preferences. However, we recognize that, by quantifying properties of languages, our results may be misinterpreted to assign relative values to one language over another (e.g., higher complexity being better). We insist that our framework 1) has only been applied to narrow domains that do not capture any language's entirety and 2) does not prioritize any term and that any tradeoff between these terms is equally valid.

Acknowledgments and Disclosure of Funding

M.T. was supported by an Amazon Science Hub Fellowship. N.Z. was supported by a K. Lisa Yang Integrative Computational Neuroscience (ICoN) Postdoctoral Fellowship. We thank Rahma Chaabouni for useful discussions on their prior work.

References

[1] Luc Steels and Tony Belpaeme. Coordinating perceptually grounded categories through language: A case study for colour. *Behavioral and Brain Sciences*, 28(4):469–489, 2005.

[2] Jakob N. Foerster, Yannis M. Assael, Nando de Freitas, and Shimon Whiteson. Learning to communicate with deep multi-agent reinforcement learning. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, pages 2145–2153, Red Hook, NY, USA, 2016. Curran Associates Inc.

[3] Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pages 6382–6393, Red Hook, NY, USA, 2017. Curran Associates Inc.

[4] Igor Mordatch and Pieter Abbeel. Emergence of grounded compositional language in multi-agent populations. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'18/IAAI'18/EAAI'18. AAAI Press, 2018.

[5] Angeliki Lazaridou, Alexander Peysakhovich, and Marco Baroni. Multi-agent cooperation and the emergence of (natural) language. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

[6] Angeliki Lazaridou and Marco Baroni. Emergent multi-agent communication in the deep learning era. *arXiv preprint arXiv:2006.02419*, 2020.

[7] Rahma Chaabouni, Eugene Kharitonov, Emmanuel Dupoux, and Marco Baroni. Anti-efficient encoding in emergent communication. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 6290–6300, 2019.

[8] Satwik Kottur, José Moura, Stefan Lee, and Dhruv Batra. Natural language does not emerge 'naturally' in multi-agent dialog. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2962–2967, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1321.

[9] Tom Eccles, Yoram Bachrach, Guy Lever, Angeliki Lazaridou, and Thore Graepel. Biases for emergent communication in multi-agent reinforcement learning. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[10] Rahma Chaabouni, Eugene Kharitonov, Emmanuel Dupoux, and Marco Baroni. Communicating artificial neural networks develop efficient color-naming systems. *Proceedings of the National Academy of Sciences*, 118(12):e2016569118, 2021. doi: 10.1073/pnas.2016569118.

[11] Edward Gibson, Richard Futrell, Steven Piantadosi, Isabelle Dautriche, Kyle Mahowald, Leon Bergen, and Roger Philip Levy. How efficiency shapes human language. *Trends in Cognitive Sciences*, 23:389–407, 2019.

[12] Charles Kemp, Yang Xu, and Terry Regier. Semantic typology and efficient communication. *Annual Review of Linguistics*, 4:109–128, 01 2018. doi: 10.1146/annurev-linguistics-011817-045406.

[13] Noga Zaslavsky, Charles Kemp, Terry Regier, and Naftali Tishby. Efficient compression in color naming and its evolution. *Proceedings of the National Academy of Sciences*, 115(31): 7937–7942, 2018. doi: 10.1073/pnas.1800521115.

[14] Noga Zaslavsky, Terry Regier, Naftali Tishby, and Charles Kemp. Semantic categories of artifacts and animals reflect efficient coding. In *Proceedings of the 41st Annual Meeting of the Cognitive Science Society*, 2019.

[15] Noga Zaslavsky, Mora Maldonado, and Jennifer Culbertson. Let's talk (efficiently) about us: Person systems achieve near-optimal compression. In *Proceedings of the 43rd Annual Meeting of the Cognitive Science Society*, 2021.

[16] Francis Mollica, Geoff Bacon, Noga Zaslavsky, Yang Xu, Terry Regier, and Charles Kemp. The forms and meanings of grammatical markers support efficient communication. *Proceedings of the National Academy of Sciences*, 118(49):e2025993118, 2021. doi: 10.1073/pnas.2025993118.

[17] Noga Zaslavsky, Karee Garvin, Charles Kemp, Naftali Tishby, and Terry Regier. The evolution of color naming reflects pressure for efficiency: Evidence from the recent past. *Journal of Language Evolution*, 04 2022. ISSN 2058-458X. doi: 10.1093/jole/lzac001. lzac001.

[18] Naftali Tishby, Fernando C. Pereira, and William Bialek. The Information Bottleneck method. In *Proceedings of the 37th Annual Allerton Conference on Communication, Control and Computing*, 1999.

[19] Alexander A. Alemi, Ian Fischer, Joshua V. Dillon, and Kevin Murphy. Deep variational information bottleneck. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

[20] Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pages 6309–6318, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

[21] Mycal Tucker, Huao Li, Siddharth Agrawal, Dana Hughes, Katia P. Sycara, Michael Lewis, and Julie A. Shah. Emergent discrete communication in semantic spaces. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 10574–10586, 2021.

[22] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013.

[23] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1532–1543, 2014. doi: 10.3115/v1/d14-1162.

[24] Ł ukasz Kuciński, Tomasz Korbak, PawełKoł odziej, and Piotr Mił oś. Catalytic role of noise and necessity of inductive biases in the emergence of compositional communication. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 23075–23088. Curran Associates, Inc., 2021.

[25] Igor Mordatch and Pieter Abbeel. Emergence of grounded compositional language in multi-agent populations. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), Apr. 2018.

[26] Rundong Wang, Xu He, Runsheng Yu, Wei Qiu, Bo An, and Zinovi Rabinovich. Learning efficient multi-agent communication: An information bottleneck approach. In *ICML*, pages 9908–9918, 2020.

[27] Claude E. Shannon. Coding theorems for a discrete source with a fidelity criterion. *IRE Nat. Conv. Rec*, 4(142-163):1, 1959.

[28] Matti Miestamo. On the feasibility of complexity metrics. In *FinEst linguistics, proceedings of the annual Finnish and Estonian conference of linguistics, Tallinn*, pages 11–26, 2004.

[29] Mathieu Rita, Rahma Chaabouni, and Emmanuel Dupoux. "LazImpa": Lazy and impatient neural agents learn to communicate efficiently. In *Proceedings of the 24th Conference on Computational Natural Language Learning*, pages 335–343, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.conll-1.26.

[30] Michael Cogswell, Jiasen Lu, Stefan Lee, Devi Parikh, and Dhruv Batra. Emergence of compositional language with deep generational transmission. *CoRR*, abs/1904.09067, 2019.

[31] Yi Ren, Shangmin Guo, Matthieu Labeau, Shay B. Cohen, and Simon Kirby. Compositional languages emerge in a neural iterated learning model. In *International Conference on Learning Representations*, 2020.

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

[33] Toru Lin, Jacob Huh, Christopher Stauffer, Ser Nam Lim, and Phillip Isola. Learning to ground multi-agent communication with autoencoders. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 15230–15242. Curran Associates, Inc., 2021.

[34] Naftali Tishby and Daniel Polani. Information theory of decisions and actions. In Vassilis Cutsuridis, Amir Hussain, and John G. Taylor, editors, *Perception-Action Cycle: Models, Architectures, and Hardware*, pages 601–636. Springer New York, New York, NY, 2011.

[35] Susanne Still and Doina Precup. An information-theoretic approach to curiosity-driven reinforcement learning. *Theory in Biosciences*, 131(3):139–148, 2012.

[36] Pedro A. Ortega and Daniel A. Braun. Thermodynamics as a theory of decision-making with information-processing costs. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 469(2153):20120683, 2013. doi: 10.1098/rspa.2012.0683.

[37] Roy Fox, Ari Pakman, and Naftali Tishby. Taming the noise in reinforcement learning via soft updates. In *32nd Conference on Uncertainty in Artificial Intelligence (UAI)*, 2016.

[38] Anirudh Goyal, Riashat Islam, DJ Strouse, Zafarali Ahmed, Hugo Larochelle, Matthew Botvinick, Sergey Levine, and Yoshua Bengio. Transfer and exploration via the information bottleneck. In *International Conference on Learning Representations*, 2019.

[39] David Lewis. *Convention: A philosophical study*. John Wiley & Sons, 2008.

[40] Brent Berlin and Paul Kay. *Basic Color Terms: Their Universality and Evolution*. University of California Press, Berkeley and Los Angeles, 1969.

[41] Terry Regier, Paul Kay, and Naveen Khetarpal. Color naming reflects optimal partitions of color space. *Proceedings of the National Academy of Sciences*, 104(4):1436–1441, 2007. doi: 10.1073/pnas.0610341104.

[42] Paul Kay, Brent Berlin, Luisa Maffi, William R. Merrifield, and Richard Cook. *The World Color Survey*. Stanford: Center for the Study of Language and Information, 2009.

[43] Mike Dowman. Explaining color term typology with an evolutionary model. *Cognitive Science*, 31(1):99–132, 2007.

[44] Will Monroe, Robert X.D. Hawkins, Noah D. Goodman, and Christopher Potts. Colors in context: A pragmatic neural model for grounded language understanding. *Transactions of the Association for Computational Linguistics*, 5:325–338, 2017.

[45] Mikael Kågebäck, Emil Carlsson, Devdatt Dubhashi, and Asad Sayeed. A reinforcement-learning approach to efficient communication. *PLOS ONE*, 15(7):1–26, 07 2020.

[46] Toby Berger. *Rate distortion theory; a mathematical basis for data compression*. Prentice-Hall, Englewood Cliffs, NJ, 1971.

[47] Peter Harremoës and Naftali Tishby. The Information Bottleneck revisited or how to choose a good distortion measure. In *IEEE International Symposium on Information Theory*, pages 566–571, June 2007.

[48] Noga Zaslavsky and Naftali Tishby. Deterministic annealing and the evolution of optimal information bottleneck representations. 2019.

[49] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

[50] Yoshua Bengio, Nicholas Léonard, and Aaron C. Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *CoRR*, abs/1308.3432, 2013.

[51] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, second edition, 2018.

[52] Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach. Learn.*, 8(3–4):229–256, may 1992. ISSN 0885-6125. doi: 10.1007/BF00992696.

[53] Sainbayar Sukhbaatar, Arthur Szlam, and Rob Fergus. Learning multiagent communication with backpropagation. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, pages 2252–2260, Red Hook, NY, USA, 2016. Curran Associates Inc.

[54] Jennifer Hu, Roger Levy, and Noga Zaslavsky. Scalable pragmatic communication via self-supervision. In *ICML Workshop on Self-Supervised Learning for Reasoning and Perception*, 2021.

[55] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017.

[56] Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017.

Checklist

The checklist follows the references. Please read the checklist guidelines carefully for information on how to answer these questions. For each question, change the default **[TODO]** to [Yes] , [No] , or [N/A] . You are strongly encouraged to include a **justification to your answer**, either by referencing the appropriate section of your paper or providing a brief inline description. For example:

- Did you include the license to the code and datasets? [Yes] See Section TODO.
- Did you include the license to the code and datasets? [No] The code and the data are proprietary.
- Did you include the license to the code and datasets? [N/A]

Please do not modify the questions and only use the provided macros for your answers. Note that the Checklist section does not count towards the page limit. In your paper, please delete this instructions block and only keep the Checklist section heading above along with the questions/answers below.

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 abstract, introduction, and findings

(b) Did you describe the limitations of your work? [Yes] See Section 6, in which we note that we tested in limited domains and Section 3.4.1, in which we note that an approximation we made may be improved upon in future work.

(c) Did you discuss any potential negative societal impacts of your work? [Yes] See Section 6 in which we discuss how readers may misconstrue quantitative measures of languages as assigning value to one over the other.

(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] See footnote for link to code

   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Appendix B

   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] See error bars in plots and Table 1, in Section 5. All scatter plots directly included the raw results from all 5 random trials.

   (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] Training times and computational resources are listed in Appendix B.

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 the World Color Survey (WCS) data, as well as the publicly-available code by Zaslavsky et al. [13]. Both sources are cited.

   (b) Did you mention the license of the assets? [Yes] We note that both are publicly-available and link to them in Footnote 1.

   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] Code is linked to in a footnote.

   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A] We only used the existing WCS dataset.

   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A] We only used the existing WCS dataset.

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]

A Appendix: VQ-VIB After

We proposed an additional architecture, **VQ-VIB After**, that is closely related to our main VQ-VIB method. In VQ-VIB After, an encoder to generates a discrete representation ζ, whereupon a communication vector c is sampled from a Gaussian in the continuous space centered around ζ. This "sample-after" method is depicted in Figure 8, and corresponds to an intuitive notion of choosing a word (a discrete embedding) deterministically but then being uncertain about the form (by sampling in the continuous space around it).

This architecture fits into the standard Information Bottleneck framework, as complexity can be penalized via standard variational methods. However, it is distinct from discrete communication

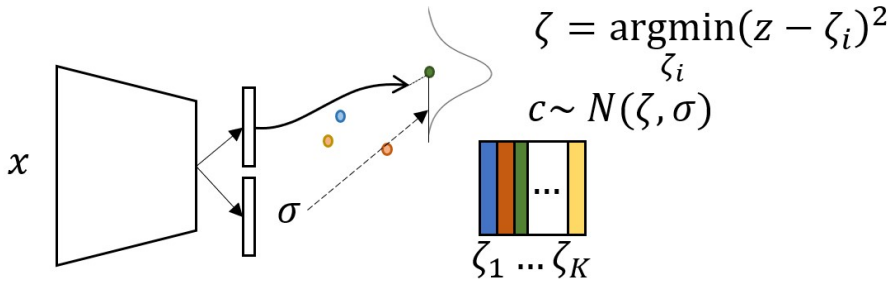


Figure 7: In the VQ-VIB After architecture, the speaker deterministically maps x to the nearest quantized encoding, ζ, and then c is sampled from a Gaussian centered at ζ.

methods because communication vectors are ultimately sampled from a continuous space. Thus, this work does not fit naturally into the main focus of this work, but we believe it could have applications in other settings, such as speech processing (wherein discrete words are pronounced in a continuous auditory space).

B Appendix: Implementation Details

Here, we include implementation details that we omitted from the main paper that can be used to reproduce our experiments.

B.1 Multi-Agent Reinforcement Learning Experiments

We trained agents in the 9 points and Uniform environments with Multi-Agent Deep Deterministic Policy Gradient (MADDPG), a standard method for multi-agent reinforcement learning [3].

Agent architectures were identical in both environments. Regardless of the speaker architecture used, all teams used common decoder and listener agents (except for variations due to different communication dimensions, explained later). The decoder used 2 fully-connected ReLU layers with hidden dimension 64; the listener used 3 fully-connected ReLU layers with hidden dimension 64.

Because of the common architecture for the decoder and listener agents across teams, variation arose due to differences among the speaker agents. A speaker agent consisted of 2, 64-neuron fully-connected layers with ReLU activations, followed by a speaker "head." The head implemented the specific communication method (e.g., continuous, VQ-VIB, etc.) by taking the 64-dimensional representation from the speaker and going through one of 5 possible architectures to produce a vector in the communication space, $\mathbb{R}^d$. (As discussed later, d varied by environment and architecture slightly.)

- **Continuous:** The continuous communication head consisted of two fully connected layers branching off the speaker encoding to generate a μ and `logvar` (the log of the variance) from which the communication vector was sampled using the reparametrization trick. Complexity was measured as the KL divergence between the parameters μ and `logvar` and a unit Gaussian.
- **Onehot:** The onehot communication head consisted of one fully connected layer from the speaker encoding to $\mathbb{R}^d$. Onehot vectors were generated using gumble softmax sampling, with temperature 1. Complexity was measured as the KL divergence between the distribution from which the onehot vector was sampled and a uniform categorical distribution. Thus, complexity corresponds to the difference between the conditional and unconditional entropies of messages, unlike non-IB frameworks that often penalize unconditional entropy. We used greater communication dimensionality for onehot agents and therefore rescaled the first layer of the decoder and listener agents to accept such communication.
- **VQ-VIB:** The VQ-VIB communication head first passed the speaker encoding through two fully connected layers, branching off the encoding, to generate a μ and `logvar`, as in the continuous communication case. A communication vector was sampled from the distribution,

whereupon it was discretized to the nearest quantized vector, ζ, in the communication space (as done in a standard VQ-VAE, without the sampling). Complexity was measured as the KL divergence between the parameters μ and `logvar` and a unit Gaussian.

- **VQ-VIB After:** The VQ-VIB After communication head first passed the speaker encoding through two fully connected layers to generate an encoding, z, and `logvar`. The encoding, z, was discretized to the nearest quantized vector, ζ. Lastly, a communication vector, c, was sampled from a normal distribution centered at ζ with variance specified via `logvar`. Complexity was measured as the KL divergence between the parameters ζ and `logvar` and a unit Gaussian.
- **Proto:** The Proto head mapped the speaker encoding through a single fully-connected layer to a categorical distribution over k neurons, for k representing the number of prototypes. A k-dimensional onehot vector was sampled using gumble softmax; a prototype, p was selected by multiplying the prototype matrix by the onehot vector. A different fully-connected layer mapped from the speaker encoding to `logvar`. Lastly, a communication vector, c, was sampled from a Gaussian centered at p with variance via `logvar`. Complexity was measured as the KL divergence between the parameters p and `logvar` and a unit Gaussian. This closely resembles the VQ-VIB After architecture, but with a different discretization mechanism.

All agents were trained with an Adam optimizer with learning rate 0.01 and batch size 1024.

B.1.1 9 points Environment

For all agent architectures except for onehot, the communication dimension was 2; onehot used communication dimension 9 so that the number of discrete representations allowed by the architecture was not the limiting factor for complexity. Similarly, the VQ-VIB and Proto agents used 9 discrete embeddings.

In our results with annealed complexity weight (presented in Figure 11), agents were trained for 10,000 episodes, with each episode lasting 100 timesteps. The listener agent was always spawned at the origin, equidistant from each landmark. The informativeness loss weight, λ_I, was constant at 1.0. For the continuous and VQ-VIB communication methods, the complexity loss weight, λ_C increased by $8 * 10^{-6}$ per episode. The onehot and Proto methods suffered from communication collapse at that rate, so results λ_C increased half as quickly: at $4 * 10^{-6}$ per episode.

Training a team in the 9 points environment for 10,000 episodes took approximately 15 minutes on a desktop computer with 16 Intel i9 cores for all architectures, except for onehot, which took roughly 20 minutes. This difference in time is attributable to the larger communication dimension needed for onehot.

B.1.2 Uniform Environment

In the Uniform environment, we used identical parameters as in the 9 points environment, except for the changes noted below.

For all agent architectures except for onehot, the communication dimension was 3; for onehot, we experimented with communication dimensions of 9 and 100. Similarly, we tested VQ-VIB and Proto agents when using 9 or 100 discrete embeddings. The listener agent was spawned at a starting location sampled uniformly at random from the map (bounded between -200 and 200 for both x and y). Otherwise, we used the same number of training episodes, episode duration, and complexity annealing rates as in 9 points.

Training a team in the Uniform environment for 10,000 episodes took approximately 15 minutes, except for onehot, which took roughly 30 minutes for 100-dimensional communication.

B.2 Color Reference Game Experiments

Here, we include the implementation details used in our color reference game experiments.

Agents were trained in the overall architecture depicted in Figure 8. The speaker agent was parametrized with 3 fully connected layers with ReLU activations and hidden dimension 64, followed

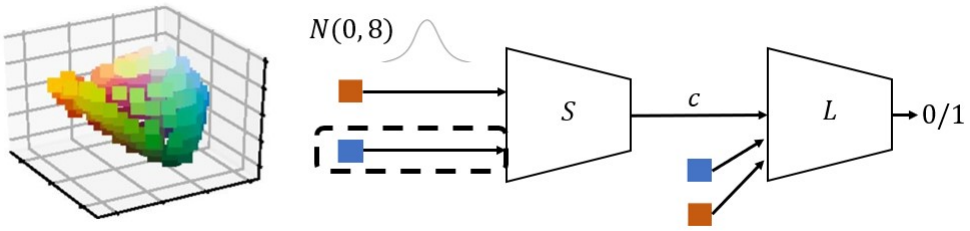


Figure 8: Target and distractor colors were drawn uniformly without replacement from the set of 330 colors in CIELAB space (left). A speaker, S, observed the target and (in the Common Ground setting) distractor, corrupted by Gaussian noise. The listener, L, had to identify which of the shuffled colors it saw was the target color, given the speaker's communication, c.

by a communication head. The communication heads used the same architectures as specified in the particle world experiments: gumbel softmax was used for generating onehot vectors, and a straight-through estimator was used for discretization in VQ-VIB architectures [55, 56]. In all experiments, the communication dimension was set to 32 for all agents except onehot, for which the communication dimension was set to 330 (to match the expressive power of other discrete agents, that each had 330 discrete tokens in the 32 dimensional space). The decoder and listener agents used the same feed-forward architecture as in the particle world experiments.

Without annealing the complexity loss, 50 training epochs sufficed for agents to converge to a stable solution. Thus, in our experiments with $\lambda_C = 0$, we only trained for 50 epochs. For later experiments, wherein we annealed complexity, we first trained agents for 50 epochs before beginning to anneal the complexity term for the next 40 epochs (for a total of 90 epochs). We note that, for VQ-VIB, we set the complexity loss to 0 during this "burnin" period, but for onehot communication, it appeared necessary to set the loss to 0.05 to prevent numerical instability during the first epoch after the burnin period. During complexity loss annealing, we increased λ_C by 0.01 during each epoch for VQ-VIB, and by 0.1 for onehot. These different rates were necessary to induce similar ranges in complexity of learned communication.

In the Basic color reference game, the speaker observed a only the 3-dimensional, corrupted target color. In the Common Ground setting, the speaker observed a 6-dimensional vector, generated by concatenating noisy versions of the target and distractor colors. In both settings, the listener observed the communication output by the speaker, as well as the target and distractor colors (ordered randomly, without added noise). Results were analyzed using code partially-based on publicly-available code released by Zaslavsky et al. [13].

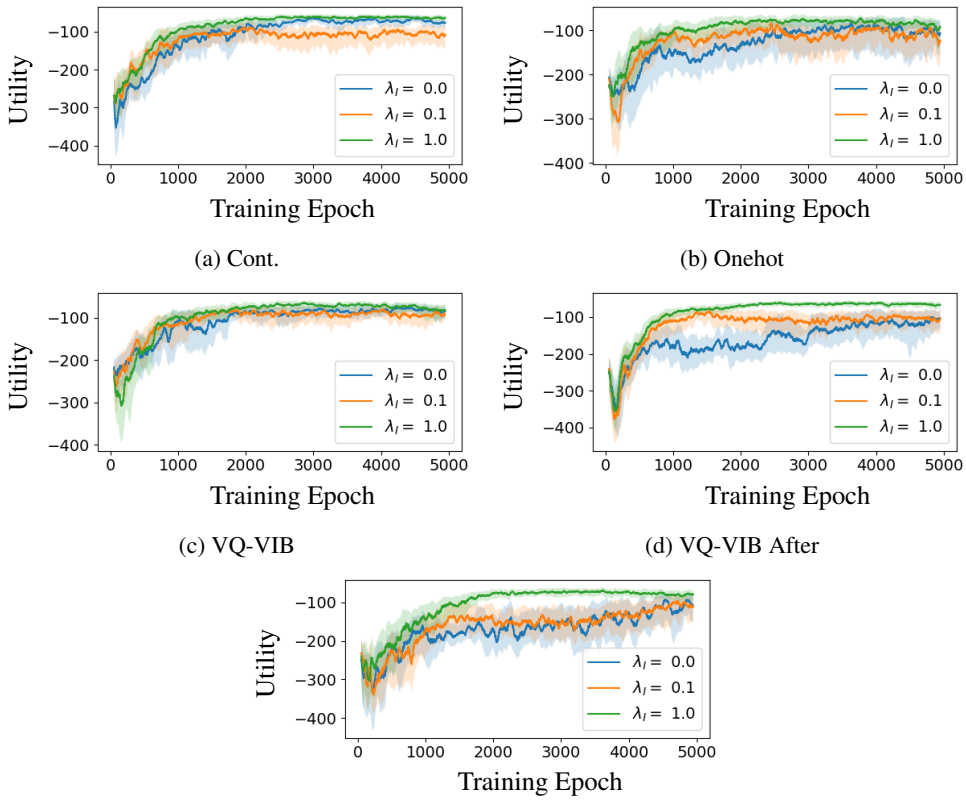
We trained agents on a desktop computer with 16 Intel i9 CPUs and 1 NVIDIA GeForce RTX 2080. Training for 90 epochs in both the Basic and Common Ground settings took approximately 3 minutes for all agent architectures. Agents were trained with an Adam optimizer with default parameters, with batch size 1024.

C Appendix: Full Particle World Results

For brevity, we only included partial results from the particle-world experiments in the main paper. Here, we include the full results.

First, we experimented with varying λ_I for different environments and neural architectures, without annealing complexity. Training curves for both environments, using varying λ_I for each neural architecture, are included in Figures 9 and 10. For each configuration (a particular speaker architecture and λ_I), we trained 5 teams from scratch for 5,000 episodes and plotted the median and quartile rewards, smoothed over a 50 episode window.

A consistent trend emerged: higher λ_I led to faster convergence to a higher reward. Indeed, this corroborates findings by Lin et al. [33], who used an autoencoder architecture for communication in multiagent settings, but here we showed good convergence properties when mixing utility and informativeness terms (whereas they reported worse results when adding utility losses). Overall,



(a) Cont.

(b) Onehot

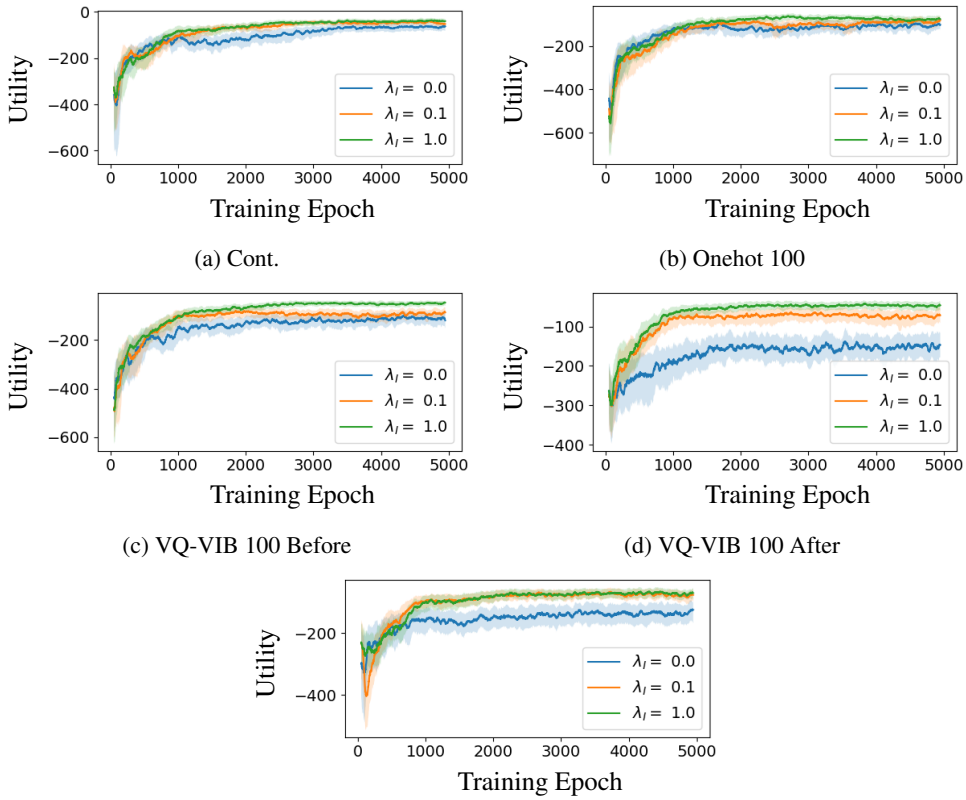
(c) VQ-VIB

(d) VQ-VIB After

(e) Proto

Figure 9: In the 9 points environment, using a higher weight for informativeness (λ_I) resulted in faster convergence to higher rewards for each communication architecture. Medians and quartiles plotted over 5 trials for each configuration.

these findings motivated training agents to convergence with high λ_I before beginning to penalize complexity with λ_C.



(a) Cont.

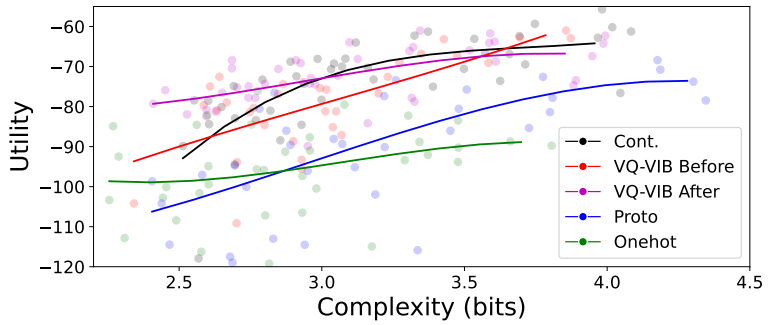
(b) Onehot 100

(c) VQ-VIB 100 Before

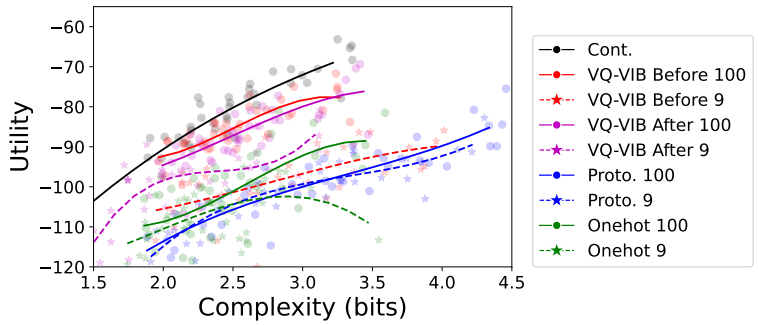
(d) VQ-VIB 100 After

(e) Proto 100

Figure 10: In the Uniform environment, we observed similar trends as in 9 points: greater λ_I improved convergence rates and rewards. Medians and quartiles plotted over 5 runs.



(a) 9 points: Reward vs. Complexity



(b) Uniform: Reward vs. Complexity

Figure 11: In the 9-points and Uniform environments, agent reward (utility) decreased with complexity. Each point represents the mean complexity and reward for agent checkpoints at 1,000 episode increments during training, over 5 random seeds. Curves are the best cubic fits to the data. VQ-VIB architectures typically outperformed other discrete methods by achieving higher reward for the same complexity, and over-parametrizing with additional embeddings (100 vs. 9) also helped.

Next, we include the results of annealing complexity in both the 9 points and Uniform environments, as depicted in Figure 11. In both environments, as we increased λ_C, complexity and utility decreased. Comparing both VQ-VIB architectures to other discrete communication methods showed that VQ-VIB agents tended to obtain a higher mean reward, for a given complexity, than other methods. In the Uniform environment, we also compared different variants of VQ-VIB, Proto, and onehot agents based on the maximum number of discrete messages allowed by the neural architectures. We tested agents parametrized with either 100 or 9 unique messages; we hypothesized that initializing agents with many discrete messages and then gradually decreasing complexity would result in better performance than training with few messages from the start. Indeed, we observed that was the case. As seen in Figure 11 b, VQ-VIB agents parametrized with 100 embeddings consistently achieved higher reward for the same complexity than VQ-VIB agents parametrized with only 9 embeddings. This trend held even when the complexity was lower than $\log_2(9)$ bits.

Ultimately, results from both particle world environments corroborate the findings from the color reference games: encouraging informativeness improved convergence, VQ-VIB achieved greater utility than onehot, and decreasing complexity led to coarser discretizations (main paper, Figure 4). The varied domains and training mechanisms (MARL vs. backpropagation for reference games) leading to consistent results suggest broad generalizability of our framework and VQ-VIB.

D Appendix: Color Reference Game Full Results

We omitted some of the color reference game results from the main paper due to space constraints. Here, we include our complete results.

D.1 Basic Color Reference Game

First, we include results from the Basic setting, in which we trained agent with different values of λ_I without penalizing complexity. These results are included in Figure 12. Setting $\lambda_I = 0$ induced similar conditions to those used by Chaabouni et al. [10]; any positive λ_I favored more informative (and therefore more complex) communication. Given the monotonically increasing nature of the IB bound, an optimal agent trained with any positive λ_I would maximize utility and informativeness by maximizing complexity, up to a maximum of $\log_2(330)$ bits. Instead, we found that agents appeared to converge to lower complexity values, although increasing λ_I tended to somewhat increase complexity and informativeness. For example, onehot communication increased in complexity from roughly 2.3 bits for $\lambda_I = 0.0$ to 3.0 bits for $\lambda_I = 1.0$. This alone is a novel result compared to Chaabouni et al. [10], who trained agents with no informativeness loss and observed lower-complexity communication. Curiously, VQ-VIB agents automatically converged to informativeness around 3.5 bits regardless of the λ_I value, although increasing λ_I did improve training stability. This difference between onehot and VQ-VIB indicates that the differing architectures themselves have different inductive biases.

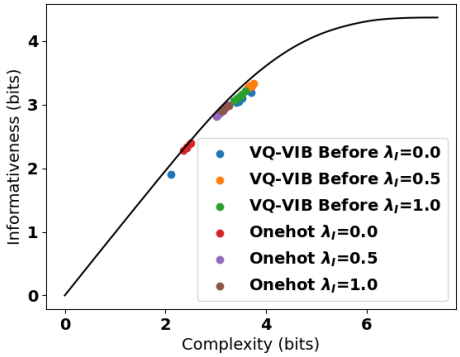
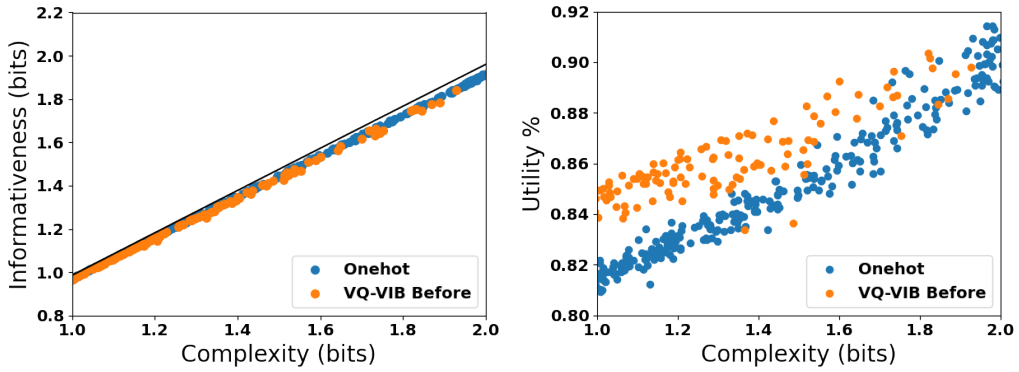


Figure 12: Training agents in the Basic color reference game for various λ_I but no complexity loss. Without positive λ_I, onehot learns low-informativeness communication, but increasing λ_I generated more complex agents, especially for onehot.

We conducted further experiments to compare agents with the neural agents presented by Chaabouni et al. [10], wherein onehot agents were parametrized with a communication dimension of 1024. We therefore tested VQ-VIB and onehot agents with 1024 tokens, instead of the 330 used for the results in our main paper. This hyperparameter change largely did not alter results, which is unsurprising given that agents with 330 tokens did not use their full vocabulary. As shown in Figure 13, both VQ-VIB and onehot achieved nearly optimal informativeness for a given complexity, and VQ-VIB achieved higher utility than onehot (results zoomed in to complexity values between 1 and 2 bits for clearest plotting of these trends).

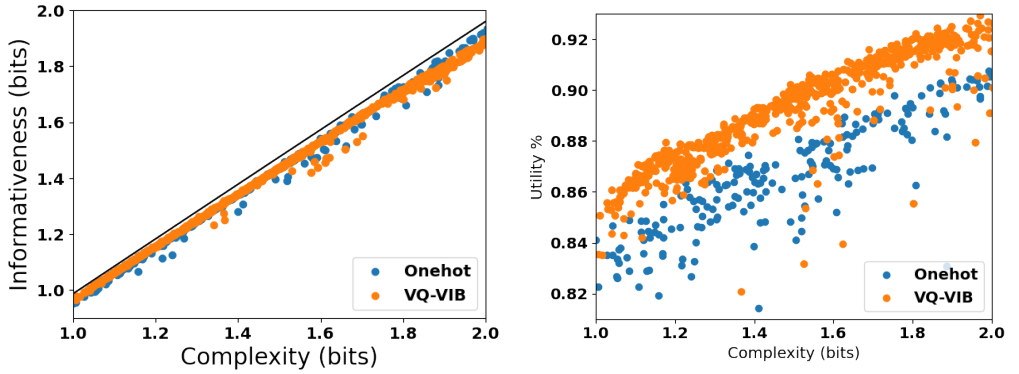
Lastly, we performed additional experiments to investigate the effect of training with REINFORCE, instead of the supervised learning framework used in our main experiments (e.g., backpropagating through gumbel softmax) [52]. Chaabouni et al. [10] had found that training with REINFORCE typically resulted in lower-complexity communication than when training with messages generated via a gumbel softmax layer, and that training with REINFORCE was less stable. Results from our REINFORCE experiments, using 330 tokens for onehot and VQ-VIB, are included in Figure 14. We trained agents with $\lambda_U = 1.0$, $\lambda_I = 50.0$, and λ_C was annealed from 0 to 0.1, using code adapted from Rita et al. [29].

Our results from these experiments align with all prior findings: VQ-VIB and onehot achieved high communicative efficiency, and VQ-VIB produced greater utility than onehot did. Beyond those corroborating results, our experiments aligned with Chaabouni et al. [10]'s findings and



(a) (b)

Figure 13: Communicative informativeness (a) and utility (b) as a function of complexity for color agents trained with 1024 tokens in the Basic game. These results largely reproduced results from our main paper with 330 tokens, wherein VQ-VIB and onehot were similarly efficient, but VQ-VIB achieved higher utility than onehot.



(a) (b)

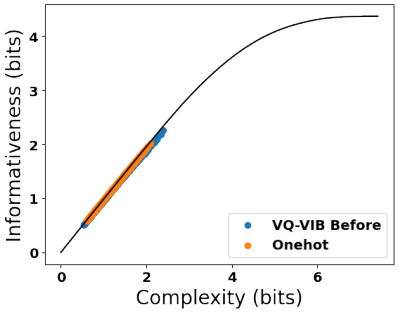
Figure 14: Informativeness (a) and utility (b) versus complexity for agents trained with 330 tokens, using REINFORCE, in the Basic reference game. These results closely tracked prior experiments: both onehot and VQ-VIB were efficient, and VQ-VIB achieved greater utility, for the same complexity, than onehot.

further illustrated the importance of informativeness losses and the VQ-VIB architecture. With zero informativeness loss (i.e., $\lambda_I = 0$) onehot agents often failed to converge to any useful communication (as indicated in prior art), but by increasing λ_I, agents consistently converged to useful communication. Indeed, with $\lambda_I = 50$ and $\lambda_C = 0$, onehot agents learned to communication with complexity of up to 2.4 bits, greater than any complexity value found by Chaabouni et al. [10] using REINFORCE over 180 seeds (see Fig. 5 in that paper). Furthermore, we observed that VQ-VIB agents typically converged to higher informativeness (and higher complexity) communication than onehot when complexity was not penalized, indicating that further investigation of inductive biases is a promising direction for future work. Overall, this experiment largely replicates our findings from the main paper.

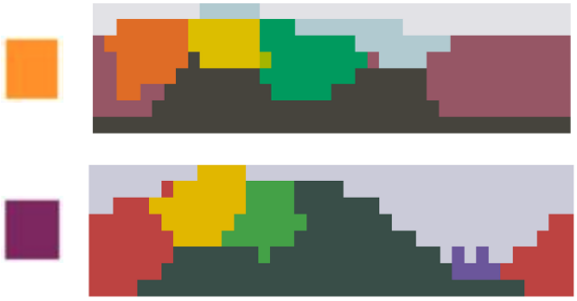
Thus, findings from our REINFORCE experiments indicate that the main conclusions of our paper hold true across several training mechanisms, and furthermore that our framework (including in particular the informativeness loss) can address issues already noted in prior work.

D.2 Common Ground Color Reference Game

Here, we include the full results from our Common Ground color reference game experiments. Crucially, these experiments tested agent behavior when utility and informativeness were not identical: utility could be maximized by the speaker only communicating about the target color, whereas informativeness was maximized by communicating about both target and distractor. Given the lack of human-labeled data for such a setting, we were unable to compare agent communication with human languages. However, by analyzing communication when varying the target for a fixed distractor, or by varying the distractor for a fixed target, we established that agents conditioned communication both on the target and distractor colors.



(a) Informativeness and Complexity for a gray distractor



(b) Communication by the same agent conditioned on purple or orange distractors differed, showing context-dependent communication.

Figure 15: In Common Ground experiments, both VQ-VIB and onehot communication demonstrated high informativeness for a fixed distractor (left), and varied communication according to distractor (right).

We trained agents in the Common Ground setting by randomly sampling target and distractor colors, as in the Basic setting, but here the speaker observed noisy versions of the distractor as well as the target. Based on insights that training to increase informativeness produced more complex communication, we trained agents for 30 episodes with $\lambda_I = 1.0$. For the next 70 episodes, we annealed the complexity weight, λ_C, using the same rates as in the Basic setting; this produced communication at different levels of complexity. As in the Basic setting, we found that VQ-VIB achieved higher utility, for the same complexity, than onehot (Figure 16).

We further analyzed communication in two ways: using a fixed distractor with varying targets and using a fixed target with varying distractors. First, for a fixed, gray distractor (#46 in the WCS data), we measured the standard metrics of informativeness and complexity. That is, we set the distractor color to gray, and we recorded, for each of the other 329 colors as the target, the speaker's communication. We then performed the same complexity and informativeness analysis as in the Basic

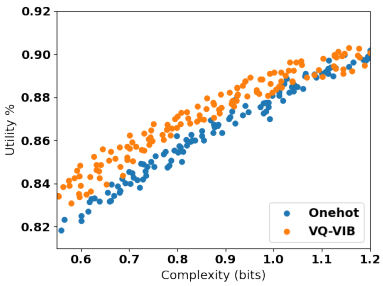


Figure 16: In the Common Ground color reference game (as well as the Basic reference game, presented in the main paper), VQ-VIB achieved greater utility, for the same complexity, than onehot communication.

setting, and found that both VQ-VIB and onehot communication remained near-optimally efficient, as shown in Figure 15 a.

Second, we measured whether agents conditioned their communication on the distractor color. Unlike the prior evaluation, in which we varied the target color for a fixed distractor, here, we evaluated communication for different distractors. Plots of a VQ-VIB agent's naming scheme, using either an orange (color #212) or purple (color #325) distractor are shown in Figure 15 b.

As expected, the agent displayed different resolutions in regions of color depending upon the distractor. Specifically, for an orange distractor, the speaker had finer resolution around shades of orange and yellow; conversely, the speaker had finer resolution around shades of purples for a purple distractor. This aligns with expectations that, when trying to differentiate between colors, one must use more complex or accurate words to distinguish similar colors.

Jointly, our findings indicate that many of the themes identified in the Basic setting – nearly-optimal communicative efficiency, the importance of annealing λ_C to generate varied complexity, and the advantages of VQ-VIB over onehot – held in the Common Ground setting as well.