Using Principles from Cognitive Science to Train and Analyze Language-Related Neural Networks

by

Mycal Tucker

B.S., Massachusetts Institute of Technology, 2015
M.Eng., Massachusetts Institute of Technology, 2016

Submitted to the Department of Aeronautics and Astronautics
in partial fulfillment of the requirements for the degree of
DOCTOR OF PHILOSOPHY IN ARTIFICIAL INTELLIGENCE
at the
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
April 2024

Authored by: Mycal Tucker
Department of Aeronautics and Astronautics
April 25, 2024

Certified by: Julie A. Shah
H.N. Slater Professor of Aeronautics and Astronautics, MIT
Thesis Supervisor

Certified by: Roger Levy
Professor of Department of Brain and Cognitive Sciences, MIT
Thesis Committee Member

Certified by: Ellie Pavlick
Manning Assistant Professor of Computer Science and Assistant Professor of Linguistics, Brown University
Thesis Committee Member

Certified by: Been Kim
Senior Staff Research Scientist, Google DeepMind
Thesis Committee Member

Accepted by: Jonathan P. How
R. C. Maclaurin Professor of Aeronautics and Astronautics
Chair, Graduate Program Committee

Using Principles from Cognitive Science to Train and Analyze Language-Related Neural Networks

by

Mycal Tucker

Submitted to the Department of Aeronautics and Astronautics
on April 25, 2024, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Artificial Intelligence

Abstract

Natural language, while central to human experience, is not uniquely the domain of humans. AI systems, typically neural networks, exhibit startling language processing capabilities from generating plausible text to modeling simplified language evolution. To what extent are such AI models learning language in a "human-like" way?

Defining "human-like" generally may be an impossible problem, but narrower definitions of aspects of human-like language processing, borrowed from cognitive science literature, afford metrics for evaluating AI models. In this thesis, I borrow two theories about human language processing for such analysis. First, human naming systems (e.g., a language's words for colors such as "red" or "blue") appear near-optimal in an information-theoretic sense of compressing meaning into a small number of words; I ask how one might train AI systems that behave similarly. Second, people understand and produce language according to hierarchical representations of structure; I study whether large language models use similar representations in predicting text. Thus, in this thesis, I show how to train and analyze neural networks according to cognitive theories of human language processing.

In my first branch of work, I introduce a method for neural network agents to communicate according to cognitively-motivated pressures for utility, informativeness, and complexity. Utility represents a measure of task success and induces task-specific communication; informativeness is a task-agnostic measure of how well listeners understand speakers and leads to generalizable communication; complexity captures how many bits are allocated for communication and can lead to simpler communication systems. All three terms are important for human-like communication. In experiments, training artificial agents according to different tradeoffs between these properties led them to learn different naming systems that closely aligned with existing natural languages.

In my second branch of work, rather than training neural agents from scratch, I probe pre-trained language models and found that some use representations of syntax in making predictions. Humans use hierarchical representations of sentence structure in understanding and producing language, but it is unclear if large language models, trained on simple tasks like next-word-prediction, should learn similar representations. I introduce a causal probing method that sheds light on this topic. By creating counterfactual representations

of syntactically ambiguous sentences, I measured how model predictions changed for different structural interpretations of the same sentence. For example, I recorded model predictions to ambiguous inputs like "The girl saw the boy with the telescope. Who had the telescope?" with different syntactic structures. For some (but not all) models, I found that models use representations of syntax (e.g., change their answers to the previous question). Thus, I offer novel insight into pre-trained models and a new method for studying such models for other properties.

The two halves of my thesis represent complementary approaches towards more human-like AI; training new models and analyzing pre-trained ones closes an AI development feedback loop. In this thesis, I explain my contributions to both parts of this loop and identify promising directions for future research.

Thesis Supervisor: Julie A. Shah
Title: H.N. Slater Professor of Aeronautics and Astronautics, MIT

Thesis Committee Member: Roger Levy
Title: Professor of Department of Brain and Cognitive Sciences, MIT

Thesis Committee Member: Ellie Pavlick
Title: Manning Assistant Professor of Computer Science and Assistant Professor of Linguistics, Brown University

Thesis Committee Member: Been Kim
Title: Senior Staff Research Scientist, Google DeepMind

Dissertation Reader: Jacob Andreas
Title: Associate Professor of Electrical Engineering and Computer Science, MIT

Dissertation Reader: Edward Gibson
Title: Professor of Department of Brain and Cognitive Sciences, MIT

Acknowledgments

I am deeply grateful to the many people who helped me in ways both personal and professional throughout my PhD.

First, I thank Julie, my advisor. I first met Julie when I was a freshman at MIT; despite my lack of research experience, Julie offered me a position as an undergraduate researcher. That summer job gave me my first view into research and showed me a wonderful and collaborative side of academia. Later, when starting my PhD at MIT, Julie yet again supported me as I first began exploring language and AI despite having little background in such topics. She allowed me to study the problems I wanted and trusted that I would eventually learn what was needed. Lastly, throughout our time working together, Julie has always been personally – as well as professionally – supportive in a way that allowed me to feel secure in my academic home. I am particularly grateful for Julie's personal support through the years. Thank you.

Beyond Julie, I thank my committee members – Roger Levy, Ellie Pavlick, and Been Kim – and thesis readers – Jacob Andreas and Ted Gibson – for their deep involvement in my academic interests and thesis work. In selecting committee members and readers, I prioritized both technical excellence and personal qualities of curiosity and kindness. Each person listed here has written works that deeply influence how I think about language and AI, and each has spent valuable time with me to talk about my own research. I am particularly grateful and indebted to Roger for working with me so closely despite having no formal affiliation with me. Thank you.

Outside of the formal structure of my thesis, I thank my labmates, friends, and collaborators for shaping how I think in both technical and non-technical ways. It is impossible to draw exact distinctions between labmates, friends, and collaborators – some collaborators became friends, and some friends became collaborators. Outside MIT, other friends provided needed balance to my everyday research concerns. Thank you.

Lastly, I thank my family most of all. Thanking my family for help during my thesis is like thanking the sun or the air; without them, certainly, a particular finding or figure in my thesis would not be possible, but neither would so many more important things.

Nevertheless, in the narrow context of this thesis and academia, I am thankful to my family. I thank my parents for supporting me academically, in particular by always prioritizing education. I thank my sister and brother in law for modeling human-centered careers across fields from literature to medicine. I thank Regina for always obeying a strong moral vision while pursuing a technical degree, and for being a wonderful partner with whom to speak about my own studies. Lastly, I thank little baby Arlen for being the spark of joy we all need. Thank you.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Natural language, while central to human experience, is not uniquely the domain of humans. In recent years, AI models, often parameterized as neural networks, have demonstrated impressive language capabilities. For example, AI agents, trained in cooperative settings, can learn their own communication systems that serve as a simplified language [Lowe et al., 2017, Wang et al., 2020, Sukhbaatar et al., 2016a]. At the same, other AI models, trained on natural language corpora, exhibit impressive word prediction and generalization capabilities and can seemingly converse with humans [Devlin et al., 2019, Brown et al., 2020].

What are such AI models learning about language? Just as linguists and pyscholinguists consider fundamental properties of language and how humans process it, I seek to better understand AI language processing, especially compared to theories of human cognition. In this thesis, I contribute to two areas within the field of language AI systems (Figure 1-1). First, I show how cognitively motivated pressures for simple but useful lexicons can induce more "human-like" communication among AI agents. Second, I develop tools to analyze pre-trained large language models (LLMs) and show that some models use representations of sentence structure in making predictions. Thus, my two contributions in this thesis are complementary, offering on the one hand control over training new systems, and on the other hand insight into studying existing systems.

The remainder of this chapter includes a brief summary of relevant aspects of computational linguistics (Section 1.1) and executive summaries of the contributions I have made in

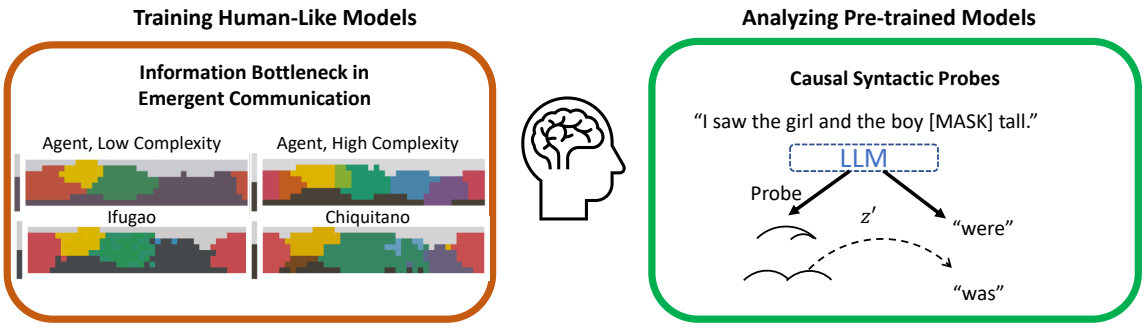


Figure 1-1: I seek to induce more human-like communication (left; Chapter 2) and analyze pre-trained language models for human-like processing (right; Chapter 3). Each high-level field of training or analyzing contains many specific problems; I focus on training information-theoretic methods in emergent communication and analyzing the use of syntax in pre-trained models.

this field (Sections 1.2 and 1.3). Each executive summary is then expanded into technical chapters on information-theoretic emergent communication (Chapter 2) and causal probing of LLMs (Chapter 3). I conclude with a brief summary of my work and directions for future research.

1.1 Related Works

In this thesis, I seek to train and analyze language-related neural networks, using principles from cognitive science. This work is situated within the broader fields of computational linguistics and AI, wherein researchers seek to develop computational models for producing and processing language. On the one hand, some researchers consider how to build new AI tools to act, at least in some way, like humans in language processing tasks. Large Language Models (LLMs), for example, are a type of AI model trained on next-word prediction tasks to produce human-like language [Brown et al., 2020, Touvron et al., 2023]. On the other hand, other researchers analyze pre-trained models to assess how they behave, often in comparison to humans [Linzen et al., 2016, Giulianelli et al., 2018]. For example, one may ask whether a model trained on a next-word prediction objective inherently learns rules governing syntax. These two sub-areas of research are complementary: affording new techniques to train models and new tools to understand them. Here, I explore some of the

major trends in these sub-areas of research that I build upon in the remainder of the thesis.

1.1.1 Developing Computational Models of Language

A longstanding goal in computational cognitive science has been the development of computer models that replicate important aspects of human language processing. Beyond simply observing how people use language, computational models distill predictions into testable mathematical theories of language. If a computational model accurately predicts human behavior, it may reveal important insights into human mental processes – conscious or unconscious. Of course, depending upon the linguistic behavior being predicted (e.g., word production, phonetic classification, etc.) different computational models or data may be used.

While classical computational modeling research has explored a variety of modeling methods, neural networks have become broadly popular in recent years. In general, neural networks, comprising a large number of artificial "neurons" made up of learnable weights and functions, are over-parametrized function approximators that, with enough training data, can at least in theory approximate diverse functions. Typically, neural networks are initialized with random parameter values that are updated according to training data. In practice, different neural network implementations tend to learn some behaviors from data better than others. For example, recurrent neural networks (e.g., [GRUs Cho et al., 2014], [LSTMs Hochreiter and Schmidhuber, 1997], etc.) generate fixed-size representations of lengthy inputs by iteratively incorporating more context into a hidden representation. While in some ways cognitively plausible (after all, humans remember a long series of input in a fixed representation space – the brain), such recurrent neural networks often fail to represent information over very long contexts. More recently, transformer-based neural networks, using self-attention mechanisms that support efficient and selective contextualization, have supplanted recurrent neural networks [Vaswani et al., 2017, Devlin et al., 2019]. Thus, while the exact form may vary, neural networks, and in particular large, transformer-based neural networks, are the current standard model to train using linguistic data.

Complementing model choice, the training data and objectives used in conjunction with

learning-based models play important roles in inducing human-like language processing. An untrained neural network alone is a poor model for language processing; models are trained to learn desired behaviors by optimizing the weights of neural nets to minimize the expected loss over some dataset. Depending upon the desired behavior, the training data and objective may vary. For example, some models may be trained using carefully curated text pulled from articles in the Wall Street Journal [Marcus et al., 1993], while others exploit the benefits of large data sources by using hundreds of billions of words [Devlin et al., 2019, Brown et al., 2020]. Similarly, the training objective may vary: some models are trained to predict a hidden word in the middle of a sentence while others may be trained to answer questions by highlighting a portion of text [Rajpurkar et al., 2016]. Next-word prediction, wherein models are trained to predict the next word, given previous words, is a particularly common training objective [Brown et al., 2020, Touvron et al., 2023]. Some methods appear to improve language model performance according to computational metrics but are clearly not cognitively plausible (e.g., infants learn the rudiments of language from far fewer linguistic inputs than the language models use during training). Such variation in training data and objective reveal the open nature of current research in language modeling.

Complementing standard language modeling approaches such as next-word prediction, and of particular relevance to this thesis, research in Emergent Communication (EC) considers an alternative training objective to encourage agents to learn grounded communication that may not align with natural language [Lowe et al., 2017, Havrylov and Titov, 2017]. In standard EC work, artificial agents, parametrized as neural networks, are trained in simulated settings and, by training to maximize a shared reward function, learn to communicate. Just as two humans who do not speak any languages in common may learn a simple vocabulary for cooperating towards a common goal, agents their own communication. As in standard language modeling research, EC research must consider the important role of model class (e.g., type of neural net), data (e.g., training environment), and objective (e.g., task reward). Emergent communication differs substantially from language modeling, however, by inducing fundamentally grounded communication: agents communicate about (simulated) objects. At the same time, EC agents necessarily do not communicate via natural language, as they lack human data. Thus, emergent communication research

offers an alternative computational model of communication.

In Chapter 2, I explain a framework for developing more "human-like" communicative agents by expanding traditional EC approaches with cognitively-motivated training objectives. My work is broadly similar to many of the works summarized in this section (developing computational models of language) and more narrowly related to grounded emergent communication research.

1.1.2 Analyzing LLMs with Interpretability Tools

Complementing the work described in the prior section, some researchers seek to better understand pre-trained LLMs from a human-centered perspective. This work falls under the broad umbrella of AI interpretability or explainability, wherein researchers develop methods for understanding AI models (typically, neural networks) in human-understandable terms. Generally, interpretability research complements model design research much as a sensor does an actuator; interpretability tools can be applied to trained AI models to reveal what behaviors they have learned. Here, I broadly characterize LLM analysis in two categories depending upon whether analysis is based upon surface level features or latent representations.

Using feature-level explanations, researchers seek to explain model behavior with respect to the raw inputs fed into models. Generally, feature-attribution methods reveal how model decisions would change as individual features change [Ribeiro et al., 2016, Lundberg and Lee, 2017, Selvaraju et al., 2017]. For example, natural language classification decisions may be explained via linear combinations of words (e.g. the word "good" may contribute to a model classifying a review positively), or image classifications may be explained by highlighting pixels that, if altered, would change the classifier's prediction. While the utility of such feature-attribution explanations for human understanding remains debatable [Zhou et al., 2022a,b], it is generally true that such explanations are by definition limited to only provide explanations with respect to input features.

In contrast to feature-level explanations, other works seek to explain model behavior with respect to latent concepts. Many "interpretable-by-design" neural networks, for ex-

ample, embed inputs into learnable prototypes that capture high-level concepts that a single pixel or word cannot express (e.g., a beak of a bird). To the extent prototypes align with human-interpretable concepts, these prototypes therefore provide latent-concept explanations. When faced with traditional black-box models that do not use such interpretable representations, researchers may nevertheless still employ latent-space explanations by exploring what information models encode and (optionally) how model decisions change with respect to such information. Latent concept explanations are inherently generalizations of feature-space explanations and can, in theory, provide more meaningful explanations by invoking high level concepts. This flexibility comes at the cost, however, of difficulty in identifying how latent concepts are encoded in high-dimensional latent spaces.

In Chapter 3, I describe my work that falls within the broad field of LLM analysis, in which I consider how representations of syntactic structure mediate pre-trained model predictions. I consider standard pre-trained LLM models (thus, I do not assume I can train an interpretable-by-design language model) and analyze them with respect to a latent structure (syntax).

1.2 Training Communicative Agents: Emergent Communication

1.2.1 Goal: Human-Like Emergent Communication

In emergent communication literature, cooperative AI agents are often trained to in partially-observable environments to maximize a shared reward function specifying a desired goal or behavior [Lowe et al., 2017, Lazaridou et al., 2018, Kottur et al., 2017, Havrylov and Titov, 2017]. By learning to optimize this reward function, agents automatically learn grounded communication with no supervision. Unfortunately, even in simple settings, such emergent communication is quite different from human languages or communication [Chaabouni et al., 2019, Eccles et al., 2019, Lin et al., 2021, Kottur et al., 2017].

Emergent communication and human naming systems often differ in their complexity – a measure of how much information in bits speakers encode in words. Across languages

and semantic domains, human naming systems appear at least partially guided by a pressure for lower complexity in an information-theoretic sense [Zaslavsky et al., 2018, 2021, Mollica et al., 2021, Zaslavsky et al., 2019]. For example, while people could in theory communicate about colors by describing precise RGB values, natural languages typically employ high-level words (e.g., "red" or "blue" in English). Such behavior of using high-level words to represent complex inputs is consistent with Information-Bottleneck (IB) pressures driving human language evolution. Unlike humans, traditional emergent communication agents are rarely trained with such pressures.

In my work, presented in Chapter 2, I seek to build AI agents that automatically learn human-like emergent communication by including cognitively motivated pressures. Training with such pressures provides a computational testbed for theories of language evolution: if certain pressures lead to human-like naming systems among artificial agents, that provides some evidence that such pressures may guide human language evolution.

1.2.2 Approach: Information-Theoretic Emergent Communication

To train human-like emergent communication systems, I make two technical contributions: 1) I describe an Information-Theoretic Emergent Communication (ITEC) framework and 2) I propose a family of neural network methods that perform well within this framework.

In my first contribution, I describe a cognitively-motivated ITEC training framework that trades off utility, informativeness, and complexity. Utility corresponds to how well a team of agents can accomplish a task using learned communication; traditional emergent communication literature often uses utility or reward maximization to drive communication. Unlike utility, informativeness is a task-agnostic measure of how well a listener can understand a speaker, regardless of context. Lastly, complexity, which we measure as the mutual information between a speaker's meaning and its communication, reflects roughly how many bits are allocated for communication. Prior studies indicate that the IB tradeoff of maximizing informativeness while minimizing complexity explains aspects of human naming systems.

In my framework, I explore how different relative pressures for utility, informativeness,

and complexity lead to different communication systems. Broadly, I seek to maximize utility, maximize informativeness, and minimize complexity, regulated by scalar tradeoff parameters. Just as different human naming systems in different languages appear to correspond to different tradeoffs between informativeness and complexity, my framework allows me to train agents according to different tradeoffs among the three terms I consider.

While one can train a variety of neural network agents within my ITEC framework, in my second contribution, I propose a family of neural network methods for learning complexity-limited discrete emergent communication. My method, dubbed the Vector Quantized Variational Information Bottleneck (or VQ-VIB), combines notions of an Information Bottleneck (to limit complexity) with vector quantization (which supports learning a finite set of representations, embedded in a continuous space). Compared to other neural networks designs, VQ-VIB agents more naturally support variational bounds on complexity while learning meaningful embeddings spaces.

1.2.3 Results

I found that the ITEC framework afforded control over important characteristics of communication, and VQ-VIB models outperformed other neural network methods. Experiments were conducted in three domains (a color reference game, an open-domain image reference game, and a grounded communication game in a simulated 2D world), motivated by prior studies in human naming systems and emergent communication. Results were largely consistent across domains, indicating the wide applicability of my technical contributions. First, the ITEC framework allowed direct tuning of the utility, informativeness, and complexity of communication, which led to indirect control over high level properties such as convergence rate (how long it took for agents to learn a shared protocol) and similarity to human naming systems. Second, VQ-VIB models, by embedding discrete communication vectors within a continuous space, consistently outperformed other strong baselines trained in the ITEC framework.

Across domains, the ITEC framework allowed for direct and indirect control over important communication properties. Most directly, varying the tradeoffs between utility,

informativeness, and complexity in the ITEC optimization caused agents to learn communication systems with varying utility, informativeness, and complexity measures. Given the importance of these terms in human language evolution, it is important to control them in inducing human-like emergent communication. Beyond such direct goals, training in the ITEC framework allowed for indirect control over other aspects of communication. For example, increasing the pressure for informativeness caused agents to converge more quickly to successful communication that generalized well to novel inputs, while limiting the complexity of communication to match human-like levels was important for aligning with representations of natural language. Such results that link information-theoretic pressures to behavioral communicative changes reveal important computational evidence for how hypothesized pressures on human naming systems lead to observed natural languages.

Within the ITEC framework, I found that VQ-VIB models tended to outperform other neural network methods, likely due to the continuous embedding space VQ-VIB models use. Prior emergent communication literature has typically employed onehot vectors to represent discrete communication vectors that are all orthogonal and equidistant from each other. Conversely, VQ-VIB models embed learnable discrete tokens in a continuous space; in experiments, the relative positions of different tokens within this space suggests a semantic interpretation of the learned embeddings. The continuous embedding space also supported behavioral improvements for VQ-VIB models such as greater informativeness and more human-aligned representations.

1.3 Analyzing Pre-trained Language Models: Causal Probing for Syntax

1.3.1 Goal: Analyzing Pre-trained Models

Complementing research on training methods to induce more human-like communication, other researchers analyze pre-trained models. In recent years, large neural networks, trained to predict a word given that word's context (e.g., preceding words in a sentence), have demonstrated impressive human-like performance at many tasks. Despite some be-

havioral similarities, however, it is unclear if such trained models use similar representations of language to those that humans do.

Recent works in probing pre-trained models indicate that large language models (LLMs) encode some syntactic information in their internal representations, but it is unclear if such representations mediate model predictions. Generally, probing literature seeks to expose learned patterns of a neural language model by training small neural networks to map from model representations to human-interpretable properties [Alain and Bengio, 2017, Conneau et al., 2018, Reif et al., 2019, Giulianelli et al., 2018, Stanczak et al., 2023, Tenney et al., 2019]; syntactic probes are trained to predict syntactic properties such as plurality from an embedding [Giulianelli et al., 2018] or positions in a dependency structure [Hewitt and Manning, 2019]. Probes are often intentionally parametrized as simple models to prevent probes from "doing too much;" this reflects a fundamental desire to only probe for salient properties in representations rather than building entirely separate models [Hall Maudslay et al., 2020a, Belinkov, 2021]. Unfortunately, such probes are fundamentally limited to only perform correlative, as opposed to causal, analysis [Amini et al., 2023]. That is, probes can reveal what information is present in model representations but now how such information is used in making predictions.

In Chapter 3, I seek to answer how (and if) representations of syntax in mediate model predictions. Answering this question would provide important insight into the narrow topic of how hierarchical representations of syntactic structure, which are thought to be fundamental to human language processing [Chomsky, 1965], are used in language models. More generally, syntactic structure may be viewed as a latent property of language; the techniques I develop to study representations of syntax may be broadly applied to a variety of other latent properties such as tone or intent.

1.3.2 Approach: Causal Probing via Dropout Probes

To assess how representations of syntax mediate model predictions, I develop a gradient-based intervention method to create counterfactual latent representations, and I introduce a new type of probe for improved causal analysis. Overall, my causal analysis comprises

three steps: 1) generating a syntactically-ambiguous input, 2) using a probe to create counterfactual representations corresponding to different parses of the same input, and 3) recording model predictions for the counterfactual representations. While one can use existing probing methods within the second step of my framework, I propose a novel type of "dropout probe" to address aspects of redundancy in model representations.

Causal Probing In the first step, I design ambiguous test suites that support multiple syntactic interpretations, which should induce different model predictions. For example, consider a sentence, "I saw the girl and the boy [MASK] tall," where "[MASK]" is a special token that denotes a masked word the model must predicted. This sentence supports different structural interpretations: should one treat "the girl and the boy" as one joint noun phrase, or should one treat "I saw the girl" as a standalone phrase, with "the boy" starting a new phrase? Depending upon ones structural interpretation, one should predict different verbs in the [MASK] spot: "I saw (the girl and the boy) were tall" vs. "(I saw the girl) and (the boy was tall)." More generally, I generate sentences with different structural ambiguities that should therefore affect model predictions.

In the second step, I generate counterfactual latent representations corresponding to different syntactic interpretations of a single sentence. I assume access to a language model, a trained syntactic probe (I propose a new family of probes inspired by those introduced by Hewitt and Manning [2019]), and two syntactic structures for each sentence. Using the model, probe, and desired structures, I create a latent representation for each sentence at model layer k, from which I generate two counterfactual representations. Each counterfactual is generated via gradient descent in the latent space: starting with the original latent representation of a sentence, the latent is updated according to the gradient of the loss between the probe's prediction and a desired syntactic structure. In the example from the previous paragraph, one desired structure would correspond to "the girl and the boy" being a single noun phrase, whereas the other structure would be a coordination of sentence phrases. (This gradient-descent process is explained more fully in Section 3.3.2.) Using the two syntactic structures generates two distinct latent representations, each corresponding to, intuitively, what the model's representation of a sentence would be if the syntactic

structure were specified.

In the third and final step, I record model predictions for each of the counterfactual representations generated in the previous step. To assess model predictions for counterfactual representations, I passed each counterfactual through the remaining layers of the pre-trained model. By design, different syntactic interpretations of the sentence afford different valid predictions (e.g., "was" or "were" in the running example). I therefore measured whether model predictions systematically shifted in valid ways for one counterfactual representation of another.

Dropout Probes The causal probing framework described above uses probes to generate counterfactual representations; while one can use probes introduced in prior art (e.g., [Hewitt and Manning, 2019]), I introduce a novel type of probe to better detect redundantly encoded syntactic information. For example, if a language model encodes the same syntactic information in multiple neuron activations, a probe may only learn to use a single copy of that information, which would lead to (gradient-based) counterfactual representations only updating some representations of syntax. I introduce "dropout probes" to mitigate this risk. By pre-pending a dropout layer that randomly masks probe inputs with probability α during training, I force high-α dropout probes to use all representations of syntactic structure in a model's latent representation. In causal probing experiments, therefore, I used dropout probes, trained with varying α values, to create counterfactual representations.

1.3.3 Results

Representations of syntax appear to mediate model predictions, although there is important variation across syntactic structure and language model. Furthermore, I found that models appear to redundantly encode syntactic information in their representations, validating the importance of dropout probes in causal probing.

The primary findings in Chapter 3 indicate that several models use representations of syntax in making predictions. For example, given the input "I saw the girl and the boy [MASK] tall," models tended to shift their predictions from "were" to "was" when using counterfactual representations to favor singular, rather than plural, verbs. Such behavior is

consistent with using representations of syntax in understanding and producing language. Beyond that specific sentence, I tested BERT-based models on a series of syntactic ambiguities and prediction tasks (e.g., predicting a masked word or answering a question). Although the magnitude of effects varied somewhat, for several models and syntactic suites, I found that models systematically changed their predictions in syntactically-expected ways.

My experiments further indicate the importance of dropout probes in causally probing language models. When using standard probes that lack dropout, as described in prior literature, the counterfactual representations tended to reveal only small changes in model predictions. Conversely, model predictions changes by over an order of magnitude more when using high-dropout probes (e.g., $\alpha = 0.8$) to create counterfactuals. Such results, in conjunction with further analysis confirming that models encode syntactic information redundantly, confirm the importance of using dropout probes.

1.4 Thesis Findings

While the two technical chapters in this thesis consider separate problems, they jointly indicate important progress towards more human-like AI systems.

First, in Chapter 2, I show how to train AI systems that communicate via human-like lexicons. An information-theoretic emergent communication framework, representing cognitively-motivated pressures for language evolution, leads to naming systems that closely resemble aspects of human communication. This is important evidence that such pressures could drive human language evolution. Beyond this broad framework, I also find value in a suite of neural network architectures that I propose – the Vector Quantized Variational Information Bottleneck – that outperforms standard neural networks. This technical contribution should inform future AI research.

Second, in Chapter 3, I find that some pre-trained language models automatically learn to use human-like latent concepts in reasoning about language. In particular, the BERT language model [Devlin et al., 2019], trained to predict randomly-masked words in a sentence, uses representations of syntactic structure in predicting such words. This is a novel finding that, beyond implications for syntactic processing alone, provides evidence that neural

networks can learn to use human-like representations.

Considered jointly, the findings in this thesis present two views of the best way to create more human-like AI systems. On the one hand, in some domains, carefully-designed losses based on cognitive insights are required to induce desired behaviors. On the other hand, with enough human-generated data, some models appear to already conform to some theories of human cognition. The tension between these terms underpins the generality of many of the ideas presented in this thesis. Training or analyzing human-like AI systems need not be restricted to language domains; ongoing and future research extend some of the core tools I propose in this thesis to new areas. In Chapter 4, I briefly discuss such efforts in inducing multi-task abstractions (training) or enforcing in-distribution causal probing mechanisms (analyzing). I am excited to see how the ideas I explore in this thesis inform and transform within future human-like AI research.

Chapter 2

Training New Models: Information-Theoretic Emergent Communication

In this chapter, I consider how to train communicative models to learn aspects of human-like communication using cognitive theories for driving human communication. Two fundamentally contrasting theoretical views are available to characterize optimization in linguistic communication and the resulting pressures on language evolution: on the one hand, task-specific utility maximization; on the other hand, a task-agnostic cognitive pressure for maximizing informativeness (how well a listener can understand a speaker) while minimizing complexity (roughly, how many bits are allocated for communication). Here, I integrate these two views (a utility-based framing for accomplishing tasks, with an IB pressure for efficiency) and propose a new information-theoretic framework for emergent communication that trades off utility, informativeness, and complexity. To train agents within my framework, I develop a method, called Vector-Quantized Variational Information Bottleneck (VQ-VIB), that allows agents to interact using information-constrained discrete communication embedded in a continuous vector space. I test this approach in three domains and show that pressure for informativeness facilitates faster learning and task-agnostic communication, which generalizes better to novel domains. At the same time, limiting complexity yields better alignment with actual human languages. Lastly, I find that VQ-VIB outper-

forms previously proposed emergent communication methods; I posit that this is due to the semantically-meaningful communication space that VQ-VIB affords. Overall, my work considers the role of cognitively-motivated pressures in inducing aspects of human-like communication among artificial agents. [1]

2.1 Introduction

How can language emerge from local interactions? One common answer to this question stems from a game-theoretic approach of language evolution that emphasizes *utility* maximization [Steels and Belpaeme, 2005, Still and Precup, 2012, Chaabouni et al., 2021b]. In this view, agents interact to increase a task-specific utility or reward function. For example, a passenger in a car who wishes to avoid crashing (low utility for crashing, high utility for driving safely) might learn to communicate to an inattentive driver about red, yellow, or green traffic lights. In computational experiments modeling language emergence, this utility-based framework has been shown to capture several important aspects of language [Lowe et al., 2017, Chaabouni et al., 2021a]; however, this framework is limited in several ways. Notably, while a utility pressure can induce task-specific communication, it is unlikely to lead to communication systems that generalize across different tasks. For example, a communication system that is tailored specifically to my traffic light setting might not necessarily support the emergence of a more general notion of "yellow" that can be generalized for productive use in other tasks, such as determining whether a banana is ripe; nor will optimization in the traffic-light scenario alone support communication of other colors (e.g., "blue"), which are irrelevant for traffic lights but are useful in other settings.

Complementing a utility-based framing of language evolution, others propose that language is driven by a pressure for more task-agnostic efficient communication [for reviews see Kemp et al., 2018, Gibson et al., 2019]. Most relevant to my work is a recent body of literature that shows that human language is characterized by near-optimal compression of meanings into words [Zaslavsky et al., 2018]. Necessarily, languages that employ a

[1] Much of the work in this chapter is described in Tucker et al. [2022b, Under Review].

finite set of words to describe infinite items lose information. For example, when categorizing colors, a language that employs 10 words to describe all colors will discard subtle information about shades or brightness; at the same time, the 10 color words retain more salient information, such as distinguishing between yellow and black. Thus, whether describing colors or other attributes, human naming systems are a form of lossy compression. Zaslavsky et al. [2018] have characterized this lossy compression mechanism in human languages as an information bottleneck (IB) tradeoff [Tishby et al., 1999] between the informativeness (roughly, how well a listener can understand a speaker, regardless of context) and complexity (roughly, how many bits are allocated for communication) of the lexicon. Increasing informativeness can lead to more specific naming systems (e.g., color words like "crimson" or "vermillion") but comes at the cost of greater complexity (e.g., having to learn a larger lexicon). More generally, in any domain, there is a theoretical maximum informativeness that is achievable for a given complexity; in a wide variety of human languages and semantic domains, languages approach this maximum informativeness value for their complexity levels [Zaslavsky et al., 2018, Mollica et al., 2021, Zaslavsky et al., 2019, 2021]. In other words, for a fixed complexity budget, languages convey as much information as possible. At the same time, different languages exhibit different complexity levels, representing different tradeoffs between complexity and informativeness.

These two frameworks for communication – utility maximization and IB efficiency – represent different interpretations of what drives language: I emphasize that utility and informativeness are distinct concepts. Utility is a task-specific measure, while informativeness represents a task-agnostic pressure for conveying meanings. In my traffic light example, a utility-driven speaker might refer to yellow lights as "red" to force a driver to slow down; this accomplishes the desired goal of avoiding crashes (high utility) but may lead the driver to think the traffic light is actually red (low informativeness). Conversely, an IB-optimal speaker might refer to yellow and green lights as a single word, given the visual similarity between their shades; such communication might be IB-efficient, but it ignores safe driving goals.

While prior investigations into utility and IB pressures on naming systems illustrate important aspects on human communication, several important questions remain largely

unaddressed. First, it is unclear what type of agent utility functions and learning dynamics may lead to naming systems that are near-optimal in the IB sense. Small changes in the training environment can lead to large differences in learned communication [Chaabouni et al., 2021a] and even in the same environment agents often randomly converge to different communication protocols corresponding to different complexity levels [Kågebäck et al., 2020, Carlsson et al., 2021]. More dramatic changes in training, such as iterated learning to simulate cultural transmission, likewise tend to alter information-theoretic aspects of communication in unclear ways [Carlsson et al., 2023]. Second, applications of the IB framework to naming systems have focused on finite domains in which exact calculations of the IB tradeoff is tractable. Therefore, it has been unclear how to apply this framework at scale, and in particular, in large open domains.

In this chapter, I address these open questions by integrating utility-based and IB-based approaches to language evolution. Specifically, I develop a scalable computational framework for training artificial intelligence systems to communicate while guided by pressures for communicative utility, informativeness, and complexity. This framework builds upon prior literature in Emergent Communication (EC). In traditional EC work, artificial agents (typically, neural networks) are trained in cooperative settings to accomplish some goal and, by endowing agents with the ability to broadcast vectors to each other, agents learn to communicate with each other. For example, in a simulated driving environment, agents learn to warn other agents when they approach an intersection, thus preventing collisions [Sukhbaatar et al., 2016a]. Critically, communication emerges only due to a utility-based framing (e.g., avoid crashes while driving) rather than direct supervision of how to communicate.

In contrast to the utility-based framework of prior EC works, I adopt the cognitively-motivated terms for informativeness and complexity while training EC agents and propose a utility-informativeness-complexity framework for emergent communication. Agents are trained to simultaneously maximize utility (how well they do at a task), maximize informativeness, and minimize complexity. As in classic IB, these terms may be in conflict, so agents optimize a tradeoff between these three quantities. This general framework captures important notions of communication, such as goal-directedness, context-independent in-

formation transfer, and limited computation. Unlike exact IB calculations, training agents according to this framework is highly scalable.

I propose a novel neural network method to train within my framework: the Vector-Quantized Variational Information Bottleneck (VQ-VIB), for complexity-limited discrete communication. Deterministic neural networks are incompatible with the information-theoretic notions of complexity or informativeness in my framework. Conversely, VQ-VIB and other stochastic methods support variational bounds on these quantities, which allow us to reward or penalize complexity and informativeness. In experiments, VQ-VIB methods outperform other neural architectures by generating a discrete set of representations in a continuous space, much like word embeddings in Natural Language Processing literature [Pennington et al., 2014].

Beyond the specifics of VQ-VIB, I demonstrate the flexibility of my framework and the importance of each term it contains. Utility-based training enables agents to solve particular tasks but fails to generalize to novel settings (e.g., when trained and tested on semantically distinct inputs). Highly informative communication allows agents to learn faster and generalize better to harder tasks, but such communication is often overly-complex compared to human communication. Thus, limiting the complexity of communication is necessary to induce aspects of human-like communication.

2.2 Background

In this chapter, I present an information-theoretic emergent communication (ITEC) framework that combines task-specific utility with task-agnostic informativeness and complexity pressures. Furthermore, I develop VQ-VIB, a method that fits within my framework. In this section, I review the relevant technical background. First, I review IB systems, including IB analysis measuring the tradeoff between complexity and informativeness in human naming systems, and variational methods for training neural nets with the IB objective. Second, I explain the technical details of the Vector-Quantized Variational Autoencoder (VQ-VAE), a neural network architecture for learning discrete representations in a continuous space; my VQ-VIB method combines ideas from VQ-VAE with concepts from IB.

Third, I summarize related literature in Emergent Communication, highlighting how prior literature often uses a utility-based framework for inducing communication.

2.2.1 The Information Bottleneck for Semantic Systems

My work extends Zaslavsky et al. [2018]'s information-bottleneck framework for semantic systems, which I review in this section. In this framework, a speaker and listener optimize the IB tradeoff between informativeness and complexity of communication. [2]

In IB semantic systems, I assume a probabilistic source over meanings that a speaker wishes to encode: $m \sim \mathbb{P}(m)$. A speaker is characterized as a probabilistic encoder mapping from m to communication w: $q(w|m)$. Conversely, a listener seeks to recover a reconstructed meaning from communication: $\hat{m} = q(m|w)$.

Within this framework, one may measure the complexity and informativeness of communication. Complexity is measured as the mutual information between the speaker's inputs and communication: $I(m; w)$. Intuitively, this corresponds to the number of bits allocated for communication, although it is more general than deterministic discrete coding schemes. Simultaneously, informativeness corresponds to notions of similarity between the speaker's and listener's belief states and can be measured via the negative expected Kullback-Leibler (KL) divergence: $-\mathbb{E}[D_{\mathrm{KL}}[m\|\hat{m}]]$. Lower KL divergence values arise from more similar distributions, so decreasing the KL divergence leads to a listener "understanding" a speaker better.

Zaslavsky et al. [2018] propose that, in natural language naming systems, the speaker and listener jointly optimize the encoder and decoder functions, $q(w|m)$ and $q(m|w)$, according to a tradeoff between complexity and informativeness, modulated by a scalar parameter $\beta > 0$:

$$\text{minimize} \quad I_S(m; w) + \beta \mathbb{E}_S[D_{\mathrm{KL}}[m\|\hat{m}]] \tag{2.1}$$

The tradeoff parameter, β, represents the importance afforded to the informativeness

[2] Zaslavsky et al. [2018] use the term "accuracy" where I use the term "informativeness;" in some of my experiments in which I train agents, there is a notion of team accuracy that is measured by a utility function, so "informativeness" better illustrates the distinction between these terms.

term relative to the complexity term. For small β, systems converge to low-complexity and low-informativeness communication; for large β, speakers and listeners tolerate greater complexity to achieve increased informativeness.

This theoretical IB tradeoff for semantic systems yields two important insights: 1) at least in simple systems, one can derive optimal communication schemes for varying β, and 2) one can compare human naming systems to IB-optimal systems by measuring their informativeness and complexity values. In a variety of works, covering diverse semantic domains (e.g., colors, pronouns, containers, etc.) and hundreds of languages, human languages are consistently near-optimal in the IB sense, while exhibiting different β tradeoff values [Zaslavsky et al., 2018, 2021, Mollica et al., 2021, Zaslavsky et al., 2019]. That is, each language achieves near-maximal informativeness for its complexity level, but different languages settle on different complexity levels.

2.2.2 The Variational Information Bottleneck

While exact computation of IB tradeoffs is possible in simple systems, typical approaches fail to scale to complex settings; the Variational Information Bottleneck (VIB) is a scalable variational approximation method for IB. As introduced by Alemi et al. [2017], VIB comprises a stochastic neural network encoder and decoder. In VIB, a neural encoder with weights θ, $q_\theta(z|x)$, maps input x to parameters of a $d-$dimensional Gaussian distribution – $\mu(x)$ and $\Sigma(x)$ – from which a continuous latent variable, $z \in \mathbb{R}^d$, is sampled. A neural decoder with weights ϕ, $q_\phi(y|z)$, reconstructs a target feature, y, from z. In standard VIB literature, the encoder and decoder are jointly trained according to a tradeoff between decoder accuracy and the complexity of representations but, rather than using $I(x;z)$ directly, VIB uses a variational bound on complexity:

$$I_{q_\theta}(x;z) \leq \mathbb{E}\left[D_{\mathrm{KL}}[q_\theta(z|x)\|r(z)]\right] \tag{2.2}$$

which holds for any distribution $r(z)$ (typically set to $\mathcal{N}(0, I_d)$). Overall, therefore, the VIB objective is:

$$\text{maximize}_{\phi,\theta} \quad I_{q_\phi}(z;y) - \beta I_{q_\theta}(x;z) \tag{2.3}$$

$$\leq I_{q_\phi}(z;y) - \beta \mathbb{E}\left[D_{\mathrm{KL}}[q_\theta(z|x)\|r(z)]\right] \tag{2.4}$$

$$\tag{2.5}$$

This closely resembles Zaslavsky et al. [2018]'s IB framework for semantic systems, with tradeoffs between complexity and informativeness (although note some flipped signs and the role of β, due to terms for informativeness vs. distortion, and maximization vs. minimization). Unlike in standard IB, however, VIB uses variational bounds for complexity and, depending upon the predicted feature, y, for informativeness as well. The flexibility of VIB has in turn supported applications of Information Bottleneck methods across disciplines, including economics [Aridor et al., 2024] and modeling of human intelligence [Malloy, 2022] .

2.2.3 Vector-Quantized Variational Autoencoder

While VIB generates complexity-limited continuous representations, the Vector-Quantized Variational Autoencoder [VQ-VAE, van den Oord et al., 2017] generates discrete, but potentially highly complex, representations. VQ-VAE models comprise a neural network encoder and decoder, mediated by a $d-$dimensional latent space. A codebook of K vectors, $\zeta_i \in \mathbb{R}^d, i \in [1, ..., K]$, defines a set of learnable discrete representations within the latent space. To generate a latent representation of an input, x, a deterministic encoder maps from x to a continuous latent representation $z(x) \in \mathbb{R}^d$, which it then discretizes by selecting the index of the closest element of the codebook: $i = \mathrm{argmin}_j \, ||z(x) - \zeta_j||^2$. The final discrete representation is this closest element, $\zeta_i(x)$. Given $\zeta_i(x)$, a deterministic decoder network seeks to reconstruct the encoder's input.

During training, the weights of the encoder, the decoder, and the codebook are updated using gradient descent; I represent these weights as Θ. Passing gradients through the non-differentiable discretization process (specifically, the `argmin` operation) is challenging; VQ-VAE uses a straight-through estimator, a common method for estimating gradients

through non-differentiable processes. VQ-VAE is trained according to the loss function in Equation 2.6, combining the evidence lower bound (ELBO) with two vector-quantization terms that encourage continuous embeddings and codebook elements to cluster.

$$l_{\text{VQ-VAE}} = \log p(x|\zeta_i(x);\Theta) + \|\texttt{sg}[z(x)] - \zeta_i(x)\|^2 + \alpha\|z(x) - \texttt{sg}[\zeta_i(x)]\|^2 \tag{2.6}$$

Here, the first term represents the evidence lower bound (ELBO), a measure of the estimated likelihood of training data. The second and third terms are clustering losses (using `sg` to stand for the `stopgradient` operator) that cause continuous embeddings and codebook elements to move closer together in the latent space. Lastly, α is a scalar tradeoff hyperparameter controlling the relative importance of clustering terms, typically set to 0.25 [van den Oord et al., 2017].

Overall, VQ-VAE models learn discrete representations in a continuous space (the codebook elements) that enable high-quality reconstructions of inputs. My work builds upon aspects of the VQ-VAE architecture but differs both in training objective and implementation. By introducing complexity bounds in training and using a different neural architecture to support such bounds, my VQ-VIB methods allow me to vary the complexity of communication, which I show enables more human-like communication.

2.2.4 Emergent Communication

My work combines ideas of IB and discrete representation learning (reviewed in the previous two sections) with the utility-based framing of emergent communication. In traditional EC work, agents are trained in cooperative multi-agent environments to maximize a utility function (sometimes called the reward) [Lowe et al., 2017, Lazaridou et al., 2018, Kottur et al., 2017, Havrylov and Titov, 2017]. In partially-observable environments, endowed with "cheap-talk" channels that allow agents to broadcast vectors to each other, agents often learn to communicate relevant information to each other. For example, if one agent can see a goal location in a 2D world, and another agent can move in the world (with utility based on proximity of the second agent to the goal), communication may "emerge" by

the first agent learning to broadcast information about goal location and the second agent simultaneously learning to interpret such communication [Lowe et al., 2017, Wang et al., 2020]. Crucially, communication emerges based only upon task performance rather than explicit supervision of how to communicate or what to communicate about.

The utility-based EC framework is a powerful mechanism for inducing task-specific communication: in various settings, agents may coordinate in simulated environments to find target locations [Lowe et al., 2017], to avoid collisions in simulated road intersections [Sukhbaatar et al., 2016b], or to refer to a specific photo among a set of images of faces [Chaabouni et al., 2021b]. Despite the flexibility of such methods, traditional EC agents often exhibit undesirable properties, such as being overly-complex [Chaabouni et al., 2019], slow to converge [Eccles et al., 2019, Lin et al., 2021], or unable to generalize to novel inputs [Kottur et al., 2017]. As an example of overly-complex communication, whereas humans might use only a small number of words to categorize colors, agents might output a distinct communication vector for each color.

Based in part by the desire to create more "human-like" EC, some recent works seek to induce complexity-limited discrete communication (much as words are complexity-limited discrete communication in natural languages) [Kottur et al., 2017]. In recent discrete EC works, agents communicate via onehot vectors, and the dimensionality of these vectors specifies the maximum vocabulary size [Lowe et al., 2017, Chaabouni et al., 2021b, Rita et al., 2020]. In some works, therefore, authors decrease the vocabulary size to a small number k, which sets a maximum communication complexity limit of $\log_2(k)$ bits. Beyond such hard-coded limits on complexity, two recent studies of discrete EC add corrupting noise to communication which affects communication complexity through environmental, rather than agent architecture, choices [Tucker et al., 2021a, Kuciński et al., 2021].

Lastly, Lin et al. [2021] indirectly explores the role of informativeness in guiding emergent communication, by using a reconstruction loss to generate communication encodings. This reconstruction loss encourages communication to contain more decodable information and is closely aligned with notions of informativeness. The authors find that their method tends to induce faster convergence (i.e., the team performs well at a task faster) using their method, but they note that their method might induce unnecessarily complex

communication.

2.3 Technical Approach

My technical contributions are twofold: first, I introduce an information-theoretic emergent communication (ITEC) framework that incorporates a utility-based loss into the IB informativeness-complexity tradeoff; second, I propose a neural network method, named the Vector Quantized Variational Information Bottleneck (VQ-VIB), which may be trained within my framework.

2.3.1 Information-Theoretic Emergent Communication Framework

I propose the ITEC framework that extends traditional utility-based EC to include terms for informativeness and complexity. Consider a simple EC setting with two agents: a speaker and a listener (S and L), depicted in Figure 2-1 a. Given a global state, x, the speaker receives a (potentially noisy) partial observation of the state and encodes it as a meaning, m, representing a belief state or probability distribution over x. The speaker stochastically maps m to an output communication vector, w: $w \sim S(w|m)$. Based on w and its own partial observation of the state (o_l), the listener simultaneously reconstructs the meaning ($\hat{m}$) and takes an action $y \in Y$: $y \sim L(y|w, o_l)$. The utility of actions is measured as a function of the state and the listener's action, $U(x, y)$.

Figure 2-1 a shows not only how a speaker and listener can coordinate to learn a communication protocol, but also how functions of different terms in the communication process reflect important quantities like informativeness and complexity. Just as IB optimization depends upon the tradeoff between informativeness and complexity, regulated by a scalar parameter β, I consider a maximization of three terms: utility, informativeness, and complexity. Thus, the ITEC objective is:

$$\text{maximize} \quad \lambda_U \mathbb{E}[U(x, y)] - \lambda_I \mathbb{E}[D_{\mathrm{KL}}[m \| \hat{m}]] - \lambda_C I_S(m; w)\,, \tag{2.7}$$

where λ_U represents the scalar weight for increasing utility, λ_I for increasing informa-

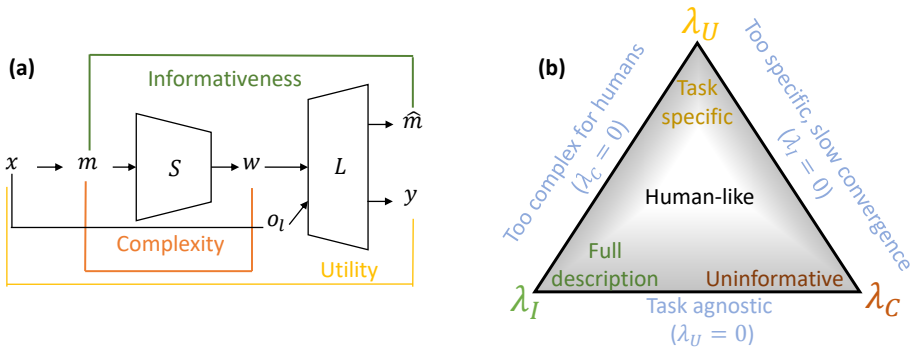


Figure 2-1: a) Measuring utility, informativeness, and complexity for a speaker and listener. b) Tradeoffs between these three terms control a variety of important communicative behaviors, and describe a region of human-like communication that balances competing pressures.

tiveness (or, equivalently, minimizing distortion), and λ_C for minimizing complexity.

The relative weights of these three scalar terms dictate important properties of optimal emergent communication, as depicted in Figure 2-1 b. For example, setting λ_C too high might lead to uninformative communication, but setting $\lambda_C = 0$ might lead to overly complex communication. Similarly, λ_U controls the task specificity of communication, and λ_I controls a task-agnostic measure of informative communication. Human communication likely optimizes a tradeoff between these three terms.

2.3.2 Vector-Quantized Variational Information Bottleneck

Equation 2.7 trades off terms for utility, informativeness, and complexity, but directly solving this maximization is intractable in large domains [Alemi et al., 2017]. Therefore, I train neural network agents to maximize a tractable variational bound of the same objective.

My method, named the Vector-Quantized Variational Information Bottleneck, or VQ-VIB, is a variational method for learning complexity-limited discrete representations in a continuous space. I was inspired by word embeddings: words in natural languages are complexity-limited, and word embedding methods represent words as a finite set of points in a semantically-meaningful space [Pennington et al., 2014, Mikolov et al., 2013].

Intuitively, VQ-VIB combine notions from the variational information bottleneck to

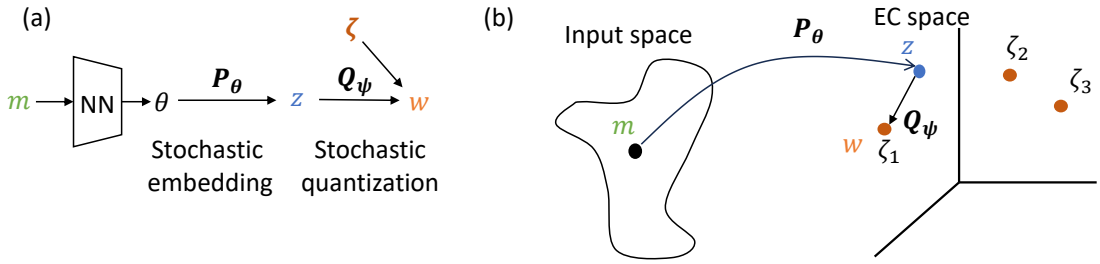


Figure 2-2: a) VQ-VIB employs a stochastic embedder (P_θ) that samples a continuous embedding, z, based on parameters extracted from a meaning, m, and a stochastic quantizer (Q_ψ) that maps z to a discrete word, w, using a codebook, $\boldsymbol{\zeta}$. The neural net mapping from m to θ, and the codebook $\boldsymbol{\zeta}$ are learnable parameters. b) The two-stage process first maps from a potentially high-dimensional input space into an embedding space, where discrete vectors, ζ, are used for quantization. In this chapter, I propose two implementations of VQ-VIB by providing different P_θ and Q_ψ.

limit complexity with vector quantization for discretization. While VQ-VIB borrows from prior art, it is distinct in several important ways. Unlike VQ-VAE, it uses a stochastic encoder, which is necessary for variational bounds on complexity, and unlike VIB, it generates discrete representations.

My general VQ-VIB method combines a stochastic embedding mechanism with stochastic discretization, as depicted in Figure 2-2 a. First, given a meaning, m, I generate parameters of a probability distribution, θ, from which a continuous latent embedding, z is sampled: $z \sim P_\theta(m)$. Next, z is quantized (i.e., set to a discrete vector) via a stochastic process to generate a discrete embedding, $w \sim Q_\psi(z)$. The neural network to produce θ, as well as the set of discrete embeddings, $\boldsymbol{\zeta}$, are parametrized by learnable weights.

The changes in representation formats in VQ-VIB are depicted graphically in Figure 2-2 b. First P_θ maps from a high-dimensional input space into the EC space. Within the EC space, I assume VQ-VIB models have K learnable codebook elements: $\zeta_i, i \in [1, K]; \boldsymbol{\zeta} = [\zeta_1, \zeta_2, ..., \zeta_K]$. (In the diagram, $K = 3$.) Lastly Q_ψ stochastically maps from the continuous representation, z, to one of ζ_i, to produce the final output w. In this chapter, I propose two different implementations of VQ-VIB, which differ by providing different implementations of P_θ or Q_ψ.

Vector-Quantized Variational Information Bottleneck – Normal Distribution

My first encoder architecture, named the Vector-Quantized Variational Information Bottleneck – Normal, or VQ-VIB$_\mathcal{N}$, draws its name from the normal distribution it uses for P_θ. That is, $z \sim \mathcal{N}(\mu(m), \Sigma(m)) \in \mathbb{R}^d$. This stochastic encoding process via a normal distribution is similar to standard VIB parametrizations [Alemi et al., 2017]. Next, VQ-VIB$_\mathcal{N}$ deterministically discretizes z by selecting the closest element of the codebook, $\boldsymbol{\zeta}$. That is, Q_ψ is set to the argmin operation over distance in the EC space. This quantization step is borrowed from VQ-VAE literature. Overall, the VQ-VIB$_\mathcal{N}$ encoder is a probabilistic function given by:

$$S_\mathcal{N}(w|m) = \mathbb{P}(w = \underset{\zeta_i \in \boldsymbol{\zeta}}{\operatorname{argmin}} ||z(m) - \zeta_i||^2) \tag{2.8}$$

where w is a discrete communication vector, m is the speaker's meaning, $z(m)$ is the continuous latent variable sampled from the Gaussian distribution, and $\boldsymbol{\zeta}$ is the codebook,

Vector-Quantized Variational Information Bottleneck – Categorical Distribution

My second encoder architecture, named the Vector-Quantized Variational Information Bottleneck – Categorical, or VQ-VIB$_\mathcal{C}$, differs from VQ-VIB$_\mathcal{N}$ by its implementations of P_θ and Q_ψ. First, P_θ is a deterministic function based on a feedforward encoder, rather than sampling from a Gaussian: $z(m) = P_\theta(m)$. VQ-VIB$_\mathcal{C}$ uses a stochastic Q_ψ, however, by sampling a discrete representation according to probabilities based on the negative Euclidean distance from z to each ζ_i. That is,

$$S_\mathcal{C}(w = \zeta_i|m) \propto e^{-||z(m)-\zeta_i||^2}, \tag{2.9}$$

which generates a probability distribution by normalizing for all $\zeta_i \in \boldsymbol{\zeta}$.

Overall, VQ-VIB$_\mathcal{N}$ and VQ-VIB$_\mathcal{C}$ reflect two different interpretations and implementations of the same overall idea. Both fit within the general VQ-VIB method by providing implementations of P_θ and Q_ψ. The difference in sampling mechanisms can be interpreted as uncertainty over a continuous semantic space (for VQ-VIB$_\mathcal{N}$) that is deterministically

discretized, or uncertainty over how to discretize a deterministically-generated continuous embedding (for VQ-VIB$_\mathcal{C}$). I note that future work may propose alternate VQ-VIB implementations that provide different sampling mechanisms; such methods still fit within my broader framework.

Lastly, I note that both VQ-VIB$_\mathcal{N}$ and VQ-VIB$_\mathcal{C}$ support a combinatorial token architecture that increases the effective codebook size of models without increasing the number of parameters in the network. Intuitively, rather than generating a single continuous representation in $\mathbb{R}^d$, an encoder can generate n representations in $\mathbb{R}^{\frac{d}{n}}$ and discretize each of those representations. Full discussion of this method, and diagrams of specific implementation architectures, are included in Appendix 5.1; conceptually this change in architecture does not alter the underlying variational bounds or discrete nature of encodings, it only changes the neural net implementation to support a greater number of discrete representations. Supporting a greater number of representations enables more complex and informative communication but, as I show in experiments, leads to less "human-like" communication.

Learning Objective

Regardless of the specific architecture, VQ-VIB agents can be trained according to variational bounds of the ITEC objective defined in Equation 2.7.

I used the same bound on informativeness (the second term in Equation 2.7) for both VQ-VIB architectures. Assuming that true states, $x \in \mathbb{R}^n$, are corrupted by zero-mean Gaussian noise with some variance Σ, belief states, m, are given by $m = \mathcal{N}(x, \Sigma)$. Under this assumption, $\mathbb{E}[D_{\mathrm{KL}}[m\|\hat{m}]] \leq \frac{1}{2}\mathbb{E}[||m-\hat{m}||^2] + \text{const}$. Thus, training a simple decoder, D, that outputs a reconstructed meaning based on communication, $\hat{m} = D(w)$, provides an upper bound on informativeness.

For complexity (the third term in Equation 2.7), I used architecture-specific variational bounds. For VQ-VIB$_\mathcal{N}$,

$$I_S(m; w) \leq I_S(m; z) \leq \mathbb{E}[D_{\mathrm{KL}}[q_\theta(z|m)\|r(z)]]] \tag{2.10}$$

The first inequality follows from the data-processing inequality, and the second follows

standard VIB bounds for any marginal distribution, $r(z)$. In my implementations, I set $r(z) = \mathcal{N}(0, \boldsymbol{I})$. Intuitively, therefore, for VQ-VIB$_\mathcal{N}$, I bottleneck the sampling process prior to discretization, which in turn limits the complexity of the downstream, discrete representation.

For VQ-VIB$_\mathcal{C}$, complexity is bounded via

$$I_S(m; w) \leq \mathbb{E}[D_{\mathrm{KL}}[q_\theta(w|m) \| r(w)]] \tag{2.11}$$

Note that this bound differs from the one for VQ-VIB$_\mathcal{N}$ by measuring the stochasticity the discretization process via Q_ψ, whereas VQ-VIB$_\mathcal{C}$ uses P_θ. Given the categorical distribution for $q_\theta(w|m)$, $r(w)$ represented a categorical prior over codebook elements, set in my experiments to a uniform distribution.

Combining terms for informativeness and complexity, the overall variational bound of the ITEC optimization in Equation 2.7 is

$$\text{maximize} \quad \mathcal{L}_{\text{var}} = \lambda_U \mathbb{E}[U(x, y)] - \lambda_I \mathbb{E}[||m - \hat{m}||^2] - \lambda_C I_{\text{var}}(m; w)\,, \tag{2.12}$$

where I_{var} is the architecture-specific variational bound on complexity, defined in Equation 2.10 for VQ-VIB$_\mathcal{N}$ and Equation 2.11 for VQ-VIB$_\mathcal{C}$.

Lastly, I actually trained VQ-VIB models by combining the ITEC variational bound, $\mathcal{L}_{\text{var.}}$ with prototype clustering losses from VQ-VAE methods (as introduced in Equation 2.6) and a tie-breaking entropy loss:

$$\text{maximize} \quad \mathcal{L}_{\text{var}} - \|\texttt{sg}[z(m)] - \zeta_i(m)\|^2 - \alpha\|z(m) - \texttt{sg}[\zeta_i(m)]\|^2 - \epsilon\lambda_C \mathbb{H}(w) \tag{2.13}$$

The final entropy term, $\mathbb{H}(w)$ represents the estimated entropy over the codebook; penalizing high-entropy communication (weighted by a small scalar value, ϵ, times λ_C) biased agents towards more human-like naming systems for a given complexity class. Further discussion of the entropy term is included in Appendix 5.2.

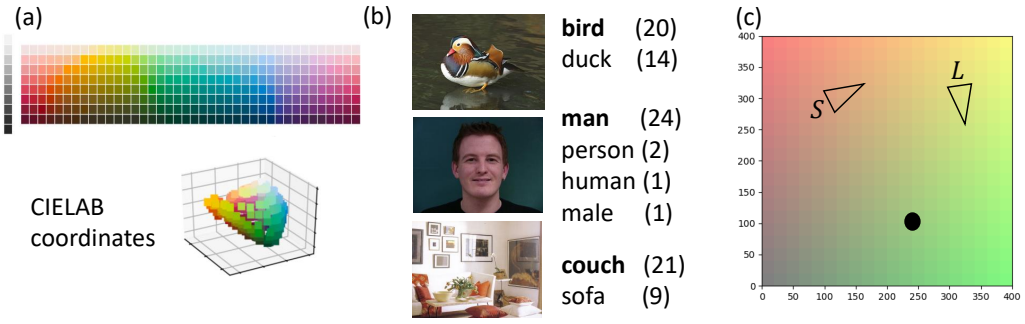


Figure 2-3: I conducted experiments in 3 domains: a color-naming reference game (a), a reference game of natural images, accompanied by human-provided names (b), and a 2D navigation environment (c).

2.4 Experiment Design

In a series of experiments, presented in the subsequent sections, I sought to characterize important aspects of information-theoretic emergent communication and my VQ-VIB method. First, could VQ-VIB agents learn IB-optimal communication, as in human naming systems in simple domains? Second, in open-domain communication, with an unbounded set of possible objects to refer to, what are the roles of informativeness and complexity pressures on generalization (the ability to refer to novel objects) and similarity to human languages? Third, beyond object-naming experiments, could artificial agents learn similar IB tradeoffs in simulated grounded settings?

I studied these questions in three experimental domains, depicted in Figure 2-3. Overall, I found that VQ-VIB agents, trained within the ITEC framework, learned similar complexity-informativeness tradeoffs to humans. In all domains, encouraging informativeness led to faster convergence (agents learning meaningful communication), but penalizing complexity was important for inducing more human-like communication. Lastly, I observed consistent trends across all three domains, indicating the generality and robustness of the ITEC training framework.

2.5 Experiment 1: Learning IB Naming Systems

In my first experiment, I considered whether VQ-VIB agents could learn IB-optimal communication systems in a color reference game, depicted in Figure 2-3 a, and whether doing so would induce aspects of human-like communication. Color-naming has been central to many cognitive theories of semantics, as languages must encode the continuous color spectrum into a finite set of words [Steels and Belpaeme, 2005, Berlin and Kay, 1969, Regier et al., 2007]. Data in the World Color Survey (WCS) demonstrate how 110 non-industrialized societies name color, providing a large corpus of human data [Kay et al., 2009]. Thus, I tested the ITEC framework for training agents to communicate about color, and compared the resulting EC agents to human naming systems and analytically-computed IB systems.

2.5.1 Experiment Setup

I trained a team of agents, comprising a speaker and a listener, in a reference game, or Lewis signalling game, as depicted in Figure 2-4 [Lewis, 2008]. In the game, a speaker observed a randomly-drawn target color (in the figure, a reddish-orange color), corrupted by Gaussian noise, and emitted communication, w. (The CIELAB representation of colors, as well as parameters for the observation noise, were motivated by prior work in human color perception [Mokrzycki and Tatol, 2012, Zaslavsky et al., 2018, Chaabouni et al., 2021a].) That communication vector was passed to a listener agent, comprising a decoder, D, that reconstructed the speaker's observation, and an actor, A, that observed the reconstruction as well as the target and a distractor color. The listener had to predict which of the two candidates was the target color.

Achieving high team accuracy (correctly identifying the target color) requires the speaker and listener to develop a shared understanding of communication about color. Given a complexity-limited discrete communication channel, agents should learn discretizations of the color space that maintain high accuracy for a given complexity level.

Throughout the experiments, I set $\lambda_U = 1.0, \lambda_I = 1.0$; this tended to lead to highly accurate team performance for low λ_C. After team convergence, I then incrementally in-

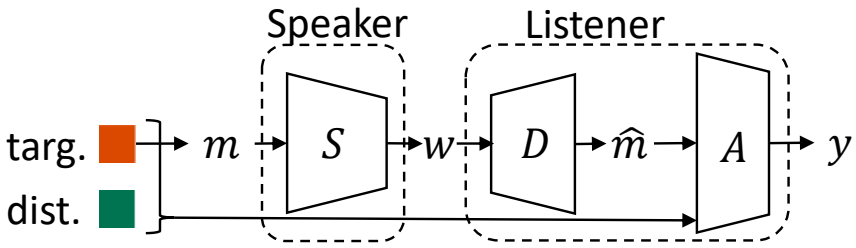


Figure 2-4: Reference game setup for the color experiment. Given a noisily-observed target color, the speaker communicated to a listener, which predicted, from a set of the target and a distractor color, the target. Utility was based on team accuracy.

creased λ_C by a fixed step size every training episode, which induced a spectrum of communication complexity levels.

2.5.2 Results

I found that controlling the complexity of communication enabled EC agents to learn human-like color naming systems, and that VQ-VIB agents learned semantically-meaningful communication spaces. Visualizations of the results, for VQ-VIB$_{\mathcal{N}}$, are included in Figure 2-5.

Figure 2-5 a shows how VQ-VIB$_{\mathcal{N}}$ EC messages were nearly optimally-efficient, just under the IB-optimal curve for informativeness vs. complexity. Furthermore, by varying λ_C during training, VQ-VIB$_{\mathcal{N}}$ communication spanned the range of complexities observed in human color naming systems recorded in the WCS dataset. Crucially, this shows that VQ-VIB agents can learn IB-optimal communication and can be controlled similarly to analytical IB methods.

Snapshots of VQ-VIB$_{\mathcal{N}}$ naming systems, at different complexity levels, are shown in Figure 2-5 c-f. At low complexity (c and d), agents used only 5 communication vectors, representing high-level color categories. At high complexity (e and f), agents used more communication vectors and partitioned the color space more finely. By changing the complexity of EC, VQ-VIB$_{\mathcal{N}}$ agents learned communication systems similar to different hu-

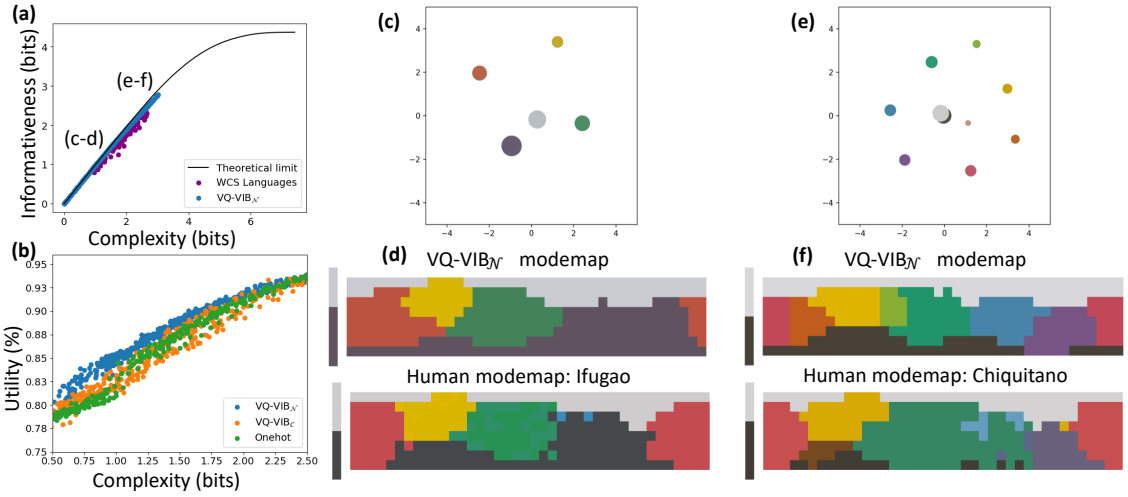


Figure 2-5: Color reference game results. Within the ITEC framework, I annealed λ_C to generate a spectrum of near-IB-optimal naming systems, closely matching human-like diversity (a). Within the range of human-like complexities, VQ-VIB$_\mathcal{N}$ tended to achieve greater utility, for the same complexity, as other methods (b). At low (c - d) and high (e - f) complexity levels, VQ-VIB$_\mathcal{N}$ used fewer or more distinct tokens for colors, and closely aligned with human languages at similar complexity levels. Notably, the tokens were embedded in a semantically-meaningful communication space, as visualized using 2D PCA (c, e).

man naming systems (e.g., Ifugao at low complexity and Chiquitano at high complexity). [3] Videos of communication evolution, for VQ-VIB$_\mathcal{N}$ and VQ-VIB$_\mathcal{C}$ are available online, showing similar behavior for both model types, and smooth interpolation of behavior across complexity levels.

Beyond matching human languages, VQ-VIB$_\mathcal{N}$ learned a meaningful communication space. 2-dimensional principle component analysis (PCA) of the communication space at different complexity levels (Figure 2-5 c and e) show how vectors representing similar colors were located in similar locations in the communication space. For example, the embedding for orange-like colors is close to the embedding for yellow colors, and far from the embedding for blue colors. This embedding space likely supported VQ-VIB$_\mathcal{N}$'s improved utility relative to other methods (Figure 2-5 b).

In addition to comparing different variational methods within the ITEC framework, I trained baseline non-variational onehot and VQ-VAE models without informativeness or

[3] For completeness, I repeated these experiments using a REINFORCE training mechanism in which gradients were not passed between the listener and speaker. Results were largely unchanged; details are included in Appendix 5.5.

	VQ-VIB$_{\mathcal{N}}$	Onehot	VQ-VAE	IB-optimal
Human-agent gNID	0.151 (0.00)	0.42 (0.16)	0.38 (0.15)	0.18 (0.10)
Efficiency loss	0.024 (0.00)	0.06 (0.01)	0.08 (0.01)	NA, 0 by definition

Table 2.1: Quantitative evaluation of artificial color communication systems, compared to human naming data and an IB-optimal bound. VQ-VIB$_{\mathcal{N}}$ models, trained in the ITEC framework, were closer to human languages (lower gNID - generalization of Normalized Information Distance) and closer to the IB bound (lower efficiency loss) than utility-only methods (onehot and VQ-VAE). Compared to the IB-optimal color naming system (from Zaslavsky et al. [2018]), VQ-VIB$_{\mathcal{N}}$ was more human-like, suggesting an important role of utility language development.

complexity losses. (Recall that VQ-VAE is a non-variational discrete representation learning mechanism similar to VQ-VIB methods but without a stochastic encoder or variational bounds on complexity.) Without variational methods and information-theoretic pressures, I was only able to induce variations among agents by explicitly encoding different codebook sizes. Figure 5-4 in Appendix 5.5 shows plots of resulting behavior.

Quantitative evaluation of these non-variational models showed that they were both less optimal (in the IB sense) and less "human-like" than my variational approaches. Table 2.1 includes results for comparing EC communication to the WCS languages (measured via the generalized Normalized Information Distance, or gNID) and deviation from the IB theoretical bound; lower values are better for both metrics. VQ-VIB$_{\mathcal{N}}$ was both more human-like and more efficient than utility-based Onehot or VQ-VAE agents. Furthermore VQ-VIB$_{\mathcal{N}}$ agents were more human-like (lower gNID) than standard IB methods which ignore utility measures [Zaslavsky et al., 2018]. For completeness, I include these metrics for VQ-VIB$_{\mathcal{C}}$ and a variational extension of traditional onehot communication in Table 5.1. These methods achieve similar results to VQ-VIB$_{\mathcal{N}}$ and confirm that variational methods, trained in the ITEC framework, are more efficient and human-like than traditional EC methods.

Overall, results from this domain indicate that agents, trained in the ITEC framework, learn to communicate according to similar IB tradeoffs found in human naming systems. Furthermore, VQ-VIB models in particular learn meaningful embedding spaces that represent semantic relationships.

2.6 Experiment 2: Open-Domain Communication

Beyond studying communication a color-naming domain, I considered how the ITEC framework influenced learned communication in a broader and richer domain. Specifically, I used the ManyNames dataset to study aspects of generalization and alignment with natural language embeddings [Silberer et al., 2020]. In training agents, I found that encouraging informativeness led to higher self-play rewards (evaluating teams of agents that have trained together), including in harder evaluation settings in which agents communicated about types of images never seen during training. At the same time, limiting the complexity of communication was important for optimal human-agent alignment.

2.6.1 Experiment Setup

I trained agents in a reference game using the ManyNames dataset [Silberer et al., 2020], which is particularly appropriate for studying alignment of EC and natural languages. It is composed of 25,000 images, each of which is annotated with roughly 36 English responses. (The varying number of annotations is an artifact of the filtering process applied during collection of the dataset; I refer to Silberer et al. [2020] for further details.) Unlike most labeled image datasets with a closed set of prescribed labels, therefore, ManyNames reflects open-domain communication and captures important aspects of the probabilistic nature of human naming [Gualdoni et al., 2022, Mädebach et al., 2022]. Examples of images in the dataset, with associated responses, are included in Figure 2-3 b.

The images in Figure 2-3 b reflect important characteristics of the dataset. First, there is a wide variety of the types of images: from outdoor scenes of wildlife to indoor scenes of furniture [Silberer et al., 2020]. Second, there is important variation in the naming data [Gualdoni et al., 2022, Mädebach et al., 2022]. For example, while most participants labeled the top image in Figure 2-3 b as a "bird," others used the label "duck." Both labels are correct but reflect different complexity levels. I hoped that, in training EC agents on this dataset and controlling the complexity of communication, I could induce human-like EC.

Beyond inducing similarities between EC and natural language, I was interested in how

well EC generalized to novel inputs. Thus, rather than training on the full ManyNames dataset, I constructed semantically-distinct training- and test-sets from the full dataset. Before training, I recorded the most common response for each image (named the `topname`, and shown in bold in Figure 2-3). There were 442 `topnames` in the full dataset; I selected a random 20% of those names for the training set. All images with the matching topname were selected for the training set, while the test set was generated by finding all images for which *no* response matched a training-set topname. For example, if "duck" were a training-set topname, the top image in Figure 2-3 b would be in neither the training set (because it's topname is "bird") nor the test set (because "duck" is in the responses). This train-test split procedure tended to produce semantically-distinct sets of similar sizes.

I trained agents in the reference game setup shown in Figure 2-6. As in the color reference game, a speaker agent observed a noisy version of a target input image drawn from the ManyNames dataset, and a listener agent had to identify the target among a set of C candidate images. During training, I set $C = 2$, but in some evaluation settings, I increased C to increase the difficulty of the task. Because of the high-dimensionality of the images, I used a pre-trained ResNet feature extractor to generate 512-dimensional representations for each images [He et al., 2016]. These 512-dimensional vectors were passed through a pre-trained Variational Autoencoder (VAE) to simulate perceptual noise. Thus, agents observed noisy version of the features extracted from each image. Lastly, I note that during training candidate images were selected, by design, to always have distinct `topname` labels; this introduced an important distinction between utility (which could be maximized via unique words for each possible `topname`) and informativeness (which would be maximized at a much higher complexity level, representing fine-grained details in the target image).

Agents were trained via the ITEC losses, setting $\lambda_U = 1$ and $\lambda_C = 0.01$, with different λ_I across trials to investigate the effect of informativeness pressures on communication. I found that these λ values tended to cover a range of interesting behaviors from at-chance accuracy (reflecting uninformative communication) to accuracy and complexity surpassing estimates based on English naming data. Additional studies with $\lambda_U = 1$ and $\lambda_C \in \{0.001, 0.0001\}$ led to similar results, confirming that in these experiments the

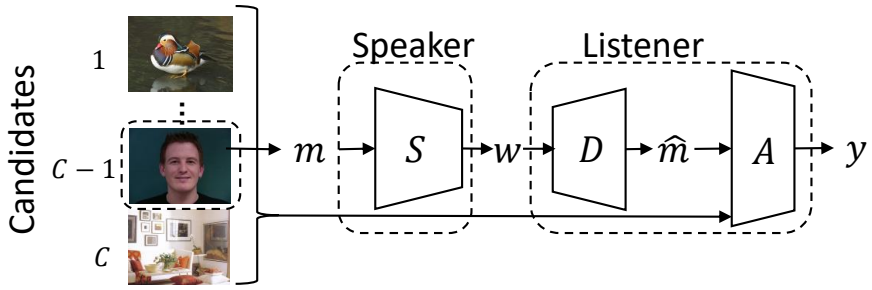


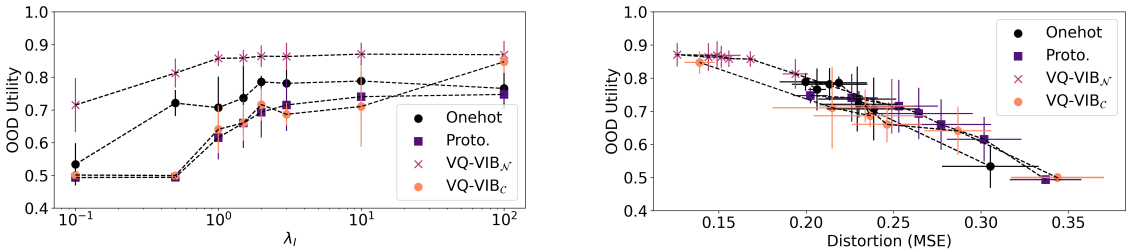
Figure 2-6: Reference game setup for ManyNames domain. The speaker observed a randomly-drawn candidate image and communicated to a listener, which decoded the speaker's meaning and predicted which candidate image was the target.

pressure for low complexity was quite small.

In evaluation, I studied both agent generalizability and aspects of EC-English alignment. To test generalizability, I measured team accuracy when evaluated on the held-out test set. Recall that these out-of-distribution (OOD) inputs were semantically distinct from the training images used by the team during training; by measuring the listener's accuracy in identifying the target image from a set of candidates, I captured one aspect of how generalizable the EC was to novel inputs.

Beyond generalization, I sought to measure alignment between pre-trained EC agents and GloVe embeddings [Pennington et al., 2014]. Notions of "alignment" typically correspond to representational similarity of some sort: two aligned representation spaces might encode the same inputs in similar locations, for example [Sucholutsky and Griffiths, 2023, Moschella et al., 2023]. In this work, I used two specific measures of alignment: functional alignment and relative representation alignment.

The first metric, functional alignment, captured how well EC and GloVe embedding spaces could be aligned to perform well on a reference game. Intuitively, this corresponded to ideas of translation between the two spaces. Using a pre-trained EC speaker, I fit a linear mapping from EC vectors for images to GloVe embeddings of an English label for the image (drawn from the responses associated with each image in the dataset). I then evaluated team accuracy on the ManyNames reference game using a simulated English speaker (again, drawing responses associated with each image) and the pre-trained EC listener, me-



(a) OOD Utility vs. λ_I

(b) OOD Utility vs. Informativeness

Figure 2-7: Team accuracy on OOD inputs as a function of λ_I (a) or informativeness (b). By increasing λ_I, I increased the communicative informativeness for all models, which in turn increased utility on OOD inputs.

diated by the linear mapping. High team accuracy indicated that a linear mapping could capture important similarities between EC and GloVe embedding spaces, at least for the purposes of the reference game task.

The second metric, relative representation alignment, reflected similarities in distances between encodings in different latent spaces [Moschella et al., 2023]. Concretely, O sampled 100 random images (and for each image, a sampled English response, as before) to generate EC and GloVe embeddings. I then measured the pairwise distance between each embedding in each space; this generated 10000 distances in each space (counting duplicates of distances from A to B and B to A, for example). Lastly, to generate a final value, O computed the Spearman correlation coefficient between these distances across the spaces. A large positive value would indicate that points that were close together in one space were close together in another space. Conversely, completely "unaligned" spaces should have correlation coefficients of 0.

2.6.2 Generalization Results

In generalization experiments, increased informativeness led to greater team accuracy on OOD inputs. Figure 2-7 shows team accuracy as a function of λ_I and informativeness.

Jointly, the plots indicate how informativeness pressure induced greater OOD utility. Figure 2-7 a shows that by increasing λ_I, I increased OOD scores. Given different inductive biases, some model architectures increased their OOD scores more quickly as a function

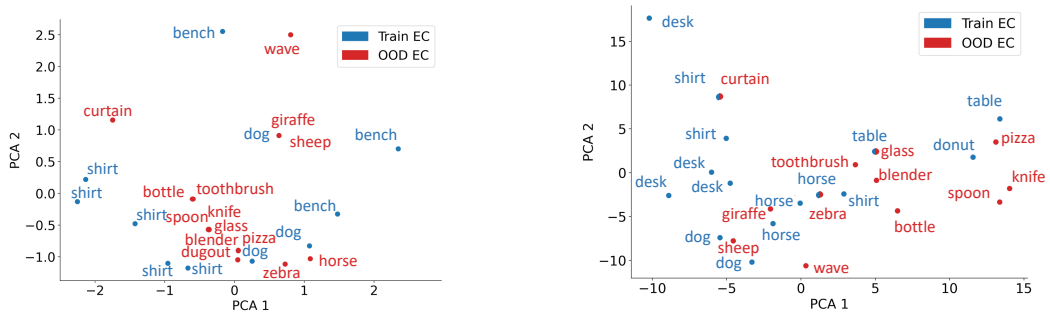
of λ_I (e.g., VQ-VIB$_\mathcal{N}$) and some architectures' utility values plateaued as λ_I grew larger.[4] However, in general, all architectures tended to achieve greater utility with greater λ_I. Recall that agents were evaluated on OOD inputs; increased scores from greater λ_I indicate an important generalization benefit of informativeness pressures.

Figure 2-7 b reveals an even closer relationship between informativeness and OOD utility. Each point represents the mean distortion (inversely related to informativeness) and utility for a particular model architecture and λ_I. By increasing λ_I, distortion tended to decrease (left along the x axis) and utility increased (up along the y axis). The close relationship between informativeness and utility across architectures also explains model performance differences: while all models tended to achieve similar utility for the same informativeness, some models were not able to learn highly-informative communication, which limited performance.

Lastly, visualization of VQ-VIB$_\mathcal{N}$ communication space in Figure 2-8 indicates how agents generalized well to OOD inputs, and why increasing informativeness increased utility. For each token in the EC agent, I recorded which images caused the speaker to generate that token, and labeled the token with the most common `topname` among that set of images. I then selected all tokens with greater than 1% likelihood of being emitted by the speaker and plotted them in blue using 2D PCA. This is similar to the modal coloring scheme used in prior visualizations. I then repeated this process, using the same PCA projection, for OOD inputs, which I plotted in red.

These visualizations reveal two important characteristics of VQ-VIB$_\mathcal{N}$ tokens. First, increasing the informativeness of communication increased the number of tokens used and the specificity of their meanings. For example, for $\lambda_I = 0.1$, many of the tokens were most associated with images of shirts – this indicates a relatively stochastic sampling process that emitted different tokens for the same input. At high informativeness, however, tokens for more specific types of images (e.g., "donut" or "horse") emerged that had not been distinctly encoded at low informativeness. Second, the tokens formed a semantically-meaningful space that extended to OOD inputs. For example, in Figure 2-8 b, the OOD

[4] In experiments, onehot agents never converged to greater than 50% utility for $\lambda_C = 0.01$, as was used for other models. I therefore set $\lambda_C = 0.0$ for onehot, which biased communication to have higher utility and informativeness, for the same λ_I and λ_U.



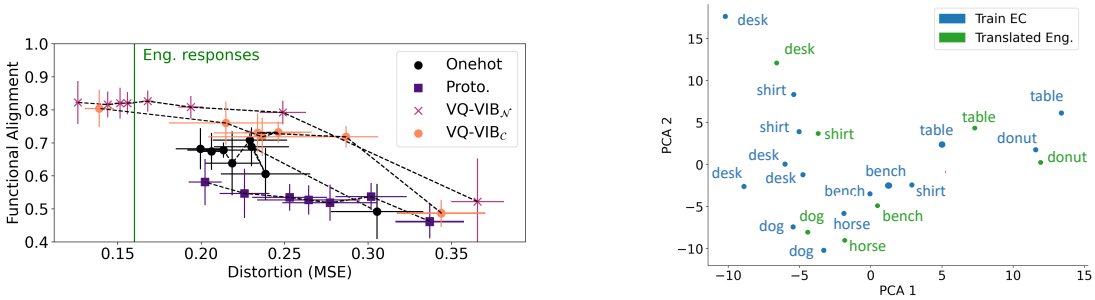
(a) VQ-VIB$_\mathcal{N}$, $\lambda_I = 0.1$ (b) VQ-VIB$_\mathcal{N}$, $\lambda_I = 100$

Figure 2-8: Visualization of VQ-VIB$_\mathcal{N}$ communication spaces at low (a) or high (b) informativeness. Each point represents the 2D PCA projection of the EC token for training inputs (blue) or OOD inputs (red). At low informativeness, there were few commonly-used tokens, and the meanings associated with each token was highly stochastic. At higher informativeness, distinct images were more often encoded differently, and the communication space was semantically-interpretable, which often generalized to OOD inputs.

input of an image of a sheep was located near training-set images of dogs and horses. Similarly, food-related images clustered near images of tables and donuts. This semantic embedding space, learned by both the speaker and the listener, supports generalization to novel inputs, similarly to how word embeddings improve natural language processing generalization to new words.

For the results plotted in Figure 2-7, I evaluated OOD generalization with 2 candidate images at test time. As motivated by Chaabouni et al. [2021b], who advocate for evaluation in more challenging settings with more candidate images, I repeated such evaluations for $C = 16$ and $C = 32$. Generalization results in such settings are included in Appendix 5.6. In general, I found that increasing C worsened team performance, as expected, and that VQ-VIB agents continued to outperform other architectures.

Lastly, I note that similar generalization trends hold, and are more obvious, for VQ-VIB agents with combinatorial codebooks. Recall that VQ-VIB$_\mathcal{N}$ and VQ-VIB$_\mathcal{C}$ support multiple discretizations that are concatenated together for a final communication vector. Increasing the number of concatenated vectors (n) decreased communication distortion, which improved OOD utility (see Appendix 5.6, Figure 5-5). For example, VQ-VIB$_\mathcal{N}$ for $n = 4$ achieved OOD accuracy for $C = 32$ of roughly 60%, more than three times better than onehot or prototype agents in similar evaluation. Thus, I combinatorial codebook



(a) Functional alignment vs. distortion

(b) 2D PCA of linearly-aligned embeddings

Figure 2-9: Functional alignment (a) and visualization of embeddings (b) for GloVe and EC models. Functional alignment improved as distortion decreased until reaching the estimated English informativeness, at which point performance plateaued.

results corroborate trends from the main paper and show how combining tokens allows agents to develop more complex communication.

2.6.3 Alignment Results

Using the same pre-trained agents from the prior experiments, I evaluated the functional alignment and relative representation alignment of EC agents with GloVe embeddings. In functional alignment experiments, I fit a linear transform from EC tokens to GloVe embeddings and then evaluated the team accuracy for a simulated English speaker and an EC listener, mediated by this linear "translator." In relative representation alignment experiments, I computed the Spearman correlation coefficient between EC embedding distances and GloVe embedding distances.

Functional alignment, for different models as a function of informativeness, is plotted in Figure 2-9 a. At high distortion (low informativeness), functional alignment improves as distortion decreases. This closely matches OOD trends. However, unlike in the generalization experiments, performance largely plateaued once EC informativeness decreased below English response informativeness. That is, there was no benefit to training more informative EC agents beyond a distortion value around 0.16. This indicates that performance was bottlenecked by the English speaker, and further informativeness simply added unnecessary complexity to the EC communication.

Visualization of translated GloVe embeddings in Figure 2-9 b show similarities between

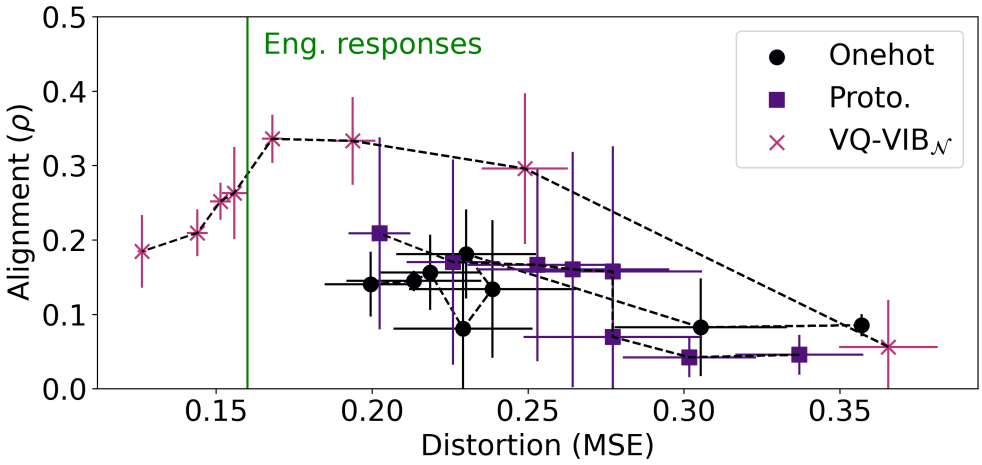


Figure 2-10: Relative representation alignment (ρ) between EC and GloVe embedding spaces. VQ-VIB$_\mathcal{N}$ models had greater alignment than other models, and alignment peaked when EC model informativeness matched English response informativeness.

the EC and GloVe embedding spaces. As before, each blue point represents an EC embedding for images from the training set. The green points show the embeddings generated by passing a GloVe embedding through the linear transformation. The semantic structure of the GloVe embeddings was largely preserved through this linear mapping, indicating substantial similarities between the EC embedding space and the semantically-meaningful GloVe space.

Lastly, I measured relative representation alignment as a function of distortion and plotted results in Figure 2-10. This alignment metric demonstrates the importance of tuning EC informativeness to the right level even more starkly than the functional alignment experiments. At high distortion, all relative representation alignment values (ρ) were roughly 0, indicating no consistent relationship between EC spaces and GloVe embeddings. As in the functional alignment experiments, as distortion decreases, ρ increases for VQ-VIB$_\mathcal{N}$, until reaching English response levels. At that point, ρ *decreases*, reflecting a worsening alignment between the spaces as informativeness increases further. (Similar trends hold for VQ-VIB$_\mathcal{C}$ and are plotted in Figure 5-8, omitted here for clarity.) Thus, VQ-VIB models achieve peak relative representation alignment by matching the informativeness of the simulated English speaker. This is a key result in the experiments, highlighting the importance of matching information-theoretic properties of EC and natural language for the greatest

alignment.

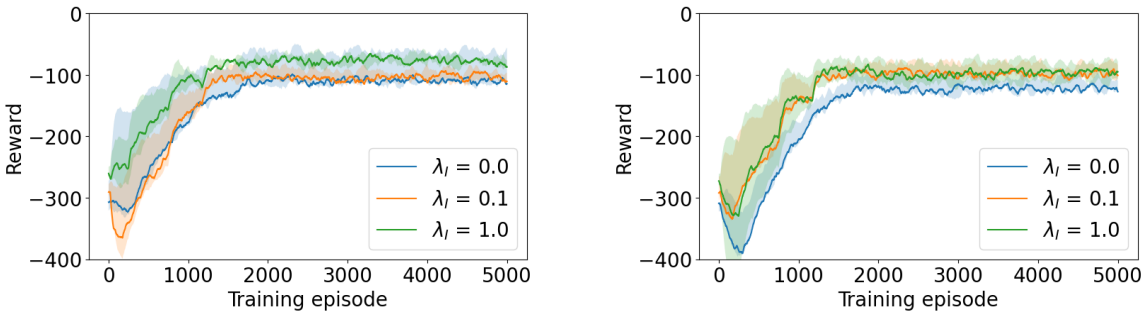
2.7 Experiment 3: Generalizing to 2D Navigation

In the first two experiments, I showed how agents could learn to communicate in reference games, and how terms in the ITEC framework regulated important aspects of communication; in my final experiment, I showed how the same framework can be applied to training agents in (simulated) grounded environments. As before, I found that penalizing complexity led to simpler systems. At the same time, encouraging informativeness improved the convergence of communication to more meaningful protocols, suggesting an important pressure for language emergence.

2.7.1 Experiment Setup

I developed a two-dimensional simulated world, depicted in Figure 2-3 c. In the world, a speaker agent observed a target, spawned at a location generated uniformly at random in the map, while a listener agent only observed its own location in the map (but not the target location). Both agents achieved reward equal to the negative Euclidean distance from the listener to the target, so utility was maximized if the listener moved straight to the target. Given that the listener could not observe the target location, the speaker and listener had to jointly learn to use the environment's communication channel, in which the speaker could broadcast communication at the first timestep in the environment, which the listener could observe. Thus, the optimal policy to maximize reward would consist of the speaker communicating about the location of the target. Agents were trained using a standard policy-gradient method (Multi-Agent Deep Deterministic Policy Gradient [MADDPG, Lowe et al. [2017]]), as well as the informativeness and complexity terms introduced in the ITEC framework.

I conducted two types of experiments in this domain by varying λ_I or λ_C. In the first experiment, I trained new teams of agents from scratch with $\lambda_U = 1.0$ and $\lambda_C = 0.01$ while varying λ_I across trials. (I found largely similar results for other small values of $\lambda_C \in \{0.001, 0.0001\}$.) This exposed how different informativeness pressures led to different



(a) VQ-VIB$_{\mathcal{N}}$ (b) VQ-VIB$_{\mathcal{C}}$

Figure 2-11: Training curves for VQ-VIB$_{\mathcal{N}}$ and VQ-VIB$_{\mathcal{C}}$. During training, mean reward increased for all methods, but increasing λ_I led to faster convergence and higher mean rewards.

training speeds. In the second experiment, I fixed $\lambda_U = 1.0$ and $\lambda_I = 1.0$ while slowly increasing λ_C within a trial. This revealed how decreasing complexity led to different communication strategies.

2.7.2 Results

I observed three main trends in the 2D Navigation experiments: 1) increasing λ_I led to faster convergence to higher reward for all agent architectures, 2) increasing λ_C led to less complex communication and coarser discretizations of the 2D space, and 3) VQ-VIB methods outperformed other agent architectures in achieving greater utility, for the same complexity, as other agents, likely due to the semantically-meaningful communication space that VQ-VIB agents learned.

As shown in Figure 2-11, greater λ_I led to faster convergence of team performance to higher rewards. Each curve in Figure 2-11 represents the team utility (calculated as the average distance from the listener to the goal) over the course of training VQ-VIB$_{\mathcal{N}}$ and VQ-VIB$_{C}$ teams with different λ_I values, averaged over 5 random trials. As shown for these VQ-VIB architectures, and for baselines architectures included in Appendix 5.4, which showed similar trends, increasing λ_I led to faster convergence to higher mean rewards. This demonstrates an important benefit of informativeness pressures in the emergence of communication among intelligent systems.

In the next experiment, I trained teams with $\lambda_U = 1.0, \lambda_I = 1.0$ and slowly increased

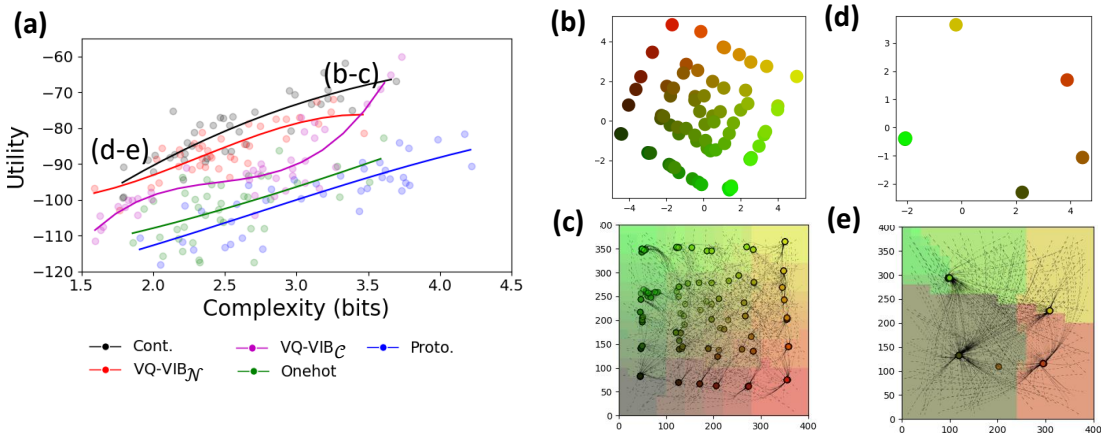


Figure 2-12: Varying complexity solutions in the 2D world environments. (a) Utility vs. complexity for different architectures reflect tradeoffs between referring to specific target locations (high utility) and using less complex communication. For a given complexity level, continuous communication represents an optimal upper bound for utility. Among discrete methods, VQ-VIB methods outperformed other architectures. Visualizations of VQ-VIB$_\mathcal{C}$ agent traces and end locations at high (c) and low (e) complexity show how agents discretized the space into increasingly large regions. Visualization of the communication for these agents (b and d) show that communication vectors were embedded in a meaningful space: nearby tokens (in communication space) referred to nearby targets (in physical space). A video of communication evolution for VQ-VIB$_\mathcal{N}$ shows smooth evolution between high and low complexity solutions.

λ_C over the course of training. This smoothly decreased the complexity of communication for all agents. Results from that experiment are shown in Figure 2-12.

Comparing across architectures, I found that both VQ-VIB models achieved greater utility, for the same complexity, than other methods. Figure 2-12 a plots team utility for different architectures and random trials, recorded while annealing λ_C. (The plotted complexity metric is calculated via the variational bound used during training, providing an upper bound on true complexity.) Increasing λ_C led to less-complex communication (moving to the left along the x axis) and lower utility (moving down along the y axis). All agent architectures demonstrated the expected decrease in utility as complexity decreased, but some architectures (VQ-VIB) achieved greater utility than others for the same complexity value. The black curve for continuous communication represents a theoretical upper bound on utility, which the VQ-VIB agents nearly match despite being a discrete communication

method.

Visualizations of the VQ-VIB$_\mathcal{C}$ communication space, and discretizations of the 2D space, suggests that VQ-VIB agents outperformed other discrete methods due to their semantically-meaningful communication space (similar to in the reference games in prior experiments). Figure 2-12 (c and e) depict how VQ-VIB$_\mathcal{C}$ agents discretized the 2D space at high and low communicative complexity values, respectively. Each background color represents the set of locations referred to by the same communication vector, and each point represents the mean location of the listener agent, based on that communication vector (with actual paths traced in black). Thus, Figure 2-12 shows that high-complexity communication led to fine-grained discretization of the continuous space, while low-complexity communication led to much cruder discretizations. This aligns with intuitions of humans modulating the complexity of spatial navigation from highly complex (e.g., GPS coordinates when navigating to a precise location) to crude (e.g., "North," "South," "East," or "West" for high-level directions).

Visualizations of the VQ-VIB$_\mathcal{C}$ communication space, included in Figure 2-12 (b and d), suggest that VQ-VIB agents achieved greater utility than other methods by communicating via a discrete set of symbols embedded in a semantically-meaningful space. Figures 2-12 (b and d) show the learned communication vectors at high and low complexity. (Agents were trained to communicate via 2D tokens to support direct visualization of the communication vectors.) Each token is visualized as a point in the communication space, colored by the mean location it referred. At both high and low complexity levels, the communication tokens clearly reflect a semantically-meaningful space: nearby tokens (in communication space) referred to nearby targets (in the simulated world). These results closely parallel the findings from the color and ManyNames reference games, wherein VQ-VIB agents learned semantically-meaningful embeddings, thus demonstrating consistent trends across domains.

2.8 Discussion

Across experiments, two trends emerged: 1) controlling complexity and informativeness, within the ITEC framework, enabled faster learning, greater generalization, and more human-like communication than standard EC methods, and 2) VQ-VIB methods were particularly adept at learning semantically-meaningful complexity-limited communication.

In all experiments, for all neural network architectures, I found that controls over informativeness and complexity regulated important aspects of communication. Increasing pressure for informativeness (by increasing λ_I) tended to induce faster convergence and greater communicative generalization, as evidenced in the 2D world and ManyNames results. Simultaneously, limiting the complexity of communication (by increasing λ_C) was important for aspects of human-like communication, as shown via comparisons to color-naming systems and natural language word embeddings. These findings suggest that informativeness and complexity, thought to regulate aspects of human naming systems, can play important roles in EC settings.

Beyond establishing general characteristics of terms in the ITEC framework, I specifically found that VQ-VIB methods outperformed other neural agent architectures, likely due to a semantically-meaningful communication embedding space. In all experiments, VQ-VIB tended to achieve greater utility, for the same complexity, as other discrete EC methods. At the same time, all methods exhibited similar informativeness-complexity tradeoffs. This is unsurprising: the IB informativeness-complexity tradeoff makes no assumptions about how information is encoded. However for high task performance, using a task-appropriate representation space can be important. Unlike onehot agents, for which all communication vectors were equidistant and orthogonal, VQ-VIB agents embedded discrete communication in a continuous space and, through visualization in all experimental domains, I found that this space encoded important semantic properties of the inputs.

These findings raise important questions for future work, including how communicative pressures are instantiated in the real world. First, while utility may be reasonably modeled via the success of actions in the world, it is unclear *a priori* how human speakers and listeners could estimate the informativeness of communication without direct access to others'

mental states. New research inspired by Theory of Mind (ToM) capabilities could shed important light on such questions. At the same time, while I trained emergent communication agents with a single utility function, one might also consider a multi-task framework in which agents must accomplish many rewards. The relationship between multi-task utility and informativeness appears particularly interesting.

2.9 Conclusion

In this work, I combined cognitively-motivated terms with utility-based rewards to produce information-theoretic emergent communication. While human languages are thought to be guided by pressures on informativeness and complexity, prior art in emergent communication often adopts a task-specific training mechanism. Using my framework, I found that explicit informativeness and complexity losses yielded important human-like benefits to emergent communication. Increased pressure for informativeness, for example, led to greater generalization of EC agents, while decreasing complexity was necessary for better alignment with natural language representations.

Within the context of my information theoretic emergent communication framework, I found that a novel neural network method, the Vector Quantized Variational Information Bottleneck, outperformed other discrete communication mechanisms. Unlike prior methods, it supports complexity-limited discrete representations embedded in a continuous space. Across experiments, I found that this embedding space reflected important semantic properties, which in turn supported high team performance and better human-agent alignment.

Overall, I believe that emergent communication is a powerful *in silico* testbed for measuring the effect of different pressures on language evolution. Therefore, I hope future work extends my studies by considering the role of other cognitively-motivated pressures, beyond complexity and informativeness, on emergent communication.

Chapter 3

Analyzing Pre-trained Models: Causal Probing for Syntax

In the prior chapter, I considered how to train more human-like emergent communication systems. That work represents a "bottom-up" approach towards building cognitively-plausible AI systems. In contrast to such work, a different, and recently successful, approach to AI development has been to train Large Language Models (LLMs) on vast quantities of data. Subtle tuning of cognitively-motivated pressures plays less of a role in training such massive models – instead, they are often trained on simple tasks such as next-word prediction. What do such LLMs learn about language?

Answering the above question in general is challenging, but one can consider more narrow questions about how LLMs represent specific aspects of language. In particular, humans naturally learn to use latent concepts such as syntactic structure, plurality, or gender to understand and produce language. Do LLMs learn to use similar latent concepts?

I propose a method for causal analysis of latent concepts in model representations and conduct experiments showing that some models use representations of syntax in making predictions. First, inspired by prior work, I train a small model, dubbed a probe, that predicts aspects of a sentence's syntactic structure from an LLM's embedding. Second, I use a gradient descent mechanism to create counterfactual embeddings that change probe decisions. Third, I record model predictions using the counterfactual embeddings. A simplified schematic of this approach is depicted in Figure 3-1. Subject to some variation across

model and probe types, I find important evidence that some models use representations of syntax in making their predictions.[1]

3.1 Introduction

Large neural models like BERT and GPT-3 have established a new state of the art in a variety of challenging linguistic tasks [Devlin et al., 2019, Brown et al., 2020]. These connectionist models, trained on large corpora in a largely unsupervised manner, learn to map words into numerical representations, or embeddings, that support language-reasoning tasks. Fine-tuning these models on tasks like extractive question answering specializes these generic models into performant, task-specific models [Wolf et al., 2019].

In conjunction with the rise of these powerful neural models, researchers have investigated what the models have learned. Probes, tools built to reveal properties of a trained model, are a favored approach [Hall Maudslay et al., 2020b, Conneau et al., 2018]. For example, Hewitt and Manning [2019] have uncovered compelling evidence that several models encode syntactic information in their embeddings. That is, by passing embeddings through a trained probe, one may recover information about a sentence's syntax.

Although these results are impressive, they fall short of clearly demonstrating what linguistic information the language models actually *use*. Syntactic information is present in sentences; that embeddings also encode syntax does not imply that a model uses syntactic knowledge.

In order to truly query a model's understanding, one must use causal analysis. Recently, several authors have done so by generating counterfactual data to test models [Kaushik et al., 2020, Goyal et al., 2019, Elazar et al., 2020]. They either create new input data or ablate parts of embeddings and study how model outputs change. I extend this prior art via a new technique for generating counterfactual embeddings by using structural probes to manipulate embeddings according to syntactic principles, as depicted in Figure 3-1. Because I conduct experiments with syntactically ambiguous inputs, I am able to measure how models respond to different valid parses of the same sentence instead of, for exam-

[1] Much of the work in this chapter is described in Tucker et al. [2021b, 2022a].

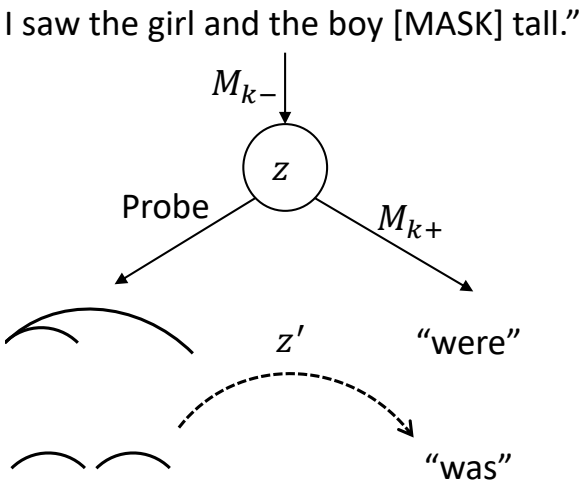


Figure 3-1: A language model, M, generates latent representations at layer k. These representations are used to output model predictions (e.g., "were") and probe predictions (e.g., a dependency structure). I use probes to generate counterfactual representations, z', based on syntactic manipulations, to reveal aspects of the model's causal structure.

ple, removing all syntactic information. Thus, my technique uncovers not only what parts of its embeddings a model uses to represent syntax, but also how those parts influence downstream behavior.

In this chapter, I make three contributions. First, I develop a gradient-based algorithm to generate counterfactual embeddings, informed by trained probes. Second, I show how standard probes likely use only a subset of the syntactic information in model embeddings, indicating a disconnect between the information probes and models use; I introduce a novel type of "dropout" probe to address this limitation. Third, in experiments using my technique and dropout probes, I find that some BERT-based models, trained on word-masking tasks, appears to leverage features of syntax in making predictions.

3.2 Related Work

3.2.1 Neural Language Model Probes

Transformer-based models like GPT-3 and BERT have recently advanced the state of the art in numerous language-related problems [Brown et al., 2020, Devlin et al., 2019, Wolf

et al., 2019]. These large models appear to learn meaningful representations of words and sentences, enabling high performance when fine-tuned for a specific task.

In conjunction with these models, probes have been developed to uncover what principles models have learned. Such probes have been used in a wide variety of contexts, from image structure to syntax and semantics in language models ([Alain and Bengio, 2018, Conneau et al., 2018, Hewitt and Manning, 2019, Reif et al., 2019], among others). In standard probing literature, a neural network probe, p, is trained to map from embeddings, Z, to a predicted property, S: $S = p(Z)$. For example, Hewitt and Manning [2019] define two types of probes that map from Z to representations of a sentence's syntax. Their "depth" probe predicts words' depths in a parse tree; "distance" probes predict the distance between pairs of words in a parse tree. In my work, I assume S refers to syntactic information, but probing techniques are general. Given a corpus comprising (Z, S) pairs, probes are trained using supervised learning to minimize some supervised loss.

Recent work considers how to properly parametrize probe models. On the one hand, if probes are too expressive, they may reveal their own learning instead of a model's [Liu et al., 2019, Hewitt and Liang, 2019a]. Out of caution of over-interpreting trained probe results, Li et al. [2022] avoiding probes altogether and instead seek to find prompts (i.e., additional tokens concatenated to standard inputs) to reveal model reasoning. On the other hand, Pimentel et al. [2020] argue from an information-theoretical perspective that more expressive probes are always preferable. Regardless of exact parametrization, some recent works seek to contextualize probe performance by reporting aspects of probe uncertainty [Wang et al., 2023] or comparing probe performance for different test suites [Hewitt and Liang, 2019b].

My work differs from much prior art in probe design by leveraging causal analysis, which uses counterfactual data to test probes and models. This provides direct evidence of whether a model uses the same features as a probe, allowing me to experiment beyond linear probes (and indeed, I found that more complex probes offered an advantage in some cases).

3.2.2 Causal Analysis of Language Models

Motivated by the limitations of traditional, correlative probes, researchers have recently turned to causal analysis to better understand language models. Goyal et al. [2019] and Kaushik et al. [2020] generate counterfactual inputs to language models, while Vig et al. [2020] study individual neurons and attention heads to uncover gender biases in pre-trained networks.

My work is most closely related to that of Elazar et al. [2020], who, as in this work, used probes to generate counterfactual embeddings within a network. Their amnesiac counterfactuals are generated by suppressing features in embeddings that a linear probe uses. In contrast, I use a continuous, gradient-based approach to generate counterfactuals, yielding insight into how features are used, as opposed to if they are used at all.

3.3 Technical Approach

Here, I propose a causal framework for understanding how representations of syntax mediate model predictions (Section 3.3.1), design a method for generating counterfactual representations within this framework (Section 3.3.2), and show how care must be taken in creating counterfactual representations when information is redundantly encoded (Section 3.3.3).

3.3.1 Causal Problem Formulation

One may characterize a transformer-based language model, M, trained on a specific task, as a function mapping from an input string, x, to an output y: $M(x) = y$. In order to reveal embeddings for analysis by probes, one may decompose M into two functions: M_{k-} and M_{k+}. M_{k-} represents the first k layers of the model; M_{k+} represents the layers of M after layer k; M is the composition of these functions: $M = M_{k+} \circ M_{k-}$. I label the embeddings output by M_{k-} as z. This decomposition of models to reveal internal embeddings mirrors the formulation for layer-specific probes [Hewitt and Manning, 2019]. As noted earlier, a probe may be defined as a function p that maps from an embedding, Z, to a property

S about the input, x: $p(M_{k-}(x)) = S$. (For the remainder of this chapter, I focus on syntactic probes, but my reasoning may be extended to other linguistic properties.) The relationships between model inputs, probe inputs, and model and probe outputs are depicted in the structural causal diagram in Figure 3-2.

Figure 3-2 further reveals important decompositions of the information in model embeddings, Z. In particular, in this chapter, I consider whether latent representations of syntactic structure mediate model predictions. Therefore, Z is decomposed into four, non-overlapping parts to isolate the parts of Z that encode different information about S (syntactic structure) and Y (model predictions). Specifically, Z comprises 1) Z_S, the part of Z that only mediates predictions of S, 2) $Z_{S \bigwedge Y}$, the part of Z that mediates predictions of S and predictions of Y, 3) Z_Y, the part of Z that only mediates predictions of Y, and 4) $Z_{\bar{S} \bigwedge \bar{Y}}$ that does not mediate predictions of S or Y. Each of these four components support intuitive interpretations: Z_S encodes syntactic information that does not affect word predictions, $Z_{S \bigwedge Y}$ encodes syntactic information that is important for predicting words, Z_Y encodes semantic information independent of syntax, and $Z_{\bar{S} \bigwedge \bar{Y}}$ encodes irrelevant information for word or syntactic predictions (e.g., capitalization of words). In standard probing approaches, training a neural network probe that predicts S accurately is insufficient evidence for determining whether syntactic information does mediate model predictions (in $Z_{S \bigwedge Y}$) or does not (only in Z_S).

In this chapter, I seek to uncover whether representations of syntax mediates model predictions.[2] Borrowing notation from Figure 3-2, I consider the link between $Z_{S \bigwedge Y}$ and Y to show if there is any part of Z that mediates both probe predictions and model predictions. In Pearl's *do*-calculus notation, I ask if $P(Y|do(Z_{S \bigwedge Y})) = P(Y|do(Z_{S' \bigwedge Y}))$: do predictions of Y change for different syntactic structures S and S' [Pearl and Mackenzie, 2018]?

[2]More specifically, I ask whether representations of syntax mediate model predictions in linguistically consistent ways. Merely mediating predictions but in an unexpected or a-grammatical way is less relevant to questions of human-like language processing.

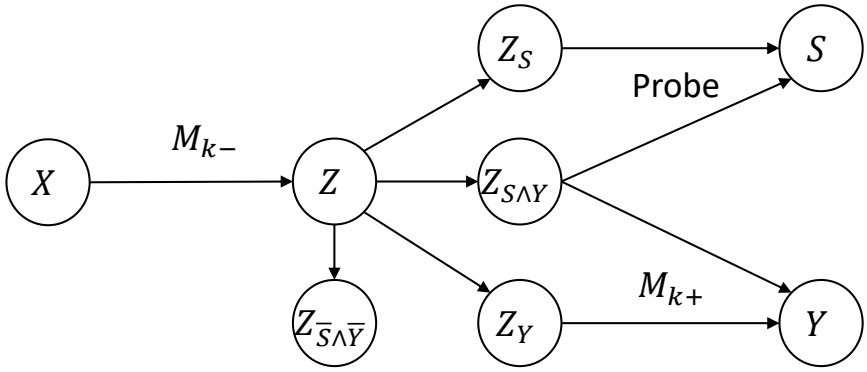


Figure 3-2: The structural casual diagram describing probed models. A language model, M, generates latent representations Z at layer k. Given a probed property, S, and a model prediction, Y, Z may be decomposed into the part of Z that only mediates predictions of S (Z_S), the part of Z that mediates predictions of both S and Y ($Z_{S \bigwedge Y}$), the part of Z that only mediates predictions of Y (Z_Y), and 4) the part of Z that mediates neither predictions S nor Y ($Z_{\bar{S} \bigwedge \bar{Y}}$). (Figure partially inspired by Veitch et al. [2021].)

3.3.2 Probe-Based Counterfactuals

To study such a causal question, I generate counterfactual embeddings, Z', that modify probe outputs, starting from standard embeddings Z. I borrow the term "counterfactual" from causal literature because Z' represents what Z would have been if Z_S and $Z_{S \bigwedge Y}$ had been different [Pearl and Mackenzie, 2018]. I am particularly interested in finding Z' that changed both probe and model outputs; if Z' only changes probe outputs, that could indicate that the probe was acting as an independent parser instead of reflecting model reasoning (more formally, that Z_S exists, but $Z_{S \bigwedge Y}$ does not) [Hall Maudslay et al., 2020b].

In typical counterfactual literature, one might seek to find minimally-different Z' from Z that change probe predictions. More formally, given a sentence embedding, Z, a probe, p, a probe loss function, L, a desired syntactic structure, S, and a "threshold loss", T, one can find the closest Z' in latent space according to the following optimization problem:

$$Z' = \underset{Z'}{\operatorname{argmin}} ||Z - Z'||_2^2 \quad \text{s.t.} \quad L(p(Z'), S) \leq T \tag{3.1}$$

That is, the counterfactual embedding Z' is the closest (via Euclidean distance) embedding to Z such that the loss between the probe's prediction of Z' and the desired probe

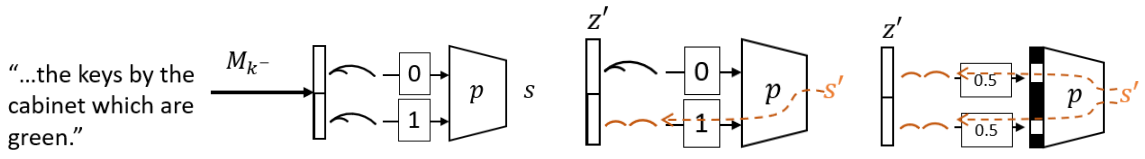
prediction is less than or equal to T. Intuitively, this corresponds to the minimal change to Z to make the probe predict the desired property value.

While well-formulated, the optimization in Equation 3.1 is challenging to solve in practice. In particular, given non-linear probes, small changes in Z' may induce large changes in $p(Z')$, so direction optimization of Equation 3.1 would require exhaustive exploration of the (continuous) embedding space.

Instead, I developed an approximate but tractable gradient-based method to generate Z'. Assuming a differentiable probe (e.g., a neural network) and loss function, one can compute the gradient of the loss with respect to the embeddings: $\nabla_{Z'} L(p(Z'), S)$. Standard gradient descent methods may then be employed to decrease the probe loss (barring failures from converging to local minima). Given Z and p, I constructed a counterfactual embedding, Z', by initializing Z' as Z and updating Z' via gradient descent of the loss. Updating Z' may be terminated based on various stopping criteria (e.g., local optimality, loss below a threshold T, etc.), yielding the final counterfactual Z'.[3] In experiments, I studied how Z's changed model outputs when passed through M_{k+}.

Although my technique bears some resemblance to gradient-based adversarial attacks [Szegedy et al., 2014], it may more broadly be thought of as guided search in a latent space. Adversarial images are often characterized by changes that are imperceptible to humans but change model behaviors to be incorrect. In contrast, I seek to find embeddings that change both probe and language model outputs. Furthermore, by design, I use syntactically ambiguous sentences in experiments and generate counterfactuals according to valid parses. Thus, unlike adversarial attacks on images that seek to switch model classification to an incorrect class, I merely guide embeddings among a set of valid interpretations. Lastly, even uncovering instances of embeddings that change a probe's outputs but not a model's is important as it indicates a misalignment of probe and model reasoning.

[3]Note: this gradient-based optimization decreases the probe loss from Equation 3.1 but does not consider that equation's measure of distance between Z and Z'. Future work may wish to use gradient methods to optimize a different objective that combines both counterfactual distance and probe predictions via, e.g., Lagrange multipliers to enforce distance constraints.



(a) (b) (c)

Figure 3-3: If a model encodes the dependency structure of a sentence twice its embedding, a probe, p, may learn to ignore one copy of the information (indicated by learned weight 0) and only use the other (via learned weight 1) to predict s (a). In such cases, the gradient of s with respect to the embedding (dashed orange) only flows from one of the copies, so only that copy will be updated in counterfactual embeddings (b). However, by introducing a dropout layer that masks random inputs to the probe, dropout probes learn to use all informative parts of embeddings, which distributes the gradient across the whole embedding (c).

3.3.3 Addressing Limitations from Redundancy: Dropout Probes

The previous section introduces a method for generating counterfactual embeddings to test whether models use representations of syntax in making predictions, but such an approach could yield false negatives for particular model and probe architectures. In particular, in this section, I show how if models redundantly encode syntactic information in embeddings, probes and models could use different representations of the same information, which in turn could lead to uninformative causal analysis results. I propose a new probe architecture, dubbed "dropout probes," that addresses this limitation by encouraging probes to use all sources of information in embeddings.

Challenges from Redundancy

I show by example how standard causal probing methods may fail to reveal causal uses of syntactic information in language models. Here, I use a simplified example; in experiments I demonstrate that trained models exhibit similar phenomena.

I assume access to a trained model, M, and probe, p, using the same notation depicted in Figure 3-2. That is, the model generates an embedding, Z, from an input, X, and Z is used for probe predictions, S, and model predictions, Y. For the purposes of this example, I state that M uses syntactic information and specifically that Z is informative of the syntactic structure of X.

Let us assume that the dependency structure of X may be represented within a vector, Z_D, and that M_{k-} produces embeddings, Z, which are two identical copies of Z_D. Using pythonic notation, $Z = [Z_D] + [Z_D]$. Thus, Z contains syntactic information and, when I state that M "uses" syntactic information, I formally mean that $\nabla_{Z_D} M_{k+}(z) \neq 0$.

Building upon this example, let us label the two copies of Z_D as Z_{D^1} and Z_{D^2}, although the two vectors remain identical. If one trains a probe to predict syntactic forms from Z, it may arbitrarily learn to use any aspects of Z that are informative of its prediction, S. Let us say that the probe learns to use only Z_{D^2}, again defined as $\nabla_{Z_{D^2}} p(Z) \neq 0$. At the same time, the rest of the model, M_{k+} may only use Z_{D^1}: the copy that the probe does not use.

This example, while simplified, demonstrates a potential scenario in which causal probing techniques could return false negative results. Specifically, if one generates counterfactual embeddings, Z', by changing Z according to the activations that change the probe's outputs, only Z_{D^2} will change. Because M_{k+} uses only Z_{D^1} for predictions, the model's output will not change. This example is depicted in Figure 3-3.

Figure 3-2 also reveals, from a theoretical perspective, how redundant embeddings may lead to false negative causal results. Note that the probe makes predictions of S based upon Z_S and $Z_{S \bigwedge Y}$. If there is identical syntactic information in Z_S and $Z_{S \bigwedge Y}$, the probe could learn to use either copy in making predictions. If the probe does not use the information in $Z_{S \bigwedge Y}$ (i.e., $\nabla_{Z_{S \bigwedge Y}} p(Z) = 0$), then gradient-based counterfactuals generated via the probe will necessarily not change model predictions. Ultimately, without considering the redundancy in a model's internal representation, standard methods could fail to uncover the fact that M uses representations of syntax causally.

Dropout Probes

In this section, I propose a neural probe architecture to address the limitations of standard probe design by encouraging probes to use all syntactic information present in Z. The desired behavior is depicted in Figure 3-3 c: if the probe uses all activations that are informative of syntax, that will necessarily be a superset of the activations that the model uses for downstream processing (if the model uses syntax). Therefore, when generating counterfactual embeddings using such probes, every activation encoding syntactic information

would be updated, which in turn would change the model's output.

My approach was inspired by an idea of creating a mixture of probes, each trained to use a different masked subset of activations in Z. The full set of such probes would have to learn to use all activations in Z that are informative of S. One may approximate creating such a set by introducing a dropout layer as the first layer to a single probe. At training time, the dropout layer masks a random subset of the input; the mask itself changes with every training batch. I dub such probes "dropout probes."

3.4 Experiments

Here, I report the results from three types of experiments. First, I find evidence of redundantly-encoded syntactic information in model embeddings by calculating the mutual information between various activations in trained networks. This motivates using dropout probes. Second, I used dropout probes to investigate the role of representations of syntax in mediating model predictions. For some models, I *did* find evidence supporting the causal role of syntax, particularly when using dropout probes. Lastly, given my finding that models use syntax causally, I demonstrated how one could "inject" syntactic information into models to improve performance in syntactically-challenging tasks.

Experiments were conducted on four models, all based on huggingface's `bert-base-uncased` [Wolf et al., 2019]. The Mask model was the original model, trained on a masked language modeling task and next-sentence prediction [Devlin et al., 2019]. The QA model was fine-tuned on the Stanford Question Answering Dataset 2.0 [Rajpurkar et al., 2016].[4] Lastly, I trained two models, dubbed NLI and NLI-HANS, that were finetuned on the Multi-Genre Natural Language Inference dataset or that dataset augmented with the Heuristic Analysis for NLI Systems (HANS) dataset, respectively [Williams et al., 2018, McCoy et al., 2019]. I conducted analysis across multiple pre-trained models to account for variation in how different training objectives and data may influence each model's causal structure. Indeed, prior literature indicates that finetuning models can cause them to be less aligned

[4]The QA model was downloaded from huggingface model repository under "twmkn9/bert-base-uncased-squad2"

	$I(Z_1, D)$	$I(Z_2, D)$	$I(Z, D)$
Mask	2.2	2.6	2.7
QA	2.7	2.8	2.8
NLI	2.3	2.7	2.8

Table 3.1: The mean in nats of $I(Z, D)$ is less than $I(Z_1, D) + I(Z_2, D)$, indicating that information about D is redundantly encoded in embeddings. Standard deviation under 0.2 for all values over 5 trials.

with human representations of language [Gauthier and Levy, 2019].

3.4.1 Measuring Redundancy in Embeddings

First, I found that language models redundantly encoded syntactic information in their embeddings, which motivated using dropout probes.

I used a technique from prior art, Mutual Information Neural Estimator (MINE), which is a neural-network based approach for estimating the mutual information between two random variables [Belghazi et al., 2018]. It does so by computing a lower bound of mutual information and training a neural network to maximize that value. This provides a conservative but tight estimate of mutual information. I refer readers to Appendix 6.1 for further details of my implementation.

In using MINE, I defined four random variables of interest. The first, D, was the depth of each word in a sentence's parse tree; in other words, the labels used to train depth probes in prior literature [Hewitt and Manning, 2019]. The second random variable, Z, was the 768-dimensional embeddings generated by a language model for each token in an input sentence. Lastly, the third and fourth random variables (Z_1 and Z_2) corresponded to the first and second halves of Z for each token. That is, these variables comprised the starting and ending 384 units for each token's embedding. By measuring the mutual information between different pairs of these variables, one may formalize my redundancy hypothesis into the following test: $I(Z, D) < I(Z_1, D) + I(Z_2, D)$. Intuitively, if the test holds, there is shared syntactic information between Z_1 and Z_2.

I trained a MINE neural network on the first 5,000 examples from the Penn TreeBank to estimate mutual information between random variables [Marcus et al., 1993]. Embeddings

were taken from the fourth layer of the MASK, QA, and NLI models, although they may be generated elsewhere. Results are presented in Table 3.1. For all models, $I(D, Z) < I(D, Z_1) + I(D, Z_2)$; i.e., one gains little to no information for predicting D from the full Z instead of from just Z_1 or just Z_2. For example, for the Masked Language Model, there were 2.7 nats of mutual information between the full embedding and D ($I(Z, D) = 2.7$), and 2.6 of those 2.7 nats were present just in Z_2; thus, Z_1 only contained 0.1 additional nats of information about D. This is evidence of highly redundant syntactic information in Z.

In these experiments using MINE, I demonstrated how Z_1 and Z_2 could be defined as the subsets of redundant activations depicted in Figure 3-3. One could define other Z_1 and Z_2 to better characterize redundancy by choosing other subsets of Z; here, I merely claim that at least some redundancy is present in the model embeddings. Indeed, in independent work, published subsequent to these experiments, researchers have found evidence of redundant encodings in other large language [Nanda et al., 2023] and vision [Doimo et al., 2022] models.

3.4.2 Causal Probing for Syntax

The prior section established that language models encode syntactic information redundantly; here, I show that my gradient-based counterfactual update method, in conjunction with dropout probes, reveals that some models use representations of syntax in making predictions.

I trained both distance- and depth-based probes, the two types of syntactic probes proposed by Hewitt and Manning [2019]. I trained a new probe for each layer of each model, conducting 5 trials with random seeds 0 through 4. All probes were implemented as 3-layer, non-linear neural nets that mapped from model embeddings (of dimension 768) through 2 ReLU layers of dimension 1024, to a final layer to predict a word's depth or distance in the parse tree from other words. Probes were trained for up to 100 epochs, with early stopping based on validation set loss, using the Penn TreeBank dataset [Marcus et al., 1993]. I found that these hyperparameters produced more accurate probes than typically used in prior art, which capped training at 30 epochs and used single-layer probes.

Each probe was prefixed by a dropout layer with a parameter, α, that specified the proportion of inputs that were masked before being fed to the probe. At one extreme, probes with no dropout ($\alpha = 0$) might work if syntactic information was not redundantly encoded. Given results of such redundancy in the prior section, however, a positive value of α would likely reveal greater causal effects. (I tested the other extreme with $\alpha = 1$, corresponding to default probes that could not observe model embeddings; unsurprisingly, counterfactuals generated by such probes had no systematic effect on model predictions.) Counterfactual embeddings were created via gradient descent through trained probes (with dropout disabled), as described in Section 3.3.2. That is, new embeddings, Z', were generated to decrease the loss between $p(Z')$ and a desired parse. This loss is dubbed the counterfactual loss.

I recorded two types of results from my experiments. First, I visualized the effect of interventions, by layer, for a particular dropout rate and counterfactual loss. This revealed that, typically, earlier layers in models were more susceptible to interventions. Second, I devised an aggregate metric for the average difference, across all layers, in model outputs for counterfactuals generated with different parses. This showed how lower counterfactual losses (i.e., more interventions) and higher dropout typically revealed larger effects.

Additionally, I note that the probes were trained to parse single sentences, but two of the models (QA and NLI) accepted two sentences as inputs. For both models, counterfactual embeddings were creating by only updating the syntactically-ambiguous sentence and then concatenating it to the unaltered other embeddings.

Masked Language Model

I found the Mask model uses representations of syntax causally. I tested the model with two test suites exhibiting different forms of structural ambiguity. In the Coordination suite, exhibiting ambiguous coordination, one sentence reads, “The man saw the girl and the dog [MASK] tall.” One may plausibly insert either a plural or singular noun in the masked location, depending upon the syntactic interpretation of the sentence. In the NP/Z suite, masked words could be either adverbs or nouns, depending upon syntactic interpretations of the sentence. For example, one such sentence read, “As the author wrote the book

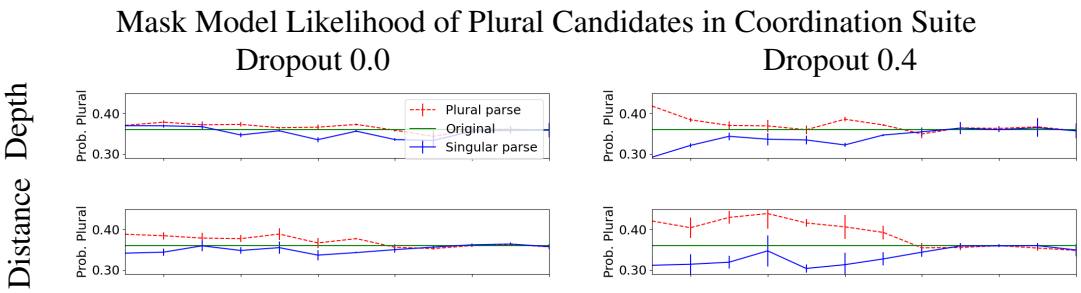


Figure 3-4: Mean and standard deviation probabilities over 5 trials for plural candidates using the original embeddings (green) or counterfactual embeddings favoring plural (dashed red) or singular (solid blue) parses. Counterfactual embeddings generated by both depth- and distance-based probes with greater dropout rates caused the greatest shift in model predictions.

[MASK] grew" where acceptable predictions might include "it" or "quickly." I generated test suite sentences using a template-based method; details of the prompts (and all prompts in this work) are included in Appendix 6.2.

The results of passing Z' generated from different parses in the Coordination suite through the rest of the Mask model are plotted in Figure 3-4. The three plotted lines correspond to the model output using the normal embeddings (green), using Z' generated according to a parse favoring plural verbs (red dashed), or using Z' generated using parses implying singular verbs (blue solid). The y axis corresponds to the probability the model assigned to words implying a plural interpretation ("were," "are," and "as") fitting in the masked location, normalized by the sum of probabilities assigned to those plural words or singular words ("was" and "is"). If the Mask model uses syntactic representations correctly, counterfactuals from plural parses should increase the probability of plural words.

I indeed found that effect, although it is clearest when using dropout probes. The causal effects using standard probes with no dropout are plotted in the left column; notably, distance-based probes revealed the desired effect, but depth-based probes had little to no effect. Conversely, when using dropout probes with $\alpha = 0.4$ (right column), I found much larger effects.

Averaging across all layers, I also measured the mean difference in output when using counterfactual embeddings generated according to different parses. Intuitively, this generated a single number that captured the average difference between the red and blue lines in

the plots in Figure 3-4.

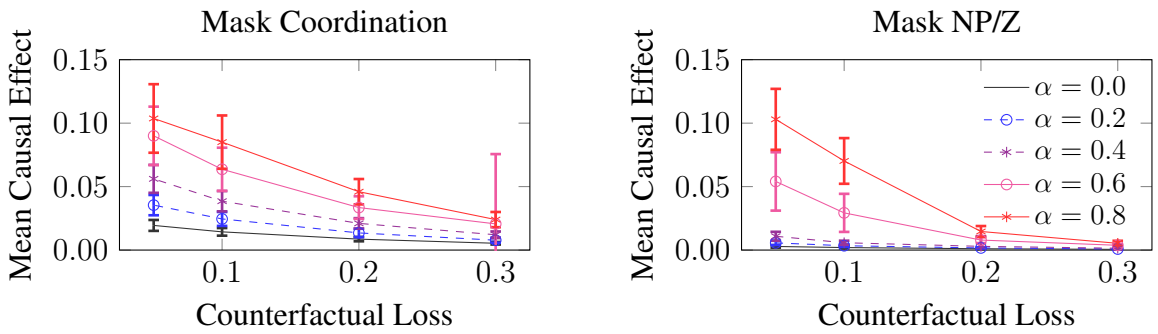


Figure 3-5: For the Coordination (left) and NP/Z (right) suites, interventions to a lower counterfactual loss (x axis) and with higher-dropout probes (different curves) revealed the greatest causal effects. Means and standard errors.

For a range of dropout values and counterfactual losses, I plotted the mean causal effect for the Coordination and NP/Z suites in Figure 3-5, using distance probes. For a given counterfactual loss, using higher dropout probes produced larger effects. In addition, lower counterfactual losses (corresponding to more gradient steps) induced greater effects. These trends also held true for depth-based probes (Appendix 6.4). Overall, using the Mask model, I found important evidence that models use redundant representations of syntax in making predictions.

QA Model

I also found that, to varying degrees depending upon the syntactic structure, the QA model used representations of syntax causally through a series of similar causal analysis experiments using syntactically-ambiguous inputs. The QA model is a BERT-based model fine-tuned on a question-answering task to map from context and a question to a continuous span of the context that answered the question [Rajpurkar et al., 2016].

I performed experiments using depth- and distance-based probes, using dropout values at increments of 0.1 from 0 to 0.9. I used three test suites for analyzing the causal use of syntax in the QA model: "Coordination", "Relative Clause" (RC), and a "Noun Phrase/Verb Phrase" (NP/VP) suite. The Coordination suite consisted of 256 prompts with coordination ambiguity like, "I saw the men and the women were tall. Who was tall?" The

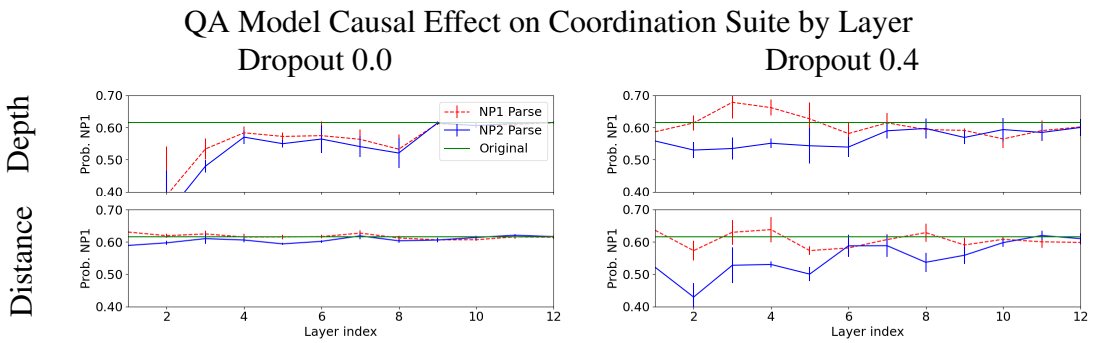


Figure 3-6: Causal effects for the QA model using depth- (top row) and distance- (bottom row) based probes with dropout of 0 (left column) or 0.4 (right column) on the Coordination corpus to counterfactual loss 0.05. Dropout probes produce more stable and larger effect sizes. Means and standard deviations over 5 trials plotted.

RC suite consisted of 193 prompts with attachment ambiguity of a relative clause like, "I saw the women and the men who were tall. Who was tall?" The NP/VP suite consisted of 256 prompts like, "The girl saw the boy with the telescope. Who had the telescope?" Prompts were designed such that answers were dictated by syntactic interpretations.

Findings for the Coordination suite are plotted in Figure 3-6. On the y axis, I plotted the model's prediction of words in the first noun phrase (NP1) starting the answer. Correct causal use of representations of syntax would move the red line (corresponding to parses indicating NP1) above the original outputs, in green, and the blue line (for the other parse) below.

As in the Mask model experiments, I found evidence that QA models use representations of syntax causally. For zero-dropout probes, plotted in the left column of Figure 3-6, I found that depth probes produced noisy results and distance-based probes had only small effects. In contrast to the standard probes, higher-dropout probes, plotted in the right column, revealed much larger effects of syntactic interventions.

More systematic analysis for all dropout rates, using distance and depth-based probes for all 3 test suites confirmed these trends. I plotted the aggregate metrics for all suites using depth probes in Figure 3-7. The causal effects were smaller in the RC and NP/VP suites than in the Coordination suite, indicating that the model may have learned a weaker causal link for these syntactic relations. Nevertheless, all suites demonstrate the importance of using dropout in probes: without dropout (solid black curve), the causal effects were

smaller than for any positive dropout rate.

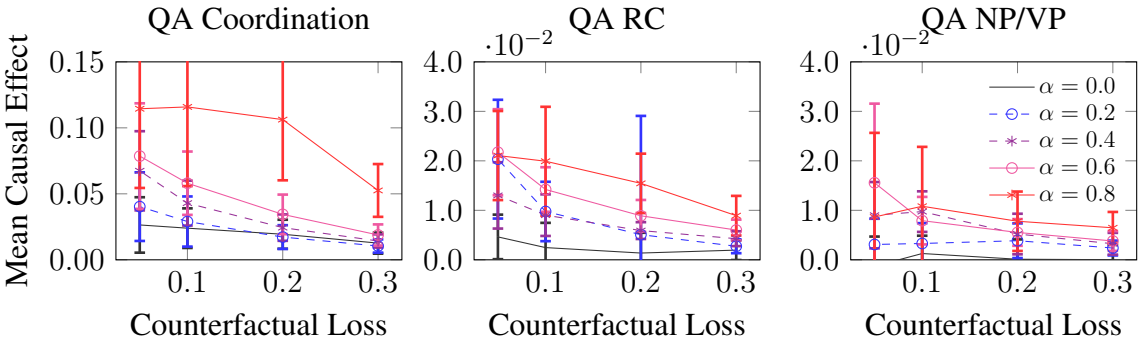


Figure 3-7: Mean causal effects when using depth-based probes for the QA test suites. Smaller counterfactual losses and higher dropout rates typically induced greater effects, although the scale of the effects varied by suite (note different axis scales). Means and standard deviations over 5 trials.

I note briefly that the causal effects uncovered by dropout probes may not be solely attributed to dropout probes performing better at their parsing task. In fact, adding dropout worsened probe performance according to typical probe performance metrics (Appendix 6.5).

NLI Model

Lastly, I performed similar causal analysis on the NLI and NLI-HANS models and, in contrast to the Mask and QA models, I found no evidence for the causal use of syntax using any probes for either model. The NLI model was finetuned on just the MNLI corpus, and the NLI-HANS model was finetuned with both the MNLI and HANS corpora, based on code from Gao et al. [2021]. The NLI model had a test set accuracy of 86%, and the NLI-HANS model had test set accuracy of 93%.

I used a test suite based on the Coordination suite already introduced in this work: an example prompt was "The person saw the keys in the cabinets which are green. The keys are green." The models had to classify these inputs among the three classes of entailment, contradiction, or neutrality.

Ultimately, I failed to find any evidence that either the NLI or the NLI-HANS model used syntactic information causally. The models always predicted entailment for all prompts, whether using original embeddings or counterfactuals generated for different parses. I used distance probes with dropout values from 0 to 0.9 and created counterfactuals for losses

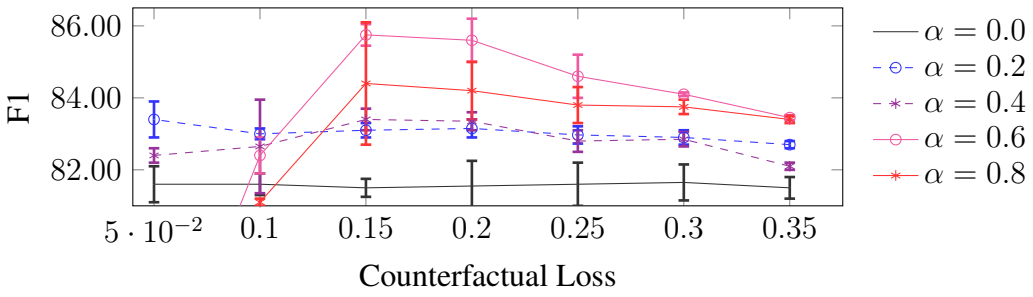


Figure 3-8: Using higher dropout probes (different curves) and lower counterfactual losses (x axis) allowed me to inject structural information into model embeddings, boosting model F1 scores. Care must be taken, however, to not perturb embeddings too much (high dropout with low counterfactual loss). Means and standard deviations plotted over 5 trials.

from 0.05 to 0.3 and never observed a shift in predicted probability mass of more than 1% when using counterfactuals. Unfortunately, this suggests that simply augmenting the MNLI dataset with HANS may not be enough to produce a model that uses syntactic information causally.

3.4.3 Boosting Performance with Probes

Earlier, I demonstrated that the QA model causally used representations of syntax for predictions; here, I show that one can improve QA model performance at test time by "injecting" syntactic information into embeddings.

I designed a new, syntactically challenging "Intervene" test suite of 288 prompts for the QA model. Example prompts include "The person saw the keys by the cabinet which was green. What was green?" and "The person saw the keys by the cabinet which were green. What was green?" Answering correctly ("the cabinet" first and "the keys" second) depends upon using noun-verb agreement. I used template-generated parse trees for each sentence and distance probes to create counterfactual embeddings for each sentence at layer 4 of the QA model. Layer 4 was chosen based on performance on a validation dataset (Appendix 6.3).

I passed the original and counterfactual embeddings through the QA model and measured performance on a test suite. F1 performance is plotted in Figure 3-8; exact match metrics had similar trends. Typically, higher-dropout probes improved performance more,

although the highest-dropout probes deteriorated for the lowest counterfactual losses. I hypothesize that this deterioration corresponded to generating out-of-distribution embeddings, but this topic warrants further study.

Lastly, I performed a similar experiment using the NLI and NLI-HANS models using 486 prompts drawn from the HANS dataset like "The doctor near the actor danced. The actor danced" [McCoy et al., 2019]. The NLI model achieved 50% accuracy (always predicting entailment) and the NLI-HANS model achieved 99% accuracy. Neither model's accuracy changed significantly when using counterfactuals with the correct parse for the first sentence, yet again indicating that these models may not use representations of syntax causally.

3.5 Contributions and Conclusion

In this chapter, I designed and evaluated a method for causal analysis of trained language models. A series of technical contributions combined to inform my method: 1) using causal, rather than correlative, methods enables novel insight into model behavior and 2) intuitions of redundantly encoded information in model embeddings motivated "dropout probes" a new neural probing architecture. Results from experiments confirm such intuitions and reveal important directions for future research. First, I showed that models indeed encode syntactic information redundantly. Second, dropout probes, unlike standard probes proposed in prior literature, reveal the large role representations of syntax can play in mediating model predictions Lastly, by injecting syntactic information at test time in syntactically-challenging domains, I showed how to increase model performance without retraining.

Despite this step towards better understanding of pretrained models, important future work remains. Natural extensions include studying pretrained models beyond those considered in this work. (In ongoing preliminary work, I have found some evidence that representations of syntax mediate Mistral-7B predictions, although the effect size appears smaller than for BERT.) Further analysis of the probe-based interventions is also warranted. How much do embeddings change during gradient updates? Do other interventions, such as

Elazar et al. [2020] induce similar effects, both in embedding and prediction spaces? I leave such questions to future research that can build upon my particular method using gradient-based updates and dropout probes.

Chapter 4

Contributions

In this thesis, I have explored two broad areas of research. In Chapter 2, I investigated how cognitively motivated losses could induce neural network agents to learn more human-like communication systems; in Chapter 3, I developed a causal probing method to test whether pre-trained models used human-like representations of language. While both chapters considered language-related domains, the underlying computational methods of many of these works apply to many settings, which fall outside the scope of this thesis (Figure 4-1). I have begun investigating such extensions, but important future work, both in language and other domains, remains.

4.1 Summary of Contributions

In each of the two technical chapters of this thesis, I introduced novel technical methods that, when applied, shed some light on human-like AI systems.

4.1.1 Chapter 2 - Information Theoretic Emergent Communication

In Chapter 2, I introduced a Information Theoretic Emergent Communication (ITEC) framework to induce more human-like communication among artificial agents. The framework combines cognitively-motivated terms for utility, informativeness, and complexity when training agents. Utility reflects a pressure for task-specific communication, informative-

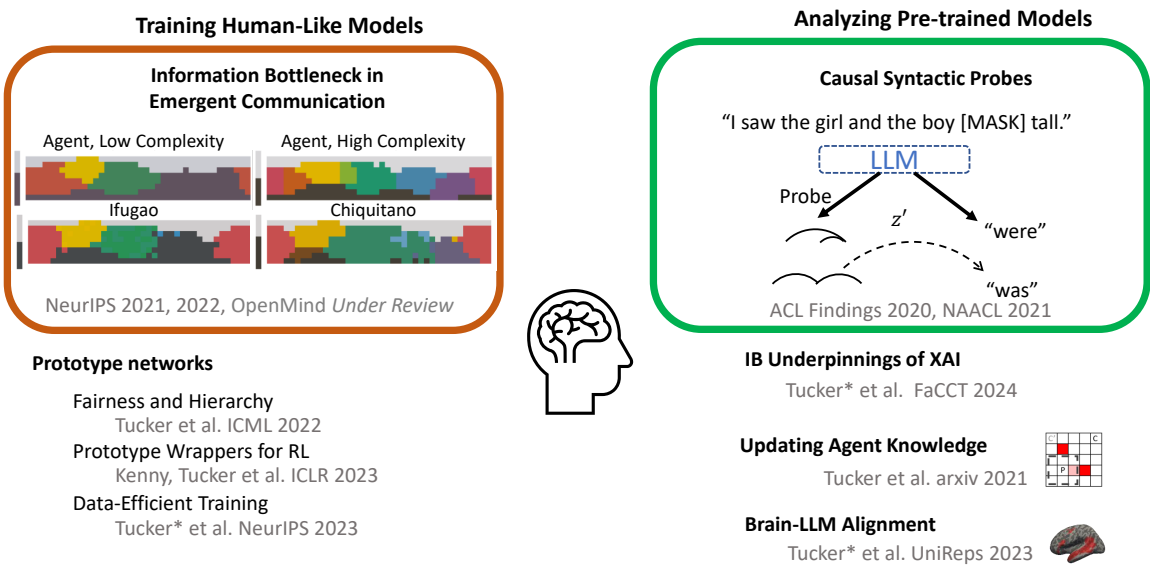


Figure 4-1: In this thesis, I discuss an Information Bottleneck method for inducing more human-like emergent communication (Chapter 2; left in diagram) and a causal probing mechanism for studying the role of syntax in model predictions (Chapter 3; right in diagram). These are particular instances of broader research directions towards training more human-like models or analyzing pre-trained models, some of which I have pursued in papers not included in this thesis (bottom).

ness a pressure for task-agnostic communication, and complexity for simpler lexicons. The ITEC allows one to control the relative importance of each of the three terms.

I proposed a novel method that can be trained within the ITEC framework to induce complexity-limited discrete communication, embedded in a continuous space. My method, dubbed the Vector Quantized Variational Information Bottleneck (VQ-VIB), supports variational bounds on complexity, which is used to penalize complexity in training. Unlike prior Variational Information Bottleneck methods, VQ-VIB employs vector quantization to produce *discrete* representations, reflecting a desire to match the discrete nature of words in natural language. The VQ-VIB methods is a general framework that extends beyond a single implementation, and indeed we experiment with several, subtly different, neural network architectures.

Overall, the ITEC framework allows one to directly control the utility, informativeness, and complexity of communication, which indirectly control high-level properties such as generalization and alignment. In three experiment domains, I showed how tun-

ing the weights for each of the ITEC terms led to expected changes such as simpler lexicons as complexity decreased. Furthermore, in domains with human data, I showed that controlling the complexity of emergent communication was necessary to achieve optimal alignment with natural language representations. Lastly, across domains, VQ-VIB models outperformed even strengthened baselines, likely due to its ability to represent discrete meanings in a continuous vector space.

4.1.2 Chapter 3 - Causal Probing for Syntax

In Chapter 3, I developed a causal probing method for establishing whether representations of syntax mediate language model predictions. Using this causal probing technique, I found that some model predictions appeared to be affected by representations of syntactic structure, much as human language understand is. My contributions are thus twofold: a causal probing technique and the findings from applying this technique.

The causal probing technique addresses limitations of correlative probing methods by intervening in a pre-trained model's latent representation. At a high level, my method uses a gradient-descent approach and pre-trained probes to change latent representations according to probe predictions. This method generates counterfactual representations, which can be passed through the remainder of the pre-trained model. Unlike correlative probing methods, therefore, my method reveals how models actually *use* representations.

In applying my technique to pre-trained BERT models with syntactic probes, I found that some, but not all, models appeared to use representations of syntax in mediating their predictions. This is an interesting, if slightly mixed, result. While humans use representations of syntax in understanding and producing language, it is unclear why some models may or may not learn to do so.

4.2 Limitations and Extensions

While I have made some contributions towards guiding and analyzing language-related neural networks in this thesis, many important questions, even in the narrow domains I consider, remain. These questions indicate limits of the current work, as well as excit-

ing directions for the future. Here, I include discussion of only a subset of such myriad questions.

4.2.1 Utility and Informativeness

How do tradeoffs between utility and informativeness regulate abstractions in neural network and natural language communication?

In Chapter 2, I explain the ITEC framework, which maximizes utility, maximizes informativeness, and minimizes complexity. Clearly, complexity is in tension with high utility or high informativeness. The relationship between utility and informativeness, however, is less clear. For fixed complexity level, increasing pressures for utility or informativeness would likely shift the nature of communicated information to be more about actions or inputs, respectively.

Although the ITEC framework exposes the dials to control the tradeoff between utility and informativeness, the experiments do not fully explore the effects of such tradeoffs. In both the color and 2D world domains, utility and informativeness are nearly identical. In the ManyNames domain, however, utility and informativeness are distinct because the target and distractor colors are selected from different categories by design. Nevertheless, further experiments in domains in which utility and informativeness are more clearly different could shed important new light on this complex tradeoff. In preliminary experiments in highly simplified domains, I have found that utility and informativeness pressures indeed induce expected effects on VQ-VIB$_{\mathcal{C}}$ communication. In future experiments, one could better explore how varying task-specific vs. task-agnostic information in communication (modulated via informativeness and utility losses) aligns with human representations.

4.2.2 Multi-task Abstractions

How do multi-task frameworks affect (complexity-limited) representations?

The ITEC framework models utility as a task-specific reward function to maximize; this corresponds to a notion of humans communicating in order to accomplish a goal. In practice, humans often seek to accomplish many goals, and speakers and listeners must

reason jointly about the distribution over tasks. A natural extension of the ITEC framework, as yet unexplored in this thesis, would be to train agents according to a distribution of utility functions. Such multi-task training will likely change the types of representations that agents learn, which in turn could raise important questions about meta-learning or task-conditioned efficient representations. Of course, care must be taken in justifying a cognitively-plausible distribution of training tasks.

4.2.3 In-Distribution Causal Probing

Can probe-based interventions remain in-distribution?

The probe-based interventions in Chapter 3 are designed to modify representations of a sentence's syntactic structure; it is important to ask if such interventions modify any other parts of the sentence's representation. Ideally, the counterfactual representations should correspond to "what would the representation of this sentence be if the syntactic structure change?" In traditional causal literature, therefore, such counterfactual representations would therefore correspond to minimal changes to the original representations.

Unfortunately, ascertaining what constitutes a minimal change to a representation, or even if it remains in-distribution, is challenging. Syntax is a fundamentally latent concept, so counterfactual generation must occur in the largely uninterpretable latent space. In the experiments presented in this thesis, model behavior was largely consistent with most interventions being in-distribution: if the counterfactual representations were out of distribution, one might expect random changes in model predictions rather than the syntactically-consistent shifts in predictions that I observed. Nevertheless, further advances in counterfactual evaluation, as well as theoretical advances in causal problem framings, would strengthen my results.

In ongoing work, I have begun exploring one method of evaluation the probe-based counterfactual method. Instruction-tuned models are trained to respond appropriately to explicit user instructions or prompts, and in early testing of simple cases, they appear somewhat sensitive to syntactic prompts. For example, one may input to a model, "In the following sentence, 'telescope' modifies the seeing. The girl saw the boy with the tele-

scope. Who had the telescope?' Varying prompts tends to change model predictions. This prompt-based intervention method offers an important alternative to generate counterfactual representations that are, by definition, in distribution. Comparing such representations to probe-based counterfactual representations is an important direction for calibrating both methods and better understanding model behavior.

4.2.4 Probing beyond Syntax

Can probe-based counterfactuals be used for non-syntactic analysis?

In this thesis, I explored how probe-based counterfactuals can change representations of syntax in pre-trained language models, but one may also ask if the same technique may be applied to different domains. Indeed, it can. Dropout probes and a gradient-based intervention method, can be generally used to modify different neural network representations in a variety of human-interpretable ways. In early experiments, I showed how dropout probes may be used to modify an image-classifier's predictions according to semantic traits (animal vs. vehicle) and the actions of neural network policies in a simulated 2D world. In general, to apply my method, one needs a pre-trained neural model and a probe trained to predict the desired property to manipulate. Because the probes modify latent concepts, in theory this technique is widely applicable; I note, however, that applications must consider the same limitations listed in the prior section of remaining in-distribution.

Bibliography

Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes. *ArXiv*, abs/1610.01644, 2017.

Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes, 2018.

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.

Afra Amini, Tiago Pimentel, Clara Meister, and Ryan Cotterell. Naturalistic causal probing for morpho-syntax. *Transactions of the Association for Computational Linguistics*, 11: 384–403, 2023. doi: 10.1162/tacl_a_00554. URL https://aclanthology.org/2023.tacl-1.23.

Guy Aridor, Rava Azeredo da Silveira, and Michael Woodford. Information-constrained coordination of economic behavior. Technical report, National Bureau of Economic Research, 2024.

Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeshwar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and Devon Hjelm. Mutual information neural estimation. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 531–540. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/belghazi18a.html.

Yonatan Belinkov. Probing Classifiers: Promises, Shortcomings, and Advances. *Computational Linguistics*, pages 1–13, 11 2021. ISSN 0891-2017. doi: 10.1162/coli_a_00422. URL https://doi.org/10.1162/coli_a_00422.

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

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark

Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020.

Emil Carlsson, Devdatt P. Dubhashi, and Fredrik Johansson. Learning approximate and exact numeral systems via reinforcement learning. *ArXiv*, abs/2105.13857, 2021.

Emil Carlsson, Devdatt Dubhashi, and Terry Regier. Iterated learning and communication jointly explain efficient color naming systems, 2023.

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.

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, 2021a.

Rahma Chaabouni, Florian Strub, Florent Altché, Eugene Tarassov, Corentin Tallec, Elnaz Davoodi, Kory Wallace Mathewson, Olivier Tieleman, Angeliki Lazaridou, and Bilal Piot. Emergent communication at scale. In *International Conference on Learning Representations*, 2021b.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In Alessandro Moschitti, Bo Pang, and Walter Daelemans, editors, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar, October 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1179. URL https://aclanthology.org/D14-1179.

Noam Chomsky. *Aspects of the Theory of Syntax*. MIT press, 1965.

Alexis Conneau, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2126–2136, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1198. URL https://aclanthology.org/P18-1198.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the*

2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://www.aclweb.org/anthology/N19-1423.

Diego Doimo, Aldo Glielmo, Sebastian Goldt, and Alessandro Laio. Redundant representations help generalization in wide neural networks. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=lC5-Ty_0FiN.

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.

Yanai Elazar, Shauli Ravfogel, Alon Jacovi, and Yoav Goldberg. Amnesic Probing: Behavioral Explanation with Amnesic Counterfactuals. *arXiv e-prints*, art. arXiv:2006.00995, June 2020.

Yang Gao, Nicolò Colombo, and Wei Wang. Adapting by pruning: A case study on BERT. *CoRR*, abs/2105.03343, 2021. URL https://arxiv.org/abs/2105.03343.

Jon Gauthier and R. Levy. Linking artificial and human neural representations of language. In *EMNLP/IJCNLP*, 2019.

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.

Mario Giulianelli, Jack Harding, Florian Mohnert, Dieuwke Hupkes, and Willem Zuidema. Under the hood: Using diagnostic classifiers to investigate and improve how language models track agreement information. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 240–248, Brussels, Belgium, November 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-5426. URL https://aclanthology.org/W18-5426.

Yash Goyal, Amir Feder, Uri Shalit, and Been Kim. Explaining classifiers with causal concept effect (cace). *arXiv preprint arXiv:1907.07165*, 2019.

Eleonora Gualdoni, Andreas Mädebach, Thomas Brochhagen, and Gemma Boleda. What's in a name? a large-scale computational study on how competition between names affects naming variation. 2022.

Rowan Hall Maudslay, Josef Valvoda, Tiago Pimentel, Adina Williams, and Ryan Cotterell. A tale of a probe and a parser. In *Proceedings of the 58th Annual Meeting of*

the Association for Computational Linguistics, pages 7389–7395, Online, July 2020a. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.659. URL https://aclanthology.org/2020.acl-main.659.

Rowan Hall Maudslay, Josef Valvoda, Tiago Pimentel, Adina Williams, and Ryan Cotterell. A tale of a probe and a parser. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7389–7395, Online, July 2020b. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.659. URL https://www.aclweb.org/anthology/2020.acl-main.659.

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

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

John Hewitt and Percy Liang. Designing and interpreting probes with control tasks. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2733–2743, Hong Kong, China, November 2019a. Association for Computational Linguistics. doi: 10.18653/v1/D19-1275. URL https://aclanthology.org/D19-1275.

John Hewitt and Percy Liang. Designing and interpreting probes with control tasks. In *Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2019b.

John Hewitt and Christopher D Manning. A structural probe for finding syntax in word representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4129–4138, 2019.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9 (8):1735–1780, nov 1997. ISSN 0899-7667. doi: 10.1162/neco.1997.9.8.1735. URL https://doi.org/10.1162/neco.1997.9.8.1735.

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

Divyansh Kaushik, Eduard Hovy, and Zachary Lipton. Learning the difference that makes a difference with counterfactually-augmented data. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=Sklgs0NFvr.

Paul Kay, Brent Berlin, Luisa Maffi, William R Merrifield, and Richard Cook. *The world color survey*. CSLI Publications Stanford, CA, 2009.

Charles Kemp, Yang Xu, and Terry Regier. Semantic typology and efficient communication. *Annual Review of Linguistics*, 4:109–128, 01 2018.

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.

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.

Angeliki Lazaridou, Karl Moritz Hermann, Karl Tuyls, and Stephen Clark. Emergence of linguistic communication from referential games with symbolic and pixel input. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018.

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

Jiaoda Li, Ryan Cotterell, and Mrinmaya Sachan. Probing via prompting. In Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz, editors, *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1144–1157, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-main.84. URL https://aclanthology.org/2022.naacl-main.84.

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.

Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. Assessing the ability of lstms to learn syntax-sensitive dependencies. *Transactions of the Association for Computational Linguistics*, 4:521–535, 2016.

Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. Linguistic knowledge and transferability of contextual representations. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1073–1094,

Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1112. URL https://aclanthology.org/N19-1112.

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.

Scott M. Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 4768–4777, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

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.

Andreas Mädebach, Ekaterina Torubarova, Eleonora Gualdoni, and Gemma Boleda. Effects of task and visual context on referring expressions using natural scenes. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 44, 2022.

Tyler Malloy. *Resource-Rational Cognitive Modelling: An Information-Theoretic Approach*. PhD thesis, Rensselaer Polytechnic Institute, 2022.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2): 313–330, 1993. URL https://aclanthology.org/J93-2004.

Tom McCoy, Ellie Pavlick, and Tal Linzen. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1334. URL https://aclanthology.org/P19-1334.

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.

WS Mokrzycki and M Tatol. Colour difference ΔE - a survey. *Machine Graphic and Vision*, 8, 2012.

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.

Luca Moschella, Valentino Maiorca, Marco Fumero, Antonio Norelli, Francesco Locatello, and Emanuele Rodolà. Relative representations enable zero-shot latent space communication. In *The Eleventh International Conference on Learning Representations*, 2023.

Vedant Nanda, Till Speicher, John P Dickerson, Krishna P. Gummadi, Soheil Feizi, and Adrian Weller. Diffused redundancy in pre-trained representations. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=LyAuNoZkGP.

Judea Pearl and Dana Mackenzie. *The Book of Why: The New Science of Cause and Effect*. Basic Books, Inc., USA, 1st edition, 2018. ISBN 046509760X.

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.

Tiago Pimentel, Josef Valvoda, Rowan Hall Maudslay, Ran Zmigrod, Adina Williams, and Ryan Cotterell. Information-theoretic probing for linguistic structure. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4609–4622, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.420. URL https://aclanthology.org/2020.acl-main.420.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ Questions for Machine Comprehension of Text. *arXiv e-prints*, art. arXiv:1606.05250, 2016.

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.

Emily Reif, Ann Yuan, Martin Wattenberg, Fernanda B Viegas, Andy Coenen, Adam Pearce, and Been Kim. Visualizing and measuring the geometry of bert. 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. URL https://proceedings.neurips.cc/paper/2019/file/159c1ffe5b61b41b3c4d8f4c2150f6c4-Paper.pdf.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should i trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 1135–1144, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450342322. doi: 10.1145/2939672.2939778. URL https://doi.org/10.1145/2939672.2939778.

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.

Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *ICCV*, pages 618–626. IEEE Computer Society, 2017. ISBN 978-1-5386-1032-9. URL http://dblp.uni-trier.de/db/conf/iccv/iccv2017.html#SelvarajuCDVPB17.

Carina Silberer, Sina Zarieß, and Gemma Boleda. Object Naming in Language and Vision: A Survey and a New Dataset. In *Proceedings of the 12th International Conference on Language Resources and Evaluation (LREC 2020)*, 2020.

Karolina Stanczak, Lucas Hennigen, Adina Williams, Ryan Cotterell, and Isabelle Augenstein. A latent-variable model for intrinsic probing. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37:13591–13599, 06 2023. doi: 10.1609/aaai.v37i11.26593.

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.

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

Ilia Sucholutsky and Thomas L. Griffiths. Alignment with human representations supports robust few-shot learning, 2023.

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, 2016a. Curran Associates Inc.

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

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations*, 2014.

Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R Thomas McCoy, Najoung Kim, Benjamin Van Durme, Sam Bowman, Dipanjan Das, and Ellie Pavlick. What do you learn from context? probing for sentence structure in contextualized word representations. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=SJzSgnRcKX.

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.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023.

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, 2021a.

Mycal Tucker, Peng Qian, and Roger Levy. What if this modified that? syntactic interventions with counterfactual embeddings. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 862–875, Online, August 2021b. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-acl.76. URL https://aclanthology.org/2021.findings-acl.76.

Mycal Tucker, Tiwalayo Eisape, Peng Qian, Roger Levy, and Julie Shah. When does syntax mediate neural language model performance? evidence from dropout probes. In Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz, editors, *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5393–5408, Seattle, United States, July 2022a. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-main.394. URL https://aclanthology.org/2022.naacl-main.394.

Mycal Tucker, Roger P. Levy, Julie Shah, and Noga Zaslavsky. Trading off utility, informativeness, and complexity in emergent communication. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022b.

Mycal Tucker, Julie Shah, Roger Levy, and Noga Zaslavsky. Towards Human-Like Emergent Communication via Utility, Informativeness, and Complexity. *Open Mind*, Under Review.

Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *Proceedings of the 31st International Conference on Neural Infor-*

mation Processing Systems, NIPS'17, pages 6309–6318, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Victor Veitch, Alexander D'Amour, Steve Yadlowsky, and Jacob Eisenstein. Counterfactual invariance to spurious correlations in text classification. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=BdKxQp0iBi8.

Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Simas Sakenis, Jason Huang, Yaron Singer, and Stuart Shieber. Causal mediation analysis for interpreting neural nlp: The case of gender bias, 2020.

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.

Zi Wang, Alexander Ku, Jason Baldridge, Thomas L Griffiths, and Been Kim. Gaussian Process Probes (GPP) for Uncertainty-Aware Probing. *arXiv preprint arXiv:2305.18213*, 2023.

Adina Williams, Nikita Nangia, and Samuel Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics, 2018. URL http://aclweb.org/anthology/N18-1101.

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

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. Huggingface's transformers: State-of-the-art natural language processing. *CoRR*, abs/1910.03771, 2019. URL http://arxiv.org/abs/1910.03771.

Noga Zaslavsky. *Information-Theoretic Principles in the Evolution of Semantic Systems*. Ph.D. Thesis, The Hebrew University of Jerusalem, 2020.

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.

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.

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.

Yilun Zhou, Serena Booth, Marco Tulio Ribeiro, and Julie Shah. Do feature attribution methods correctly attribute features? In *Proceedings of the 36th AAAI Conference on Artificial Intelligence*. AAAI, Feb 2022a.

Yilun Zhou, Marco Tulio Ribeiro, and Julie Shah. Exsum: From local explanations to model understanding. In *Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, July 2022b.

Chapter 5

Appendix: Emergent Communication:

5.1 Combinatorial codebook

In Chapter 2, I proposed two VQ-VIB architectures, VQ-VIB$_{\mathcal{N}}$ and VQ-VIB$_{\mathcal{C}}$, both of which use a learnable codebook to discretize communication. In this section, I show how simple changes to the neural network architectures support dramatic increases in the codebook sizes while retaining support for the variational bounds on complexity from the original architectures. Diagrams of both updates architectures are included in Figure 5-1.

In VQ-VIB$_{\mathcal{N}}$, a feedforward encoder outputs $[\mu_1, ..., \mu_n]$ and $[\Sigma_1, ..., \Sigma_n]$ for $\mu_i, \Sigma_i \in \mathbb{R}^{d/n}$. That is, rather than producing the mean and variance for a single $d-$dimensional normal distribution, the encoder outputs the parameters for n normal distributions, each in $\mathbb{R}^{d/n}$. The learnable codebook, $\mathcal{F} \approx \mathcal{D} = \zeta_i \in \mathbb{R}^{d/n}; i \in [1, K]$ is similarly updated so that each codebook element is in $\mathbb{R}^{d/n}$. Thus, each $z_i \sim \mathcal{N}(\mu_i, \Sigma_i)$ is discretized via the standard discretization process of snapping to the nearest element of the codebook. Lastly, the overall discrete communication is generated by concatenating all discrete subtokens, generating a final discrete communication vector $w \in \mathbb{R}^d$

In VQ-VIB$_{\mathcal{C}}$, the feedforward encoder and codebook are similarly updated to generate n, $\frac{d}{n}-$dimensional discrete representations that are concatenated together. As shown in Figure 5-1 b, the encoder generates n continuous representations: $z_i \in \mathbb{R}^{d/n}, i \in [1, n]$. As in VQ-VIB$_{\mathcal{N}}$, the learnable codebook defines K vectors: $\mathcal{F} \approx \mathcal{D} = \zeta_i \in \mathbb{R}^{d/n}; i \in [1, K]$. Each z_i is discretized using the VQ-VIB$_{\mathcal{C}}$ sampling process based on negative distance from

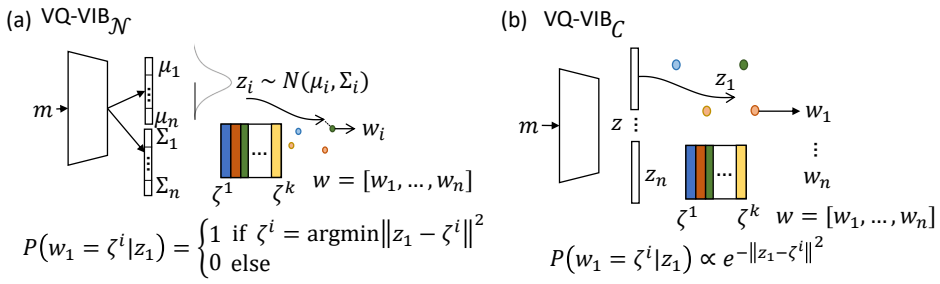


Figure 5-1: VQ-VIB$_{\mathcal{N}}$ (a) and VQ-VIB$_{\mathcal{C}}$ (b) speaker architectures for combinatorial codebooks.

z_i to each element of the codebook. Lastly, these discrete representations are concatenated into $w \in \mathbb{R}^d$.

Conceptually, updating VQ-VIB$_{\mathcal{N}}$ and VQ-VIB$_{\mathcal{C}}$ to support concatenations of discrete tokens only differs from the prior architecture by expanding the finite set of possible communication vectors. The architectures presented in Chapter 2 may be thought of as a special case of this more general combinatorial setup by setting $n = 1$. By increasing n, however, one can dramatically increase the effective codebook size of agents: e.g., for $K = 1024$, as used in the ManyNames experiments, $n = 1$ generates 1024 discrete representations, whereas $n = 4$ generates $1024^4 \approx 10^{12}$ representations, without increasing the number of weights in the neural nets. This increased codebook size in turn supported more complex and informative representations which in some cases improved model performance (e.g., for OOD generalization) but in others was unnecessary (e.g., maximum representation alignment was achieved at low complexity, so increasing n did not help alignment).

5.2 Penalizing Entropy

In Chapter 2, I defined the VQ-VIB training objective in Equation 2.13, which included a tie-breaking entropy loss. Here, I explain the motivation behind this term and how it was calculated for VQ-VIB$_{\mathcal{N}}$ and VQ-VIB$_{\mathcal{C}}$ models.

Prior art shows that, for any given complexity level, there are many possible IB-optimal solutions with different effective codebook sizes $k = |\{\zeta_i \in \boldsymbol{\zeta} : \sum_m \mathbb{P}(m)q(\zeta_i|m) > 0|$ [Zaslavsky, 2020]. This measurement of effective codebook size captures how many distinct tokens are used by a speaker. Intuitively, for the same complexity level, a speaker may use relatively few tokens (each token encoding a quite distinct meaning), or use many different tokens that are sampled quite stochastically (e.g., two tokens might encode the same meaning, and the speaker randomly chooses between the tokens).

I was interested in EC systems with the minimal k, for a given complexity level, as they encode the same amount of information with the smallest effective codebook size. Therefore, to bias EC agents towards small k, I wished to penalize the entropy of the categorical distribution over tokens, breaking the tie within a complexity equivalence class to favor small-codebook solutions.

For VQ-VIB$_\mathcal{C}$, I could directly measure and penalize the entropy of the categorical distribution, as it was calculated while sampling tokens: $\mathbb{P}(\zeta_i|m) \propto e^{||\zeta_i - z(m)||^2}$. VQ-VIB$_\mathcal{N}$, however, does not support exact calculation of the categorical distribution over tokens. I therefore approximated the categorical distribution by using the same probability function as for VQ-VIB$_\mathcal{C}$: proportional to $e^{||\zeta_i - z(m)||^2}$.

As shown in Equation 2.13, I weighted the entropy loss term by a small scalar value: $\epsilon * \lambda_C$. Given that I merely sought to penalize entropy within a complexity equivalence class, rather than use this term directly to control agents, I typically set ϵ to small values: in the 2D world environments, $\epsilon = 0.01$, and in the color and ManyNames domains, $\epsilon = 0.05$. These values were chosen by training teams to low complexity solutions and evaluating how many tokens were used. ϵ was increased until the number of tokens shrank to roughly $\log_2$ of the complexity.

5.3 Baseline architectures

Here, I elaborate upon some of the neural network architectures used in experiments other than the VQ-VIB methods.

The two discrete EC architectures I started with were onehot and Proto.. As discussed

in the main text, onehot agents mapped inputs to a onehot vector. Proto. agents, introduced by Tucker et al. [2021a], internally compute a onehot vector, which is then multiplied by a "prototype matrix" to generate discrete representations in a continuous space. While both methods generate discrete communication, neither supports variational bounds on complexity. Thus, I adapted these methods to generate strengthened baselines that fit within the ITEC framework.

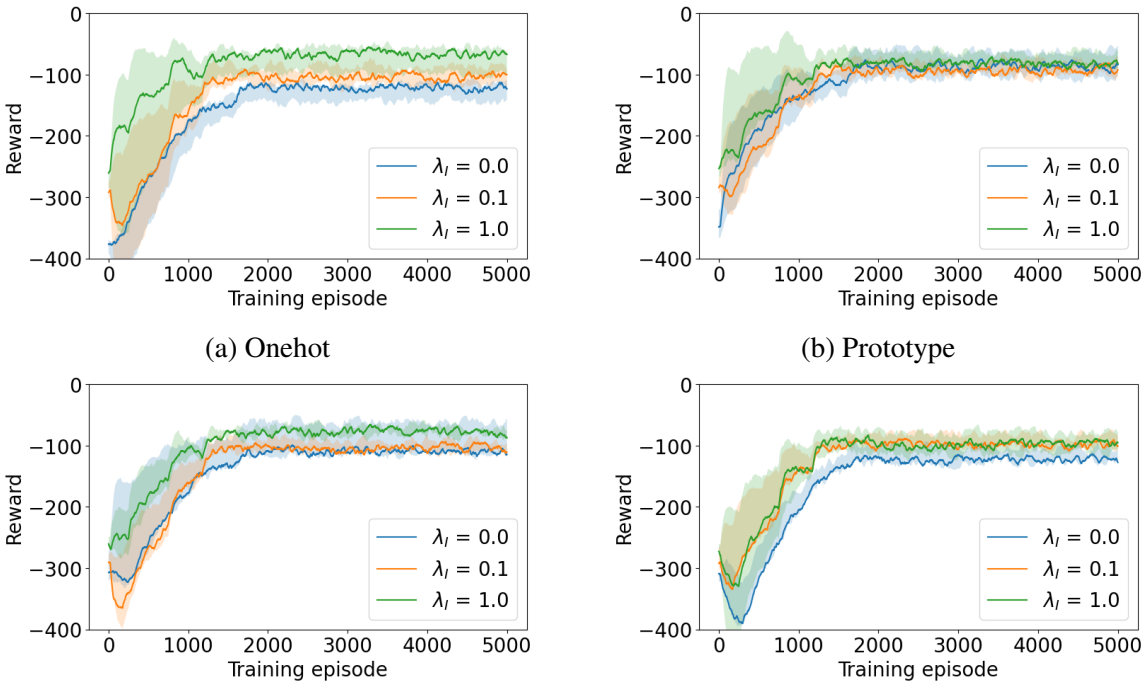
My updated onehot baseline used a variational bound on the categorical distribution over the communication dimension. As in VQ-VIB$_{\mathcal{C}}$, onehot agents generated a categorical distribution over which token to emit and sampled from this distribution using the gumbel-softmax trick [Maddison et al., 2017]. I therefore similarly bounded complexity via $I(m; z) \leq \mathbb{E}[D_{\mathrm{KL}}[q_\theta(w|m) \| r(w)]]$ where I set $r(w)$ to a uniform categorical distribution.

My updated Proto. baseline used a variational bound on complexity based on normal distributions. Variational Proto. agents used a standard Proto. speaker, which generates a discrete representation, z, in a continuous space. Agents also produced Σ, representing the variance of a normal distribution. Lastly, the Proto. agent's communication was generated by sampling from a normal, centered at the discrete representation, with variance Σ: $w \sim \mathcal{N}(z, \Sigma)$. Thus, Proto. agent communication passed through a discrete bottleneck but was ultimately continuous. Given this sampling mechanism, I used the variational bound on complexity often used in VAEs: $I(m; w) \leq \mathbb{E}[D_{\mathrm{KL}}[q_\theta(w|m) \| r(w)]]$, which is tractable for a unit normal prior, $r(w)$.

5.4 Training curves

In Figure 2-11, I showed training curves in the 2D simulated word for VQ-VIB$_{\mathcal{N}}$ and VQ-VIB$_{\mathcal{C}}$ agents. Increasing λ_I, the pressure for informativeness, caused both model types to converge faster and to a higher mean reward. For completeness, I include similar training curve results for onehot and Proto. agents in Figure 5-2.

As before, I found that, for all architecture types, increasing λ_I continued to induce higher-reward policies earlier in training. This indicates that informativeness can be a pow-



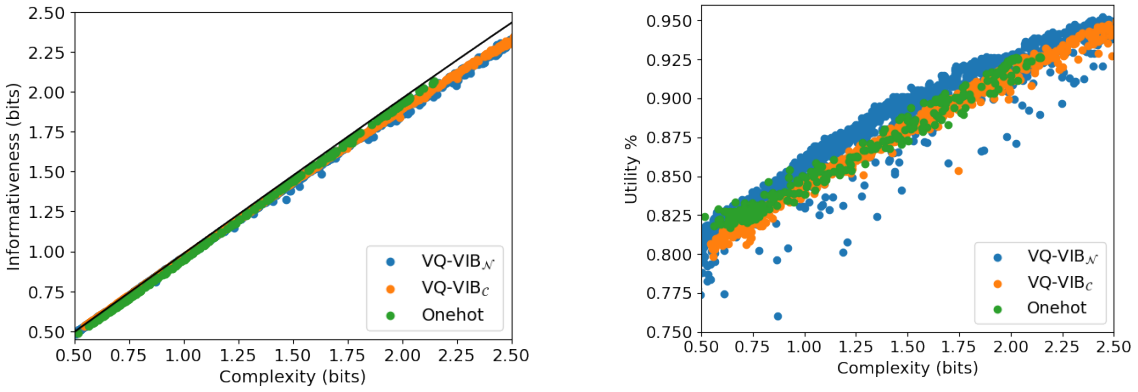
(a) Onehot

(b) Prototype

(c) VQ-VIB$_{\mathcal{N}}$

(d) VQ-VIB$_{\mathcal{C}}$

Figure 5-2: Team utility over as a function of training episode for different model architectures (different plots) and different λ_I values (different curves). Increasing λ_I improved performance for all model types.



(a) (b)

Figure 5-3: Informativeness vs. complexity (a) and utility vs. complexity (b) in the color reference game for models trained with REINFORCE. I observed similar trends as when training via backpropagation: all models were similarly IB-optimal (a), and VQ-VIB$_{\mathcal{N}}$ tended to achieve higher utility, for the same complexity, as other models.

erful indirect pressure towards high-utility policies, especially in multi-agent reinforcement learning settings, which are notoriously hard to train models in [Eccles et al., 2019].

5.5 Color Reference Game Further Results

In the main color reference game experiments, I trained agents via a supervision loss, which I backpropagated through the speaker and listener.

Here, I briefly present results from a similar setup but with a different training method: REINFORCE [Williams, 1992] Using this reinforcement-based training mechanism, the speaker was only indirectly trained to maximize team accuracy, without the ability to back-propagate through the listener. Prior art in EC has employed both REINFORCE and supervisory methods for reference games (including, in particular, color) and found that training with REINFORCE was less stable and led to communication with lower communicative complexity [Chaabouni et al., 2021a].

Results from my agents, trained with REINFORCE, are depicted in Figure 5-3. As before, I found that all models achieved near optimal informativeness and that VQ-VIB$_{\text{norm}}$ achieved greater utility than other methods (for a given complexity level).

Furthermore, I note that training with positive λ_I consistently induced more complex

communication than cases with $\lambda_I = 0$, as done in prior art. Chaabouni et al. [2021a] trained 180 teams from scratch to overcome random failures of model training; in my results I trained 5 teams from scratch and each team converged to useful communication. In addition, whereas the 180 teams trained by Chaabouni et al. [2021a] never learned communication more complex than 2.4 bits, I found that VQ-VIB models consistently surpassed 2.5 bits.

Thus, I found that the results in Chapter 2 generalized to different training mechanisms and, moreover, the ITEC framework appeared capable of addressing important limitations identified in earlier work.

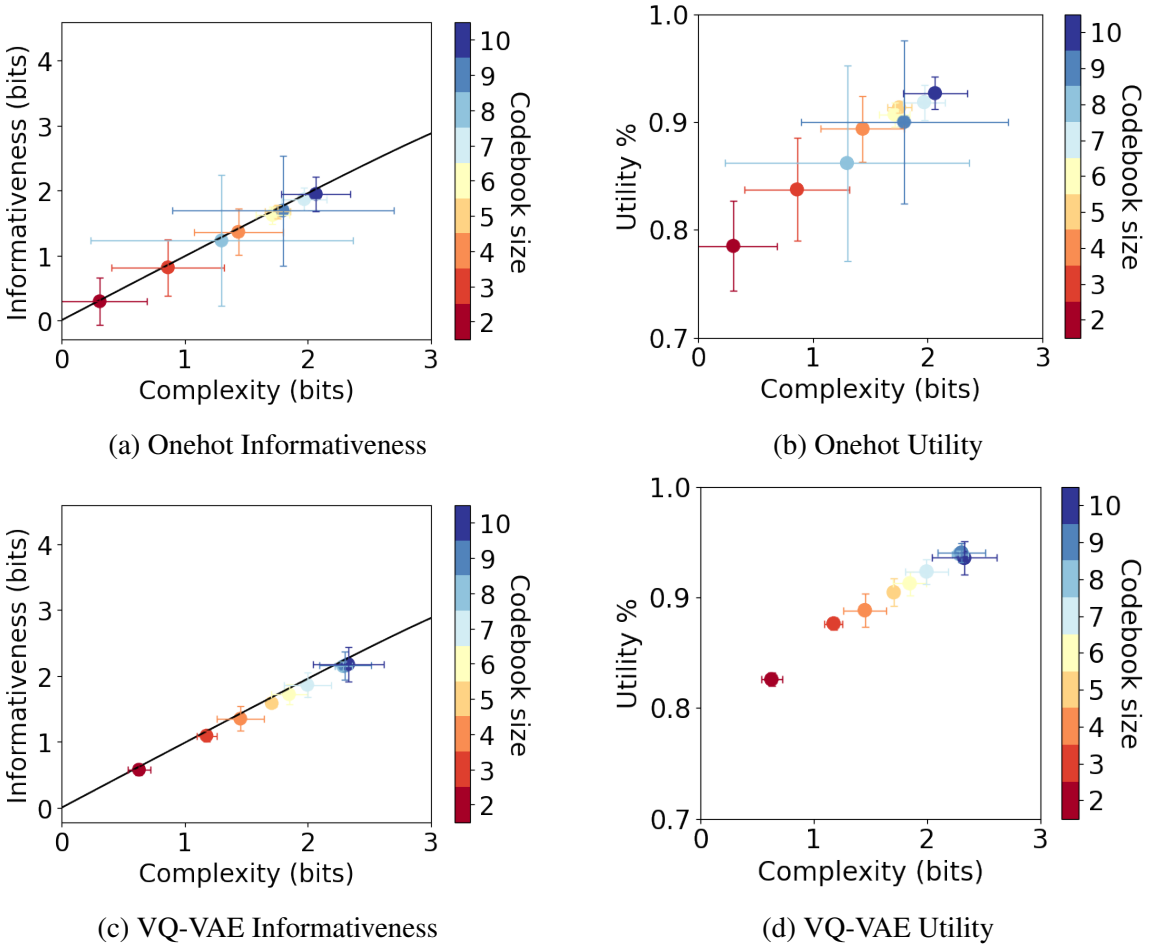
Lastly, when training agents with a supervision loss (not REINFORCE), I found, yet again, the importance of the ITEC framework in generating human-like naming systems. I compared to onehot and VQ-VAE agents trained with only utility rather than the full ITEC framework ($\lambda_I = \lambda_C = 0$). Results for such agents are depicted in Figure 5-4.

Each point in Figure 5-4 corresponds to the behavior for an agent with a hardcoded codebook size from 2 to 10. By varying the codebook size, I could indirectly control the resulting complexity of communication. This indirect control, however, affords less finegrained control than the ITEC framework, and is cognitively less plausible.

Therefore, in the final set of color experiments, I trained a variational version of onehot communication. Training within the ITEC framework, I reproduced important behaviors of smoothly modifying the complexity of communication. Metrics of efficiency and similarity to human languages for this improved onehot baseline, as well as VQ-VIB$_\mathcal{C}$ (and results already reported in the main paper) are included in Table 5.1. I note that, even as the Onehot - Variational efficiency metrics are similar to VQ-VIB models, VQ-VIB$_\mathcal{N}$ continued to achieve greater utility than onehot, due to the semantically-meaningful communication space.

5.6 ManyNames further results

In Chapter 2, I included OOD utility, functional alignment, and relative representation alignment results for some of the models in the ManyNames reference game. Here, I



(a) Onehot Informativeness

(b) Onehot Utility

(c) VQ-VAE Informativeness

(d) VQ-VAE Utility

Figure 5-4: Informativeness and utility vs. complexity for onehot and VQ-VAE models, for different hardcoded codebook sizes. Without informativeness or complexity pressures, varying the architecture was the only way to indirectly control important model behaviors.

	VQ-VIB$_{\mathcal{N}}$	VQ-VIB$_{\mathcal{C}}$	Onehot - Var.	Onehot	VQ-VAE
gNID	0.151 (0.00)	0.153 (0.01)	0.154 (0.00)	0.42 (0.16)	0.38 (0.15)
Efficiency loss	0.024 (0.00)	0.022 (0.00)	0.023 (0.01)	0.06 (0.01)	0.08 (0.01)

Table 5.1: Quantitative evaluation of artificial color communication systems, compared to human naming data and an IB-optimal bound. The three variational methods (VQ-VIB$_{\mathcal{N}}$, VQ-VIB$_{\mathcal{C}}$, and Onehot - Var.), trained in the ITEC framework were closer to human languages (lower human-agent gNID - generalization of Normalized Information Distance) and closer to the IB bound (lower efficiency loss) than utility-only methods (Onehot and VQ-VAE).

include results from varying n and varying C at test time.

5.6.1 Generalization

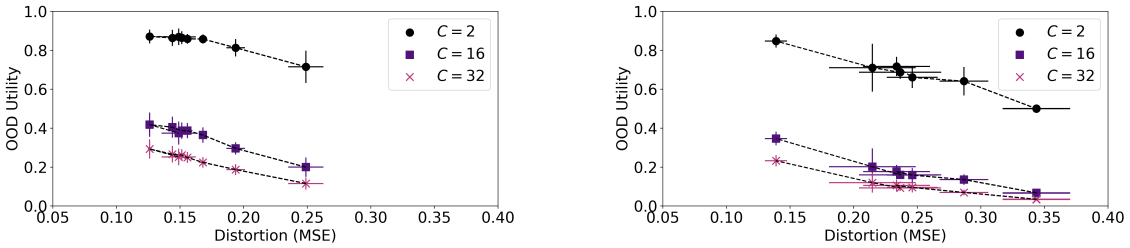
Trends from the main generalization experiments indicated that increasing λ_I and informativeness tended to enable greater generalization performance across models. Here, I examine such trends as I varied important task and model parameters.

First, by varying the number of candidate images used at test time, C, I exposed important differences between model capabilities. Chaabouni et al. [2021b] found that such analysis was important for uncovering differences in generalization capabilities. Therefore, I tested agents with $C \in [2, 16, 32]$; note, however, that all models were trained with $C = 2$. Thus, increasing C tested a different sort of generalization to a harder task setting. Second, I evaluated VQ-VIB models for $n \in [1, 2, 4]$. Recall that n is the combinatorial parameter that set how many codebook elements to concatenate into a single message.

OOD generalization results, for varying C and n, are included in Figure 5-5. As expected, increasing C decreased utility, as the task became substantially more challenging. Notably, for all C, the trend from Chapter 2 of lower distortion supporting greater utility continued to hold.

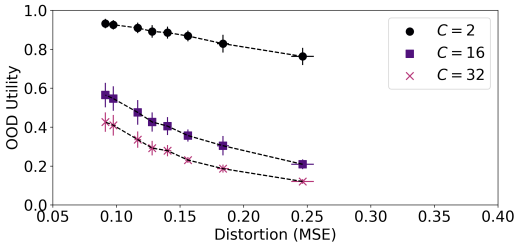
Examining results for $n > 1$, one sees that this trend held even as distortion continued to decrease further. Increasing n clearly decreased distortion (note how curves shifted to the left along the x axis as n increased) which in turn was associated with greater utility. These improvements are particularly notable in comparison to onehot or Proto. performance, plotted in Figure 5-6. For example, for $C = 32$, VQ-VIB$_{\mathcal{N}}$ achieves up to 60% accuracy for $n = 4$, whereas onehot and Proto. accuracy peaks at approximately 17% and 13%, respectively.

Overall, I found a strong trend between informativeness and OOD utility across architectures. This trend also explains, to a large degree, inter-architecture differences: VQ-VIB models tended to achieve lower distortion, which in turn led to greater utility.

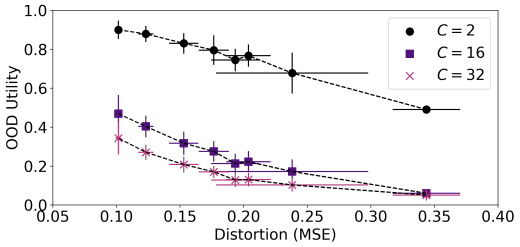


(a) VQ-VIB$_{\mathcal{N}}$ $n = 1$

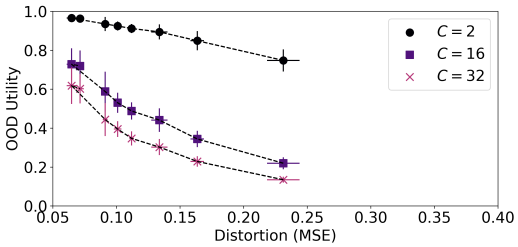
(b) VQ-VIB$_{\mathcal{C}}$ $n = 1$



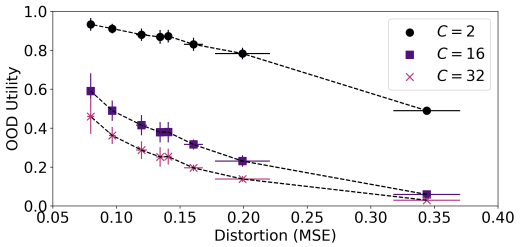
(c) VQ-VIB$_{\mathcal{N}}$ $n = 2$



(d) VQ-VIB$_{\mathcal{C}}$ $n = 2$

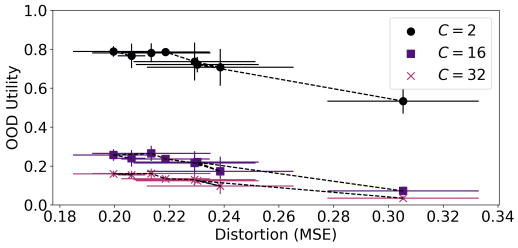


(e) VQ-VIB$_{\mathcal{N}}$ $n = 4$

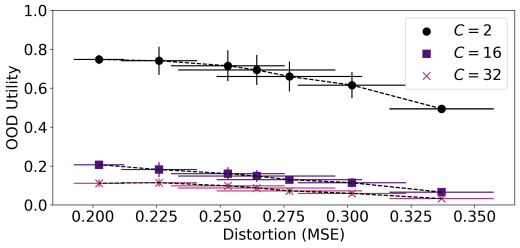


(f) VQ-VIB$_{\mathcal{C}}$ $n = 4$

Figure 5-5: OOD accuracy vs. distortion for VQ-VIB$_{\mathcal{N}}$ and VQ-VIB$_{\mathcal{C}}$ (different columns) for different combinatorial codebook factors (different rows). For all architectures, OOD accuracy as distortion decreased, and using greater n supported lower distortion. Note how increasing n shifted all curves to the left, corresponding to lower MSE.

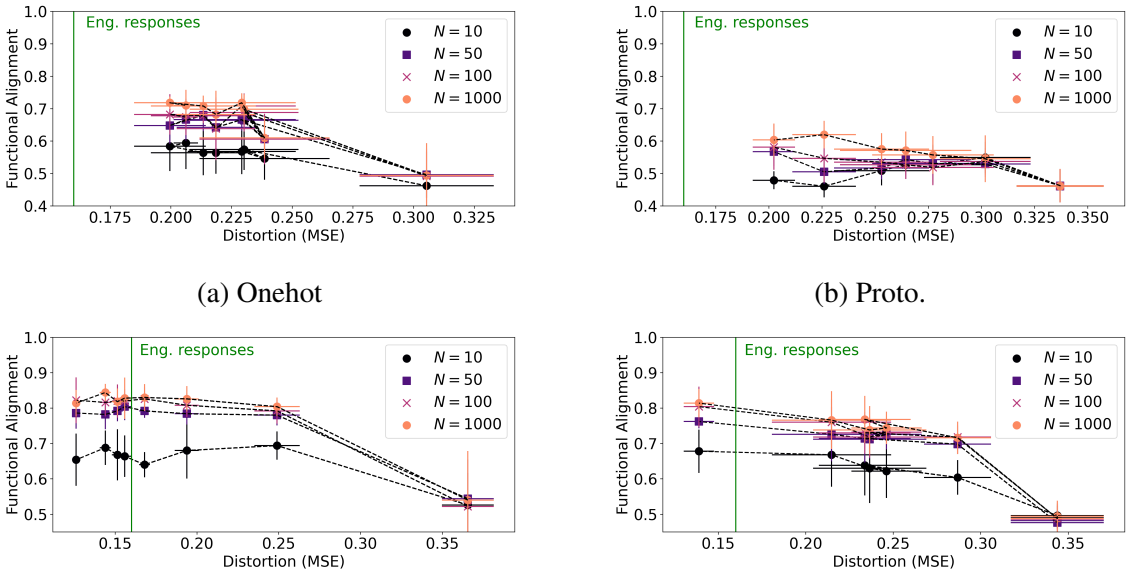


(a) Onehot



(b) Proto.

Figure 5-6: OOD accuracy vs. distortion for onehot and Proto. architectures, and varying C. As in Figure 5-5 for VQ-VIB models, decreasing distortion was associated with increased OOD utility.



(a) Onehot

(b) Proto.

(c) VQ-VIB$_{\mathcal{N}}$

(d) VQ-VIB$_{\mathcal{C}}$

Figure 5-7: Functional alignment for different architectures for varying amounts of alignment data, N. Increasing N improved alignment slightly, but performance typically plateaued after reaching English response informativeness.

5.6.2 Functional alignment analysis

In some of the ManyNames experiments, I trained a linear model to map from GloVe embeddings to EC vectors. In Chapter 2, I reported results based on fitting this model with $N = 100$ randomly-selected images and associated labels. In Figure 5-7, I present results for different N.

As expected, increasing N increases translation performance, but only up to a point. More importantly, regardless of N, all plots exhibit similar plateauing behavior: at high distortion (low informativeness), performance improved as distortion decreased, but only up to the estimated English informativeness level, at which point performance roughly plateaued. Lastly, VQ-VIB architectures tended to outperform the other architectures, both by achieving greater alignment for the same distortion and by reaching lower distortion values.

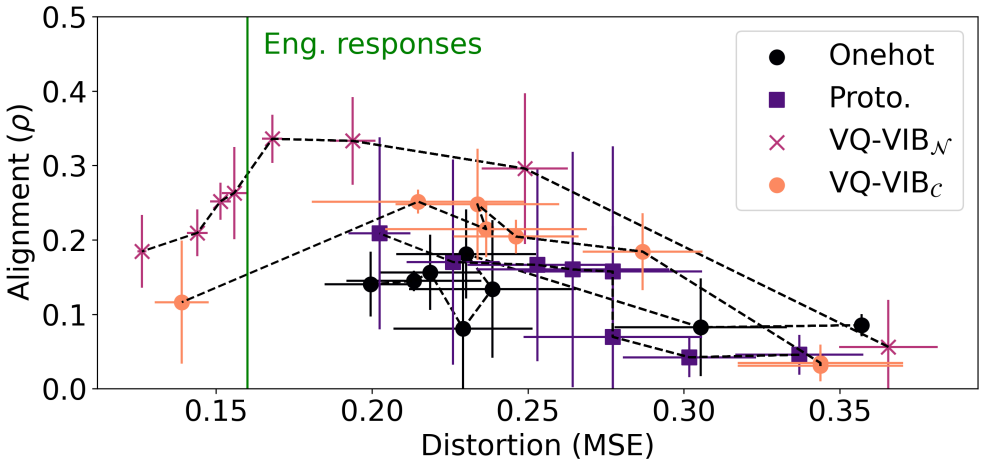


Figure 5-8: Relative representation alignment (ρ) between EC and GloVe embedding spaces. Results for VQ-VIB$_\mathcal{C}$ models are overlayed over data from Figure 2-10. VQ-VIB$_\mathcal{C}$ exhibits similar, although slightly worse, alignment trends to VQ-VIB$_\mathcal{N}$.

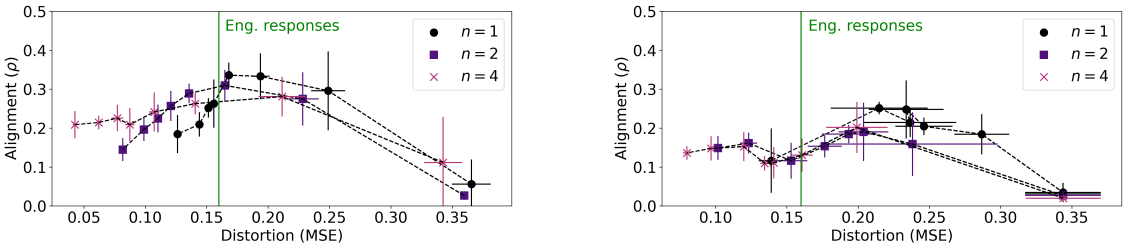
5.6.3 Relative representation alignment

Just as I conducted further analysis for functional alignment in the previous section, in this section I examined relative representation alignment trends for different neural architectures. In Chapter 2, I found that alignment peaked when EC distortion roughly matched English distortion, and that VQ-VIB$_\mathcal{N}$ models tended to achieve greater alignment than other models.

First, complementing Figure 2-10, I plotted relative representation alignment for all model architectures, including VQ-VIB$_\mathcal{C}$ in Figure 5-8. Notably, both VQ-VIB methods achieve greater alignment than other models, and both achieve greatest alignment near the estimated distortion from English responses.

Next, I examined alignment for both VQ-VIB models for varying n, the parameter specifying how to combine codebook elements into discrete communication. Results are plotted in Figure 5-9.

Two main trends are apparent in Figure 5-9. First, as identified in the OOD experiments for varying n, increasing n led to lower distortion (higher informativeness). Second, and most importantly, despite the increased informativeness, increasing n tended to result in *lower* relative representation alignment. This reinforces the importance of setting the



(a) VQ-VIB$_{\mathcal{N}}$ (b) VQ-VIB$_{\mathcal{C}}$

Figure 5-9: Relative representation alignment for VQ-VIB$_{\mathcal{N}}$ and VQ-VIB$_{\mathcal{C}}$, for varying combinatorial token settings (n). Increasing n decreased the distortion for both model types (note the changed x axis scale) but worsened alignment.

"right" informativeness to match the English data, and that simple architectures can, at least in this domain, learn the most human-like representation space.

Chapter 6

Appendix: Causal Probing

6.1 MINE Details

The Mutual Information Neural Estimator technique works by training a neural net to compute and maximize a lower bound on mutual information between two random variables [Belghazi et al., 2018]. I describe the intuition of the technique, as well as my implementation, in this section; I refer readers to the full paper for theoretical analysis of MINE.

The mutual information between two variables is defined via the KL Divergence between the joint distribution of the variables and the product of their marginals: $I(X, Y) = D_{KL}(P(XY)||P(X)P(Y))$. For notational simplicity, I describe the joint distribution as P and the product of the marginals as Q. Let me further state that P and Q define outputs that are jointly in R^D.

A lower-bound for the KL divergence is as follows, setting F as any class of functions that maps from R^D to R:

$$D_{KL}(P||Q) \geq \text{sup}_{T \in F}\, \mathbb{E}_P[T] - (\,\mathbb{E}_Q[e^T]) \tag{6.1}$$

In other words, one can calculate a lower bound for the mutual information by finding a function, T, that maximizes the difference between the two terms in Equation 6.1. Belghazi et al. [2018] do so with functions parametrized as a neural net that maps from

the concatenation of two inputs (one for each random variable) to a single-valued output. Training the neural net is conducted to maximize the value described by Equation 6.1.

In my experiments, I create neural networks with separate, linear layers of size 64 for each input. The embeddings from those two layers are concatenated, passed through two 1024-dimensional layers with ReLU activations, and then passed through a linear layer with a single output. I thus map from the two inputs to a single, real-valued output.

Training was performed using batch size 32 over 50 epochs, at which point the mutual information estimates appeared to have converged.

6.2 Test Suite Creation

Here, I specify the details of the test suites used to evaluate models for reproducibility.

The Mask model Coordination test suite comprised sentences like "The man saw the girl and the dog [MASK] tall." More generally, sentences followed the following template: "The NN1 V the NN2 and the NN3 [MASK] ADJ." I created all sentences by iterating through the combinations of the words described in Table 6.1. This generated 243 sentences, and each sentence was associated with 2 parses: one described as a conjunction of sentences (e.g., "(The man saw the girl) and (the dog [MASK] tall.)") and one as a single sentence with a conjunction of noun phrases (e.g., "The man saw (the girl and the dog) [MASK] tall.").

Category	Words
NN1	man, woman, child
NN2	boy, building, cat
NN3	dog, girl, truck
V	saw, feared, heard
ADJ	tall, falling, orange

Table 6.1: Words used for sentence generation in the Mask Coordination test suite.

The mask model NP/Z test suite comprised sentences like, "When the dog scratched the vet [MASK] ran." More generally, sentences followed the following template: "When the NN1 V1 the NN2 [MASK] V2." Each sentence was associated with two parses, favoring either adverbs (e.g., "When the dog scratched the vet quickly ran" or nouns, "When the

dog scratch the vet she ran"). I used the word tuples described in Table 6.2 to generate 150 sentences.

NN1	V1	NN2	V2
(dog/child)	(scratched/bit)	(vet/girl/boy)	(ran/screamed/smiled)
author	wrote	book	grew
(doctor/professor)	lectured	student	listened
(girls/boys)	raced	(kids/children)	(watched/cheered)
(people/spectators)	watched	(show/movie)	(stopped/paused)
(lawyers/judges)	(studied/considered)	case	(languished/proceeded)
(people/viewers)	(notice/spot)	actor	(departs/stays)
(band/conventions)	left	(hotel/stalls)	closed

Table 6.2: Words used for sentence generation in the Mask NP/Z test suite.

The QA model Coordination test suite comprised prompts like "Who was tall? The happy stranger saw the angry men and the angry women were tall." More generally, the prompts followed the following template: "Who was ADJ1? The ADJ2 NN1 V the ADJ3 NN3 and the ADJ4 NN4 were ADJ1." I created 256 prompts by iterating through combinations of the words in Table 6.3. "None" adjectives were excluded from the text.

Category	Words
ADJ1	tall, short
ADJ2	happy, None
ADJ3	angry, None
ADJ4	angry, None
NN1	stranger, child
NN2	men, women
NN3	women, men
V	saw, believed

Table 6.3: Words used for sentence generation in the QA Coordination test suite.

The QA model NP/VP suite comprised prompts like "Who had the telescope? The girl saw the boy with the telescope." The prompts followed the following template: "Who had the NN1? The ADJ1 NN2 ADV V the ADJ2 NN3 with the ADJ3 NN4." In this suite, the choice of V and NN4 was tightly coupled - one may see with a telescope but not see with a stick, for example. Table 6.4 details the combinations of words used to fill out the template, including V-NN4 pairs. Overall, I generated 256 prompts.

Category	Words
V - NN4	(saw, telescope), (poked, stick)
ADJ1	tall, None
ADJ2	short, None
ADJ3	special, None
NN1	man, woman
NN2	boy, girl

Table 6.4: Words used for sentence generation in the QA NP/VP test suite.

The QA model RC suite comprised prompts like "Who was desperate? The women and the men who were desperate bribed the politician." The prompts followed the following template: "Who was ADJ1? The ADJ2 NN1 and the ADJ3 NN2 who were ADJ1 V the NN3." I generated 192 example prompts by iterating over combinations of the words listed in Table 6.5, excluding sentences in which NN1 and NN2 or ADJ2 and ADJ3 would have been the same.

Category	Words
ADJ1	corrupt, desperate
ADJ2	tall, smart, rich
ADJ3	tall, smart, rich
NN1	men, women
NN2	men, women
NN3	judge, politician

Table 6.5: Words used for sentence generation in the QA RC test suite.

The Intervention suite for the QA model comprised prompts like "What was green? The human saw the keys by the cabinet which were green." More generally, prompts were created via the following template: "What was ADJ1? The NN1 V the NN2 by the NN3 which was/were ADJ1." By changing the plurality of NN2 or NN3 and replacing "was" with "were," the correct answer should change. Overall, I generated 288 sentences by iterating over all combinations of the words listed in Table 6.6, such that exactly one of NN1 and NN2 was plural at a time.

Category	Words
ADJ1	green, large, dirty
NN1	human, stranger, child
NN2	key, keys, gadget, gadgets
NN3	cabinet, cabinets, vase, vases

Table 6.6: Words used for sentence generation in the QA intervention experiments.

6.3 Hyperparameter Selection

In the intervention experiments in Section 3.4.3, I performed interventions at layer 4, based on results of a validation study shown below. I reported the results for probes with different dropout rates and for varying counterfactual losses, but I had to choose the layer of the QA model at which to perform interventions.

Therefore, I created a validation suite based on the Intervention template, using new nouns, verbs, and adjectives. For dropout rates from 0.0 to 0.3, ranging over counterfactual losses, and layers from 1 to 7, I computed the QA model's F1 and Exact Match scores on the validation suite. These results are included in Table 6.7, and strongly suggest that performance, for all probes, was most increased via interventions at layer 4.

6.4 Varying Dropout Rates

In the main paper, I reported only some of the results for distance- and depth-based probe interventions. Here, I first show, in more detail, how increasing the dropout rate grows the causal effect with the QA attachment suite and distance probes of varying α. Next, I include the mean causal effect plots for Mask and QA models using both types of probes on the 5 total suites.

First, I plotted an example of how increasing the dropout rate grew the causal effect in the QA attachment suite in Figure 6-1. I found that positive dropout values consistently outperformed probes with no dropout. Furthermore, for α ranging from 0.1 to 0.4, increasing the dropout rate seemed to increase the effect size. Considering only interventions at layer 2, for example, normal probes shifted model predictions by at most 2% for different parses; for probes with dropout 0.5, probabilities shifted by roughly 20%.

α/Loss	Layer	0.05	0.1	0.2	0.3
	1	71.9/59.4	72.7/60.9	73.4/60.9	73.4/60.9
	2	69.5/56.3	71.9/60.9	71.9/60.9	71.9/59.4
	3	71.1/60.9	71.1/59.4	71.9/59.4	71.9/59.4
Dist. 0.0	**4**	**71.9/62.5**	**72.6/60.4**	**71.9/59.4**	**73.4/60.9**
	5	68.8/57.8	68.8/56.3	72.7/60.9	73.4/62.5
	6	68.8/57.8	69.5/59.4	71.9/60.9	72.7/62.5
	7	70.3/60.9	70.3/60.9	72.6/62.5	72.6/62.5
	1	69.5/56.3	71.1/59.4	71.9/59.4	71.9/59.4
	2	68.8/60.9	70.3/60.9	69.3/59.4	71.1/59.4
	3	67.2/56.4	69.5/60.9	72.7/60.9	73.4/62.5
Dist. 0.1	**4**	**75.8/64.1**	**72.7/60.9**	**72.7/60.9**	**72.7/60.9**
	5	68.8/56.3	70.3/59.4	71.9/57.8	71.1/56.3
	6	75.0/59.4	72.7/60.9	73.4/62.5	73.4/62.5
	7	72.7/60.9	72.7/62.5	72.7/62.5	72.7/60.9
	1	69.5/54.7	70.3/56.3	72.7/59.4	73.4/60.9
	2	73.4/60.9	74.2/59.4	74.2/62.5	74.2/62.5
	3	70.3/59.4	69.5/56.3	71.1/57.8	71.9/57.8
Dist. 0.2	**4**	**74.2/65.6**	**75.0/65.6**	**75.8/65.6**	**75.0/64.1**
	5	71.1/62.5	71.9/64.1	71.1/62.5	71.9/60.9
	6	73.4/62.5	71.8/59.4	74.2/62.5	74.2/62.5
	7	71.9/59.4	73.4/62.5	72.7/62.5	72.7/60.9
	1	67.2/54.7	70.3/59.4	73.4/62.5	72.7/60.9
	2	68.8/60.9	71.1/60.9	72.7/62.5	71.9/60.9
	3	61.7/53.1	64.8/56.3	71.9/64.1	72.3/65.6
Dist. 0.3	**4**	**67.2/59.4**	**71.9/64.1**	**75.0/65.6**	**75.8/65.6**
	5	62.5/56.3	68.8/59.4	70.3/62.5	70.3/62.5
	6	71.1/62.5	711/64.1	70.3/60.9	71.1/60.9
	7	75.0/64.1	72.7/62.5	71.9/62.5	73.4/62.5

Table 6.7: Validation Coord. suite results (F1/Exact Match) using distance probes. For each probe type, I iterated over intervention layer and counterfactual loss value. The small validation suite was useful for rapid identification of good hyperparameter settings. All probes had the best performance at layer 4 (in bold).

QA Model Causal Effect on Attachment Suites Using Dropout Distance Probes

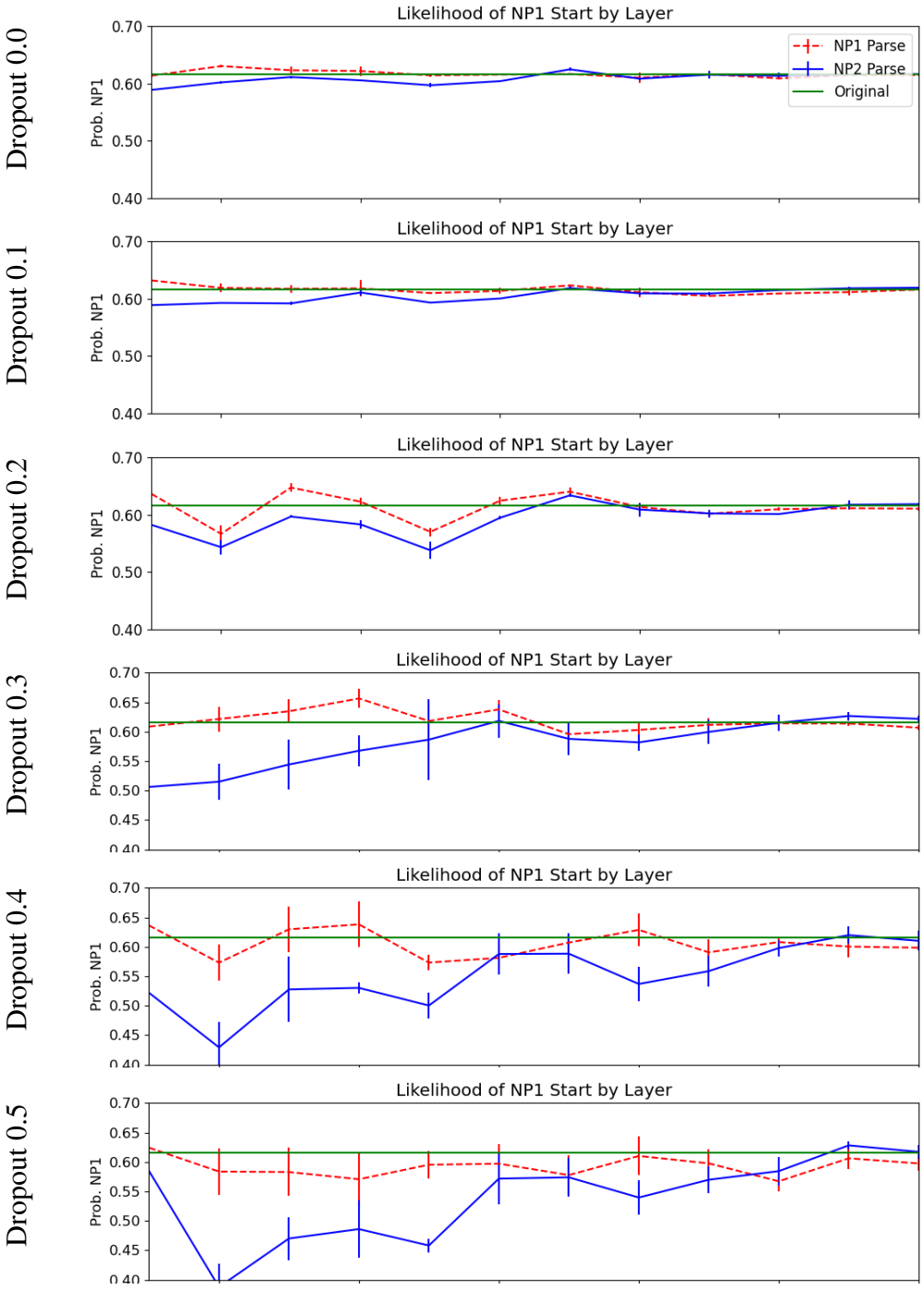


Figure 6-1: Dropout distance probes with dropout rates from 0.0 to 0.5 showed how, to a point, increasing the dropout rate increased the effect size for QA models on the Coord. suite.

Finally, I included results for all dropout rates and counterfactual losses in Figures 6-2 and 6-3.

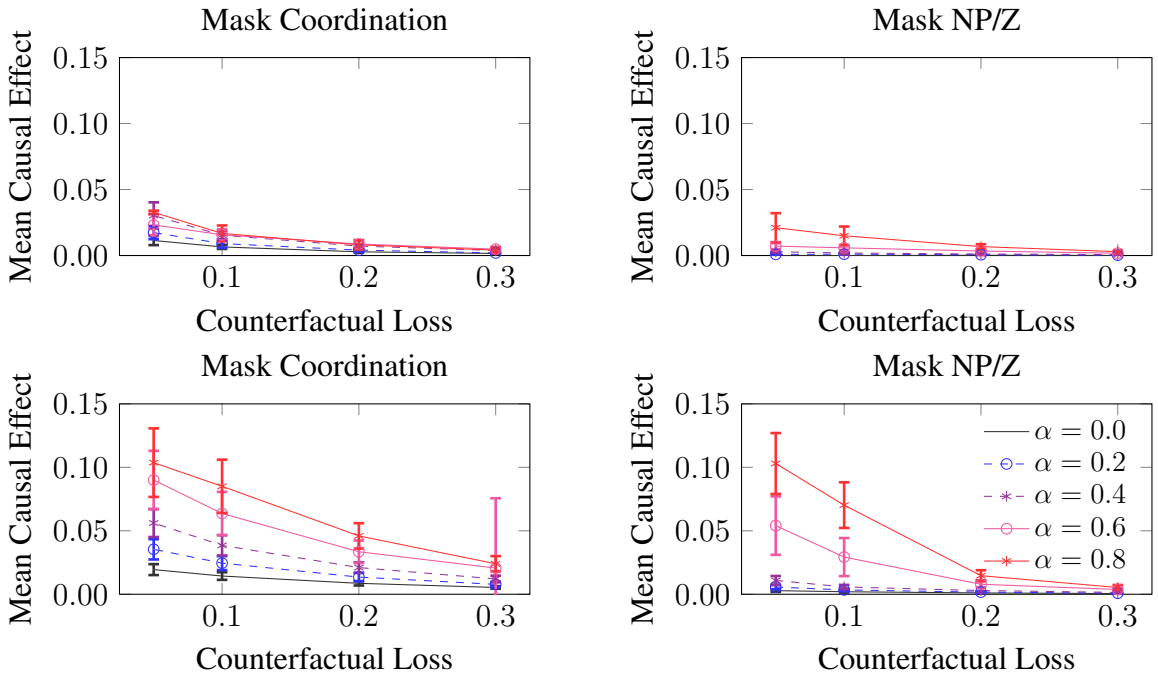


Figure 6-2: Mask mean causal effects using depth- (top) or distance-based (bottom) probes. Depth probes revealed smaller effects than distance-based probes, but a similar pattern of benefiting from lower counterfactual loss and higher dropout.

6.5 Probe Performance Metrics

In the main paper, I demonstrated the benefits of using dropout probes for creating counterfactual embeddings. One could hypothesize that the dropout enables better counterfactuals because the probes are prevented from overfitting to the training data. I found that that was not the case.

In Figure 6-4, I plotted probe performance metrics for the distance- and depth-based probes. For the distance probe, I reported the spearman correlation coefficient between predicted and actual pairwise distances between words in a sentence's parse tree. For the depth probe, I reported the accuracy of the probe in predicting the word at the root of the syntax tree. Both metrics were used in prior probing literature [Hewitt and Manning, 2019].

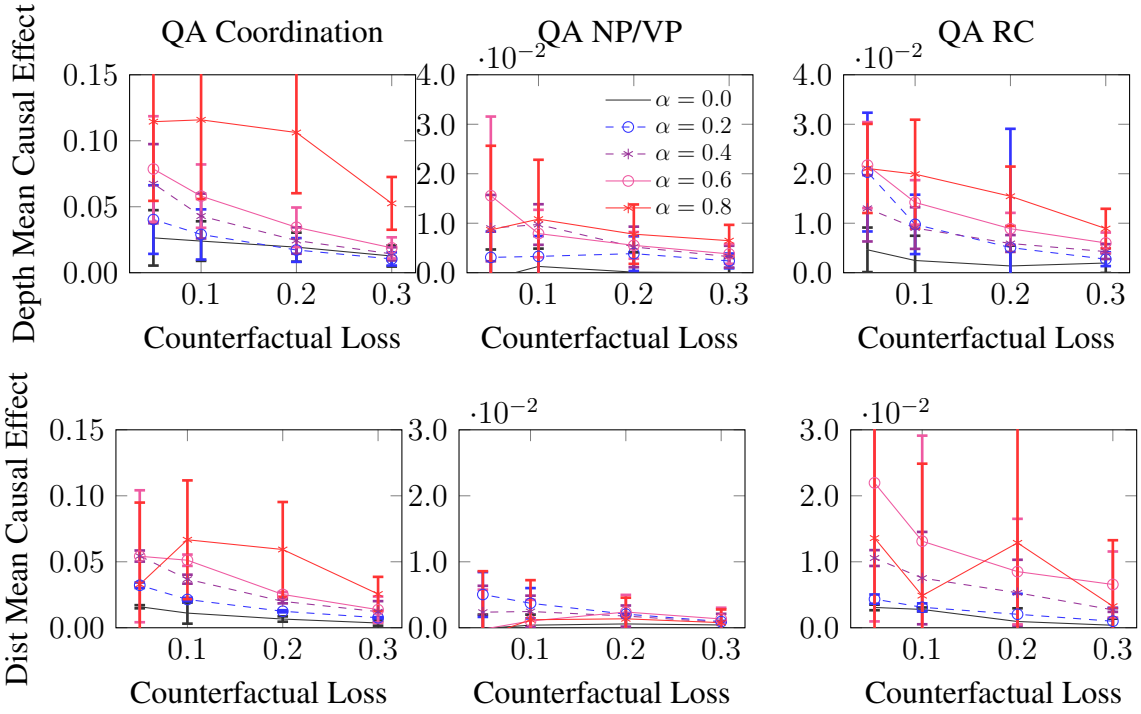
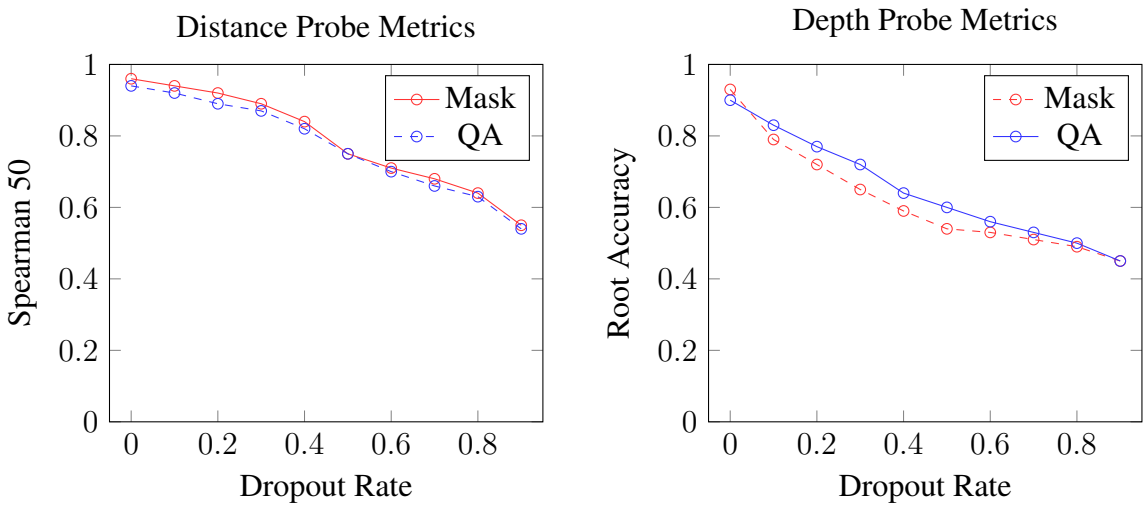


Figure 6-3: QA mean causal effects using depth-based (top) or distance-based (bottom) probes.

I found that, while using non-linear probes boosted probe performance compared to linear probes, adding dropout actually worsened probe performance. This suggests that the benefits from dropout in counterfactual generation arose from a phenomenon other than higher-performing probes.



(a) Distance probe metrics

(b) Depth probe metrics

Figure 6-4: Metrics for the distance (left) and depth (right) probes showed that introducing dropout worsened probe performance as measured on the probe prediction tasks. Means over 5 trials plotted. All standard deviations less than 0.01.