Disentangling Categorization in Multi-agent Emergent Communication

Washington Garcia
University of Florida
Gainesville, FL, USA
w.garcia@ufl.edu

Hamilton Scott Clouse
ACT3, Air Force Research Laboratory
Wright-Patterson AFB, OH, USA
hamilton.clouse.1@afrl.af.mil

Kevin R.B. Butler
University of Florida
Gainesville, FL, USA
butler@ufl.edu

Abstract

The emergence of language between artificial agents is a recent focus of computational linguistics, as it offers a synthetic substrate for reasoning about human language evolution. From the perspective of cognitive science, sophisticated categorization in humans is thought to enable reasoning about novel observations, and thus compose old information to describe new phenomena. Unfortunately, the literature to date has not managed to isolate the effect of *categorization power* in artificial agents on their inter-communication ability, particularly on novel, unseen objects. In this work, we propose the use of disentangled representations from representation learning to quantify the categorization power of agents, enabling a differential analysis between combinations of *heterogeneous systems*, e.g., pairs of agents which learn to communicate despite mismatched concept realization. Through this approach, we observe that agent heterogeneity can cut signaling accuracy by up to 40%, despite encouraging compositionality in the artificial language. We conclude that the reasoning process of agents plays a key role in their communication, with unexpected benefits arising from their mixing, such as better language compositionality.

1 Introduction

A recent interest in the design of multi-agent systems is the unsupervised emergence of complex behaviors (Havrylov and Titov, 2017). Rather than completely supervising agents to perform a mapping betweem inputs and outputs, it is observed that with enough capacity and bandwidth, agents eventually learn to produce emergent properties for their own benefit, e.g., the creation of artificial language to solve a task more efficiently (Resnick et al., 2020; Lee et al., 2018; Chaabouni et al., 2019). With carefully designed constraints, the agents may eventually produce a proto-language that mimics the complex properties of human languages, such as compositionality and basic interpretability (Kottur et al., 2017).

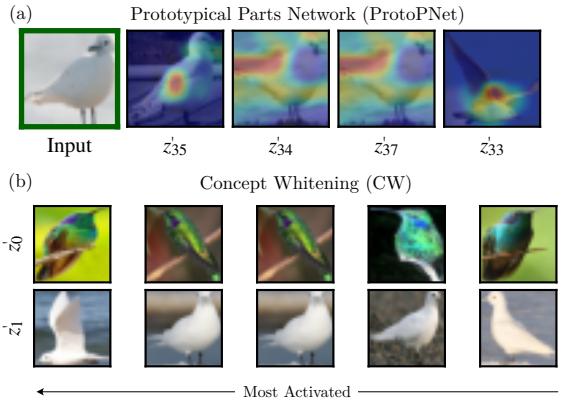


Figure 1: Approximating categorization ability in agents through disentangled representations (denoted $\mathbf{z}'$), generated by either (a) prototypical parts network (ProtoPNet) (Chen et al., 2019), where each dimension is the input's activation of a prototypical part previously seen by the agent, or (b) concept whitening (CW) (Chen et al., 2020), where each dimension corresponds to a previously-observed data category.

Despite recent progress, previous works have not investigated inter-agent communication in the context of abstract categorization, an ability that allows humans to represent new observations by recalling previous experiential knowledge of abstract concepts or classes (Hofstadter, 2007; Gentner, 2010). Abstract categorization in the human brain can be attributed to a combination of recognition and recall of episodic memory, although identifying the exact mechanism remains an open problem (Tulving, 2002). Due to the power and flexibility of categorization, theories of human intelligence often place *categorical reasoning* at the center of human cognition, e.g., the notion of analogical symbolic categorization by Gentner (2010), or categorization through situated templates, i.e., frames (and corresponding frame-systems) by Minsky (1988). In the study of artificial multi-agent systems, we propose

Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 4523 - 4540

to operationalize the notion of categorization, and subsequent ability to reason over new observations, through the use of *disentangled neural networks* as alternate image encoders in agents. Disentangled neural networks are a suitable choice since they can leverage a fixed number (k) of pre-defined or self-learned concepts to build representations of the observable environment. As illustrated in Figure 1, disentangled agents extract meaning from an observation by leveraging their past experiential knowledge, which is analogous to categorical reasoning in theories of human intelligence (Minsky, 1988; Gentner, 2010).

Due to the subjectivity of experiential knowledge, a natural consequence of the categorical reasoning theory is that two human minds may be heterogeneous, i.e., they do not share the same internal categorization of a situation or object. Natural language serves to mediate these cognitive differences, thus enabling the communication of ideas (Chandler, 2007). Under our synthetic analogue of communicating multi-agent systems, *how well can heterogeneous agents mediate their differences in categorization ability through the use of emergent language?* In this paper, we offer a differential analysis of agents leveraging heterogeneous techniques for abstract categorization, taking full advantage of disentangled representations as a computational analogue for categorical reasoning. Contrary to the scenario of different internal representations (Chaabouni et al., 2020), our agents can possess either different or identical means of categorical abstraction, which influences the course of their language evolution, e.g., improving compositionality or signaling success of the final protolanguage. Our analysis offers a substrate to answer **four main research questions** about heterogeneous multi-agent communication, revealing subsequent findings as a result:

Q1. What is gained or lost with agent heterogeneity? A1: We find that *heterogeneity is at odds with signaling performance*. Despite this, heterogeneous systems exhibit *better potential survival* through higher compositionality.

Q2. Does categorization power and heterogeneity affect signaling ability on novel objects? A2: Simpler disentangled agents with poor performance in supervised tasks can *outperform all other agents on novel objects* by up to 9% on signaling accuracy. In some cases, mismatched agents manage to improve compositionality score.

Q3. Can disentangled representations directly inform agent utterances? A3: Through our proposed Latent Self-attention (LSA) module, incorporating the disentangled representation into agent utterances can encourage input-message alignment and improve signaling accuracy by up to 27%.

Q4. Does increased categorization ability imply better communication? A4: Agents with complex concept realization are potentially poor receivers, despite a successful upstream classification task. Signaling can be performed up to 33% better with "simpler", i.e., smaller k, agents.

To encourage reproducibility, we share the code for our experiments online.[1]

2 Background

2.1 Multi-agent Referential Games

We model the commonly-used Lewis signaling game from cognitive science (Lewis, 1969; Skyrms and Press, 2010). A sender agent S is shown a target object, and must describe that object to a receiver agent R using a combination of discrete tokens from a fixed vocabulary $\mathcal{V}$. R then signals the object out of a set of candidate objects (called the *candidate set*). Our setup is based on that of Lazaridou et al. (2018), who investigated emergent language with the same game. The sender S and receiver R each use their own encoder (e.g., image CNNs) f^S and f^R to encode any pre-linguistic object $\mathbf{o}$ into an initial dense representation $\mathbf{z}_0$. The encoder is parameterized by agent-specific weights $\theta_f^{\cdot}$, e.g., $f^S(\mathbf{o}, \theta_f^S) = \mathbf{z}_0$, where S denotes the sender. Since the receiver processes the candidate set, which consists of multiple pre-linguistic objects, the receiver's dense representations form a matrix denoted Z. Given $\mathcal{V}$ and $\mathbf{z}_0$, the message is obtained by sampling tokens at each time step t from a recurrent policy defined as $g^S(\mathbf{z}_{t-1}, \theta_g^S) = \mathbf{m}_t$. The decoding process ends as soon as an end-of-sequence token or maximum length L is reached. In practice, the recurrent decoder is implemented as a recurrent neural network (RNN). The receiver's only input from the sender

[1] https://github.com/FICS/disentangling_categorization

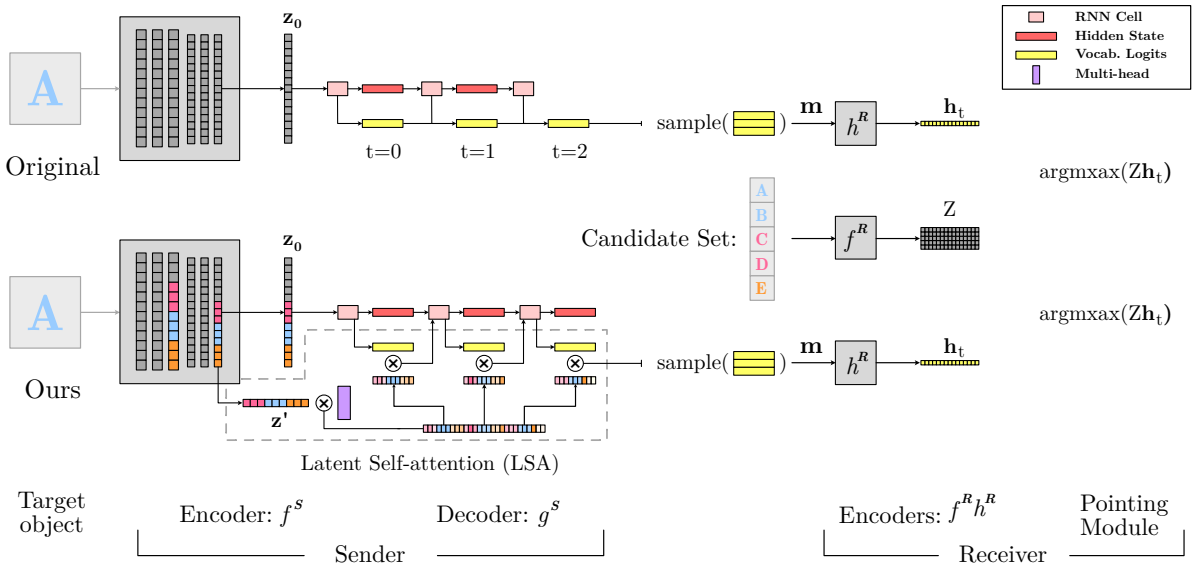


Figure 2: Whole-system perspective of our approach. (Top - Original) Traditional Sender-Receiver architecture for playing the Lewis Signaling game, with entangled deep networks as encoders f^S and f^R. (Bottom - Ours) Proposed approach which leverages disentangled networks to instantiate f^S or f^R (denoted by latent dimension colors matching candidate set colors), and our new Latent Self-Attention (LSA) module for senders, which enforces alignment between utterances (yellow rectangles) and the disentangled structure over time steps.

is message $\mathbf{m}$, which is encoded using encoder h^R at each time step as $h^R(\mathbf{m}_t, \theta_h^R) = \mathbf{h}_t$. Similar to Lazaridou et al. (2018), the receiver predicts the position of the described object within the candidate set as $\text{argmax}(Z\mathbf{h}_t)$. The agents are successful during a round of the signaling game if the receiver correctly predicts the index of the target object at the final time-step. The typical whole-system perspective of this game is shown in the top method (Original) of Figure 2.

2.2 Learning

We must jointly optimize sender and receiver policy weights $\Theta := \{\theta_g^S, \theta_h^R\}$, such that they minimize a game cost function $\mathcal{L}$, i.e., $\min_\Theta \mathcal{L}$. However, no weights are shared between sender and receiver during the game, since each agent has a uniquely instantiated encoder, and the sender can only transmit discrete signals to the receiver during the game. As a result the gradient of the game cost function is not defined, so it is common to rely on gradient estimation algorithms such as REINFORCE (Schulman et al., 2015) or continuous relaxations of the discrete channel, such as the Gumbel-Softmax estimator (Havrylov and Titov, 2017). We use the straight-through Gumbel-Softmax estimator due to its functional flexibility and performance improvement compared to REINFORCE at test time.

2.3 Disentangled Representations

A common technique is to employ intermediate feature representations of CNNs as the encoder function $f^{\cdot}$. The purpose of the encoder function is to approximate a mapping function from raw image space to a dense representation $\mathbf{z} \in \mathbb{R}^d$. In general, this representation is a point in lower-dimensional latent space that adequately describes useful conceptual priors of the data. From the perspective of model interpretability, it is expected that points closer together in latent space correspond to image samples which are conceptually similar, and likewise, dimensions in the latent space correspond to human understandable concepts (Rudin et al., 2022). However, it was shown that traditional CNN architectures learn latent representations which may not achieve concept separation (Chen et al., 2020), and in fact can activate dimensions for concepts that are completely unrelated (Zhou et al., 2015, 2018). Thus several methods have been proposed in order to *disentangle* the latent representation of deep networks into k categories or concepts. In this work, we propose to leverage two such methods, shown in Figure 1, in order to analyze the effect of categorization power in multi-agent communication: (a) Prototypical Parts Network (ProtoPNet) by Chen et al. (2019), which recalls prototypical parts of previously-observed images to describe the current one, and (b) Con-

cept Whitening (CW) by Chen et al. (2020), which allocates an observed data class to each of the k dimensions in $\mathbf{z}'$.

3 Methodology

In order to answer the research questions in Section 1, we implement agents which can play a Lewis Signaling game (described in Section 2.1) using disentangled neural networks, e.g., Prototypical Parts Network (ProtoPNet) by Chen et al. (2019) and Concept Whitening (CW) by Chen et al. (2020). ProtoPNet is trained through unsupervised disentanglement, where given a pre-defined number of concepts per class (totaling k over all classes), it extracts prototypical image patches representing each concept. Concept Whitening instead aligns orthogonal directions in the encoded input's latent space with k pre-defined (supervised) concepts through training. In practice, CW can only align a subset of the pre-existing latent space dimensions, leaving the rest of the dimensions to handle residual information. To avoid ambiguity, we denote a sample from this aligned subspace as $\mathbf{z}' \in \mathbb{R}^k$. Since ProtoPNet learns a new latent space, it can be said to align the entire latent space to extracted concepts, thus $\mathbf{z}' = \mathbf{z}$. In either case, we use the latent vector $\mathbf{z}$ as the initial input to the sender agent's decoder, g^S. The whole-system perspective of this approach is shown in Figure 2, where the bottom case (Ours) corresponds to the use of disentangled networks, compared to a traditional approach above (Original). We can manipulate which agents are paired (e.g., ProtoPNet talking to CW) in order to simulate scenarios where two agents have different categorical reasoning. Likewise, we can model categorization *power* by increasing or decreasing the number of realized concepts k in disentangled agents. This enables the study of incongruous agents to answer Q1, Q2, and Q4.

3.1 Latent Self-attention (LSA)

Unfortunately, in order to synthesize the discrete message and eventually communicate with the receiver, the sender's disentangled representation must recurrently pass through the decoder g^S, which defines a different (entangled) latent space. This could mean the grounding to the original disentangled space is lost, since traditional RNN decoding is not aware of the encoder's concepts, thus we propose a module named Latent Self-Attention (LSA), inspired by the design of multi-head attention in transformer networks (Vaswani et al., 2017). We treat the aligned latent dimensions from $\mathbf{z}'$ as units that align one-to-one with the vocabulary logits of each time step, which can be weighted by a combination of aligned concepts. This mechanism is illustrated in the bottom method (Ours) of Figure 2, distinguished by the dashed rectangle. In order to satisfy the one-to-one alignment, we assume that (1) for the aligned representation $\mathbf{z}' \in \mathbb{R}^k$, $|\mathcal{V}| = k$, and (2) agents communicate with fixed message length L (due to discarding end-of-sequence token).

More formally, consider the recurrent output $\mathbf{z}_t \in \mathbb{R}^d$ from the sender decoder at time t. At each time step, the recurrent output is projected to dimension $|\mathcal{V}| = k$ using a fully-connected network, of which the output is treated as logits over vocabulary space (yellow rectangles), and is simply denoted $\mathbf{v}_t$. During the training process, the sender agent learns an L-head matrix $M \in \mathbb{R}^{k \times (L \cdot k)}$ (purple rectangle). which is softmaxed row-wise to obtain a weighting function over k concepts for each time step. The weighting of concepts is applied as $\mathbf{z}'^T M$ and reshaped to form matrix $M' \in \mathbb{R}^{L \times k}$. During decoding, the t-th row of M' is element-wise multiplied with $\mathbf{v}_t$. Through this mechanism, we can experimentally validate Q3 from Section 1. In our experiments, the receiver is implemented the same as previous works (shown as Original in Figure 2) (Lazaridou et al., 2017). To experimentally isolate the effect of the LSA module, we leave the receiver as-is, treating the disentangled networks as a drop-in selection for f^R. Incorporating the aligned vector $\mathbf{z}'$ into the pointing module is left for future work.

3.2 Metrics

We adopt standard metrics from the literature to quantify our study of Q1-Q4, and propose one new variant named disentangled similarity.

Receiver Accuracy (Recv. Acc.). A standard metric in the study of referential games is the communication success rate, i.e., the receiver's effective signaling accuracy on objects from a held-out test set (Skyrms and Press, 2010; Lazaridou et al., 2017). The metric is defined as Recv. Acc. $= \frac{\#\text{correct target objects}}{\#\text{total target objects shown}}$.

Compositionality (Top.Sim.) A common concern in the study of language emergence is compositionality of the proto-language, which can be

described informally as the tendency to re-use vocabulary to describe similar objects (Brighton and Kirby, 2006; Chaabouni et al., 2020). Brighton and Kirby (2006) operationalized the measure of language compositionality with topographic similarity (Top.Sim.), the Spearman correlation between pairwise input distances and their corresponding pairwise message distances. We compute pairwise cosine distances between latent vectors $\mathbf{z}$ (inputs), and use edit distance to compute pairwise distance between messages. The reported Top.Sim. value is the Spearman correlation of distances (and where available, the associated p-value is reported).

Disentangled Similarity (Dis.Sim.) Since CW only aligns a subset of the latent space dimensions, it is possible that Top.Sim. will describe correlation to unaligned dimensions. Thus we complement the calculation of Top.Sim. with a new metric, disentangled similarity (Dis.Sim.), which only uses the aligned dimensions to calculate the pairwise input distances. In practice, this is implemented by calculating pairwise cosine distances over $\mathbf{z}'$ instead of $\mathbf{z}$. Consequently, the reported values for ProtoPNet will be equal to Top.Sim., since in that case $\mathbf{z} = \mathbf{z}'$.

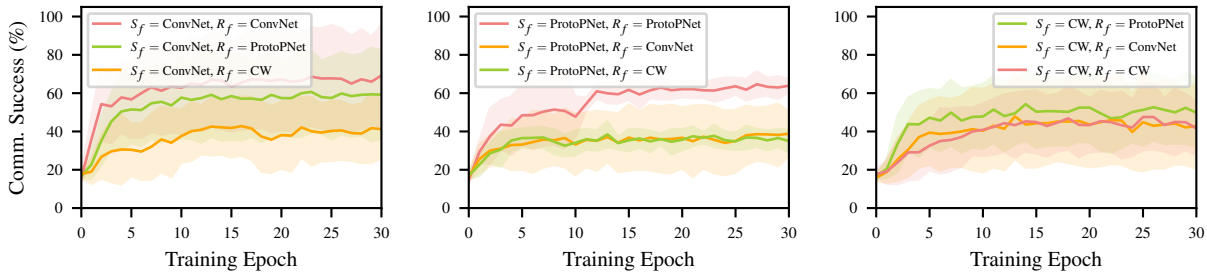
4 Experiments

Preliminaries. To address Q1-Q4, we investigate mixed/same-encoder variants of multi-agent systems. Pairs of agents play the Lewis signaling game over five random seeds. Each agent in a two-player game is instantiated with distinct image encoder checkpoints, thus requiring ten image encoder checkpoints total for each variant (with each checkpoint using a different seed). In all reported results of the main text, we compare against fine-tuned ResNet-50 encoders (denoted ConvNet) as baselines (He et al., 2016), and subsequently use them as the base models for training ProtoPNet and CW variants. We offer supplemental results for VGG-16 (Simonyan and Zisserman, 2015) and DenseNet-161 (Huang et al., 2017) variants in Appendix A.4.1. With ResNet-50 as the base architecture, each disentangled variant has a similar parameter count on the order of 24M (reported in Appendix Table 12), thus any performance difference due to parameter count is negligible. We source prelinguistic objects from two benchmark image data sources: 10-class subsets of Caltech-UCSD CUB-200 (Wah et al., 2011) (denoted CUB10) and the entire few-shot mini-ImageNet dataset (Vinyals et al., 2016). For mini-ImageNet experiments, we train agents using the 64-class meta-training split, and report communication success rate (Recv. Acc.) on the meta-test set. For CUB10, we uniform randomly sample ten classes for each random seed, ensuring sender and receiver encoders are trained using the same subset. Our communicating agents are derived from those implemented in the publicly-available EGG framework (Kharitonov et al., 2019). We only employ our proposed LSA module for Q3 results, as it is not salient to Q1, Q2, and Q4. Details of training and hyper-parameters are given in Appendices A.1 and A.2, respectively.

Heterogeneous multi-agent systems (Q1): To examine the effect of heterogeneous multi-agent systems, we train nine permutations of mixed/same-encoder systems over five random seeds on CUB10 data (45 permutations total), setting $k = 100$ for ProtoPNet (default value from Chen et al. (2019)) and $k = 10$ for CW (using CUB10 classes as concept classes). In Figure 3, we initially compare the communication success rate (y-axis) over training epochs (x-axis), with each sub-figure corresponding to a fixed sender encoder architecture. The results exhibit a tendency for same-encoder systems to outperform their heterogeneous counterparts. This trend is evidenced by ConvNet-ConvNet systems in Figure 3a (red line) and ProtoPNet-ProtoPNet systems in Figure 3b (red line). CW models generally under-performed as receivers in Figure 3a-b, scoring 47.978 ± 14.325 and 42.001 ± 5.203 receiver accuracy, respectively. In terms of disentangled categorization power, ProtoPNet is the strongest with $k = 100$ and up to 93% accuracy on the CUB10 classification task (compared to 78% of CW).[2]

In Table 1 we report Top. Sim. and Dis. Sim. measures for systems from Figure 3 at the best epoch on the test set (according to Recv. Acc.). Although CW models generally under-performed in Figure 3, they score the best compositionality (0.405) paired with ConvNet receivers. This is followed by ConvNet senders (top three rows), with 0.316–0.363 Top.Sim. score. CW-ConvNet scored the highest Dis. Sim. score despite having a Recv. Acc. of 48.939 ± 22.226, aligning with Chaabouni et al. (2020), who found that compositionality does not necessarily emerge from generalization pressure. Based on Figure 3 and Table 3, we conclude:

[2]We provide encoder accuracy scores from the upstream classification task in Appendix Table 12.



(a) (b) (c)

Figure 3: Comparison of communication success rate on CUB10 testing data for permutations of a) ConvNet senders, b) ProtoPNet senders, and c) CW senders. Shaded regions denote standard deviation over five random seeds (each swapping in a new CUB10 test split, sender encoder checkpoint, and receiver encoder checkpoint), with $L = 10$.

S_f	R_f	Top. Sim.	Dis. Sim.
ConvNet	ConvNet	$0.360_{\pm 0.104}$	-
ConvNet	CW	$0.363_{\pm 0.130}$	-
ConvNet	ProtoPNet	$0.316_{\pm 0.056}$	-
ProtoPNet	ProtoPNet	$0.349_{\pm 0.114}$	$0.349_{\pm 0.114}$
ProtoPNet	CW	$0.224_{\pm 0.092}$	$0.224_{\pm 0.092}$
ProtoPNet	ConvNet	$0.209_{\pm 0.043}$	$0.209_{\pm 0.043}$
CW	CW	$0.238_{\pm 0.047}$	$0.305_{\pm 0.010}$
CW	ConvNet	$0.405_{\pm 0.057}$	$0.415_{\pm 0.052}$
CW	ProtoPNet	$0.287_{\pm 0.064}$	$0.298_{\pm 0.092}$

Table 1: Language compositionality under different ResNet-50 encoder variants f for sender and receiver on CUB10 test set (higher is better). All measures were statistically significant with $p \leq 1^{-97}$.

S_f	R_f	Top. Sim.	Dis. Sim.
ConvNet	ConvNet	$0.254_{\pm 0.054}$	-
ConvNet	ProtoPNet	$0.323_{\pm 0.039}$	-
ConvNet	CW	$0.181_{\pm 0.028}$	-
ProtoPNet	ProtoPNet	$0.395_{\pm 0.097}$	$0.395_{\pm 0.097}$
ProtoPNet	ConvNet	$0.357_{\pm 0.215}$	$0.357_{\pm 0.215}$
ProtoPNet	CW	$0.450_{\pm 0.012}$	$0.450_{\pm 0.012}$
CW	CW	$0.380_{\pm 0.071}$	$0.162_{\pm 0.134}$
CW	ProtoPNet	$0.335_{\pm 0.056}$	$0.052_{\pm 0.035}$
CW	ConvNet	$0.259_{\pm 0.078}$	$0.131_{\pm 0.012}$

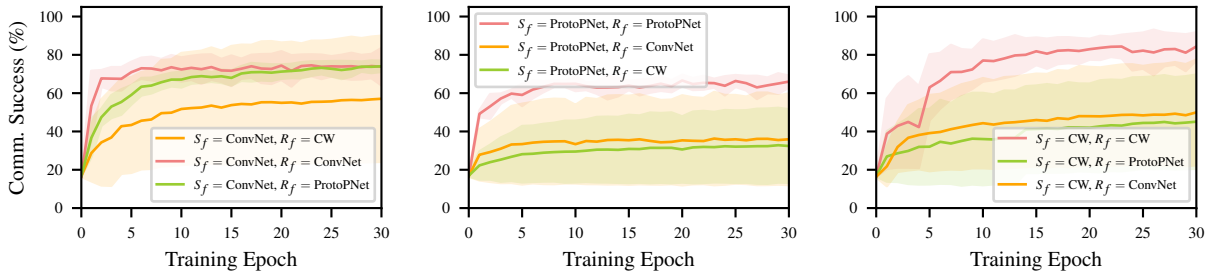
Table 2: Language compositionality under different encoder functions f for sender and receiver on mini-ImageNet test set (higher is better) . All measures were statistically significant with $p \approx 0$.

A1: Heterogeneous systems appear to underperform compared to same-encoder (e.g., homogeneous) systems, although the former can exhibit higher compositionality in the emerged language.

Categorization power on novel objects (Q2): To better understand the description of novel objects, in Figure 4 we repeat the previous Q1 experiment by swapping CUB10 for mini-ImageNet, whose test set consists of object classes not seen during training. We observe a similar trend that aligns with A1, except CW-CW outperforms other methods, at best scoring 85.585 ± 5.476 signaling accuracy compared to ConvNet-ConvNet (76.586 ± 6.400) and ProtoPNet-ProtoPNet (67.542 ± 3.001) despite realizing the least amount of concepts ($k = 10$) compared to ProtoPNet ($k = 100$). In some instances, e.g., ConvNet-ProtoPNet, the mismatched system can match the ConvNet-ConvNet performance within 3% accuracy.

In Table 2, we capture Top.Sim. and Dis.Sim. measures for mini-ImageNet. Despite only reaching a signaling accuracy of 33.328 ± 19.324, the highest Top.Sim. and Dis.Sim. scores (0.450) are realized by mismatched ProtoPNet-CW, thus compositionality was able to emerge despite the absence of generalization ability, as originally reported by Chaabouni et al. (2020). CW-CW and ProtoPNet-ProtoPNet scored similarly, obtaining 85.585 ± 5.476 and 67.542 ± 3.001 best epoch signaling accuracy, respectively. The best mixed-encoder system in Figure 4 was ConvNet-ProtoPNet (74.929 ± 3.079 accuracy), with a similarly high topographic similarity. We thus conclude:

A2: Heterogeneous systems evidence lower signaling performance compared to same-encoder baselines on novel objects, although the former are capable of improving the language compositionality. Holistically, disentangled networks either matched or outperformed the ConvNet baselines.



(a) (b) (c)

Figure 4: Comparison of communication success rate on mini-ImageNet few-shot testing data for permutations of a) ConvNet senders, b) ProtoPNet senders, and c) CW senders. Shaded regions denote standard deviation over five random seeds (each swapping in new sender encoder and receiver encoder checkpoints), with $L = 10$. Agents in this experiment have never seen the testing objects before.

Sender Arch.	Top. Sim.	Dis. Sim.	Recv. Acc.
ProtoPNet z' Utterance	0.523 $\pm$0.006	0.523 $\pm$0.006	81.625 $\pm$1.031
ProtoPNet	0.362 $\pm$0.078	0.362 $\pm$0.078	80.343 $\pm$0.681
ProtoPNet+LSA	0.383 $\pm$0.033	0.383 $\pm$0.033	81.496 $\pm$1.665
ConvNet	0.391 $\pm$0.055	-	86.753 $\pm$0.651

Table 3: Comparison of LSA module, 1-Length utterance baseline, and normal RNNs on CUB10 testing data using ProtoPNet disentangled structure to weight the vocabulary ($|V| = 10$).

Informing design (Q3): Tables 3 and 4 examine the aggregate effect on CUB10 test data after integrating concept-aligned latent vectors into the agent utterance, using our LSA module described in Section 3.1. We report statistics using each system's best signaling checkpoint. Since $|V| = k$, for ease of comparison we set $|V| = k = 10$ for both ProtoPNet and CW, and constrain all agents to communicate with fixed message lengths $L = 10$. In addition to the normal variants, we compare against a baseline which uses the latent vector $\mathbf{z}'$ directly as the vocabulary logits to produce a single utterance, establishing the potential usefulness of the disentangled vector for communication. In the case of ProtoPNet in Table 3, we observe that the LSA variant is competitive with the ConvNet baseline, and slightly improves over the regular ProtoPNet baseline. In fact, the 1-Length baseline was sufficient for generalization (81.625 ± 1.031 signaling accuracy). In Table 4 for CW models, we observe that the LSA module had significant impact, increasing signaling accuracy from 52.523 ± 17.600 to 79.859 ± 2.417. Notably, the 1-Length utterance was scored 75.670 ± 1.820, indicating that the learned concepts were more useful than the proto-language created by the regular CW variant. Despite this, the result with CW+LSA evidences that incorporating both sources of information is beneficial for signaling performance.

Apart from incorporating new information into the sender's message decoding process, our proposed LSA module may enable better diversity in the agent utterances. Recall from Section 3.1 that the disentangled representation is linearly recombined and multiplied with message logits at each time-step, thus we posit that the LSA module encourages agents to use vocabulary that is otherwise ignored, which may explain the higher topographic similarity scores observed in Tables 3 and 4. We study this idea for CW in Figure 5 (and ProtoPNet in Appendix A.5.1) by taking the top-performing multi-agent system for ConvNet (top row), CW (middle row), and CW+LSA (bottom row). Each inset heat map is the activated vocabulary usage (x-axis) across input data classes (y-axis) in the CUB10 test set at a given time step for each figure column (i.e., columns $t = 0$ to $t = 10$), with the cumulative sum over time displayed in the rightmost column. We quantify the vocabulary usage by calculating normalized area (A) of shaded heatmap pixels, shown below each cumulative matrix. ConvNet and CW variants (top and middle row, respectively) evidence a tendency to re-use vocabulary at each time-step and do not explore different combinations of vocabulary across time steps. LSA variants (bottom row) instead have a higher distribution of vocabulary across time, which can be seen qualitatively by comparing early time-steps to

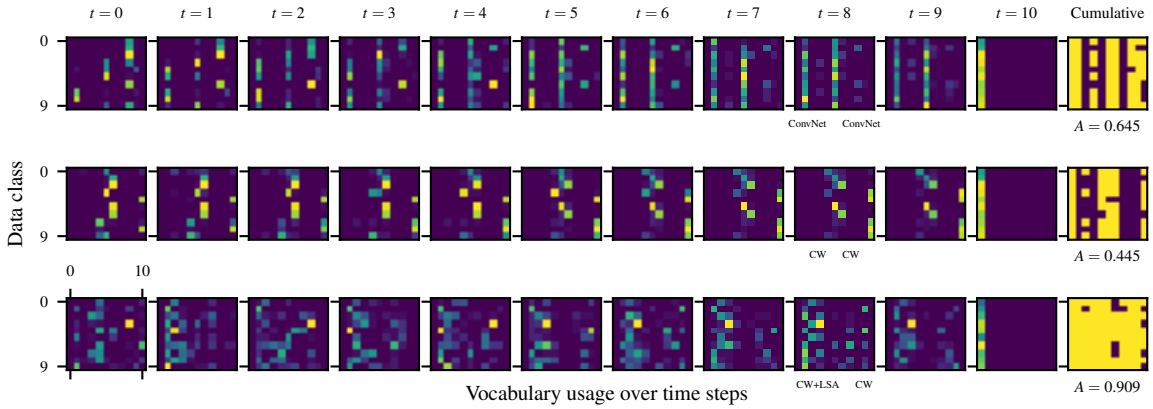


Figure 5: Heat-maps of the best-performing agent's vocabulary usage across time and data classes over the entire CUB10 test set for CW ($|V| = k = 10$), where agents communicate for ten time-steps (with end-of-sentence symbol '0' at the final time-step). The final column shows the cumulative sum over all time steps, subtitled by the normalized area of non-zero cells.

Sender Arch.	Top. Sim.	Dis. Sim.	Recv. Acc.
CW	0.325	0.503	75.670
z' Utterance	±0.031	±0.006	±1.820
CW	0.278	0.339	52.523
	±0.079	±0.044	±17.600
CW+LSA	0.266	0.439	79.859
	±0.087	±0.076	±2.417
ConvNet	0.391	-	86.753
	±0.055		±0.651

Table 4: Comparison of LSA module, 1-Length utterance baseline, and normal RNNs on CUB10 testing data using CW disentangled representation to weight the vocabulary ($|V| = 10$).

later time-steps. For example, the CW+LSA variant (bottom row) uses the vocabulary space $[6, 10)$ across classes early in the message, whereas this space is less used in later parts of the message (e.g., compare space $[6, 10)$ at $t = 2$ against $t = 9$). Notably, this space-swapping behavior can be seen in the ConvNet variant (top row), which causes higher cumulative area than the regular CW variant (middle row). In this analysis, CW+LSA obtains the highest cumulative area, so it can be said that this variant leverages the most tokens in the vocabulary, which may help explain the high disentangled similarity (0.439) in Table 4. We thus conclude:

A3: Incorporating the aligned subspace information into the vocabulary logits can have substantial impact on disentangled agents, remaining competitive with the non-disentangled ConvNet baseline accuracy, and improving or matching concept alignment (Dis. Sim.) compared to the baseline CW or ProtoPNet variant.

Categorization power and communication (Q4): Throughout the evaluation, we have compared against ProtoPNet ($k = 100$) and CW ($k = 10$). To better isolate the concept realization implied by k, we train ProtoPNet encoders for k values in $\{1, 10, 100\}$, then play signaling games with mixed ProtoPNet variations (denoting each as S_k or R_k) using fixed message length $L = 10$. We report the communication success rate over epochs in Figure 6, where each sub-figure is a fixed S_k. For $S_k = 10$ in Figure 6a, it is capable of generalization when paired with $R_k \in \{10, 100\}$, and suffers with $R_k = 1000$. Although $R_k = 1000$ under-performs in each case (Figures 6a-c), we observe that ProtoPNet encoders with $k = 1000$ have sufficient generalization in the upstream classification task, scoring at least 91% accuracy on CUB10 (we report all upstream scores in Appendix Table 13). This finding is nuanced by considering additional model architectures, such as DenseNet-161 in Appendix Section A.4.1, which shows that the complexity penalty observed in Figures 6a-c is mitigated by using the larger DenseNet-161 architecture (illustrated in Appendix Figure 10). Thus certain model architectures have unexpected benefits when deploying more complicated sender/receiver pairs. We thus conclude:

A4: Better concept realization does not always equate to better reception, despite performance on an upstream task such as classification. Otherwise, mixed-k agents have a tendency to generalize similarly, despite differences in the number of concepts.

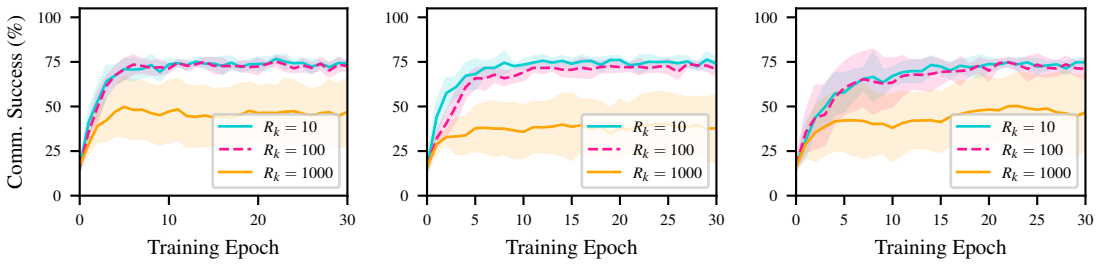


Figure 6: Modulating the sender's categorization ability over realized concepts k for a) $S_k = 10$, b) $S_k = 100$, and c) $S_k = 1000$, when paired with receivers of varying categorization ability.

5 Related Work

The properties of language acquisition and emergence of its structural properties (e.g., compositionality and recursion) were investigated early on by Kirby (1998), who proposed that natural language can evolve to possess an internal structure as a result of observational learning. Due to the availability of complex real-world data and the scalability of deep learning, recent studies in language evolution have focused on artificial, yet realistic analogues of language evolution (Lazaridou et al., 2017; Havrylov and Titov, 2017; Lazaridou and Baroni, 2020). Notably, artificial language emergence has been shown to exhibit an inner structure as originally investigated by Kirby (1998), e.g., compositionality (Lazaridou et al., 2018; Chaabouni et al., 2020) and encoding preference (Chaabouni et al., 2019). During artificial language formation, agents modulate the structure of the language in service of jointly solving the learning task. For example, by negotiating symbols in the artificial language, Hagiwara et al. (2019) evidenced the ability for same-encoder agents to realize data categories in an unsupervised way, leading to a form of emergent categorization. The most related work to ours is by Chaabouni et al. (2020), who study concept formation in same-encoder artificial agents through the lens of disentanglement. However, their study focuses on proposing new metrics of compositionality in small-scale, categorical data. We instead investigate the case of mixed-encoder agents, each realizing a different level of disentangled categorization ability on large-scale RGB image data.

6 Conclusion

Our results suggest that agents in a cooperative game setting are capable of communication despite differences in concept realization. Despite this, we find that heterogeneous agents can be at odds with each other, often under-performing compared to their same-encoder variants. In this way, the reasoning process of an agent can largely determine its interaction with other agents. On novel objects, there is a notable advantage to using disentangled networks, and in some cases mismatched agents, evidenced by higher signaling success and language compositionality. Our proposed Latent Self-attention module illustrates the advantage of incorporating an agent's categorization power into utterances, evidenced by higher disentangled similarity in experiments. The interplay between disentanglement and interpretability of the emerged language will be investigated in future work.

Acknowledgements

This work was funded by the Air Force Research Laboratory's Early Career Award program in conjunction with the AFOSR Center of Excellence on Assured Autonomy (AFRL-FA9550-19-1-0169) and NSF grant CNS-2055123.

References

Henry Brighton and Simon Kirby. 2006. Understanding linguistic evolution by visualizing the emergence. *Artificial Life*, 12(2):229–242.

Rahma Chaabouni, Eugene Kharitonov, Diane Bouchacourt, Emmanuel Dupoux, and Marco Baroni. 2020. Compositionality and generalization in emergent languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4427–4442, Online. Association for Computational Linguistics.

Rahma Chaabouni, Eugene Kharitonov, Emmanuel Dupoux, and Marco Baroni. 2019. Anti-efficient encoding in emergent communication. In *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.

Daniel Chandler. 2007. *Semiotics: The basics*, 2nd edition. Basics (Routledge (Firm). Routledge, London; New York. OCLC: ocm70864411.

Chaofan Chen, Oscar Li, Daniel Tao, Alina Barnett, Cynthia Rudin, and Jonathan Su. 2019. This looks like that: Deep learning for interpretable image recognition. In *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 8928–8939.

Zhi Chen, Yijie Bei, and Cynthia Rudin. 2020. Concept whitening for interpretable image recognition. *Nature Machine Intelligence*, 2(12):772–782. ArXiv: 2002.01650.

Dedre Gentner. 2010. Bootstrapping the mind: Analogical processes and symbol systems. *Cognitive Science*, 34(5):752–775.

Yoshinobu Hagiwara, Hiroyoshi Kobayashi, Akira Taniguchi, and Tadahiro Taniguchi. 2019. Symbol Emergence as an Interpersonal Multimodal Categorization. *Frontiers in Robotics and AI*, 6:134.

Serhii Havrylov and Ivan Titov. 2017. Emergence of language with multi-agent games: Learning to communicate with sequences of symbols. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 2149–2159.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778. IEEE Computer Society.

Douglas R. Hofstadter. 2007. *I am a strange loop*. Basic Books, New York. OCLC: ocm64554976.

Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. 2017. Densely connected convolutional networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 2261–2269. IEEE Computer Society.

Eugene Kharitonov, Rahma Chaabouni, Diane Bouchacourt, and Marco Baroni. 2019. EGG: a toolkit for research on emergence of lanGuage in games. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations*, pages 55–60, Hong Kong, China. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Simon Kirby. 1998. Learning, bottlenecks and the evolution of recursive syntax. In *In E. Briscoe (Ed.), Linguistic*. University Press.

Satwik Kottur, José Moura, Stefan Lee, and Dhruv Batra. 2017. 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. Association for Computational Linguistics.

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

Angeliki Lazaridou, Karl Moritz Hermann, Karl Tuyls, and Stephen Clark. 2018. 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.

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

Jason Lee, Kyunghyun Cho, Jason Weston, and Douwe Kiela. 2018. Emergent translation in multi-agent communication. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

David Lewis. 1969. Convention. *Cambridge, MA*.

Marvin Minsky. 1988. *Society Of Mind*. Simon and Schuster. Google-Books-ID: bLDLllfRpdkC.

Cinjon Resnick, Abhinav Gupta, Jakob Foerster, Andrew M. Dai, and Kyunghyun Cho. 2020. Capacity, bandwidth, and compositionality in emergent language learning. *arXiv:1910.11424 [cs, stat]*. ArXiv: 1910.11424.

Cynthia Rudin, Chaofan Chen, Zhi Chen, Haiyang Huang, Lesia Semenova, and Chudi Zhong. 2022. Interpretable Machine Learning: Fundamental principles and 10 grand challenges. *Statistics Surveys*, 16:1–85.

John Schulman, Nicolas Heess, Theophane Weber, and Pieter Abbeel. 2015. Gradient estimation using stochastic computation graphs. In *Advances in Neural Information Processing Systems 28: Annual*

Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada, pages 3528–3536.

Karen Simonyan and Andrew Zisserman. 2015. Very deep convolutional networks for large-scale image recognition. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Brian Skyrms and Oxford University Press. 2010. *Signals: Evolution, learning, and information*. Signals: Evolution, learning, & information. OUP Oxford.

Endel Tulving. 2002. Episodic memory: From mind to brain. *Annual Review of Psychology*, 53(1):1–25.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Oriol Vinyals, Charles Blundell, Tim Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. 2016. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 3630–3638.

Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. 2011. The Caltech-UCSD Birds-200-2011 dataset. Technical Report CNS-TR-2011-001, California Institute of Technology.

Bolei Zhou, David Bau, Aude Oliva, and Antonio Torralba. 2018. Interpreting deep visual representations via network dissection. *IEEE transactions on pattern analysis and machine intelligence*, 41(9):2131–2145.

Bolei Zhou, Aditya Khosla, Àgata Lapedriza, Aude Oliva, and Antonio Torralba. 2015. Object detectors emerge in deep scene CNNs. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

A Appendix

A.1 Training details

In our experiments, every pair of sender and receiver is trained with distinct seeds, thus they never share the same checkpoint, only the training data. We use the publicly available code to train encoders from the respective authors (Chen et al., 2019, 2020). The underlying model for CW and ProtoPNet encoders is a standard ResNet-50 CNN (He et al., 2016) trained on the full-size ImageNet dataset[3], and fine-tuned on the respective dataset (CUB10 or mini-ImageNet) for five epochs over ten random seeds (i.e., unique sender/receiver seeds for five systems).

The Gumbel-Softmax estimator samples from the Gumbel distribution with a temperature term, which controls how similar the relaxation is to the true argmax distribution. As recommended by Havrylov and Titov (2017), we allow the sender agent to learn the inverse of the temperature in order to avoid manual tuning. Agent parameters are tuned by stochastic gradient descent (SGD) using the Adam optimization algorithm (Kingma and Ba, 2015). The recurrent sender decoder g and receiver encoder h are instantiated in practice with recurrent neural networks (RNNs) using LSTM cells. We leverage the NVIDIA DALI library to accelerate data loading through the use of GPGPU hardware. [4]

A.2 Hyperparameter selection

In order to achieve the best performance for agents and encoders, we performed a grid-search over the following hyperparameter combinations:

- Number of epochs in $\{30, 50, 100\}$,
- Hidden dimension of LSTM hidden state in $\{64, 128, 256\}$,
- Number of distractors in $\{1, 3, 5\}$,
- Message length L in $\{10, 100\}$,
- ConvNet classifier fine-tuning learning rate in $\{1^{-2}, 1^{-3}\}$,
- ProtoPNet dense representation network f learning rate in $\{1^{-4}, 1^{-5}\}$,
- ProtoPNet interface layer learning rate in $\{3^{-5}, 3^{-4}, 3^{-3}\}$,
- ProtoPNet prototype layer learning rate in $\{3^{-5}, 3^{-4}, 3^{-3}\}$,
- ProtoPNet classifier learning rate in $\{1^{-5}, 1^{-4}\}$,
- CW classifier and whitening layer learning rate $\{1^{-2}, 1^{-3}\}$,

[3] https://pytorch.org/vision/stable/models.html

[4] https://github.com/NVIDIA/DALI

- Sender learning rate in $\{1^{-5}, 1^{-4}, 1^{-3}, 1^{-2}\}$, and
- Receiver learning rate in $\{1^{-5}, 1^{-4}, 1^{-3}, 1^{-2}\}$.

As a result of the grid-search, we selected the best performing hyperparameter combination which balanced training time and communication success rate performance (e.g., receiver signaling accuracy). This combination is shown in Table 5.

A.3 Hardware description

All experiments were performed on an NVIDIA DGX-2 compute node equipped with 16 NVIDIA Tesla V100 Tensor Core GPUs, high-speed NVMe flash storage, 1TB main system memory, and 40 Intel Xeon Platinum CPU cores.

A.4 Supplemental Results

A.4.1 Results on VGG-16 and DenseNet-161 variants

The disentangled representations discussed in the main text are not restricted to residual networks such as ResNet-50, in fact they are flexible enough to interact with other network designs such as VGG-16 (Simonyan and Zisserman, 2015) and DenseNet-161 (Huang et al., 2017). To expand the empirical contribution, we provide additional results addressing Q1, Q3, and Q4 of the main text using disentangled image encoders built on top of VGG-16 and DenseNet-161, as well as their non-disentangled baselines.

Q1 (VGG-16 & DenseNet-161). In Figures 7 and 8, we see the same general tendency for same-encoder systems to either match or outperform the mixed-encoder versions. This is evidenced on VGG-16 (Figure 7, red line) across each sender variant, with CW being a notable exception due to performing similarly between CW-CW and CW-ConvNet variants. On DenseNet-161, we observe a failure mode for CW (Figure 8c) due to low accuracy on the upstream task (Table 16). We experimentally observed that CW causes training instability for DenseNet-161 on CUB10. In this failure mode, there is little difference between same- and mixed-encoder agent systems, which may indicate that the performance discrepancies are tied to success of the upstream task.

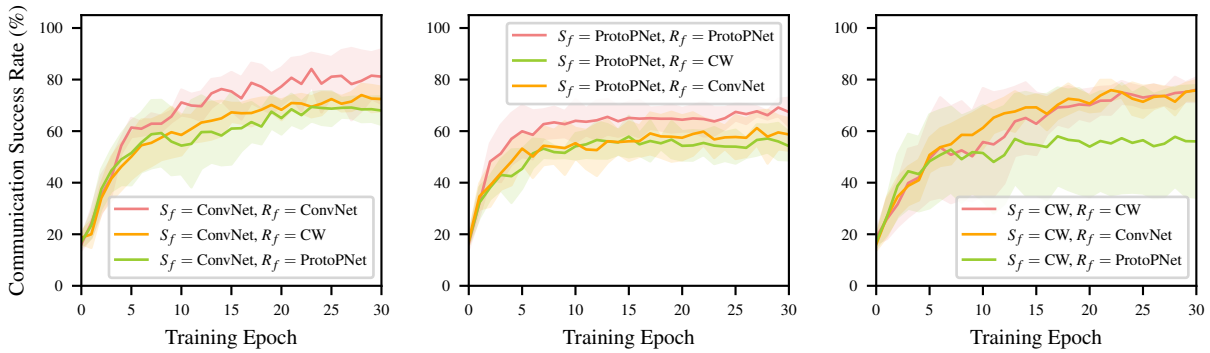
As in the main text, we provide measures of the language compositionality under different encoder variants for VGG-16 (Table 6) and DenseNet-161 (Table 7). In this context, topographic similarity or disentangled similarity for mixed-encoder systems can be higher in some cases (e.g., VGG-16 CW-ConvNet Dis. Sim. of 0.435 compared to 0.401 of CW-CW in Table 6, or VGG-16 ConvNet-CW Top.Sim. of 0.312 compared to 0.305 of ConvNet-ConvNet). In general, same-encoder variants evidence a tendency to outperform the mixed-encoder variants, invariant to the choice of underlying architecture. Returning to the failure mode of DenseNet-161 CW seen previously, we observe that the topographic similarity scores for DenseNet-161 CW variants in Table 7 are competitive with models that did not experience failure, e.g., the CW-CW system scored Top.Sim. of 0.391 whereas ProtoPNet-ProtoPNet scored 0.369. This is further evidence of the effect observed by Chaabouni et al. (2020), which is that compositionality can emerge despite lack of generalization ability.

Q3 (VGG-16 & DenseNet-161). We investigated Q3 under the same constraints of Section 4, but instead using VGG-16 (Tables 8 and 9) and DenseNet-161 (Tables 10 and 11) instead of ResNet-50. We observe the general tendency from Section 4 for RNN+LSA variants to either match or outperform the standard RNN variants, whilst remaining competitive with the non-disentangled ConvNet baseline. The exception to this trend is the failure mode of DenseNet-161 CW variants (Table 11), despite evidencing higher disentangled similarity with low generalization ability (e.g., Dis.Sim of 0.785 for RNN+LSA with 45.487% receiver accuracy).

Q4 (VGG-16 & DenseNet-161). Under the constraints of Section 4, we repeat the experiment using encoders based on VGG-16 (Figure 9) and DenseNet-161 (Figure 10). Notably, variants based on VGG-16 exhibit the same trend from Section 4 for more complex $k = 1000$ agents to perform poorly as receivers (orange lines in Figure 9). The results for DenseNet-161-based variants introduce nuance to this trend, since complex receivers in this setup perform within the random variance of other receivers (i.e., orange lines in Figure 10 being within blue and pink shading). Referencing the upstream performance of each variant in Tables 13, 15, and 17 offers some insight. First, we observe that DenseNet-161 variants are the best performing variants of ProtoPNet (up to 97% accuracy compared to 90% and 95% of VGG-16 and

Hyperparameter	Main paper choice(s)	Remarks
Sender learning rate (LR)	1^{-3}	
Receiver LR	1^{-3}	
Lewis Signaling Game Epochs	30	
Number of distractors	5	
Message length L	10	
LSTM Hidden Dim.	256	
ConvNet fine-tuning LR	1^{-2}	Standard setting for CNNs
ProtoPNet Dense Rep. LR	1^{-5}	Same as Chen et al. (2019)
ProtoPNet Interface LR	3^{-3}	Same as Chen et al. (2019)
ProtoPNet Prototypes LR	3^{-3}	Same as Chen et al. (2019)
ProtoPNet Classifier LR	1^{-4}	Same as Chen et al. (2019)
ProtoPNet Epochs	20	
CW LR	1^{-2}	Same as Chen et al. (2020)

Table 5: Hyperparameter choices based on grid-search.



(a) (b) (c)

Figure 7: Comparison of communication success rate on CUB10 testing data for permutations of a) ConvNet senders, b) ProtoPNet senders, and c) CW senders, using image encoders based on VGG-16. Shaded regions denote standard deviation over five random seeds (each swapping in a new CUB10 test split, sender encoder checkpoint, and receiver encoder checkpoint), with $L = 10$.

ResNet-50, respectively), and second, DenseNet-161 requires more model parameters (on the order of 26M compared to 15M and 24M of VGG-16 and ResNet-50 variants). Given this result, it may be possible to mitigate the complexity penalty observed in Section 4 by leveraging parameter-heavy densely-connected neural networks, although the impact on smaller models is notable. The effect of model parameterization on the language evolution offers an interesting direction for future work.

A.4.2 Upstream task metrics

In Table 12 we provide the accuracy score for each ResNet-50 encoder's upstream classification task on CUB10 data. We use the training code open-sourced by the respective paper authors (Chen et al., 2019, 2020). Table 13 gives the accuracy scores for ProtoPNet encoders used in Section 4. The same scores are provided for supplemental variants corresponding to VGG-16 (Tables 14 and 15) and DenseNet-161 (Tables 16 and 17).

A.5 Supplemental Analysis

A.5.1 LSA vocabulary usage

In Figure 11 we repeat the same experiment from Section 4, this time for ProtoPNet variants. Although the ProtoPNet+LSA variant (bottom row) does not score the highest cumulative vocabulary usage compared to the ConvNet variant (0.445 and 0.645, respectively), it encourages more exploration of the vocabulary compared to the regular ProtoPNet variant (0.327). We observe the same trend with CW in Figure 5, which suggests the LSA mechanism can enable better vocabulary diversity on multiple types of disentangled representations.

A.5.2 LSA concept alignment

Our proposed LSA module requires the disentangled sender agent to weight their utterances by the concept weight matrix M', which is obtained by

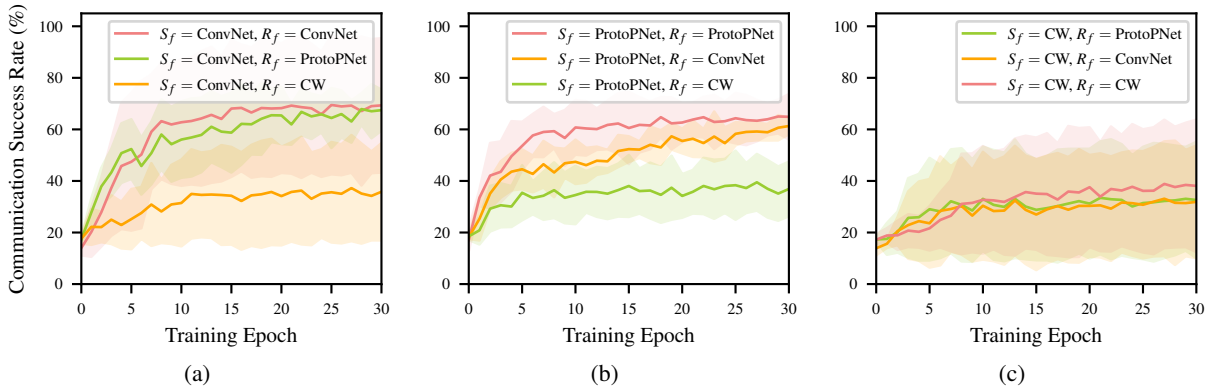


Figure 8: Comparison of communication success rate on CUB10 testing data for permutations of a) ConvNet senders, b) ProtoPNet senders, and c) CW senders, using image encoders based on DenseNet-161. Shaded regions denote standard deviation over five random seeds (each swapping in a new CUB10 test split, sender encoder checkpoint, and receiver encoder checkpoint), with $L = 100$.

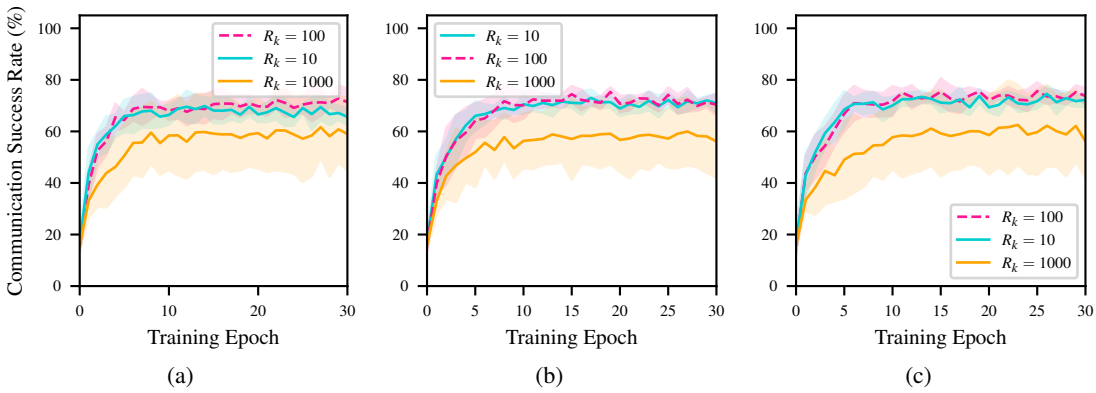


Figure 9: Modulating the sender's categorization ability over realized concepts k for a) $S_k = 10$, b) $S_k = 100$, and c) $S_k = 1000$, when paired with receivers of varying categorization ability and using VGG-16 as the base encoder architecture.

the product of the learned L-head matrix M and initial disentangled representation z' (as discussed in Section 3.1). This design enables a qualitative analysis of the agent's concept activation at each time step, i.e., at some time t, we can plot the image that corresponds to the highest-weighted dimension in that time-step, $\max(M'[t])$. This is done for six images across three data classes for the best performing LSA+ProtoPNet model (Figure 12) and LSA+CW model (Figure 13) from Section 4. We observe that during the decoding of utterances, the most active concept does not always correspond to visual semantic features of the input data class, e.g., brown and black textures are not always present in the top two rows. Instead, the sender agent may rely on an arbitrary positional encoding of known concepts to describe certain visual features. This is evident across both ProtoPNet and CW models. A notable effect of the LSA module on CW variants is an increased exploration of the agent vocabulary, as discussed in Section 4, which can be observed in Figure 13 by the lack of repetition between top categories, whereas in Figure 12 we note the same concept is often repeated several times within the same message. Given the qualitative results, we posit that agents rely on an encoding preference that can be considered incongruent with realized concepts of the disentangled models. For example, agents might rely on the position of a concept in the utterance, rather than the concept itself, to capture information that leads to better decisions by the receiver. The LSA module enables a first look into the connection between these grounded concepts and their recall during agent utterances, the enforcement of which offers an exciting direction for future work.

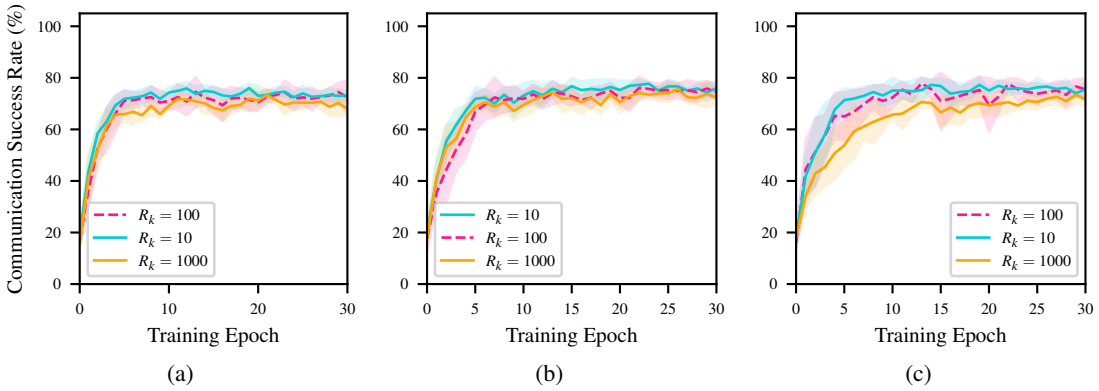


Figure 10: Modulating the sender's categorization ability over realized concepts k for a) $S_k = 10$, b) $S_k = 100$, and c) $S_k = 1000$, when paired with receivers of varying categorization ability and using DenseNet-161 as the base encoder architecture.

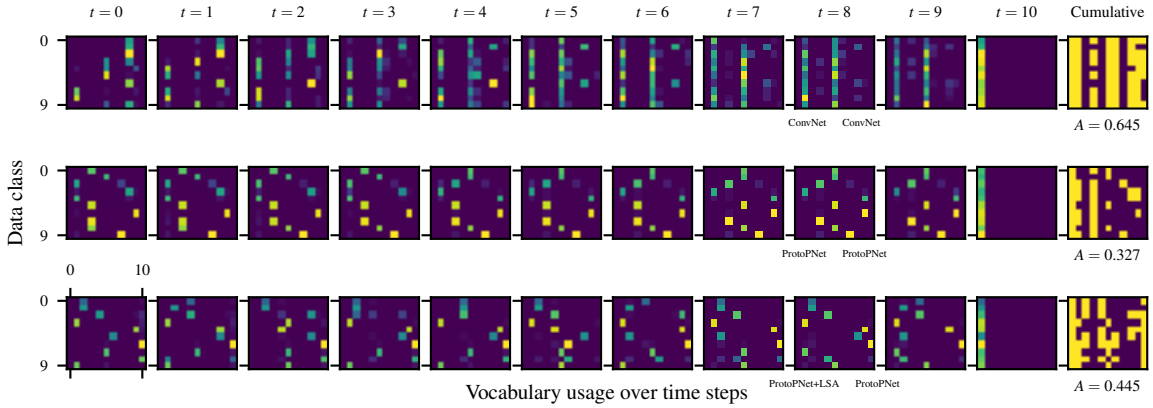


Figure 11: Heat-maps of the best-performing agent's vocabulary usage across time and data classes over the entire CUB10 test set for ProtoPNet, corresponding to experiments based on ResNet-50 in Section 4, where $|V| = k = 10$, and agents communicate for ten time-steps (with end-of-sentence symbol '0' at the final time-step). The final column shows the cumulative sum over all time steps, subtitled by the normalized area of non-zero cells.

Sender Arch.	Recv. Arch.	Top. Sim.	Dis. Sim.
ConvNet	ConvNet	0.305 $\pm$0.111	-
ConvNet	CW	0.312 $\pm$0.068	-
ConvNet	ProtoPNet	0.305 $\pm$0.066	-
ProtoPNet	CW	0.299 $\pm$0.143	0.299 $\pm$0.143
ProtoPNet	ProtoPNet	0.380 $\pm$0.101	0.380 $\pm$0.101
ProtoPNet	ConvNet	0.255 $\pm$0.063	0.255 $\pm$0.063
CW	CW	0.330 $\pm$0.042	0.401 $\pm$0.079
CW	ProtoPNet	0.186 $\pm$0.039	0.261 $\pm$0.050
CW	ConvNet	0.297 $\pm$0.032	0.435 $\pm$0.089

Table 6: Language compositionality under different VGG-16 encoder variants f for sender and receiver on CUB10 test set (higher is better). All measures were statistically significant with $p \approx 0$.

Sender Arch.	Recv. Arch.	Top. Sim.	Dis. Sim.
ConvNet	ConvNet	0.437 $\pm$0.078	-
ConvNet	CW	0.296 $\pm$0.168	-
ConvNet	ProtoPNet	0.304 $\pm$0.029	-
ProtoPNet	CW	0.208 $\pm$0.115	0.208 $\pm$0.115
ProtoPNet	ProtoPNet	0.369 $\pm$0.054	0.369 $\pm$0.054
ProtoPNet	ConvNet	0.288 $\pm$0.118	0.288 $\pm$0.118
CW	CW	0.391 $\pm$0.112	0.265 $\pm$0.068
CW	ProtoPNet	0.367 $\pm$0.033	0.186 $\pm$0.100
CW	ConvNet	0.390 $\pm$0.013	0.235 $\pm$0.056

Table 7: Language compositionality under different DenseNet-161 encoder variants f for sender and receiver on CUB10 test set (higher is better). All measures were statistically significant with $p \approx 0$.

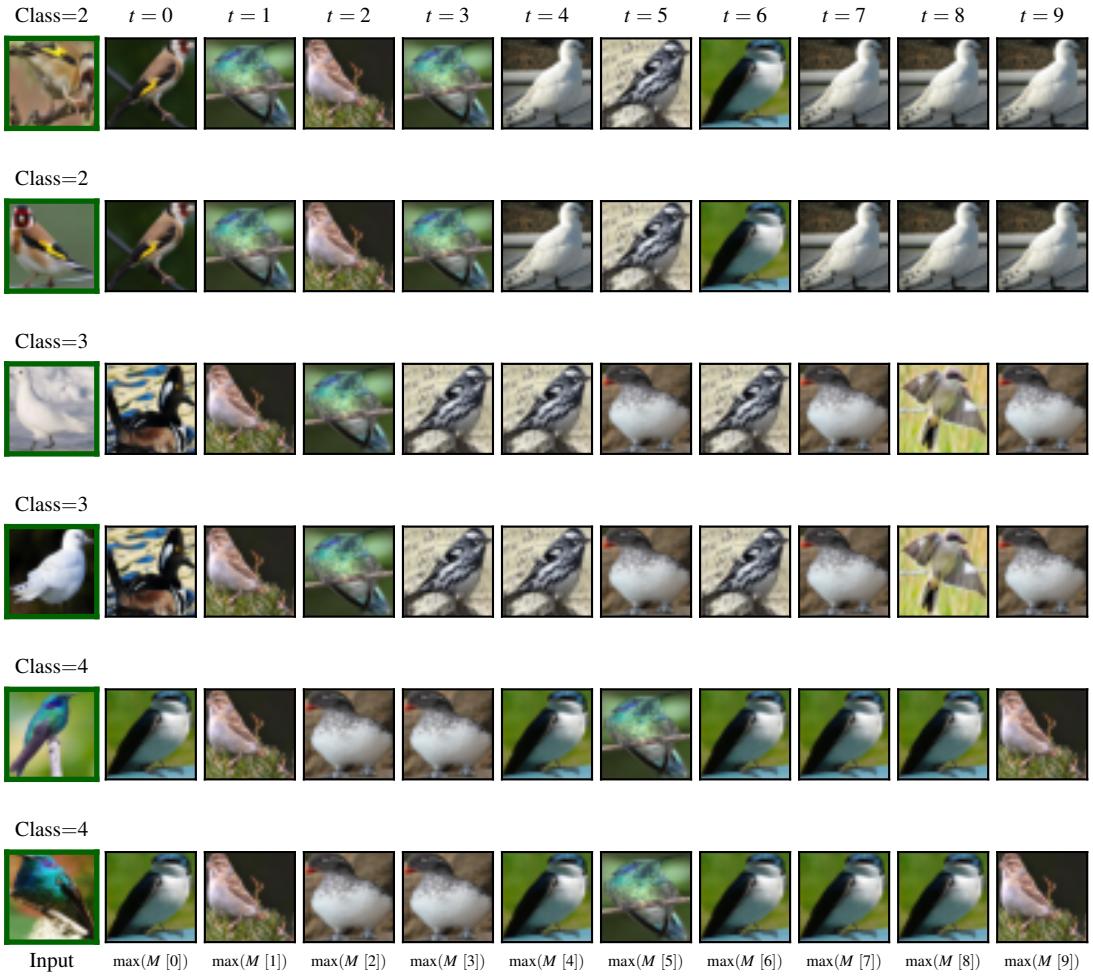


Figure 12: Visualization of the most activated prototypical image at each time-step in an LSA+ProtoPNet agent's concept weight matrix (M'), given an input image from three classes in CUB10. When decoding utterances, the most activated concept at a given time-step does not always correspond to the class of the observation.

Sender Arch.	Top. Sim.	Dis. Sim.	Recv. Acc.
ProtoPNet Utterance	0.263 ±0.039	0.263 ±0.039	58.110 ±14.306
RNN	0.468 ±0.064	0.468 ±0.064	76.737 ±1.995
RNN+LSA	0.424 ±0.109	0.424 ±0.109	76.619 ±1.863
RNN (ConvNet Baseline)	0.422 ±0.080	-	92.590 ±2.945

Table 8: Comparison of LSA module, 1-Length utterance baseline, and normal RNNs on CUB10 testing data using ProtoPNet disentangled structure to weight the vocabulary ($|V| = 10$) and VGG-16 as the base image encoder.

Sender Arch.	Top. Sim.	Dis. Sim.	Recv. Acc.
CW Utterance	0.354 ±0.032	0.496 ±0.018	77.928 ±2.208
RNN	0.313 ±0.048	0.385 ±0.058	83.167 ±2.032
RNN+LSA	0.294 ±0.035	0.520 ±0.047	86.495 ±0.622
RNN (ConvNet Baseline)	0.367 ±0.033	-	92.653 ±1.894

Table 9: Comparison of LSA module, 1-Length utterance baseline, and normal RNNs on CUB10 testing data using CW disentangled representation to weight the vocabulary ($|V| = 10$) and VGG-16 as the base image encoder.

Figure 13: Visualization of the most activated image category at each time-step in an LSA+CW agent's concept weight matrix (M'), given an input image from three classes in CUB10. When decoding utterances, the most activated concept at a given time-step does not always correspond to the class of the observation.

Sender Arch.	Top. Sim.	Dis. Sim.	Recv. Acc.
ProtoPNet Utterance	0.319 ±0.027	0.319 ±0.027	80.276 ±2.416
RNN	0.325 ±0.037	0.325 ±0.037	79.793 ±1.136
RNN+LSA	0.376 ±0.072	0.376 ±0.072	79.612 ±3.409
RNN (ConvNet Baseline)	0.407 ±0.092	-	88.929 ±1.253

Table 10: Comparison of LSA module, 1-Length utterance baseline, and normal RNNs on CUB10 testing data using ProtoPNet disentangled structure to weight the vocabulary ($|V| = 10$) and DenseNet-161 as the base image encoder.

Sender Arch.	Top. Sim.	Dis. Sim.	Recv. Acc.
CW Utterance	0.210 ±0.061	0.771 ±0.188	40.547 ±9.303
RNN	0.377 ±0.073	0.135 ±0.085	44.182 ±29.066
RNN+LSA	0.203 ±0.070	0.785 ±0.183	45.487 ±15.481
RNN (ConvNet Baseline)	0.431 ±0.048	-	89.454 ±3.403

Table 11: Comparison of LSA module, 1-Length utterance baseline, and normal RNNs on CUB10 testing data using CW disentangled representation to weight the vocabulary ($|V| = 10$) and DenseNet-161 as the base image encoder.

	f^S Acc. (# Param.)	f^R Acc. (# Param.)
ConvNet	95.280 ± 1.143 (23.52M)	94.951 ± 1.744 (23.52M)
ProtoPNet	91.559 ± 3.751 (23.81M)	93.183 ± 2.731 (23.81M)
CW	72.899 ± 8.264 (23.52M)	78.593 ± 8.293 (23.52M)

Table 12: Average classification accuracy of all Q1 CUB10 sender and receiver encoders based on ResNet-50 (i.e., the variants used for main paper results). Each entry represents the average over 5 random seeds (each row consisting of 10 seeds total). Parameter counts are applicable to Q2 experiments, due to shared model architectures.

k	f^S Acc. (# Param.)	f^R Acc. (# Param.)
10	95.884 ± 0.980 (23.78M)	94.335 ± 2.457 (23.78M)
100	91.559 ± 3.751 (23.81M)	93.183 ± 2.731 (23.81M)
1000	92.891 ± 1.017 (24.05M)	93.371 ± 1.209 (24.05M)

Table 13: Average classification accuracy of sender and receiver encoders based on ResNet-50 used in Section 4 (Q4). Each entry represents the average over 5 random seeds (each row consisting of 10 seeds total).

	f^S Acc. (# Param.)	f^R Acc. (# Param.)
ConvNet	93.327 ± 1.860 (134.30M)	94.191 ± 1.822 (134.30M)
ProtoPNet	86.558 ± 3.036 (14.83M)	88.799 ± 2.449 (14.83M)
CW	87.419 ± 3.736 (134.30M)	88.127 ± 2.352 (134.30M)

Table 14: Average classification accuracy of all Q1 CUB10 sender and receiver encoders based on VGG-16 (from supplemental results). Each entry represents the average over 5 random seeds (each row consisting of 10 seeds total).

k	f^S Acc. (# Param.)	f^R Acc. (# Param.)
10	90.001 ± 4.486 (14.80M)	89.523 ± 3.719 (14.80M)
100	86.558 ± 3.036 (14.83M)	88.799 ± 2.449 (14.83M)
1000	90.884 ± 3.125 (15.07M)	89.482 ± 3.464 (15.07M)

Table 15: Average classification accuracy of sender and receiver encoders based on VGG-16 for Q4 supplemental results. Each entry represents the average over 5 random seeds (each row consisting of 10 seeds total).

	f^S Acc. (# Param.)	f^R Acc. (# Param.)
ConvNet	97.974 ± 0.472 (26.49M)	98.175 ± 0.759 (26.49M)
ProtoPNet	96.485 ± 1.063 (26.79M)	94.793 ± 2.188 (26.79M)
CW	25.690 ± 24.975 (26.49M)	25.545 ± 30.644 (26.49M)

Table 16: Average classification accuracy of all Q1 CUB10 sender and receiver encoders based on DenseNet-161 (from supplemental results). Each entry represents the average over 5 random seeds (each row consisting of 10 seeds total).

k	f^S Acc. (# Param.)	f^R Acc. (# Param.)
10	97.085 ± 0.679 (26.77M)	96.756 ± 0.666 (26.77M)
100	96.485 ± 1.063 (26.79M)	94.793 ± 2.188 (26.79M)
1000	95.767 ± 2.818 (27.03M)	95.398 ± 0.966 (27.03M)

Table 17: Average classification accuracy of sender and receiver encoders based on DenseNet-161 for Q4 supplemental results. Each entry represents the average over 5 random seeds (each row consisting of 10 seeds total).