VQEL:
Enabling Self-Play in Emergent Language Games via Agent-Internal Vector Quantization

Anonymous authors
Paper under double-blind review

Abstract

Emergent Language (EL) focuses on the emergence of communication among artificial agents. Although symbolic communication channels more closely mirror the discrete nature of human language, learning such protocols remains fundamentally difficult due to the non-differentiability of symbol sampling. Existing approaches typically rely on high-variance gradient estimators such as REINFORCE or on continuous relaxations such as Gumbel–Softmax, both of which suffer from limitations in training stability and scalability when learning a language from scratch. Motivated by cognitive theories that emphasize intrapersonal processes preceding communication, we explore self-play as a substrate for language emergence prior to mutual interaction. We introduce Vector Quantized Emergent Language (VQEL), a novel architecture that incorporates vector quantization into the message generation process. VQEL enables agents to perform self-play using discrete internal representations derived from a learned codebook while preserving end-to-end differentiability. By grounding the vocabulary through dense gradients in self-play, VQEL completely avoids the cold-start instability of reinforcement learning. The resulting vector-quantized codebook naturally induces a symbolic vocabulary that serves as a highly robust initialization for subsequent REINFORCE-based fine-tuning during mutual play with other agents. Empirical results show that agents pretrained via VQEL self-play achieve more consistent symbol alignment and higher task success when later engaged in mutual interaction. These findings position self-play as a principled and effective mechanism for learning discrete communication protocols, addressing key optimization and representational challenges in emergent language systems.

1 Introduction

Emergent Language (EL) studies how communication protocols arise when artificial agents must coordinate to solve cooperative tasks in multi-agent environments Lazaridou et al. (2017); Havrylov & Titov (2017). In such settings, agents typically start without any predefined language and must learn to exchange information to achieve shared objectives. A central motivation of this research is twofold: to shed light on the mechanisms underlying human language evolution and acquisition, and to build artificial systems that can ultimately communicate with humans via natural language Mordatch & Abbeel (2018); Lazaridou et al. (2020); Chaabouni et al. (2021a). For this reason, many EL frameworks explicitly target *symbolic* communication channels. Unlike continuous message vectors, symbolic channels require agents to produce sequences of discrete tokens, better matching the discrete nature of human language Lazaridou et al. (2018a); Bouchacourt & Baroni (2018); Peters et al. (2025).

Training agents to communicate with discrete symbols, however, introduces a core technical obstacle: sampling discrete tokens is non-differentiable. As a result, gradients from the receiver cannot be backpropagated through the sender's discrete decisions. To address this, prior work has largely relied on (i) stochastic gradient estimators such as REINFORCE (Williams, 1992) and (ii) continuous relaxations such as the Gumbel-Softmax estimator (Jang et al., 2017b). However, both approaches suffer from high variance and training

instability. Instead of leveraging the informative directional guidance of gradients, they rely on weak scalar rewards, hindering convergence and often leading to suboptimal communication protocols.

Beyond these optimization concerns, theoretical perspectives in cognitive science and linguistics argue that language learning is not only driven by interpersonal exchange, but is also grounded in intrapersonal cognitive processes. Accounts such as the "Language of Thought" hypothesis (Fodor, 1975) and theories of "Inner Speech" (Alderson-Day & Fernyhough, 2015) suggest that agents may first develop internal conceptual structures, a private representational system used to organize experience, before aligning meanings through social interaction.

Motivated by these insights, we investigate whether self-play mechanisms in the context of EL can be leveraged to construct a smoother substrate for language emergence. From this perspective, a self-play–supported EL enables an agent to autonomously form, refine, and stabilize grounded concepts through self-interaction prior to communication with others. Crucially, such a self-play mechanism allows the agent's internal learning to occur directly within the representation space, producing rich learning signals that extend far beyond the sparse, scalar rewards typically employed in methods like REINFORCE. This leads to a central question: how can self-play be effectively instantiated and integrated with mutual play to alleviate the intrinsic difficulty of learning discrete communication protocols?

To answer the above question, we propose a novel architecture based on Vector Quantization (VQ). By integrating VQ into the agent's Message Generation Module, we provide a mechanism that discretizes continuous internal representations into a finite codebook of embedding vectors. This architecture solves the dilemma of Self-Play: it allows the agent to conduct internal games using discrete representations (via the codebook) while maintaining differentiability through the straight-through estimator or commitment losses associated with VQ. Consequently, the agent can "invent" a language internally without the instability of REINFORCE or the continuous relaxation of Gumbel-Softmax. Furthermore, the discrete cluster indices derived from the VQ codebook can be directly mapped to symbols, allowing the internally developed language to be seamlessly transferred and aligned during *Mutual-Play* with other agents.

The contributions of this paper are twofold:

1. We introduce VQEL (Vector Quantized Emergent Language), an architecture that leverages Vector Quantization to facilitate emergent language. By learning a foundational language via differentiable self-play, VQEL provides a stable, highly structured initialization that overcomes the optimization hurdles of standard REINFORCE and Gumbel-Softmax.

2. We demonstrate the efficacy of self-play in emergent language. Through extensive experiments, we illustrate that agents pretrained with VQEL self-play achieve better alignment and task success compared to those trained solely through mutual interaction from scratch.

2 Related work

2.1 Emergent Language

Emergent communication studies how artificial agents evolve protocols to solve cooperative tasks without predefined linguistic rules. The standard testbed is the referential game (Lewis signaling game), where a sender communicates a target perception to a receiver Lewis (2008); Lazaridou et al. (2016); Havrylov & Titov (2017). Recent work has expanded this framework to include multi-turn dialogue Evtimova et al. (2018); Jorge et al. (2016); Das et al. (2017); Graesser et al. (2019), population dynamics Ren et al. (2020); Fitzgerald (2019); Chaabouni et al. (2021c), and embodied environments Mordatch & Abbeel (2018). While most research focuses on symbolic transmission, others explore continuous signals Mihai & Hare (2021) or use communication as a means to solve non-communicative downstream goals rather than as the objective itself Chaabouni et al. (2019); Brandizzi et al. (2022); Eccles et al. (2019).

A primary challenge in symbolic emergent language is the non-differentiability of discrete message channels, which prevents standard backpropagation. Two predominant optimization strategies address this: Policy Gradients and Continuous Relaxations. The REINFORCE algorithm Williams (1992), widely used for its

implementation simplicity Foerster et al. (2016); Lazaridou et al. (2016); Bernard & Mickus (2023), treats communication as an action but suffers from high variance and instability Brandizzi (2023). Alternatively, the Gumbel-Softmax relaxation Jang et al. (2017a); Maddison et al. (2016) allows gradients to flow via reparameterization. While Gumbel-Softmax often yields higher performance Havrylov & Titov (2017); Chaabouni et al. (2020); Kharitonov et al. (2020), it relies on continuous approximations during training that deviate from strict discrete communication constraints.

Recent efforts have sought alternatives to these standard paradigms. For example, Carmeli et al. (2023) investigate message quantization, utilizing continuous communication during training and discretizing only during inference. However, this setup creates a discrepancy between training and testing phases and relies on simple scalar quantization. In contrast, our proposed VQEL method enforces discreteness throughout the learning process while maintaining semantic depth, addressing the limitations of prior quantization approaches.

2.2 Vector Quantization

Vector Quantization (VQ) is a technique widely used in the domain of signal processing and machine learning. It involves mapping a large set of input vectors into a finite set of output vectors, essentially discretizing continuous input space into a discrete representation space. Vector Quantization discretizes continuous data into discrete representations via a codebook—a finite set of vectors acting as centroids in the embedding space. During encoding, input vectors are mapped to the nearest codebook vector, converting continuous inputs into discrete representations and indices. Optimization refines the embedding space and codebook vectors using techniques like the straight-through estimator and moving average updates of codebook vectors. The most famous and well-known in this field is VQ-VAE, in which VQ is used for image generation Van Den Oord et al. (2017). After the success of VQ-VAE, attention to VQ increased, and it began to be used more in the domains of image and speech. Various modifications were applied to it based on the application and the specific problem. Residual vector quantizationZeghidour et al. (2021), initializing the codebook by the means Zeghidour et al. (2021), having the codebook in a lower dimension Yu et al. (2021), orthogonal regularization loss on codebook Shin et al. (2023), multi-head vector quantizationMama et al. (2021) and expiring stale codes Zeghidour et al. (2021) are some of this works. In this work, we used expiring stale codes to make better use of the codebook and to avoid falling into collapse.

3 Method

In traditional research methodologies for investigating emergent language, a significant obstacle is the inability to propagate gradients from the receiver back to the sender during language acquisition and development processes. Although techniques such as REINFORCE and Gumbel–Softmax offer partial solutions, they do not fully model realistic communication environments. Moreover, both approaches rely on approximate gradient estimates for the sender, which can negatively affect optimization and result in reduced accuracy.

Motivated by these limitations, this study aims to investigate the feasibility of enabling an agent to autonomously invent and develop a language without requiring interaction with another agent. This is pursued through mechanisms such as internal game-play or self-dialogue. Our proposed solution must meet two essential criteria: First, it should facilitate gradient propagation within the agent itself, distinguishing it from traditional two-agent systems where effective gradient transmission is hindered. Second, the agent must engage in self-interaction using structured linguistic formats, specifically discrete representations, to ensure these representations can be seamlessly employed when interfacing with another agent.

Drawing inspiration from vector quantization, we focused on embedding this mechanism within the agent, targeting two primary objectives. The first objective is to derive discrete representations conducive to effective gradient-based training during internal dialogue within the agent. The second objective is to utilize these discrete representations as the basis for generating symbols that can be directly applied in interactions with external agents.

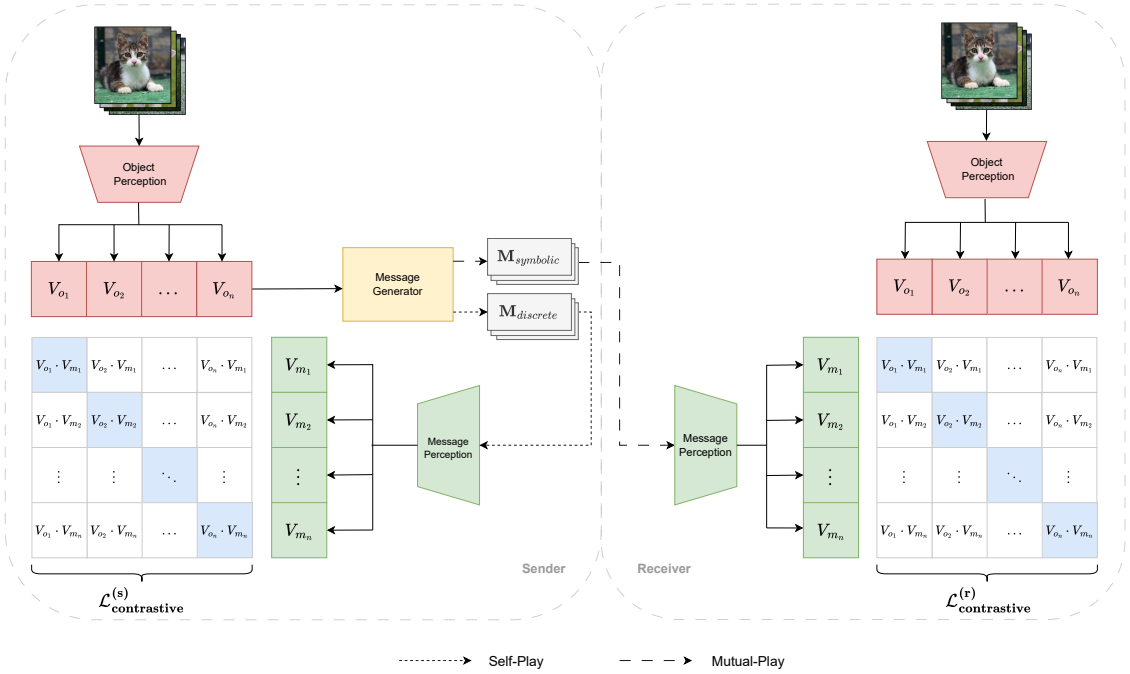

Figure 1: Overview of self-play and mutual play. In self-play, the sender independently develops a symbolic language using a contrastive loss. The sender then interacts with the receiver in mutual play, where the self-developed language is refined through communication using the same contrastive loss.

3.1 Architecture

In our proposed approach, both agents have a similar architecture. Depending on whether they act as a sender or a receiver, they employ different components of this architecture. The architecture of each agent consists of three parts: the **Object Perception Module**, the **Message Generation Module**, and the **Message Perception Module**.

3.1.1 Object Perception Module

The Object Perception Module, parameterized by a neural network f_θ, maps an input object o to a continuous vector representation $\mathbf{v}_o \in \mathbb{R}^d$. Formally, this mapping is defined inline as $\mathbf{v}_o = f_\theta(o)$, where f_θ serves as the feature encoder for the environment objects.

3.1.2 Message Generation Module

This module takes $\mathbf{v}_o$, the output of the Object Perception Module for object o, and returns a sequence of symbols $\mathbf{M}_{\text{symbolic}} = w_1 w_1 \dots w_L$, where $w_i \in \mathcal{V}$ and $\mathcal{V}$ is the vocabulary of possible symbols (i.e. semantic units or words). Additionally, when used in a self-play scenario, this module outputs a sequence of embedding vectors $\mathbf{M}_{\text{discrete}} = \mathbf{e}_{w_1} \mathbf{e}_{w_1} \dots \mathbf{e}_{w_L}$ corresponding to the sequence of symbols. This module internally consists of a recurrent neural network and a vector quantization mechanism. This module operates as described in Algorithm 1.

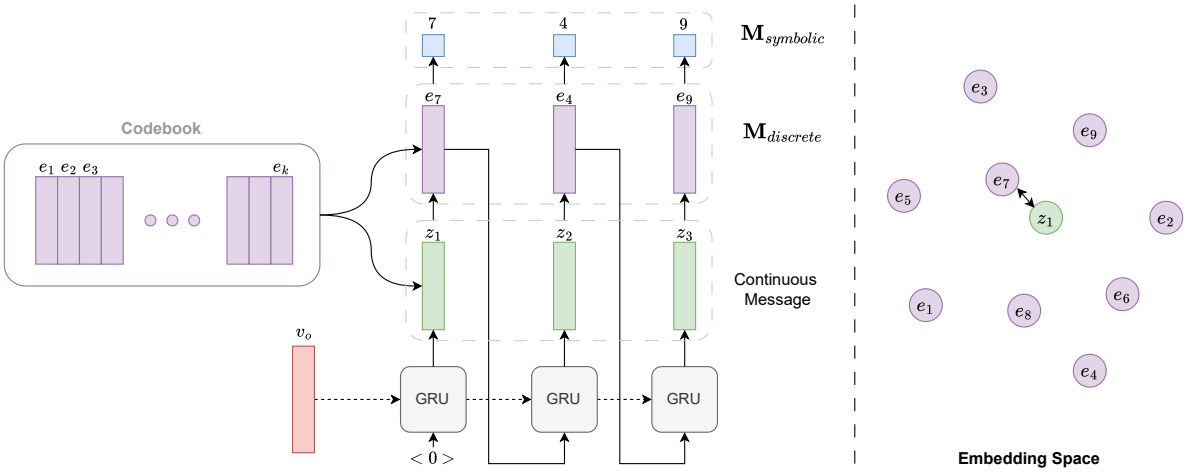

Figure 2: **Left:** Overview of the message generator and the corresponding messages produced during self-play and mutual play. **Right:** Visualization of the embedding space, where the GRU output $\mathbf{z}_1$ is mapped to its nearest codebook embedding $\mathbf{e}_7$.

Algorithm 1 Message Generation Module

Input: $\mathbf{v}_o \in \mathbb{R}^d$; L: message length
Module params: $\mathbf{e}_k \in \mathbb{R}^d, k \in 1, 2, \ldots, K$: codebook; `RNN`; g: linear projection
  $\mathbf{h}_0 = \mathbf{v}_o$
  for $t = 1 \ldots L$
    $\mathbf{h}_t = \texttt{RNN}(\mathbf{h}_{t-1}, \texttt{last\_word})$
    $\mathbf{z}_t = g(\mathbf{h}_t)$ # linear projection of the hidden state layer
    $w_t = \arg\min_k \|\mathbf{z}_t - \mathbf{e}_k\|^2$ # hard assignment; see Equation 1 for soft sampling
    $\texttt{last\_word} = \mathbf{e}_{w_t}$
  $\mathbf{M}_{\text{discrete}} = \{\mathbf{e}_{w_1}, \mathbf{e}_{w_2}, \ldots, \mathbf{e}_{w_L}\}$
  $\mathbf{M}_{\text{symbolic}} = \{w_1, w_2, \ldots, w_L\}$
  return $\mathbf{M}_{\text{discrete}}, \mathbf{M}_{\text{symbolic}}$

Alternatively, we can sample w_t from a probability distribution over codebook embeddings, for example using softmax probabilities:

$$P(w_t = k) = \frac{\exp(-\|\mathbf{z}_t - \mathbf{e}_k\|^2 / \tau)}{\sum_{i=1}^{K} \exp(-\|\mathbf{z}_t - \mathbf{e}_i\|^2 / \tau)}, \tag{1}$$

where τ is a temperature parameter.

3.1.3 Message Perception Module

In the mutual-play scenario, this module takes a sequence of symbols $\mathbf{M}_{\text{symbolic}}$ as input and produces a vector representation $\mathbf{v}_m$ for that sequence. When used in a self-play scenario, it takes a sequence of vectors $\mathbf{M}_{\text{discrete}}$ as input and produces a vector representation $\mathbf{v}_m$. Structurally, this module consists of an embedding function f_ω and a recurrent neural network f_ϕ:

$$\mathbf{v}_m = \begin{cases} f_\phi(\mathbf{e}_{w_1}, \ldots, \mathbf{e}_{w_L}), & \text{if } \mathbf{M} \text{ is discrete} \\ f_\phi(f_\omega(w_1), \ldots, f_\omega(w_L)), & \text{if } \mathbf{M} \text{ is symbolic} \end{cases} \tag{2}$$

where f_ω maps discrete symbols to dense vectors, and f_ϕ processes the sequence to form the final message representation. As detailed above, both the Message Generation and Message Perception networks accept

dual inputs/outputs: discrete representations (vectors drawn from the learned codebook) used during self-play, and symbolic representations (explicit symbol indices) used during mutual play.

3.2 Training Algorithm

We have two types of learning processes: **Self-Play** and **Mutual-Play**.

3.2.1 Self-Play

In this setting, a single agent plays the referential game with itself, as illustrated for the sender agent in Figure 1. As discussed later, the sender or the receiver can perform the self-play themselves; therefore, we use the superscript a to denote the agent in the following equations. The self-play scenario proceeds as follows.

1. Encode the object to obtain its representation $\mathbf{v}_o^{(a)}$.
2. Generate the discrete message embeddings $\mathbf{M}_{\text{discrete}}$ from $\mathbf{v}_o^{(a)}$ (Algorithm 1).
3. Encode $\mathbf{M}_{\text{discrete}}$ to obtain the representation of whole message $\mathbf{v}_m^{(a)}$ (Equation 2).
4. Compute the loss function. We use a Contrastive Loss similar to the CLIP loss, defined as:

$$\mathcal{L}_{\text{contrastive}}^{(a)} = -\log \frac{\exp\left(\text{sim}(\mathbf{v}_m^{(a)}, \mathbf{v}_o^{(a)})\right)}{\sum_{o' \in \mathcal{O}} \exp\left(\text{sim}(\mathbf{v}_m^{(a)}, \mathbf{v}_{o'}^{(a)})\right)}, \tag{3}$$

where $\text{sim}(\cdot, \cdot)$ is a similarity function (e.g., the dot product), and $\mathcal{O}$ is the set of objects including the target object and distractors.

We also have a loss related to commitment in vector quantization, which is calculated as follows:

$$\mathcal{L}_{\text{commitment}} = \|\mathbf{z}_t - \text{sg}[\mathbf{e}_{w_t}]\|_2^2, \tag{4}$$

where sg stands for the stopgradient operator. The final self-play loss is given by:

$$\mathcal{L}_{\text{self-play}}^{(a)} = \mathcal{L}_{\text{contrastive}}^{(a)} + \beta \mathcal{L}_{\text{commitment}}, \tag{5}$$

where β acts as the weighting factor for the commitment loss, allowing it to be scaled appropriately relative to the contrastive loss.

Since self-play operates over discrete word embeddings (rather than the discrete symbols themselves), we can leverage the straight-through estimator (Van Den Oord et al., 2017) to copy gradients from the discrete representation to its continuous counterpart. This design preserves end-to-end differentiability, allowing all three agent modules to be optimized jointly through backpropagation of $\mathcal{L}_{\text{self-play}}^{(a)}$. The codebook embeddings $\mathcal{C}$ are updated using an exponential moving average procedure following Van Den Oord et al. (2017).

3.2.2 Mutual-Play

Upon entering the mutual-play phase, the sender possesses a highly structured language developed during self-play. In principle, this foundation is robust enough that we could strictly freeze the sender's parameters, requiring only the receiver to align with this established vocabulary via standard backpropagation. However, to foster dynamic co-adaptation between the agents, we instead choose to fine-tune the sender based on the receiver's feedback. Because the physical transmission of discrete symbols creates a non-differentiable bottleneck, this fine-tuning necessitates policy gradient methods. Crucially, by initializing the sender with the grounded vocabulary from VQ self-play, we shift REINFORCE from an unstable, high-variance engine that typically struggles to learn from scratch, to a highly stable fine-tuning mechanism operating over an informative discrete prior.

Formally, the mutual-play referential game involves two agents, one acting as the sender (s) and the other as the receiver (r), and proceeds as follows:

1. **Sender Agent**:
   (a) Encodes the target object to obtain its vector representation $\mathbf{v}_o^{(s)}$.
   (b) Generates the discrete sequence of symbols $\mathbf{M}_{\text{symbolic}}$, and transmits it to the receiver agent over the communication channel. During this step, the commitment loss is also calculated according to Equation 4.
2. **Receiver Agent**:
   (a) Encodes the received message $\mathbf{M}_{\text{symbolic}}$ to obtain the message embedding $\mathbf{v}_m^{(r)}$.
   (b) Encodes the target object and all distractors to obtain their vector representations $\mathbf{v}_{o'}^{(r)}$, $\forall o' \in \mathcal{O}$.
   (c) Computes $\mathcal{L}_{\text{contrastive}}^{(r)}$, the contrastive loss for the receiver agent, as defined in Equation 3.

The parameters of the receiver agent's modules (the Message Perception Module and the Object Perception Module) are fully differentiable with respect to the task objective and are updated directly by backpropagating the gradients from $\mathcal{L}_{\text{contrastive}}^{(r)}$.

To fine-tune the sender agent, we apply the REINFORCE algorithm to update its generative policy:

$$\mathcal{L}_{\text{RL}} = -R \sum_{t=1}^{L} \log P(w_t \mid \mathbf{h}_t), \tag{6}$$

where $P(w_t \mid \mathbf{h}_t)$ is the probabilistic sampling distribution defined in Equation 1. The reward R is derived from the receiver's performance and is formalized as $R = -\mathcal{L}_{\text{contrastive}}^{(r)}$.

Finally, the overall loss for the mutual-play phase is defined as:

$$\mathcal{L}_{\text{mutual-play}} = \mathcal{L}_{\text{contrastive}}^{(r)} + \mathcal{L}_{\text{RL}} + \beta\, \mathcal{L}_{\text{commitment}}, \tag{7}$$

where the first term optimizes the receiver, and the combination of the policy gradient and commitment loss fine-tunes the sender. Ultimately, by performing the end-to-end self-play game prior to mutual play, the system constructs a robust internal communication foundation that greatly simplifies the complexities of multi-agent alignment.

4 Experimental Setup

4.1 Datasets

Synthetic Objects. This dataset is based on EGG's object game (Kharitonov et al., 2019), and designed to cover the full space of categorical attribute combinations. It contains 10,000 unique objects, each defined by four categorical attributes with ten possible values each. Objects are represented as 40-dimensional vectors formed by concatenating four one-hot encodings. A key challenge of this dataset is that the inputs are discrete, in contrast to datasets with continuous or visual representations.

ShapeWorld. This dataset consists of synthetic images of single geometric objects rendered in different colors on a black background (Kuhnle & Copestake, 2017). It enables explicit control over compositional structure. In our setup, the training and test splits differ in compositionality: some color–shape combinations are seen during training, while others are held out and appear only at test time.

DSprites. It is a synthetic dataset for studying disentangled representations (Matthey et al., 2017). It includes 737,280 black-and-white 64×64 images generated by varying five latent factors: shape, scale, rotation, and x- and y-position. Its explicitly structured latent space makes it well suited for evaluating disentanglement through emergent communication.

CelebA. This dataset contains 202,599 face images of size 178×218 pixels from 10,177 identities, each annotated with 40 binary facial attributes (Liu et al., 2015). Compared to the synthetic datasets, CelebA introduces the additional complexity of natural image data.

4.2 Variations of the Proposed Model

Sender Self-Play. In this version, the sender first undergoes self-play before interacting with the receiver in the mutual-play. During the mutual-play, the sender can either be frozen, fine-tuned using REINFORCE (RL) (see Equation 6), or fine-tuned using both the REINFORCE and self-play objectives (see Equation 5) (RL+Pres). In the first scenario, the sender's language remains fixed; in the second, it is optimized for communication; and in the third, it improves for communication while attempting to preserve its original language.

Sender and Receiver Self-Play. We designed an experiment in which each agent first invents its own language during the self-play and then communicates with the other agent in the mutual-play to converge to a shared language. To encourage the development of different languages during self-play, agents are initialized with different seeds.

Receiver Self-Play. We also conducted a receiver self-play experiment, in which the receiver first undergoes self-play before interacting with the sender in mutual-play. The results of this experiment are reported in Appendix B.

4.3 Baselines

We compare VQEL against two commonly used baselines for emergent communication in referential games: REINFORCE (Williams, 1992) and GS (Jang et al., 2017b; Maddison et al., 2016). To maintain backwards differentiability in GS, we use the straight-through (ST) trick (Havrylov & Titov, 2017) with learning the inverse-temperature with a multilayer perceptron (Havrylov & Titov, 2017):

$$\frac{1}{\tau(h_i)} = \log(1 + \exp(\mathbf{w}_\tau^T \mathbf{h}_i)) + \tau_0 \tag{8}$$

where τ_0 controls maximum possible value for the temperature.

4.4 Agents' Architecture

The Object Perception Module is a simple embedding layer for the Objects dataset, and a simple CNN encoder architecture adopted from Prototypical Networks (Snell et al., 2017) for ShapeWorld and dSprites. In contrast, for CelebA we use a small pretrained DINOv2 network (Oquab et al., 2023) with a linear layer on top. The DINOv2 network's parameters are frozen during training, and only the linear layer is trained.

The Message Generation Module comprises a GRU and a VQ module. In the VQ module, we use cosine similarity as the distance metric. We observe that constraining the code vectors to lie on a hypersphere leads to improved communication success. For completeness, we also report results obtained using Euclidean distance in Appendix A.

The Message Perception Module consists of a GRU and an embedding layer, which converts symbolic messages into embeddings before passing them through the GRU.

All three methods share the same overall architecture, except that GS-ST and REINFORCE do not include VQ module. The number of parameters is identical across all methods.

4.5 Evaluation Metrics

We evaluate all experiments using four metrics. **Accuracy (ACC)** measures communication success as the fraction of correctly identified target objects. **Active Words (AW)** (Lazaridou et al., 2016) represents

the fraction of the vocabulary that is used at least once during communication. **Topographic Similarity (TopSim)** (Brighton & Kirby, 2006; Lazaridou et al., 2018b) measures the structural alignment between attribute representations and generated messages, computed as the correlation between Hamming distances in the attribute space and message space. **Entropy of the concept given the message, $\mathbf{H}(\mathbf{C} \mid \mathbf{M})$** (Rita et al., 2021), measures the uncertainty over concepts conditioned on the received message.

4.6 Training Details

In each experiment, the dataset is split into training, validation, and test sets with proportions of 80%, 10%, and 10%, respectively. All methods use the Adam optimizer with a weight decay of 1×10^{-5}. The learning rate is tuned by searching the range $[10^{-6}, 10^{-3}]$ with a step size of 0.1. The sampling temperature is optimized over the range $[10^{-5}, 1]$ with a step size of 0.1, and τ_0 is tuned by selecting the best value from $[0.1, 1.5]$ with a step size of 0.1.

Baseline methods are trained for 100 epochs, while VQEL is trained for 50 epochs in the self-play phase, followed by an additional 50 epochs in mutual play. The vocabulary size is set to 10, and the message length L is 4 in all experiments. During training, agents are trained with a batch size of 32 (corresponding to 31 distractors), whereas at test time, the main results are reported using a batch size of 100. Additionally, we evaluate methods under different batch sizes to analyze their robustness.

5 Results

For simplicity, in the following tables we denote self-play and mutual play as SP and MP, respectively. The agent performing self-play is indicated in subscript; for example, SP_S refers to sender self-play. We use the + notation to indicate sequential training phases (e.g., SP_S + MP denotes sender self-play followed by mutual play). Entries denoted solely as SP (e.g., VQEL-SP_S) report the performance of the agent's internal language established during the self-play phase, prior to any mutual interaction.

To preserve comparability, the GS-ST and REINFORCE results are copied from Table 1 into the subsequent tables. In these tables, the highest accuracy for each setting is highlighted in bold, and when relevant, the second-highest accuracy is also indicated to facilitate comparison between the most competitive results.

5.1 Sender Self-Play

The results are shown in Table 1. Across all datasets except CelebA, the accuracy of all three modes exceeds that of both REINFORCE and GS-ST. Moreover, allowing the sender to improve its language during communication consistently increases accuracy. For CelebA, REINFORCE achieves the best performance, likely due to the use of pretrained encoders and the limited portion of the model that can take advantage of our method. However, as shown in Section 5.2, there exists another scenario in which our method even outperforms REINFORCE on CelebA. As shown in Figure 3, VQEL demonstrates greater robustness to an increasing number of distractors during testing, with a significantly smaller decline in accuracy relative to the baseline methods.

Interestingly, although the vocabulary size is limited to 10, both GS-ST and REINFORCE fail to utilize the full vocabulary (AW < 1). In contrast, VQEL fully exploits 100% of the vocabulary across all datasets and modes. Additionally, our method yields substantially lower entropy, demonstrating more efficient message encoding.

As previous works (Yao et al., 2022; Chaabouni et al., 2021b) have shown no clear relationship between accuracy and TopSim metric, we observe that VQEL can sometimes improve TopSim, while in other cases it remains comparable to REINFORCE.

Finally, as illustrated in Figure 4, which depicts the number of unique messages generated by different methods across datasets, Our method more effectively utilizes channel capacity. Consequently, it generates a higher number of unique messages compared to both REINFORCE and GS-ST.

Dataset	Method	Sender Update	ACC ↑	AW ↑	TopSim ↑	H(C|M) ↓
Objects	GS-ST	-	$0.78_{\pm 0.01}$	$0.93_{\pm 0.06}$	$0.21_{\pm 0.02}$	$1.04_{\pm 0.03}$
	REINFORCE	-	$0.51_{\pm 0.21}$	$0.47_{\pm 0.12}$	$0.14_{\pm 0.07}$	$2.21_{\pm 1.28}$
	VQEL-SPS	-	$0.82_{\pm 0.01}$	$1.00_{\pm 0.00}$	$0.19_{\pm 0.01}$	$0.12_{\pm 0.02}$
	VQEL-SPS+MP	Frozen	$0.85_{\pm 0.01}$	$1.00_{\pm 0.00}$	$0.19_{\pm 0.01}$	$0.12_{\pm 0.02}$
	VQEL-SPS+MP	RL	$\mathbf{0.86_{\pm 0.01}}$	$1.00_{\pm 0.00}$	$0.19_{\pm 0.01}$	$0.12_{\pm 0.02}$
	VQEL-SPS+MP	RL+Pres	$\mathbf{0.86_{\pm 0.01}}$	$1.00_{\pm 0.00}$	$0.19_{\pm 0.01}$	$0.12_{\pm 0.02}$
Shape	GS-ST	-	$0.82_{\pm 0.01}$	$1.00_{\pm 0.00}$	$0.02_{\pm 0.01}$	$1.21_{\pm 0.11}$
	REINFORCE	-	$0.86_{\pm 0.00}$	$0.83_{\pm 0.15}$	$0.01_{\pm 0.01}$	$0.88_{\pm 0.11}$
	VQEL-SPS	-	$0.87_{\pm 0.01}$	$1.00_{\pm 0.00}$	$0.05_{\pm 0.03}$	$0.38_{\pm 0.10}$
	VQEL-SPS+MP	Frozen	$0.89_{\pm 0.01}$	$1.00_{\pm 0.00}$	$0.05_{\pm 0.03}$	$0.38_{\pm 0.10}$
	VQEL-SPS+MP	RL	$\mathbf{0.91_{\pm 0.01}}$	$1.00_{\pm 0.00}$	$0.05_{\pm 0.02}$	$0.39_{\pm 0.10}$
	VQEL-SPS+MP	RL+Pres	$0.91_{\pm 0.02}$	$1.00_{\pm 0.00}$	$0.05_{\pm 0.03}$	$0.39_{\pm 0.10}$
DSprites	GS-ST	-	$0.81_{\pm 0.01}$	$0.90_{\pm 0.00}$	$0.10_{\pm 0.01}$	$1.80_{\pm 0.05}$
	REINFORCE	-	$0.88_{\pm 0.02}$	$0.80_{\pm 0.10}$	$0.06_{\pm 0.00}$	$1.06_{\pm 0.13}$
	VQEL-SPS	-	$0.91_{\pm 0.02}$	$1.00_{\pm 0.00}$	$0.07_{\pm 0.01}$	$0.40_{\pm 0.02}$
	VQEL-SPS+MP	Frozen	$0.92_{\pm 0.01}$	$1.00_{\pm 0.00}$	$0.07_{\pm 0.01}$	$0.40_{\pm 0.02}$
	VQEL-SPS+MP	RL	$\mathbf{0.93_{\pm 0.01}}$	$1.00_{\pm 0.00}$	$0.07_{\pm 0.01}$	$0.40_{\pm 0.01}$
	VQEL-SPS+MP	RL+Pres	$\mathbf{0.93_{\pm 0.01}}$	$1.00_{\pm 0.00}$	$0.07_{\pm 0.01}$	$0.40_{\pm 0.02}$
CelebA	GS-ST	-	$0.90_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.14_{\pm 0.01}$	$1.01_{\pm 0.08}$
	REINFORCE	-	$\mathbf{0.93_{\pm 0.01}}$	$1.00_{\pm 0.00}$	$0.11_{\pm 0.03}$	$0.90_{\pm 0.06}$
	VQEL-SPS	-	$0.89_{\pm 0.01}$	$1.00_{\pm 0.00}$	$0.10_{\pm 0.04}$	$0.58_{\pm 0.10}$
	VQEL-SPS+MP	Frozen	$0.90_{\pm 0.01}$	$1.00_{\pm 0.00}$	$0.10_{\pm 0.04}$	$0.58_{\pm 0.10}$
	VQEL-SPS+MP	RL	$0.91_{\pm 0.01}$	$1.00_{\pm 0.00}$	$0.10_{\pm 0.04}$	$0.57_{\pm 0.09}$
	VQEL-SPS+MP	RL+Pres	$0.91_{\pm 0.01}$	$1.00_{\pm 0.00}$	$0.10_{\pm 0.04}$	$0.56_{\pm 0.09}$

Table 1: Performance comparison across datasets and evaluation metrics for the sender self-play game.

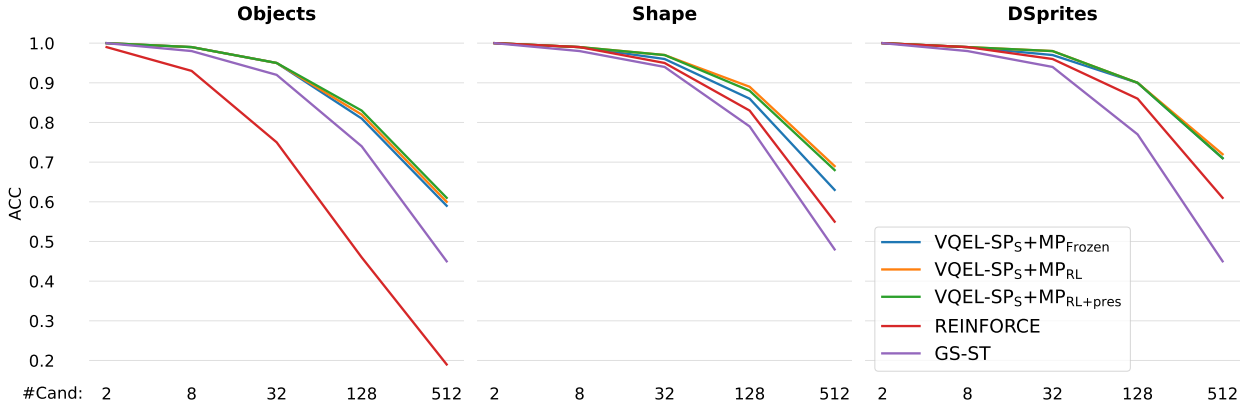

Figure 3: Accuracy of the sender self-play game compared to baseline methods for varying numbers of candidates at test time.

5.2 Sender and Receiver Self-Play

The results for this game, shown in Table 2, demonstrate that our method outperforms both REINFORCE and GS-ST across all four datasets. Notably, for the Objects and CelebA datasets, it also surpasses VQEL in the Sender Self-Play scenario (Section 5.1).

Furthermore, Figure 5 illustrates that VQEL maintains higher accuracy than the baselines as the number of test-time candidates increases, exhibiting a smaller performance drop.

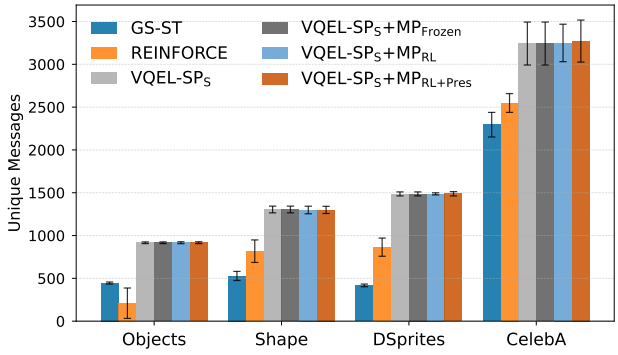

Figure 4: Comparison of the number of unique messages produced by VQEL, during the sender self-play game, and baseline models.

Dataset	Method	ACC ↑	AW ↑	TopSim ↑	H(C|M) ↓
OBJECTS	GS-ST	$0.78_{\pm 0.01}$	$0.93_{\pm 0.06}$	$0.21_{\pm 0.02}$	$1.04_{\pm 0.03}$
	REINFORCE	$0.51_{\pm 0.21}$	$0.47_{\pm 0.12}$	$0.14_{\pm 0.07}$	$2.21_{\pm 1.28}$
	VQEL-SPS	$0.82_{\pm 0.01}$	$1.00_{\pm 0.00}$	$0.19_{\pm 0.01}$	$0.12_{\pm 0.02}$
	VQEL-SPR	$0.81_{\pm 0.01}$	$1.00_{\pm 0.00}$	$0.19_{\pm 0.01}$	$0.12_{\pm 0.00}$
	VQEL-SPS,R+MP	$\mathbf{0.90_{\pm 0.01}}$	$1.00_{\pm 0.00}$	$0.17_{\pm 0.02}$	$0.14_{\pm 0.02}$
SHAPE	GS-ST	$0.82_{\pm 0.01}$	$1.00_{\pm 0.00}$	$0.02_{\pm 0.01}$	$1.21_{\pm 0.11}$
	REINFORCE	$0.86_{\pm 0.00}$	$0.83_{\pm 0.15}$	$0.01_{\pm 0.01}$	$0.88_{\pm 0.11}$
	VQEL-SPS	$0.87_{\pm 0.01}$	$1.00_{\pm 0.00}$	$0.05_{\pm 0.03}$	$0.38_{\pm 0.10}$
	VQEL-SPR	$0.87_{\pm 0.01}$	$1.00_{\pm 0.00}$	$0.05_{\pm 0.03}$	$0.38_{\pm 0.10}$
	VQEL-SPS,R+MP	$\mathbf{0.91_{\pm 0.00}}$	$1.00_{\pm 0.00}$	$0.04_{\pm 0.01}$	$0.42_{\pm 0.02}$
DSPRITES	GS-ST	$0.81_{\pm 0.01}$	$0.90_{\pm 0.00}$	$0.10_{\pm 0.01}$	$1.80_{\pm 0.05}$
	REINFORCE	$0.88_{\pm 0.02}$	$0.80_{\pm 0.10}$	$0.06_{\pm 0.00}$	$1.06_{\pm 0.13}$
	VQEL-SPS	$0.91_{\pm 0.02}$	$1.00_{\pm 0.00}$	$0.07_{\pm 0.01}$	$0.40_{\pm 0.02}$
	VQEL-SPR	$0.90_{\pm 0.01}$	$1.00_{\pm 0.00}$	$0.07_{\pm 0.01}$	$0.38_{\pm 0.03}$
	VQEL-SPS,R+MP	$\mathbf{0.92_{\pm 0.01}}$	$1.00_{\pm 0.00}$	$0.09_{\pm 0.00}$	$0.45_{\pm 0.04}$
CELEBA	GS-ST	$0.90_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.14_{\pm 0.01}$	$1.01_{\pm 0.08}$
	REINFORCE	$0.93_{\pm 0.01}$	$1.00_{\pm 0.00}$	$0.11_{\pm 0.03}$	$0.90_{\pm 0.06}$
	VQEL-SPS	$0.89_{\pm 0.01}$	$1.00_{\pm 0.00}$	$0.10_{\pm 0.04}$	$0.58_{\pm 0.10}$
	VQEL-SPR	$0.89_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.11_{\pm 0.04}$	$0.58_{\pm 0.12}$
	VQEL-SPS,R+MP	$\mathbf{0.94_{\pm 0.00}}$	$1.00_{\pm 0.00}$	$0.11_{\pm 0.01}$	$0.53_{\pm 0.07}$

Table 2: Performance comparison across datasets and evaluation metrics for the sender and receiver self-play game.

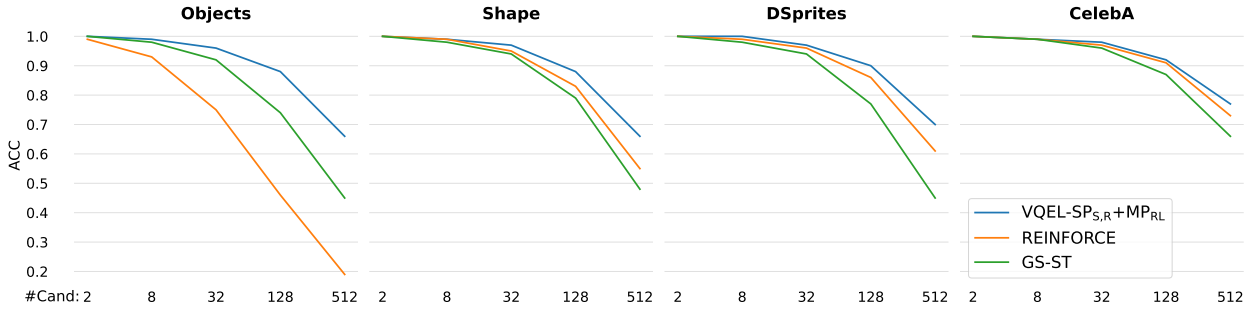

Figure 5: Accuracy of the sender and receiver self-play game compared to baseline methods for varying numbers of candidates at test time.

5.3 Effect of Vector Quantization

The improvements in communication observed in the previous experiments raise an important question: are these gains due to the self-play technique, or are they primarily the result of using vector quantization in the message generator? To investigate this, we designed an experiment in which the model is trained for the full 100 epochs in the mutual-play scenario without any self-play. As shown in Table 3, removing self-play leads to a significant drop in accuracy, highlighting the crucial role of inventing a symbolic language during the self-play phase.

Dataset	Method	Sender Update	ACC ↑	AW ↑	TopSim ↑	H(C|M) ↓
OBJECTS	GS-ST	-	$0.78_{\pm 0.01}$	$0.93_{\pm 0.06}$	$0.21_{\pm 0.02}$	$1.04_{\pm 0.03}$
	REINFORCE	-	$0.51_{\pm 0.21}$	$0.47_{\pm 0.12}$	$0.14_{\pm 0.07}$	$2.21_{\pm 1.28}$
	VQEL-SP$_S$+MP	RL	$\mathbf{0.86_{\pm 0.01}}$	$1.00_{\pm 0.00}$	$0.19_{\pm 0.01}$	$0.12_{\pm 0.02}$
	VQEL-MP	RL	$0.28_{\pm 0.18}$	$1.00_{\pm 0.00}$	$0.22_{\pm 0.04}$	$2.64_{\pm 0.32}$
SHAPE	GS-ST	-	$0.82_{\pm 0.01}$	$1.00_{\pm 0.00}$	$0.02_{\pm 0.01}$	$1.21_{\pm 0.11}$
	REINFORCE	-	$0.86_{\pm 0.00}$	$0.83_{\pm 0.15}$	$0.01_{\pm 0.01}$	$0.88_{\pm 0.11}$
	VQEL-SP$_S$+MP	RL	$\mathbf{0.91_{\pm 0.01}}$	$1.00_{\pm 0.00}$	$0.05_{\pm 0.02}$	$0.39_{\pm 0.10}$
	VQEL-MP	RL	$0.88_{\pm 0.01}$	$1.00_{\pm 0.00}$	$0.01_{\pm 0.00}$	$0.67_{\pm 0.18}$
DSPRITES	GS-ST	-	$0.81_{\pm 0.01}$	$0.90_{\pm 0.00}$	$0.10_{\pm 0.01}$	$1.80_{\pm 0.05}$
	REINFORCE	-	$0.88_{\pm 0.02}$	$0.80_{\pm 0.10}$	$0.06_{\pm 0.00}$	$1.06_{\pm 0.13}$
	VQEL-SP$_S$+MP	RL	$\mathbf{0.93_{\pm 0.01}}$	$1.00_{\pm 0.00}$	$0.07_{\pm 0.01}$	$0.40_{\pm 0.01}$
	VQEL-MP	RL	$0.88_{\pm 0.02}$	$1.00_{\pm 0.00}$	$0.05_{\pm 0.01}$	$0.63_{\pm 0.09}$
CELEBA	GS-ST	-	$0.90_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.14_{\pm 0.01}$	$1.01_{\pm 0.08}$
	REINFORCE	-	$\mathbf{0.93_{\pm 0.01}}$	$1.00_{\pm 0.00}$	$0.11_{\pm 0.03}$	$0.90_{\pm 0.06}$
	VQEL-SP$_S$+MP	RL	$\underline{0.91_{\pm 0.01}}$	$1.00_{\pm 0.00}$	$0.10_{\pm 0.04}$	$0.57_{\pm 0.09}$
	VQEL-MP	RL	$0.87_{\pm 0.03}$	$1.00_{\pm 0.00}$	$0.13_{\pm 0.02}$	$0.74_{\pm 0.10}$

Table 3: Effect of self-play on the agents' communication performance.

6 Conclusion

In this work, we address the optimization challenges of discrete communication by proposing Vector Quantized Emergent Language (VQEL). Inspired by the cognitive role of "inner speech," VQEL employs Vector Quantization to bootstrap language through differentiable self-play, effectively bridging private cognition and social communication. This approach allows agents to optimize internal representations via a learned codebook, providing a robust initialization for subsequent multi-agent interaction.

Our empirical evaluation across four diverse datasets—Synthetic Objects, ShapeWorld, dSprites, and CelebA, demonstrates the efficacy of this approach. We observed that:

- **Performance and Stability:** Agents pre-trained with VQEL self-play consistently outperform or match strong baselines (REINFORCE and Gumbel-Softmax) in terms of communication accuracy. VQEL exhibits superior stability, particularly as the number of distractors increases.
- **Vocabulary Efficiency:** Unlike baseline methods, which often suffer from vocabulary collapse, VQEL utilizes the available channel capacity fully (100% active words) and achieves lower entropy in concept-message mapping, indicating more precise communication.
- **The Necessity of Self-Play:** Our ablation studies confirm that the performance gains are not merely due to the VQ architecture but are driven by the self-play phase. Removing the self-play pre-training significantly degrades task success, validating the hypothesis that intrapersonal concept stabilization is a precursor to effective interpersonal communication.

These findings suggest that self-play provides a smoother optimization landscape for emergent language than trying to learn discrete protocols from scratch in a multi-agent setting.

References

Ben Alderson-Day and Charles Fernyhough. Inner speech: development, cognitive functions, phenomenology, and neurobiology. *Psychological bulletin*, 141(5):931, 2015.

Timothée Bernard and Timothee Mickus. So many design choices: Improving and interpreting neural agent communication in signaling games. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 8399–8413, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.531. URL https://aclanthology.org/2023.findings-acl.531/.

Diane Bouchacourt and Marco Baroni. How agents see things: On visual representations in an emergent language game. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 981–985, 2018.

Nicolo' Brandizzi. Toward more human-like ai communication: A review of emergent communication research. *IEEE Access*, 11:142317–142340, 2023.

Nicolo' Brandizzi, Davide Grossi, and Luca Iocchi. Rlupus: Cooperation through emergent communication in the werewolf social deduction game. *Intelligenza Artificiale*, 15(2):55–70, 2022.

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

Boaz Carmeli, Ron Meir, and Yonatan Belinkov. Emergent quantized communication. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(10):11533–11541, Jun. 2023. doi: 10.1609/aaai.v37i10.26363. URL https://ojs.aaai.org/index.php/AAAI/article/view/26363.

Rahma Chaabouni, Eugene Kharitonov, Alessandro Lazaric, Emmanuel Dupoux, and Marco Baroni. Word-order biases in deep-agent emergent communication. *arXiv preprint arXiv:1905.12330*, 2019.

Rahma Chaabouni, Eugene Kharitonov, Diane Bouchacourt, Emmanuel Dupoux, and Marco Baroni. Compositionality and generalization in emergent languages. *arXiv preprint arXiv:2004.09124*, 2020.

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.

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*, 2021c.

Abhishek Das, Satwik Kottur, José MF Moura, Stefan Lee, and Dhruv Batra. Learning cooperative visual dialog agents with deep reinforcement learning. In *Proceedings of the IEEE international conference on computer vision*, pp. 2951–2960, 2017.

Tom Eccles, Yoram Bachrach, Guy Lever, Angeliki Lazaridou, and Thore Graepel. Biases for emergent communication in multi-agent reinforcement learning. *Advances in neural information processing systems*, 32, 2019.

Katrina Evtimova, Andrew Drozdov, Douwe Kiela, and Kyunghyun Cho. Emergent communication in a multi-modal, multi-step referential game. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=rJGZq6g0-.

Nicole Fitzgerald. To populate is to regulate. *arXiv preprint arXiv:1911.04362*, 2019.

Jerry A Fodor. *The language of thought*, volume 5. Harvard university press, 1975.

Jakob N. Foerster, Yannis M. Assael, Nando de Freitas, and Shimon Whiteson. Learning to communicate with deep multi-agent reinforcement learning. *CoRR*, abs/1605.06676, 2016. URL `http://arxiv.org/abs/1605.06676`.

Laura Harding Graesser, Kyunghyun Cho, and Douwe Kiela. Emergent linguistic phenomena in multi-agent communication 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)*, pp. 3700–3710, 2019.

Serhii Havrylov and Ivan Titov. Emergence of language with multi-agent games: Learning to communicate with sequences of symbols. *Advances in neural information processing systems*, 30, 2017.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In *International Conference on Learning Representations*, 2017a. URL `https://openreview.net/forum?id=rkE3y85ee`.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In *International Conference on Learning Representations*, 2017b.

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

Eugene Kharitonov, Rahma Chaabouni, Diane Bouchacourt, and Marco Baroni. Egg: a toolkit for research on emergence of language in games. *arXiv preprint arXiv:1907.00852*, 2019.

Eugene Kharitonov, Rahma Chaabouni, Diane Bouchacourt, and Marco Baroni. Entropy minimization in emergent languages. In *International Conference on Machine Learning*, pp. 5220–5230. PMLR, 2020.

Alexander Kuhnle and Ann Copestake. Shapeworld-a new test methodology for multimodal language understanding. *arXiv preprint arXiv:1704.04517*, 2017.

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

Angeliki Lazaridou, Alexander Peysakhovich, and Marco Baroni. Multi-agent cooperation and the emergence of (natural) language. In *International Conference on Learning Representations*, 2017.

Angeliki Lazaridou, Karl Moritz Hermann, Karl Tuyls, and Stephen Clark. Emergence of linguistic communication from referential games with symbolic and pixel input. In *International Conference on Learning Representations*, 2018a.

Angeliki Lazaridou, Karl Moritz Hermann, Karl Tuyls, and Stephen Clark. Emergence of linguistic communication from referential games with symbolic and pixel input. In *International Conference on Learning Representations (ICLR)*, 2018b.

Angeliki Lazaridou, Anna Potapenko, and Olivier Tieleman. Multi-agent communication meets natural language: Synergies between functional and structural language learning. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 7663–7674, 2020.

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

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*, pp. 3730–3738, 2015.

Chris J Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. *arXiv preprint arXiv:1611.00712*, 2016.

Rayhane Mama, Marc S Tyndel, Hashiam Kadhim, Cole Clifford, and Ragavan Thurairatnam. Nwt: towards natural audio-to-video generation with representation learning. *arXiv preprint arXiv:2106.04283*, 2021.

Loic Matthey, Irina Higgins, Demis Hassabis, and Alexander Lerchner. dsprites: Disentanglement testing sprites dataset. https://github.com/deepmind/dsprites-dataset/, 2017.

Daniela Mihai and Jonathon Hare. Learning to draw: Emergent communication through sketching. *Advances in neural information processing systems*, 34:7153–7166, 2021.

Igor Mordatch and Pieter Abbeel. Emergence of grounded compositional language in multi-agent populations. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

Jannik Peters, Constantin Waubert de Puiseau, Hasan Tercan, Arya Gopikrishnan, Gustavo Adolpho Lucas de Carvalho, Christian Bitter, and Tobias Meisen. Emergent language: a survey and taxonomy. *Autonomous Agents and Multi-Agent Systems*, 39(1):1–73, 2025.

Yi Ren, Shangmin Guo, Matthieu Labeau, Shay B Cohen, and Simon Kirby. Compositional languages emerge in a neural iterated learning model. *arXiv preprint arXiv:2002.01365*, 2020.

Mathieu Rita, Florian Strub, Jean-Bastien Grill, Olivier Pietquin, and Emmanuel Dupoux. On the role of population heterogeneity in emergent communication. In *International Conference on Learning Representations*, 2021.

Woncheol Shin, Gyubok Lee, Jiyoung Lee, Eunyi Lyou, Joonseok Lee, and Edward Choi. Exploration into translation-equivariant image quantization. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2023.

Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. *Advances in neural information processing systems*, 30, 2017.

Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.

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

Shunyu Yao, Mo Yu, Yang Zhang, Karthik R Narasimhan, Joshua B Tenenbaum, and Chuang Gan. Linking emergent and natural languages via corpus transfer. *arXiv preprint arXiv:2203.13344*, 2022.

Jiahui Yu, Xin Li, Jing Yu Koh, Han Zhang, Ruoming Pang, James Qin, Alexander Ku, Yuanzhong Xu, Jason Baldridge, and Yonghui Wu. Vector-quantized image modeling with improved vqgan. *ArXiv*, abs/2110.04627, 2021. URL `https://api.semanticscholar.org/CorpusID:238582653`.

Neil Zeghidour, Alejandro Luebs, Ahmed Omran, Jan Skoglund, and Marco Tagliasacchi. Soundstream: An end-to-end neural audio codec. *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, 30:495–507, November 2021. ISSN 2329-9290. doi: 10.1109/TASLP.2021.3129994. URL `https://doi.org/10.1109/TASLP.2021.3129994`.

A Euclidean Distance in Codebook

Table 4 reports the results of the sender self-play game when Euclidean distance is used in the codebook to select the nearest embedding vector. Across datasets, this choice leads to a 2–6% drop in accuracy compared to cosine similarity.

Dataset	Method	Sender Update	ACC ↑	AW ↑	TopSim ↑	H(C|M) ↓
OBJECTS	GS-ST	-	$0.78_{\pm 0.01}$	$0.93_{\pm 0.06}$	$0.21_{\pm 0.02}$	$1.04_{\pm 0.03}$
	REINFORCE	-	$0.51_{\pm 0.21}$	$0.47_{\pm 0.12}$	$0.14_{\pm 0.07}$	$2.21_{\pm 1.28}$
	VQEL-SP$_S$	-	$0.75_{\pm 0.02}$	$1.00_{\pm 0.00}$	$0.17_{\pm 0.02}$	$0.30_{\pm 0.03}$
	VQEL-SP$_S$+MP	Frozen	$0.81_{\pm 0.02}$	$1.00_{\pm 0.00}$	$0.17_{\pm 0.02}$	$0.30_{\pm 0.03}$
	VQEL-SP$_S$+MP	RL	$0.84_{\pm 0.01}$	$1.00_{\pm 0.00}$	$0.17_{\pm 0.02}$	$0.29_{\pm 0.03}$
	VQEL-SP$_S$+MP	RL+Pres	$0.84_{\pm 0.01}$	$1.00_{\pm 0.00}$	$0.18_{\pm 0.01}$	$0.28_{\pm 0.03}$
SHAPE	GS-ST	-	$0.82_{\pm 0.01}$	$1.00_{\pm 0.00}$	$0.02_{\pm 0.01}$	$1.21_{\pm 0.11}$
	REINFORCE	-	$0.86_{\pm 0.00}$	$0.83_{\pm 0.15}$	$0.01_{\pm 0.01}$	$0.88_{\pm 0.11}$
	VQEL-SP$_S$	-	$0.77_{\pm 0.02}$	$1.00_{\pm 0.00}$	$0.05_{\pm 0.01}$	$0.64_{\pm 0.07}$
	VQEL-SP$_S$+MP	Frozen	$0.85_{\pm 0.02}$	$1.00_{\pm 0.00}$	$0.05_{\pm 0.01}$	$0.64_{\pm 0.07}$
	VQEL-SP$_S$+MP	RL	$0.88_{\pm 0.01}$	$1.00_{\pm 0.00}$	$0.05_{\pm 0.01}$	$0.64_{\pm 0.07}$
	VQEL-SP$_S$+MP	RL+Pres	$0.88_{\pm 0.01}$	$1.00_{\pm 0.00}$	$0.05_{\pm 0.01}$	$0.63_{\pm 0.09}$
DSPRITES	GS-ST	-	$0.81_{\pm 0.01}$	$0.90_{\pm 0.00}$	$0.10_{\pm 0.01}$	$1.80_{\pm 0.05}$
	REINFORCE	-	$0.88_{\pm 0.02}$	$0.80_{\pm 0.10}$	$0.06_{\pm 0.00}$	$1.06_{\pm 0.13}$
	VQEL-SP$_S$	-	$0.78_{\pm 0.02}$	$1.00_{\pm 0.00}$	$0.09_{\pm 0.01}$	$0.85_{\pm 0.09}$
	VQEL-SP$_S$+MP	Frozen	$0.86_{\pm 0.01}$	$1.00_{\pm 0.00}$	$0.09_{\pm 0.01}$	$0.85_{\pm 0.09}$
	VQEL-SP$_S$+MP	RL	$0.87_{\pm 0.01}$	$1.00_{\pm 0.00}$	$0.09_{\pm 0.01}$	$0.85_{\pm 0.11}$
	VQEL-SP$_S$+MP	RL+Pres	$0.87_{\pm 0.01}$	$1.00_{\pm 0.00}$	$0.09_{\pm 0.01}$	$0.82_{\pm 0.07}$
CELEBA	GS-ST	-	$0.90_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.14_{\pm 0.01}$	$1.01_{\pm 0.08}$
	REINFORCE	-	$0.93_{\pm 0.01}$	$1.00_{\pm 0.00}$	$0.11_{\pm 0.03}$	$0.90_{\pm 0.06}$
	VQEL-SP$_S$	-	$0.82_{\pm 0.02}$	$1.00_{\pm 0.00}$	$0.10_{\pm 0.01}$	$0.76_{\pm 0.01}$
	VQEL-SP$_S$+MP	Frozen	$0.88_{\pm 0.01}$	$1.00_{\pm 0.00}$	$0.10_{\pm 0.01}$	$0.76_{\pm 0.01}$
	VQEL-SP$_S$+MP	RL	$0.89_{\pm 0.01}$	$1.00_{\pm 0.00}$	$0.10_{\pm 0.02}$	$0.73_{\pm 0.03}$
	VQEL-SP$_S$+MP	RL+Pres	$0.91_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.10_{\pm 0.00}$	$0.58_{\pm 0.02}$

Table 4: Performance comparison across datasets and evaluation metrics for the sender self-play game using Euclidean distance in the codebook.

B Receiver Self-Play

In this experiment, the receiver first invents its own language during the self-play phase and then communicates with the sender in the mutual-play phase. During the mutual-play, the receiver can either be fine-tuned or frozen. As shown in Table 5, VQEL performs worse in both modes compared to the baselines and to VQEL in the previous experiments (Sender Self-Play and Sender and Receiver Self-play). In particular, freezing the receiver results in lower accuracy than fine-tuning, indicating that agents are unable to effectively transfer its language as a receiver.

Technically, successful language invention requires learning an effective codebook. In the MP phase, the receiver must use the codebook established by the sender. Therefore, self-play for the receiver does not help it learn this codebook; instead, the sender must learn a new codebook from scratch during mutual-play. By contrast, in the Sender Self-Play scenario, the codebook is already learned during self-play, making optimization easier in the mutual-play phase.

Dataset	Method	Receiver Update	ACC ↑	AW ↑	TopSim ↑	H(C|M) ↓
OBJECTS	GS-ST	-	$0.78_{\pm 0.01}$	$0.93_{\pm 0.06}$	$0.21_{\pm 0.02}$	$1.04_{\pm 0.03}$
	REINFORCE	-	$0.51_{\pm 0.21}$	$0.47_{\pm 0.12}$	$0.14_{\pm 0.07}$	$2.21_{\pm 1.28}$
	VQEL-SP_R	-	$0.82_{\pm 0.01}$	$1.00_{\pm 0.00}$	$0.19_{\pm 0.01}$	$0.12_{\pm 0.02}$
	VQEL-SP_R+MP	Frozen	$0.14_{\pm 0.02}$	$0.70_{\pm 0.10}$	$0.14_{\pm 0.03}$	$3.97_{\pm 0.26}$
	VQEL-SP_R+MP	Fine-tuned	$0.43_{\pm 0.12}$	$1.00_{\pm 0.00}$	$0.18_{\pm 0.02}$	$2.17_{\pm 0.40}$
SHAPE	GS-ST	-	$0.82_{\pm 0.01}$	$1.00_{\pm 0.00}$	$0.02_{\pm 0.01}$	$1.21_{\pm 0.11}$
	REINFORCE	-	$0.86_{\pm 0.00}$	$0.83_{\pm 0.15}$	$0.01_{\pm 0.01}$	$0.88_{\pm 0.11}$
	VQEL-SP_R	-	$0.87_{\pm 0.01}$	$1.00_{\pm 0.00}$	$0.05_{\pm 0.03}$	$0.38_{\pm 0.10}$
	VQEL-SP_R+MP	Frozen	$0.40_{\pm 0.16}$	$0.83_{\pm 0.12}$	$0.03_{\pm 0.00}$	$2.04_{\pm 0.10}$
	VQEL-SP_R+MP	Fine-tuned	$0.85_{\pm 0.01}$	$1.00_{\pm 0.00}$	$0.02_{\pm 0.01}$	$0.65_{\pm 0.05}$
DSPRITES	GS-ST	-	$0.81_{\pm 0.01}$	$0.90_{\pm 0.00}$	$0.10_{\pm 0.01}$	$1.80_{\pm 0.05}$
	REINFORCE	-	$0.88_{\pm 0.02}$	$0.80_{\pm 0.10}$	$0.06_{\pm 0.00}$	$1.06_{\pm 0.13}$
	VQEL-SP_R	-	$0.91_{\pm 0.02}$	$1.00_{\pm 0.00}$	$0.07_{\pm 0.01}$	$0.40_{\pm 0.02}$
	VQEL-SP_R+MP	Frozen	$0.24_{\pm 0.09}$	$0.67_{\pm 0.06}$	$0.09_{\pm 0.01}$	$4.56_{\pm 0.52}$
	VQEL-SP_R+MP	Fine-tuned	$0.86_{\pm 0.01}$	$1.00_{\pm 0.00}$	$0.07_{\pm 0.00}$	$0.80_{\pm 0.12}$
CELEBA	GS-ST	-	$0.90_{\pm 0.00}$	$1.00_{\pm 0.00}$	$0.14_{\pm 0.01}$	$1.01_{\pm 0.08}$
	REINFORCE	-	$0.93_{\pm 0.01}$	$1.00_{\pm 0.00}$	$0.11_{\pm 0.03}$	$0.90_{\pm 0.06}$
	VQEL-SP_R	-	$0.89_{\pm 0.01}$	$1.00_{\pm 0.00}$	$0.10_{\pm 0.04}$	$0.58_{\pm 0.10}$
	VQEL-SP_R+MP	Frozen	$0.40_{\pm 0.04}$	$0.77_{\pm 0.23}$	$0.12_{\pm 0.06}$	$3.80_{\pm 0.64}$
	VQEL-SP_R+MP	Fine-tuned	$0.54_{\pm 0.05}$	$1.00_{\pm 0.00}$	$0.12_{\pm 0.01}$	$2.48_{\pm 0.24}$

Table 5: Performance comparison across datasets and evaluation metrics for the receiver self-play game.