LARGE-SCALE EMERGENT COMMUNICATION: THE DATASET, THE TASK AND THE POPULATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Emergent communication aims for a better understanding of human language evolution and building more efficient representations. We posit that reaching these goals will require scaling up, in contrast to a significant amount of literature that focuses on setting up small-scale problems to tease out desired properties of the emergent languages. We focus on three independent aspects to scale up, namely the dataset, task complexity, and population size. We provide a first set of results for large populations solving complex tasks on realistic large-scale datasets, as well as an easy-to-use codebase to enable further experimentation. In more complex tasks on larger datasets, we find that RL training can become unstable, but responds well to established stabilization techniques. We also identify the need for a different target than topographic similarity when working with large image datasets, due to the absence of a meaningful conceptual representation. Instead, we focus on the learnability of emergent languages. Finally, larger populations do not always produce the expected results of stronger generalization and more robust representations, leading us to explore different ways to take advantage of the population, such as voting and imitation learning.

1 INTRODUCTION

Language emergence is at the intersection of cognitive science and machine learning. From a cognitive science perspective, looking at artificial agents as another expressive species can shed light on the source of linguistic regularities (Guo et al., 2019; Chaabouni et al., 2021). From a machine learning perspective, language evolution is a promising direction to shape agents' representation and design interactive AI (Lazaridou et al., 2020; Li et al., 2020). Historically, this field explored diverse models, e.g., grammar, trees, Bayesian models, etc., with different optimization procedures, e.g., cellular-automaton, evolutionary algorithm etc. (Steels, 1997; Wagner et al., 2003; Kirby, 2002). Following advances in machine learning, there was a sudden shift of paradigm toward deep reinforcement learning methods (Foerster et al., 2016; Lazaridou et al., 2017). First, deep neural networks allow modeling large and ambiguous problems with little prior knowledge (He et al., 2016; Brown et al., 2020). Second, reinforcement learning optimizes difficult interactive problems (Mnih et al., 2015; Silver et al., 2017). Deep RL became a new frontier in language emergence.

Many works have revisited previous language emergence experiments through deep RL settings, assessing various hypotheses of language evolution, e.g. iterated learning (Kirby 2002 vs. Ren et al. 2019), contact linguistics (Myers-Scotton et al. 2002 vs. Graesser et al. 2019), or theory of mind (Batali 1998 vs. Choi et al. 2018). Noticeably, they all relied on different variants of the well-known Lewis game (Lewis, 1969). There, a speaker agent must describe an object to a listener agent, which must then retrieve it among a set of other objects. The two agents thus need to settle on a common communication protocol to solve the game. While agents manage to solve the Lewis game, their communication protocol was shown to be degenerate with deep RL agents, lacking core properties of human languages such as compositionality or efficient word length (Lazaridou et al., 2018; Chaabouni et al., 2019). Faced with these protocols, several works proposed ad-hoc solutions by constraining the game and agents' capacity (Kottur et al., 2017; Resnick et al., 2020). Yet, if reducing the problem complexity is tempting, leading to expected outcome, it may miss general language emergence behaviors (Hayes, 1985). We here take a different route. As scaling up datasets and tasks were crucial in many ML achievements (Sutton, 2019), we advocate that scaling up the

Lewis game is paramount to release the full potential of deep RL methods in language emergence. We here focus on three scaling dimensions: the dataset, the task complexity, and the population size.

Finally, we study three properties of emergent languages in our experiments: generalization, robustness to input noise, and ease of learning over transfer tasks. Each metric analyses different facets of the communication protocol. Note that the proposed scaling up remains computationally tractable, i.e., most of the experiments can be done within a day on a single GPU. The source code is available at `HIDDEN` to ease reproducibility.

Overall, our experiments provide a large spectrum of observations, both positive and negative. First, we found that scaling up the Lewis game quickly entails unstable RL optimization. We propose the KL-regularization as an effective way to overcome this issue. Second, we observe that complexifying the task has two positive aspects: it better discriminates between methods and increases the quality of the communication protocol. Third, we note that the wide-spread topographic similarity metric becomes limited when dealing with natural images. We thus explore transfer learning as a new evaluation metric inspired by the self-supervised learning literature, to correctly assess the communication protocol quality. Fourth, we observe little and no systematic benefit when increasing the population size on languages' properties. Yet, we highlight that populations should be leveraged differently, e.g., with voting systems or within-speakers imitation. In particular, we show that such population dynamics lead to more robust, productive, and in some cases easy-to-learn languages even when compared to the strong one-pair baseline, opening up new research opportunities. In the end, we expect that these observations, baselines, and good practices would allow the language emergent community to benefit further from deep RL advances.

2 DISCRIMINATION GAME

The discrimination game involves two players, *Speaker* and *Listener*, and is decomposed into three sequential stages. First, Speaker receives a target image x and outputs a message m that is sent to Listener. Second, Listener receives m and selects an image $\hat{x}$ among a set $\mathcal{X}$ of different images containing the target image x. The set $\mathcal{X} \setminus \{x\}$ is called *distractors*. Finally, the target image x is revealed to Listener. If Listener had selected the target image, then both players receive a positive reward of 1, otherwise, they receive a reward of 0. A graphical description of the discrimination game is provided in Fig. 6 in Appendix A.1. Speaker and Listener are parameterized by a set of parameters θ and ϕ respectively. The message $m(x, \theta) = (w_t(x, \theta))_{t=0}^{T-1}$ is a sequence of length T of words $w_t(x, \theta)$ and is a function of Speaker's parameters θ and the image x. When no confusion is possible, we omit the dependence on x and θ to simplify notations. A word w_t is an element of a discrete and finite set $\mathcal{W}$. The image selected by Listener $\hat{x}(\phi, m, \mathcal{X})$ is a function of Listener's parameters ϕ, the message m and the set $\mathcal{X}$. Similarly we omit the dependence on ϕ, m and $\mathcal{X}$ when no confusion is possible. Finally, the reward for Listener and Speaker is denoted by $R(x, \hat{x}) = 1_{x=\hat{x}}$.

Communication in a population of agents A straightforward extension of the standard discrimination game is to train a population of agents. In this case, given a population of N Speakers and N Listeners, we sample, at each training step, S Speakers and S Listeners with replacement to construct S random pairs.[1] All pairs observe the same examples and are trained independently to play the discrimination game as described in the 1-pair setting above. In our case, $S = N$ so that each agent is trained on average once per step, to have a fair comparison with the baseline of 1 pair. At inference time, we take the first P Speakers and P Listeners, and construct all the possible P^2 pairs. To compute the accuracy at evaluation, we average the rewards of all pairs.

Exploiting the population The training dynamic of the population of agents described above does not take advantage of the diversity of the population. Indeed, we treat all agents in the same way. However, one motivation for the effectiveness of the population is the variety between Speakers to invent new structures and between Listeners to avoid overfitting. In this work, we introduce two different ways to exploit the population; on Speakers' side, we add imitation, and on Listeners' side, we allow Listeners to vote to get the final prediction. Mathematical details are found in Sec. 3.3.

Imitation training consists of two different steps. First, as described above, Speakers and Listeners are paired randomly for M interaction steps to learn to communicate. Second, Speakers are

[1] In this work, we only consider populations with the same number of Speakers and Listeners.

trained for 1 imitation step as follows: we select the best Speaker (called "teacher"), and randomly sample $K-1$ other Speakers as "students". We then train all the students in a supervised way to imitate the teacher speaker. We alternate between the interaction steps and the imitation step until convergence. More details on the imitation procedure are reported in Sec. 3.3. Both M and K are hyper-parameters.

Voting is only used at inference time, irrespective of the training mode. In this setting, instead of averaging pairs' rewards to get the evaluation accuracy, we now allow Listeners to vote to get a unique prediction, which is inspired by ensemble methods to reduce prediction errors (Polikar, 2012). Prior works have also looked at the impact of training agents with different dynamics (e.g., Ren et al., 2019; Guo et al., 2019; Cogswell et al., 2019; Li & Bowling, 2019). For completeness, we also reproduce ease-of-teaching protocol as another effective way to train communicating agents in the Appendix A.8 (Li & Bowling, 2019). It aims to spur agents to develop an easy-to-learn language by periodically resetting Listeners within a population.

3 SETUP

3.1 DATASETS

ImageNet () contains XXX natural image of animals and objects over XXX labels. We generate a validation set by putting apart the last 50.000 images of the training set.

CelebA () contains XXX natural image of celebrity faces over XXX identities. Noticeably, each faces also contains XX attributes, e.g. glasses, color hair, to potentially analyse the language topography. We also split celeba into two regime, and refer them as unseen and seen splits. The official split respectively contains XXX, XXX and XXX samples in training, validation, and test set. Importantly, those sets have non-overlapping identities, leading to a zero-shot setting (Unseen). We then also evaluate a distribution by reshuffling the dataset as detailed in Appendix **??**.

Image Processing In both dataset, we resized images to 256 pixels along the shorter side using bicubic resampling, before applying a 224×224 center crop. We normalize the color channels by subtracting the average color and dividing by the standard deviation computed on ImageNet. We then use the ResNet-50 pretrained on ImageNet with the self-supervised method BYOL (Grill et al., 2020) to extract the final representation, i.e. 2048 logits of the image.

3.2 ARCHITECTURES

Speaker architecture Speaker is a neural network that receives an image x as input and outputs a message $m = (w_t)_{t=0}^{T-1}$. It is composed of a fixed (non-trainable) image encoder f that transforms the image x into an embedding $f(x)$, before being projected by a state adapter c_θ to an initial state of an RNN $z_{-1,\theta} = c_\theta(f(x))$. The RNN receives as input a word embedding $e_{t,\theta} = g_\theta(w_{t-1})$ and outputs the next state $z_{t,\theta} = h_\theta(z_{t-1,\theta}, e_{t,\theta})$. The first word $e_{0,\theta} = g_\theta(\texttt{sos})$ is initialized with a start of sequence symbol. The state $z_{t,\theta}$ is then fed to two different heads. The value head v_θ estimates the value of the expected reward knowing $z_{t,\theta}$ and the policy head computes the probability of the chosen word given $z_{t,\theta}$. Then, a sampling function s picks the word $w_t = s(\pi_\theta(.|z_{t,\theta}))$. Finally w_t is fed back to the word embedder g_θ to produce the next word and so on until the maximum message length T is reached. The sampling function s differs at training and evaluation time. At training, the word $w_t \sim \pi_\theta(.|z_{t,\theta})$ is sampled according to the policy whereas it is greedily selected at evaluation.

Listener architecture Listener is a neural network that receives a message $m = (w_t)_{t=0}^{T-1}$, which was computed by Speaker from the image x, and a set of images $\mathcal{X}$ containing the target image x. It outputs the probability over each image $\tilde{x} \in \mathcal{X}$ of being the target image x. Listener is composed of a RNN cell h_ϕ, which is initialized to the null vector $z_{-1,\phi} = \mathbf{0}$. The message m is decoded by processing the sequence of word embeddings $e_{t,\phi} = g_\phi(w_t)$ through the RNN such as $z_{t,\phi} = h_\phi(z_{t-1,\phi}, e_{t,\phi})$. The final state $z_{T-1,\phi}$ is then fed to a predictor network p_ϕ to output $p_{m,\phi} = p_\phi(z_{T-1,\phi})$. In parallel, each image $\tilde{x}$ is projected through the image encoder f and the network t_ϕ to obtain the image embedding $t_{\tilde{x}} = t_\phi(f(\tilde{x}))$. Both messages and image embeddings are then compared through a score function $\texttt{score}(m, \tilde{x}, \phi) = \cos(\frac{p_{m,\phi}}{\|p_{m,\phi}\|_2}, \frac{t_{\tilde{x}}}{\|t_{\tilde{x}}\|_2})$. The scores

over all images are normalized via a softmax to get a probability $\pi_\phi(.|m, \mathcal{X})$. Finally, Listener selects an image by taking the best guess according to π_ϕ, i.e. $\hat{x} \in \arg\max_{\tilde{x}\in\mathcal{X}} \pi_\phi(\tilde{x}|m, \mathcal{X})$.

Architecture details, hyper-parameters and graphical descriptions are provided in Appendix A.2.

3.3 Optimization

Speaker training and loss The goal of a Speaker is to optimize the message m sent to a Listener such that the expected reward of the game is the highest possible. This can be framed as a sequential decision making problem where the decision is the choice of each word w_t. Therefore, following the policy gradient approach with a baseline (Sutton et al., 2000), we train the Speaker's network by minimizing a value loss $L_V(\theta)$ to make the value head $v_\theta(z_{t,\theta})$ fit the expected reward over a batch $\mathcal{X}$: $L_V(\theta) = \frac{1}{|\mathcal{X}|}\sum_{x\in\mathcal{X}}\sum_{t=0}^{T-1}\left(R(x,\hat{x}) - v_\theta(z_{t,\theta})\right)^2$, and a policy loss function $L_\pi(\theta)$ to maximise the expected reward $L_\pi(\theta) = \frac{1}{|\mathcal{X}|}\sum_{x\in\mathcal{X}}\sum_{t=0}^{T-1} \mathtt{sg}\left(R(x,\hat{x}) - v_\theta(z_{t,\theta})\right)\log(\pi_\theta(w_t|z_{t,\theta}))$, where $\mathtt{sg}(.)$ is the stop gradient function.

In addition, it is common practice in RL and emergent language literature to minimize an entropy loss $L_\mathcal{H}(\theta)$ encouraging to explore other choices of words by the Speaker (Mnih et al., 2016; Williams & Peng, 1991; Espeholt et al., 2018). Finally, several deep RL (Schulman et al., 2015; 2017) and theoretical RL papers (Geist et al., 2019; Vieillard et al., 2020a;b) argued that minimizing a KL loss $L_{\mathrm{KL}}(\theta)$ between the online policy π_θ and a target policy $\pi_{\overline{\theta}}$ instead of or in addition to entropy regularization could be beneficial for better final performance as well as stabilizing the learning. The policy $\pi_{\overline{\theta}}$ is obtained by doing an exponential moving average of the weights θ over training. The KL minimization encourages the online policy to change slowly and smoothly. To sum up, the speaker training loss $L(\theta)$ on a batch of images $\mathcal{X}$ is: $L(\theta) = L_V(\theta) + L_\pi(\theta) + \alpha L_\mathcal{H}(\theta) + \beta L_{\mathrm{KL}}(\theta)$, where α and β are hyper-parameters. More details about how we derive the losses $L_V, L_\pi, L_\mathcal{H},$ and L_{KL} are found in the appendix A.3.

Listener training and loss The listener is also trained to maximize the reward but acts only on predicting the best guess for the game $\hat{x} \in \arg\max_{\tilde{x}\in\mathcal{X}} \pi_\phi(\tilde{x}|m, \mathcal{X})$ that depends on the listener parameters ϕ. To do so, for a given batch $\mathcal{X}$ containing x, the Listener's network should output a probability $\pi_\phi(\tilde{x}|m, \mathcal{X}) = 1$ if $x = \tilde{x}$ and a probability of $\pi_\phi(\tilde{x}|m, \mathcal{X}) = 0$ otherwise. Therefore, we use a multi-class classification loss where the correct class is the index of x in the batch $\mathcal{X}$ also called InfoNCE loss (van den Oord et al., 2018): $L(\phi) = -\frac{1}{|\mathcal{X}|}\sum_{x\in\mathcal{X}}\log\left(\pi_\phi(x|m, X)\right)$.

Imitation training among a group of speakers. In a training imitation step, a group of K speakers among the total population of N speakers is selected. Among those K speakers, one speaker is going to play the role of the teacher and $K-1$ are going to play the role of the students. To choose the teacher among the K speakers, we use as metric the exponential moving average of the accuracies over each batch on which a given speaker has been trained on, σ_i. More details about how we compute σ_i are provided in the appendix. Then the teacher is simply the speaker with the higher σ_i among the K speakers. Now for convenience, let us note θ_T the parameters of the teacher and θ_S the parameters of a student. A student θ_S is going to be trained on a batch of data $\mathcal{X}$ by imitating the messages of the teacher θ_T with the following loss: $L_I(\theta_S) = -\frac{1}{|\mathcal{X}|}\sum_{x\in\mathcal{X}}\sum_{t=0}^{T-1}\log\pi_{\theta_S}(w_t(x,\theta_T)|x, z_{\theta_S,t})$, where $L_I(\theta_S)$ is a cross-entropy loss to encourage a student to output the same words as the teacher.

Listener's voting at inference time At inference time, we can use all listeners $(\phi_i)_{i=0}^N$ of the population as an ensemble of networks to find the best guess $\hat{x}$ over a batch $\mathcal{X}$ of images for a message m coming from a speaker. One simple way of doing that consists in averaging the score probabilities of the listeners and taking the best guess of this average. More precisely, for each listener ϕ_i, each message m and batch $\mathcal{X}$, we have the score probability $\pi_{\phi_i}(.|m, \mathcal{X})$. Then the choice $\hat{x}(m, \mathcal{X}, (\phi_i)_{i=0}^N)$ of the listeners' population for the message m and the batch $\mathcal{X}$ would be the best guess of the average distribution which is $\hat{x}(m, \mathcal{X}, (\phi_i)_{i=0}^N) = \arg\max_{\tilde{x}\in\mathcal{X}}\sum_{i=0}^{N-1}\pi_{\phi_i}(\tilde{x}|m, \mathcal{X})$.

3.4 Language Properties

Generalization measures the ability of agents to communicate about new inputs. To measure generalization, we simply report the test accuracy. For CelebA dataset, we inspect two different

types of generalization; (1) *Seen Acc*, and (2) *Unseen Acc*. *Seen Acc* represents agents' accuracy on a test set distributed identically to the train set (see Section 3.1). Whereas *UnSeen Acc* captures a harder form of generalization by testing agents on never-seen identities at training. We use the official split of CelebA to compute *UnSeen Acc*. Finally, for ImageNet, we only report *Seen Acc*.

Robustness We report agents' accuracy when faced with noisy inputs at test time. To construct the new stimuli, we add for each batch a Gaussian noise, with mean 0 and a standard deviation $1/2$ of the standard deviation of the batch. Note that we sample different noises for Speakers' and Listeners' inputs. That is, while Listeners' were trained to find the exact input of the Speaker at train time, they are now required to uncover a different input.

Topographic Similarity (TopSim) *Topsim* (Brighton & Kirby, 2006) is commonly used in the emergent communication literature as a proxy for compositionality (e.g., Li & Bowling, 2019; Lazaridou et al., 2018). Hence, it is an important metric as compositionality is widely considered crucial for generalization (Fodor & Lepore, 2002; Marcus, 2003). *TopSim* tests whether close objects in the input space are described by close messages. To do so, it assumes a distance both in the input and message spaces. Following prior works, we use the edit-distance and the cosine-distance as distances in the input and message spaces.

Ease and transfer learning (ETL) This metric captures how fast and well the emergent language is transmitted to new listeners performing a distinct task. It is similar to the ease-of-teaching measure introduced in (Li & Bowling, 2019) to the exception that now listeners are trained to solve a task different from the initial task that the emergent language is optimized to solve. To measure ETL, we take X fixed speakers from the population after convergence, and feed their deterministic language to 3 newly initialized listeners.[2] We argmax from speakers' distribution to construct the deterministic languages. Finally, we report how fast the new listeners perform a given task when trained with the fixed languages. In practice, we look at the ease of learning when listeners are trained to solve a harder discrimination task and a classification task (classifying faces' identities for CelebA and classes for ImageNet). Hence, the ETL metric not only captures the generality of the language to new learners, but also its ability to transfer to new tasks.

4 EXPERIMENTS

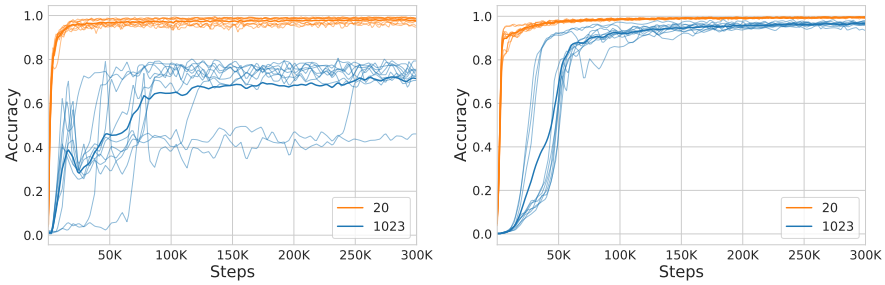
All experiments are run over 10 seeds, and we report the mean and std. Hyperparameters are provided in Appendix **??**, and the link to the source code is provided in the introduction.

4.1 TASK SCALING-UP: INCREASING THE NUMBER OF DISTRACTORS

In the emergent communication literature, agents are typically required to discriminate between a few objects (less than 20) (Mu & Goodman, 2021), or reconstruct hand-crafted attributes (Rita et al., 2020). Focusing on solving these simple tasks has been shown to lead to degenerate communication protocols (Bouchacourt & Baroni, 2018; Kottur et al., 2017). As highlighted by (Dessì et al., 2021), the Lewis game is a discrete variant of Contrastive Predictive Coding (CPC) in the self-supervised learning literature (van den Oord et al., 2018). There, it was observed that the intermediate representation quality improved when increasing the number of distractors (He et al., 2019). Here, we look at the impact of large numbers of distractors on languages generalization and the technical challenges it introduces. To this end, we train one pair of agents on the ImageNet dataset.

Scaling-up task difficulty requires to carefully tune optimization: As displayed in Figure 1(a), the training becomes unstable when increasing the number of distractors from 20 to 1023 distractors. The final mean accuracy drops from XX.X% to XX.X%, and we observe a large std across seeds. Such instability is sometimes observed in RL with complex tasks as detailed in Appendix **??**, and a common remedy consists in adding a KL-regularisation loss between the online policy and a target policy L_{KL} as described in Section 3.3. We thus add a KL regularization with a coefficient $\beta = 0.5$. In Figure 1(b), we note that, with this better regularization, 20 and 1023 distractors converge to XX.X% and XX.X% mean accuracy respectively with small stds at training. That is, adding L_{KL} both stabilizes the training and leads to better overall performance. We illustrate further the effect of this regularization across multiple sweeps in Appendix A.7.

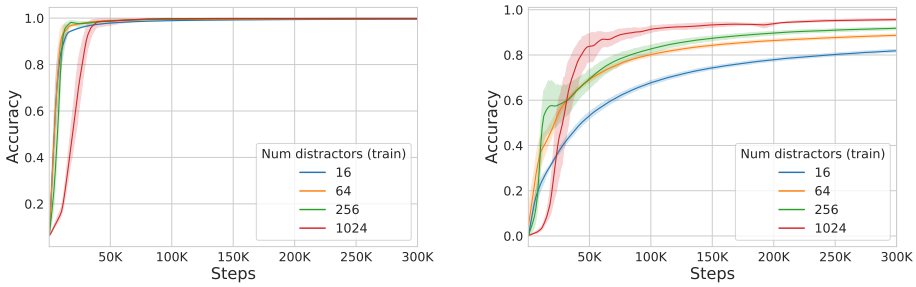
[2]For the baseline of 1 pair, we only take the unique speaker.



(a) KL coefficient $\beta = 0.0$ (b) KL coefficient $\beta = 0.5$

Figure 1: Training curves for different task complexity (20 vs. 1023 distractors) w/ and w/o KL regularization. Thin lines are the training accuracy of 10 seeds while thick lines represent the mean.

Scaling-up task difficulty is necessary to differentiate methods and enhance representation: In Figure 2(a), we train 1 pair of agents on different number of distractors ranging from 63 to 1023 before evaluating them on 63 distractors.[3] Note that, in the literature, the number of distractors rarely exceeds a dozen both at training and evaluation times (Mu & Goodman, 2021; Li & Bowling, 2019). Unfortunately, no conclusions can be drawn in such settings: all methods are within 99.0% and 99.5% accuracy range and within std. The same trend is observed until reaching 1023 distractors at evaluation as shown in Figure 2(b). There, agents achieve an accuracy of 70.4%, 78.8%, 80.7% when trained on 63, 511 and 1023 distractors respectively, differentiating the various representation's productivity. These observations are expected in light of the recent self-supervised results (Chen et al., 2020); harder training tasks lead to better representation and harder evaluation tasks better discriminate algorithms. Yet, they clearly illustrate how current language emergence experiments may be ill-scaled to have robust conclusions. We report further experiments in Appendix A.9. In the following, we use 1023 distractors at train and test to fully leverage our findings.



(a) Num distractors (eval): 15 (b) Num distractors at (eval): 1023

Figure 2: Test accuracy for different number of distractors at training time (lines) and at evaluation time (subplot). (a) Easy evaluation tasks: it is impossible to differentiate between methods. (b) Hard evaluation task: we finally observe that more distractors at training time induces higher accuracies.

4.2 Dataset Scaling-up: Rethinking Evaluation

In most emergent communication works, deep agents are situated in a simple environment containing just a handful of objects, represented as disentangled one-hot vectors (Kottur et al., 2017; Li & Bowling, 2019). As a result, the state space rarely goes beyond a few thousands of discrete samples. Besides, such representations are unambiguous as they are inherently compositional. We argue that such environments may be too simplistic to reach complex or diverse communication behaviors.

[3]We kept the training batch size to 1024, and only varied the number of distractors in the InfoNCE loss.

Accuracies		**ETL**		**Noise**	
Training	Discr.	Identity	Attributes	Input	Message
$\mathcal{X} = 16$					
$\mathcal{X} = 64$					
$\mathcal{X} = 256$					
$\mathcal{X} = 1024$					
best 1-pair					
second 1-pair					

Meantime, success stories in machine learning proved the importance of training neural networks on a rich and large amount of data to emulate complex distributions (Brown et al., 2020; Krizhevsky et al., 2012). Furthermore, if the goal is to reproduce human interaction, it is fundamental to incorporate complex stimuli to develop advanced concepts in language evolution (Miller & Johnson-Laird, 1976; Barsalou, 2008). In this subsection, we scale up the input space representation by using large natural image datasets, namely ImageNet and CelebA. This scaling-up requires enhancing the panel of evaluation protocols to better assess communication properties.

TopSim	**ImageNet**	**CelebA**	
Training	Image	Image	Attributes
$\lvert\mathcal{X}\rvert = 16$	$15.50_{\pm 1.94}$	$\mathbf{33.31_{\pm 5.61}}$	$13.46_{\pm 4.04}$
$\lvert\mathcal{X}\rvert = 64$	$\mathbf{19.01_{\pm 1.11}}$	$31.73_{\pm 4.66}$	$15.43_{\pm 3.3}$
$\lvert\mathcal{X}\rvert = 256$	$17.06_{\pm 3.44}$	$29.39_{\pm 4.86}$	$11.23_{\pm 3.88}$
$\lvert\mathcal{X}\rvert = 1024$	$16.52_{\pm 3.68}$	$32.48_{\pm 4.04}$	$\mathbf{16.35_{\pm 2.92}}$

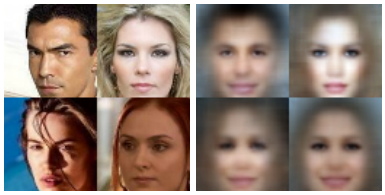


Figure 3: (left) *TopSim* with different representation of the input space (Image and Attributes) for different training difficulties, $|\mathcal{X}|$. We do not observe a correlation between *TopSim* and $|\mathcal{X}|$, and thus test accuracies. (right) Predicted images given a message of 1-pair trained with $|\mathcal{X}| = 1024$). Some features seem to be correctly encoded, e.g. sex, hair color etc, and other not, e.g. face orientation.

Topography may fall short with natural images: We compute *TopSim* using the pretrained image logits on ImageNet and CelebA (unseen) as input representation for different level of task complexity. As task difficulty, $|\mathcal{X}|$, correlates with generalization (see subsection 4.1), we expect to retrieve a correlation with *TopSim* and $|\mathcal{X}|$ by transitivity. Results are provided in in Fig. 3(left). We do not observe a consistent pattern between *TopSim* and $|\mathcal{X}|$. We also note a high variances across seeds. Although pretrained logits are excellent linear features (Grill et al., 2020), one may argue that they are not compositional. We thus reiterate the evaluation by using the attributes from CelebA. In that case too, the outcome is similar: no apparent pattern and high variance, suggesting that deep agents are succeeding in the task without communicating about the faces attributes provided in CelebA. As verification, we perform a qualitative evaluation assessing emergent languages content when $|\mathcal{X}|$=1024 (the case with the highest test accuracy). To do so, we train a new Listener, whose final layers are deconvolutions, to minimize an $L2$ loss with the target images given the deterministic emergent language. Note that this analysis only looks at the information content and not directly at compositionality. Full training details and additional samples can be found in Appendix A.6, and samples of Listener predictions are reported in Fig. 3(right). We observe that sex, face shape, makeup and hair color are fairly encoded, while others such as the smile are not.[4] However, *TopSim* does not allow detecting these differences. Finally, *TopSim* relies on strong assumptions, such as the dependence on the distance, and the use of a linear correlation between messages and inputs. Such hypothesis may not hold when moving from artificial to realistic data.

Natural images enable new and diverse evaluation protocol with Ease-of-Learning: The complexity and richness of image can be turned into an advantage. A single image entail a large variety of tasks ranging from classification, segmentation, captioning, attribute tagging etc., where each task is a proxy for a different features (Grill et al., 2020).

[4]The features sex, face shape, hair color and smile are available attributes in CelebA.

4.3 POPULATION SCALING-UP: EXPLOITING THE POPULATION DYNAMIC

The emergent communication literature often considers a single pair of agents. However, there is evidence in the multi-agent literature that such a setting may lead to extreme co-adaptation and overfitting (Lanctot et al., 2017). One counter-measure consists in sampling agents within a population (Jaderberg et al., 2018). From a linguistic perspective, different large-scale corpora analysis and human simulations support the importance of the population in shaping our languages (Gary Lupyan, 2010; Bromham et al., 2015; Raviv et al., 2019a;b). Some artificial simulations consider small population sizes up to 32 agents, but with mixed results about their advantage (Mordatch & Abbeel, 2018; Graesser et al., 2019; Tieleman et al., 2018; Cogswell et al., 2019). In this part, we scale up the population size. Specifically, we consider populations of size $N \in \{1, 10, 50\}$.[5] For a fairer evaluation of the population size, we consider the strong baseline of "best 1 pair". The latter is constructed by selecting the best seed among the 10 seeds of "1 pair" based on the validation accuracy. Hence, we investigate how training agents in a population is beneficial compared to training independent pairs in parallel. In other words, we ask if the population is beneficial given a computational power budget. We consider both CelebA and ImageNet datasets and look at the population size impact on the generalization, robustness, and ease and transfer learning of the languages.

When considering the right baseline, population size does not improve generalization and robustness: Tables 1&2 show the language generalization and robustness performances for CelebA and ImageNet respectively. One noticeable difference between both datasets is the behavior of "best 1 seed". For ImageNet, "best 1 seed" achieves a large *Seen Acc* (the best among all the tested settings) but shows the worst robustness to noise (27.8% relative drop vs. 18.9% on average for *Robustness*). In other words, "best 1 pair" converged to a degenerate solution, that overfits the task.[6] Such behavior is not observed when experimenting with CelebA where "best 1 pair" performs according to the average "1 pair" with overall higher performances. The remaining settings show the same pattern for both datasets and demonstrate a non-systematic advantage of the population. For example, for both datasets, "50 pairs" achieves a lower *Seen Acc* and *Robustness* than "10 pairs", and both population settings generate less productive and not systematically more robust languages compared to "best 1 pair". That is, with the same computational power, there is no appealing motivation to use large population sizes if what we want is more productive and robust languages.

We do not discern a systematic advantage of the population size for the *ETL*: Fig. 5 shows the same overfitting pattern of "best 1 pair" for ImagNet experiments. Indeed, if this setting performs well on the discrimination task, it is the worst setting on the classification one (15.16% accuracy at the end of the training compared to 24.69% for the average setting). Hence, *ETL* could be also used to investigate the robustness of the languages. Then, we focus on the impact of the population size. For CelebA, Fig. 4 exhibits a benefit of the population even compared to the "best 1 pair" setting. However, this benefit does not always correlate with population size where "10 pairs" outperforms "50 pairs" on the discrimination task transfer. This non-systematic benefit of the population is also found when experimenting with ImagNet where we note an advantage only for the classification task(Fig. 5(b)). However, when learning the discrimination task (Fig. 5(a)) "50 pairs" setting demonstrates the worst *ETL*.

Overall, our results suggest that there is no benefit of the population size. In particular, if in some cases we observe a slight improvement when $N > 1$ compared to the standard setting, this result is not stable across datasets and values of N. In the remaining, we explore how different dynamics of training and evaluation could lead to a larger benefit of the population. Indeed, one common hypothesis of the advantage of the population is the presence of more creative members to generate more general languages by chance (Bromham et al., 2015). However, the standard training of the population, does not take into account the diversity of agents.

A better use of the population is needed to see a larger benefit: Here, we look at the effect of imitation at training time and vote at inference when training agents in a population. Results for generalization and robustness are reported in Tables 1&2. We observe that both imitation and vote introduce a systematic improvement compared to the standard training in a population and lead to better performances compared to "1 pair" in all metrics. Moreover, when considering the stronger

[5] N=1 is equivalent to the standard Lewis game with 1-pair. For a given N, we experiment with $2N$ agents.

[6] Note that, however, the setting did not overfit to the training set as we observe high generalization performances. Better investigation of this solution would be interesting for future works.

baseline "best 1 pair", the latter only outperforms the imitation and vote settings in 1 out of 7 metrics (*Seen Acc* for ImageNet). However, as shown previously, this case developed a degenerate language. Furthermore, we observe that if both the imitation and vote introduce a systematic improvement of the communication protocol, their benefit is even more noticeable for distribution-shifted settings (both *Unseen Acc* and *Robusteness* measures). For example, the imitation for 50 pairs adds a relative improvement of 12.8% in the *Robusteness* metric compared to 2.3% for *Seen Acc* on the CelebA dataset. The same observation holds for the vote but with a lower degree.
When considering the *ETL* on the discrimination task in Figs. 4(a)&5(a), we observe that imitation leads to a systematic improvement. Indeed, for CelebA, if the standard population was already beneficial, adding the imitation strengthen the results. For ImageNet, "10/50 pairs + imitation" perform as good or better at the end of training than the overfitted "best 1 pair" with a more stable optimization and considerably better than "1 pair". Finally, the results are less stable for *ETL* on the classification task. Specifically, if CelebA experiments suggest an advantage when training a population of agents in a standard way as opposed to w/ imitation or "1 pair", this pattern is absent with ImageNet experiments where all settings with population are on par and we only distinguish a slight benefit for "10 + imitation". Still all these settings outperform significantly "1 pair" and the degenerate setting "best 1 pair".
So ... exploiting these dynamics are always good. The 2 settings where they are not is compared to a degenerate seed that overfitted the discrimination task and for *ETL* classification CelebA. But in this case, we are still as good as the strong baseline.

Table 1: Different language properties on CelebA dataset, in %. For each setting we report the mean over 10 seeds. $\pm$ denotes 1 standard error of the mean and $\downarrow$ the relative drop.

Setting	Seen Acc	Unseen Acc	Robustness
best 1 pair	90.73	92.60	58.24($\downarrow$ 35.8%)
1 pair	$89.00_{\pm 0.48}$	$90.08_{\pm 0.54}$	$55.28_{\pm 0.67}$($\downarrow$ 37.9%)
10 pairs	$91.06_{\pm 0.23}$	$90.42_{\pm 0.55}$	$56.86_{\pm 1.00}$($\downarrow$ 37.6%)
50 pairs	$90.69_{\pm 0.61}$	$90.91_{\pm 0.37}$	$55.44_{\pm 1.16}$($\downarrow$ 38.9%)
10 pairs + imitation	$91.84_{\pm 0.31}$	$92.01_{\pm 0.41}$	$59.27_{\pm 0.88}$($\downarrow$ 35.5%)
10 pairs + imitation + vote	$92.19_{\pm 0.30}$	$92.40_{\pm 0.40}$	$59.73_{\pm 0.88}$($\downarrow$ 35.2%)
50 pairs + imitation	$92.82_{\pm 0.51}$	$93.41_{\pm 0.19}$	$62.48_{\pm 1.77}$($\downarrow$ 32.7%)
50 pairs + imitation + vote	$\mathbf{93.13_{\pm 0.50}}$	$\mathbf{93.75_{\pm 0.18}}$	$\mathbf{62.95_{\pm 1.76}}$($\downarrow$ **32.4%**)

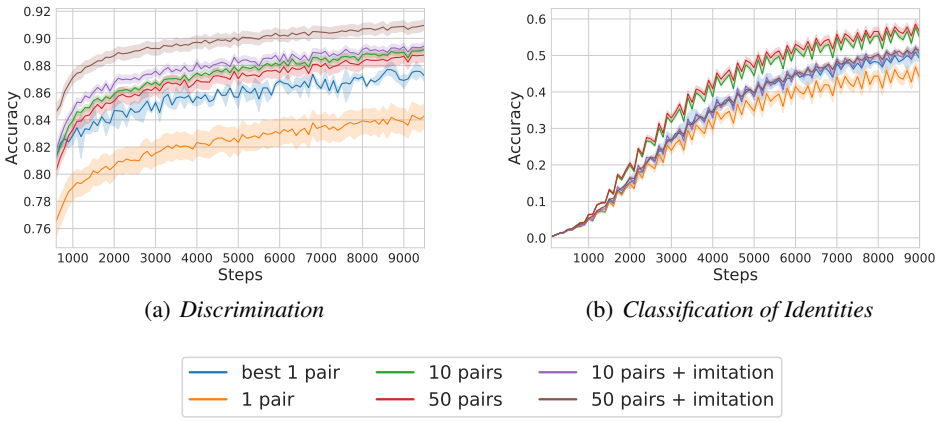
Table 2: Different language properties on ImageNet dataset, in %. For each setting we report the mean over 10 seeds. $\pm$ denotes 1 standard error of the mean and $\downarrow$ the relative drop.

Setting	Seen Acc	Robustness
best 1 pair	**97.55**	70.40 ($\downarrow$ 27.8%)
1 pair	$96.09_{\pm 0.21}$	$77.93_{\pm 1.21}$ ($\downarrow$ 18.9%)
10 pairs	$95.78_{\pm 0.29}$	$81.81_{\pm 0.77}$ ($\downarrow$ 14.6%)
50 pairs	$95.29_{\pm 0.34}$	$80.79_{\pm 0.83}$ ($\downarrow$ 15.2%)
10 pairs + imitation	$96.79_{\pm 0.12}$	$83.76_{\pm 0.37}$ ($\downarrow$ 13.5%)
10 pairs + imitation + vote	$96.95_{\pm 0.11}$	$84.08_{\pm 0.36}$ ($\downarrow$ 13.3%)
50 pairs + imitation	$96.70_{\pm 0.16}$	$84.20_{\pm 0.53}$ ($\downarrow$ 12.9%)
50 pairs + imitation + vote	$96.85_{\pm 0.15}$	$\mathbf{84.50_{\pm 0.53}}$ ($\downarrow$ **12.7%**)

5 Conclusion

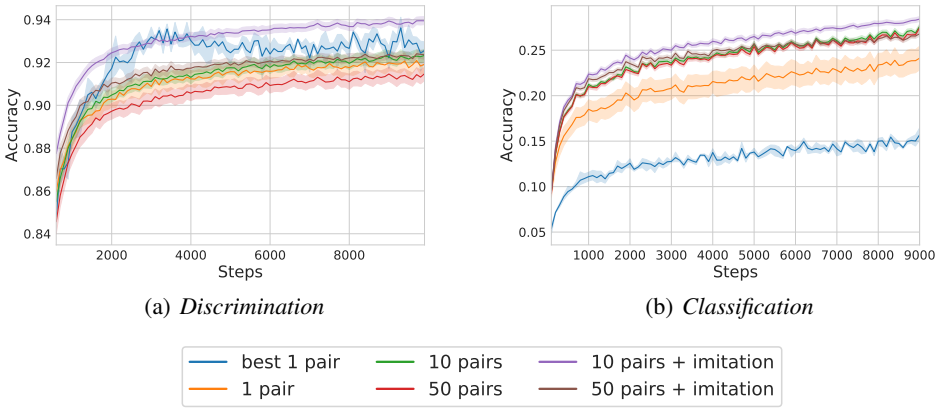
In this work, we investigated the effect of 3 scaling up dimensions:

On the dataset, mention that other works trained directly from the raw images (Dessì et al., 2021). Dessì et al. (2021) did 2 out of the 3 scaling up. Still the pair of agents communicate with only one



(a) *Discrimination* (b) *Classification of Identities*

Figure 4: *ETL* for the CelebA datase for different tasks. The results are averaged across the emergent languages of the different speakers (when it is generated in a population), newborn listeners' seeds, and across the 10 seeds of each setting. The shaded region represents the standard deviation.



(a) *Discrimination* (b) *Classification*

Figure 5: *ETL* for the ImageNet dataset for different tasks. The results are averaged across the emergent languages of the different speakers (when it is generated in a population), newborn listeners' seeds, and across the 10 seeds of each setting. The shaded region represents the standard deviation.

symbol which makes the optimization simpler, but the setting more restricted. We found that topsim bad and transfer is good

On the task, we .... We found that KL and it is also beneficial

On the population size ... Results: the importance of having the right baseline to assess fairly the impact of the population. the population size on its own has little impact on shaping the emergent languages and it is how we train agents in the population that matters the most. In this context, we explored other training and evaluation dynamics to exploit better the population [maybe if related work not complete add it here, like Other works that looked at the population dynamic; as those are the hidden underlying transitions our learners are working to model. References: Fitzgerald (2019) https://arxiv.org/pdf/1911.04362.pdf]:

Imitation: performance + evolution inspired and fitness + iterated learning + homogenous population

Vote: little benefit but completely free

In the future we should explore even other dynamics

REFERENCES

Lawrence W Barsalou. Grounded cognition. Annu. Rev. Psychol., 59:617–645, 2008.

John Batali. Computational simulations of the emergence of grammar. In James Hurford, Michael Studdert-Kennedy, and Chris Knight (eds.), Approaches to the Evolution of Language: Social and Cognitive Bases, pp. 405–426. Cambridge University Press, Cambridge, UK, 1998.

Diane Bouchacourt and Marco Baroni. How agents see things: On visual representations in an emergent language game. In Proceedings of EMNLP, pp. 981–985, Brussels, Belgium, 2018.

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

Lindell Bromham, Xia Hua, Thomas G Fitzpatrick, and Simon J Greenhill. Rate of language evolution is affected by population size. Proceedings of the National Academy of Sciences, 112(7): 2097–2102, 2015.

Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. arXiv preprint arXiv:2005.14165, 2020.

Rahma Chaabouni, Eugene Kharitonov, Emmanuel Dupoux, and Marco Baroni. Anti-efficient encoding in emergent communication. In Proceedings of NeurIPS, Vancouver, Canada, 2019. Published online: https://papers.nips.cc/book/advances-in-neural-information-processing-systems-32-2019.

Rahma Chaabouni, Eugene Kharitonov, Emmanuel Dupoux, and Marco Baroni. Communicating artificial neural networks develop efficient color-naming systems. volume 118. National Acad Sciences, 2021.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A simple framework for contrastive learning of visual representations. arXiv preprint arXiv:2002.05709, 2020.

Edward Choi, Angeliki Lazaridou, and Nando de Freitas. Compositional obverter communication learning from raw visual input. In Proceedings of ICLR Conference Track, Vancouver, Canada, 2018. Published online: https://openreview.net/group?id=ICLR.cc/2018/Conference.

Michael Cogswell, Jiasen Lu, Stefan Lee, Devi Parikh, and Dhruv Batra. Emergence of compositional language with deep generational transmission. arXiv preprint arXiv:1904.09067, 2019.

Roberto Dessì, Eugene Kharitonov, and Marco Baroni. Interpretable agent communication from scratch (with a generic visual processor emerging on the side). arXiv preprint arXiv:2106.04258, 2021.

Lasse Espeholt, Hubert Soyer, Remi Munos, Karen Simonyan, Vlad Mnih, Tom Ward, Yotam Doron, Vlad Firoiu, Tim Harley, Iain Dunning, et al. Impala: Scalable distributed deep-rl with importance weighted actor-learner architectures. In International Conference on Machine Learning, pp. 1407–1416. PMLR, 2018.

Jerry Fodor and Ernest Lepore. The Compositionality Papers. Oxford University Press, Oxford, UK, 2002.

Jakob Foerster, Ioannis Alexandros Assael, Nando de Freitas, and Shimon Whiteson. Learning to communicate with deep multi-agent reinforcement learning. In Proceedings of NIPS, pp. 2137–2145, Barcelona, Spain, 2016.

Rick Dale Gary Lupyan. Language structure is partly determined by social structure. PLoS ONE 5, 1, 2010.

Matthieu Geist, Bruno Scherrer, and Olivier Pietquin. A theory of regularized markov decision processes. In International Conference on Machine Learning, pp. 2160–2169. PMLR, 2019.

Laura Graesser, Kyunghyun Cho, and Douwe Kiela. Emergent linguistic phenomena in multi-agent communication games. In Proceedings of EMNLP, pp. 3700–3710, Hong Kong, China, 2019.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. arXiv preprint arXiv:2006.07733, 2020.

Shangmin Guo, Yi Ren, Serhii Havrylov, Stella Frank, Ivan Titov, and Kenny Smith. The emergence of compositional languages for numeric concepts through iterated learning in neural agents. https://arxiv.org/abs/1910.05291, 2019.

Patrick J. Hayes. The second naive physics manifesto. In R. Shaw & J. Bransford (ed.), Formal Theories of the Commonsense World, 1985.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In Proceedings of European Conference on Computer Vision (ECCV), 2016.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross B. Girshick. Momentum contrast for unsupervised visual representation learning. arXiv preprint arXiv:1911.05722, 2019.

Max Jaderberg, WM Czarnecki, I Dunning, L Marris, G Lever, AG Castaneda, et al. Human-level performance in first-person multiplayer games with population-based deep reinforcement learning. arxiv. arXiv preprint arXiv:1807.01281, 2018.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. CoRR, abs/1412.6980, 2014.

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

Simon Kirby, Tom Griffiths, and Kenny Smith. Iterated learning and the evolution of language. Current Opinion in Neurobiology, 28:108–114, 2014.

Satwik Kottur, José Moura, Stefan Lee, and Dhruv Batra. Natural language does not emerge 'naturally' in multi-agent dialog. In Proceedings of EMNLP, pp. 2962–2967, Copenhagen, Denmark, 2017.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In Proceedings of Advances in Neural Information Processing Systems (NeurIPS), 2012.

Marc Lanctot, Vinicius Zambaldi, Audrunas Gruslys, Angeliki Lazaridou, Karl Tuyls, Julien Pérolat, David Silver, and Thore Graepel. A unified game-theoretic approach to multiagent reinforcement learning. arXiv preprint arXiv:1711.00832, 2017.

Angeliki Lazaridou, Alexander Peysakhovich, and Marco Baroni. Multi-agent cooperation and the emergence of (natural) language. In Proceedings of ICLR Conference Track, Toulon, France, 2017. Published online: https://openreview.net/group?id=ICLR.cc/2017/conference.

Angeliki Lazaridou, Karl Moritz Hermann, Karl Tuyls, and Stephen Clark. Emergence of linguistic communication from referential games with symbolic and pixel input. In Proceedings of ICLR Conference Track, Vancouver, Canada, 2018. Published online: https://openreview.net/group?id=ICLR.cc/2018/Conference.

Angeliki Lazaridou, Anna Potapenko, and Olivier Tieleman. Multi-agent communication meets natural language: Synergies between functional and structural language learning. arXiv preprint arXiv:2005.07064, 2020.

David Lewis. Convention. Harvard University Press, Cambridge, MA, 1969.

Fushan Li and Michael Bowling. Ease-of-teaching and language structure from emergent communication. In Proceedings of NeurIPS, Vancouver, Canada, 2019. Published online: https://papers.nips.cc/book/advances-in-neural-information-processing-systems-32-2019.

Yaoyiran Li, Edoardo M Ponti, Ivan Vulić, and Anna Korhonen. Emergent communication pretraining for few-shot machine translation. arXiv preprint arXiv:2011.00890, 2020.

Gary Marcus. The Algebraic Mind. MIT Press, Cambridge, MA, 2003.

George A Miller and Philip N Johnson-Laird. Language and perception. Havard University Press, 1976.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. nature, 518(7540):529–533, 2015.

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

Igor Mordatch and Pieter Abbeel. Emergence of grounded compositional language in multi-agent populations. In Proceedings of AAAI, pp. 1495–1502, New Orleans, LA, 2018.

Jesse Mu and Noah Goodman. Emergent communication of generalizations. arXiv preprint arXiv:2106.02668, 2021.

Carol Myers-Scotton et al. Contact linguistics: Bilingual encounters and grammatical outcomes. Oxford University Press on Demand, 2002.

Robi Polikar. Ensemble learning. In Ensemble machine learning, pp. 1–34. Springer, 2012.

Limor Raviv, Antje Meyer, and Shiri Lev-Ari. Larger communities create more systematic languages. Proceedings of the Royal Society B, 286(1907):20191262, 2019a.

Limor Raviv, Antje Meyer, and Shiri Lev-Ari. Compositional structure can emerge without generational transmission. Cognition, 182:151–164, 2019b.

Yi Ren, Shangmin Guo, Serhii Havrylov, Shay Cohen, and Simon Kirby. Enhance the compositionality of emergent language by iterated learning. In Proceedings of the NeurIPS Emergent Communication Workshop, Vancouver, Canada, 2019. Published online: `https://sites.google.com/view/emecom2019/accepted-papers`.

Cinjon Resnick, Abhinav Gupta, Jakob Foerster, Andrew Dai, and Kyunghyun Cho. Capacity, bandwidth, and compositionality in emergent language learning. In Proceedings of AAMAS, Auckland, New Zealand, 2020. In press.

Mathieu Rita, Rahma Chaabouni, and Emmanuel Dupoux. " lazimpa": Lazy and impatient neural agents learn to communicate efficiently. arXiv preprint arXiv:2010.01878, 2020.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In International conference on machine learning, pp. 1889–1897. PMLR, 2015.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347, 2017.

David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. Nature, 550(7676):354, 2017.

Luc Steels. The synthetic modeling of language origins. Evolution of communication, 1(1):1–34, 1997.

Rich Sutton. The Bitter Lesson, 2019. `http://www.incompleteideas.net/IncIdeas/BitterLesson.html`.

Richard S Sutton, David A McAllester, Satinder P Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In Advances in neural information processing systems, pp. 1057–1063, 2000.

Olivier Tieleman, Angeliki Lazaridou, Shibl Mourad, Charles Blundell, and Doina Precup. Shaping representations through communication. https://openreview.net/pdf?id=HkzL4hR9Ym, 2018.

Aäron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. arXiv preprint arXiv:1807.03748, 2018.

Nino Vieillard, Tadashi Kozuno, Bruno Scherrer, Olivier Pietquin, Rémi Munos, and Matthieu Geist. Leverage the average: an analysis of kl regularization in reinforcement learning. In NeurIPS-34th Conference on Neural Information Processing Systems, 2020a.

Nino Vieillard, Olivier Pietquin, and Matthieu Geist. Munchausen reinforcement learning. arXiv preprint arXiv:2007.14430, 2020b.

Kyle Wagner, James A Reggia, Juan Uriagereka, and Gerald S Wilkinson. Progress in the simulation of emergent communication and language. Adaptive Behavior, 11(1):37–69, 2003.

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

A APPENDIX

A.1 DISCRIMINATION GAME

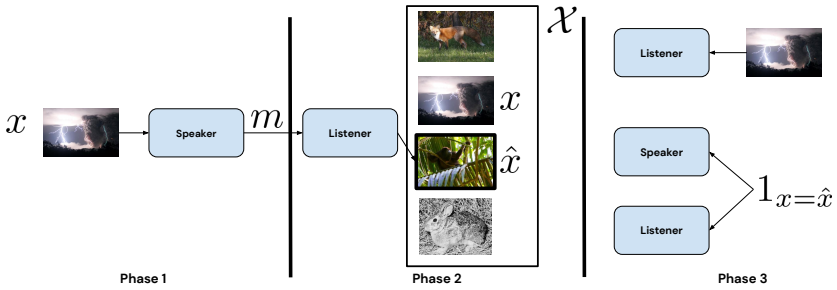


Figure 6: Example of a discrimination game on Imagenet with a set $\mathcal{X}$ of 4 images with 3 distractors. In this specific instance, Listener does not select an image $\hat{x}$ that is similar to the target image x received by Speaker. Therefore the reward received by both player $R(x, \hat{x}) = 1_{x=\hat{x}}$ will be 0.

The message $m = (w_t)_{t=0}^{T-1}$ is of a fixed length $T = 10$ with a vocabulary size of $|\mathcal{W}| = 20$. This provides an action space of size 20^{10}, significantly larger than the number of training examples of CelebA and ImageNet.

A.2 ARCHITECTURE DETAILS

Speaker The speaker's network architecture is composed of several components to transform the target image x into a message $m = (w_t)_{t=0}^{T-1}$:

- The encoder f is a fixed Resnet-50 architecture that has been previously trained on Imagenet with the BYOL algorithm. The resulting embedding $f(x)$ is of size 2048.
- The RNN h_θ used is an LSTM of hidden size 256. Therefore the core state $z_{t,\theta}$ is of size 512 and is composed of an hidden state $z_{h,t,\theta}$ of size 256 and a cell state $z_{c,t,\theta}$ of size 256.
- The core-state adapter c_θ is simply a linear layer with input size 2048 and an output size of 512 that allows to transform the embedding $f(x)$ into an appropriate core state $z_{-1,\theta} = c_\theta(f(x))$. We simply split $z_{-1,\theta}$ into two equal parts to obtain the initial hidden state $z_{h,-1,\theta}$ and the initial cell state $z_{c,-1,\theta}$.
- The word embedder g_θ associates to each discrete symbols in $\mathcal{W} \cup \{\texttt{sos}\}$ an embedding of size 10.
- The value head v_θ first selects the hidden part $z_{h,t,\theta}$ of the core state $z_{t,\theta}$ and then apply a linear layer of output size 1.
- The policy head π_θ first select the hidden part $z_{h,t,\theta}$ of the core state $z_{t,\theta}$ and then apply a linear layer of output size $|\mathcal{W}|$ to obtain logits $l_{t,\theta}$. Then a softmax layer transforms the logits into the output distribution $\pi_\theta(.|z_{t,\theta})$.

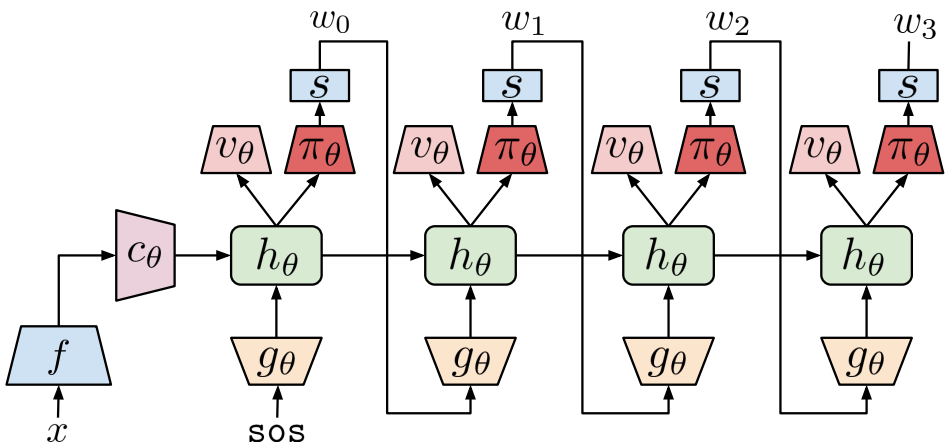


Figure 7: Graphical representation of Speaker's architecture that shows how the words $(w_t)_{t=0}^{T-1}$ are computed from the inputs x.

Listener The listener's network architecture is composed of several components to transform the message m and the input image $\tilde{x}$ into a score:

- The encoder f is a fixed Resnet-50 architecture that has been previously trained on Imagenet with the BYOL algorithm. The resulting embedding $f(x)$ is of size 2048.
- The RNN h_ϕ used is an LSTM of hidden size 512. Therefore the core state $z_{t,\phi}$ is of size 1024 and is composed of an hidden state $z_{h,t,\phi}$ of size 512 and a cell state $z_{c,t,\phi}$ of size 512.
- The word embedder g_ϕ associates to each discrete symbols in $\mathcal{W}$ an embedding of size 10.
- The target projection t_ϕ is a linear layer with output size 256.
- The core-state projection p_ϕ is a linear layer with output size 256.

Computational requirements As mentioned in the article, our approach remains computationally tractable even with widely available hardware. Table 3 summarizes these requirements to reproduce our experimental setup.

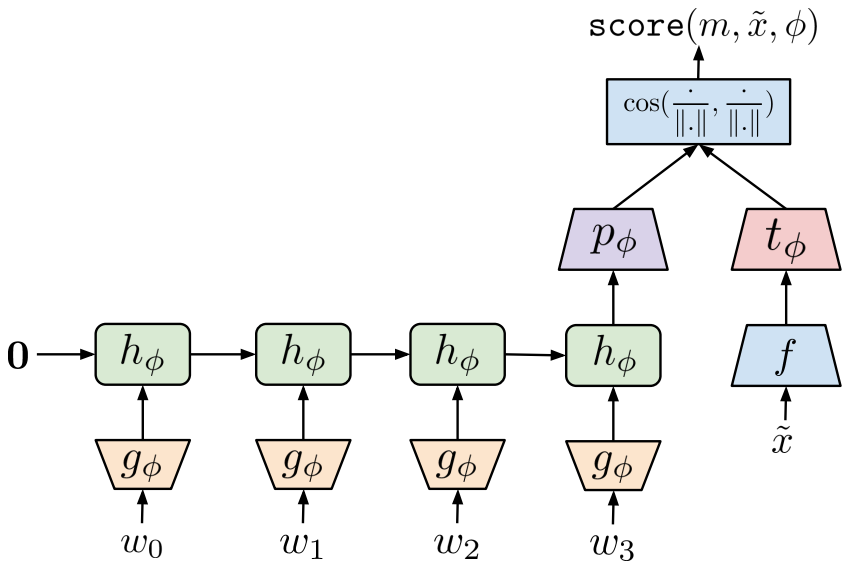


Figure 8: Graphical representation of Listener's architecture that shows how the score $\texttt{score}(m, \tilde{x}, \phi)$ is computed given a message $(w_t)_{t=0}^{T-1}$ and an input image $\tilde{x}$.

Dataset	Device	Pop. size	Memory (GiB)	Step time (ms)	Train. time (hours)
ImageNet	p100	1	0.29	42	10.5
		10	0.71	381	95.2
		50	2.63	1887	471.7
	v100	1	0.29	25	6.3
		10	0.71	223	55.7
		50	2.63	1089	272.1
CelebA	p100	1	0.33	89	7.4
		10	0.75	457	38.1
		50	2.60	2248	187.4
	v100	1	0.33	107	8.9
		10	0.75	293	24.4
		50	2.60	1408	117.3

Table 3: Computational requirements for our base setup.

A.3 OPTIMISATION DETAILS

Speaker training and loss The goal of a speaker, parameterized by θ, is to optimize the message m sent to a listener, parameterized by ϕ, such that the mean reward of the game is the highest possible. This can be framed as a sequential decision making problem where the decision is the choice of each word w_t. Thus, each word is sampled from a parameterized stochastic policy $\pi_\theta(.|x, (w_k)_{k=0}^{t-1})$ that depends on the image x and past words $(w_k)_{k=0}^{t-1}$, where for $t = 0$ we have $(w_k)_{k=0}^{-1} = \emptyset$. Then, the goal is to maximize the expected reward $J(\theta, \phi)$ by finding the best policy π_θ:

$$J(\theta, \phi) = \mathbb{E}_{x\sim\rho}\left[\mathbb{E}_{\pi_\theta,x}[R(x, \hat{x})]\right],$$

where the expectation $\mathbb{E}_{x\sim\rho}$ is over the dataset of training images and the expectation $\mathbb{E}_{\pi_\theta,x}$ is over all possible sequences $(x, (w_t)_{t=0}^{T-1})$ that can be generated by π_θ starting from x. For a given image x, we define the value $V^{\pi_\theta}(x) = \mathbb{E}_{\pi_\theta,x}[R(x, \hat{x})]$ as the expected reward for a given image, here we left the dependence over the choice of the listener $\hat{x}$ for ease of notations as the speaker cannot act on it. For a parameterized policy π_θ, we can use the policy gradient theorem to obtain the gradient:

$$\partial_\theta V^{\pi_\theta}(x) = \mathbb{E}_{\pi_\theta,x}\left[\sum_{t=0}^{T-1}\left(R(x, \hat{x}) - V_{t-1}^{\pi_\theta}(x)\right)\partial_\theta \log(\pi_\theta(w_t|x, (w_k)_{k=0}^{t-1}))\right],$$

where $V_{t-1}^{\pi_\theta}(x) = \mathbb{E}_{\pi_\theta,x}\left[R(x, \hat{x})|(w_k)_{k=0}^{t-1}\right]$ is the value conditioned on all the information revealed at time $t-1$. For our particular choice of speaker network, we encode the policy $\pi_\theta(.|x, (w_k)_{k=0}^{t-1})$ and the value $V_{t-1}^{\pi_\theta}(x)$ with the policy head $\pi_\theta(.|z_{t,\theta})$ and the value head $v_\theta(z_{t,\theta})$ respectively. Doing so is legitimate because by construction of our recurrent speaker network the embedding $z_{t,\theta}$ is a function of the image x and the past words $(w_k)_{k=0}^{t-1}$. We train then our speaker network by minimizing a value loss $L_V(\theta)$ to make $v_\theta(z_{t,\theta})$ fit $V_{t-1}^{\pi_\theta}(x)$ over a batch $\mathcal{X}$ of images:

$$L_V(\theta) = \frac{1}{|\mathcal{X}|}\sum_{x\in\mathcal{X}} L_V(\theta, x) = \frac{1}{|\mathcal{X}|}\sum_{x\in\mathcal{X}}\sum_{t=0}^{T-1}\left(R(x, \hat{x}) - v_\theta(z_{t,\theta})\right)^2,$$

and a policy loss function $L_\pi(\theta)$ to maximise the expected reward:

$$L_\pi(\theta) = \frac{1}{|\mathcal{X}|}\sum_{x\in\mathcal{X}} L_\pi(\theta, x) = \frac{1}{|\mathcal{X}|}\sum_{x\in\mathcal{X}}\sum_{t=0}^{T-1}\mathtt{sg}\left(R(x, \hat{x}) - v_\theta(z_{t,\theta})\right)\log(\pi_\theta(w_t|z_{t,\theta})),$$

where $\mathtt{sg}(.)$ is the stop gradient function. If we assume that $v_\theta(z_{t,\theta})$ fits perfectly $V_{t-1}^{\pi_\theta}(x)$, one can easily verify that $\partial_\theta L_\pi(\theta, x)$ is an unbiased estimate of $-\partial_\theta V^{\pi_\theta}(x)$ therefore $\partial_\theta L_\pi(\theta)$ is also an unbiased estimate of $-\partial_\theta J(\theta, \phi)$. In addition to following the gradient $\partial_\theta V^{\pi_\theta}(x)$, it is common practice in RL and emergent language literature to maximize an entropy term encouraging to explore other choices of words by Speaker:

$$\begin{aligned}\mathcal{H}(\pi_\theta) =& \mathbb{E}_{x\sim\rho}\left[\mathbb{E}_{\pi_\theta,x}\left[\sum_{t=0}^{T-1}\mathcal{H}(\pi_\theta(.|x, (w_k)_{k=0}^{t-1}))\right]\right],\\ =& \mathbb{E}_{x\sim\rho}\left[\mathbb{E}_{\pi_\theta,x}\left[\sum_{t=0}^{T-1}\mathbb{E}_{w\sim\pi_\theta(.|x,(w_k)_{k=0}^{t-1})}\left[\pi_\theta(w|x, (w_k)_{k=0}^{t-1})\log(\pi_\theta(w|x, (w_k)_{k=0}^{t-1}))\right]\right]\right].\end{aligned}$$

In practice, a sampled version of the negative entropy $L_\mathcal{H}(\theta)$ is minimized with the speaker network:

$$L_\mathcal{H}(\theta) = -\frac{1}{|\mathcal{X}|}\sum_{x\in\mathcal{X}}\sum_{t=0}^{T-1}\mathbb{E}_{w\sim\pi_\theta(.|z_{t,\theta})}\left[\pi_\theta(w|z_{t,\theta})\log(\pi_\theta(w|z_{t,\theta}))\right].$$

Recently, several deep RL (Schulman et al., 2015; 2017) and theoretical RL papers (Geist et al., 2019; Vieillard et al., 2020a;b) argued that minimizing the KL between the online policy π_θ and a target policy $\pi_{\overline{\theta}}$ instead of or in addition to entropy regularization could be beneficial for better final performance as well as stabilizing the learning. Generally, $\pi_{\overline{\theta}}$ is an older policy or an exponential moving average of past policies. Therefore, we also consider the following KL regularization

$\mathtt{KL}(\pi_\theta, \pi_{\overline{\theta}})$:

$$\begin{aligned}\mathtt{KL}(\pi_\theta, \pi_{\overline{\theta}}) =& \mathbb{E}_{x\sim\rho}\left[\mathbb{E}_{\pi_\theta,x}\left[\sum_{t=0}^{T-1}\mathtt{KL}(\pi_\theta(.|x,(w_k)_{k=0}^{t-1}), \pi_{\overline{\theta}}(.|x,(w_k)_{k=0}^{t-1}))\right]\right],\\ =& \mathbb{E}_{x\sim\rho}\left[\mathbb{E}_{\pi_\theta,x}\left[\sum_{t=0}^{T-1}\mathbb{E}_{w\sim\pi_\theta(.|x,(w_k)_{k=0}^{t-1})}\left[\pi_\theta(w|x,(w_k)_{k=0}^{t-1})\log(\frac{\pi_\theta(w|x,(w_k)_{k=0}^{t-1})}{\pi_{\overline{\theta}}(w|x,(w_k)_{k=0}^{t-1})})\right]\right]\right].\end{aligned}$$

In practice, the policy $\pi_{\overline{\theta}}$ is obtained by doing an exponential moving average of the weights θ over training. Then, a sampled version of the KL is minimized $L_{\mathtt{KL}}(\theta)$ with our specific speaker network:

$$L_{\mathtt{KL}}(\theta) = \frac{1}{|\mathcal{X}|}\sum_{x\in\mathcal{X}}\sum_{t=0}^{T-1}\mathbb{E}_{w\sim\pi_\theta(.|z_{t,\theta})}\left[\pi_\theta(w|z_{t,\theta})\log(\frac{\pi_\theta(w|z_{t,\theta})}{\pi_{\overline{\theta}}(w|z_{t,\overline{\theta}})})\right].$$

To sum up, the speaker training loss $L(\theta)$ on a batch of images $\mathcal{X}$ is:

$$L(\theta) = L_V(\theta) + L_\pi(\theta) + \alpha L_{\mathcal{H}}(\theta) + \beta L_{\mathtt{KL}}(\theta),$$

where α and β are hyper-parameters.

Imitation training among a group of speakers. In a training imitation step, a group of K speakers among the total population of N speakers is selected. Among those K speakers, one speaker is going to play the role of the teacher and $K-1$ are going to play the role of the students. To choose the teacher among the K speakers, we use as metric the exponential moving average of the accuracies over each batch on which a given speaker has been trained on. To be more precise, let θ_i be the i-th speaker and $\mathcal{X}$ the present batch on which θ_i has just been trained on, then σ_i is updated according to the following rule:

$$\sigma_i \leftarrow \mu\sigma_i + (1-\mu)\frac{1}{|\mathcal{X}|}\sum_{x\in\mathcal{X}} 1_{x=\hat{x}}$$

and is the exponential moving average of coefficient μ of the accuracies over each batch on which speaker i has been trained. Then the teacher is simply the speaker with the higher σ_i among the K speakers. Now for convenience, let us note θ_T the parameters of the teacher and θ_S the parameters of a student. A student θ_S is going to be trained on a batch of data $\mathcal{X}$ by imitating the messages of the teacher θ_T with the following loss:

$$L_I(\theta_S) = -\frac{1}{|\mathcal{X}|}\sum_{x\in\mathcal{X}}\sum_{t=0}^{T-1}\log\pi_{\theta_S}(w_t(x,\theta_T)|x, z_{\theta_S,t}).$$

The loss $L_I(\theta_S)$ is a cross-entropy loss that encourages the student to output the same words as the teacher.

Optimizer hyper-parameters Each member of the population (listener or speaker) uses its own optimiser. All optimisers are Adam optimisers (Kingma & Ba, 2014) with the same set of hyper-parameters:

- learning rate: $1e-4$,
- β_1: 0.9,
- β_2: 0.999,
- ϵ: $1e^{-8}$.

For Speakers' loss, we use $\alpha = 0.0001$ for the entropy regularization and $\beta = 0.5$ for the KL regularization. See Sec. A.7 for more details about the impact of these parameters.

A.4 IMITATION HYPER-PARAMETERS

In all our experiments, we perform a grid search over the parameters of M (number of interaction steps) and K (number of sampled speakers at the imitation step). For the case of a population of $N = 10$, K is chosen from $\{1, 4, 9\}$ and M from $\{1, 10, 100\}$ according to the validation accuracy. For $N = 50$, K is selected from $\{4, 9, 24\}$ and M from $\{1, 10, 100\}$.

Table 4

Dataset	max steps	$\lvert\mathcal{X}\rvert$	10 pairs + imitation (K, M)	50 pairs + imitation (K, M)	ease of learning discrim. $\lvert\mathcal{X}\rvert$
CelebA	300k	1024	(1, 10)	(9, 10)	4096
ImageNet	900k	1024	(1, 100)	(9, 10)	4096

A.5 DATASETS DETAILS

Selecting the best with respect to the listener loss at validation. At evaluation $P = min(N, 10)$ to have a fast evaluation for large populations.

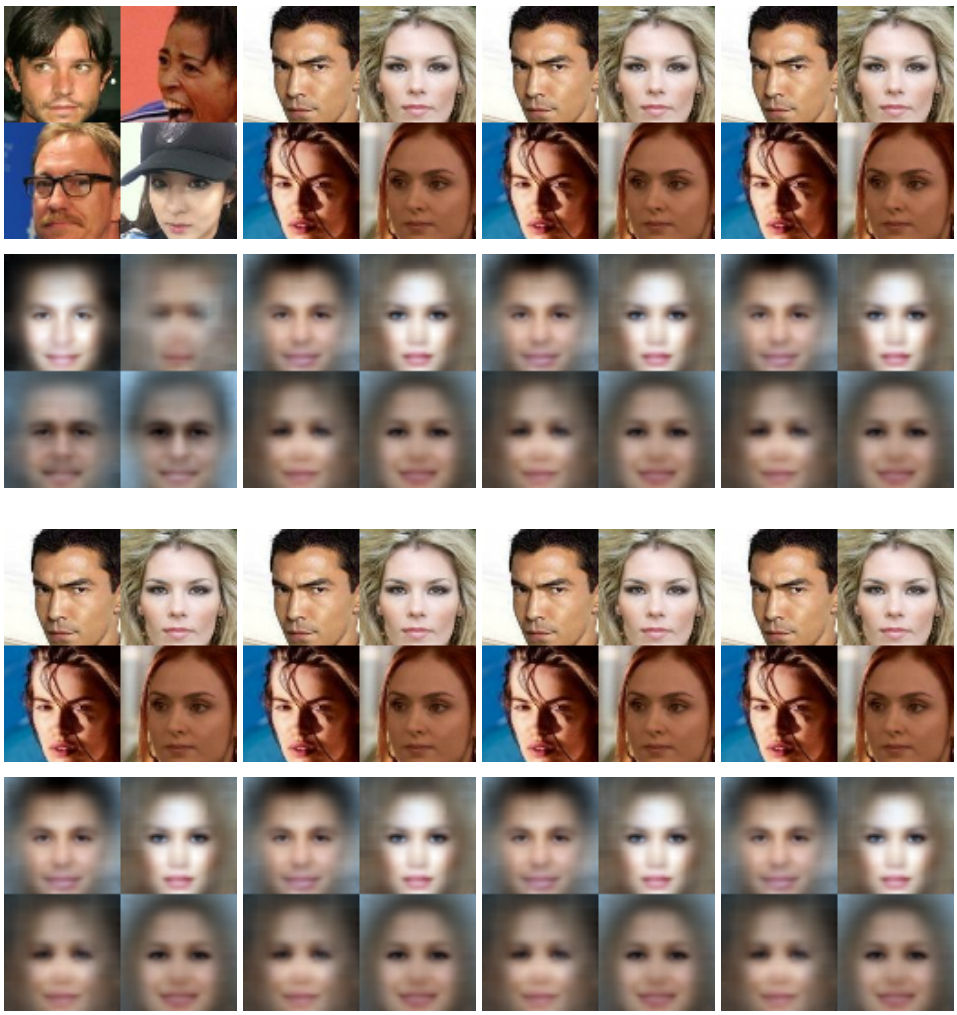
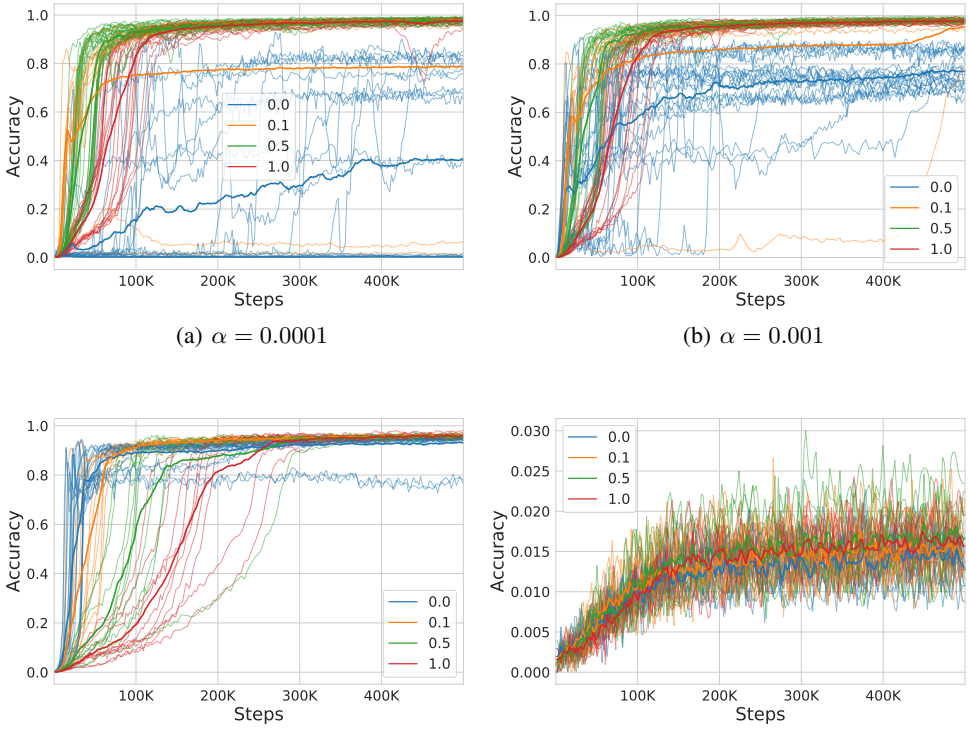


Figure 9: Different face reconstructions.

A.6 FACE RECONSTRUCTION



(a) $\alpha = 0.0001$

(b) $\alpha = 0.001$

(c) $\alpha = 0.01$

(d) $\alpha = 0.1$

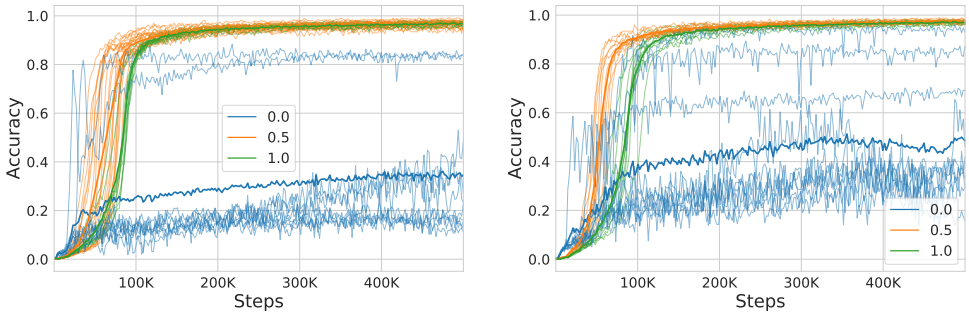
Figure 10: Training accuracy for a 1 communicating pair when varying the KL coefficient β. Each sub-figure represents the results for a fixed entropy coefficient α. Thin lines represent the training accuracy curves of different seeds. Thick lines represent the average across 10 seeds.

A.7 Impact of the KL regularization on training stability

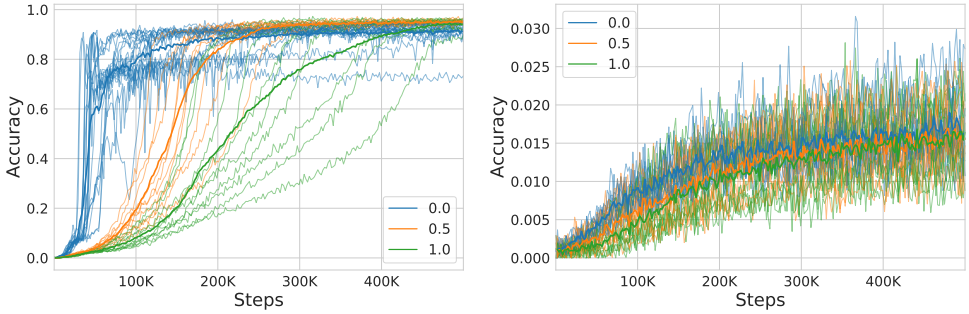
We showed in the main paper how training a pair of agents to solve a complex task (a discrimination game with 1023 distractors) becomes unstable when using common optimization algorithms. In this section, we look at the training curves when agents are trained with and without KL regularization, while varying the entropy coefficient $\alpha \in \{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$. Results are shown with a population of size 1 (Fig. 10) and 10 (Fig. 11). In both cases, agents are trained to solve the complex discrimination task with 1023 distractors.

First, we observe that the lower α is, the more crucial applying a KL regularization is. For example, in Fig. 10(a), we go from a chaotic setting that converges to an accuracy of $\sim 40\%$ when the KL coefficient $\beta = 0$ to a significantly more stable optimization with almost perfect accuracy when $\beta \geq 0.5$. Second, we observe that, if the KL regularization is useful for stable optimization, it comes at the cost of the rapidity of convergence. This is clearly shown in Figs. 10(c) and 11(c), where we observe that the larger β is, the slower the convergence is. In other words, one should select the best β optimizing the stability/rapidity trade-off. Third, in the presence of a KL regularization, training stability depends less on the entropy coefficient α. Indeed, we observe a stable and high training curves for $\alpha \in \{10^{-4}, 10^{-3}, 10^{-2}\}$.[7] However, training without KL regularization (blue curves with $\beta = 0$) is very sensitive to the value of α and the most stable case is observed for $\alpha = 0.01$. Still even in this best case without KL regularization, having a $\beta > 0$ is useful as shown in Fig. 12.

[7] It is also the case for $\alpha = 0$. Not shown here.

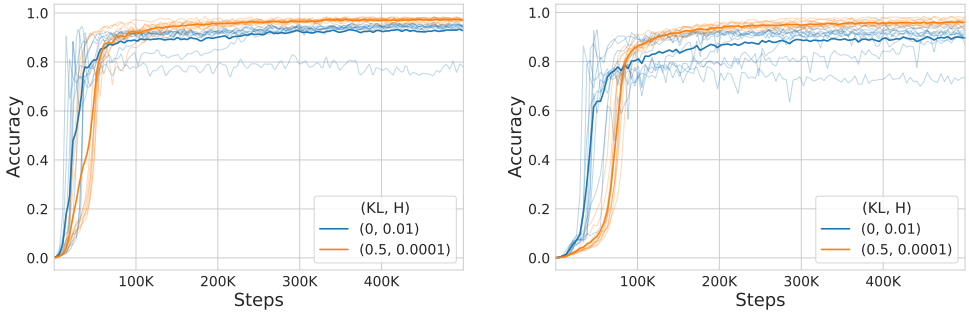


(a) $\alpha = 0.0001$ (b) $\alpha = 0.001$



(c) $\alpha = 0.01$ (d) $\alpha = 0.1$

Figure 11: Training accuracy for a 10 communicating pairs when varying the KL coefficient β. Each sub-figure represents the results for a fixed entropy coefficient α. Thin lines represent the training accuracy curves of different seeds. Thick lines represent the average across 10 seeds.



(a) 1 pair (b) 10 pairs

Figure 12: Training accuracy for a 1 and 10 pairs. Each sub-figure compares the best setting with no KL regularization (with entropy coefficient of 0.01) and the selected setting for our experiments, with a KL coefficient of 0.5 and entropy coefficient of 0.0001. Thin lines represent the training accuracy curves of different seeds. Thick lines represent the average across 10 seeds. In both cases, the setting with the KL regularization, $(0.5, 0.0001)$, exhibits a more stable convergence with a larger difference for population of 10 pairs.

A.8 STUDY OF THE RESETTING ON THE CELEBA DATASET

In this work, we argue that training deep agents to communicate in a population benefits from exploiting its richness, as opposed to a standard training (Mordatch & Abbeel, 2018). In this context, we introduced imitation and voting mechanisms. However, other prior works have stipulated a similar argument by focusing instead on the idea of the expressivity/compressibility trade-off (Kirby et al., 2014). The latter attests that our language is shaped by the two competing pressures of expressivity and compressibility. In practice, there were different methods to implement this trade-off in deep agents communication such as iterated learning (Ren et al., 2019), cultural transmission (Cogswell et al., 2019), or ease-of-teaching (Li & Bowling, 2019). In this section, we compare imitation and voting to the ease-of-teaching.[8] To encourage languages' ease-of-teaching, Li & Bowling (2019) trained deep agents to solve a discrimination task, while periodically resetting listeners. In this section, we study the impact of this baseline on the different emergent languages' properties introduced in the main paper. Specifically, we consider 3 additional settings: (1) "1 pair + reset" that consists of resetting the only listeners, (2) "10 pairs + reset" where we reset a randomly selected listener among the 10 available ones, and (3) "50 pairs + reset" where we reset a randomly selected listener among the 50 available ones. In all cases, resetting takes place every 51k steps. Note that if (1) is identical to Li & Bowling (2019) setting, (2) and (3) present 10 and 50 Speakers respectively unlike 1 Speaker as it is the case in (Li & Bowling, 2019).

We observe in Table 5 that resetting has a slight effect only in the case of "1 pair" on the generalization and robustness of the languages. For example, we note an average *Seen Acc* of 88.93% for "1 pair" vs. 90.30% for "1 pair + reset". However, there is no systematic improvement for when agents are trained in a population, ("10 pairs" vs. "10 pairs + resetting") and ("50 pairs" vs. "50 pairs + resetting"). This is in line with Li & Bowling (2019)'s results which show that an abrupt change during training leads to better results. Still, "1 pair + reset" performs worse than training agents in a population when imitation and voting are at play.

However, as mentioned in the original paper, the purpose of resetting is not to boost agents' generalization or robustness, but instead to incentivize agents to develop more compositional and easy-to-transmit languages. In the following, we look at these two metrics. First, similar to (Li & Bowling, 2019), we use *TopSim* as a proxy for compositionality. Results are shown in Table 6. Second, we report the ease of learning metric when new listeners are required to discriminate between 4096 distractors (Fig. 13(a)), to classify identities of the images (Fig. 13(b)), and to classify the attributes provided in CelebA for each image (Fig. 13(c)). Note that here, unlike (Li & Bowling, 2019), we evaluate languages' ease of learning when new listeners are required to solve a distinct task from the initial one used to develop the languages.

Results for ease of learning show the same pattern as robustness and generalization. That is, we only note a benefit of the resetting mechanism in the case of the one pair training. Indeed, "1 pair + reset" achieve only a slightly worse performance compared to training agents in a population and hence significantly better than the "1 pair" setting. However, the remaining reset settings perform on par with the standard population training. Overall, our results suggest that resetting listeners is only beneficial in the one-pair agents. However, even with the improvement provided by this dynamic, this setting still provides less easy to learn languages compared to training agents in a population.

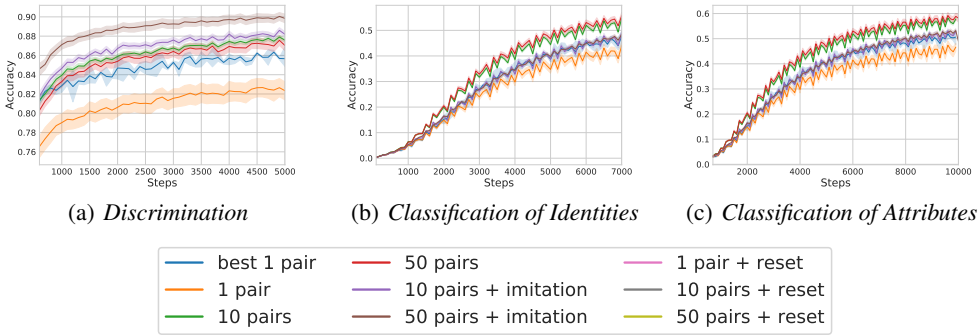
Finally, in Table 6, we do not discern a clear pattern. Indeed, if reset encourages the emergence of more compositional languages, this improvement is still within the standard deviation, suggesting a little impact of this mechanism. Also, as in the main paper, we do not see a correlation between *TopSim* and generalization or ease of learning.

A.9 IMPACT OF THE NUMBER OF DISTRACTORS AT TRAINING AND EVALUATION TIME

[8] We also considered the setup of Cogswell et al. (2019). However, preliminary results showed systematically worse results compared to the ease-of-teaching settings. Our codebase provides both options.

Table 5: Different language properties on CelebA dataset, in %. For each setting we report the mean over 10 seeds. ± denotes 1 SEM.

Setting	Seen Acc	Unseen Acc	Robustness
best 1 pair	90.97	77.97	58.78(↓ 40.0%)
1 pair	$88.93_{\pm 0.55}$	$75.84_{\pm 0.51}$	$49.99_{\pm 1.66}$(↓ 43.8%)
10 pairs	$90.64_{\pm 0.36}$	$76.66_{\pm 1.01}$	$52.91_{\pm 1.13}$(↓ 41.6%)
50 pairs	$90.11_{\pm 0.81}$	$77.61_{\pm 1.10}$	$50.42_{\pm 1.45}$(↓ 44.0%)
10 pairs + imitation	$93.09_{\pm 0.30}$	$78.27_{\pm 0.87}$	$57.35_{\pm 1.28}$(↓ 38.4%)
10 pairs + imitation + vote	$93.43_{\pm 0.28}$	$79.07_{\pm 0.86}$	$57.88_{\pm 1.28}$(↓ 38.0%)
50 pairs + imitation	$93.57_{\pm 0.29}$	$82.52_{\pm 0.41}$	$60.34_{\pm 1.54}$(↓ 35.5%)
50 pairs + imitation + vote	$\mathbf{93.90}_{\pm \mathbf{0.28}}$	$\mathbf{83.24}_{\pm \mathbf{0.39}}$	$\mathbf{60.89}_{\pm \mathbf{1.55}}$(↓ **35.1**%)
1 pair + reset	$90.30_{\pm 0.45}$	$78.76_{\pm 0.31}$	$55.11_{\pm 1.25}$(↓ 39.0%)
10 pairs + reset	$90.60_{\pm 0.37}$	$76.42_{\pm 0.91}$	$53.68_{\pm 1.14}$(↓ 40.8%)
50 pairs + reset	$90.09_{\pm 0.83}$	$76.89_{\pm 1.24}$	$50.55_{\pm 1.55}$(↓ 44.0%)

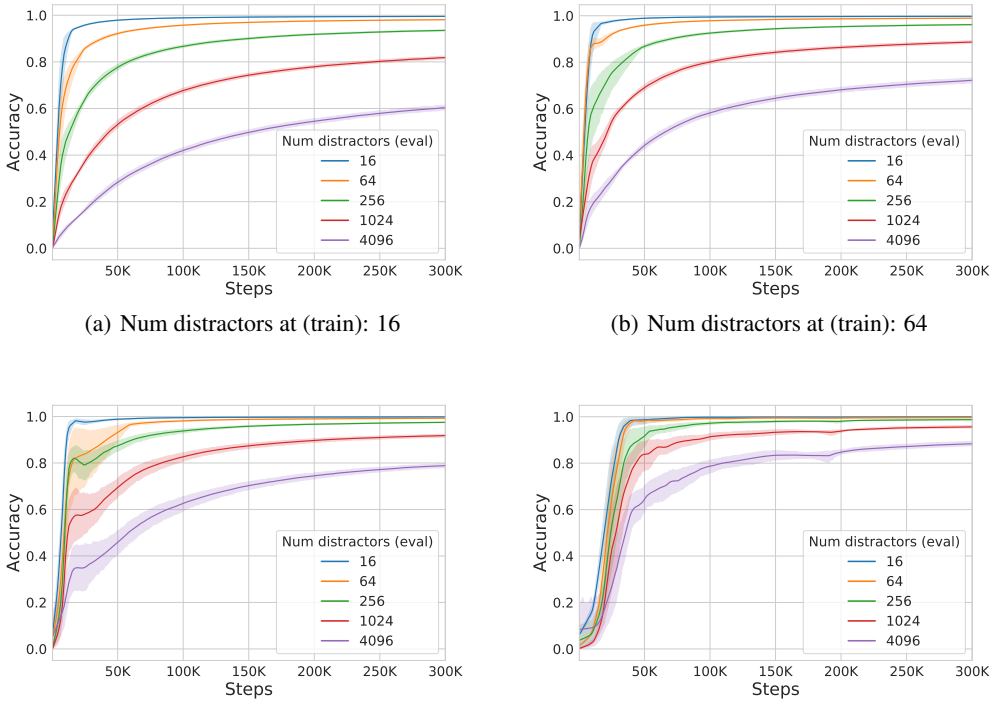


(a) *Discrimination* (b) *Classification of Identities* (c) *Classification of Attributes*

Figure 13: Ease of learning for the CelebA dataset of the emergent languages for different tasks. The results are averaged across the emergent languages of the different speakers (when it is generated in a population), newborn listeners' seeds, and across the 10 seeds of each setting. The shaded region represents the standard deviation.

Table 6: Topographic Similarity on CelebA dataset, in %. For each setting we report the mean over 10 seeds. ± denotes 1 SEM.

Setting	TopSim ↑
best 1 pair	12.08
1 pair	$12.47_{\pm 0.89}$
10 pairs	$14.38_{\pm 0.54}$
50 pairs	$10.57_{\pm 0.97}$
10 pairs + imitation	$13.13_{\pm 0.79}$
50 pairs + imitation	$10.75_{\pm 0.66}$
1 pair + reset	$13.88_{\pm 1.20}$
10 pairs + reset	$\mathbf{15.32}_{\pm \mathbf{0.64}}$
50 pairs + reset	$10.67_{\pm 0.90}$



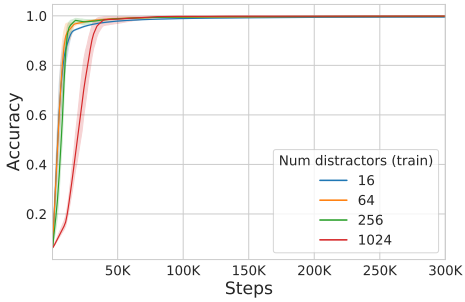
(a) Num distractors at (train): 16

(b) Num distractors at (train): 64

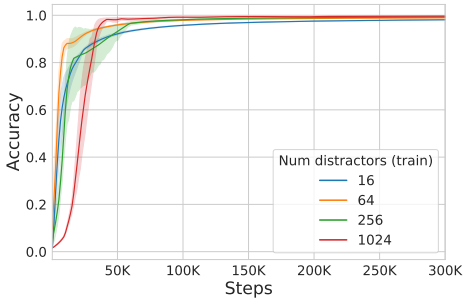
(c) Num distractors at (train): 256

(d) Num distractors at (train): 1024

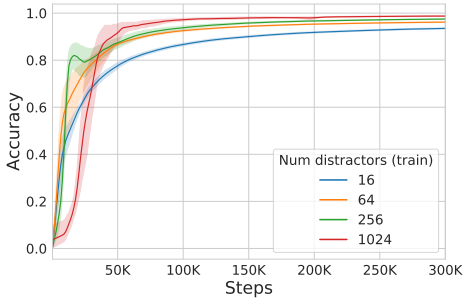
Figure 14: Test accuracy for different number of distractors at training time (subplot) and at evaluation time (lines) on ImageNet. As one would expect, given one model trained with $\mathcal{X}$ distractors, the more complex the evaluation task, the lower is the accuracy.



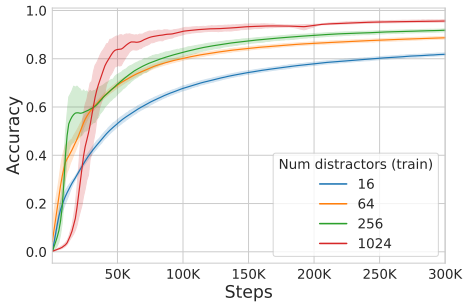
(a) Num distractors at (eval): 16



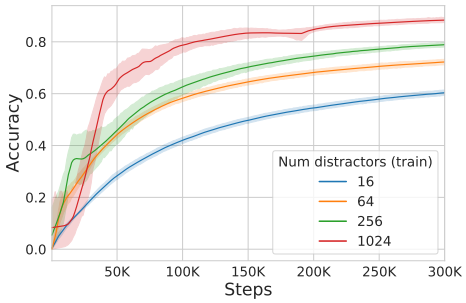
(b) Num distractors at (eval): 64



(c) Num distractors at (eval): 256



(d) Num distractors at (eval): 1024



(e) Num distractors at (eval): 4096

Figure 15: Test accuracy for different number of distractors at training time (lines) and at evaluation time on ImageNet. The more we complexify evaluation, the easier it is too discriminate among representations.