Entropy minimization in emergent languages

Eugene Kharitonov, Rahma Chaabouni, Diane Bouchacourt, Marco Baroni

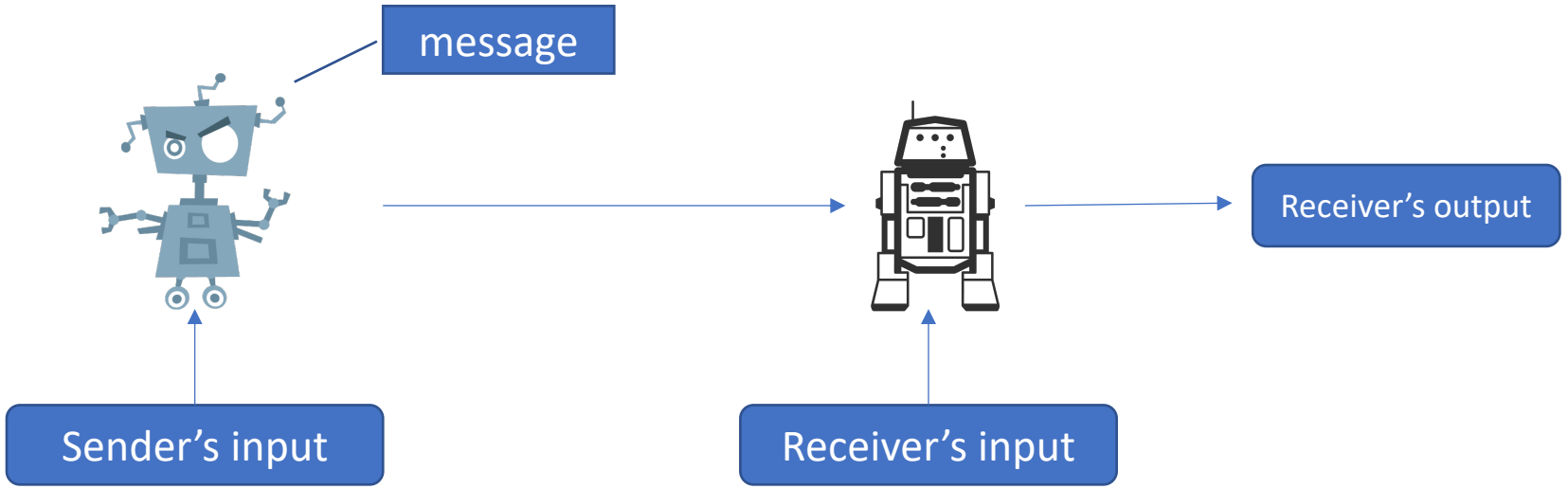
FACEBOOK Artificial Intelligence

ICREA

LSCP Laboratoire de Sciences Cognitives et Psycholinguistique

Setup: signalling game (Lewis, 1969)

- Two deterministic neural agents, Sender and Receiver, solving a task collaboratively
- Each has its own individual input
- Sender sends a discrete message (one- or multi-symbol) to Receiver
- Based on its own input and the message, Receiver performs an action

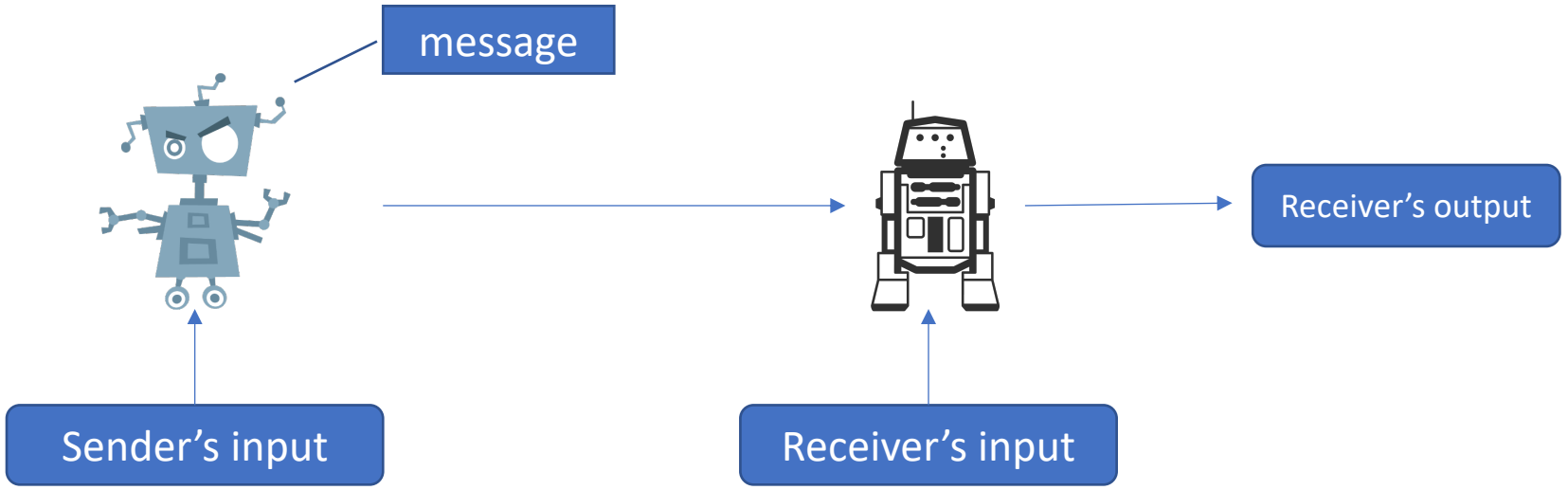


Setup: signalling game (Lewis, 1969)

- The goal is for Receiver to perform some task
- Both agents get the same reward that depends on Receiver's action
- No supervision on the emergent protocol

Motivated by

- developing agents that are able to communicate with humans (Mikolov et al., 2016)
- Better understanding natural language itself (Hurford, 2014)



Setup

Suppose Receiver has only a part of the information required to perform a task, while Sender has all available information

Two opposite scenarios of successful communication:

- Sender tries to transmit all the information in its message
  - “Complex” protocol, encodes a lot of information
- Sender only sends what Receiver lacks
  - “Simple” protocol, encodes the required mininum

We measure complexity of the protocol by its entropy

Data processing inequalities (discrete inputs)

Processing its input, Sender non-increases entropy

Conditioning does not increase entropy

Again, applying a function does not increase the entropy

When task is solved, outputs ***o*** are (almost) equal to ground-truth ***l***

$$H(i_s) \geq H(S(i_s))$$
$$= H(m) \geq H(m|i_r)$$
$$\geq H(R(m, i_r)|i_r) = H(o|i_r)$$
$$\approx H(l|i_r)$$

Entropy of the messages is bounded between entropy of Sender's inputs and the amount of information that Receiver needs to solve the task

Q: How complex the communication protocol would be?

Why is this question interesting?

Efficiency pressures are frequently observed in language and other biological communication systems (Ferrer i Cancho et al., 2013; Gibson et al., 2019)

- Color naming: for a given accuracy, lexicon complexity is minimized (Zaslavsky et al., 2018, 2019)

Why is this question interesting?

Would something similar happen when two agents are communicating with each other?

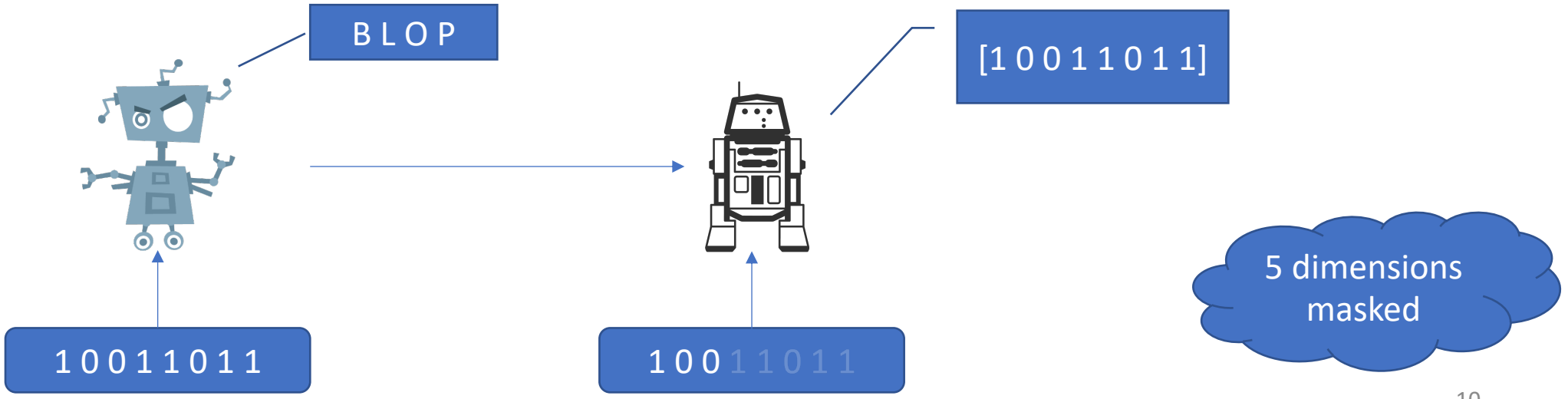
- Can it be a general property of discrete communication systems?
- Can it have some beneficial properties?

Methodology

- We build two games, that allow us to vary the amount of information Receiver needs to perform a task
- We achieve that in two ways:
  - By controlling the amount of information Receiver has as its own input
  - By controlling the complexity of the task itself, via changing the entropy of the ground-truth outputs

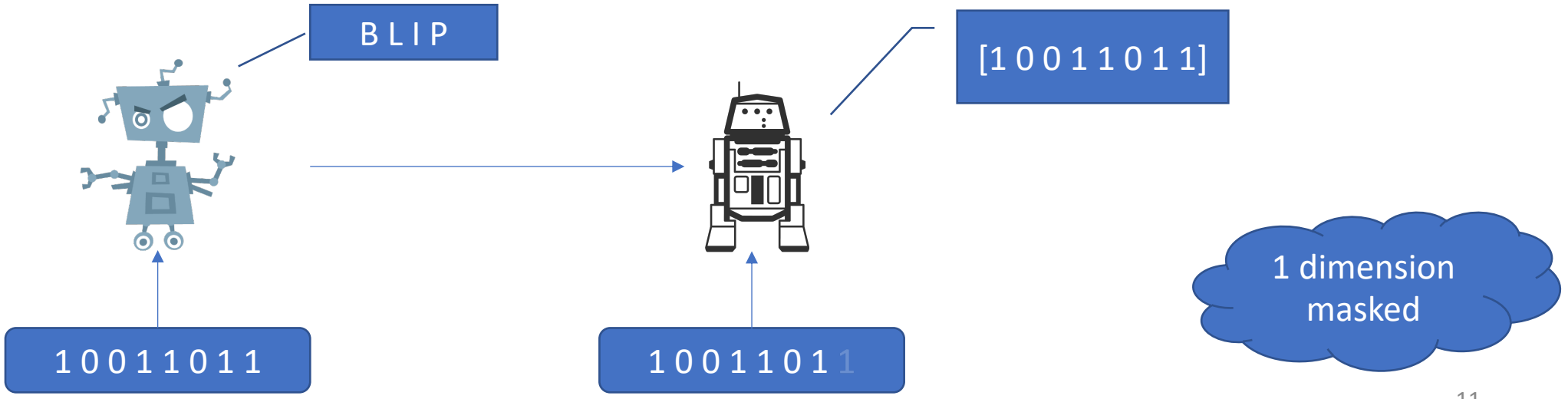
Game 1: Guess Number

- Sender gets a 8-dim binary vector an input
  - components are i.i.d. Bernouilli variables
- Receiver gets the same vector, but only k (0 … 8) dimensions are not masked
- Goal is to recover the original vector



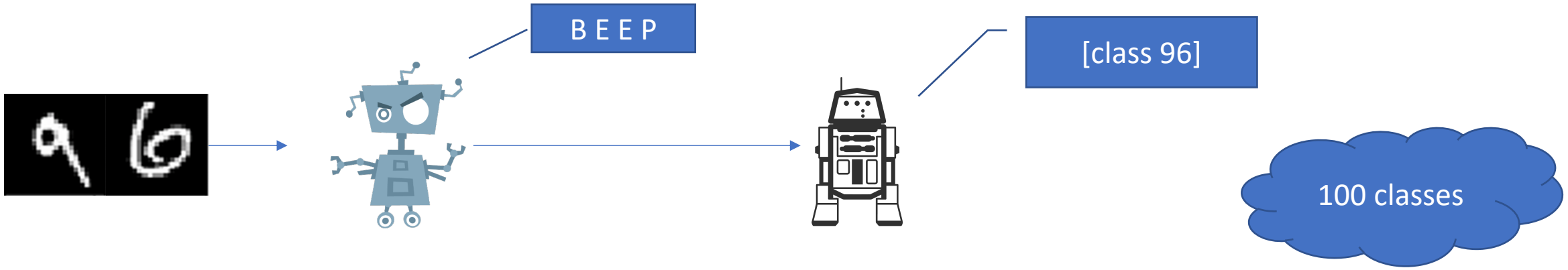
Game 1: Guess Number

- Sender gets a 8-dim binary vector an input
  - components are i.i.d. Bernouilli variables
- Receiver gets the same vector, but only k (0 … 8) dimensions are not masked
- Goal is to recover the original vector



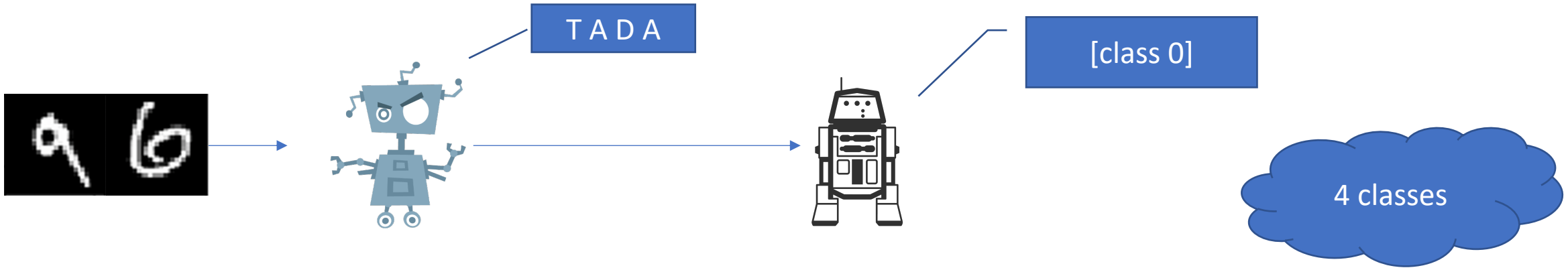
Game 2: Image Classification

- Sender gets two concatenated MNIST images, representing a two-digit number (00 … 99) (uniformly sampled from MNIST train data)
- Numbers are split in 2, 4, 10, 20, 25, 50, 100 equally sized classes
- Receiver has no side input
- Agents' goal is for Receiver to output the class



Game 2: Image Classification

- Sender gets two concatenated MNIST images, representing a two-digit number (00 … 99) (uniformly sampled from MNIST train data)
- Numbers are split in 2, 4, 10, 20, 25, 50, 100 equally sized classes
- Receiver has no side input
- Agents' goal is for Receiver to output the class



Methodology

We experiments with:

- Different architectures of agents,
- Different lengths of the messages & vocabulary size,
- Different approaches for learning with the discrete channel:
  - Gumbel-Softmax relaxation (Maddison et al., 2016; Jang et al., 2016),
  - REINFORCE for training both agents (Williams, 1992),
  - SCG: REINFORCE for Sender + standard backpropagation for Receiver (Stochastic Computational Graph) (Schulman et al., 2015)

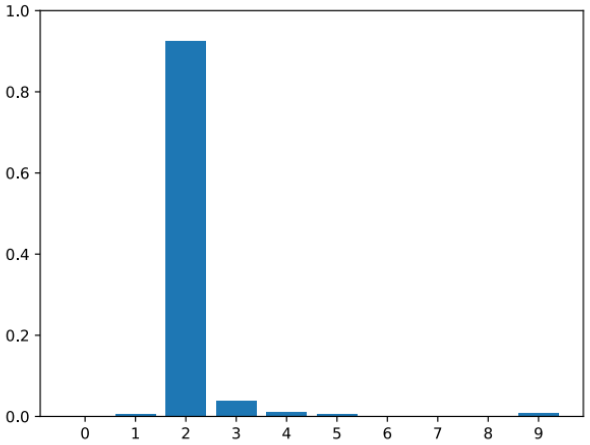
- We vary hyperparameters/seeds and select the game instances where agents are sucessful in solving the task
  - Game success rate: 20% REINFORCE, 50..75% of Gumbel-Softmax and SCG

- Measure entropy of the discrete protocol

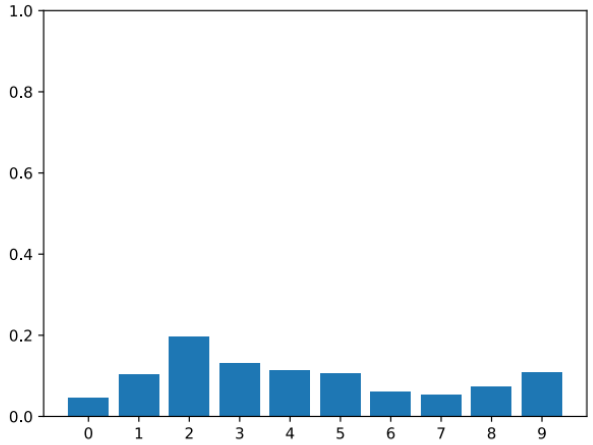
Gumbel-Softmax relaxation

- Closer approximates discrete messages as temperature gets lower
- Allows to “interpolate” between discrete and continuous

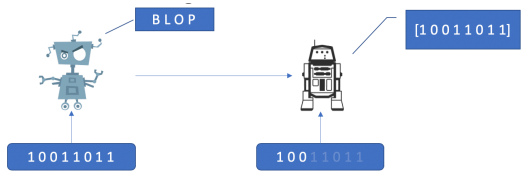
temperature (τ) at 0.25



temperature (τ) at 2



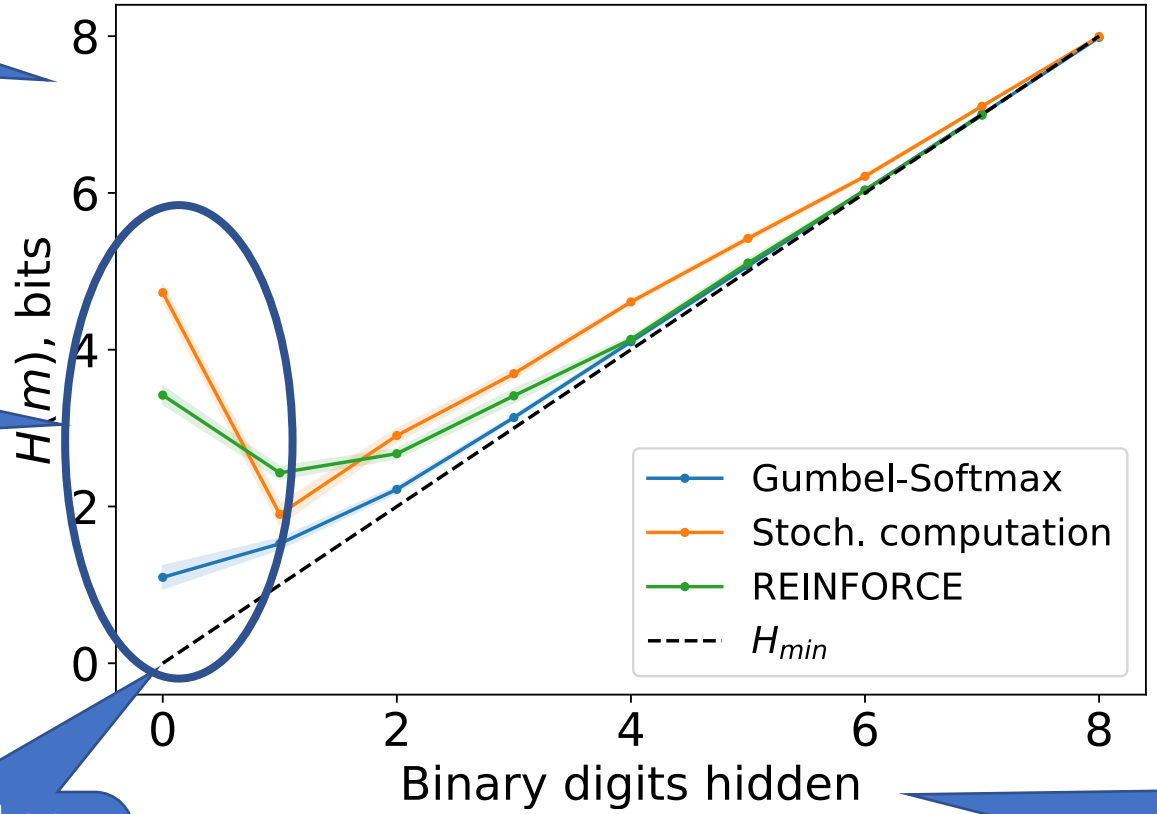
Results: Guess Number



Entropy of the messages

Degenerate case of non-communication

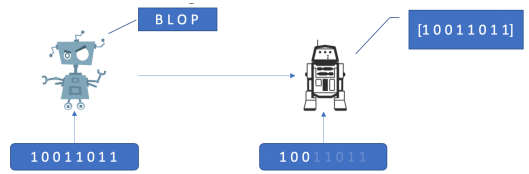
Lower bound on the information required for solving the task

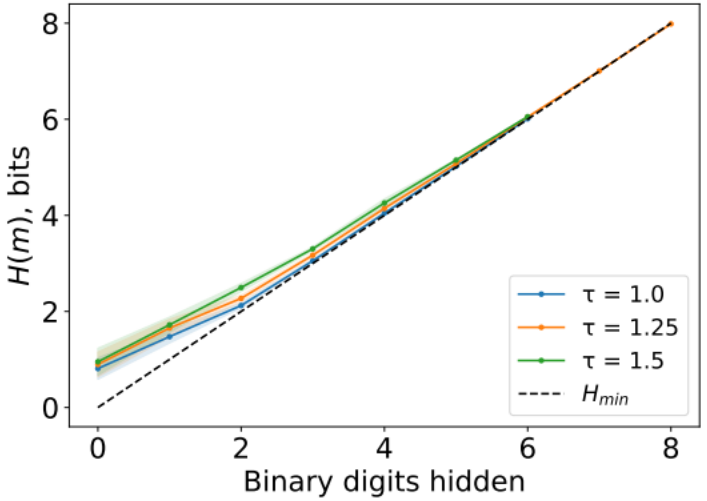


Upper bound on the entropy: 8 bits

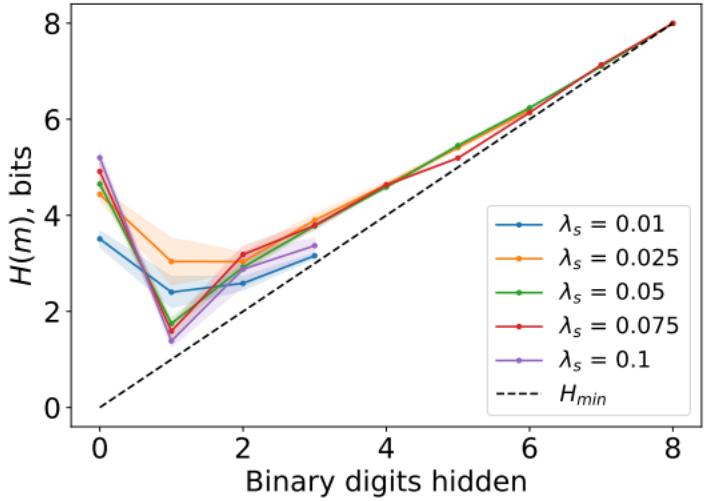
How much information Receiver needs to perform the task

Results: Guess Number

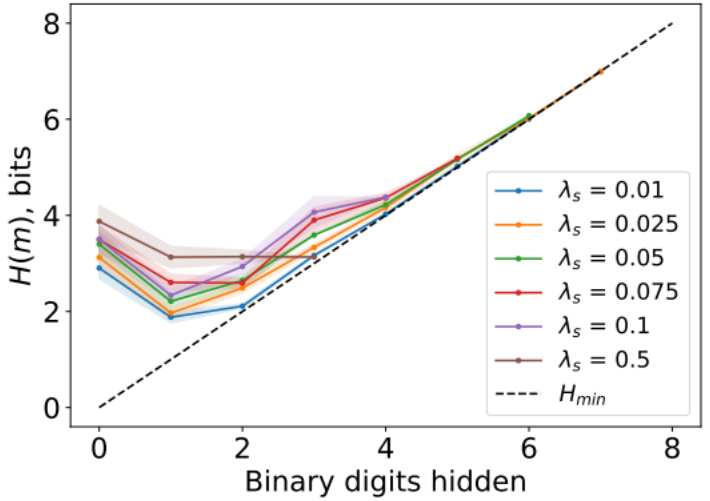




(a) Guess Number, Gumbel-Softmax relaxation.

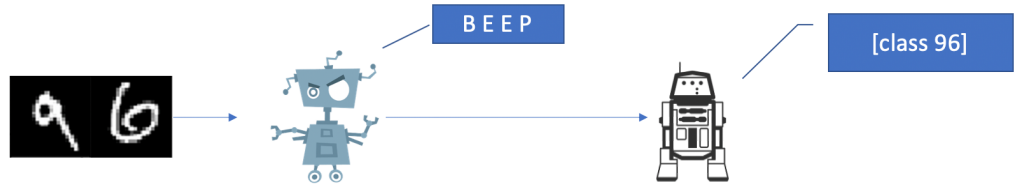


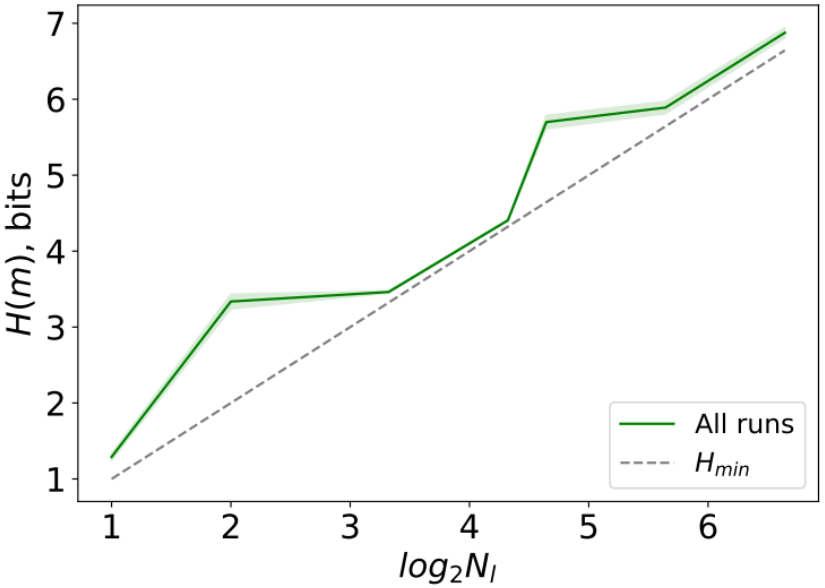
(b) Guess Number, Stochastic Computation Graph.



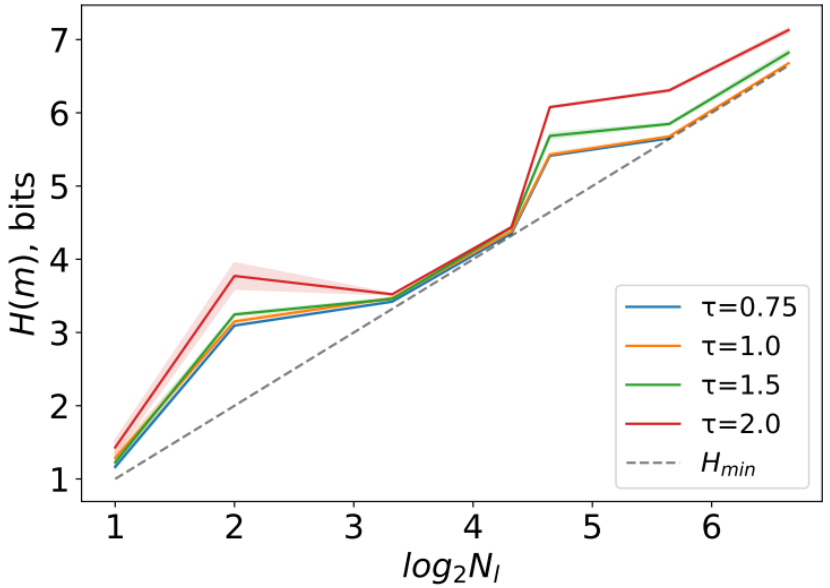
(c) Guess Number, REINFORCE.

Results: Image Classification





(a) Successful runs pooled together.



(b) Successful runs grouped by temperature.

Upper bound on the entropy: 10 bits

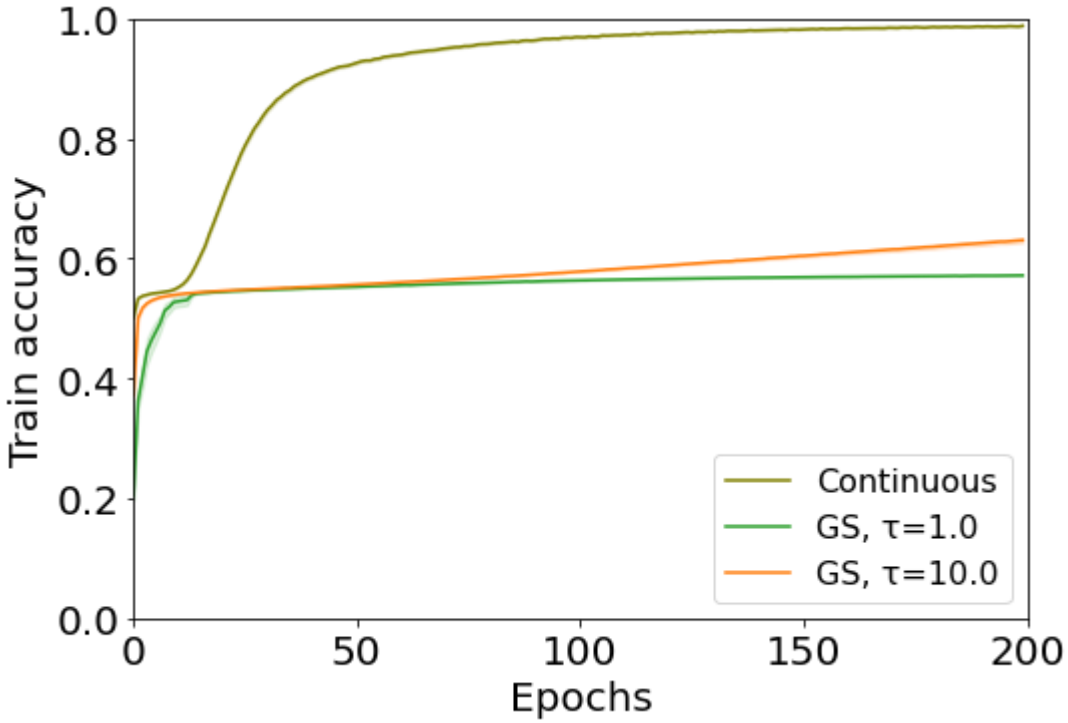
Entropy Minimization

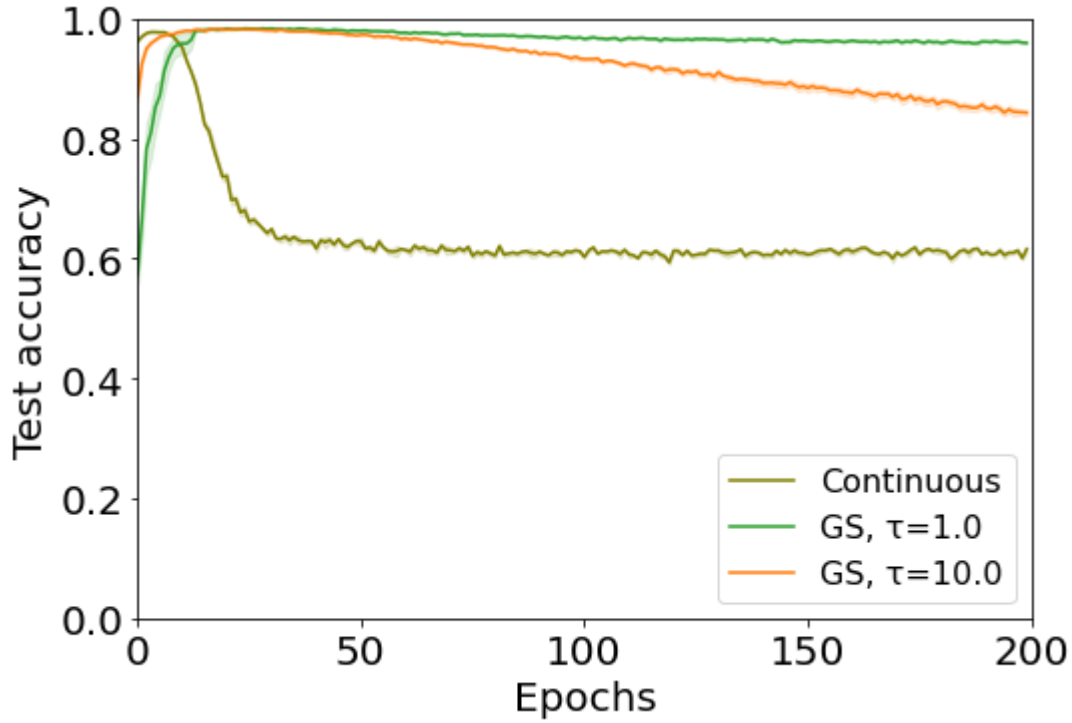
- The agents only develop protocols with higher entropy when this is necessary
- Entropy approaches the lower bound

- Does discrete channel have other *desirable* properties?
  - Robustness to overfitting

Results: Robustness

- Image Classification (10 classes): shuffle labels for random ½ of the digit images





Our findings

The entropy of the protocol consistently approaches the lower bound that still allows to solve a task

- In other words, the agents develop the simplest protocol they can get away with, while still solving the task

The level of discreteness of this protocol impacts the tightness of this approximation

Discrete channel has useful properties:

- Robustness to overfitting random labels
- Robustness against adversarial attacks (see the paper)

Why is it interesting?

Efficiency pressures arise in artificial discrete communication systems

- A common cause - hardness of discrete communication?

Discrete protocols have useful properties

- Good reasons for agents to communicate in a discrete language
- That’s why (human) language is discrete in the first place?

Why is it interesting?

Agents wouldn’t develop complex languages (protocols) unless that is necessary

- Echoes earlier findings in the literature (Bouchacourt & Baroni, 2018)
- If we want agents to develop complex languages, we should make sure that is absolutely required

Thank you!