Developmentally motivated emergence of compositional communication via template transfer

Tomasz Korbak
Institute of Philosophy and Sociology
Polish Academy of Sciences
tomasz.korbak@gmail.com

Julian Zubek
Human Interactivity and Language Lab
Faculty of Psychology,
University of Warsaw, Poland
zubekj@gmail.com

Łukasz Kuciński
Institute of Mathematics
Polish Academy of Sciences
lukasz.kucinski@gmail.com

Piotr Miłoś
Institute of Mathematics
Polish Academy of Sciences,
deepsense.ai
pmilos@mimuw.edu.pl

Joanna Rączaszek-Leonardi
Human Interactivity and Language Lab
Faculty of Psychology,
University of Warsaw, Poland
joanna.leonardi@gmail.com

Abstract

This paper explores a novel approach to achieving emergent compositional communication in multi-agent systems. We propose a training regime implementing *template transfer*, the idea of carrying over learned biases across contexts. In our method, a sender–receiver pair is first trained with disentangled loss functions and then the receiver is *transferred* to train a new sender with a standard loss. Unlike other methods (e.g. the obverter algorithm), our approach does not require imposing inductive biases on the architecture of the agents. We experimentally show the emergence of compositional communication using topographical similarity, zero-shot generalization and context independence as evaluation metrics. The presented approach is connected to an important line of work in semiotics and developmental psycholinguistics: it supports a conjecture that compositional communication is scaffolded on simpler communication protocols.

1 Introduction

Language-like communication protocols can arise in environments that require agents to share information and coordinate behavior (Foerster et al., 2016; Lazaridou et al., 2016; Jaques et al., 2018). A crucial feature of human languages and some animal communication systems is *compositionality*: there are complex signals constructed through the combination of signals. Furthermore, compositionality is essential for general intelligence because it facilitates generalization —adaptability to novel situations—and productivity—an infinite number of meanings can be created using a finite set of primitives (Lake et al., 2016).

The contribution of this paper is its demonstration that communication protocols exhibiting compositionality can emerge via adaptation of pre-existing, simpler non-compositional protocols to a new environment. This procedure is an instance of *template transfer* (Barrett and Skyrms, 2017).

Preprint. Under review.

We are motivated in this work by semiotic theoretical framework and empirical research in language development. Deacon (1998), after Peirce (1998) presented a semiotic framework detailing how (i) iconic and (ii) indexical communication protocols precede (iii) complex compositional communication protocols. Similarly, language development research demonstrates that children learn to speak compositionally in a structured social environment designed for teaching progressively more complex utterances through simple language games (Stern, 1974; Bruner, 1983; Nomikou et al., 2017; Rączaszek-Leonardi et al., 2018).

Our model implements the idea of template transfer by sharing agents across games of varying complexity. We decompose learning compositional communication into three phases: (i) learning a visual classifier, (ii) learning non-compositional communication protocols, and (iii) learning a compositional communication protocol. This decomposition closely follows distinctions established in semiotics and is more plausible in the light of human language development than other approaches. Crucially, the biases learned in simple games in phase (ii) are sufficient to incentivize a compositional communication protocol to emerge in phase (iii). We compare the template transfer approach with other method of achieving compositionality—the obverter algorithm (Batali, 1998; Choi et al., 2018)—on three different metrics: zero-shot generalization, context independence and topographical similarity. The results demonstrate that the ability to communicate compositionality can emerge in a model less cognitively demanding than the obverter approach. The key idea—that complex communication protocols are easier to learn when bootstrapped on pre-existing simpler protocols—may generalize to other problems in multi-agent communication.

2 Related work

Recent work on emergent communication in artificial intelligence shows that compositionality requires strong inductive biases to be imposed on communicating agents (Kottur et al., 2017). One recurring idea is placing pressure on agents to use symbols consistently across varying contexts. To that end, Kottur et al. (2017) and Das et al. (2017) reset the memory of an agent between producing or receiving subsequent symbols, which helps to obtain a consistent symbol grounding. A more psychologically plausible approach is explored by Choi et al. (2018) and Bogin et al. (2018), who take inspiration from the obverter algorithm (Oliphant and Batali, 1997; Batali, 1998).

The obverter (from the Latin *obverto*, to turn towards) algorithm (Batali, 1998; Oliphant and Batali, 1997) is based on the assumption that an agent can use its own responses to messages to predict other agent's responses, and thus can iteratively compose its messages to maximize the probability of the desired response. A limitation of the obverter is that it makes strong assumptions about the agents and task: to be able to use themselves as models of others, the agents must share an identical architecture and the task must be symmetric (the agents must be able to exchange their roles). This excludes games with functional specialization of agents.

A different family of approaches, more similar in spirit to the approach described in this work, includes population-based training that incentivizes the creation of communication protocols that are easy to teach to new agents (Brighton, 2002; Li and Bowling, 2019) and gradually increasing task complexity that incentivizes reusing existing patterns of communication (De Beule and Bergen, 2006). Contributing to this line of thinking, we show how the history of acquiring simpler communication protocols may lead to more complex types of communication.

3 Method

In this section we present two Lewis signaling games (Lewis, 2011; Skyrms, 2010). The first is a standard *object naming game*. Based on it, we propose a new game that incorporates the idea of template transfer (Barrett and Skyrms, 2017).

Object naming game Two agents, a sender and a receiver, learn to communicate about colored geometric objects. The sender observes an object (an RGB image) and sends a message (a sequence of discrete symbols) to the receiver; the receiver must correctly indicate both the color and the shape of the object. Formally, the game is stated as maximization of the following log likelihood:

$$\mathcal{L}(\theta, \psi) := \mathbb{E}_x \mathbb{E}_{m \sim s_\theta(\cdot|x)}[-\log r_\psi(x_c, x_s|m)],$$

where s_θ is the policy of the sender (i.e. $s_\theta(m|x)$ is the probability of sending message m when observing image x), r_ψ is the policy of the receiver, x is the representation of the object and x_c and

x_s are its color and shape, respectively. Parameters θ and ψ are learnable parameters of the polices. For more details, see Algorithm 1 in the Appendix.

Template transfer game This game consists of two phases. In the first, two senders communicate with one receiver in two sub-games. The sub-games are disentangled in the sense that their tasks are to correctly indicate one aspect of the object (color or shape), as formalized by the following loss functions:

$$\mathcal{L}_1(\theta_1, \psi) := \mathbb{E}_x \mathbb{E}_{m \sim s_{\theta_1}(\cdot|x)}[-\log r_\psi(x_c|m)], \quad \mathcal{L}_2(\theta_2, \psi) := \mathbb{E}_x \mathbb{E}_{m \sim s_{\theta_2}(\cdot|x)}[-\log r_\psi(x_s|m)],$$

where $r_\theta(x_c|m)$ is the marginalization of $r_\theta(x_c, x_s|m)$, viz. $r_\theta(x_c|m) := \sum_{x_s} r_\theta(x_c, x_s|m)$. Analogously, one can define $r_\theta(x_c|m)$.

These two losses are optimized simultaneously (crucially with the shared parameters ψ of the receiver) until a desired level of accuracy is met. Then, the second phase follows, in which the receiver is passed (via template transfer) to the object naming game (as described in the previous paragraph) with a new sender. See Figure 1 and Algorithm 2 in the Appendix for more details.

The communication protocol acquired in the first phase serves as a training bias in the second phase. Informally, the new sender learns to emulate messages sent by the two specialized senders of the previous phase. Our experiments (Section 4) indicate that two-phase learning is a sufficient incentive for compositionality to emerge.

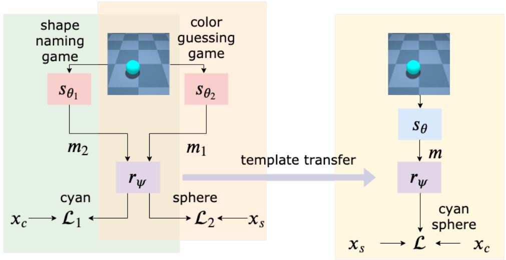


Figure 1: Template transfer consists of pre-training the receiver r_ψ on two games with disentangled losses $\mathcal{L}_1$ and $\mathcal{L}_2$ and transferring r_ψ to a new object naming game.

Agents Both the sender and the receiver are implemented as recurrent neural networks. The sender is equipped with a pre-trained convolutional neural network to process visual input. After observing the object, the sender generates a sequence of T discrete messages sampled from a closed vocabulary of 10 symbols. During the object naming game, $T = 2$, while during the template transfer game, $T = 1$. To prevent distribution shift with respect to message length between games, a random uniformly sampled symbol is prepended to s_1's messages and appended to s_2's messages. Due to the discrete nature of the communication channel, Gumbel–Softmax relaxation (Maddison et al., 2016; Jang et al., 2016) is used to backpropagate through $s_\theta(m|x)$. For details concerning the architecture and hyperparameters, please see the Appendix.

4 Experiments and results

Measuring compositionality We utilize three metrics of compositionality of a communication protocol: *zero-shot generalization accuracy*, *context independence* and *topographical similarity*. During evaluation we use the deterministic sender given by $s(x) := \arg\max_m s_\theta(m|x)$, where x is an object.

We quantify zero-shot generalization by measuring the accuracy of the agents on a test set, containing pairs of shapes and colors not present in the training set.

Context independence was introduced by Bogin et al. (2018) as a measure of alignment between the symbols in an agent's messages and the concepts transmitted. We denote the set of symbols used to compose messages by V and by K the set of concepts, which in our case is the union of available colors and shapes. Given sender s, and assuming a uniform distribution of objects, we define $p(v|k)$ as the probability that symbol $v \in V$ appears when the sender observes an object with property $k \in K$. We define $p(k|v)$ in the same manner. Further, let $v^k := \arg\max_v p(k|v)$. The context independence metric is defined as $\mathbb{E}(p(v^k|k) \cdot p(k|v^k))$; the expectation is taken with respect to the uniform distribution on K. Context independence is sometimes considered restrictive, as it may be low in the presence of synonyms (Lowe et al., 2019).

Finally, we introduce topographical similarity (Brighton and Kirby, 2006; Lazaridou et al., 2018), also known as *representational similarity* (Kriegeskorte, 2008; Bouchacourt and Baroni, 2018). We denote the random variable $L_t := L((x_c^1, x_s^1), (x_c^2, x_s^2))$, where L is the Levenshtein (1966) distance and x^1 and x^2 are independently sampled objects with the subscripts denoting their shapes and colors. Note that in our case $L_t \in \{0, 1, 2\}$. Let $L_m := L(s(x^1), s(x^2))$ be the distance between messages sent by the sender after observing x^1 and x^1. Topographical similarity is the the Spearman ρ correlation of L_t and L_m.

The metrics described above can be regarded as measures of compositionality. Indeed, high zero-shot generalization indicates that the agents correctly map the implicit compositional structure of inputs to explicate one of the outputs. The other two metrics focus directly on the transmitted messages, comparing them to the ground truth, fully disentangled (color, shape) representation.

Obverter baseline In the obverter algorithm, two agents exchange the roles of the sender and the receiver. If an agent is the receiver, it behaves as in the object naming game. If an agent is the sender, it sends message that would have produced the most accurate prediction of color and shape, if it had received such a message as a receiver (i.e. instead of the greedy decoding used in the original implementation of Batali (1998), we simply choose the message maximizing accuracy). Accuracy is evaluated against the predictions of the visual classifier. For details, consult Algorithm 3 in the Appendix.

The effect of template transfer on compositionality We compared our approach with several baselines (random, the same architecture without pre-training games, and our implementation of the obverter approach) on games with five shapes and five colors[1]. Topographical similarity and context independence were computed on the full dataset (train and test); the results are presented in Table 1. Template transfer clearly leads to highly compositional communication protocols. While all methods struggled to generalize to unseen objects, template transfer was the most successful. For examples of communication protocols representative of the experiments conducted, see Table 2.

Table 1: The effect of template transfer on compositionality. The metrics are train and test set accuracies (the rate of correctly predicted both x_c and x_s); average over the individual accuracies for x_c and x_s; and context independence (CI) and topographical similarity (Topo). The models are random baseline (untrained agents); baseline architecture (without template transfer); template transfer (TT); and obverter algorithm. All reported metrics are averaged over ten random seeds and standard deviations are reported in brackets.

	Accuracy				
Model	Train (both)	Test (both)	Test (avg)	CI	Topo
Random	0.04	0.04	0.2	0.04 (± 0.01)	0.13 (± 0.03)
Baseline	0.99 (± 0.01)	0.02 (± 0.05)	0.47 (± 0.09)	0.08 (± 0.01)	0.30 (± 0.05)
Obverter	0.99 (± 0)	0.24 (± 0.23)	0.51 (± 0.19)	0.12 (± 0.02)	0.55 (± 0.13)
TT (ours)	1 (± 0)	0.48 (± 0.10)	0.74 (± 0.06)	0.18 (± 0.01)	0.85 (± 0.03)

5 Conclusions

Template transfer — transferring skills from simpler to more complex Lewis signaling games — can be used to model a variety of semiotic, social and cognitive phenomena (Barrett and Skyrms, 2017; Barrett et al., 2019, 2018). In this paper we demonstrated that compositional communication can emerge via template transfer, presenting a model-free approach more general than the obverter algorithm. The one assumption that we make is that the loss function can be decomposed into two disentangled loss functions, as in the case of decomposing $\mathcal{L}$ into $\mathcal{L}_1$ and $\mathcal{L}_2$. (Note that there is no need for inputs to be disentangled.) This shows that biases necessary for compositional communication can be learned rather than imposed by design.

[1] The code accompanying this paper is anonymously released on GitHub under `https://github.com/sc2PjMS8/compositional-communication-via-template-transfer`.

Table 2: Two example communication protocols, one that emerged via the baseline architecture (2a), and one via template transfer (2a). Gray cells indicate objects not seen during training. In (2b), symbols exhibit clear association with colors and shapes, e.g. symbol 8 is consistently associated with the color magenta (when on first position) and boxes (when on second position).

(a) A non-compositional communication protocol (topographical similarity 0.25)

	box	sphere	cylinder	torus	ellipsoid
blue	1 0	4 5	1 0	4 5	5 0
cyan	9 0	4 0	3 0	4 0	7 0
gray	3 5	6 5	3 2	6 5	5 3
green	0 0	7 6	3 0	6 0	7 6
magenta	1 5	5 5	1 2	1 5	5 2

(b) A highly compositional communication protocol (topographical similarity 0.85)

	box	sphere	cylinder	torus	ellipsoid
blue	1 8	1 9	1 5	1 6	1 4
cyan	4 8	4 9	4 5	4 6	4 4
gray	6 8	6 9	6 5	6 6	6 9
green	9 8	9 9	9 5	9 6	9 4
magenta	8 8	8 9	8 5	8 8	8 4

References

Barrett, J., Skyrms, B., and Cochran, C. (2018). Hierarchical Models for the Evolution of Compositional Language.

Barrett, J. A. and Skyrms, B. (2017). Self-assembling Games. The British Journal for the Philosophy of Science, 68(2):329–353.

Barrett, J. A., Skyrms, B., and Mohseni, A. (2019). Self-Assembling Networks. The British Journal for the Philosophy of Science, 70(1):301–325.

Batali, J. (1998). Computational simulations of the emergence of grammar. Approach to the Evolution of Language, pages 405–426.

Bogin, B., Geva, M., and Berant, J. (2018). Emergence of Communication in an Interactive World with Consistent Speakers. arXiv:1809.00549 [cs]. arXiv: 1809.00549.

Bouchacourt, D. and Baroni, M. (2018). How agents see things: On visual representations in an emergent language game. arXiv:1808.10696 [cs]. arXiv: 1808.10696.

Brighton, H. (2002). Compositional Syntax From Cultural Transmission. Artificial Life, 8(1):25–54.

Brighton, H. and Kirby, S. (2006). Understanding Linguistic Evolution by Visualizing the Emergence of Topographic Mappings. Artificial Life, 12(2):229–242.

Bruner, J. S. (1983). Child's talk: learning to use language. W.W. Norton, New York, 1st ed edition.

Choi, E., Lazaridou, A., and de Freitas, N. (2018). Compositional Obverter Communication Learning From Raw Visual Input. ICLR 2018. arXiv: 1804.02341.

Das, A., Kottur, S., Moura, J. M. F., Lee, S., and Batra, D. (2017). Learning Cooperative Visual Dialog Agents with Deep Reinforcement Learning. 2017 IEEE International Conference on Computer Vision (ICCV), Venice, 2017. arXiv: 1703.06585.

De Beule, J. and Bergen, B. K. (2006). On the emergence of compositionality. In The Evolution of Language, pages 35–42, Rome, Italy. World scientific.

Deacon, T. W. (1998). The symbolic species: the co-evolution of language and the brain. Norton, New York, NY, norton paperback edition. OCLC: 254499872.

Foerster, J. N., Assael, Y. M., de Freitas, N., and Whiteson, S. (2016). Learning to Communicate with Deep Multi-Agent Reinforcement Learning. NIPS'16 Proceedings of the 30th International Conference on Neural Information Processing Systems. arXiv: 1605.06676.

Jang, E., Gu, S., and Poole, B. (2016). Categorical Reparameterization with Gumbel-Softmax. arXiv:1611.01144 [cs, stat]. arXiv: 1611.01144.

Jaques, N., Lazaridou, A., Hughes, E., Gulcehre, C., Ortega, P. A., Strouse, D. J., Leibo, J. Z., and de Freitas, N. (2018). Social Influence as Intrinsic Motivation for Multi-Agent Deep Reinforcement Learning. arXiv:1810.08647 [cs, stat]. arXiv: 1810.08647.

Kharitonov, E., Chaabouni, R., Bouchacourt, D., and Baroni, M. (2019). EGG: a toolkit for research on Emergence of lanGuage in Games. arXiv:1907.00852 [cs]. arXiv: 1907.00852.

Kingma, D. P. and Ba, J. (2014). Adam: A Method for Stochastic Optimization. arXiv:1412.6980 [cs]. arXiv: 1412.6980.

Kottur, S., Moura, J. M. F., Lee, S., and Batra, D. (2017). Natural Language Does Not Emerge 'Naturally' in Multi-Agent Dialog. arXiv:1706.08502 [cs]. arXiv: 1706.08502.

Kriegeskorte, N. (2008). Representational similarity analysis – connecting the branches of systems neuroscience. Frontiers in Systems Neuroscience.

Lake, B. M., Ullman, T. D., Tenenbaum, J. B., and Gershman, S. J. (2016). Building Machines That Learn and Think Like People. arXiv:1604.00289 [cs, stat]. arXiv: 1604.00289.

Lazaridou, A., Hermann, K. M., Tuyls, K., and Clark, S. (2018). Emergence of Linguistic Communication from Referential Games with Symbolic and Pixel Input. arXiv:1804.03984 [cs]. arXiv: 1804.03984.

Lazaridou, A., Peysakhovich, A., and Baroni, M. (2016). Multi-Agent Cooperation and the Emergence of (Natural) Language. arXiv:1612.07182 [cs]. arXiv: 1612.07182.

Levenshtein, V. I. (1966). Binary Codes Capable of Correcting Deletions, Insertions and Reversals. Soviet Physics Doklady, 10:707.

Lewis, D. K. (2011). Convention: a philosophical study. Blackwell, Oxford, nachdr. edition. OCLC: 837747718.

Li, F. and Bowling, M. (2019). Ease-of-Teaching and Language Structure from Emergent Communication. arXiv:1906.02403 [cs]. arXiv: 1906.02403.

Lowe, R., Foerster, J., Boureau, Y.-L., Pineau, J., and Dauphin, Y. (2019). On the Pitfalls of Measuring Emergent Communication. arXiv:1903.05168 [cs, stat]. arXiv: 1903.05168.

Maddison, C. J., Mnih, A., and Teh, Y. W. (2016). The Concrete Distribution: A Continuous Relaxation of Discrete Random Variables. arXiv:1611.00712 [cs, stat]. arXiv: 1611.00712.

Nomikou, I., Koke, M., and Rohlfing, K. J. (2017). Verbs in Mothers' Input to Six-Month-Olds: Synchrony between Presentation, Meaning, and Actions Is Related to Later Verb Acquisition. Brain Sciences, 7(12):52.

Oliphant, M. and Batali, J. (1997). Learning and the Emergence of Coordinated Communication. Center for Research on Language Newsletter, 11.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., and Lerer, A. (2017). Automatic differentiation in PyTorch.

Peirce, C. S. (1998). Collected papers of Charles Sanders Peirce. Thoemmes Press, Bristol, England. OCLC: ocm39692049.

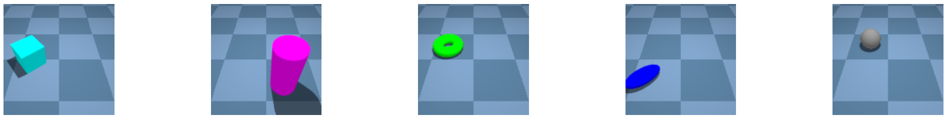
Rączaszek-Leonardi, J., Nomikou, I., Rohlfing, K. J., and Deacon, T. W. (2018). Language Development From an Ecological Perspective: Ecologically Valid Ways to Abstract Symbols. Ecological Psychology, 30(1):39–73.

Skyrms, B. (2010). Signals: evolution, learning, & information. Oxford University Press, Oxford ; New York. OCLC: ocn477256653.

Stern, D. N. (1974). The Goal and Structure of Mother-Infant Play. Journal of the American Academy of Child Psychiatry, 13(3):402–421.

A Dataset

We conduct our experiments on a dataset consisting of 2500 images of colored three-dimensional objects. Each image has dimensions of 128 x 128 x 3 pixels. The dataset includes images of five shapes (box, sphere, cylinder, torus, ellipsoid) and five colors (blue, cyan, gray, green, magenta). One hundred images generated using POV-Ray ray tracing engine,[2] differing in the position of the object on a surface, are included for each color–shape pair. (An analogous dataset was previously used by Choi et al. (2018) and Bogin et al. (2018).) We choose pairs for the test set by taking one of each figure and color, i.e. the test set is composed of blue boxes, cyan spheres, gray cylinders, green tori and magenta ellipsoids. Example images from the dataset are shown in Figure 2.



(a) Cyan box (b) Magenta cylinder (c) Green torus (d) Blue ellipsoid (e) Gray sphere

Figure 2: Examples of images from the dataset.

B Agent architecture

Vision module We pre-train a simple convolutional neural network on the training subset of our datatset to predict colors and shapes. The network is composed of two layers of filters (20 and 50 filters with kernel size $5x5$ and stride 1), each followed by a ReLU (rectified linear unit) activation and max pooling. The output of convolutional layers is then projected into a 25-dimensional image embedding using a fully-connected layer. During pre-training, the image embedding is passed to two linear classifiers (for color and shape) and the whole vision module is optimized with negative log likelihood as a cost function.

Sender During naming games, the vision module is kept frozen (i.e. it is not updated during training). The sender generates its messages using a single-layer recurrent neural network (RNN) with a hidden state size of 200. The 25-dimensional image embedding for each image is projected to 200 dimensions to initialize the hidden state of the RNN. Let T be a fixed length of the message. Then, at each time-step $t < T$, the output of the RNN is used to parameterize a Gumbel-Softmax distribution (together with a temperature τ that is a trainable parameter as well). A symbol is sampled from this distribution at each time-step t. After reaching T, the RNN halts and the generated symbols are concatenated to form a message, which is then passed to the receiver.

Receiver The receiver processes a message symbol-by-symbol using a single-layer recurrent neural network with a hidden state size of 200. After processing the entire sequence, the last output is passed to a two-layer neural network classifier with two softmax outputs for color and shape.

Hyperparameters All models are optimized using Adam (Kingma and Ba, 2014). The batch size is always 32. During the pre-traning phase of template transfer, both sender and receiver, as well as the vision classifier, are trained with learning rate 10^{-3}. During the object naming game, the sender is trained with learning rate 10^{-3} and receiver with learning rate 10^{-5}. During experiments with obverter, the receiver is trained with learning rate 10^{-5}.

C Implementations

Algorithms 1–3 describe the particular implementations of baseline training, template transfer pre-training and obverter used in the experiments. All experiments are implemented using PyTorch (Paszke et al., 2017) and EGG (Kharitonov et al., 2019).

[2] The dataset was generated using code available from `https://github.com/benbogin/obverter`.

Algorithm 1 Baseline training

```
Initialize sender $s_\theta$, receiver $r_\psi$, and training set $\mathcal{D}$
for $x, x_c, x_s \in \mathcal{D}$ do
    $m \sim s_\theta(x)$
    $\hat{x}_c, \hat{x}_s = r_\psi(m)$
    $\mathcal{L}$ = -log_likelihood($x_c$, $\hat{x}_c$) - log_likelihood($x_s$, $\hat{x}_s$)
    optimize($\mathcal{L}(\theta, \psi)$)
```

Algorithm 2 Template transfer

```
Initialize senders $s_{\theta_1}$, $s_{\theta_2}$, $s_\theta$, receiver $r_\psi$, and training set $\mathcal{D}$
for $x, x_c, x_s \in \mathcal{D}$ do
    $m_1 \sim s_{\theta_1}(x)$                                    ▷ Color naming game
    $m' \sim$ vocabulary
    $\hat{x}_c, \hat{x}_s = r_\psi([m_1, m'])$
    $\mathcal{L}_1$ = -log_likelihood($x_c$, $\hat{x}_c$)
    $m_2 \sim s_{\theta_2}(x)$                                    ▷ Shape naming game
    $m'' \sim$ vocabulary
    $\hat{x}_c, \hat{x}_s = r_\psi([m'', m_2])$
    $\mathcal{L}_2$ = -log_likelihood($x_s$, $\hat{x}_s$)
    optimize(($\mathcal{L}_1(\theta_1, \psi) + \mathcal{L}_2(\theta_2, \psi)$))
for $x, x_c, x_s \in X$ do
    $m \sim s_\theta(x)$                                          ▷ Object naming game
    $\hat{x}_c, \hat{x}_s = r_\psi(m)$
    $\mathcal{L}$ = -log_likelihood($x_c$, $\hat{x}_c$) - log_likelihood($x_s$, $\hat{x}_s$)
    optimize($\mathcal{L}(\theta, \psi)$)
```

D Loss derivation

We assume that we are given a dataset $\mathcal{D} = \left\{ (x^{(i)}, x_c^{(i)}, x_s^{(i)}) \right\}_{i=1}^{n}$, where entries are i.i.d., $x^{(i)}$ is an RGB image and $x_c^{(i)}$, $y_s^{(i)}$ are (ground truth) labels for $x^{(i)}$.

We assume that each data-point comes from a distribution (X, X_s, X_c, M), where X_s and X_c are two labels for image X and M is a latent variable.

We are interested in minimizing the negative log likelihood of ground truth labels given the image

$$\mathbb{E}_{(x,x_c,x_s)\sim\mathcal{D}}[-\log p(x_c, x_s|x)]. \tag{1}$$

We assume that (X_c, X_s) and X are conditionally independent given M, i.e. $p(x_s, x_c|x, m) = p(x_s, x_c|m)$. Consequently,

$$\log p(x_c, x_s|x) = \log \sum_m p(x_c, x_s|m)p(m|x) \geq \mathbb{E}_{m\sim p(\cdot|x)}[\log p(x_c, x_s|m)] \tag{2}$$

where the last inequality follows from Jensen's inequality. We will be optimizing the lower bound; hence, our surrogate loss function is

$$\mathbb{E}_{(x,x_c,x_s)\sim\mathcal{D}}\mathbb{E}_{m\sim p(\cdot|x)}[-\log p(x_c, x_s|m)]. \tag{3}$$

During template transfer pre-training (color naming game and shape naming game) we additionally assume that X_c is conditionally independent from X_s given X, i.e.

$$p_{\theta_1,\theta_2,\psi}(x_c, x_s|x) = p_{\theta_1,\psi}(x_s|x)p_{\theta_2,\psi}(x_c|x). \tag{4}$$

This implies that the surrogate loss is equal to

$$\mathbb{E}_{(x_c,x)\sim\mathcal{D}}[-\log p_{\theta_1,\psi}(x_c|x)] + \mathbb{E}_{(x_s,x)\sim\mathcal{D}}[-\log p_{\theta_2,\psi}(x_s|x)]. \tag{5}$$

Algorithm 3 Obverter

```
Initialize agents $a_1, a_2$, visual module $v$, training set $\mathcal{D}$
Initialize the set $M$ of all possible messages $m$
for $x, x_c, x_s \in \mathcal{D}$ do
    $s_\theta, r_\psi \sim \{a_1, a_2\}$        ▷ Randomly assigning the roles of sender and receiver
    $m = \arg\min_{m \in M}$ evaluate_message($s_\theta$, $m$)
    $\hat{y}_c, \hat{y}_s = r_\psi(m)$
    $\mathcal{L}$ = - log_likelihood($x_c$, $\hat{x}_c$) - log_likelihood($x_s$, $\hat{x}_s$)
    optimize($\mathcal{L}(\psi)$)
procedure EVALUATE_MESSAGE(model, $m$)
    $x_c, x_s = v(x)$        ▷ Using visual classifier predictions as a proxy for ground truth labels
    $\hat{x}_c, \hat{x}_s$ = model($m$)
    $\mathcal{L}'$ = - log_likelihood($x_c$, $\hat{x}_c$) - log_likelihood($x_s$, $\hat{x}_s$)
    return $\mathcal{L}'$
```

Furthermore, we assume that

$$p_{\theta_1,\psi}(x_c|x) = \sum_m s_{\theta_1}(m|x) r_\psi(x_c|m), \qquad p_{\theta_2,\psi}(x_s|x) = \sum_m s_{\theta_2}(m|x) r_\psi(x_s|m), \tag{6}$$

where both marginal distributions share the same r_ψ.

During the object naming game (the second phase of template transfer), we have

$$p_{\theta_3,\psi}(x_c, x_s|x) = \sum_m s_{\theta_3}(m|x) r_\psi(x_c, x_s|m), \tag{7}$$

where r_ψ is the same as in the color naming and shape naming games.