USING LOGIC TO EVOLVE MORE LOGIC
COMPOSING LOGICAL OPERATORS VIA SELF-ASSEMBLY

TRAVIS LACROIX

Abstract. I consider how complex logical operations might self-assemble in a signalling-game context via composition of simpler underlying dispositions. On the one hand, agents may take advantage of pre-evolved dispositions; on the other hand, they may co-evolve dispositions as they simultaneously learn to combine them to display more complex behaviour. In either case, the evolution of complex logical operations can be more efficient than evolving such capacities from scratch. Showing how complex phenomena like these might evolve provides an additional path to the possibility of evolving more or less rich notions of compositionality. This helps provide another facet of the evolutionary story of how sufficiently rich, human-level cognitive or linguistic capacities may arise from simpler precursors.

Keywords — Signalling Games; Logical Operations; Modular Composition

1. Signalling and Self-Assembly

The signalling game, presented in a classical context in Lewis (1969) and an evolutionary context in Skyrms (1996, 2010), has been well-studied. The basic sender-receiver model has been extended to shed light on a variety of philosophically interesting phenomena that arise in communicative contexts.[1] Barrett and Skyrms (2017) ask how such games might evolve in the first place. In some cases, actors may interact in such ways that can usefully be characterised *as* a game. This is the focus of their notion of *self-assembly*. Under this description, individuals with prior strategies for solving decision problems may come to interact. These interactions

Department of Logic and Philosophy of Science, University of California, Irvine
Mila, (Québec AI Institute / Institut Québécois d'Intelligence Artificielle)
E-mail address: `tlacroix@uci.edu`.
Date: Draft of October, 2019; *PLEASE CITE PUBLISHED VERSION, IF AVAILABLE.*

[1]For example, differentiating indicatives and imperatives (Huttegger, 2007; Zollman, 2011); signalling in social dilemmas (Wagner, 2014); network formation (Pemantle and Skyrms, 2004; Barrett et al., 2017b); deception (Zollman et al., 2012; Martínez, 2015; Skyrms and Barrett, 2018); meta-linguistic notions of truth and probability (Barrett, 2016, 2017); syntactic structure (Barrett, 2006, 2007, 2009); compositionality (Franke, 2014, 2016; Steinert-Threlkeld, 2016, 2018; Barrett et al., 2018); epistemic representations (Barrett and LaCroix, 2019); etc.

then compose to form games. Once such simple games have arisen, they may themselves compose to form more complex games—ones that can, perhaps, deal with novel phenomena more efficiently than learning new dispositions from scratch. This may happen in a variety of plausible ways—via ritualisation, template transfer, analogical reasoning, or modular composition. Thus, self-assembly concerns how games and the dispositions that they implement might evolve together.

Here, I provide novel models to show how modular compositional processes may provide some benefit over learning complex dispositions from scratch. I use logical operations as a testbed for comparing the efficacy and efficiency of several different models. In one case, the agents may utilise pre-evolved dispositions, which then self-assemble, via modular composition, to exhibit more complex behaviour. In another case, simple dispositions may evolve even while the means of composing them into more complex dispositions evolves.[2]

Each of these models utilises only simple reinforcement learning, with no punishment.[3] Reinforcement learning is often considered the simplest dynamic that we can study, regarding the agents' computational resources. All other things being equal, this is the simplest dynamic that readily illustrates the particular phenomena of interest.[4] Further, since this is the dynamic that Barrett and Skyrms (2017) discuss, using the same dynamic allows for more simple comparison of their results (as a baseline) with the results of the present models.

In the *unary logic* game (Section 2), I show how a function that takes a single proposition as its input, and which outputs a truth value, can be modelled in a sender-receiver context, like a signalling game. Once this architecture is available

[2]See also LaCroix (2018b).

[3]This dynamic has a long psychological pedigree is a standard model for human and non-human animal learning. The role that the 'law of effect' plays in animal learning was discussed by Thorndike (1905, 1911, 1927); this notion was formalised by Herrnstein (1961, 1970) and empirically tested as a model for human learning (Roth and Erev, 1995, 1998). Argiento et al. (2009) analyse limiting results of signalling games under simple reinforcement. For alternative dynamics, see Barrett (2006); Barrett and Zollman (2009); Skyrms (2010); Huttegger et al. (2010); Alexander et al. (2012); Barrett et al. (2017a). Sutton and Barto (1998) provide a computational perspective on reinforcement learning.

[4]Though some bandit-type problems, which simple heuristics like Herrnstein reinforcement readily solve, cause problems for more 'sophisticated' learners. See Vermorel and Mohri (2005); Kuleshov and Precup (2014).

to the agents, I demonstrate how we can understand a binary logical operator—specifically, NAND—as the modular composition of two unary operations (Section 3.1). Thus, the key idea is that rather than having to evolve a binary disposition from scratch, the agents may learn to compose two pre-evolved unary dispositions appropriately. In one way, this is an extension of previous work on self-assembly, insofar as the complex signalling behaviour examined here is the same as that which is examined in Barrett and Skyrms (2017)—namely, the evolution of logical operations; however, the mechanism by which this behaviour evolves is wholly different.[5] I compare the efficacy and efficiency of several means for evolving logical operations by examining the *cumulative success rates* of the agents' evolving strategies under simulation.[6] Finally, understanding binary operations as the composition of independent (pre-evolved) unary operations, I show how this model can be extended in several ways to obtain a more general picture of how such dispositions may *co-evolve* (Sections 3.2, 3.3, 3.4, 3.5).

Demonstrating how complex phenomena like these might evolve provides an additional path to the possibility of evolving more-or-less rich notions of compositionality. Independent modelling accomplishments of this sort make successful composition that much more plausible as a natural phenomenon. With sparse cognitive resources, agents can exhibit a surprising and subtle degree of complexity. This provides another facet of the evolutionary story of how sufficiently rich, human-level cognitive or linguistic capacities may arise from simpler precursors. Such an explanation is indispensable for research in language origins, insofar as most researchers take complex (i.e., compositional, hierarchical) syntax to be a defining

[5]Yet another mechanism for self-assembly via modular composition is examined in Barrett (2019).

[6]Cumulative success of a run is calculated simply by dividing the number of successful plays in that run by the total number of plays. The cumulative success *rate* is the proportion of runs that achieved a cumulative success above some threshold. The cumulative success rate approaches 1 when the players have fixed upon a signalling system—i.e., a maximally-efficient communication convention. A different measure of efficacy is given by the expected payoff of the players' strategies at a particular play of the game—the *communicative success rate*. The communicative success rate is a 'snapshot' which is better suited than the cumulative success rate at providing a picture of the estimated size of pooling equilibria: a sub-optimal random walk could spend some time above the expected cumulative success rate for pooling equilibria, though this becomes less likely as the threshold for the cumulative success rate increases. Still, neither of these measures guarantees what the results will be in the limit.

characteristic of human linguistic capacities.[7] This has additional implications for the evolution of cognition, inasmuch as a broader picture of self-assembly would show how completely general algorithms might evolve. Furthermore, this sheds light on how a cognitive system might self-assemble to perform epistemic functions, such as representation, communication, and inference.[8]

Though the models presented here are couched in the language of *logical operations*, it should be clear from the outset that these operators provide a simple testbed for the broader sort of compositional phenomena under consideration. This should be understood as providing a *how-possibly* explanation of the type that is common in this field (Hempel, 1965; Resnik, 1991).[9] In the conclusion (Section 5), I discuss in more detail empirical examples that display the more general sort of compositional processes, of which the logic games considered here provide a specific example.

2. Simple Unary Logic Games

I will begin with an excessively simple model. Whereas Barrett and Skyrms (2017) use the binary operator NAND to compare the evolution of new dispositions, we will see that we can understand such a binary operation as the modular composition of two *unary* logical operators. Consider the following four (exhaustive) unary operations on a single input (proposition) in the context of a two-player sender-receiver game: identity (ID), negation (NEG), tautology (TAUT), and contradiction (CONT); see Table 1.

p	ID(p)	NEG(p)	TAUT(p)	CONT(p)
1	1	0	1	0
0	0	1	1	0

Table 1. Unique outputs for unary functions

[7]See discussion in Progovac (2019); LaCroix (2019).

[8]See discussion in Barrett (2019); Barrett and LaCroix (2019).

[9]For example, the evolutionary signalling game described in Skyrms (2010) shows how communication *might* evolve without natural salience, but not that or how it *actually* did so.

Each of these unary functions can be modelled as a sender-receiver game, with each game being differentiated by the composition of its respective payoff matrix. I will denote the game that models the ID function as the 'ID game', and similarly for the other unary functions. The payoff matrices for the games corresponding to each of these unary logical operations are given in Table 2.[10] The ID game is equivalent

	a_0	a_1
s_0	1	0
s_1	0	1

(A) ID Game

	a_0	a_1
s_0	0	1
s_1	1	0

(B) NEG Game

	a_0	a_1
s_0	0	1
s_1	0	1

(C) TAUT Game

	a_0	a_1
s_0	1	0
s_1	1	0

(D) CONT Game

TABLE 2. Four possibilities for payoffs matching states to acts: (a) is a coordination game, which corresponds to the unary identity function; (b) is an anti-coordination game, corresponding to the unary negation function; (c) and (d) are, what we might call, pooling games, corresponding to the tautology and contradiction functions, respectively.

to the standard 2×2 signalling game. On this model, there are two states (the truth values of p), two messages, and two actions (the truth values defined by the operation that the game models). Reinforcement is modelled as an urn-learning process. The sender has an urn for each state, and each urn initially contains a ball for each message. Similarly, the receiver has an urn for each message, and each of these urns initially contains a ball for each action.

On a given play of the game, nature picks a truth value for the proposition, randomly and without bias. The sender sees the state and chooses a message randomly from the appropriate urn for the current state. The receiver sees the message but

[10] I follow the convention that 's_0' ['a_0'] corresponds to input [output] 0, and 's_1' ['a_1'] corresponds to input [output] 1. Since the payoffs are also denoted by '0' and '1', this is meant to help keep the state, act, and payoff disambiguated. I will refer to the input [output] as either '0' and '1', or 's_0' and 's_1' ['a_0' and 'a_1'] depending upon the context and which notation allows for the most clarity.

not the state; she chooses an action randomly from the urn for the current message. If the action 'matches' the state (as defined by the payoff tables in Table 2), then the play is counted as a success. In this case, the players each replace the ball they chose on that round to the urn from which it was chosen and add another ball of the same type to that urn. If the play was not successful, then they return the ball to the urn from which it was chosen. Thus, over time, their propensities to act shift proportional to their past successes.

Given that the ID game just is a 2×2 signalling game, we know that the players will (eventually, but usually quickly) coordinate upon one or the other signalling system with probability 1, when nature is unbiased (Argiento et al., 2009). Indeed, the NEG game is functionally equivalent to the ID game—modulo a permutation of the payoffs, or a re-labelling of the states or acts. Thus, it follows immediately that the NEG game also gives rise to one or the other signalling system with probability 1 (subject to the same caveats). Further, the receiver in the TAUT- and CONT games has an action available to her which is strictly dominant. Therefore, each of these will converge to an optimal strategy under simple reinforcement learning, since this dynamic converges on a dominant strategy, if there is one (Beggs, 2005). Accordingly, success is guaranteed eventually in each of these four simple unary logic games under reinforcement learning.

However, analytic results do not capture what happens in the short term, or how quickly these effective strategies might arise. On simulation, the average cumulative success rates (1000 runs each) for the ID- and NEG games are both around 0.70 after 10^2 plays per run. In contrast, by 10^2 plays per run, the average cumulative success rates for the TAUT- and CONT games are already near 0.92. A comparison of the cumulative success rates for each of these games for several thresholds is given in Table 3. Figures 1 and 2 illustrate the difference in speed of convergence between these games graphically. The relative speed of the TAUT- and CONT games will be necessary for explaining the results discussed in Section 3 below. Therefore, it is

	ID game		NEG game		TAUT game		CONT game	
	10^2	10^6	10^2	10^6	10^2	10^6	10^2	10^6
μ	**0.692**	**0.999**	**0.700**	**0.998**	**0.916**	**1.00**	**0.917**	**1.00**
0.99	0.00	0.98	0.00	1.00	0.05	1.00	0.04	1.00
0.95	0.00	1.00	0.00	1.00	0.20	1.00	0.21	1.00
0.90	0.03	1.00	0.03	1.00	0.76	1.00	0.78	1.00
0.80	0.27	1.00	0.29	0.98	1.00	1.00	1.00	1.00

Table 3. Comparison of cumulative success rates for unary logic games with a variety of thresholds for success

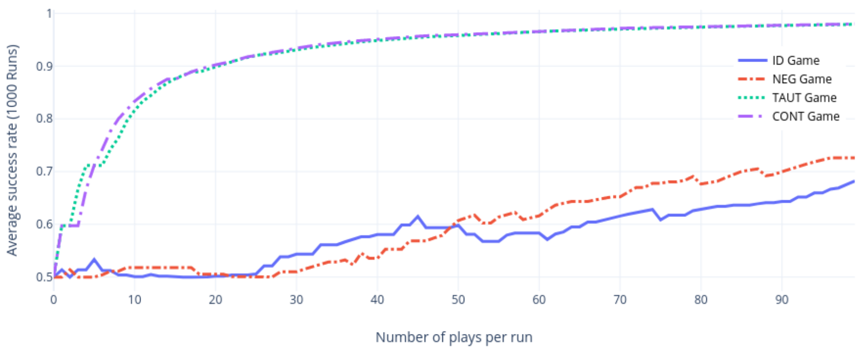


Figure 1. Comparison of average cumulative success rates for unary-input logic games in the short term (10^2 plays per run)

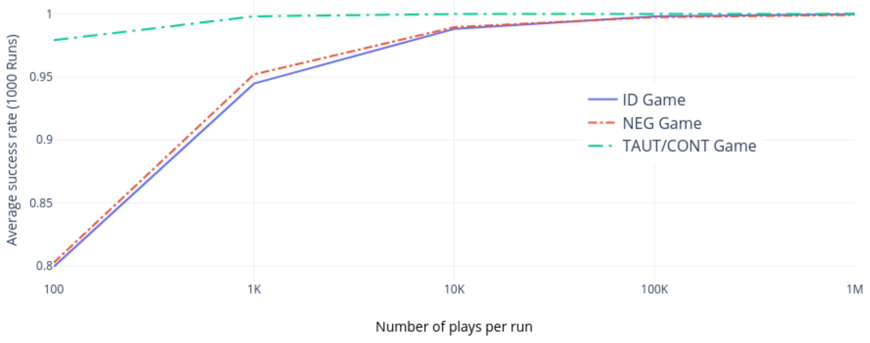


Figure 2. Comparison of average cumulative success rates for unary-input logic games in the long term (10^6 plays per run)

instructive to discuss why the TAUT- and CONT games should evolve faster than the ID- and NEG games.

In the ID- and NEG games, the players *must* co-evolve their strategies to reach a maximally-effective set of strategies—namely, the 'signalling systems' of these sender-receiver games.[11] However, there are more maximally-effective sets of pure strategies available to the players in the CONT- and TAUT games.[12] Taking account of mixed strategies implies a continuum of maximally-effective strategies for the TAUT- and CONT games: every mixed strategy for the sender in conjunction with the appropriate pure total-pooling strategy for the receiver will result in a maximal payoff in the TAUT- and CONT games, whereas no mixed strategy proffers maximal payoff in the ID- and NEG games.

Thus, it is sufficient, but not necessary, that the players co-evolve their strategies in the TAUT- and CONT games to achieve maximal payoff; the receiver may learn that the signal does not matter—she should always choose action 0 in the CONT game and action 1 in the TAUT game.

Unary functions are simple enough that they are essentially guaranteed to emerge in a sender-receiver context under simple reinforcement learning.[13] With these initial results outlined, I describe how a binary logical operator can be composed out of unary logical operators.

3. Composing Unary Functions for Binary Inputs

In this section, I demonstrate how actors in a signalling context might compose unary dispositions into more complex binary dispositions. I begin with the relevant

[11]Technically, the maximally-effective sets of strategies are only signalling systems in the ID- and NEG games since these are properly signalling games, which *require* coordination on the part of the agents. In the TAUT- and CONT games, the players need not coordinate to achieve maximal payoff—signals need not carry any information—so it makes little sense to talk of 'signalling systems' or 'coordination conventions' in these latter contexts.

[12]In the atomic n-game, there are n^{2n} strategy combinations, $n!$ of which are signalling systems. Thus, for the ID- and NEG games there are $2^4 = 16$ combinations of pure strategies, and $2! = 2$ of these are signalling systems. However, in the TAUT game and the CONT game, there are 4 maximally-effective sets of (pure) sender-receiver strategies.

[13]This guarantee requires that nature is not too biased for the ID- and NEG games (Hofbauer and Huttegger, 2008); no such assumption is required for the TAUT- and CONT games.

background, against which I will compare the novel models in this paper. This includes learning to evolve a binary disposition from scratch via *syntactic* signalling (Barrett, 2006, 2007, 2009) and appropriating a pre-evolved binary disposition for a novel context via *template transfer* (Barrett and Skyrms, 2017). In Section 3.1, I present a novel means for agents to utilise pre-evolved dispositions (the unary logic dispositions discussed in Section 2) to learn more complex novel dispositions (a binary-input NAND disposition). This model presupposes that the agents have already learned the unary logical operations and further learn to combine them into binary logical operations. However, the assumption that the unary logical operations are pre-evolved is relaxed in Section 3.2 using a hierarchical model similar to the one employed in Barrett et al. (2018). This too involves several simplifying assumptions, which are incrementally relaxed in Sections 3.3, 3.4, and 3.5.

Each of the models I discuss uses only simple reinforcement learning, where the propensities for actions are proportional to the accumulated rewards for prior actions. Nature is unbiased, and players' initial weights are always 1 so that the probability distribution over any player's actions is uniform to start. Finally, the rewards are always 1 for successful actions. There is no discounting, no error rate, no punishment, and no bounds in any of these models. In terms of our urn-learning metaphor, each urn always starts with one ball of each type, and a single ball of the appropriate type is added when agents act successfully.[14]

Since Barrett and Skyrms (2017) examine template transfer for NAND, I will use the same operator to analyse the evolution of binary logical functions via modular composition of the unary operations that were discussed in Section 2.[15] There are

[14]This is the most straightforward case, but there is good reason to think that the results presented will be robust to variation of these parameters. For example, Barrett et al. (2017a) discuss a low-rationality hybrid of simple reinforcement and the "win-stay/lose-randomise" learning dynamic and show that it is reliable, stable, and exceptionally fast for learning in signalling contexts. LaCroix (2018a) discusses a novel learning rule which helps avoid partial pooling, even in complex games. Similarly, adding punishment or forgetting can help agents evolve optimal signalling conventions (Barrett and Zollman, 2009).

[15]Barrett (2018) also discusses the evolution of NAND in the context of a *sender-predictor* game. However, his simulations use bounded reinforcement with punishment, which is radically different (and importantly more sophisticated) than the generic, straightforward dynamic I consider. Still, his general remarks regarding the effectiveness of appropriation of logical operations are relevant

several ways that we might model the evolution of a NAND game using reinforcement learning. The first, which I will refer to as the *atomic two-sender* NAND *game*, is shown in Figure 3. There are two senders, called *Sender A* and *Sender B*. There

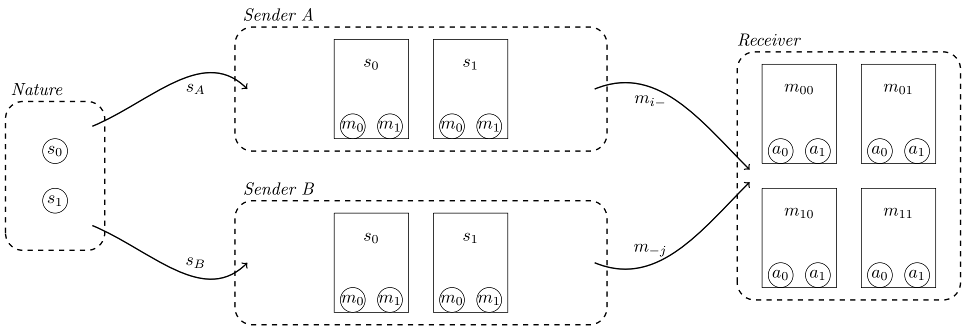


FIGURE 3. Reinforcement learning model for simple binary-input logic game with two senders. Nature chooses twice from one of two states; each sender chooses one of two messages from the urn matching the state received; the receiver chooses one of two actions from the urn matching the two-input signal received. Both the senders and receiver reinforce just in case the act corresponds (s_A NAND s_B)—i.e., the state shown to sender A and the state shown to sender B, respectively.

are two values for the state of the world, $s_0 = 0$ and $s_1 = 1$, and a full input state is an ordered pair. Thus, there are four input states, $\langle 0,0\rangle, \langle 0,1\rangle, \langle 1,0\rangle$ and $\langle 1,1\rangle$.

To be clear on notation, I will use s_i to indicate the state-value $i \in \{0,1\}$ and s_{ij} to denote the full binary state, $\langle i,j\rangle$. I will use s_A to refer to the state seen by sender A and s_B to refer to the state seen by sender B. Nature chooses each state independently, and each sender only sees one aspect of the full state.[16] m_{i-} denotes A's message, and m_{-j} denotes B's message. m_{ij} denotes the full 2-bit message that the receiver observes. Consistent with previous work on two-sender signalling games, I assume that the receiver knows which sender sends which message.[17] Further, the senders' messages are independent in the sense that neither sender knows which

here—specifically, the empirical case of appropriation in the context of rule-following in pinyon and scrub jays (Bond et al., 2003).

[16]This game can also be modelled so the senders each see the full state of nature and must coordinate on how they partition nature, to encode complete information about the state. Barrett (2018) highlights that a truth-functional rule is more likely to evolve when the senders each have access to different, independent states of nature than when they have access to the full state. However, such a constraint is not artificial, since the receiver must use information from both senders; hence, this allows for a generalisable truth-functional operation. In either case, I discuss role-free senders in Section 3.5.

[17]See Skyrms (2000, 2010); Barrett (2006, 2007, 2009, 2018, 2019); Barrett and Skyrms (2017).

message the other sent. Finally, there is no requirement that the two senders be interpreted as distinct agents—they might be understood as functional components of a single organism.

The atomic two-sender NAND game involves learning the appropriate outputs for NAND from scratch. Nature chooses a state-value randomly to send to A and separately chooses another state-value to send to B. The ordered combination of values $\langle s_A, s_B \rangle$ constitutes the full state, which can be interpreted as the binary input for the logical function in question. Each sender chooses a message from the urn that matches the state that she sees and sends that message to the receiver. The receiver sees which message each sender sent, and who sent the message, and chooses an output action, $a_0 (= 0)$ or $a_1 (= 1)$, from the urn corresponding to the full 2-bit message. In the NAND game, a play is successful just in case the receiver chooses the act corresponding to $(s_A \text{ NAND } s_B)$—namely, she should choose a_0 when both state-values are 1, and she should choose a_1 otherwise. The payoff table for the NAND game, with both the state and act labels, s_{ij} and a_k, and the actual input-output values, 0 or 1, is shown in Table 4.

		a_0	a_1
		0	**1**
s_{00}	$\langle 0, 0 \rangle$	0	1
s_{01}	$\langle 0, 1 \rangle$	0	1
s_{10}	$\langle 1, 0 \rangle$	0	1
s_{11}	$\langle 1, 1 \rangle$	1	0

TABLE 4. Payoff table (states and acts) for atomic NAND game

On simulation, after 10^6 plays per run, the cumulative success rate is 0.9053, on average (1000 runs), when the players must learn a NAND disposition from scratch. More often than not (0.54), the players achieve a cumulative success rate higher than 0.95, and about one-quarter of the time (0.26), they achieve a near-perfect cumulative success rate (≥ 0.99). In approximately one-quarter (0.26) of the runs, the agents appear to get caught in a partial-pooling equilibrium. Here, they fail to learn a maximally-efficient signalling convention for a reduced expected payoff of 0.75. In such cases, the receiver always chooses a_1.[18]

Once the agents have learned a NAND disposition, they may appropriate this disposition, via *template transfer*, for use in a novel context (Barrett and Skyrms, 2017). This process is an order of magnitude more efficient than learning the same disposition in the new context from scratch. A schematic for the template-transfer model is shown in Figure 4. We suppose that the agents have already coordinated

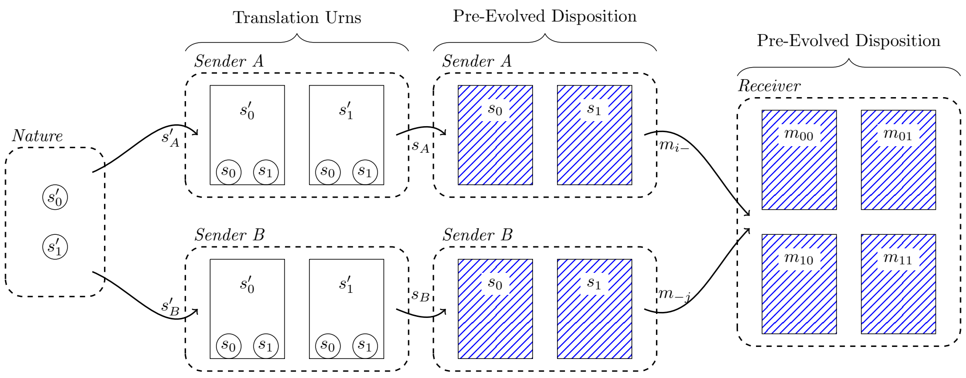
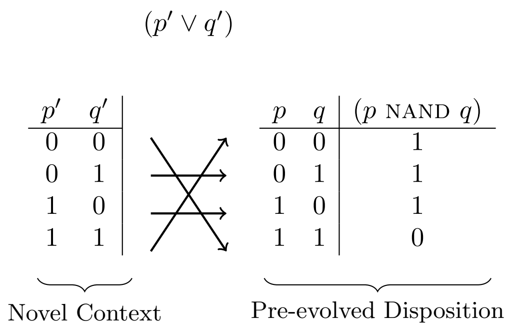


FIGURE 4. Transfer learning model for simple binary-input logic game with two senders. Nature chooses twice from one of two states; each sender chooses one of two messages from the urn matching the state received; the receiver chooses one of two actions from the urn matching the two-input signal received. Both the senders and receiver reinforce just in case the act corresponds (s_A NAND s_B).

upon a convention for outputting (s_A NAND s_B) on input $\langle s_A, s_B \rangle$. Thus, the urns for these pre-evolved dispositions are already populated and fixed. Now, given a new context with novel state-values, s'_0 and s'_1, the senders learn to appropriate the previously evolved disposition by *translating* the novel states into their analogues in the prior context. This model additionally shows how individuals who have already learned NAND can quickly learn a different logical operation, such as OR. See Figure 5.

Barrett and Skyrms (2017) report that on 1000 runs with 10^5 plays per run, 0.78 of the runs exhibit a cumulative success rate of better than 0.80, 0.61 of the runs

[18]On 10^7 plays per run, the agents still get caught in partial pooling at a rate of about 0.25.



p'	q'
0	0
0	1
1	0
1	1

p	q	$(p$ NAND $q)$
0	0	1
0	1	1
1	0	1
1	1	0

FIGURE 5. Example of translating a novel context (OR) into a pre-evolved disposition (NAND) via template transfer

better than 0.90, and 0.50 of the runs better than 0.95. This is roughly the same level of success that the atomic two-sender NAND game achieves on 10^6 plays per run.

The template-transfer model for NAND is meant to be suggestive. However, this set up has several short-comings—for example, though the players might evolve OR via template transfer on a pre-evolved NAND disposition, it is less clear how (or whether) they might be able to use this pre-evolved NAND disposition to evolve, e.g., AND, since this operation has a different number of '0' values. I will discuss these and other considerations in more detail in Section 4 below. But first, we will see how modular composition can be applied to the evolution of binary logical operations out of the unary logical operations that were presented in Section 2.

3.1. **Utilising Pre-Evolved Dispositions.** In this section, I present one of the key insights of this paper: the payoff table for the atomic NAND game (Table 4) can be understood as *the composition of two unary logic games*—in this particular case, TAUT and NEG; see Table 5. This insight is crucially important for understanding the subsequent models presented in this paper, as each of these builds off of and relaxes certain assumptions of the basic model shown here. The sense in which I mean that the binary logical operation NAND can be understood as the composition of the unary logic operations TAUT and NEG is as simple as this: the truth table for NAND (Table 4) *just is* the truth table for unary TAUT stacked on top of the

		a_1	a_2	
		$\mathbf{0}$	$\mathbf{1}$	
s_{00}	$\langle \mathbf{0}, 0\rangle$	0	1	TAUT Game
s_{01}	$\langle \mathbf{0}, 1\rangle$	0	1	
s_{10}	$\langle \mathbf{1}, 0\rangle$	0	1	NEG Game
s_{11}	$\langle \mathbf{1}, 1\rangle$	1	0	

Table 5. Payoff Table for NAND Game as the Composition of Two Unary Games

truth table for unary NEG (Table 5). This compositional idea is at the core of every model that follows in this section.

This may seem somewhat trivial; however, the order of the inputs plays a particular role in this game: the second input corresponds to the input of the unary sub-game, whereas the first input differentiates the two sub-games that compose the NAND game. Put another way, the first input tells us whether we are in the top half or bottom half of the (binary-NAND) truth table, and the second input codes for which output is appropriate *given* the context that is differentiated by the first input. Therefore, if sender A codes for the first input, and sender B codes for the second input, then the receiver might learn to *interpret* the first signal as specifying which unary game should take the state that is encoded by the second signal as input, and then output the appropriate value for that game and input.

If we assume that players have *pre-evolved* the unary sub-game (so their dispositions are already fixed), then the binary-input NAND game with pre-evolved unary sub-games can be modelled as in Figure 6. I will refer to this model as the *pre-evolved composition* model. There are two senders, as with the atomic NAND game (Figure 3). A's message codes for the first input from Nature—i.e., the unary logic sub-game that ought to be played. The receiver must learn the meaning of this message. B's message codes for the second input from Nature—i.e., the *input* of the unary sub-game. We assume that B's dispositions are fixed so that she always sends m_i on input s_i. Thus, the receiver must choose a unary sub-game to make use of B's message; however, once the receiver chooses a unary sub-game, she already

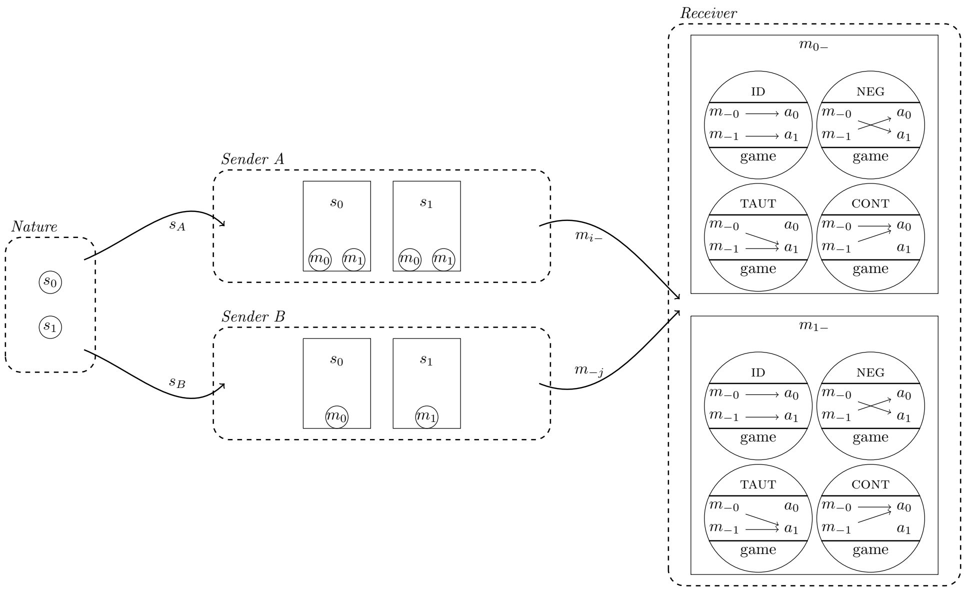


FIGURE 6. Reinforcement learning model for binary-input logic game assuming a pre-evolved unary-input logic game. Nature chooses one of two states for the sender; the sender chooses one of two messages to send to the receiver; the receiver acts according to her pre-evolved disposition for the message received. The solid rectangles (the urns) are determined by the input seen by a particular player. Once an urn is determined, the circles (the balls) show what choices are available to the agent.

knows how to proceed to choose an output based on B's message, since the four unary sub-games are pre-evolved.[19]

On a given play of the game, nature chooses a state-value randomly to show to A and separately chooses a state-value to show to B. A chooses a message from the urn matching the observed state to send to the receiver. B's choices are fixed. The receiver sees each of the messages and chooses, from the urn matching A's message, a unary sub-game to play. Once the sub-game is chosen, everything else is determined: the receiver performs the action corresponding to the message received from B, given the sub-game which she has chosen to play. There are ways in which the players can miscoordinate for a partial payoff in this game. The full payoff

[19]This model assumes the senders' roles are fixed—namely, A codes for the *game* and B codes for the *input*. However, this assumption is not necessary. A role-free game with pre-evolved unary sub-game dispositions has almost identical results as the fixed-role game presented here. Hence, I have opted to show the simpler of these models; however, I discuss role-free senders in detail in the co-evolutionary game presented in Section 3.5.

table is shown in Table 6.[20] As with the atomic binary-input NAND game, when

		a_0	a_1	a_2	a_3
		ID	NEG	TAUT	CONT
s_{0-}	$\langle \mathbf{0}, - \rangle$	0.5	0.5	1	0
s_{1-}	$\langle \mathbf{1}, - \rangle$	0	1	0.5	0.5

TABLE 6. Payoff Table for NAND Game as the Composition of Two Unary Games

the players act randomly, the chance payoff is 0.5; if they fail to evolve a signalling system, efficient pooling strategies have a success rate of 0.75.

On simulation, after 10^6 plays per run, the cumulative success rate for the pre-evolved composition game is 0.9440, on average (1000 runs). The players often (0.66) achieve a cumulative success rate higher than 0.95, and about one-third of the time (0.31) achieve a near-perfect cumulative success rate (≥ 0.99). Less than 10% of the time (0.08), they fail to evolve a maximally-efficient signalling convention and get caught in pooling equilibria—in these rare cases, the receiver fails to learn the differentiating feature of the first sender's signal and always chooses the TAUT game as the unary sub-game.

For ease of comparison, the results for 10^5 and 10^6 plays per run, for each of the atomic, template-transfer, and pre-evolved NAND games, are displayed in Table 7.[21] As with template transfer, composition that takes advantage of pre-evolved unary dispositions appears to allow the agents to learn the binary disposition an order of magnitude faster than learning it from scratch. When the agents learn to compose pre-evolved unary dispositions, they do comparably well to template transfer on a pre-evolved binary NAND disposition. However, although template transfer and

[20]Compare this with Table 5. Recalling that the unary logic sub-game dispositions are pre-evolved, and so fixed, consider the following situation. Suppose $s_A = 1$ and the receiver chooses to play the NEG game. Then regardless of what s_B is, the receiver will choose the correct action, affording each player a payoff of 1. Suppose, however, that the receiver chooses the ID game. Then regardless of what S_B is, the receiver will choose the wrong action, affording each player a payoff of 0. Finally, suppose that the receiver chooses to play the TAUT game. Then, if s_B was 0, she outputs 1, which was the correct action (since she should have chosen the NEG game, which outputs 1 on input 0); however, if s_B was 1, then she again outputs 1, which is the incorrect action (since she should have chosen the NEG game, which outputs 0 on input 1). Therefore, the players will average a payoff of 0.5, since the states are uniformly distributed. The same is true if the receiver chooses the CONT game when she should have chosen the NEG game.

[21]The results for the template-transfer game are as reported in Barrett and Skyrms (2017); they do not report results for 10^6 plays.

	Atomic NAND		Template Transfer		Pre-Evolved Composition	
	10^5	10^6	10^5	10^6	10^5	10^6
0.95	0.35	0.54	0.50	—	0.47	0.66
0.90	0.50	0.63	0.61	—	0.64	0.79
0.80	0.66	0.74	0.78	—	0.84	0.92

TABLE 7. Comparison of evolutionary efficacy for learning NAND (a) as a novel disposition, from scratch; (b) via template transfer on a pre-evolved NAND disposition; and (c) via simple reinforcement on pre-evolved *unary* dispositions

pre-evolved unary composition both take advantage of prior dispositions, the model presented here is more efficient at evolving a NAND disposition than with template transfer, in the following sense.

The atomic NAND game sometimes (0.25 on 10^7 plays per run) fails to evolve NAND, but pools strategies for an expected payoff of 0.75. Barrett and Skyrms (2017) report that their template-transfer game fails to evolve NAND at about the same rate (0.23 on 10^7 plays per run). Therefore, when the players do learn the disposition, they learn more quickly with template transfer than atomically. However, they fail to learn as often with template transfer as they do atomically. In contrast, utilising pre-evolved unary sub-games to learn a composed NAND disposition fails less often than either template transfer or atomic NAND—only 0.08 of the runs fail to evolve NAND on 10^6 plays per run, and only 0.16 of the runs fail to evolve NAND on 10^5 plays per run. Therefore, the agents in a pre-evolved NAND game learn as quickly as in template transfer, but they also learn significantly more often. Composing pre-evolved dispositions is both as efficient and more effective.[22]

Since the discussion of template transfer in Barrett and Skyrms (2017) presupposes that the underlying disposition is already evolved, the assumptions in my pre-evolved composition model seem relevantly justified—at least for comparing

[22]There is a subtle point to be made clear here: recall that the cumulative success rate gives a measure of all of the plays over the course of all of the runs. If agents are slow to learn, then early failures may not be truly washed out. However, the *communicative* success rate is not history-dependent in the same way. In this case, the pre-evolved NAND game has a failure rate between 0.02 and 0.05 by 10^6 plays per run, as compared with a failure rate between 0.15 and 0.25 in the atomic case. This implies that some of the runs that count as failures in the pre-evolved game are just slow to learn. However, as was noted above, increasing the number of runs in the atomic case does *not* change the failure rate—the runs that are counted as failures appear genuinely to be caught in partial-pooling equilibria.

these results.[23] However, by cashing out binary operators in terms of the composition of (pre-evolved) unary operators, this model can do slightly more.

I mentioned previously that Barrett and Skyrms (2017); Barrett (2019) suggest that an OR disposition can easily be transferred from a NAND disposition by dint of the parity of their truth tables—both OR and NAND have three inputs yielding '1' and one input yielding '0'. However, it is not so obvious how this translation can be generalised: NAND cannot be transferred to learn an XOR disposition effectively since its truth values do not exhibit this sort of parity with NAND—XOR has *two* inputs yielding '1' and two inputs yielding '0'. Thus, understanding binary operations in terms of the composition of unary operations has at least this theoretical virtue over and above simple template transfer.

We have seen how NAND can be composed of unary TAUT and NEG operations; similarly, OR can be composed by unary ID and TAUT operations, XOR can be composed by unary ID and NEG, etc. Since there are 4 distinct unary operators, and a binary operator consists of some particular permutation of these (with replacement, so we can account for, e.g., a 2-input tautology or contradiction), we have $2^4 = 16$ unique permutations of unary operations, which correspond precisely to the 16 unique binary logical operators.

Furthermore, we know from the results of Section 2 that all of the unary operations are a 'sure thing' when nature is not too biased, and TAUT and CONT are a sure thing regardless of whether nature is biased. Therefore, it is not unreasonable to assume that these simple dispositions come pre-evolved. In such a case, the agents can learn a complex disposition by merely learning how to code for which unary disposition is appropriate in which context. Though the assumption that the unary sub-game comes pre-evolved is perhaps theoretically justified, we would like a more general picture that does not necessarily presuppose such favourable circumstances are already in place. In Section 3.2, this simple model is extended to account for dispositions that *co-evolve*.

[23] Note that the roles of the senders are also fixed in the template-transfer model.

3.2. **Co-Evolving Logical Dispositions.** The *co-evolutionary* logic game is a variant of the special composition game presented in Barrett et al. (2018); it is a cooperative game with two base senders (whom we will call A and B) and one base receiver. The agents in this game must evolve a particular sort of signalling system to be uniformly successful. Additionally, there are two 'hierarchical' agents—an 'executive' sender and an 'executive' receiver—who can learn to influence the behaviour of the base agents. See Figure 7.[24]

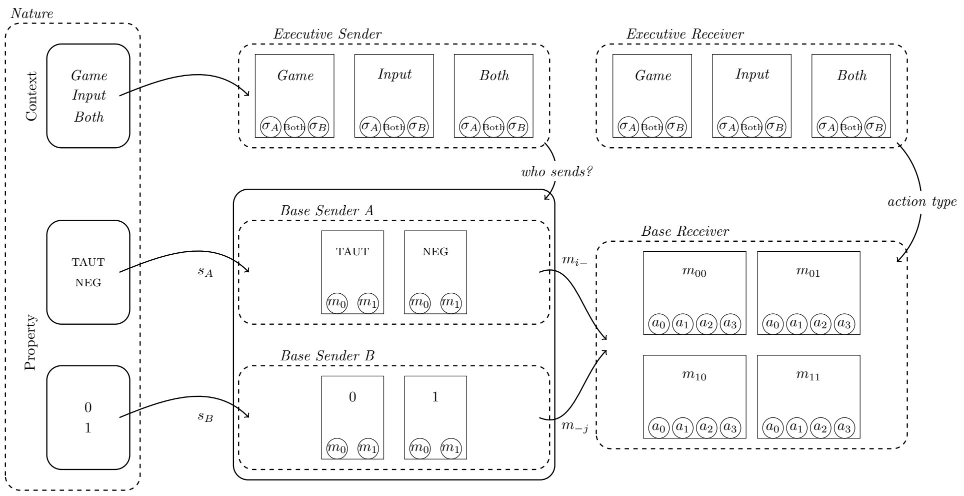


FIGURE 7. Reinforcement learning model for hierarchical, co-evolutionary binary-input logic game.

A complete specification of the state is given by two *properties* (or *features*) and a *context*. The properties are *game* (i.e., the unary sub-game) and *input*. Each of the properties has two *values*. Considering NAND as a concrete example, the value of the *game* property can be TAUT or NEG, and the value of the *input* property can be 0 or 1. Therefore, the state on a particular play of the game will be either ⟨TAUT, 0⟩, ⟨TAUT, 1⟩, ⟨NEG, 0⟩, or ⟨NEG, 1⟩. The context indicates which *aspect* of the state—i.e., which of the two properties—needs to be known by the receiver for her to perform a successful action on that particular play of the game. Thus, the values for the context are *game*, *input*, or *both*.

[24]Note that I shift the notation for representing the first index from nature as "0" to explicitly representing it as the game. It should be clear that "s_{00}" "$\langle 0, 0\rangle$", and "⟨TAUT, 0⟩" are different representations for the exact same thing. See Table 5.

The game is played by two base senders, an executive sender, one base receiver, and an executive receiver. Each base sender is assigned a particular property and only has access to that aspect of nature. (This condition is dropped in Section 3.5, where I discuss role-free senders.) As such, one base sender sees the *game*, and the other base sender sees the *input*. Initially, the executive sender randomly determines whether the game sender (A), the input sender (B), or both will send a signal. Over time, the executive sender may learn what type of signal the current context demands—i.e., a unary logical operation or a binary logical operation.

The base receiver sees the signals sent by the base senders and knows which sender sent each signal. The executive receiver also sees who sends the signals, and she determines whether the base receiver will interpret the signal as a 1-bit message—*game* or *input*—or a 2-bit message—*both*. The receiver performs an action based upon her interpretation. The actions for the receiver are *represented* by ⟨*game, input*⟩-pairs, matching the states of nature. We assume, for now, that the receiver 'just knows' what the output is for this representation. (This condition is dropped in Section 3.3.)

This can be interpreted as follows: the receiver understands the appropriate output—0 or 1—for a given state; however, she does not have access to the current state. Thus, the players must co-evolve a communication system whereby the receiver gains knowledge about the current state. Once the receiver knows the current state, she *automatically* knows the correct output. Thus, the actions are represented by ⟨TAUT, 0⟩, ⟨TAUT, 1⟩, ⟨NEG, 0⟩, and ⟨NEG, 1⟩. Note that, on this model, the receiver does not have access to representations for CONT or ID. (This condition is dropped in Section 3.4.)

The agents are successful on a particular play of the game just in case (1) the base receiver performs the correct action given the current context *and* (2) the base senders only sent the signals required for success given the context. Thus, to be successful, the receiver's action must be appropriate for the state, and the senders must be as efficient as possible—i.e., they must not send any irrelevant signals.

The 'efficiency' condition requires the base senders to coordinate, additionally, on something like a pragmatic maxim of *relation*—namely, the signal sent must be *relevant* to the current type of context (Grice, 1975). This condition is what drives the *co-evolution* of the unary dispositions and their composition.[25] See Figure 8 for an example.

Nature	Base Senders	Base Receiver	
Context	**Who Sends?**	**State**	**Action**
Both	*Sender A* *Sender B*	$\langle \mathbf{NEG}, \mathbf{0} \rangle \longrightarrow$	$\langle \mathbf{NEG}, \mathbf{0} \rangle$
Game	*Sender A Only*	$\langle \mathbf{NEG}, 0 \rangle \longrightarrow$	$\begin{cases} \langle \mathbf{NEG}, 0 \rangle \\ \langle \mathbf{NEG}, 1 \rangle \end{cases}$
Input	*Sender B Only*	$\langle \textsc{neg}, \mathbf{0} \rangle \longrightarrow$	$\begin{cases} \langle \textsc{neg}, \mathbf{0} \rangle \\ \langle \textsc{taut}, \mathbf{0} \rangle \end{cases}$

FIGURE 8. Example of success conditions for the three types of context. Each row specifies what is required for success in that row's context. Not pictured here are the roles of the executive players; however, they play a part in the success conditions, insofar as the executive sender *determines* which sender sends a signal, and the executive receiver *determines* how the sender interprets the signal.

Again, agents learn via simple reinforcement. On each play, nature determines a state by choosing a value for each of the two properties and the context randomly and with uniform probabilities. The executive sender is equipped with an urn for each of the three context-values—*game*, *input*, and *both*. Each urn begins with one ball of each type: *Sender A*, *Sender B*, and *Both*. The executive sender observes the context and randomly draws a ball from the corresponding urn. The drawn ball determines who will send a signal.

[25]For example, when the value of the context is *game*, the input is irrelevant, so the sender need only pick an action having to do with the game-value of the state on that round; and, *mutatis mutandis* when the value of the context is *input*. It may seem counterintuitive to 'successfully' play a game without knowing the appropriate input, or to 'successfully' choose an output without knowing the game. However, recall that 'game' and 'input' are peculiar to the structure of the *logic game* being discussed. The game itself, as was mentioned in the introduction, is only meant to serve as a relatively clear testbed for the *types* of compositional processes of interest here. Nonetheless, I discuss real-world interpretations in Section 5. An alternative way of enforcing efficiency is to posit a cost for signals. However, to maintain consistency with the parameters of the other models presented here—i.e., no punishment—a more stringent condition is placed on what counts as a success.

A is equipped with an urn labelled TAUT, and an urn labelled NEG; each initially contains a ball labelled m_0, and a ball labelled m_1. If the executive sender draws a ball requiring A to send a signal, then A randomly draws a ball from the urn corresponding to the property she observes, and she sends the corresponding signal. Similarly, B is equipped with an urn labelled 0, and an urn labelled 1—each initially containing a ball labelled m_0, and a ball labelled m_1. If required by the executive sender, she draws a ball from the urn corresponding to the property that she sees and sends that signal to the receiver.

The receiver has four urns, one for every ordered pair of signals she might receive from A and B, respectively: m_{00}, m_{01}, m_{10}, and m_{11}. Each urn begins with one ball for each of the game-input pairs: $\langle$TAUT, 0$\rangle$, $\langle$TAUT, 1$\rangle$, $\langle$NEG, 0$\rangle$, or $\langle$NEG, 1$\rangle$. If both senders send a signal, then the receiver draws a ball randomly from the corresponding urn. If only one sender sends a signal, then the receiver randomly chooses, with unbiased probabilities, one of the two urns corresponding to the sender's signal then draws a ball randomly from that urn.

The executive receiver determines how the receiver will interpret the type of signal she received. This interpretation, in conjunction with the ball the receiver drew determines how the receiver will act. The executive receiver is equipped with a *game-sender* urn, an *input-sender* urn, and a *both* urn. Each of these initially contains a *game* ball, an *input* ball, and a *both* ball. The ball drawn by the executive receiver determines what type of act the receiver takes as salient given the signal(s) that she has received.

If a play of the game is successful, as per the conditions described above, then each agent who was involved in that particular play returns the ball she drew to the urn from which she drew it and adds another ball of the same type to that urn. Otherwise, each agent simply returns the ball she drew to the urn from which she drew it.

On simulation, the agents nearly always evolve a successful and optimally efficient communication system. After 10^6 plays per run, the cumulative success rate is

0.9716, on average (1000 runs). The players usually (0.88) achieve a cumulative success rate higher than 0.95, and most of the time (0.63) they achieve a near-perfect cumulative success rate ($\geq$ 0.99). Rarely (0.04), they fail to evolve a maximally-efficient signalling convention and get caught in pooling equilibria.[26]

Comparing the cumulative success rates, we see that the co-evolutionary NAND game evolves an order of magnitude faster than the pre-evolved NAND game, which in turn evolved an order of magnitude faster than the atomic NAND game on the same learning dynamic. This is despite the fact that the chance payoff for the co-evolved NAND game is less than the chance payoff for the atomic or pre-evolved games. (Since the receiver has twice as many options, the chance payoff is 0.25, rather than 0.50.) Thus, this game starts with a significant handicap, but still outperforms learning NAND from scratch by at least an order of magnitude. See Table 8. Part of the reason for this is the full set of conditions for what success

	Atomic NAND		Template Transfer		Pre-Evolved Composition		Co-Evolved Composition	
	10^5	10^6	10^5	10^6	10^5	10^6	10^5	10^6
0.95	0.35	0.54	0.50	—	0.47	0.66	0.64	0.88
0.90	0.50	0.63	0.61	—	0.64	0.79	0.79	0.92
0.80	0.66	0.74	0.78	—	0.84	0.92	0.89	0.96

TABLE 8. Comparison of evolutionary efficacy for learning NAND (a) as a novel disposition, from scratch; (b) via template transfer on a pre-evolved NAND disposition; (c) via simple reinforcement on pre-evolved *unary* dispositions; and (d) via co-evolved unary dispositions

consists in. The payoffs structure the dispositions of the agents so that they cannot be successful in any way unless they are successful in every way. This is discussed in more detail in Section 4.

There are a significant number of simplifying assumptions made in this model, which one may worry are allowing for the high rates of success that we see on simulation. First, I assumed that the receiver 'just knows' how to interpret an

[26]Again, if we examine the 'snapshot' measure of the communicative success rate, the results are slightly better. The average expected payoff after 10^6 plays per run is 0.9747. More than three-quarters of the time, the agents achieve a near-perfect communication convention for a payoff greater than 0.99. Still, 0.04 runs fail to exceed a payoff that could be got by a partial-pooling convention.

action, such as $\langle$TAUT, 1$\rangle$—i.e., by outputting 1. Second, I assumed that the only possible states for the game were TAUT and NEG. This is a simplification, given that there are two additional unary operations which the agents may well learn in a general framework, but which happen not to be useful for producing the appropriate action in the NAND context. Finally, I assumed that each base sender is assigned a particular property—*game* or *input*—and only has access to that aspect of nature. Thus, the roles of the senders are fixed.

I relax the first assumption in Section 3.3, where the receiver must also learn which output, 0 or 1, is appropriate for which state. The second assumption is relaxed in Section 3.4, where I extend the composition game to account for the full action space of unary operations. Finally, in Section 3.5, I drop the assumption that the roles of the base senders are fixed.[27] These results are discussed in more detail in Section 4.

3.3. **Learning Appropriate Outputs.** In this section, I drop the condition that the receiver 'just knows' what to do with the 'action' she has chosen from her urn. This game is modelled precisely as the co-evolutionary logic game, except for the following modification. Instead of 4 balls with the state labels, each of the receiver's urns has 8 balls with the state-labels *plus* an output—0 or 1. Thus, each ball has three-part label corresponding to the *game* component of the state, the *input* component of the state, and the *output* component of the state—thus, balls on this interpretation are labelled $\langle$*game*, *input*, *output*$\rangle$.

Now, it is not pre-supposed that the receiver 'just knows' what action she ought to perform when she draws the ball $\langle$TAUT, 0$\rangle$, since she has balls labelled $\langle$TAUT, $0, 0\rangle$ and $\langle$TAUT, $0, 1\rangle$. Thus, she must *learn* which output is correct, given the complex state.

On this model, the players are successful in coordinating their actions just in case (1) the receiver performs the correct action for the given context, and (2) the

[27]Each of these assumptions is dropped independently of the others. This is meant only to be suggestive concerning the effects of these individual assumptions on the simulation results. It would be ideal, though due to space constraints impractical, to look at dropping combinations of assumptions to see whether there are interaction effects between these several parameters.

senders only sent the signals required for success given the context, *and* (3) the receiver chooses the correct *output* given the action selected. The rest of the game is as was described before.

On the co-evolutionary logic game with learned inputs, the agents still effectively *always* learn a successful and optimally efficient communication system. After 10^6 plays per run, the cumulative success rate is 0.9313, on average (1000 runs). More than half the time (0.57), the senders and receiver achieve a cumulative success rate higher than 0.95, though they rarely (0.03) achieve a near-perfect cumulative success rate (≥ 0.99). In very few cases (0.07), they fail to evolve a maximally-efficient signalling convention and get caught in pooling equilibria. These results are comparable to the basic co-evolutionary NAND game, where it is assumed that the receiver knows what to do given the state, even though the players start at a significant disadvantage—the chance payoff at the outset is half that of the basic co-evolved NAND game and one-quarter that of the pre-evolved and atomic NAND games.

3.4. **Taking Account of the Full State Space of Unary Games.** In this section, I drop the assumption that the only games available to the receiver are TAUT and NEG. Though the ID and CONT games are not appropriate for the NAND game, one might argue that the receiver must learn that these actions are inappropriate. This condition is dropped by extending the basic co-evolutionary NAND game (3.2) to a more general logic game. Now, there are 8 actions that the receiver might choose, corresponding to the eight combinations of TAUT, CONT, ID and NEG with the inputs 0 and 1.[28]

On the co-evolutionary logic game where the receiver needs to differentiate the appropriate action from the full state space of unary sub-games, the results are similar to the case where the agent needs to learn the correct output for a given state—it is slightly less efficient and slower to learn initially. Both of these facts

[28]Formally, this model is similar to the model of Section 3.3, where the receiver must additionally learn the appropriate output for a given state. However, the success conditions are different when the context is *game* only; thus, these models are not functionally equivalent.

make sense since the models are similar in complexity, but, in this case, there are fewer situations that constitute a success. After 10^6 plays per run, the cumulative success rate is 0.9458, on average (1000 runs). More than three-quarters of the time (0.78), the agents achieve a cumulative success rate higher than 0.95, and they often (0.43) achieve a near-perfect cumulative success rate (≥ 0.99). In some cases (0.10), they fail to evolve a maximally-efficient signalling convention and appear to get caught in pooling equilibria.

3.5. **Role-Free Composition.**

The final assumption that I examine is that the roles of the base senders are fixed. As was mentioned previously, assuming that the roles of the base senders are pre-assigned imposes a fair amount of structure on the hierarchical co-evolutionary NAND model. For one, this guarantees that the executive agents always learn to coordinate. On the other hand, a truth-functional rule is more likely to evolve when the senders each have access to different, independent states of nature than when they have access to the full state of nature.[29]

On the role-free co-evolutionary logic game, the base senders have no pre-assigned representational roles. Instead, they are both shown the full state of nature. In this case, the base senders must learn to coordinate their roles to partition nature fully, and the executive agents must learn what roles the base senders are playing. The executive agents thus co-evolve their dispositions even while the base agents are learning their representational roles.

Since there are no stipulated roles for the base senders, there is no stipulated game sender or input sender. On each play of the game, the senders are both shown one of the four states (but not the context-value). These, again, are $\langle$TAUT, 0$\rangle$, $\langle$TAUT, 1$\rangle$, $\langle$NEG, 0$\rangle$, and $\langle$NEG, 1$\rangle$. Each sender has an urn for each of these states, and each urn contains balls labelled m_0 and m_1. Each sender still has only two available messages, and so neither sender can convey full information about the state of the world. Thus, to be successful, they must coordinate so that they partition nature fully—that is, their signals ought to give complementary information about

[29]See discussion in Barrett (2018).

the full state of nature. So, one sender ought to learn to code for the *game*, and the other sender ought to learn to code for the *input*.

Since there are no pre-assigned roles in the role-free composition game, the conditions for success are also slightly different. A play now counts as a success just in case (1) the receiver performs the correct action given the current context (as before), and (2) Both senders send a signal *if and only if* the context given by nature requires both *game* and *input*.

After 10^6 plays per run, the cumulative success rate is 0.9450, on average (1000 runs). About two-thirds of the time (0.67), the agents achieve a cumulative success rate greater than 0.95, and they sometimes (0.21) achieve a near-perfect cumulative success rate (≥ 0.99). Rarely (0.05), they fail to evolve a maximally-efficient signalling convention and appear to get caught in pooling equilibria.

4. Discussion

4.1. **Efficacy and Efficiency of Learning Complex Dispositions.** Several subtleties should be noted about the results discussed in the previous sections. Particularly, a distinction can be made between how *effective* the agents are at learning a signalling disposition and how *efficient* they are at learning that disposition. Efficacy is highlighted by the long-run results—particularly by the avoidance of partial-pooling equilibria. In this case, as we have already seen, composing simple dispositions is *always* more effective than learning a complex disposition from scratch to achieve a maximally-efficient signalling strategy in the NAND game. See Table 9.

This comparison is made clear in terms of the communicative success rate (average expected payoff) over the long-term (10^6 plays per run) course of these runs in Figure 9. The co-evolved disposition (3.2) is the most effective, followed by the model that utilises a pre-evolved unary sub-game disposition (3.1). The extended models of Sections 3.3, 3.4, and 3.5 perform slightly worse in the long run than the base models; however, they all outperform the atomic case.

	Atomic NAND		Template Transfer		Pre-Evolved Composition		Co-Evolved Composition	
	10^5	10^6	10^5	10^6	10^5	10^6	10^5	10^6
0.95	0.35	0.54	0.50	—	0.47	0.66	0.64	0.88
0.90	0.50	0.63	0.61	—	0.64	0.79	0.79	0.92
0.80	0.66	0.74	0.78	—	0.84	0.92	0.89	0.96

	Learned Outputs		Full State-Space		Role-Free Composition	
	10^5	10^6	10^5	10^6	10^5	10^6
0.95	0.43	0.78	0.10	0.57	0.31	0.67
0.90	0.65	0.84	0.41	0.82	0.60	0.85
0.80	0.79	0.90	0.77	0.93	0.87	0.95

TABLE 9. Comparison of evolutionary efficacy for learning NAND (a) as a novel disposition, from scratch; (b) via template transfer on a pre-evolved NAND disposition; (c) via simple reinforcement on pre-evolved *unary* dispositions; and (d) via co-evolved unary dispositions. These base models are compared with (e) learning the appropriate outputs, (f) learning from the full state-space, and (g) learning with role-free agents

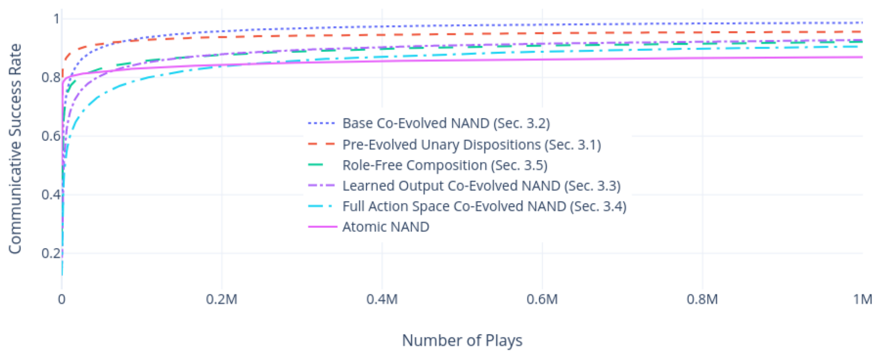


FIGURE 9. Comparison of average communicative success rates over 10^6 plays per run (efficacy considerations)

The reason for this has to do with the efficacy of the learning rule in the following sense. When the agents do learn a maximally-efficient signalling disposition in the atomic NAND game, they generally do as well as in any other game. However, they fail to evolve such a maximally-efficient communication system more often than in the other cases, which brings the average down. A signalling system obtains a maximal payoff of 1 in the limit; however, if 0.25 of the runs get caught in partial-pooling equilibria, then the average payoff will rise no higher than 0.937 in the limit. Thus, these averages tell us something about how effective each model is at avoiding partial-pooling equilibria.

Upon reflection, this ordering makes some sense. The co-evolutionary NAND game takes advantage of a more complex, hierarchical structure than the flat pre-evolved and atomic NAND game models. The base co-evolutionary model contains fewer situations that constitute success, which in turn enhances the ability of the agents to avoid pooling. Furthermore, we can analyse the game in terms of its structural components: there are situations in which *only* the sub-game dispositions are relevant. Since we saw in Section 2 that TAUT and CONT are learned more quickly than ID and NEG, this comes to bear in a significant way on the co-evolution of complex dispositions. That is, the complex disposition can be broken up into structural components which themselves vary in how difficult they are to learn. The extensions of the co-evolutionary model do slightly worse because the receiver has more choice points available to her; however, these complex games still perform better than the atomic case because they still have components which, in some rounds, are decoupled from the more complex game itself. In the atomic case, no such decomposition is available to the agents.

These long-run results say nothing about how quickly the agents learn such dispositions. To get a sense of how efficient learning is, we can compare the short-run results of each of these models. Although the atomic NAND game is the least effective of these models, it is also one of the most efficient. However, because it is not as effective as the other models, it is eventually surpassed in every case. This comparison is made clear in terms of the average expected payoff over the short term (10^4 plays) in Figure 10. Note further that the co-evolutionary games, due to their complexity, start at a significant disadvantage from the atomic game—the chance payoff is significantly lower in each of these cases. Thus, learning is slow and steady in each of the co-evolved instances, but it is also extremely effective.

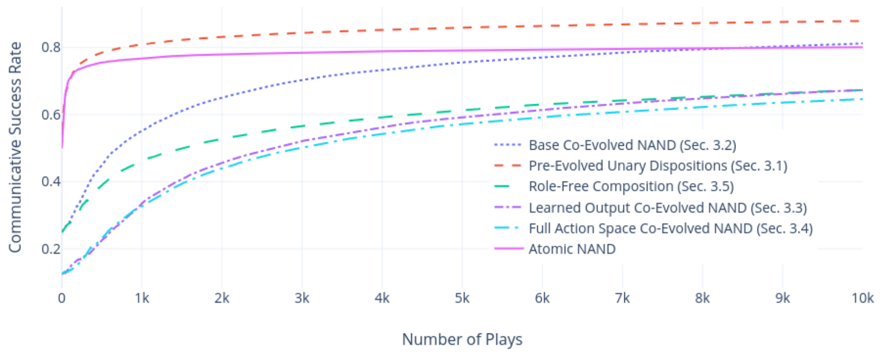


FIGURE 10. Comparison of average communicative success rates over 10^4 plays per run (efficiency considerations)

A final note about the efficacy of composing unary functions compared to template transfer. Barrett and Skyrms (2017) suppose that a NAND disposition has already been evolved. So, evolving a NAND disposition, and learning to apply that disposition to a novel context, are modelled as *independent* processes. However, having learned a NAND disposition is a necessary condition for there to be a template to transfer to a novel context. As such, these should be understood as *serial* processes.

The simulation results for learning an atomic NAND game from scratch suggest that approximately 0.25 of the runs result in pooling equilibria. Suppose we have a population of pairs of senders and receivers. Then, around 0.75 of these pairs will learn a NAND disposition, on average. Those who failed to learn NAND would not be able to transfer these dispositions successfully. We also know from the simulation results that learning a NAND disposition is not sufficient for successful transfer in a novel context: approximately 0.25 of the template transfer runs also result in pooling equilibria, failing to learn NAND in the new context successfully. Thus, when considering the serial evolution of the complex disposition, slightly more than half of the population might be successful.

In contrast, we know that the unary NEG disposition is a sure thing. Therefore, every pair of individuals will learn this disposition. Further, TAUT is also a sure thing, so that every pair of individuals will also learn this disposition. Finally, based on the simulation results of Section 3.1, success in coordination is high so that approximately 0.92 of those individuals who learned the pre-evolved unary dispositions will also learn to combine them appropriately into a NAND disposition. Thus, almost everyone in the population will learn this disposition when we consider the evolution of these dispositions as a combined serial process.

4.2. **Other Binary Operations.** There is a subtle distinction that comes to light when we understand binary operators in terms of unary operators. We saw in Section 2 that TAUT and CONT evolve more quickly than ID and NEG. So, the composition of, e.g., TAUT and CONT together should evolve faster than ID and NEG, since each of these dispositions is, independently, easier to learn in the former case than in the latter. I discussed the NAND game, since this is the logical operation that Barrett and Skyrms (2017); Barrett (2019) discuss. However, the NAND game—which is composed of TAUT and NEG—should then evolve faster than, e.g., an IFF game, which is composed of ID and NEG. We should also expect a binary operation that is composed of two completely pooling unary operations—for example, a binary TAUT operator, to evolve more readily.

Simulation results suggest that this intuition is correct. A comparison of the cumulative success rates for IFF, NAND, and TAUT—being completely representative of the combinations of completely-separating, half-pooling, and totally-pooling subgames—are shown in Table 10. This shows a clear ordering concerning how easy it is to (quickly) learn a binary disposition as a function of how easy it is to learn the constituent unary dispositions which underlie it.

The binary tautology (which takes two inputs and always outputs 1) is easiest to evolve because, in essence, the signal does not matter: the receiver needs only to react to any signal with the same disposition—a_1. This is quickly learned even in the atomic case. In the case of atomic IFF—a binary operation which has no

	Atomic IFF		Atomic NAND		Atomic TAUT	
	10^4	10^6	10^4	10^6	10^4	10^6
Average	**0.7502**	**0.8997**	**0.8326**	**0.9053**	**0.9968**	**0.9999**
0.99	0.02	0.56	0.00	0.26	1.00	1.00
0.95	0.25	0.72	0.14	0.54	1.00	1.00
0.90	0.35	0.76	0.29	0.63	1.00	1.00
0.80	0.50	0.80	0.53	0.74	1.00	1.00

Table 10. Comparison of three different ways of combining unary operations (cumulative success rate) over the short term (10^4 plays per run) and long term (10^6 plays per run)

pooling and thus *requires* co-evolution of strategies—the results are as expected. Specifically, NAND is more likely to get caught in partial-pooling equilibria than IFF and IFF is more efficient than NAND overall; however, because of the difficulty in co-evolving strategies, IFF is more difficult for agents to learn—the variance in the payoffs is significantly larger for IFF (1.74×10^{-1}) than it is for NAND (9.95×10^{-2}). Thus, Barrett and Skyrms (2017) do not look at the simplest case when they discuss template transfer, but they also do not look at the most challenging case.

4.3. **To Infinity, and Beyond.**

How well do these results generalise? There are two different ways that agents can learn a ternary-input logical operation: they might learn to compose *two* binary-input logical operations appropriately, or they might learn to compose *four* unary input logical operations. These two possibilities are illustrated in Table 11 for evolving NAND.

In the unary case, the *last* index of the ternary state provides the unary input for the unary sub-game. The first index of the ternary state distinguishes the top two possibilities from the bottom two, and the second index distinguishes the top unary sub-game from the bottom one (for each partition). In the binary case, there are 16 unique binary operations that we might take account of, assuming the order of the outputs matters. When the sender and receiver play a game which takes advantage of an underlying binary predisposition, we have a signalling game with two states, two signals, and 16 actions.

As with the binary-input NAND game, the players might co-evolve their strategies. The co-evolved ternary-input NAND game may also be modelled in several different

		a_1 $\mathbf{0}$	a_2 $\mathbf{1}$	
s_{000}	$\langle \mathbf{0}, \mathbf{0}, 0 \rangle$	0	1	TAUT
s_{001}	$\langle \mathbf{0}, \mathbf{0}, 1 \rangle$	0	1	
s_{010}	$\langle \mathbf{0}, \mathbf{1}, 0 \rangle$	0	1	TAUT
s_{011}	$\langle \mathbf{0}, \mathbf{1}, 1 \rangle$	0	1	
s_{100}	$\langle \mathbf{1}, \mathbf{0}, 0 \rangle$	0	1	TAUT
s_{101}	$\langle \mathbf{1}, \mathbf{0}, 1 \rangle$	0	1	
s_{110}	$\langle \mathbf{1}, \mathbf{1}, 0 \rangle$	0	1	NEG
s_{111}	$\langle \mathbf{1}, \mathbf{1}, 1 \rangle$	1	0	

(A) Ternary logical operator as the composition of two binary logical operators

		a_1 $\mathbf{0}$	a_2 $\mathbf{1}$	
s_{000}	$\langle \mathbf{0}, 0, 0 \rangle$	0	1	2-TAUT
s_{001}	$\langle \mathbf{0}, 0, 1 \rangle$	0	1	
s_{010}	$\langle \mathbf{0}, 1, 0 \rangle$	0	1	
s_{011}	$\langle \mathbf{0}, 1, 1 \rangle$	0	1	
s_{100}	$\langle \mathbf{1}, 0, 0 \rangle$	0	1	2-NAND
s_{101}	$\langle \mathbf{1}, 0, 1 \rangle$	0	1	
s_{110}	$\langle \mathbf{1}, 1, 0 \rangle$	0	1	
s_{111}	$\langle \mathbf{1}, 1, 1 \rangle$	1	0	

(B) Ternary logical operator as the composition of two binary logical operators

TABLE 11. Two different ways of composing a ternary NAND operation. (a) shows the composition of four unary operations, whereas (b) shows the composition of two binary operations.

ways, depending on whether or not the underlying co-evolved game is a logic game with unary inputs or binary inputs. As the dimension of the game increases, so too do the degrees of freedom concerning modelling decisions.

5. Conclusion

One thing that came out in the discussion of unary functions is that the ID- and NEG games are structurally more similar to one another than they are to either of the TAUT- and CONT games, and vice-versa. This highlights a subtlety that the analysis of Barrett and Skyrms (2017) ignores. Barrett and Skyrms (2017) suggest that "[o]nce NAND has evolved, it may be appropriated to a new context by template transfer to play the role of a different logical operator more efficiently than that operator might evolve on its own" (350-1). They report that evolving OR from NAND happens an order of magnitude faster than evolving OR from scratch.

In a suggestive footnote, they point out that "[m]ore generally, if one had five binary logical operators, one for each number of false outputs, one could get the other eleven by template transfer. And each would evolve an order of magnitude faster this way than on its own." However, it would not do to have *four* binary logical operators. Further, template transfer fails to explain how one might be able to evolve, e.g., AND from NAND, given that they have a different number of 'false' values in their truth tables. A virtue of the model presented here is that it is obvious how one would evolve each binary operator from combinations of two unary operators—it was mentioned that the unique permutations of two unary sub-games cover all 16 binary games. Finally, template transfer does not show how to evolve a ternary-input logical operator from a binary-input logical operator. Though Barrett and Skyrms (2017) do not suggest this, their process of modular composition *more generally* should be sufficient for this purpose, as has been shown here.

The key insight was to start with *unary* logical operations, and show how these can compose to more efficiently evolve *binary* logical operations. Further, the assumption that the underlying simple disposition(s) must be pre-evolved was relaxed. Finally, this compositional extends in a natural way to learn *ternary* logical operations from two binary logical operations or four unary logical operations, and so on *ad infinitum*.

In some sense, the efficacy of template transfer on a pre-evolved disposition should come as no surprise. Since the sender dispositions in the original context are *fixed*, the sender in the new context is effectively playing a *sensory-manipulation* game. Furthermore, since the two senders are entirely independent, these branches constitute two independent 2×2 sensory-manipulation games—the work of composing these two games has already been achieved by the pre-evolved disposition. We have now seen, in a more general case, once the agents in a signalling context have evolved simple unary operators, they might be able to use these previously evolved dispositions to learn new, more complex binary, ternary, etc. operations.

Though I have discussed logical operations as a concrete example, it should be clear that it is the *process* of modular composition itself that gives rise to efficacy in the evolution of complex dispositions.

Such compositional processes (more generally) might be instantiated in nature in terms of the computational principles or neurobiological underpinnings of any adaptive decision-making process. Modular composition may arise, and aid efficacy or efficiency of such processes, in several different settings; for example, composing multiple sensory modalities, such as tactile and visual stimulation (Fazeli et al., 2019); arranging dominance-relations to form a hierarchical representation of a social group (Seyfarth and Cheney, 2018); cognitive reasoning involving hierarchically organised decision-making (Sarafyazd and Jazayeri, 2019); or other such functional-demand protocols in nature, such as the availability of food, density of populations, and presence of predators in migratory species (Hopcraft et al., 2014).

When signals mediate these processes, the models provided here illustrate particular circumstances under which we might expect modular composition to be successful. When the agents in a signalling game are understood as distinct organisms, this may give rise to complex social behaviour; when they are interpreted as functional components of a single organism, this may give rise to complex cognition.

In the context of language origins, this sort of explanation is essential since an adequate description of how *linguistic* capacities might arise from simple communicative precursors is a diachronic story of how language gets to be complex over time via a combination of genetic and cultural evolution. Results of this sort help carve out the space of possibilities for how such dispositions may have arisen in the first place.

References

Alexander, Jason McKenzie, Skyrms, Brian, and Zabell, Sandy L. (2012). Inventing New Signals. *Dynamic Games and Applications*, 2:129–145.

Argiento, Raffaelle, Pemantle, Robin, Skyrms, Brian, and Volkov, Stanislav (2009). Learning to Signal: Analysis of a Micro-Level Reinforcement Model. *Stochastic Processes and Their Applications*, 119:373–390.

Barrett, Jeffrey A. (2006). Numerical Simulations of the Lewis Signaling Game: Learning Strategies, Pooling Equilibria, and Evolution of Grammar. *Technical Report, Institute for Mathematical Behavioral Science.*

Barrett, Jeffrey A. (2007). Dynamic Partitioning and the Conventionality of Kinds. *Philosophy of Science*, 74:527–546.

Barrett, Jeffrey A. (2009). The Evolution of Coding in Signaling Games. *Theory and Decision*, 67:223–237.

Barrett, Jeffrey A. (2016). On the Evolution of Truth. *Erkenntnis*, 81:1323–1332.

Barrett, Jeffrey A. (2017). Truth and Probability in Evolutionary Games. *Journal of Experimental and Theoretical Artificial Intelligence*, 29(1):219–225.

Barrett, Jeffrey A. (2018). The Evolution, Appropriation, and Composition of Rules. *Synthese*, 195(2):623–636.

Barrett, Jeffrey A. (2019). Self-Assembling Games and the Evolution of Salience. Unpublished Manuscript. March, 2019. PDF File.

Barrett, Jeffrey A., Cochran, Calvin T., Huttegger, Simon, and Fujiwara, Naoki (2017a). Hybrid Learning in Signaling Games. *Journal of Experimental & Theoretical Artificial Intelligence*, 29(5):1–9.

Barrett, Jeffrey A. and LaCroix, Travis (2019). Epistemology and the Structure of Language. Unpublished Manuscript. September, 2019. PDF File.

Barrett, Jeffrey A. and Skyrms, Brian (2017). Self-Assembling Games. *The British Journal for the Philosophy of Science*, 68(2):329–353.

Barrett, Jeffrey A., Skyrms, Brian, and Cochran, Calvin (2018). Hierarchical Models for the Evolution of Compositional Language. Unpublished Manuscript. May, 2018. PDF File.

Barrett, Jeffrey A., Skyrms, Brian, and Mohseni, Aydin (2017b). Self-Assembling Networks. *British Journal for the Philosophy of Science*. Forthcoming.

Barrett, Jeffrey A. and Zollman, Kevin (2009). The Role of Forgetting in the Evolution and Learning of Language. *Journal of Experimental and Theoretical Artificial Intelligence*, 21(4):293–309.

Beggs, A. W. (2005). On the Convergence of Reinforcement Learning. *Journal of Economic Theory*, 122:1–36.

Bond, Alan B., Kamil, Alan C., and Balda, Russell P. (2003). Social Complexity and Transitive Inference in Corvids. *Animal Behaviour*, 65(3):479–487.

Fazeli, N., Oller, M., Wu, J., Wu, Z., Tenenbaum, J. B., and Rodriguez, A. (2019). See, Feel, Act: Hierarchical Learning for Complex Manipulation Skills with Multisensory Fusion. *Science Robotics*, 4(26):eaav3123.

Franke, Michael (2014). Creative Compositionality from Reinforcement Learning in Signaling Games. In Cartmill, Erica A., Roberts, Seán, Lyn, Heidi, and Cornish, Hannah, editors, *The Evolution of Language: Proceedings of the 10th International Conference*, volume 10 of *Evolang*, pages 82–89. World Scientific, Singapore.

Franke, Michael (2016). The Evolution of Compositionality in Signaling Games. *Journal of Logic, Language and Information*, 25(3):355–377.

Grice, H. Paul (1975). Logic and conversation. In Cole, Peter and Morgan, Jerry L., editors, *Syntax and Semantics, Vol. 3: Speech Acts*, pages 41–58. Academic Press, New York.

Hempel, Carl G. (1965). Aspects of Scientific Explanation, and Other Essays in the Philosophy of Science.

Herrnstein, Richard J. (1961). Relative and Absolute Strength of Responses as a Function of Frequency of Reinforcement. *Journal of the Experimental Analysis of Behaviour*, 4:267–272.

Herrnstein, Richard J. (1970). On the Law of Effect. *Journal of Experimental Analysis of Behavior*, 13:243–266.

Hofbauer, Josef and Huttegger, Simon (2008). The Feasibility of Communication in Binary Signaling Games. *Journal of Theoretical Biology*, 254:843–849.

Hopcraft, J. Grant C., Morales, Juan Manuel, Beyer, H. L., Borner, Markus, Mwangomo, Ephraim, Sinclair, A. R. E., Olff, Han, and Haydon, Daniel T. (2014). Competition, Predation, and Migration: Individual Choice Patterns of Serengeti

Migrants Captured by Hierarchical Models. *Ecological Monographs*, 84(3):355–372.

Huttegger, Simon M. (2007). Evolutionary Explanations of Indicatives and Imperatives. *Erkenntnis*, 66:409–436.

Huttegger, Simon M., Skyrms, Brian, Smead, Rory, and Zollman, Kevin J. S. (2010). Evolutionary Dynamics of Lewis Signaling Games: Signaling Systems vs. Partial Pooling. *Synthese*, 172(1):177–191.

Kuleshov, Volodymyr and Precup, Doina (2014). Algorithms for Multi-Armed Bandit Problems. *arXiv preprint arXiv:1402.6028*.

LaCroix, Travis (2018a). On Salience and Signaling in Sender-Receiver Games: Partial Pooling, Learning, and Focal Points. *Synthese*. Forthcoming.

LaCroix, Travis (2018b). The Correction Game or, How Pre-Evolved Communicative Dispositions Might Affect Communicative Dispositions. Unpublished Manuscript. December, 2018. PDF File.

LaCroix, Travis (2019). Accounting for Polysemy and Role Asymmetry in the Evolution of Compositional Signals. Unpublished Manuscript. June, 2019. PDF File.

Lewis, David (2002/1969). *Convention: A Philosophical Study*. Blackwell, Oxford.

Martínez, Manolo (2015). Deception in Sender-Receiver Games. *Erkenntnis*, 80(1):215–227.

Pemantle, Robin and Skyrms, Brian (2004). Network Formation by Reinforcement Learning: the Long and the Medium Run. *Mathematical Social Sciences*, 48:315–327.

Progovac, Ljiljana (2019). *A Critical Introduction to Language Evolution: Current Controversies and Future Prospects*. Springer, Berlin.

Resnik, David B. (1991). How-Possibly Explanations in Biology. *Acta Biotheoretica*, 39(2):141–149.

Roth, Alvin and Erev, Ido (1995). Learning in Extensive Form Games: Experimental Data and Simple Dynamical Models in the Intermediate Term. *Games and Economic Behavior*, 8:164–212.

Roth, Alvin and Erev, Ido (1998). Predicting How People Play Games: Reinforcement Learning in Experimental Games with Unique, Mixed Strategy Equilibria. *American Economic Review*, 88(4):848–881.

Sarafyazd, Morteza and Jazayeri, Mehrdad (2019). Hierarchical Reasoning by Neural Circuits in the Frontal Cortex. *Science*, 364(6441):eaav8911.

Seyfarth, Robert M. and Cheney, Dorothy L. (2018). The Social Origins of Language. In Seyfarth, Robert M., Cheney, Dorothy L., and Platt, Michael L., editors, *The Social Origins of Language*, pages 9–33. Princeton University Press, Princeton.

Skyrms, Brian (2000). Evolution of Inference. In Kohler, Tim and Gumerman, George, editors, *Dynamics of Human and Primate Societies*, pages 77–88. Oxford University Press, New York.

Skyrms, Brian (2010). *Signals: Evolution, Learning, & Information*. Oxford University Press, Oxford.

Skyrms, Brian (2014/1996). *Evolution of the Social Contract*. Cambridge University Press, Cambridge.

Skyrms, Brian and Barrett, Jeffrey A. (2018). Propositional Content in Signals. *Studies in the History and Philosophy of Science C*. Forthcoming.

Steinert-Threlkeld, Shane (2016). Compositional Signaling in a Complex World. *Journal of Logic, Language, and Information*, 25(3–4):379–397.

Steinert-Threlkeld, Shane (2018). Function Words and Context Variability. PSA 2018: The 26th Biennial Meeting of the Philosophy of Science Association, 1–4 November 2018.

Sutton, Richard S. and Barto, Andrew G. (1998). *Reinforcement Learning: An Introduction.* MIT Press, Cambridge.

Thorndike, Edward L. (1905). *The Elements of Psychology.* The Mason Press, Syracuse.

Thorndike, Edward L. (1911). *Animal Intelligence: Experimental Studies.* The Macmillan Company, New York.

Thorndike, Edward L. (1927). The Law of Effect. *American Journal of Psychology*, 39:212–222.

Vermorel, Joannès and Mohri, Mehryar (2005). Multi-armed Bandit Algorithms and Empirical Evaluation. In Gama, J., Camacho, R., Brazdil, P. B., Jorge, A. M., and Torgo, L., editors, *Proceedings of ECML*, volume 3720 of *Lecture Notes in Computer Science*, pages 437–448. Springer, Berlin, Heidelberg.

Wagner, Elliott O. (2014). Conventional Semantic Meaning in Signalling Games with Conflicting Interests. *British Journal for the Philosophy of Science*, 66(4):751–773.

Zollman, Kevin J. S. (2011). Separating Directives and Assertions Using Simple Signaling Games. *The Journal of Philosophy*, 108(3):158–169.

Zollman, Kevin J. S., Bergstrom, Carl T., and Huttegger, Simon M. (2012). Between Cheap and Costly Signals: The Evolution of Partially Honest Communication. *Proceedings of the Royal Society B: Biological Sciences*, 280(1750):20121878.