Supplementary Methods
Model Overview, Design Concepts, and Details

This model description follows the ODD (Overview, Design concepts, Details) protocol for describing individual- and agent-based models [1], as updated by Grimm et al. [2].

Overview

1. Purpose

The purpose of this model is to illustrate how a preference for aggregating with familiar individuals can promote the evolution of cooperation. In particular, it demonstrates how a preference for aggregating with those an individual has interacted with before can make cooperation an evolutionarily stable strategy (ESS) in fission-fusion populations in the absence of other mechanisms for the evolution of cooperation [3], including association among genetically related individuals [4], reciprocal cooperation [5], and tag-based recognition of cooperative or similar individuals [6].

2. Entities, State Variables, and Scales

Entities: This model is composed of 1000 agents moving around in space, which represent individuals of some focal species, analogous to schooling fish, flocking birds, or herding ungulates. There is also a global simulation environment or "Model" which manages the initialization of the model based on simulation parameters and the gathering of data (all attributes of the Model are parameters which do not change over time, and so will be defined alongside the submodels that employ them).

Scales: The model employs a two-dimensional continuous space with periodic (toroidal) boundaries, representing an infinitely large area without edge-effects. We varied the dimensions of the space to account for different population densities, around a typical space size of 90x90 units. Each unit of space with a radius of one is roughly analogous to the "personal space" of a single agent to allow for schooling or flocking interactions (after [7]) within a sizable population. We also varied the amount of space required by each agent.

Time is modeled in discrete steps, each step representing sufficient time for all agents to move a short distance, allowing for precise simulation of schooling or flocking behavior. Cooperative interactions occur on the same timescale. We varied the average generation time around a typical mean lifespan of 500 timesteps, which provided agents with sufficient time to aggregate as well as accumulate payoffs to reproduce. We ran the simulation for 100 generations, or 50,000 timesteps.

Table 1: State Variables

Agent				
Variable	Meaning	Dynamic or Static?	Type	Range
Cooperative strategy	Whether the agent employs the cooperator or defector strategy when it encounters another in a prisoner's dilemma.	Static	Boolean	True (Cooperator), False (Defector)
Familiar bias	Weight in favor of aggregating with familiar agents as opposed to unfamiliar agents.	Static	Double	From 0 (only aggregate with unfamiliar) to 1 (only aggregate with familiar)
Familiar agents	How familiar it is with the agents that it considers to be familiar.	Dynamic	Directed network of agents with double connection strengths	
Interacted agents	How familiar it is with the agents it has interacted with.	Dynamic	Directed network of agents with double connection strengths	
Memory capacity	How many other agents it can consider to be familiar at a time.	Static	Double	Greater than or equal to 0
Familiarity threshold	How strong its connection with another agent must be to consider that agent as familiar.	Static	Double	Greater than or equal to 0
Familiarity decay	How much weaker each connection gets on each step.	Static	Double	Between 0 and 1
Accumulated resources	How much payoff it has accumulated so far between prisoner's dilemmas and any exogenous resources.	Dynamic	Double	Between 0 and the Model *reproduction threshold*
Maximum lifespan	Number of steps after which it will die.	Static	Double	Greater than or equal to 0
Age	Number of steps it has been alive so far.	Dynamic	Integer	Between 0 and its *maximum lifespan*
Location	Its x-y coordinates in the space.	Dynamic	Pair of doubles	Between 0 and the *space size*
Direction of movement	Previous movement on the x and y axes.	Dynamic	Pair of doubles	Between 0 and the Model *step size*

3. Process Overview and Scheduling

On each timestep, all agents are randomized and each in turn carries out the following processes in the following order:

1. If there are any agents within the Model *interaction radius* of the agent, it chooses a random other agent to interact with and:
   (a) Both agents execute their Update Familiarity subroutine, which updates their *interacted agents* and *familiar agents* to increment the strength of their connections to each other, and determine whether they now consider each other to be familiar.
   (b) The agent executes its Interact subroutine, which updates both agents' *accumulated resources* based on the outcome of a prisoner's dilemma game.
2. The agent executes its Move subroutine, which updates its *location* and *direction of movement* based on its current *direction of movement* and the *location*s, *direction*s *of movement*, and familiarity of the agents within the Model *aggregation radius* of the agent.
3. The agent executes its Prune Network subroutine, which updates *interacted agents* and *familiar agents* according to its *familiarity decay*.
4. The agent adjusts its *accumulated resources* by the Model *external payoff* parameter.
5. The agent increments its *age* by one.
6. The agent executes its Reproduce subroutine to determine whether it has enough *accumulated resources* to create a new agent, and if so, adjusts its *accumulated resources* according to the Model *reproduction cost*.
7. The agent executes its Die subroutine to determine whether it needs to be removed from the population based on its *age* and *accumulated resources*.

On each step, the order in which the agents act is randomized to avoid order effects. Typically, all agents move a much smaller distance in the space than their *interaction radius*, so the order in which Move and Interact occurs is largely arbitrary. Prune Network occurs after Move, so that the agents have an opportunity to aggregate with newly familiar individuals before their connection is decremented, possibly resulting in that agent becoming no longer familiar. All resources are accumulated before the agent carries out the Reproduce and Die subroutines, enabling it to more accurately assess whether it has sufficient *accumulated resources* to reproduce on that step and whether it is truly out of resources. The order of Reproduce and Die is largely arbitrary because agents only reproduce if they have sufficient resources, and the *reproduction cost* is never higher than the *reproduction threshold* so agents never die immediately after reproduction. The only exception is when an agent has passed its *lifespan*; Reproduce comes before Die to give the agent a final chance to reproduce before it dies, but this should make little difference given how long agents' *lifespan*s are.

Algorithm 1 Step. On each step, all agents are randomized and each may have the opportunity to interact with a neighboring agent and acquire resources and update their familiarity accordingly, move, prune any inactive connections, reproduce, and die.

if *this* agent has any neighbors within *interaction radius* units of its *location* **then**
  CHOOSE a random *neighbor*
  UPDATE FAMILIARITY between *this* agent and its *neighbor*
  INTERACT with its *neighbor*
end if
MOVE
PRUNE NETWORK of *this* agent's *familiar agents* and *interacted agents*
ADJUST *this* agent's *accumulated resources* by *external resources*
INCREMENT *this* agent's *age* by 1
if *this* agent's *accumulated resources* $\geq$ *reproduction threshold* **then**
  REPRODUCE
end if
if *this* agent's *age* $>$ its *lifespan* **or** (its *accumulated resources* < 0 **and** *harsh environment*) **then**
  DIE
end if

4. Design Concepts

Basic Principles

This model is built upon an extensive literature on the evolution of cooperation (e.g. [3]). Many models have explored how various kinds of movement in space can promote or inhibit the evolution of cooperation (e.g. [8, 9]). This model differs from most other models in the field in that it does not rely on population viscosity, in which offspring remain near their parents, allowing for kin-biased interactions, and therefore kin-selection to favor the evolution of cooperation. Instead, in this model, offspring are placed randomly in the space, simulating wide dispersal, and giving them an at best random chance of interacting with relatives. We likewise eliminate other known mechanisms for promoting the evolution of cooperation, such as reciprocity [5], where agents respond to the past actions of their partners, and tag-based cooperation [6], where cooperators identify each other based on phenotypic tags. Instead, our agents cooperate or defect indiscriminately, and aggregate with no regard to phenotype or past behavior, and with no *a priori* mechanism for familiarity to preferentially arise between cooperators.

This model is also motivated by an empirical literature on widespread preferences for associating and cooperating with familiar individuals across taxa (e.g. [10, 11]). To our knowledge, the closest models have gotten to exploring this phenomenon is simulating the evolution of cooperation on social networks (e.g. [3, 12]), which are typically not instantiated in populations of agents that also move in space, unlike the animals that they represent. Also, similar to the majority of spatial models exploring the evolution of cooperation, most simulations of the evolution of cooperation on social networks rely on population viscosity in the form of offspring being initialized with connections to their parents, which we likewise do not employ, instead allowing cooperation to emerge from the population structure alone.

The theories and past approaches that motivate our implementation of the various submodels will be addressed in those submodels.

Details

5. Initialization

The model is initialized with the maximum number of agents - i.e. *maximum population size* - (created by the Create Agent submodel), to reflect a stable population of cooperators at carrying capacity. At initialization, all agents are placed at random x-y *location*s in continuous space and into empty familiarity and interaction networks. Each agent's *age* is randomized by drawing uniformly from 0 to its *maximum* *age* (determined in Create Agent), and its *accumulated resources* are randomized by drawing uniformly from agents' *starting resources* at birth to the *reproduction threshold*, or the maximum amount of resources required for reproduction, to reflect a typical state of a population with overlapping generations and minimize initialization effects. To further decrease the effects of initial conditions, there is a burn-in period of 1000 steps, or approximately 2 generations, in which there is no mutation, to allow aggregations and familiarity to form before cooperative strategy is given the chance to evolve - or go extinct.

Algorithm 2 Initialize. Create a population of the maximum number of agents with randomly drawn ages and accumulated resources.

for 0 to *maximum population size* **do**
  Create Agent a
  SET agent a's *cooperative strategy* = **true** (Cooperator)
  INITIALIZE agent a's $age \sim \mathcal{U}(0,\ a\text{'s } lifespan)$
  INITIALIZE agent a's $accumulated\ resources \sim \mathcal{U}(starting\ resources, reproduction\ threshold)$
end for

6. Input Data

This model does not use input data to represent time-varying processes.

7. Submodels

Parameters

Table 2: Parameters

Name	Meaning	Type	Default Value (and Units)	Range Analyzed	Source
		Initialization			
Maximum population size	Initial and maximum number of agents.	Integer	1000 agents	1000	
Space dimensions	Length and width of the space.	Integer	90 units	50-100	
		Create Agent			
Starting resources	Amount of accumulated resources all agents are initialized with.	Double	0 resources	0-1	
Mean lifespan	Mean for drawing agents' maximum lifespan from a gamma distribution.	Double	500 steps	100-5000	
Standard deviation in lifespan	Standard deviation from drawing agents' maximum lifespan from a gamma distribution.	Double	.1	0-.9	
Mean familiar bias	Mean for drawing agents' familiar bias from a beta distribution.	Double	.9	.5-1	
Mean familiarity decay	Mean for drawing agents' familiarity decay from a beta distribution.	Double	.1 units of familiarity	0-.7	

Mean familiarity threshold	Mean for drawing agents' familiarity threshold from a gamma distribution.	Double	5 units of familiarity	1-100	
Mean memory capacity	Mean for drawing agents' memory capacities from a gamma distribution.	Integer	5 agents	1-100	
Variance in familiarity parameters	Degree of variation for drawing agents' familiar bias, familiarity threshold, familiarity decay, and memory capacity.	Integer	.1	0-1	
			Interact		
Interaction radius	Distance at which an agent looks for other agents to interact with.	Integer	2 units	1-50	Joshi et al. [7]
Benefit of cooperation	Benefit provided by cooperators in the prisoner's dilemma.	Double	.5 resources	.001-1	
Cost:benefit ratio of cooperation	Proportion of the benefit of cooperation paid by cooperators as a cost in the prisoner's dilemma.	Double	.5	.01-1	
			Move		
Repulsion radius	Distance at which an agent moves away from its neighbors.	Double	1 unit	1-6	Joshi et al. [7]
Aggregation radius	Distance at which an agent moves toward and aligns with its neighbors.	Double	6 units	2-6	[7]
Flocking weight	Relative weight of aligning with other agents as opposed to moving toward them.	Double	.5	0-1	Joshi et al. [7]
Persistence weight	Relative weight of continuing in the same direction as opposed to aggregating with others.	Double	.2	0-1	Joshi et al. [7]
Movement error	Standard deviation of random error in movement.	Double	.05	0-1	Joshi et al. [7]
Maximum rotation	Maximum change in direction of movement.	Double	50 degrees	25-360	Joshi et al. [7]
Step size	Distance moved on each timestep.	Double	.2 units	.05-2	Joshi et al. [7]
			Adjust Resources		
External payoff	Amount of resources all agents lose or gain on each step irrespective of interaction.	Double	-.05	-.1-1	Andras et al. [13]
			Reproduce		
Reproduction threshold	Accumulated resources required to reproduce.	Double	1 resource	1 (Sets the scale)	
Reproduction cost	Resources lost when an agent reproduces.	Double	1 resource	0-.9	
Mutation rate	Probability of having offspring of the opposite cooperative strategy from their parent.	Double	.01	0-.5	
			Die		
Harsh environment	Whether agents die if their accumulated resources drop below 0.	Boolean	True	True, False	Smaldino et al. [14]

Create Agents

An agent is created at a random *location* in the space at a pair of uniformly drawn x-y coordinates, with a random *direction of movement* drawn uniformly from 0 to 360 degrees. It is initialized with no *interacted agents* or *familiar agents*. Its *lifespan* is drawn from a gamma distribution [15] with the Model *mean lifespan* and *standard deviation of lifespan*, and a minimum of 1. Its *accumulated resources* are initialized at the Model *starting resources*. The agent's *familiar bias* and *familiarity decay* are drawn from a beta distribution, as they both range from 0 to 1, with a mean (μ) equal to the Model *mean familiar bias* and *mean familiarity decay* respectively, and "sample size" [16] derived from the Model *variance in familiarity parameters* (v; Equation 1) to allow for a range of variance from none when *variance in familiarity parameters* is 0, to a nearly uniform distribution when it is equal to 1. This results in α and β parameters

$$\alpha = \frac{\mu}{(\frac{3}{4}v)^2} \qquad \beta = \frac{1-\mu}{(\frac{3}{4}v)^2}. \tag{1}$$

The agent's *familiarity threshold* and *memory capacity* are drawn from a gamma distribution with a mean of the Model *mean familiarity threshold* and *mean memory capacity* respectively, and standard deviation equal to the Model *variance in familiarity parameters*. The agent's *cooperative strategy* is provided, either in model initialization (see Initialization) or by the agent's parent (see Reproduce).

Algorithm 3 Create Agent. When a new agent is created, it's given randomly drawn familiarity parameters and lifespan, an age of zero, the initial amount of resources for that simulation, and a random location and direction of movement.

SET *lifespan* $\sim \Gamma(mean\ lifespan,\ standard\ deviation\ in\ lifespan)$
SET *familiar bias* $\sim$Beta$(mean\ familiar\ bias,\ variation\ in\ familiarity\ parameters)$
SET *memory capacity* $\sim \Gamma(mean\ memory,\ variation\ in\ familiarity\ parameters)$
SET *familiarity threshold* $\sim \Gamma(mean\ familiarity\ threshold,\ variation\ in\ familiarity\ parameters)$
SET *familiarity decay* $\sim$Beta$(mean\ familiarity\ decay,\ variation\ in\ familiarity\ parameters)$
INITIALIZE *age* $= 0$
INITIALIZE *accumulated resources* $=$ *starting resources*
INITIALIZE *location* $=$ random coordinates in the space $\sim \mathcal{U}(0, space\ dimensions)$
INITIALIZE *direction of movement* $\sim \mathcal{U}(0, 360)$

Interact

If any other agents are within this agent's interaction radius (i.e. within *interaction radius* units of its *location*), then one of the other agents is chosen at random to play this agent in a prisoner's dilemma to test the most stringent case for the evolution of cooperation. If an agent is a cooperator, then its *accumulated resources* are decremented by the *benefit of cooperation* times the *cost:benefit ratio of cooperation* (c) and its partner's *accumulated resources* are incremented by the *benefit of cooperation* (b), while defectors pay no cost and provide their partners with no benefit (Table).

		Player 2: *Cooperate*	Player 2: *Defect*
Player 1	*Cooperate*	$(b-c,\ b-c)$	$(-c, b)$
	Defect	$(b, -c)$	$(0, 0)$

Table 3: Prisoner's Dilemma Payoff Matrix.

Algorithm 4 Interact. Agents play a prisoner's dilemma and gain or lose resources accordingly.

GIVEN a *partner* agent randomly chosen from within *this* agent's *interaction radius*
if *this* agent's *cooperative strategy* = **true** (Cooperator) **then**
  DECREMENT *this* agent's *accumulated resources* by the *cost of cooperation*
  INCREMENT its *partner*'s *accumulated resources* by the *benefit of cooperation*
end if
if *partner*'s *cooperative strategy* = **true** (Cooperator) **then**
  DECREMENT *partner*'s *accumulated resources* by the *cost of cooperation*
  INCREMENT *this* agent's *accumulated resources* by the *benefit of cooperation*
end if

Update Familiarity

Whenever a pair of agents interact, they each increment their familiarity with each other by one (see Equation 9 for the net change in connection strength on each step) and then decide whether to consider the other to be familiar. If an agent's partner is not in its *interacted agents*, then the agent creates a new connection to its partner with a strength of one, otherwise, the strength of their existing connection is just incremented. If an agent considers its partner to be familiar, then the strength of their connection in its *familiar agents* is also incremented, and there is nothing more it needs to do.

If an agent does not consider its partner to be familiar and the strength of their connection in *interacted agents* ($I_{i,j}$) is greater than the agent's *familiarity threshold* (h_i), then it can become familiar ($F_{i,j}$) if the agent's number of *familiar agents* is lower than its *memory capacity* (m_i) or the strength of the agent's connection to its partner is greater than the strength of the agent's weakest connection in *familiar agents* (M_i; Equation 2). If the agent's memory is at capacity (i.e. its number of *familiar agents* is equal to its *memory capacity*), then, upon becoming familiar, its partner replaces the weakest connection already in *familiar agents.*

$$
\begin{aligned}
F_{i,j} &= 1 \text{ if agent } i \text{ considers agent } j \text{ to be familiar, and 0 otherwise} \\
I_{i,j} &= \text{the strength of the connection between agents } i \text{ and } j \\
h_i &= \text{the } \textit{familiarity threshold} \text{ of agent } i \\
m_i &= \text{the } \textit{memory capacity} \text{ of agent } i \\
M_i &= \text{a vector of all of agent } i\text{'s familiar connections and their strengths}
\end{aligned}
$$

$$
F_{i,j} = \begin{cases} 1, & \textbf{if } I_{i,j} > h_i \textbf{ and } (\sum F_i < m_i \textbf{ or } MIN(M_i) < I_{i,j}) \\ 0, & \textbf{otherwise} \end{cases} \tag{2}
$$

Algorithm 5 Update Familiarity. When two agents interact, increase the strength of their connection and determine whether they become familiar.

GIVEN a *partner* agent randomly chosen from within *this* agent's *interaction radius*
if *partner* is not in *this* agent's *interacted agents* **then**
  ADD *partner* to *interacted agents*
  SET *connection strength* = 1
else
  INCREMENT *connection strength* by 1
end if
if *partner* is not in *this* agent's $familiar\ agents$ **then**
  if *connection strength* > *this* agent's *familiarity threshold* **then**
    if the size of *this* agent's $familiar\ agents$ < *this* agent's *memory capacity* **then**
      ADD *partner* to $familiar\ agents$
    else if *connection strength* > MIN(connection in $familiar\ agents$) **then**
      REMOVE MIN(connection in $familiar\ agents$) from $familiar\ agents$
      ADD *partner* to $familiar\ agents$
    end if
  end if
end if

Move

On each step, each agent (i) updates its *location* (l_i) and *direction of movement* (d_i) according to the active movement model of Joshi et al. [7], modified to take into account familiarity between agents ($F_{i,j}$).

$$\begin{aligned}
d_i &= \text{the } direction\ of\ movement \text{ of agent } i \\
l_i &= \text{the } \textit{location} \text{ of agent } i \\
R_i &= \text{array of agents within the } \textit{repulsion radius} \text{ of agent } i \\
A_i &= \text{array of agents within the } \textit{aggregation radius} \text{ of agent } i \\
w_f &= flocking\ weight \\
w_p &= persistence\ weight \\
x &= maximum\ rotation \\
s &= step\ size \\
\epsilon &= \text{random error in movement } \sim \mathcal{N}(0, movement\ error) \\
b_i &= \text{the } familiar\ bias \text{ of agent } i \\
F_{i,j} &= 1 \text{ if agent } i \text{ considers agent } j \text{ to be familiar, and 0 otherwise}
\end{aligned}$$

If there are any other agents (R_i) within the agent's repulsion radius (i.e. within *repulsion radius* units of its *location*), then the agent moves in the opposite direction of their average location (Equation 3). Repulsion takes precedence over all other movement decisions.

$$r_i = \frac{1}{|R_i|} \sum_{j}^{R_i} \frac{l_i - l_j}{\|l_i - l_j\|} \tag{3}$$

Otherwise, if there are any other agents (A_i) within the agent's aggregation radius (i.e. within *aggregation radius* units of its *location*), then the agent moves to align with (Equation 4)

$$f_i = \frac{1}{|A_i|} \sum_{j}^{A_i} d_j b_i^{F_{i,j}} (1 - b_i)^{1 - F_{i,j}} \tag{4}$$

and move toward the other agents (Equation 5)

$$o_i = \frac{1}{|A_i|} \sum_{j}^{A_i} \frac{l_j - l_i}{\|l_j - l_i\|} b_i^{F_{i,j}} (1 - b_i)^{1-F_{i,j}} \tag{5}$$

weighted by the *flocking weight* (w_f) and the agent's *familiar bias* (b_i), with some tendency to continue moving in the same direction, according to *persistence weight* (w_p; Equation 6).

$$a_i = w_f(1 - w_p)f_i + (1 - w_f)(1 - w_p)o_i + w_p d_i \tag{6}$$

Otherwise, if there are no other agents present, then the agent continues in its current *direction*. In all cases, the agent's movement is adjusted by some random error (ϵ), drawn from a normal distribution, with a mean of 0 and a standard deviation of the Model *movement error* (Equation 7).

$$d_i = \epsilon + \begin{cases} r_i, & \textbf{if } |R_i| > 0 \\ a_i, & \textbf{if } |A_i| > 0 \\ d_i, & \textbf{otherwise} \end{cases} \tag{7}$$

All movement is constrained by the *maximum rotation*, such that, if the difference between the agent's current *direction of movement* ($d_i(t-1)$) and the new calculated direction (d_i) is greater than *maximum rotation*, its *direction of movement* is only incremented by the *maximum rotation* in that direction (Equation 8). The agent will then move *step size* units in the calculated direction.

$$d_i(t) = \begin{cases} d_i, & \textbf{if } \text{turn}(d_i) < \textit{maximum rotation} \\ d_i(t-1) + \textit{maximum rotation}, & \textbf{otherwise} \end{cases} \tag{8}$$

Algorithm 6 Move. Each step, each agent moves according to the active movement model of Joshi et al. [7]. If there are any agents within its repulsion radius, the agent moves away from them. Otherwise, if there are any agents within its aggregation radius, it moves toward them and aligns with their direction of movement. Otherwise, the agent persists in moving at its previous trajectory with some random error.

if *this* agent has any *neighbors* within *repulsion radius* units of its *location* **then**
  SET *new direction* = average direction away from *neighbors*
else if *this* agent has any *neighbors* within *aggregation radius* units of its *location* **then**
  for each *neighbor* **do**
    if n is in *this* agent's *familiar agents* **then**
      SET *weight* = *this* agent's *familiar bias*
    else
      SET *weight* = 1 - *this* agent's *familiar bias*
    end if
    INCREMENT *flocking direction* by *neighbor*'s *direction of movement* $*$ *weight*
    INCREMENT *aggregating direction* by the direction from *this* agent to *neighbor* $*$ *weight*
  end for
  SET *new direction* = $(1 - \textit{persistence weight}) * \textit{flocking weight} * \textit{flocking direction}$
      $+ (1 - \textit{persistence weight}) * (1 - \textit{flocking weight}) * \textit{aggregating direction}$
      $+ \textit{persistence weight} * \textit{direction of movement}$ + error $(\sim \mathcal{N}(0, \textit{movement error}))$
else
  SET *new direction* = current *direction of movement* + error $(\sim \mathcal{N}(0, \textit{movement error}))$
end if
if |*new direction* - *direction of movement*| > *maximum rotation* **then**
  SET *this* agent's *direction of movement* = current *direction of movement* + *maximum rotation* in the desired direction
else
  SET *this* agent's *direction of movement* = *new direction*
end if
SET *this* agent's *location* = its previous *location* + *step size***direction of movement*

Prune Connections

On each step, each agent goes through its *interacted agents* (I_i) and *familiar agents* and decrements each connection by its *familiarity decay*, resulting in a net change on a given step as described by Equation 9.

$$\Delta I_{i,j} = \begin{cases} 1 - decay_i, & \textbf{if } i \text{ and } j \text{ interacted on this step} \\ -decay_i, & \textbf{otherwise} \end{cases} \tag{9}$$

If any connection strength drops below 0, the connection is removed from both *interacted agents* and *familiar agents*, and it is no longer considered to be familiar.

Algorithm 7 Prune Connections. Each step, each agent goes through its connections, decrements their strength by its decay rate, and removes the ones whose strength is less than 0.

```
for each connection in this agent's interacted agents do
  DECREMENT the strength of this connection by this agent's decay rate
  if the strength of this connection < 0 then
    REMOVE this connection from interacted agents
    if this connection is in familiar agents as well then
      REMOVE this connection from familiar agents
    end if
  end if
end for
```

Reproduce

If an agent's total *accumulated resources* is greater than the *reproduction threshold*, it reproduces, subtracting the *reproduction cost* from its *accumulated resources*. If the actual number of agents in the population at that time is less than the *maximum population size*, then a new agent is created with the same *cooperative strategy* as its parent, with a probability of switching to the other strategy equal to the *mutation rate*. Otherwise, the offspring is assumed to have died due to high juvenile mortality, and no agent is created. This is analogous to a Moran process [17], but allows for the population size to vary below a maximum.

Algorithm 8 Reproduce. If an agent reaches the reproduction threshold for the simulation, it has a chance to reproduce. If the population size is not at maximum, a new agent is actually created with the same strategy as its parent, with a set chance of mutation, and placed on the grid within a set distance. Either way, the agent loses the resources required for reproduction. Each agent also may lose or gain a set amount of resources each step, depending on the condition.

```
ADJUST this agent's resources by the amount lost/gained irrespective of cooperation for this simulation
if this agent's accumulated resources > reproduction threshold then
  DECREMENT this agent's accumualted resources by the reproduction cost
  if the total number of agents < maximum population size then
    CREATE AGENT
    if a draw ~ U(0,1) < simulation mutation rate then
      SET the new agent's cooperative strategy = the opposite of this agent's cooperative strategy
    else
      SET the new agent's cooperative strategy = this agent's cooperative strategy
    end if
  end if
end if
```

Die

If an agent's *age* is greater than its *maximum lifespan*, then it is removed from the simulation. Additionally, if *harsh environment* is true and the agent's *accumulated resources* drop below 0, then it is removed from the simulation irrespective of its *age*. Such harsh environments have previously been found to promote the evolution of cooperation [14].

Algorithm 9 Die. If an agent has surpassed its lifespan, it is removed from the population. Agents can also be removed if their accumulated resources drop below zero in harsh environments.

```
if this agent's age > this agent's lifespan then
  REMOVE this agent from the population
else if harsh environment and this agent's accumulated resources < 0 then
  REMOVE this agent from the population
end if
```

References

[1] Grimm, V., Berger, U., Bastiansen, F., Eliassen, S., Ginot, V., Giske, J., Goss-Custard, J., Grand, T., Heinz, S. K., Huse, G., Huth, A., Jepsen, J. U., Jørgensen, C., Mooij, W. M., Müller, B., Pe'er, G., Piou, C., Railsback, S. F., Robbins, A. M., … DeAngelis, D. L. (2006). A standard protocol for describing individual-based and agent-based models. *Ecological Modelling*, *198*, 115–126. https://doi.org/10.1016/j.ecolmodel.2006.04.023

[2] Grimm, V., Railsback, S. F., Vincenot, C. E., Berger, U., Gallagher, C., Deangelis, D. L., Edmonds, B., Ge, J., Giske, J., Groeneveld, J., Johnston, A. S., Milles, A., Nabe-Nielsen, J., Polhill, J. G., Radchuk, V., Rohwäder, M. S., Stillman, R. A., Thiele, J. C., & Ayllón, D. (2020). The odd protocol for describing agent-based and other simulation models: A second update to improve clarity, replication, and structural realism. *Journal of Artificial Societies and Social Simulation*, *23*, 7. https://doi.org/10.18564/jasss.4259

[3] Nowak, M. A. (2006). Five rules for the evolution of cooperation. *Science*, *314*, 1560–1563. https://doi.org/10.1126/science.1133755

[4] Hamilton, W. D. (1964). The genetical evolution of social behaviour. i. *Journal of Theoretical Biology*, *7*, 1–16.

[5] Axelrod, R., & Hamilton, W. D. (1981). The evolution of cooperation. *Science*, *211*, 1390–1396. https://doi.org/10.5840/tpm201467114

[6] Sigmund, K. (2009). Sympathy and similarity: The evolutionary dynamics of cooperation [short intro]. *Proceedings of the National Academy of Sciences of the United States of America*, *106*, 8405–8406. https://doi.org/10.1073/pnas.0903947106

[7] Joshi, J., Couzin, I. D., Levin, S. A., & Guttal, V. (2017). Mobility can promote the evolution of cooperation via emergent self-assortment dynamics [Adhesion/attraction acts as a green-beard to signal cooperation, resulting in fluctuating adhesion and cooperation as cooperators try to quot;outpacequot; defectors and defectors try to catch up]. *PLoS Computational Biology*, *13*, e1005732. https://doi.org/10.1371/journal.pcbi.1005732

[8] Smaldino, P. E., & Schank, J. C. (2012). Movement patterns, social dynamics, and the evolution of cooperation. *Theoretical Population Biology*, *82*, 48–58. https://doi.org/10.1016/j.tpb.2012.03.004

[9] Aktipis, A. C. (2004). Know when to walk away: Contingent movement and the evolution of cooperation. *Journal of Theoretical Biology*, *231*, 249–260. https://doi.org/10.1016/j.jtbi.2004.06.020

[10] Griffiths, S. W., & Ward, A. (2011). *Social recognition of conspecifics*. https://doi.org/10.1002/9781444342536.ch9

[11] Temeles, E. J. (1994). The role of neighbours in territorial systems: When are they 'dear enemies'? *Animal Behaviour*, *47*, 339–350.

[12] Szabó, G., & Fáth, G. (2007). Evolutionary games on graphs. *Physics Reports*, *446*, 97–216. https://www.sciencedirect.com/science/article/pii/S0370157307001810

[13] Andras, P., Lazarus, J., & Roberts, G. (2007). Environmental adversity and uncertainty favour cooperation. *BMC Evolutionary Biology*, *7*, 1–8. https://doi.org/10.1186/1471-2148-7-240

[14] Smaldino, P. E., Schank, J. C., & McElreath, R. (2013). Increased costs of cooperation help cooperators in the long run. *American Naturalist*, *181*, 451–463. https://doi.org/10.1086/669615

[15] Lai, C. D. (2010). Generalized erlang and mortality levelling off distributions. *Mathematical and Computer Modelling*, *51*, 1268–1276. https://doi.org/10.1016/j.mcm.2010.01.011
[16] Kruschke, J. (2010). *Doing bayesian data analysis: A tutorial introduction with r*. Elsevier Science.
[17] Moran, P. A. P. (1962). *The statistical processes of evolutionary theory.* Clarendon Press.