Supplementary materials—Social contact patterns can buffer costs of forgetting in the evolution of cooperation

Jeffrey R. Stevens
Jan K. Woike
Lael J. Schooler
Stefan Lindner
Thorsten Pachur

R packages

This project used R version 3.4.4, [1] and the R packages *car* (version 3.0-0, [2]), *ggplot2* (version 2.2.1, [3]), and *here* (version 0.1, [4]).

Additional simulation methods

Social network size

To assess the effect of different social network sizes on cooperation, we conducted a set of simulations in which we paired each agent not only with the other agents in their social network of 10 but also with a 'filler' agent. This increased the total number of interactions experienced by the population agents. The filler agents had no strategy and we did not track their behavior or payoffs. This allowed us to increase the number of intervening interactions between interactions for each pair to estimate the effects of larger social networks without the computational load of tracking an increased number of interactions. We added 100, 400, and 1,400 filler interactions to the existing 100 core interactions to mimic social network sizes of 20, 50, and 150 partners. Figure 1b illustrates the original social network size (10 partners/100 interactions), plus these larger networks across the three contact pattern skews for the $\lambda = 0.87$, $\psi = 0.22$, skipping condition.

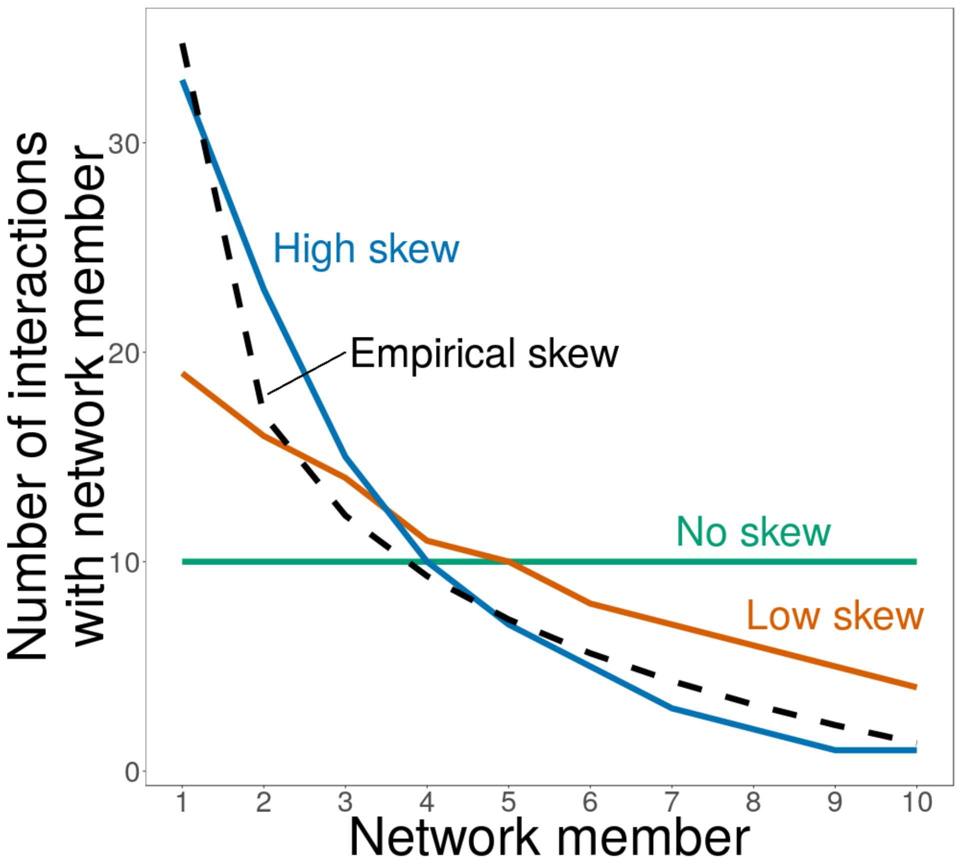
Table S1: Decision rules determining each strategy's actions.

Name	Strategy	Round 1	Round $t-2$ (Agent)	Round $t-2$ (Partner)	Round $t-1$ (Agent)	Round $t-1$ (Partner)	Reaction	Source
ALLC	Always cooperate	C	*	*	*	*	C	
ALLD	Always defect	D	*	*	*	*	D	
TFT	Tit-for-tat: cooperate in the first round, then copy partner's most recent move	C	*	*	*	C	C	[5]
		C	*	*	*	D	D	
TF2T	Tit-for-two-tats: Like TFT but only reciprocates partner's defection following two consecutive defections	C	*	D	*	D	D	[6]
		C	*	*	*	C	C	
		C	*	C	*	*	C	
GTFT	Generous tit-for-tat: mirrors partner's move stochastically, with a higher probability of copying following cooperation ($p = 0.99$) than following defection ($p = 0.67$)	C	*	*	*	C	0.99C/0.01D†	[7]
		C	*	*	*	D	0.33C/0.67D	
CTFT	Contrite tit-for-tat: behaves as TFT but forgives defection that was a response to a mistake on its part	C	*	*	*	C	C	[8]
		C	D	C	*	D	C	
		C	C	*	*	D	D	
		C	*	D	*	D	D	
GRIM	Grim trigger or Friedman: cooperates until the other party defects once and then defects in all subsequent interactions	C	*	*	*	D	D	[9]
		C	*	*	D	*	D	
		C	*	*	C	C	C	
WSLS	Win-stay-lose-shift, Simpleton, or Pavlov: starts with a cooperative choice and then cooperates when, in the previous round, the actions matched (both cooperated or both defected) and defects when the actions did not match	C	*	*	C	C	C	[5,10,11]
		C	*	*	C	D	D	
		C	*	*	D	C	D	
		C	*	*	D	D	C	
RAND	Random: chooses randomly	0.5C/0.5D	*	*	*	*	0.5C/0.5D	

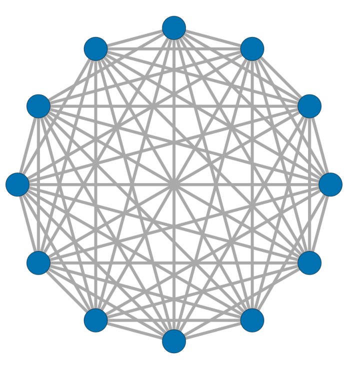
Note: * = any value would trigger the rule, C stands for cooperation and D for defection; numbers represent probabilities when reactions are stochastically determined.

† We chose $p = 0.99$ because Nowak and Sigmund (1992) found that this stochastic version of GTFT outperformed other stochastic variants in a simulation. But the results do not change if this probability is deterministic, $p = 1.0$ (Figure S8).

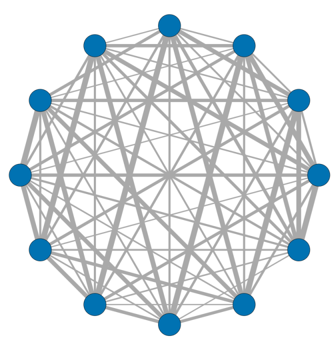
(a)



(b)



(c)



(d)

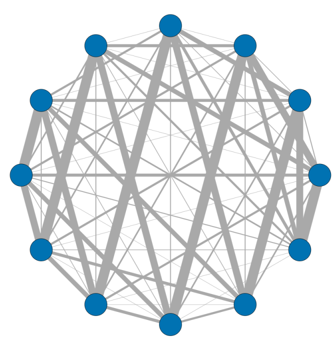


Figure S1: (a) We tested three contact patterns in our simulation. In the no-skew distribution, agents experienced 10 interactions with all 10 network members. In the low-skew distribution, agents experienced between 4-19 interactions with their network members. In the high-skew distribution, agents experienced between 1-33 interactions with their network members. The high skew distribution mirrored an empirical estimate of contact frequency [11] with slight modifications based on the constraints of the simulation. Networks graphs of subsamples of the full networks illustrate example contact patterns for (b) no skew, (c) low skew, and (d) high skew. More interactions are illustrated by thicker lines in the edge connections.

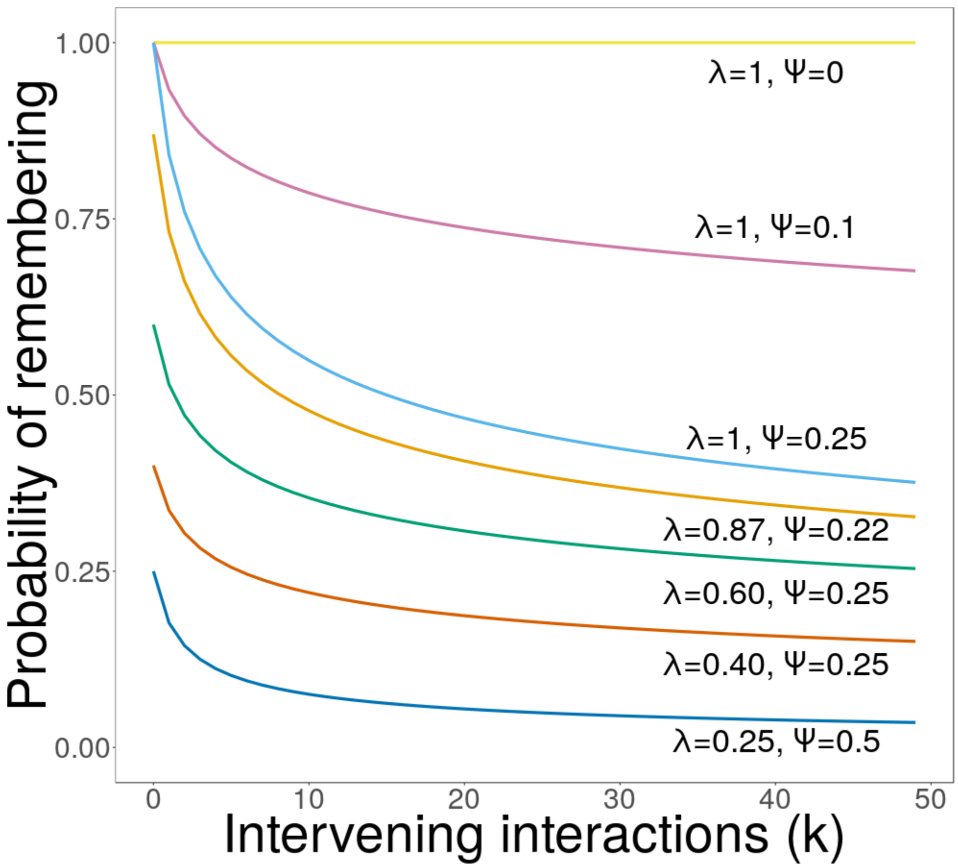


Figure S2: The probability of correctly remembering an interaction decreased with the number of intervening interactions. The curves were generated using the forgetting function: $p = \lambda(k+1)^{-\psi}$, where λ and ψ specify the starting point and decay rate of the forgetting function, and k represents the number of interactions between the target and the current interaction ($k = 0$ if the target interaction was the most recent interaction).

(a) (b)

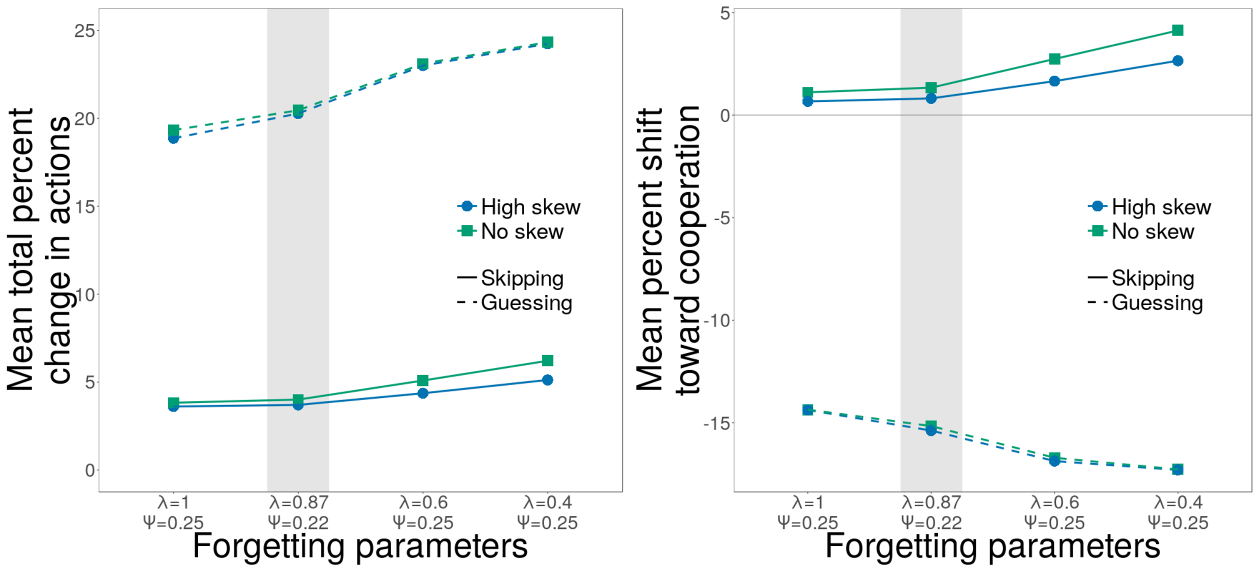


Figure S3: Effects of skew, memory error type, and forgetting rate on action selection. We conducted a series of simulations that measured how forgetting changed the actions selected by agents. For each of four sets of memory parameters, two skew levels, and the two memory error types plus perfect memory, we conducted 20 first generation (i.e., with equal numbers of strategies) simulations and tracked the frequencies of the C and D actions selected by agents. We conducted 20 repetitions of the simulations in which we used the same social network structure (i.e., identical connections between agents/strategies) for all 24 of the skew/forgetting/memory error conditions. The 20 repetitions allowed us to average over stochasticity introduced by GTFT and RAND. We then conducted 1,000 replications with randomly chosen network structure. For each simulation, we compared actions under perfect memory with those under imperfect memory for both memory error types (guessing and skipping). We calculated the percent change in C actions (Δ_C) and D actions (Δ_D) from perfect to imperfect memory (with positive values representing a change toward cooperation and negative values representing a change toward defection). (a) Summing the absolute values ($abs(\Delta_C)+abs(\Delta_D)$) gives the total percent change in actions—that is, how many differences in actions imperfect memory caused. Guessing errors generated 4-5 times as many action changes than skipping errors. (b) We then calculated the net change in actions by summing the changes in C and D actions ($\Delta_C + \Delta_D$) to generate a percent shift toward cooperation. Guessing errors resulted in large shifts toward defection, whereas skipping errors caused shifts toward more cooperation. Shaded areas highlight the empirically derived forgetting parameters.

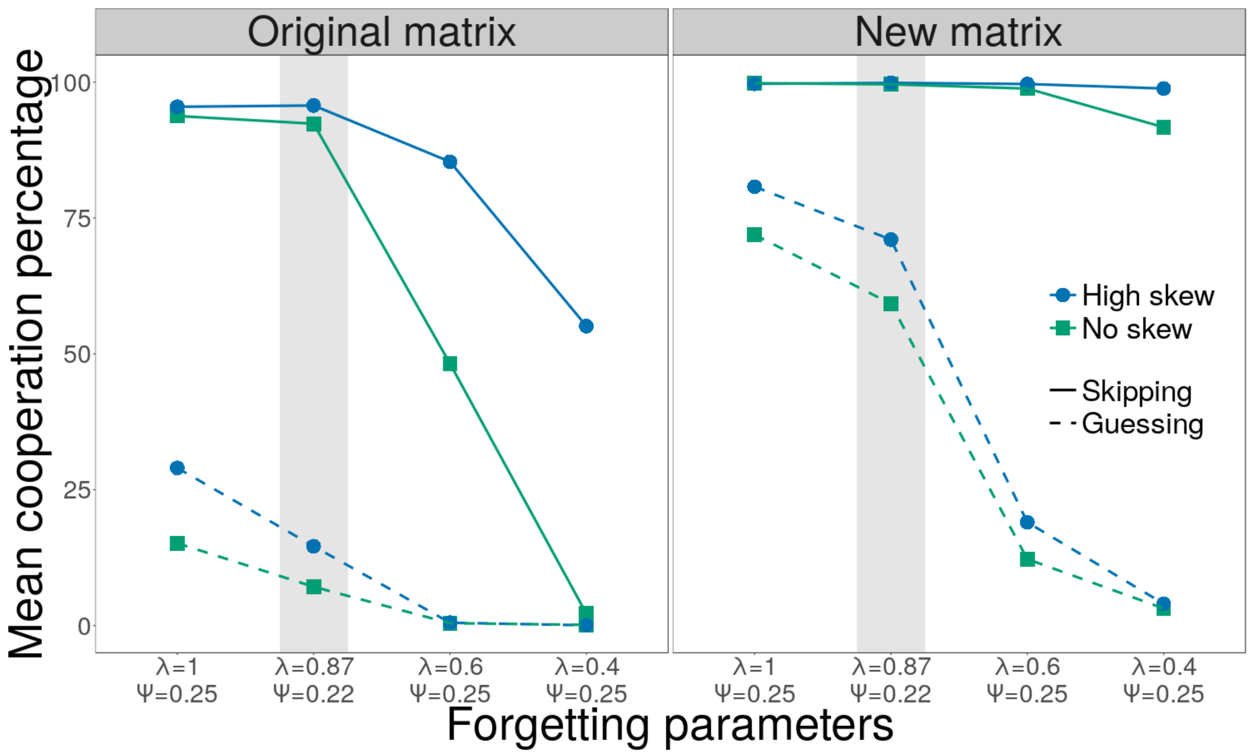


Figure S4: Effect of payoff matrix on cooperation. We used Axelrod's classic payoff matrix for the original analysis: CC=3, DC=5, CD=0, DD=1. To test the robustness of the findings to the payoff matrix, we conducted simulations with a new matrix (CC=4, DC=5, CD=0, DD=1) for a subset of the conditions (four sets of forgetting parameters, two memory error types, and two skew conditions) for 250 generations and 1,000 replicates. The new matrix increased cooperation, but skew and memory error types had similar effects as with the original matrix. Shaded area highlights the empirically derived forgetting parameters. Standard error of the mean error bars are not shown because they are smaller than the data point symbols.

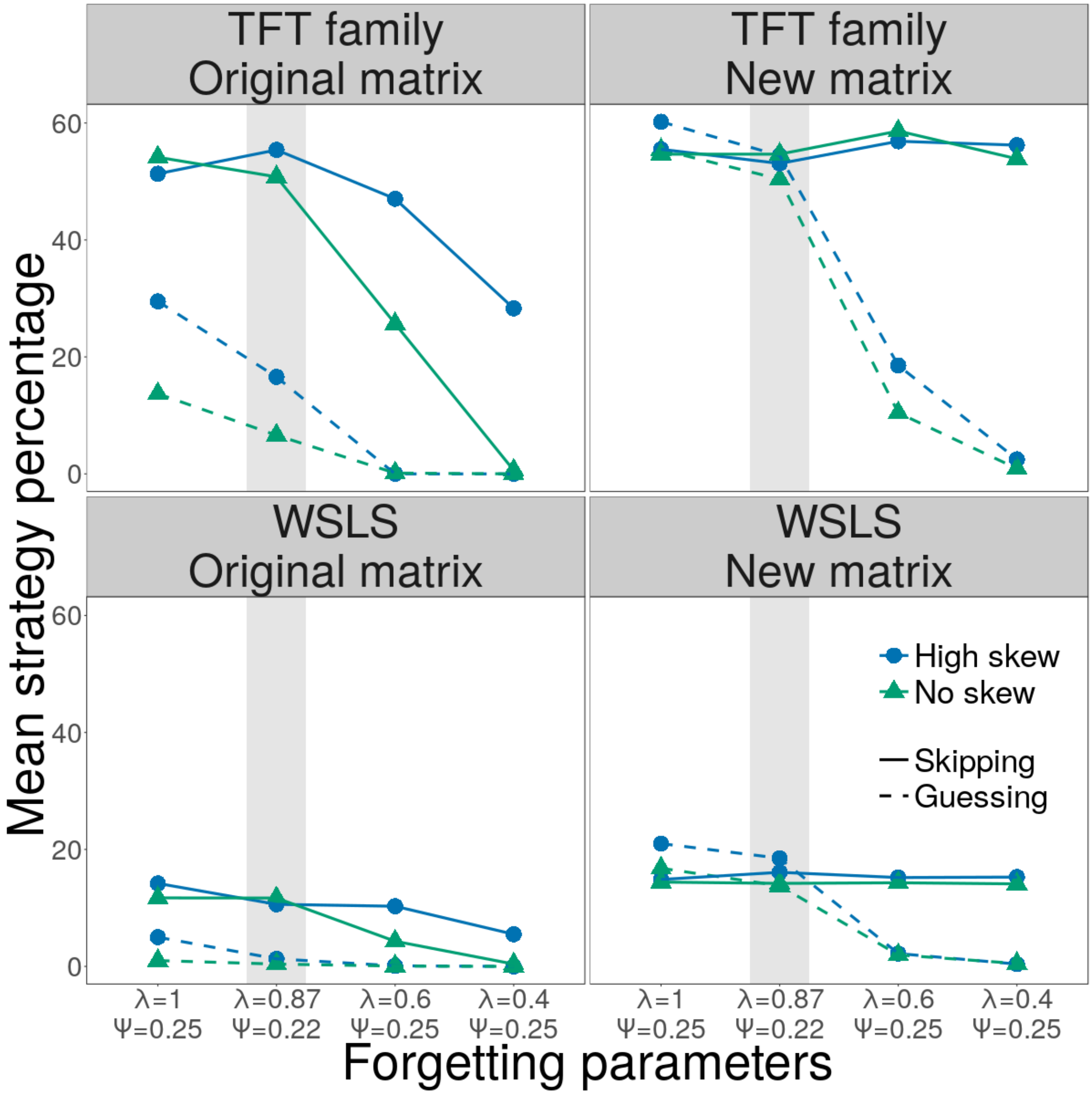


Figure S5: Effect of payoff matrix on strategy frequency for TFT family and WSLS. We used Axelrod's classic payoff matrix for the original analysis: CC=3, DC=5, CD=0, DD=1. To test the robustness of the findings to the payoff matrix, we conducted simulations with a new matrix (CC=4, DC=5, CD=0, DD=1) for a subset of the conditions (four sets of forgetting parameters, two memory error types, and two skew conditions) for 250 generations and 1,000 replicates. The strategy frequencies differed between the two matrices, but the overall pattern remained the same: if there was a difference across skews, TFT family and WSLS were more frequent with high skew. TFT family was always more frequent than WSLS in both matrices. Shaded area highlights the empirically derived forgetting parameters. Standard error of the mean error bars are not shown because they are smaller than the data point symbols.

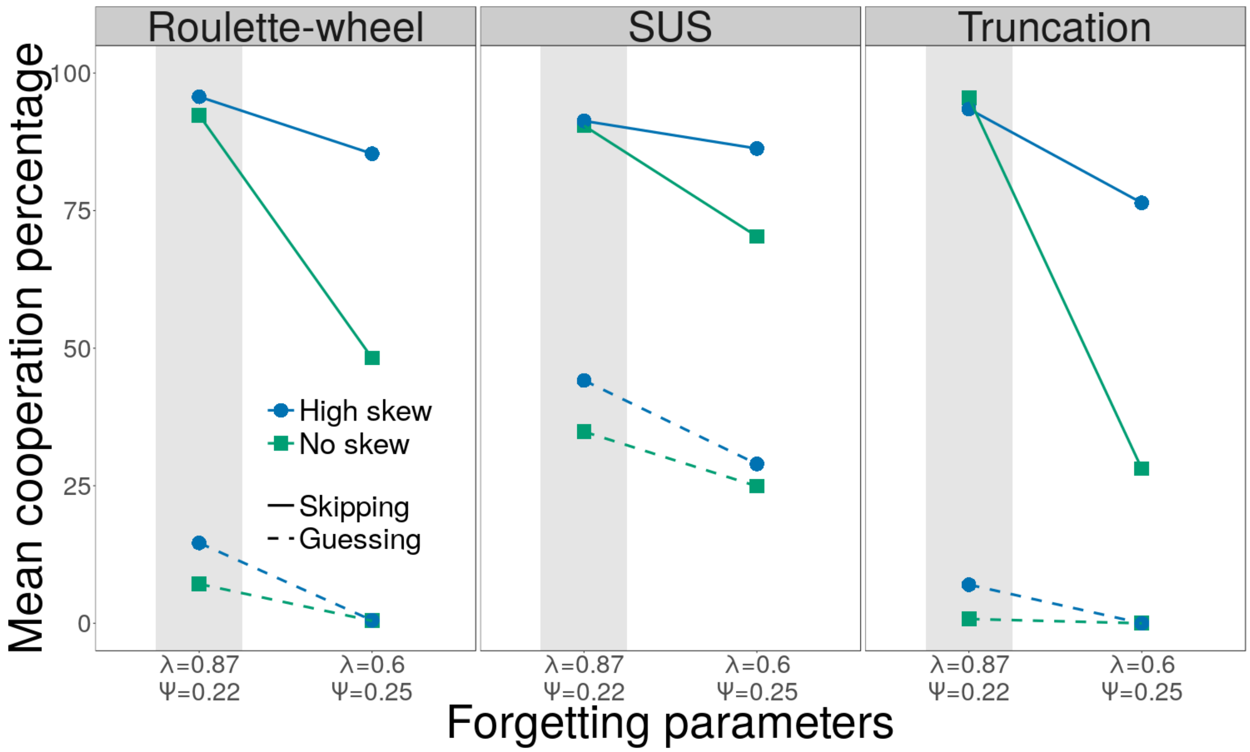


Figure S6: Effect of selection operator on cooperation. We used a standard roulette-wheel fitness selection operator for the original analysis. To test the robustness of the findings to the selection operator, we conducted simulations with two other selection operators using a subset of the conditions (two forgetting parameters, two memory error types, and two skew conditions) for 250 generations and 1,000 replicates. Stochastic universal sampling (SUS) is an unbiased selection operator that samples at evenly spaced intervals [13]. Truncation selection only samples from the top fraction of the population [14]: our simulation allowed the top 20 of the 90 agents to populate the next generation. Though the specific cooperation levels varied between selection operators, the overall effects of skew and memory error types remained. Shaded area highlights the empirically derived forgetting parameters. Standard error of the mean error bars are not shown because they are smaller than the data point symbols.

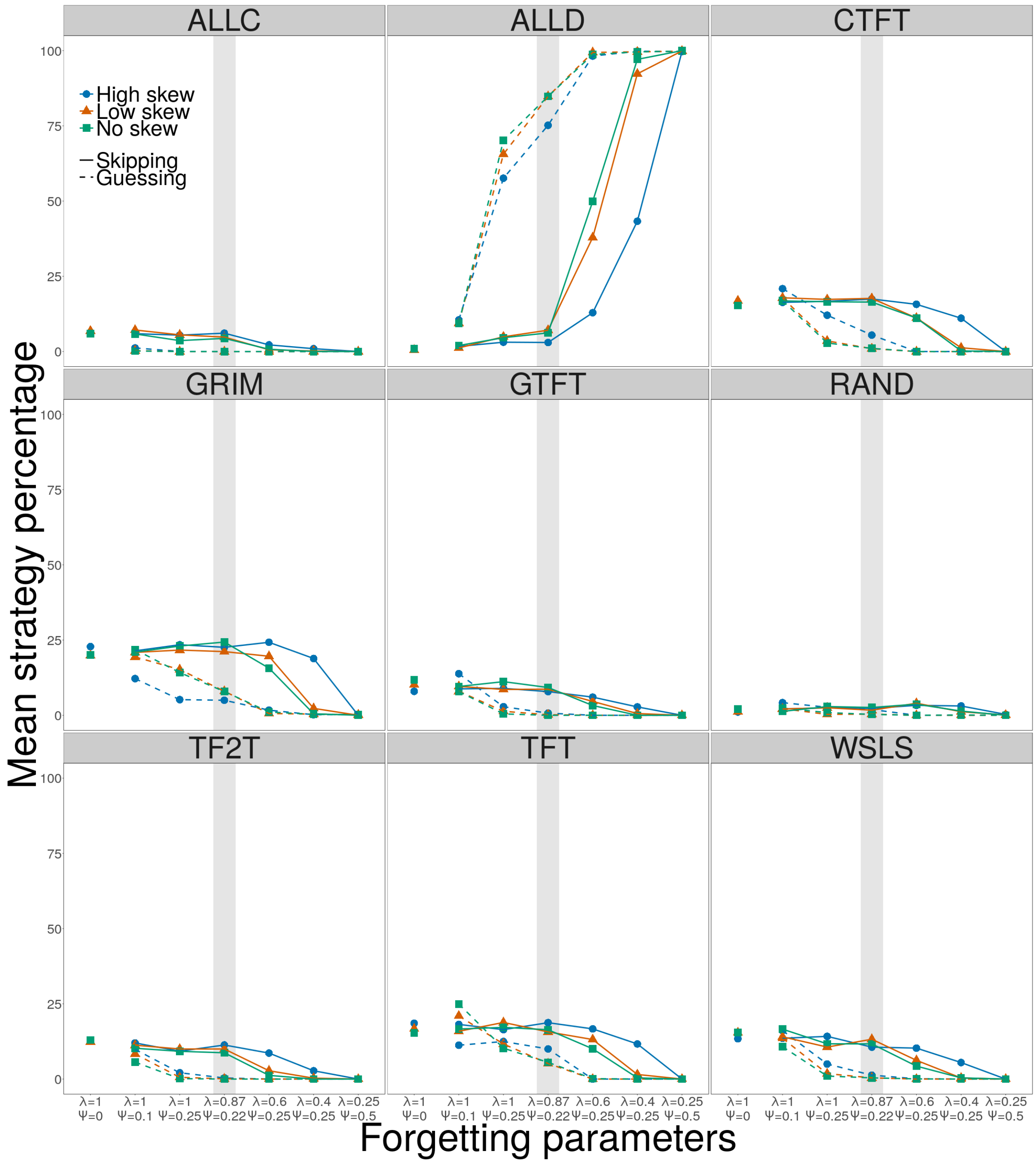


Figure S7: Mean percentage of each strategy in the population varied based on level of skew in contact pattern (no skew, low skew, high skew), forgetting rate (λ specifies starting point, ψ specifies decay rate), and type of memory error (guessing or skipping). Shaded area highlights the empirically derived forgetting parameters. Standard error of the mean error bars are not shown because they are smaller than the data point symbols.

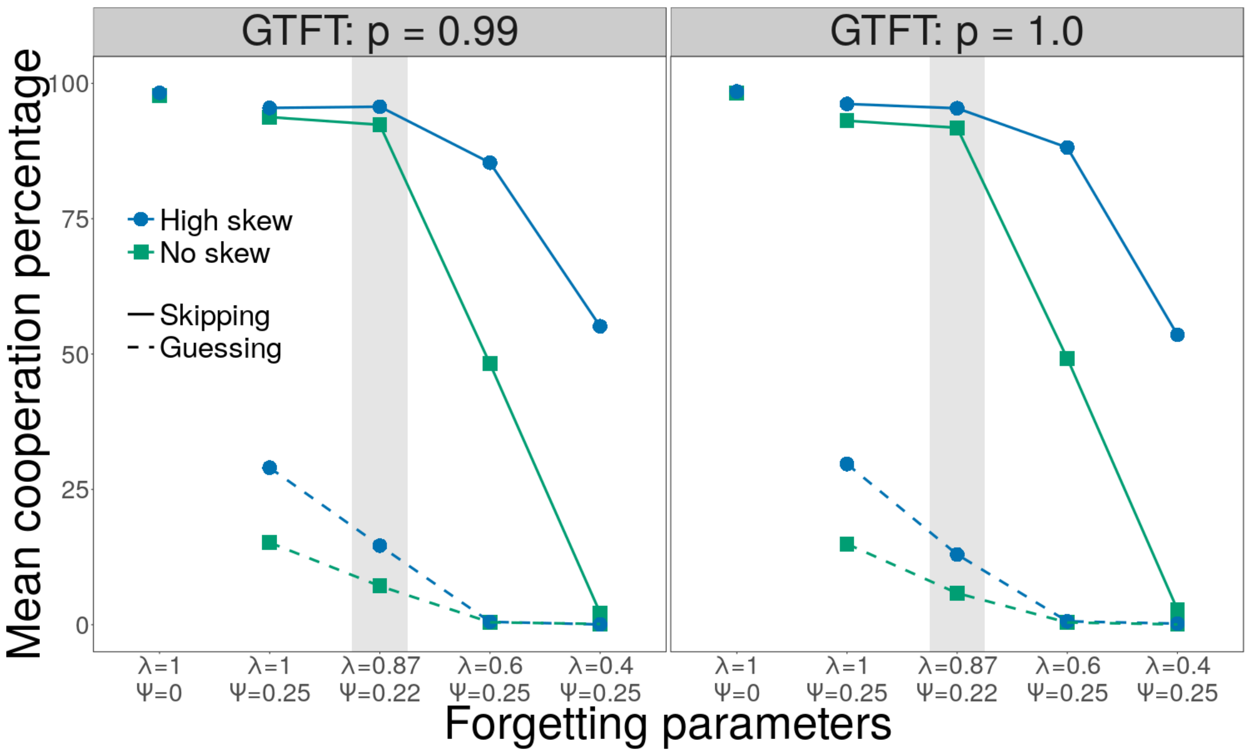


Figure S8: Effect of different GTFT versions on cooperation. GTFT has a probability p of cooperating following the partner cooperating. The primary analyses defined $p = 0.99$ based on Nowak and Sigmund's (1992) stochastic simulation illustrated in their Figure 1. We also conducted deterministic simulations defining $p = 1$ using a subset of the conditions (five forgetting parameters, two memory error types, and two skew conditions) for 250 generations and 1,000 replicates. These two values of p yielded the same cooperation rates.

References

1. R Core Team 2018 *R: A language and environment for statistical computing.* Vienna, Austria: R Foundation for Statistical Computing.

2. Fox, J. & Weisberg, S. 2011 *An R companion to applied regression.* Second edition. Thousand Oaks, CA: Sage.

3. Wickham, H. 2016 *ggplot2: Elegant graphics for data analysis.* 2nd edn. New York: Springer.

4. Müller, K. 2017 *Here: A simpler way to find your files.* R package version 0.1: https://CRAN.R-project.org/package=here.

5. Rapoport, A. & Chammah, A. N. 1965 *Prisoner's dilemma: A study in conflict and cooperation.* Ann Arbor, MI: University of Michigan Press.

6. Axelrod, R. 1980 More effective choice in the Prisoner's Dilemma. *Journal of Conflict Resolution* **24**, 379–403. (doi:10.1177/002200278002400301)

7. Nowak, M. A. & Sigmund, K. 1992 Tit-for-tat in heterogeneous populations. *Nature* **355**, 250–253. (doi:10.1038/355250a0)

8. Boyd, R. 1989 Mistakes allow evolutionary stability in the repeated prisoner's dilemma game. *Journal of Theoretical Biology* **136**, 47–56. (doi:10.1016/S0022-5193(89)80188-2)

9. Friedman, J. W. 1971 A non-cooperative equilibrium for supergames. *Review of Economic Studies* **38**, 1–12. (doi:10.2307/2296617)

10. Kraines, D. & Kraines, V. 1989 Pavlov and the prisoner's dilemma. *Theory and Decision* **26**, 47–79. (doi:10.1007/BF00134056)

11. Nowak, M. A. & Sigmund, K. 1993 A strategy of win-stay, lose-shift that outperforms tit-for-tat in the prisoner's dilemma game. *Nature* **364**, 56–58. (doi:10.1038/364056a0)

12. Pachur, T., Schooler, L. J. & Stevens, J. R. 2014 We'll meet again: Revealing distributional and temporal patterns of social contact. *PLoS ONE* **9**, e86081. (doi:10.1371/journal.pone.0086081)

13. Baker, J. E. 1987 Reducing bias and inefficiency in the selection algorithm. In *Proceedings of the second international conference on genetic algorithms and their application*, pp. 14–21. Hillsdale, NJ: L. Erlbaum Associates Inc.

14. Kimura, M. & Crow, J. F. 1978 Effect of overall phenotypic selection on genetic change at individual loci. *Proceedings of the National Academy of Sciences (USA)* **75**, 6168–6171.