Supplementary Information for: Complexity synchronization networks reveal distinct modes of coordination in an adaptive predator–prey model

Korosh Mahmoodi et al.

Supplementary Information

The following supplementary sections retain the detailed model description and implementation details so that the main paper can remain focused on the conceptual result.

S1. Detailed Predator–Prey model and simulation parameters

The simulations use one Prey agent and two Predator agents moving on a square domain of length $L = 2$ with periodic boundary conditions. The discrete time step is $\Delta t = 1$. The Prey and Predators update their positions using constant-speed motion on the torus, with the Predators moving at speed $v_S = 0.1$ and the Prey moving at speed $v_F = 0.2$.

The dimensionless sensing-radius ratio is defined as

$$R = \frac{r_S}{L},$$

and is swept over

$$R = 0.24 + 0.01(1 : 21),$$

so that the effective Predator sensing radius is $r_S = R \times L$. The Prey uses a comparable sensing radius and chooses an evasive heading by predicting the mean Predator position and applying a deceptive angular deflection of $\pm\pi/6$.

Each Predator carries three adaptive thresholds associated with information sharing, trust, and payoff sharing. These thresholds are denoted by

$$I_1,\ T_1,\ P_1,\ I_2,\ T_2,\ P_2.$$

At each time step, binary decisions are generated by comparing each threshold to a uniform random number. The thresholds are then updated using payoff-based reinforcement, with learning step

$$\Delta_S = 0.1$$

and a small stochastic term with amplitude

$$\text{Noise} = 10^{-3}.$$

When at least one Predator successfully captures the Prey and trust is active, the two Predators enter a payoff-sharing interaction governed by a Prisoner's Dilemma–like payoff matrix. Writing the row player as Predator 1 and the column player as Predator 2, the payoff matrix is

	Predator 2: Cooperate	Predator 2: Defect
Predator 1: Cooperate	(2, 2)	(−2, 5)
Predator 1: Defect	(5, −2)	(−1, −1)

(1)

Cooperation corresponds to both agents sharing payoff $(2, 2)$, and the cooperation rate C_r is the cumulative count of gated mutual-sharing events divided by the number of simulation trials. Unilateral defection yields the temptation–sucker outcomes $(5, -2)$ and $(-2, 5)$, and mutual defection yields the punishment pair $(-1, -1)$. In the simulations reported here, the temptation parameter is set by $tc = 3$.

For each parameter condition, the simulation runs for

$$T = 10^6$$

trials and

$$ENS = 10$$

ensembles. After discarding the first 25% of the threshold trajectories, local scaling exponents are computed on overlapping windows of length

$$\text{Slice} = 10^4$$

with 75% overlap. MDEA is evaluated for the three representative stripe sizes

$$\{0.1,\ 0.01,\ 0.001\}.$$

These stripe sizes were selected to span coarse, intermediate, and fine event resolutions and were used consistently throughout all MDEA analyses.

Unless otherwise stated, MDEA was implemented using Rule = 1, and local scaling exponents were obtained from the entropy-growth curve by fitting the entropy-growth curve. For the public C_r–CS MDEA script, the call is `MDEA(DataX, str, 1, 0.1, 0.9, 0)`: the fit uses indices from `round(0.1*K)` through `round(0.9*K)`, where K is the number of lag values generated by `MDEA.m`. These are fractions of the lag-grid indices, not fractions of physical time. The lag grid is $\lfloor 1.2^j \rfloor$ with $j = 1, \ldots, \lfloor \log(N)/\log(1.2) \rfloor - 5$ for window length N. The repository contains figure-specific scripts; their parameters should be consulted for exact reproduction of each figure.

In the implementation used for the $C_r - -CS$ figures, the reported pairwise CS quantities correspond to three representative threshold pairs extracted from the scaling-exponent correlation matrix. The cooperation rate C_r is computed as the running fraction of mutual payoff-sharing events per time.

S2. Explicit decision rules and update equations

This section provides the explicit dynamical rules underlying the simulations, including spatial geometry, directional updates, stochastic decision-making, payoff assignment, and threshold adaptation.

Periodic geometry and directional angles. All agents move on a square domain of side length L with periodic boundary conditions. For two positions

$$\mathbf{r}_a = (x_a, y_a), \qquad \mathbf{r}_b = (x_b, y_b),$$

the minimal-image displacement is

$$\Delta x_{ab} = x_b - x_a - L\,\mathrm{round}\left(\frac{x_b - x_a}{L}\right), \qquad \Delta y_{ab} = y_b - y_a - L\,\mathrm{round}\left(\frac{y_b - y_a}{L}\right).$$

The corresponding direction from a to b is

$$\Theta(\mathbf{r}_a \to \mathbf{r}_b) = \mathrm{atan2}(\Delta y_{ab}, \Delta x_{ab}),$$

and the torus-consistent distance is

$$d_{ab} = \sqrt{\Delta x_{ab}^2 + \Delta y_{ab}^2}.$$

State variables and motion. The system consists of one Prey agent F and two Predator agents S_1 and S_2. Their positions and headings evolve as

$$\mathbf{r}(t + \Delta t) = \mathbf{r}(t) + v \begin{bmatrix} \cos\theta(t) \\ \sin\theta(t) \end{bmatrix} \Delta t \pmod{L}.$$

Predators move with speed v_S and the Prey with speed $v_F = 2v_S$. $\Delta t = 1$.

Prey directional update and deception. The Prey senses Predators within its vision radius r_F. If at least one Predator is detected, it predicts their mean future position

$$\widehat{\mathbf{r}}_S(t + \Delta t) = \frac{1}{N} \sum_j \left(\mathbf{r}_{S_j}(t) + v_S \begin{bmatrix} \cos\theta_{S_j}(t) \\ \sin\theta_{S_j}(t) \end{bmatrix} \right),$$

where the sum runs over detected Predators.

The Prey computes the direction toward this estimate:

$$\phi_F(t) = \Theta(\mathbf{r}_F(t) \to \widehat{\mathbf{r}}_S(t + \Delta t)).$$

It then generates two symmetric evasive headings:

$$\theta_F^{(\pm)}(t) = \pi + \phi_F(t) \pm \Theta_D,$$

where $\Theta_D = \pi/6$.

The realized heading is chosen stochastically:

$$\theta_F(t) \in \{\theta_F^{(+)}, \theta_F^{(-)}\}, \quad P = 1/2.$$

This two-option rule introduces deception and stochasticity in the environment.

Predator directional prediction. If the Prey is within sensing radius r_S, each Predator S_i predicts two possible future Prey positions:

$$\widehat{\mathbf{r}}_F^{(\pm)}(t + \Delta t) = \mathbf{r}_F(t) + v_F \begin{bmatrix} \cos\theta_F^{(\pm)}(t) \\ \sin\theta_F^{(\pm)}(t) \end{bmatrix}.$$

These define two candidate pursuit directions:

$$\phi_{S_i}^{(\pm)}(t) = \Theta(\mathbf{r}_{S_i}(t) \to \widehat{\mathbf{r}}_F^{(\pm)}(t + \Delta t)).$$

Each Predator randomly selects one as its own estimate:

$$\theta_{S_i}^{\mathrm{self}}(t) \in \{\phi_{S_i}^{(+)}, \phi_{S_i}^{(-)}\},$$

while the other is stored as the complementary estimate:

$$\theta_{S_i}^{\mathrm{comp}}(t).$$

Probabilistic decision rules. Each Predator carries three thresholds:

$$I_i(t), \quad T_i(t), \quad P_i(t).$$

Each generates a binary decision via

$$s_i^{(X)}(t) = \begin{cases} 1, & u > X_i(t), \\ 0, & \text{otherwise}, \end{cases} \quad u \sim \mathcal{U}(0,1).$$

Here:

$$s_i^{(I)} = 1 \to \text{share information}, \quad s_i^{(T)} = 1 \to \text{trust}, \quad s_i^{(P)} = 1 \to \text{share payoff}.$$

Trust-mediated complementary coupling. Let $j \neq i$. The realized heading of Predator S_i is

$$\theta_{S_i}(t) = \begin{cases} \theta_{S_j}^{\text{comp}}(t), & s_i^{(T)}(t) = 1, \\ \theta_{S_i}^{\text{self}}(t), & s_i^{(T)}(t) = 0. \end{cases}$$

Thus, when trust is active, a Predator adopts the complementary estimate from its partner, allowing the pair to cover both possible Prey trajectories.

Environmental capture and stage-1 payoff. Capture is determined by proximity:

$$\chi_i(t) = \begin{cases} 1, & d(\mathbf{r}_{S_i}, \mathbf{r}_F) < r_G, \\ 0, & \text{otherwise}. \end{cases}$$

Let $c_i(t)$ be the successful-capture indicator. Conditional on $\chi_i(t) = 1$, $c_i(t)$ is Bernoulli with success probability p_{env}; otherwise $c_i(t) = 0$. Proximity and successful capture are therefore distinct.

The resulting environmental payoff is

$$\Pi_i^{(0)}(t) = \begin{cases} +2, & \text{success}, \\ -2, & \text{failure}. \end{cases}$$

Definition of cooperation rate C_r. At each time step t, a payoff-sharing interaction between the two Predators is evaluated only under specific conditions.

First, a successful capture event must occur, defined by

$$c_i(t) = 1 \quad \text{for at least one } i \in \{1, 2\},$$

indicating that at least one Predator captures the Prey.

Second, the interaction must be enabled by trust. In the implementation used here, this condition is satisfied if at least one of the two Predators realizes a trust decision,

$$s_1^{(T)}(t) = 1 \quad \text{or} \quad s_2^{(T)}(t) = 1.$$

When both conditions are satisfied, the system enters a payoff-sharing stage in which each Predator makes a binary decision regarding payoff sharing:

$$s_i^{(P)}(t) = \begin{cases} 1, & \text{share payoff}, \\ 0, & \text{do not share}. \end{cases}$$

Mutual cooperation is defined as the simultaneous realization

$$s_1^{(P)}(t) = s_2^{(P)}(t) = 1.$$

The cooperation rate C_r is then computed as the running fraction of such mutual-cooperation events over time,

$$\mathrm{C_r}(t) = \frac{1}{t}\sum_{\tau=1}^{t} \mathbb{I}\Big[\mathrm{capture}(\tau) \ \wedge \ \big(s_1^{(T)}(\tau) = 1 \ \vee \ s_2^{(T)}(\tau) = 1\big) \ \wedge \ \big(s_1^{(P)}(\tau) = 1 \ \wedge \ s_2^{(P)}(\tau) = 1\big)\Big],$$

where $\mathbb{I}[\cdot]$ denotes the indicator function. This definition reflects the fact that cooperation is evaluated only within interaction events that are both environmentally successful and socially enabled.

Stage-2 Prisoner's Dilemma interaction. A payoff-sharing interaction occurs if at least one Predator succeeds and at least one trusts.

Given decisions $s_1^{(P)}, s_2^{(P)}$, the payoff matrix is

$$(\Pi_1, \Pi_2) = \begin{cases} (2,2), & (1,1), \\ (2+t_c,-2), & (0,1), \\ (-2,2+t_c), & (1,0), \\ (-1,-1), & (0,0), \end{cases}$$

with $t_c = 3$.

Otherwise, $\Pi_i(t) = \Pi_i^{(0)}(t)$.

Threshold adaptation. Each threshold evolves via payoff-driven reinforcement:

$$X_i(t+1) = \mathcal{P}_{[0,1]}\left[X_i(t) + \sigma(s_i)\,\delta_X\,(\Delta\Pi_i + \eta)\right],$$

where

$$\sigma(s) = \begin{cases} -1, & s = 1, \\ +1, & s = 0, \end{cases} \qquad \eta \sim \mathcal{N}(0, \mathrm{Noise}^2).$$

$\mathcal{P}_{[0,1]}$ projects values into $[0,1]$.

Complexity-synchronization pipeline. The six threshold time series

$$I_1,\ T_1,\ P_1,\ I_2,\ T_2,\ P_2$$

are analyzed using sliding windows. Local scaling exponents $\delta(t)$ are computed via MDEA or DFA, and CS is defined as

$$CS = \mathrm{corr}(\delta_a(t), \delta_b(t)).$$

Statistical scope. CS correlations, fitted trends, and ensemble variability are descriptive. Nominal correlation p-values in the original scripts do not account for overlapping windows and serial dependence; averages of p-values are not valid ensemble significance tests. Filled/open markers and solid/dashed edges retain legacy plotting conventions only. The restoration panels compare three equal-budget interventions following a specified change to the payoff-sharing learning channels; the main Methods give the corresponding learning vectors.

S3. Restoration in the low-R regime

The main rescue analysis was performed at $R = 0.45$, where cooperation is high and MDEA-based CS provides the most resolved description of the event-driven coordination structure. To describe restoration in a second regime, we repeated the rescue analysis at $R = 0.25$. In this low-interaction regime, cooperation is weaker and MDEA-based CS is less pronounced. DFA-based CS provides a complementary descriptive view of scaling coordination in this regime.

The same perturbation and rescue logic was used as in the main text. Damage was imposed by reducing the learning increments of the payoff-sharing thresholds P_1 and P_2. Non-targeted local rescue applied the intervention budget to a non-targeted pair, equal-budget global rescue distributed the budget across thresholds, and targeted rescue restored the payoff-sharing predefined damaged subsystem. All simulations used 10^6 trials and ENS = 10 independent stochastic ensembles.

Supplementary Fig. S1 shows that DFA-based CS is weak and heterogeneous in the damaged condition and in the non-targeted local rescue condition, whereas the targeted rescue condition recovers a network structure closer to the baseline low-R state. Supplementary Fig. S2 shows the corresponding cooperation recovery curves. Targeted rescue increases C_r toward the baseline level as rescue strength increases, while non-targeted local rescue remains near the damaged level and equal-budget global rescue produces only a small improvement. These results extend the restoration comparison to a second interaction regime and illustrate how DFA-based CS networks accompany changes in cooperative performance.

S4. Modified Diffusion Entropy Analysis (MDEA)

Diffusion construction from event sequences To quantify temporal complexity, we employ Modified Diffusion Entropy Analysis (MDEA), which converts each time series into a diffusion process driven by discrete events. Events are extracted using a stripe-based method, whereby crossings of predefined amplitude levels generate a sequence of event times. These events are then mapped onto a diffusion trajectory by assigning unit increments at each event occurrence, resulting in an effective random walk.

Scaling hypothesis The diffusion process is assumed to satisfy the scaling form:

$$F(x,t) = \frac{1}{t^{\delta}}\,\Phi\!\left(\frac{x}{t^{\delta}}\right), \tag{2}$$

where $F(x,t)$ denotes the probability density function (PDF) of the diffusion variable x at time t, δ is the scaling exponent, and $\Phi(\cdot)$ is a time-independent scaling function. This form implies statistical self-similarity, meaning that the shape of the distribution remains invariant under appropriate rescaling. Consequently, the characteristic spread of the process follows $x \sim t^{\delta}$.

Entropy scaling The scaling exponent is obtained through the Shannon entropy of the diffusion process:

$$S(t) = -\int F(x,t)\,\log F(x,t)\,dx. \tag{3}$$

Substituting the scaling form of $F(x,t)$ leads to:

$$S(t) = A + \delta \log t, \tag{4}$$

where A is a constant. The exponent δ is estimated as the slope of $S(t)$ versus $\log t$. Entropy scaling can characterize non-Gaussian diffusion, subject to the scaling hypothesis and finite-window estimation limits.

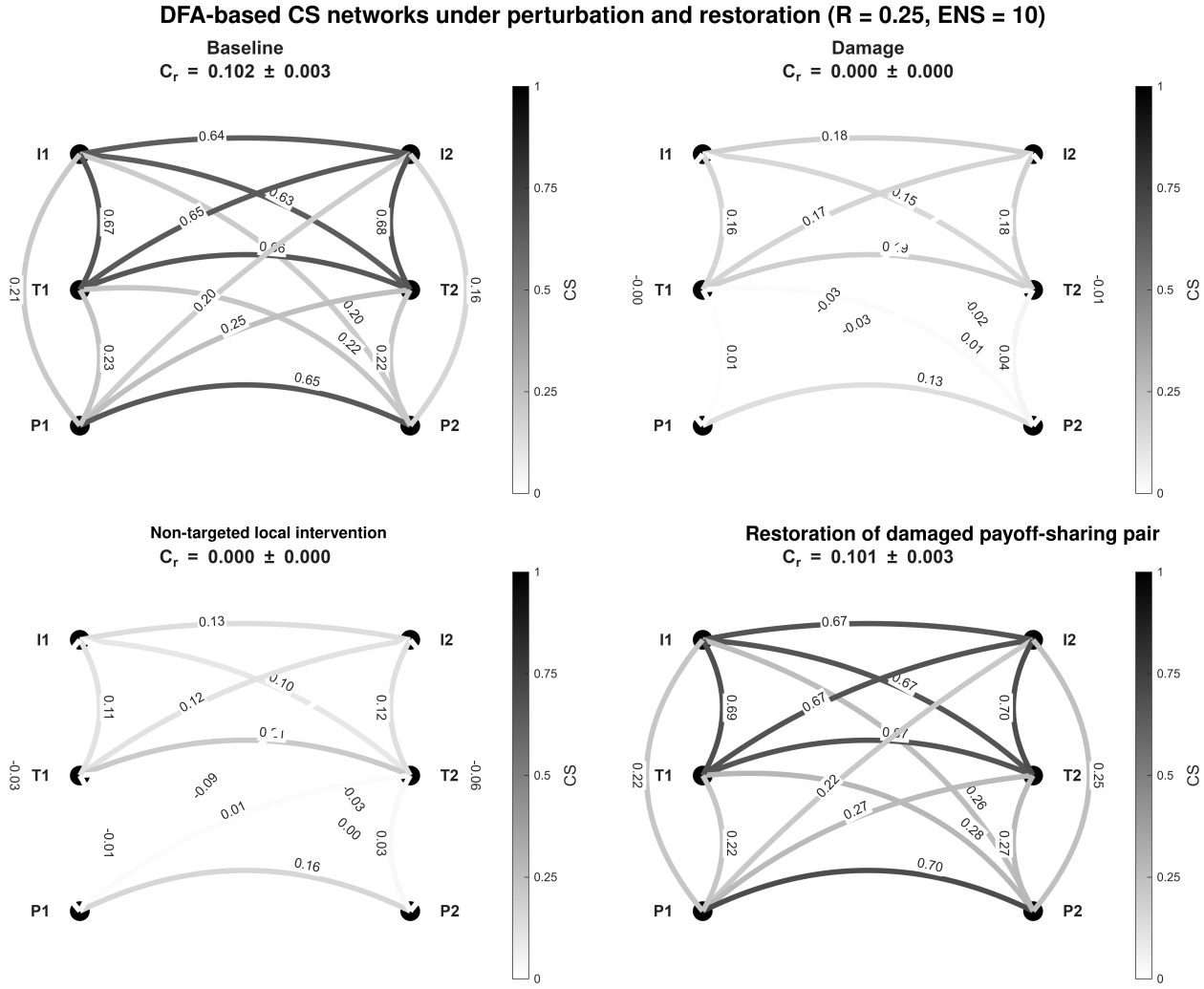

Figure S1: DFA-based complexity-synchronization (CS) networks under baseline, perturbation, and rescue conditions ($R = 0.25$, ENS $= 10$). Graph representations of pairwise CS among the six adaptive thresholds $(I_1, T_1, P_1, I_2, T_2, P_2)$, computed using detrended fluctuation analysis (DFA). Nodes correspond to threshold variables, and edges represent CS links. Edge thickness and grayscale intensity encode CS strength, dashed and solid edges retain the original nominal-p-value plotting convention and do not establish statistical significance, and edge labels report the corresponding CS values. Panels show (top left) baseline, (top right) perturbed (damage), (bottom left) non-targeted local rescue, and (bottom right) targeted rescue. The cooperation rate, C_r (mean ± SD across ensembles), is reported in each panel. The color scale spans the interval $[0, 1]$.

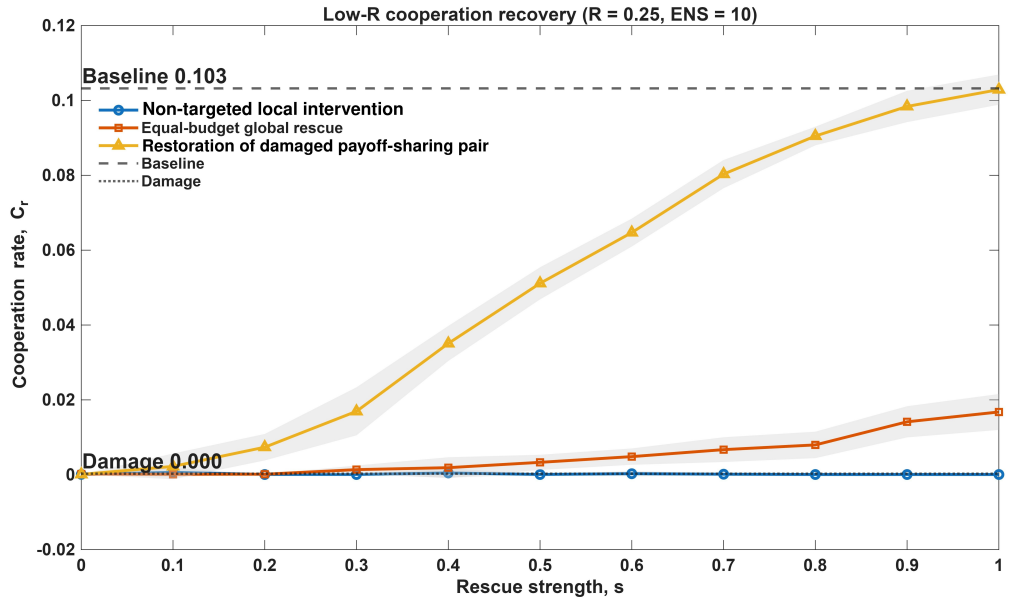

Figure S2: Cooperation recovery in the low-interaction regime ($R = 0.25$, ENS $= 10$) using DFA-based complexity-synchronization (CS). The cooperation rate, C_r, is shown as a function of rescue strength, s, for three intervention strategies: non-targeted local rescue, equal-budget global rescue, and targeted intervention. Curves show the mean across ensembles, and shaded bands indicate $\pm$ SD. Horizontal dashed lines denote the baseline and perturbed (damage) levels.

Relation to renewal dynamics For renewal processes characterized by inverse power-law waiting-time distributions,

$$\psi(\tau) \sim \tau^{-\mu}, \quad 2 < \mu < 3, \tag{5}$$

the scaling exponent is related to μ by:

$$\delta = \frac{1}{\mu - 1}. \tag{6}$$

This relationship links the diffusion scaling directly to the temporal statistics of events, enabling identification of long-range temporal organization.

Application in this study MDEA is applied within sliding windows to obtain a time-resolved scaling exponent $\delta(t)$ for each signal. These scaling time series are interpreted as measures of local window-based scaling. CS is then quantified as the statistical association between the scaling exponents of interacting components.

S5. Detrended Fluctuation Analysis (DFA)

Profile construction Detrended Fluctuation Analysis (DFA) is used as an alternative method to quantify scaling behavior. Given a time series $x(i)$ of length N, the first step is to construct the integrated profile:

$$Y(k) = \sum_{i=1}^{k} \left(x(i) - \langle x \rangle \right), \tag{7}$$

where $\langle x \rangle$ is the mean of the signal.

Segmentation and detrending The profile $Y(k)$ is divided into non-overlapping segments of equal length n. Within each segment, a polynomial trend (typically linear in this study) is fitted and removed. The root-mean-square fluctuation is then computed as:

$$F(n) = \sqrt{\frac{1}{N}\sum_{k=1}^{N}[Y(k) - Y_n(k)]^2}, \tag{8}$$

where $Y_n(k)$ denotes the local trend in each segment.

Scaling relationship If the signal exhibits scale-invariant behavior, the fluctuation function follows:

$$F(n) \sim n^H, \tag{9}$$

where H is the DFA scaling exponent. The exponent is obtained as the slope of $\log F(n)$ versus $\log n$.

Interpretation For stationary signals in the usual DFA interpretation, the exponent H characterizes fluctuation correlations:

- $H = 0.5$: uncorrelated (white noise)
- $0.5 < H < 1$: persistent correlations
- $0 < H < 0.5$: anti-persistent behavior

Exponents outside this stationary range require interpretation in terms of nonstationarity and preprocessing; they are not automatically Hurst exponents. Here H denotes the fitted DFA exponent.

Application in this study DFA is applied within sliding windows to obtain a time-resolved scaling exponent $H(t)$ for each signal. Similar to MDEA, these scaling time series are used to quantify complexity. CS is computed as the statistical association between the DFA scaling exponents of interacting components.

Complementarity with MDEA While DFA captures correlation-based scaling properties of the signal amplitude, MDEA characterizes the scaling behavior of event-driven diffusion processes. The two approaches therefore probe distinct aspects of temporal organization, and their combined use provides a complementary description of system dynamics.

DFA measures second-order fluctuation scaling, whereas MDEA estimates entropy scaling for an event-derived diffusion process. Under non-Gaussian or anomalous dynamics, the exponents may differ and their interpretation depends on the process and diffusion construction [1]. Neither exponent is an estimator-independent measure of all temporal structure. Relations derived for velocity-based diffusion models are not assumed to apply unchanged to the unit-event construction used here.

S6. Ordinary Pearson-correlation control analysis

To compare complexity synchronization with conventional linear similarity, ordinary Pearson correlations were computed directly from the raw adaptive threshold signals after the initial burn-in period. Pairwise Pearson correlations were evaluated for all 15 threshold pairs among the six adaptive variables (I_1, I_2, T_1, T_2, P_1, and P_2), and the ensemble-averaged pair-specific correlations were examined as functions of the sensing-radius parameter R. This analysis asks whether direct amplitude-level correlations among the adaptive thresholds reproduce the coordination structure revealed by CS. The results are summarized in Supplementary Fig. S3.

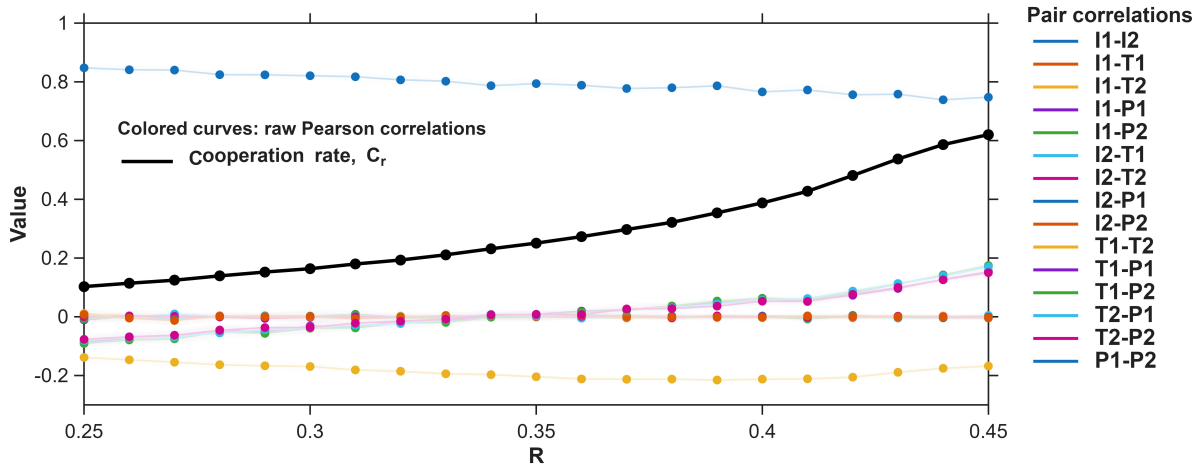

Figure S3: **Ordinary Pearson correlations as a function of sensory radius.** The ensemble-averaged cooperation rate (C_r, black) and ordinary Pearson correlations (colored curves) are shown as functions of the sensory-radius parameter R. Colored curves represent the ordinary Pearson correlation computed for each of the 15 unique pairs among the six adaptive variables (I_1, I_2, T_1, T_2, P_1, and P_2), and the right-hand legend identifies the corresponding variable pair for each curve. The figure shows that raw correlations are highly pair-specific: the payoff-sharing pair P_1–P_2 exhibits the strongest direct correlation, whereas many other threshold pairs remain weakly correlated or near zero across the R range. Simulations were performed for 10^6 time steps, the first 25% of each trajectory was discarded before analysis, and results are shown for $ENS = 10$ ensemble realizations.

S7. Renewal test / aging experiment and inter-event autocorrelation

To assess whether the events extracted from an adaptive threshold are consistent with renewal-like dynamics, we used the stripe-crossing event sequence underlying MDEA. The renewal-support analyses reported in Supplementary Figs. S4 and S5 were performed on the information-sharing threshold of Predator 1, I_1, as a representative adaptive threshold. These analyses were not averaged over all six thresholds and were not computed directly from the representative CS pairs; they were used to provide mechanistic support for the event-driven interpretation of MDEA-based CS.

For each analyzed window, I_1 was first shifted and normalized to the interval $[0, 1]$. The normalized trajectory was discretized into stripes of size 0.1, 0.01, or 0.001. An event was registered whenever the signal crossed from one stripe to another. The inter-event interval sequence

$$\{\tau_i\}$$

was then defined as the number of samples between successive stripe-crossing events.

For the renewal aging analysis in Supplementary Fig. S4, the first half of the simulation was discarded, and one window of length 10^4 samples was selected from the remaining I_1 trajectory. The aging time was fixed at

$$t_a = 100.$$

The ordinary survival function $\Psi(\tau)$ was computed from the inter-event intervals. The aged survival function was then obtained by applying the aging procedure with $t_a = 100$ to the same event sequence. A shuffled-aged control was generated by randomly permuting the inter-event intervals before applying the same aging procedure. This shuffled control preserves the one-point distribution of waiting times while removing serial dependence between consecutive intervals.

The key idea of the renewal test is that, for a genuine renewal process, consecutive waiting times are statistically independent. Therefore, if the aged survival function from the original sequence and the shuffled-aged survival function are similar, the temporal ordering of the intervals carries little additional memory, which is consistent with renewal-like dynamics. Conversely, a discrepancy between the original aged and shuffled-aged curves indicates memory across successive events. The dashed black line in Supplementary Fig. S4 is a power-law fit to the survival function over the displayed fitting range, and the reported value $\mu - 1$ summarizes the estimated survival-function scaling.

For the inter-event autocorrelation analysis in Supplementary Fig. S5, the first half of the simulation was again discarded. Ten windows of length 10^4 samples were then selected from the remaining I_1 trajectory using 75% overlap. For each window and stripe size, the inter-event interval sequence $\{\tau_i\}$ was extracted and its normalized autocorrelation was computed up to lag 100. The plotted autocorrelation curve is the mean across the 10 windows, and the shaded region represents the standard deviation across windows.

Because the autocorrelation can oscillate around zero, the memory-decay summary was obtained from the upper envelope of the absolute autocorrelation. Local peaks of $|ACF(\ell)|$ were identified as a function of lag ℓ, and a power-law fit was performed on log–log axes. The red dashed line in Supplementary Fig. S5 shows this envelope fit. The reported envelope slope quantifies the decay of inter-event memory: more negative slopes correspond to faster decay, whereas slopes closer to zero indicate more persistent correlations across event intervals.

In this way, the aging experiment and the inter-event autocorrelation provide complementary tests. The aging comparison evaluates whether temporal ordering of waiting times matters, while the autocorrelation directly quantifies memory across successive inter-event intervals. Both analyses are consistent with the interpretation that the high-cooperation regime is closer to renewal-like event organization than the low-cooperation regime.

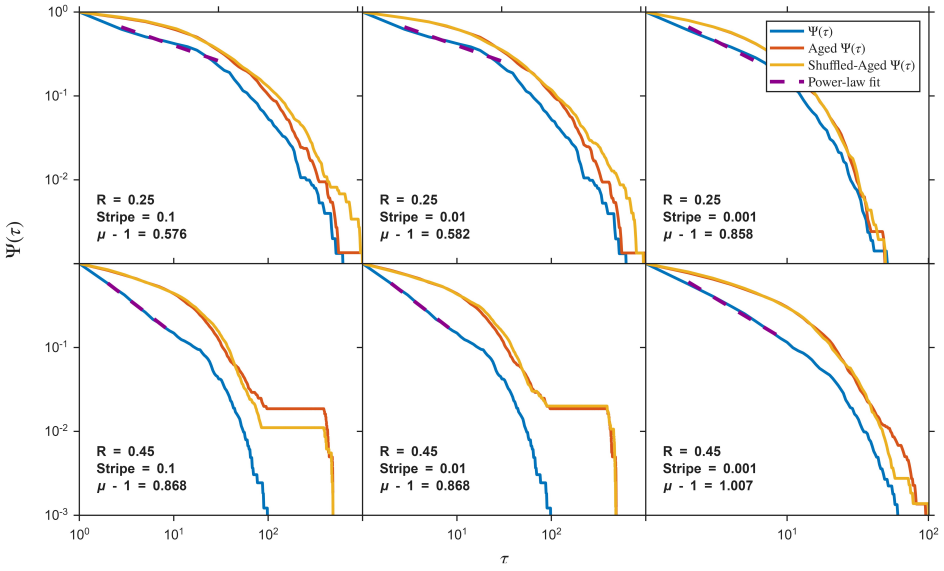

Figure S4: Renewal aging test for the representative threshold I_1. Survival functions $\Psi(\tau)$, aged survival functions, and shuffled-aged controls are shown for two sensing-radius regimes ($R = 0.25$ and $R = 0.45$) and three stripe sizes. The first half of the simulation was discarded, and one post-discard window of length 10^4 samples was analyzed. Events were defined as crossings from one stripe to another, and τ denotes the inter-event interval between successive stripe-crossing events. The aging time was fixed at $t_a = 100$. The dashed black line shows the power-law fit used to estimate $\mu - 1$.

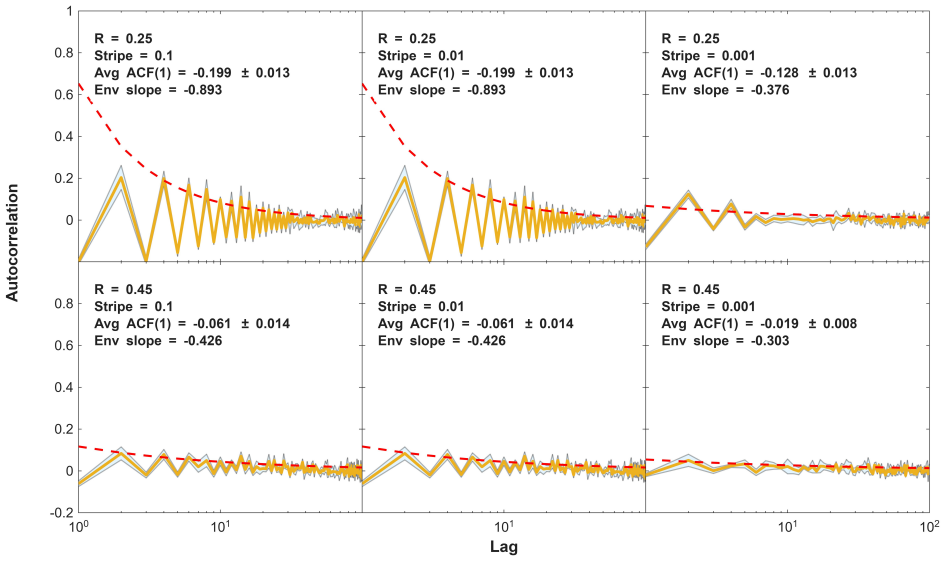

Figure S5: Autocorrelation of inter-event intervals extracted from the representative threshold I_1. The first half of the simulation was discarded, and τ sequences were extracted from 10 windows of length 10^4 samples with 75% overlap for each sensing-radius and stripe-size condition. Autocorrelation was computed on the inter-event intervals τ, not on the raw threshold signal. Curves show the mean across windows, and shaded bands indicate ± SD. The red dashed line shows a power-law fit to the upper envelope of the absolute autocorrelation, and the reported envelope slope summarizes the decay of inter-event memory with lag.

S8. Ensemble-level statistical validation of cooperation–CS slopes

The new analysis tests whether the sign of the association between cooperation and CS is reproducible across stochastic realizations. For each of ten ensembles, the cooperation rate was regressed against CS over the 21 sensing-radius conditions. Three threshold pairs were examined with both MDEA and DFA. The MDEA analysis in this validation used the intermediate stripe width of 0.01. Each ensemble supplied one slope for each estimator–pair combination. Thus the resampling unit is the simulation ensemble, not the 75%-overlapping scaling window.

We calculated percentile 95% confidence intervals for the mean ensemble slope from 5,000 bootstrap resamples of the ten ensembles. A two-sided Monte Carlo sign-flip test used 5,000 random sign assignments to evaluate the magnitude of the observed mean slope; its finite-simulation p-value used a plus-one correction. Holm adjustment was applied across all six tests. These results are conditional on the simulated sensing-radius sweep and assume that sign-flipping ensemble slopes is appropriate under the null. The tests address ensemble-level cooperation–CS slopes; individual network edges are visualized as descriptive structural features.

Table 1: Ensemble-level cooperation–CS slope validation (ten stochastic ensembles). Confidence intervals are percentile ensemble-bootstrap intervals. All adjusted p-values are from two-sided sign-flip tests with Holm correction across six comparisons.

Estimator	Pair	Mean slope	95% CI lower	95% CI upper	Holm p
MDEA	I_1–I_2	0.633	0.609	0.658	0.0084
MDEA	I_2–T_1	0.654	0.638	0.669	0.0084
MDEA	I_1–P_2	0.666	0.648	0.685	0.0084
DFA	I_1–I_2	-0.698	-0.738	-0.659	0.0084
DFA	I_2–T_1	-0.684	-0.720	-0.649	0.0084
DFA	I_1–P_2	-1.250	-1.354	-1.145	0.0084

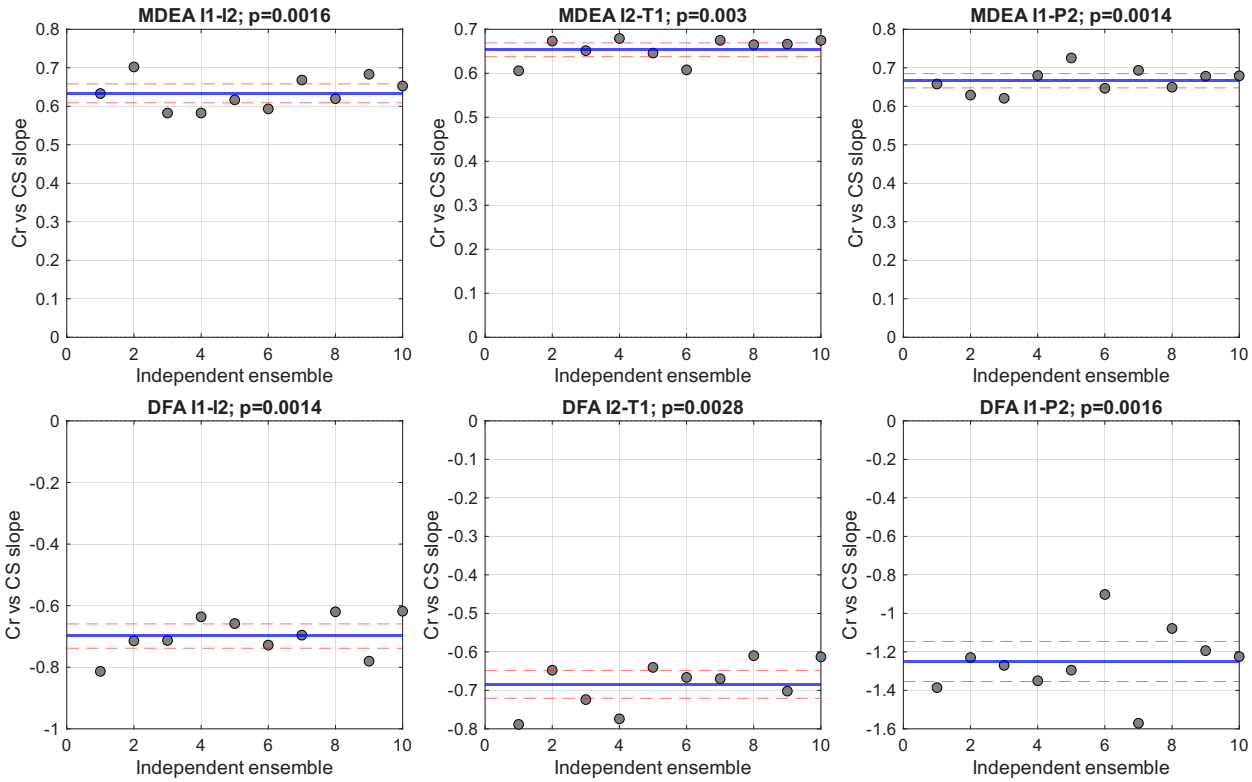

Figure S6: **Ensemble-level slopes for the association between cooperation rate and CS.** Each point represents the fitted slope across sensing-radius settings from one stochastic ensemble. Horizontal reference lines mark zero, the mean slope, and its 95% ensemble-bootstrap confidence limits. The six panels cover three threshold pairs for MDEA (intermediate stripe size 0.01) and DFA. Panel titles show unadjusted Monte Carlo sign-flip p-values; Holm-adjusted values for all six tests are 0.0084 (Table 1). These tests concern ensemble-level slopes and do not establish significance of individual network edges.

References

[1] Armen K. Kalashyan, Marco Buiatti, and Paolo Grigolini. Ergodicity breakdown and scaling from single sequences. *Chaos, Solitons & Fractals*, 39(2):895–909, 2009.