Exact computation of Transfer Entropy with Path Weight Sampling

Avishek Das[1] and Pieter Rein ten Wolde[1]
[1] *AMOLF, Science Park 104, 1098 XG, Amsterdam, The Netherlands*
(Dated: September 2, 2024)

Information processing in networks entails a dynamical transfer of information between stochastic variables. Transfer entropy is widely used for quantification of the directional transfer of information between input and output trajectories. However, currently there is no exact technique to quantify transfer entropy given the dynamical model of a general network. Here we introduce an exact computational algorithm, Transfer Entropy-Path Weight Sampling (TE-PWS), to quantify transfer entropy and its variants in an arbitrary network in the presence of multiple hidden variables, nonlinearity, transient conditions, and feedback. TE-PWS extends a recently introduced algorithm Path Weight Sampling (PWS) and uses techniques from the statistical physics of polymers and trajectory sampling. We apply TE-PWS to linear and nonlinear systems to reveal how transfer entropy can overcome naive applications of data processing inequalities in presence of feedback.

Information transfer between noisy signals underlies the functionality of diverse real-world networks such as in biochemical signaling, neuroscience, ecology, wireless communication and finance. Information theory has so far provided a useful framework to quantify the feasibility of retrieval of a noisy input signal from an output signal. In the presence of feedback loops in a network, information travels in both directions between the input and the output. An information-theoretic quantity that can measure the information transfer separately in either direction is the transfer entropy.[1,2] Transfer entropy and its variants, such as directed information, conditional transfer entropy or filtered transfer entropy, have been widely used to gain knowledge about the connectivity of a network,[3] infer causal relations from experiments,[4,5] establish fundamental bounds on network performance,[6] and estimate the minimal physical work required for a computation.[7,8] Hence, for a wide range of problems, it is vital to be able to accurately quantify transfer entropies.

However, there are currently no exact methods to compute transfer entropy in a general many-variable network model beyond linear or low-order moment closure approximations.[9,10] Systematic improvement of analytical approximations to account for high-order correlations in the trajectory ensemble is prohibitively complicated. Additionally, direct numerical estimation from experimental or simulated trajectories is intractable because the state space of long trajectories diverges exponentially with trajectory duration. Histograms in trajectory space are thus expensive to compute and are undersampled, giving rise to large and sometimes uncontrolled errors.[11,12]

Here we fill this gap by introducing TE-PWS, a numerical algorithm to estimate transfer entropies exactly in any stochastic model, including in diffusive and jump processes. The estimate is exact, *i.e.*, it is an unbiased statistical estimate of the transfer entropy. Thus it can provide *ground truth* results for any given model. TE-PWS builds on the recently developed PWS algorithm for computation of mutual information between trajectories.[13] TE-PWS exploits the idea that path likelihoods can be obtained analytically from the Langevin or master equation, from which transfer entropy is then computed via Monte-Carlo averaging in trajectory space. Additionally, long trajectories are sampled with an importance sampling scheme, solving the problem of exponential computational cost. We have applied TE-PWS to compute the transfer entropy in a three-variable motif in the presence of feedback, for both linear and nonlinear dynamics. We show that TE-PWS reproduces analytical results when available and produces novel insights on how information feedback can amplify information transfer. Specifically, the transfer entropy from an input to an output node can overcome a naive application of data processing inequalities even when the mutual information obeys one.

Diffusive process. We introduce transfer entropy and TE-PWS first in the context of diffusive processes. Consider a d-dimensional diffusive process $\mathbf{X}(t)$ modelled as a function of time t by a Langevin equation

$$\dot{\mathbf{X}}(t) = \mathbf{F}(t) + \boldsymbol{\xi}(t), \tag{1}$$

with $\mathbf{F}(t)$ a general drift, and $\boldsymbol{\xi}(t)$ a three-dimensional Gaussian white noise with a diffusion constant matrix $\mathbf{D} = [D_{ij}]$ such that $\langle \xi_i(t)\xi_j(t^{'})\rangle = 2D_{ij}\delta(t-t^{'})$. The drift may depend on the entire past history as well as on time. The transfer entropy from X_i to X_j over N timesteps of durations δt each is defined as[1]

$$\mathcal{T}_{X_i\to X_j} = \sum_{k=0}^{N-1} I\left(X_j(k+1); X_{i,[0,k]} | X_{j,[0,k]}\right) \tag{2}$$

where the index k goes over individual timesteps, $X_j(k)$ denotes X_j after k timesteps, $X_{j,[0,k]}$ denotes the trajectory over the first k timesteps, and $I(A;B)$ denotes the mutual information between two random variables A and B. $\mathcal{T}_{X_i\to X_j}$ measures the information transferred from the past trajectory of X_i to the new updates of X_j at every timestep, given the past trajectory of X_j is already known. In case the dynamics relaxes into a

steady-state, we will also talk about the transfer entropy rate, $\dot{\mathcal{T}}_{X_i\to X_j} = \lim_{N\to\infty} \mathcal{T}_{X_i\to X_i}/(N\delta t)$.

We can rewrite the transfer entropy equivalently as

$$\mathcal{T}_{X_i\to X_j} = \sum_k \Delta\mathcal{T}^{i\to j}_{[k,k+1]} = \sum_k H\left(X_j(k+1)|X_{j,[0,k]}\right) - H\left(X_j(k+1)|X_{i,[0,k]}, X_{j,[0,k]}\right) \quad (3)$$

$$= \sum_k \left\langle \ln \frac{P\left(X_j(k+1)|X_{i,[0,k]}, X_{j,[0,k]}\right)}{P\left(X_j(k+1)|X_{j,[0,k]}\right)} \right\rangle \quad (4)$$

where $\Delta\mathcal{T}^{i\to j}_{[k,k+1]}$ denotes the increment at every timestep, $H(A)$ denotes the Shannon entropy associated with the probability distribution $P(A)$ of A, and the angular brackets denote an average over the joint probability $P\left(X_{i,[0,N]}, X_{j,[0,N]}\right)$. Eq. 3 shows that transfer entropy quantifies the additional information in $X_j(k+1)$ that arrives from $X_{i,[0,k]}$ beyond that which is already present in the past trajectory $X_{j,[0,k]}$. This occurs either through direct causal action, or through a third variable X_l, schematically demonstrated in Fig. 1a. If X_i does not affect the dynamics of X_j, this additional information would be zero. In general, every $\Delta\mathcal{T}^{i\to j}_{[k,k+1]}$ is a mutual information and hence nonnegative.

For calculating transfer entropy using Eq. 4 we develop TE-PWS by extension of the PWS algorithm.[13] The central idea in TE-PWS is that trajectory likelihoods in the full d-dimensional space are analytically available on-the-fly, and that trajectory averages can be computed in a Monte-Carlo fashion. First, the average in Eq. 4 is computed as,

$$\mathcal{T}_{X_i\to X_j} = \frac{1}{M_1}\sum_\nu \sum_k \ln \frac{P\left(X^{(\nu)}_j(k+1)|X^{(\nu)}_{i,[0,k]}, X^{(\nu)}_{j,[0,k]}\right)}{P\left(X^{(\nu)}_j(k+1)|X^{(\nu)}_{j,[0,k]}\right)} \quad (5)$$

where the index ν sums over M_1 pairs of trajectories of X_i and X_j sampled from the joint probability distribution $P\left(X^{(\nu)}_{i,[0,N]}, X^{(\nu)}_{j,[0,N]}\right)$. For each pair of trajectories, the probabilties in the numerator and the denominator of Eq. 5 are not analytically available, but what is indeed available is the full joint probability $P(\mathbf{X}^{(\nu)}_{[0,N]})$ as the exponential of the Onsager-Machlup action.[14,15] We thus need to marginalize over all degrees of freedom other than X_i and X_j, denoted henceforth collectively as X_l. We illustrate this procedure first for the denominator in Eq. 5. It is obtained as $P(X^{(\nu)}_{j,[0,k+1]})/P(X^{(\nu)}_{j,[0,k]})$, where $P(X^{(\nu)}_{j,[0,k]})$ is obtained via marginalization,

$$P\left(X^{(\nu)}_{j,[0,k]}\right) = \int\int D[X_{i,[0,k]}]D[X_{l,[0,k]}]P\left(X_{i,[0,k]}, X^{(\nu)}_{j,[0,k]}, X_{l,[0,k]}\right) \quad (6)$$

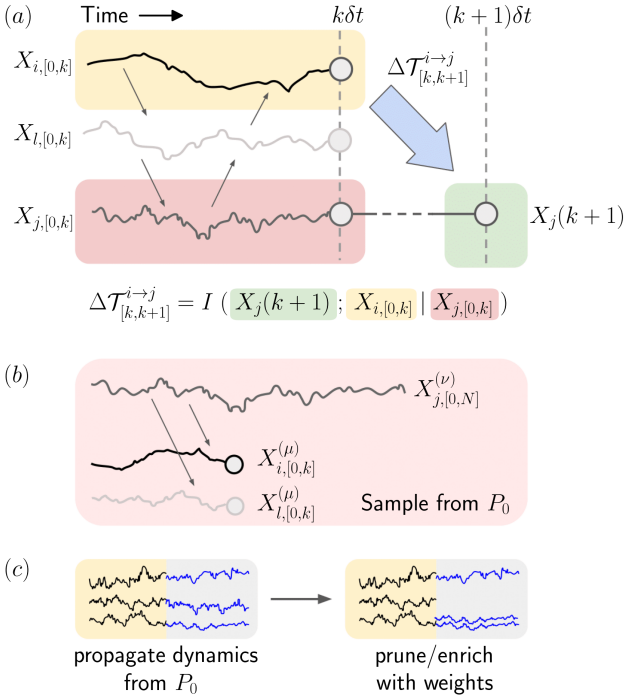

FIG. 1. (a) Schematic representation of the increase in transfer entropy from variable X_i to X_j at the $(k+1)$-th timestep. Other variables X_l may mediate information transfer even in the absence of a direct coupling from X_i to X_j. (b) Propagation of reference dynamics for X_i and X_l such that it is commensurate with the given frozen $X^{(\nu)}_j$ trajectory. (c) In the RR scheme, trajectories are propagated from a reference distribution $P_0(X_{i,[0,k]}, X_{l,[0,k]})$ and pruned and enriched periodically to turn them into the conditional distribution $P(X_{i,[0,k]}, X_{l,[0,k]}|X^{(\nu)}_{j,[0,k]})$.

For performing this average in a Monte-Carlo fashion,[16] we sample from a reference distribution, $P_0(X_{i,[0,k]}, X_{l,[0,k]})$, and correct the resultant bias by dividing by P_0,

$$P\left(X^{(\nu)}_{j,[0,k]}\right) = \frac{1}{M_2}\sum_\mu \frac{P\left(X^{(\mu)}_{i,[0,k]}, X^{(\nu)}_{j,[0,k]}, X^{(\mu)}_{l,[0,k]}\right)}{P_0\left(X^{(\mu)}_{i,[0,k]}, X^{(\mu)}_{l,[0,k]}\right)} \quad (7)$$

where the index μ sums over M_2 trajectories sampled from P_0. What is the best choice for P_0? The ideal choice would be the conditional distribution $P(X_{i,[0,k]}, X_{l,[0,k]}|X_{j,[0,k]})$, as it makes the summand in Eq. 7 equal for all μ, such that the variance of the estimate of $P(X^{(\nu)}_{j,[0,k]})$ is zero. However, this conditional distribution is not known *a priori*. We therefore generate X_i and X_l trajectories in the frozen field of $X^{(\nu)}_{j,[0,N]}$ (Fig. 1b) resulting in a distribution P_0 that is known analytically and is expected to be close to this conditional distribution. To exactly compensate for the remaining deviations of P_0 from the conditional distribution, we employ, in the spirit of Rosenbluth-Rosenbluth(RR)-PWS,[13] a

reweighing of the $X_i^{(\mu)}$ and $X_l^{(\mu)}$ trajectories on-the-fly with weights proportional to the ratio of the two distributions. We prune and enrich the trajectories with these weights after every δt time. This gives us access to the conditional distribution exactly (see Appendix).

Returning now to the numerator in Eq. 5, it is an average over another conditional distribution,

$$P\left(X_j^{(\nu)}(k+1)\middle|X_{i,[0,k]}^{(\nu)}, X_{j,[0,k]}^{(\nu)}\right) = \int D[X_{l,[0,k]}]\; P\left(X_j^{(\nu)}(k+1)\middle|X_{i,[0,k]}^{(\nu)}, X_{j,[0,k]}^{(\nu)}, X_{l,[0,k]}\right) . P\left(X_{l,[0,k]}\middle|X_{i,[0,k]}^{(\nu)}, X_{j,[0,k]}^{(\nu)}\right) \quad (8)$$

The first probability in the integral, which is the transition probability of X_j in the full d-dimensional space, is analytically available. Additionally, similar to the procedure for the denominator in Eq. 5, $P(X_{l,[0,k]}|X_{i,[0,k]}^{(\nu)}, X_{j,[0,k]}^{(\nu)})$ can also be accessed by sampling X_l trajectories from a $P_0(X_{l,[0,k]})$, and applying the RR scheme. Thus the numerator in Eq. 5 can also be evaluated in a Monte-Carlo fashion (see Appendix).

To summarize, for each of the M_1 pairs of $X_{i,[0,N]}$ and $X_{j,[0,N]}$ trajectories, we simulate a joint ensemble of M_2 new X_i and X_l trajectories to estimate the denominator in the logarithm in Eq. 5, and a separate ensemble of M_2 new X_l trajectories to estimate the numerator (Fig. 1). The trajectories in each ensemble are pruned and enriched on-the-fly with the RR scheme, giving both the numerator and denominator in Eq. 5 as Monte-Carlo averages. The computational cost thus scales as $2M_1M_2$. The transfer entropy estimate is unbiased[13] and the statistical accuracy can be arbitrarily improved by increasing M_1 and M_2. A pseudocode for the algorithm is available in the Appendix. Aside from Schreiber's transfer entropy,[1] other trajectory-based metrics of directional information transfer such as directed information,[17] conditional transfer entropy[18] and filtered transfer entropy,[19] also can be derived from conditional distributions of trajectories. Hence TE-PWS can be used to compute all such metrics at similar cost, as shown in the Appendix.

We demonstrate the validity of the method by analyzing two examples of an Ornstein-Uhlenbeck (OU) process for which transfer entropy rates are available. Eq. 1 is an OU process when it is linear with $\mathbf{F} = -\mathbf{aX}$ where $\mathbf{a}$ is a spring constant matrix. The first example is a minimal model for describing the stochastic dynamics of gene expression and growth rate in bacteria.[20] The model is non-bipartite, meaning that the diffusion constant matrix is non-diagonal and the mutual information rates are not finite. Yet transfer entropy rates are finite and numerically exact results are available.[19] The other example is an OU process with two variables, for which transfer entropy rates are analytically available.[19] As shown in Fig. 2a and b, TE-PWS results converge to known transfer entropy rates within a few multiples of the relaxation timescales in the respective systems. Details about the models are provided in the Appendix.

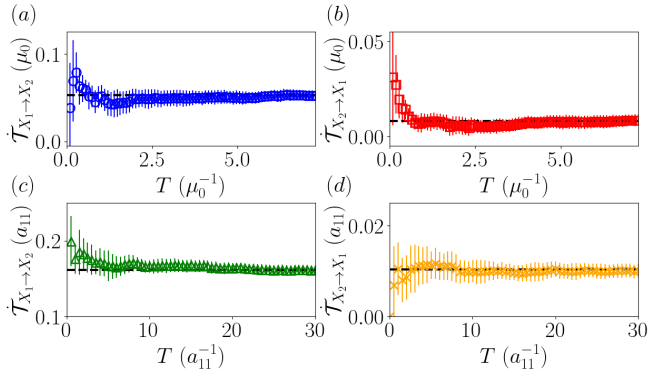

FIG. 2. Convergence of transfer entropy rate estimates. (a) and (b) respectively show forward and reverse transfer entropy rates in the stochastic gene expression model in units of the inverse relaxation timescale μ_0, as a function of the trajectory duration T. Symbols are from TE-PWS and dashed lines are numerically exact values. (c) and (d) show transfer entropy rates in the two-variable OU process. Symbols are from TE-PWS and dashed lines are analytical results.

Jump process. For systems with jumps between a discrete number of states and finite waiting times between the jumps, such as a well-stirred chemical reaction network described by a master equation, or a neural spiking process, the dynamics is governed by a jump propensity matrix $\mathbf{Q}$ of dimensions $\mathcal{N}^d \times \mathcal{N}^d$, which describes jumps among the $\mathcal{N}$ states of each of the d components of $\mathbf{X}$.[21] In contrast with diffusive processes which must be time-discretized to simulate, jump processes can be simulated exactly with event-driven kinetic Monte-Carlo algorithms such as the Gillespie algorithm.[22] Hence the transfer of information to X_j during the $(k+1)$-th trajectory segment occurs at all instants of time within the segment $X_{j,[k,k+1]}$, rather than at only the endpoint $X_j(k+1)$. Therefore in the definition of the stepwise increments to transfer entropy, $\Delta\mathcal{T}_{[k,k+1]}^{i\to j}$, in Eq. 4, the logarithms of the probabilities $P(X_j(k+1)|X_{i,[0,k]}, X_{j,[0,k]})$ and $P(X_j(k+1)|X_{j,[0,k]})$ should be replaced with functionals of the entire $X_{j,[k,k+1]}$ segment,

$$\pi_{X_i\to X_j} = -\int_{k\delta t}^{(k+1)\delta t} dt'\,\lambda_{ij}(t') + \sum_{\alpha=1}^{N_j} \ln \mathcal{Q}_{ij}(\alpha) \quad (9)$$

$$\pi_{X_j} = -\int_{k\delta t}^{(k+1)\delta t} dt'\,\lambda_{j}(t') + \sum_{\alpha=1}^{N_j} \ln \mathcal{Q}_{j}(\alpha) \quad (10)$$

$$\Delta\mathcal{T}_{[k,k+1]}^{i\to j} = \left\langle \pi_{X_i\to X_j} - \pi_{X_j} \right\rangle \quad (11)$$

where α counts the jumps that change the state of X_j, λ_{ij} and λ_j are escape propensities for X_j in the marginal spaces of (X_i, X_j) and (X_j) respectively, and $\mathcal{Q}_{ij}(\alpha)$ and $\mathcal{Q}_j(\alpha)$ similarly are marginal jump propensities for the α-th jump.[2,10] Q_{ij}, λ_{ij} and Q_j, λ_j are obtained by marginal-

izing jump propensities from the full $\mathbf{X}$-space into the (X_i, X_j) and (X_j) spaces respectively.[2,10]

The transfer entropy in Eq. 11 can be calculated with TE-PWS by marginalizing jump propensities in a Monte-Carlo fashion over conditional distributions of hidden variables, similar to the diffusive case. The average is Eq. 11 is expressed as a Monte-Carlo average over simulated trajectories of X_i and X_j, and each of $\mathcal{Q}_{ij}, \lambda_{ij}$ and $\mathcal{Q}_j, \lambda_j$ are expressed as Monte-Carlo averages over conditional distributions of hidden variables. Hidden variable trajectories are sampled from a P_0 propagated in the frozen field of the given trajectory, and the remaining bias is exactly corrected for through the RR scheme. Prior work argues that the escape terms involving λ_{ij} and λ_j in Eqs. 9 and 10 respectively can be omitted since they cancel each other on average.[2] We show however in the Appendix that the error in the transfer entropy estimate can be reduced by an order of magnitude by exploiting anti-correlated fluctuations between the escape and the jump terms.

Data processing inequality. We demonstrate the utility of TE-PWS by applying it to a three-node motif to study whether transfer entropies obey Data Processing Inequalities (DPIs). Unidirectional flow of information between different nodes in a network leads to DPIs for mutual information.[23] For a general three-variable process, if the flow of information is $X_1 \to X_2 \to X_3$, *i.e.*, without feedback, $I(X_{1,[0,N]}; X_{3,[0,N]}|X_{2,[0,N]}) = 0$; here the right arrows denote flow of information as mediated either via activation or repression. This leads by the chain rule to $I(X_{1,[0,N]}; X_{3,[0,N]}) \leq I(X_{1,[0,N]}; X_{2,[0,N]})$.[23] As transfer entropy equals mutual information in absence of feedback, it also obeys $\mathcal{T}_{X_1 \to X_3} \leq \mathcal{T}_{X_1 \to X_2}$. This relation bounds the amount of information that can be transmitted from input to output through an intermediate variable, yet is only valid in the absence of feedback. In presence of an $X_2 \to X_1$ feedback, *i.e.*, $X_1 \rightleftarrows X_2 \to X_3$, the mutual information continues to obey its DPI, while the transfer entropy formally doesn't.[24,25] This can be rationalized by considering the limit $X_1 \leftarrow X_2 \to X_3$, where X_2 controls both X_1 and X_3. Here the X_1 trajectories would still be predictive of fluctuations in X_3, thus $\mathcal{T}_{X_1 \to X_3} \geq 0$ even as $\mathcal{T}_{X_1 \to X_2} = 0$. However, where the crossover from the feedforward to the feedback dominated regime occurs, and to what extent in practice transfer entropies can overcome the DPI, is currently not understood.

We implement the motif $X_1 \rightleftarrows X_2 \to X_3$ with two diffusive models of mutual repression between X_1 and X_2, labeled A and B. In both models, X_3 rapidly copies the state of X_2 such that the information loss from X_2 to X_3 is low. Model A is a three-dimensional OU process with feedback, where the ratio of the feedback to feedforward spring constants $a_{12}/a_{21} = f^*$ is varied to study the violation of DPI. Model B is a nonlinear extension of A inspired by a genetic toggle switch[26] where the drifts for X_1 and X_2 are changed to $F_1 = -a_{11}X_1 - a_{12}(1+X_1^2)/(1+X_1^2+X_2^2)$ and $F_2 = -a_{22}X_2 - a_{21}(1+X_2^2)/(1+X_1^2+X_2^2)$. We choose $a_{12}/a_{21} = f^*$ such that the two models can be compared. Other model parameters are provided in the Appendix. Typical X_1 and X_2 trajectories at high feedback are shown in Figs. 3a and b. Model A only shows regression to the mean for both X_1 and X_2, while model B additionally shows switching between (low,high) and (high,low) regimes for X_1 and X_2 for high values of both feedforward and feedback coupling, around $f^* = 1$.

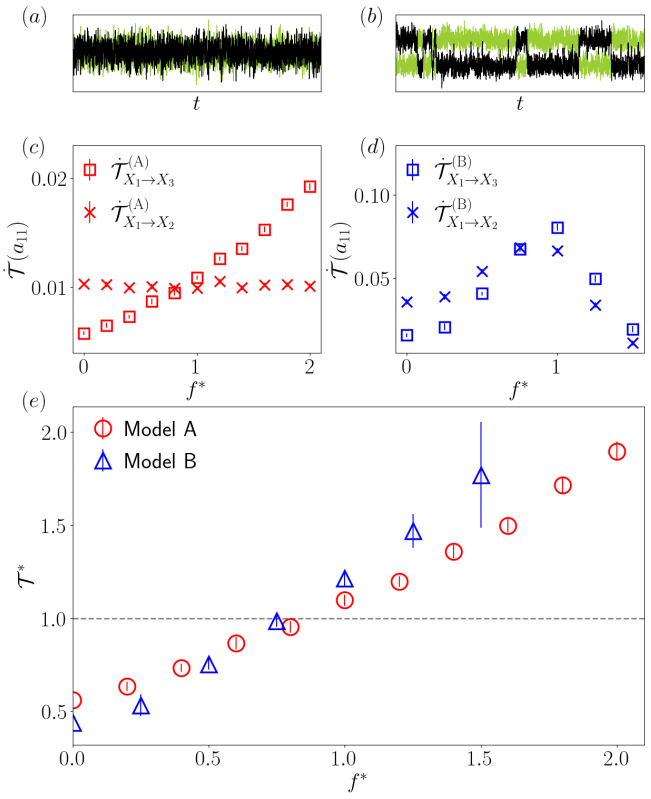

FIG. 3. (a) and (b) are typical X_1 (green) and X_2 (black) trajectories for models A and B at $a_{12} = 2a_{21} = 0.4$ and $a_{12} = a_{21} = -6$ respectively. (c) Transfer entropy rates in model A, denoted as $\dot{\mathcal{T}}^{(A)}_{X_1 \to X_3}$ and $\dot{\mathcal{T}}^{(A)}_{X_1 \to X_2}$, as a function of an increasing ratio of feedback to feedforward strength, f^*, when the feedforward strength is kept constant at $a_{21} = 0.2$. (d) Similar as (c) but for model B, where the feedforward strength is kept constant at $a_{21} = -4$. (e) Ratio of transfer entropy rates overcomes DPI bound (dashed line at $\mathcal{T}^* = 1$) with increasing feedback.

We plot in Fig. 3c and d transfer entropy rates for the two models. Each data point is obtained from steady state trajectories using TE-PWS. In model A, increasing the feedback strength keeps $\dot{\mathcal{T}}_{X_1 \to X_2}$ unchanged while monotonically increasing $\dot{\mathcal{T}}_{X_1 \to X_3}$. The former is an empirical result that we expect to formally hold for an OU process, though we have not analytically proven it. The latter arises because X_1 becomes increasingly correlated

with X_2 and the information is copied accurately by X_3. In contrast, in model B, both $\dot{\mathcal{T}}_{X_1\to X_2}$ and $\dot{\mathcal{T}}_{X_1\to X_3}$ peak near the switching regime, phenomenologically similar to how a bistable three-node motif with feedback has been shown to behave through approximate theory.[10] The values of the transfer entropy rates in model B are also amplified manyfold compared to model A due to stronger correlations between X_1 and X_2 resulting from the switching.

In Fig. 3e we have plotted the amount of DPI violation in both models quantified as a ratio of transfer entropy rates $\mathcal{T}^* = \dot{\mathcal{T}}_{X_1\to X_3}/\dot{\mathcal{T}}_{X_1\to X_2}$, as a function of the ratio of the feedback to feedforward strengths, f^*. Surprisingly, we find that regardless of the nature of the variation of the individual transfer entropy rates with increasing feedback, *i.e.*, monotonic or non-monotonic (Figs. 3c and d), the ratio $\mathcal{T}^*$ monotonically increases with increasing feedback (Fig. 3 e). Moreover, the ratio overcomes the DPI bound when the strength of the feedback becomes comparable to that of the feedforward coupling. Our results thus show that when the feedback $X_2 \to X_1$ dominates over the feedforward interaction, $X_1 \to X_2$, the feedforward entropy $\mathcal{T}_{X_1\to X_3}$ becomes larger than the feedforward entropy $\mathcal{T}_{X_1\to X_1}$. The mutual information, on the other hand, continues to obey its DPI, $I(X_{1,[0,N]}; X_{3,[0,N]}) \leq I(X_{1,[0,N]}; X_{2,[0,N]})$. We expect further analytical work in the OU process to be able to support this empirical result.

In conclusion, we have developed a method that for the first time makes it possible to compute transfer entropies exactly for any stochastic model. We have shown that TE-PWS is applicable to diverse stochastic processes and can elucidate novel physics of information transmission. We expect transfer entropies computed by TE-PWS to be used as *ground truth* for a wide range of goals, such as the characterization and design of information flow in natural and engineered information processing systems, and causality detection.

Acknowledgement We thank Manuel Reinhardt and Age Tjalma for useful discussions. This work is part of the Dutch Research Council (NWO) and was performed at the research institute AMOLF. This project has received funding from the European Research Council under the European Union's Horizon 2020 research and innovation program (grant agreement No. 885065).

Data availability A parallelized implementation of TE-PWS in Python is openly available on Zenodo at https://zenodo.org/doi/10.5281/zenodo.13617365..[27]

Appendix: Model details

Figs. 2 a and b show transfer entropies in a minimal stochastic model for gene expression.[19,20,28] The model is a three-dimensional OU process with spring constants $a_{11} = \mu_E + \mu_0 T_{\mu E}(T_{EG} - 1)$, $a_{12} = -\mu_0(T_{EG} - 1)$, $a_{13} = \mu_0 T_{EG}$, $a_{21} = -T_{\mu E}[\beta_G - \mu_E - \mu_0 T_{\mu E}(T_{EG} - 1)]$, $a_{22} = -[\mu_0 T_{\mu E}(T_{EG} - 1) - \beta_G]$, $a_{23} = -[\beta_G - \beta_\mu - \mu_0 T_{\mu E} T_{EG}]$, $a_{31} = a_{32} = 0$ and $a_{33} = \beta_\mu$, and diffusion constants $D_{11} = D_E\mu_0^2$, $D_{22} = \beta_\mu\eta_\mu^2 + \beta_G\eta_G^2 + D_E\mu_0^2 T_{\mu E}^2$, $D_{33} = \beta_\mu\eta_\mu^2$, $D_{12} = D_{21} = D_E\mu_0^2 T_{\mu E}$, $D_{13} = D_{31} = 0$ and $D_{23} = D_{32} = \beta_\mu\eta_\mu^2$, where the experimentally determined values of the parameters are $\mu_0 = 0.23h^{-1}$, $\beta_\mu = \beta_G = 0.33h^{-1}$, $\beta_E = 5.63h^{-}1$, $\eta_E = 1.03$, $\eta_\mu = 0.16$, $\eta_G = 0.22$, $T_{EE} = 1$, $T_{\mu E} = 0.7$, $T_{EG} = 1.3$, $\mu_E = \mu_0(1 + T_{\mu E} - T_{EE})$ and $D_E = \eta_E^2/\beta_E$..[19,20]

For Figs. 2c and d, the model is a two-dimensional OU process with parameters $a_{11} = 1$, $a_{22} = 2$, $a_{33} = 1.5$, $a_{12} = a_{21} = 0.5$, $a_{31} = 0.2$, $a_{32} = 0.3$, $D_{11} = 1.5$, $D_{22} = 0.5$, $D_{33} = 1$ and $a_{13} = a_{23} = D_{12} = D_{23} = D_{13} = 0$.

For models A and B in Fig. 3, we have chosen $a_{11} = a_{22} = 1$, $a_{33} = -a_{32} = 2$, $a_{13} = a_{31} = a_{23} = 0$ and $D_{ij} = \delta_{ij}$.

Error bars for all results are computed as twice the standard deviations in sets of statistically independent simulations.

TE-PWS algorithm

Central to TE-PWS is the computation of trajectory averages in a Monte-Carlo fashion over simulated trajectories, and the availability of trajectory probabilities on-the-fly in the full d-dimensional space. Thus for implementation in any stochastic model, two quantities need to be specified- a simulation method and the explicit functional form of the trajectory probability.

For diffusion processes modeled by Eq. 1, we simulate it using an Euler-Maruyama scheme with a fixed small timestep. In our examples we have taken the timestep to be equal to the duration of trajectory segments for implementing the RR scheme, δt, for convenience. The propagation equation for the $(k+1)$-th step is $\mathbf{X}(k+1) = \mathbf{X}(k) + \delta t\mathbf{F}(k) + \sqrt{\delta t}\boldsymbol{\psi}(k)$, where $\boldsymbol{\psi}$ is a Gaussian random vector with mean zero and variance $\langle\psi_i(k)\psi_j(k')\rangle = 2D_{ij}\delta_{k,k'}$. Making time discrete results in an $\mathcal{O}(\delta t)$ error which can be systematically made arbitrarily small. The probability density of the change of state $\Delta\mathbf{X}(k)$ can be written analytically through Ito discretization of the Onsager-Machlup action as[14,15]

$$P(\Delta\mathbf{X}(k)) = \frac{1}{(4\pi\delta t)^{d/2}|\mathbf{D}|}\exp\big[-(\Delta\mathbf{X}(k) - \mathbf{F}(k)\delta t)^T \cdot \mathbf{D}^{-1}(\Delta\mathbf{X}(k) - \mathbf{F}(k)\delta t)/4\delta t\big] \tag{12}$$

where $|\mathbf{D}|$ is the determinant of the diffusion constant matrix. This form also holds for systems with inertia if the generalized coordinate vector $\mathbf{X}$ contains both positions and velocities.[29,30]

For jump processes, we simulate them with a Gillespie algorithm which is exact, *i.e.*, does not make a timestep

error. The Gillespie algorithm is implemented as follows. At any given time, the waiting time till the next jump is exponentially distributed. Hence at the start of the trajectory and after every jump, denoting the time as t, we sample a uniform random number $u \in [0,1]$ and determine the wait time τ by solving

$$\ln u = -\int_t^{t+\tau} dt^{'} \lambda(t^{'}) \tag{13}$$

for τ. Here λ is the escape propensity from state $\mathbf{X}(t^{'})$, given by the sum of all jump propensities taking the system out of the state $\mathbf{X}(t^{'})$. When propagating all coordinates at once, $\lambda(t^{'})$ is independent of $t^{'}$ in between jumps, hence the equation is trivially solved for τ. Then a second uniform random number is drawn to choose which jump to fire, with probabilities proportional to their jump propensities. The probability density of a trajectory segment $\mathbf{X}_{[t,t+\delta t]}$ is written as

$$\ln P(\mathbf{X}_{[t,t+\delta t]}) = -\int_t^{t+\delta t} dt^{'} \lambda(t^{'}) + \sum_{\alpha=1}^{N_{tot}} \ln \mathcal{Q}_\alpha \tag{14}$$

where α sums over all jumps, N_{tot} in number.

The TE-PWS algorithm for both diffusive and jump processes then proceeds as follows.

- Propagate M_1 trajectories of X_i and X_j in the full d-dimensional space (for diffusive processes, see Eq. 5). These trajectories are henceforth labeled with (ν).

- For each pair of $(X_i^{(\nu)}, X_j^{(\nu)})$ trajectories, propagate M_2 trajectories of hidden variables X_l using the chosen reference distribution $P_0(X_{l,[0,k]})$ (for diffusive processes, the calculation here onwards corresponds to the computation of the numerator in Eq. 5).

- After every δt time, recalculate logarithmic weights $w^{(\mu)}$ for the trajectories defined as the logarithm of the ratio between the joint distribution $P(X^{(\nu)}_{i,[0,k]}, X^{(\nu)}_{j,[0,k]}, X_{l,[0,k]})$ and the reference distribution $P_0(X_{l,[0,k]})$. This ratio is proportional to the ratio between the conditional distribution $P(X_{l,[0,k]}|X^{(\nu)}_{i,[0,k]}, X^{(\nu)}_{j,[0,k]})$ and the reference distribution $P_0(X_{l,[0,k]})$, as shown in the next section. Then calculate the uniformity in the weights with a uniformity parameter $\kappa = (\sum_\mu \exp w^{(\mu)})^2 / \sum_\mu \exp(2w^{(\mu)})$, where sums of exponentials of weights are always performed with the Log-Sum-Exp trick.[31]

- Calculate the contribution to the transfer entropy in the space of (X_i, X_j) at the $(k+1)$-th step, denoted as $\mathcal{T}_a^{(\nu)}[k]$. This is done by computing the average in Eq. 8 for Langevin processes, and taking an expectation of Eq. 9 for jump processes.

- If $\kappa < M_2/2$, resample the M_2 trajectories with the accumulated weights $w^{(\mu)}$. This means we sample M_2 trajectories randomly with weights $w^{(\mu)}$ from the simulated ensemble of M_2 trajectories, with replacement. Set all weights to zero after every resampling.

- Go back to the second item on this list and repeat the procedure for each $(X_j^{(\nu)})$ trajectory (for diffusive processes, this calculation corresponds to the computation of the denominator in Eq. 5). Specifically, propagate M_2 trajectories of hidden variables (X_i, X_l) using the chosen reference distribution $P_0(X_{i,[0,k]}, X_{l,[0,k]})$, accumulate weights for each trajectory as the ratio between the joint distribution $P(X_{i,[0,k]}, X^{(\nu)}_{j,[0,k]}, X_{l,[0,k]})$ and the reference distribution $P_0(X_{i,[0,k]}, X_{l,[0,k]})$, compute the contribution to the transfer entropy in the (X_j) space at the $(k+1)$-th step, denoted as $\mathcal{T}_b^{(\nu)}[k]$ (Eqs. 7 and 10), and resample if the uniformity in the weights is low, *i.e.*, $\kappa < M_2/2$.

- Finally, compute $\mathcal{T}_{X_i \to X_j}$ using Eqs. 5 and 11 for Langevin and jump processes respectively.

For clarity, a pseudocode to compute transfer entropy $\mathcal{T}_{X_1 \to X_3}$ in only a three-variable process is given in Algorithm 1.

Proof of the efficiency of the RR scheme

TE-PWS performs the average in Eq. 7 with high statistical efficiency by preferentially sampling rare large values of the summand. As discussed in [13], this is implemented with a stochastic particle filter through which the ensemble of M_2 trajectories is resampled after every δt time. For example, in order to compute $P(X^{(\nu)}_{j,[0,N]})$, an ensemble of X_i and X_l trajectories is simulated. At the $(k+1)$-th step, the resampling weight used for the μ-th trajectory in the ensemble is

$$\begin{aligned}
&\widehat{g}\big[X^{(\mu)}_{l,[0,k+1]}\big] \\
&= P\left(X^{(\mu)}_{i,[k,k+1]}, X^{(\nu)}_{j,[k,k+1]}, X^{(\mu)}_{l,[k,k+1]} \middle| X^{(\mu)}_{i,[0,k]}, X^{(\nu)}_{j,[0,k]}, X^{(\mu)}_{l,[0,k]}\right) \\
&\quad / \; P_0\left(X^{(\mu)}_{i,[k,k+1]}, X^{(\mu)}_{l,[k,k+1]} \middle| X^{(\mu)}_{i,[0,k]}, X^{(\mu)}_{l,[0,k]}\right)
\end{aligned} \tag{15}$$

which is analytically available. Using this weight, we show below that iterative resampling changes the trajectory distribution optimally such that the summand in Eq. 7 is a constant, *i.e.*, has a variance of zero, achieving perfect sampling. We will show this by first showing that if after the k-th step the trajectories have an optimal distribution, they stay optimal after the $(k+1)$-th step. Then, combined with the fact that the distribution of

Algorithm 1 Transfer Entropy- Path Weight Sampling (TE-PWS)

inputs model parameters for simulating $\mathbf{X}(t)$
parameters timestep δt; number of timesteps N; number of trajectories for Monte-Carlo averages M_1 and M_2; resampling indicator κ
initialize Define trajectory labels ν and μ; timestep variable k; cumulative transfer entropy array $\mathcal{T}[0:N]$; logarithm of path probabilities of X_3 within the spaces (X_1, X_3) and (X_3) as $\mathcal{T}_a^{(\nu)}[0:N]$ and $\mathcal{T}_b^{(\nu)}[0:N]$ respectively.
$\nu \leftarrow 0$
$\mathcal{T}[0:N] = 0$
repeat
  Generate M_1 trajectories $(X^{(\nu)}_{1,[0,N]}, X^{(\nu)}_{2,[0,N]}, X^{(\nu)}_{3,[0,N]})$ jointly.
  $k \leftarrow 0$ ▷ Accessing $P(X_{2,[0,N]}|X_{1,[0,N]}, X_{3,[0,N]})$
  $\kappa \leftarrow M_2$
  initialize Generate M_2 samples of initial conditions $X_2^{(\mu)}(0)$ labeled by μ from a steady-state trajectory; weights in log scale $w^{(\mu)} = 0$.
  repeat
    if $\kappa < M_2/2$ **then**
      Resample M_2 configurations from $X_2^{(\mu)}(k)$ with weights $\exp w^{(\mu)}$.
      For a jump process, additionally regenerate random values for the first waiting time and the first jump in each trajectory using the jump propensities.
      $w^{(\mu)} \leftarrow 0$ for all μ
    end if
    Propagate reference dynamics $X^{(\mu)}_{2,[k,k+1]}$ as samples from $P_0(X_{2,[k,k+1]}|X_{2,[0,k]})$.
    $w^{(\mu)} \leftarrow w^{(\mu)} + \ln P\left(X^{(\nu)}_{1,[k,k+1]}, X^{(\mu)}_{2,[k,k+1]}, X^{(\nu)}_{3,[k,k+1]} | X^{(\nu)}_{1,[0,k]}, X^{(\mu)}_{2,[0,k]}, X^{(\nu)}_{3,[0,k]}\right) - \ln P_0\left(X^{(\mu)}_{2,[k,k+1]} | X^{(\mu)}_{2,[0,k]}\right)$
    $\kappa \leftarrow (\sum_\mu \exp w^{(\mu)})^2 / \sum_\mu \exp(2w^{(\mu)})$
    Compute $\mathcal{T}_a^{(\nu)}[k]$ using Eq. 8 or 9 for a diffusion or jump process respectively.
    $k \leftarrow k + 1$
  until $k = N$
  $k \leftarrow 0$ ▷ Accessing $P(X_{1,[0,N]}, X_{2,[0,N]}|X_{3,[0,N]})$
  $\kappa \leftarrow M_2$
  initialize Generate M_2 samples of initial conditions $(X_1^{(\mu)}(0), X_2^{(\mu)}(0))$ labeled by μ from a steady-state trajectory; weights in log scale $w^{(\mu)} = 0$.
  repeat
    if $\kappa < M_2/2$ **then**
      Resample M_2 configurations from $(X_1^{(\mu)}(k), X_2^{(\mu)}(k))$ with weights $\exp w^{(\mu)}$.
      For a jump process, additionally regenerate random values for the first waiting time and the first jump in each trajectory using the jump propensities.
      $w^{(\mu)} \leftarrow 0$ for all μ
    end if
    Propagate reference dynamics $(X^{(\mu)}_{1,[k,k+1]}, X^{(\mu)}_{2,[k,k+1]})$ as samples from $P_0(X_{1,[k,k+1]}, X_{2,[k,k+1]}|X_{1,[0,k]}, X_{2,[0,k]})$.
    $w^{(\mu)} \leftarrow w^{(\mu)} + \ln P\left(X^{(\mu)}_{1,[k,k+1]}, X^{(\mu)}_{2,[k,k+1]}, X^{(\nu)}_{3,[k,k+1]} | X^{(\mu)}_{1,[0,k]}, X^{(\mu)}_{2,[0,k]}, X^{(\nu)}_{3,[0,k]}\right)$
    $w^{(\mu)} \leftarrow w^{(\mu)} - \ln P_0\left(X^{(\mu)}_{1,[k,k+1]}, X^{(\mu)}_{2,[k,k+1]} | X^{(\mu)}_{1,[0,k]}, X^{(\mu)}_{2,[0,k]}\right)$
    $\kappa \leftarrow (\sum_\mu \exp w^{(\mu)})^2 / \sum_\mu \exp(2w^{(\mu)})$
    Compute $\mathcal{T}_b^{(\nu)}[k]$ using Eq. 7 or 10 for a diffusion or jump process respectively.
    $k \leftarrow k + 1$
  until $k = N$
  $\mathcal{T}[0:N] \leftarrow \mathcal{T}[0:N] + \mathcal{T}_a^{(\nu)}[0:N] - \mathcal{T}_b^{(\nu)}[0:N]$
  $\nu \leftarrow \nu + 1$
until $\nu = M_1$
$\mathcal{T}[0:N] \leftarrow \mathcal{T}[0:N]/M_1$

initial conditions is by construction optimal, we will conclude by induction that the distribution stays uniform during the entire duration of the trajectory.

The optimal choice for P_0 would be the hypothetical $P_0(X^{(\mu)}_{i,[0,N]}, X^{(\mu)}_{l,[0,N]}) = P(X^{(\mu)}_{i,[0,N]}, X^{(\mu)}_{l,[0,N]}|X^{(\nu)}_{j,[0,N]})$ as it makes the summand in Eq. 7 a constant independent of the index μ. The purpose of the particle filter is to bias the simulated distribution $P_0(X^{(\mu)}_{i,[0,N]}, X^{(\mu)}_{l,[0,N]})$ towards the optimal distribution $P(X^{(\mu)}_{i,[0,N]}, X^{(\mu)}_{l,[0,N]}|X^{(\nu)}_{j,[0,N]})$. For the inductive argument, assume that after the k-th step,

the trajectories are distributed optimally according to $P(X^{(\mu)}_{i,[0,k]}, X^{(\mu)}_{l,[0,k]}|X^{(\nu)}_{j,[0,k]})$. After the propagation step, the probability of each trajectory changes to a product of its previous value and the probability of the new segment,

$$\begin{aligned} w^{(\mu)} &= P\left(X^{(\mu)}_{i,[0,k]}, X^{(\mu)}_{l,[0,k]}\middle|X^{(\nu)}_{j,[0,k]}\right) \\ &\quad . P_0\left(X^{(\mu)}_{i,[k,k+1]}, X^{(\mu)}_{l,[k,k+1]}\middle|X^{(\mu)}_{i,[0,k]}, X^{(\mu)}_{l,[0,k]}\right) \\ &= P\left(X^{(\mu)}_{i,[0,k]}, X^{(\nu)}_{j,[0,k]}, X^{(\mu)}_{l,[0,k]}\right) \\ &\quad . P_0\left(X^{(\mu)}_{i,[k,k+1]}, X^{(\mu)}_{l,[k,k+1]}\middle|X^{(\mu)}_{i,[0,k]}, X^{(\mu)}_{l,[0,k]}\right) \\ &\quad / P\left(X^{(\nu)}_{j,[0,k]}\right) \end{aligned} \tag{16}$$

Then we resample the trajectories with weight $\widehat{g}$ given in Eq. 15. The probability after resampling becomes proportional to the product of the Eqs. 15 and 16,

$$w^{(\mu)}\widehat{g}\big[X^{(\mu)}_{l,[0,k+1]}\big] = \frac{P\left(X^{(\mu)}_{i,[0,k+1]}, X^{(\nu)}_{j,[0,k+1]}, X^{(\mu)}_{l,[0,k+1]}\right)}{P\left(X^{(\nu)}_{j,[0,k]}\right)} \tag{17}$$

where Bayes' theorem has been used to condense the numerator. The normalization constant for this probability is obtained by summing over all $X^{(\mu)}_{i,[0,k+1]}$ and $X^{(\mu)}_{l,[0,k+1]}$ trajectories, which gives $P\left(X^{(\nu)}_{j,[0,k+1]}\right)/P\left(X^{(\nu)}_{j,[0,k]}\right)$. Dividing Eq. 17 by this normalization constant gives the new normalized probability distribution as $P(X^{(\mu)}_{i,[0,k+1]}, X^{(\mu)}_{l,[0,k+1]}|X^{(\nu)}_{j,[0,k+1]})$.

Hence the trajectories remain distributed optimally after resampling. Additionally, initial conditions for the M_2 trajectories are sampled from the same distribution as the original M_1 trajectories. Then by induction, the resampling procedure, *i.e.*, the RR scheme, gives access to the exact conditional distribution at every step. Resampling is algorithmically performed with a stratified resampling technique which is computationally efficient.[32] The optimal distribution is then used in Eq. 7 to compute the denominator in Eq. 5, a part of the transfer entropy. The numerator is obtained similarly by accessing the corresponding conditional distribution with the RR scheme. Conditional distributions in trajectory space for all combinations of variables can be accessed by repeating this procedure.

Choice of reference distribution

The transfer entropy estimated from TE-PWS is exact for any choice of the reference probability P_0 owing to the RR scheme. However, the number of times pruning and enrichment needs to be performed depends on how large the variance of the summand in Eq. 7 is. A better choice of P_0 results in a smaller variance at the same computational cost. As discussed earlier, for computing $P(X^{(\nu)}_{j,[0,N]})$ for example, the ideal choice for P_0 would be $P_0(X^{(\mu)}_{i,[0,N]}, X^{(\mu)}_{l,[0,N]}) = P(X^{(\mu)}_{i,[0,N]}, X^{(\mu)}_{l,[0,N]}|X^{(\nu)}_{j,[0,N]})$, which is not known *a priori* and is impossible to directly sample from. We instead choose a distribution $P_0(X^{(\mu)}_{i,[0,N]}, X^{(\mu)}_{l,[0,N]})$ that uses the past trajectory of $X^{(\nu)}_{j,[0,N]}$ at every timestep to compute the drift and diffusion terms, similar to the original dynamics of the system in the full d-dimensional space. This keeps the reference distribution close to the target conditional distribution while being analytically known and easy to sample from.

As an example, consider the three-dimensional OU process discussed before, $\dot{\mathbf{X}} = -\mathbf{a}\mathbf{X} + \boldsymbol{\xi}$ where $\boldsymbol{\xi}$ is a Gaussian white noise with zero mean and covariance matrix $2\mathbf{D}$. The diffusion constant matrix $\mathbf{D}$ may have nonzero off-diagonal elements. In the full three-dimensional space, propagating the natural dynamics involves computing the drifts and sampling three correlated noise components from the joint Gaussian distribution, which we will call $\mathbb{G}(\xi_i, \xi_j, \xi_l)$. For sampling from a $P_0(X^{(\mu)}_{i,[0,N]}, X^{(\mu)}_{l,[0,N]})$ that is as close as possible to $P(X^{(\mu)}_{i,[0,N]}, X^{(\mu)}_{l,[0,N]}|X^{(\nu)}_{j,[0,N]})$, we propagate $X^{(\mu)}_i$ and $X^{(\mu)}_l$ trajectories with the equations of motion

$$\begin{aligned} \dot{X}^{(\mu)}_i(k) &= -a_{11}X^{(\mu)}_i(k) - a_{12}X^{(\nu)}_j(k) - a_{13}X^{(\mu)}_l(k) \\ &\quad + \tilde{\xi}^{(\mu)}_i(k) \end{aligned} \tag{18}$$

$$\begin{aligned} \dot{X}^{(\mu)}_l(k) &= -a_{31}X^{(\mu)}_i(k) - a_{32}X^{(\nu)}_j(k) - a_{33}X^{(\mu)}_l(k) \\ &\quad + \tilde{\xi}^{(\mu)}_l(k) \end{aligned} \tag{19}$$

where $\tilde{\xi}^{(\mu)}_i$ and $\tilde{\xi}^{(\mu)}_l$ are Gaussian white noises whose distribution should be commensurate with the existing noises in the $X^{(\nu)}_{j,[0,N]}$ trajectory. However, given only the $X^{(\nu)}_{j,[0,N]}$ trajectory, the noises $\xi^{(\nu)}_j(k)$ are not uniquely known, as equivalently the drifts $F^{(\nu)}_j(k)$ are unknown. We now make an additional choice of deriving approximate noises for $X^{(\nu)}_{j,[0,N]}$, called $\tilde{\xi}^{\nu\mu}_j$, by assuming that the drift in X_j was derived in turn from the $X^{(\mu)}_{i,[0,k]}$ and $X^{(\mu)}_{j,[0,k]}$ trajectories. Using the given $X^{(\nu)}_j$ trajectory, at every timestep, we solve

$$\begin{aligned} \dot{X}^{(\nu)}_j(k) &= -a_{21}X^{(\mu)}_i(k) - a_{22}X^{(\nu)}_j(k) - a_{23}X^{(\mu)}_l(k) \\ &\quad + \tilde{\xi}^{(\nu\mu)}_j(k) \end{aligned} \tag{20}$$

for $\tilde{\xi}^{(\nu\mu)}_j(k)$, which can be done by simply transposing the equation even if the original dynamics was not linear. Now given $\tilde{\xi}^{(\nu\mu)}_j(k)$, from which distribution should we generate $\tilde{\xi}^{(\mu)}_i(k)$ and $\tilde{\xi}^{(\mu)}_l(k)$? Here we use the conditional noise distribution in the natural dynamics of the system in order to keep the reference distribution close

to optimal. Given the multivariate Gaussian distribution $\mathbb{G}(\xi_i, \xi_j, \xi_l)$ for the original system of noises, and given that we have specified $\xi_j = \tilde{\xi}_j^{(\nu\mu)}(k)$, the distribution we need to sample from is given by the conditional distribution $\mathbb{G}(\xi_i = \tilde{\xi}_i, \xi_l = \tilde{\xi}_l | \xi_j = \tilde{\xi}_j^{(\nu\mu)}(k))$. Using the Schur complement formula, the mean and the covariance matrix of this distribution are given by $\Sigma_0 \Sigma^{-1} \tilde{\xi}_j^{(\nu\mu)}$ and covariance matrix $2\widetilde{D} - \Sigma_0 \Sigma^{-1} \Sigma_0^T$ respectively, where

$$\Sigma_0 = 2 \begin{pmatrix} D_{ij} \\ D_{lj} \end{pmatrix}, \tag{21}$$

$$\Sigma = 2 D_{jj}, \text{ and} \tag{22}$$

$$\widetilde{D} = \begin{pmatrix} D_{ii} & D_{il} \\ D_{il} & D_{ll} \end{pmatrix}. \tag{23}$$

The summary of the procedure for generating $X_i^{(\mu)}$ and $X_l^{(\mu)}$ trajectories for a given $X_j^{(\nu)}$ trajectory consists of the following steps at every timestep.

- The drifts for propagating $X_i^{(\mu)}(k)$ and $X_l^{(\mu)}(k)$ are obtained using $X_j^{(\nu)}(k)$.
- From the difference between $X_j^{(\nu)}(k)$ and $X_j^{(\nu)}(k+1)$, $\dot{X}_j^{(\nu)}(k)$ is obtained.
- From $X_i^{(\mu)}(k)$ and $X_l^{(\mu)}(k)$, together with $X_j^{(\nu)}(k)$, the drift in $\dot{X}_j^{(\nu)}(k)$ is obtained.
- From $\dot{X}_j^{(\nu)}(k)$ and the drift in X_j, the noise $\tilde{\xi}_j^{(\nu\mu)}(k)$ is obtained.
- With this noise $\tilde{\xi}_j^{(\nu\mu)}(k)$ specified, we can sample the noise $\tilde{\xi}_i(k)$ and $\tilde{\xi}_l(k)$.
- Using the drifts and the noise, we propagate $X_i^{(\mu)}(k)$ and $X_l^{(\mu)}(k)$.

Although $X_i^{(\mu)}$ and $X_l^{(\mu)}$ trajectories become overall a complicated nonlinear function of the $X_j^{(\nu)}$ trajectory due to the conditional noise sampling, the computations in each step are linear and simple. For accessing conditional distributions of trajectories of other variables, the reference dynamics is worked out similarly. Thus for every marginal probability computation for the ν-th trajectory, a unique reference dynamics is used, which is fine-tuned to that trajectory. This method of choosing a reference dynamics is a numerical analogue of constructing an approximation for the solution to the stochastic filtering equation,[10,33] albeit one whose error can be exactly corrected through trajectory reweighting. This drastically reduces the computational cost and makes TE-PWS feasible and accurate.

Reduced-variance estimator in jump process

Despite the definition of $\Delta\mathcal{T}^{i\to j}_{[k,k+1]}$ in Eq. 11 through both escape and jump propensities, only the latter terms contribute to it on average, as the escape propensities cancel, $\langle \lambda_{ij} \rangle = \langle \lambda_j \rangle$.[2] Thus only the jump propensities at the jumps, marginalized over the conditional distribution of hidden variables, are formally needed. For each marginalization, we need weights of trajectories upto the time of each jump. This is indeed available on-the-fly through the RR scheme. These weights thus provide the conditional distributions $P(X_{l,[0,t_\alpha]} | X_{i,[0,t_\alpha]}, X_{j,[0,t_\alpha]})$ and $P(X_{i,[0,t_\alpha]}, X_{l,[0,t_\alpha]} | X_{j,[0,t_\alpha]})$, where with a slight abuse of notation, the trajectories are now shown upto jump times t_α rather than $k\delta t$. We then use these two conditional distributions to compute $\mathcal{Q}_{ij}$ and $\mathcal{Q}_j$ in Eqs. 9 and 10 respectively for the α-th jump as,

$$\mathcal{Q}_{ij} = \int D[X_{l,[0,t_\alpha]}]\; Q_\alpha\; P(X_{l,[0,t_\alpha]} | X_{i,[0,t_\alpha]}, X_{j,[0,t_\alpha]}) \tag{24}$$

$$\mathcal{Q}_j = \int D[X_{i,[0,t_\alpha]}]\; D[X_{l,[0,t_\alpha]}]\; Q_\alpha \;.\; P(X_{i,[0,t_\alpha]}, X_{l,[0,t_\alpha]} | X_{j,[0,t_\alpha]}) \tag{25}$$

where Q_α is the jump propensity in the full d-dimensional space, which is analytically available. Then using only the jump terms in Eqs. 9-11, we obtain an estimate of the transfer entropy, $\mathcal{T}^{(\mathrm{J})}_{X_i \to X_j}$.

Surprisingly, we find that the statistical error in this estimate can be reduced by an order of magnitude by adding the integral of the escape terms in Eqs. 9 and 10, marginalized with a quadrature of δt. This arises because fluctuations in the jump terms in Eqs. 9 and 10 are pathwise anti-correlated to those in the escape terms, even though the latter cancel on average, as explained below.

The improved estimate is obtained by marginalizing the escape terms in Eq. 9 and 10 over conditional distributions $P\left(X_{l,[0,k]} \middle| X_{i,[0,k]}, X_{j,[0,k]}\right)$ and $P\left(X_{i,[0,k]}, X_{l,[0,k]} \middle| X_{j,[0,k]}\right)$ respectively as obtained from the RR scheme, instead of $P\left(X_{l,[0,t]} \middle| X_{i,[0,t]}, X_{j,[0,t]}\right)$ and $P\left(X_{i,[0,t]}, X_{l,[0,t]} \middle| X_{j,[0,t]}\right)$ at every instant of time. This is a quadrature approximation for marginalization. The estimate $\mathcal{T}^{(\mathrm{J})}_{X_i \to X_j}$ is then added to it to account for the jump terms in Eqs. 9 and 10 exactly. With this estimate, denoted $\mathcal{T}^{(\mathrm{J+E})}_{X_i \to X_j}$, fluctuations in the logarithm of jump propensities are suppressed by anti-correlated fluctuations in escape propensities, resulting in a much smaller variance. Physically, if a jump fires more than average, the waiting times between the jumps become that much more improbable. We can show this by calculating the fluctuations in $\sum_\alpha \ln(Q_{ij}/Q_j) - \int dt' (\lambda_{ij} - \lambda_j)$ as below. Consider the firing of only one given jump of X_j with an average propensity $\mathcal{Q}_j = \mathcal{Q}^*$, *i.e.*, an aver-

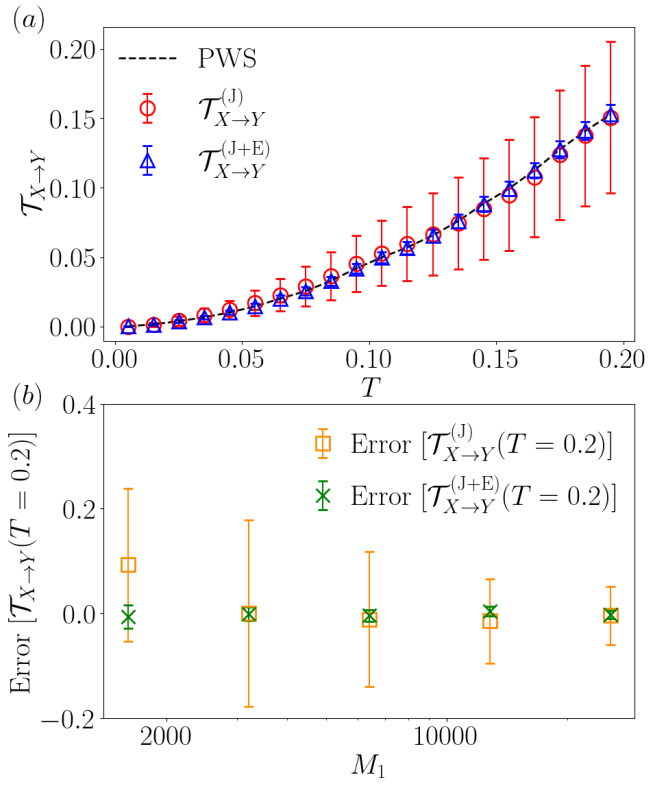

FIG. 4. Accuracy of transfer entropy estimates $\mathcal{T}^{(\mathrm{J})}_{X_i\to X_j}$ and $\mathcal{T}^{(\mathrm{J+E})}_{X_i\to X_j}$ in jump processes. (a) Transfer entropy as a function of time from the jump-based estimate $\mathcal{T}^{(\mathrm{J})}_{X_i\to X_j}$ (red circles), reduced-variance estimate $\mathcal{T}^{(\mathrm{J+E})}_{X_i\to X_j}$ (blue triangles) and exact estimate from PWS (black dashed line). (b) Error at $T = 0.2$, defined as the difference of each estimate from the PWS estimate, as a function of increasing Monte-Carlo averaging. Plotted are errors in the jump-based estimate $\mathcal{T}^{(\mathrm{J})}_{X_i\to X_j}$ (orange squares) and the reduced-variance estimate $\mathcal{T}^{(\mathrm{J+E})}_{X_i\to X_j}$ (green crosses). Initial conditions for the simulations are $n_X = 50$ and $n_Y = 500$, where n_X and n_Y are the number of species of X and Y respectively. M_1 for (a) is 25600. $M_2 = 1000$ for both subfigures.

age escape rate of $\lambda_j = \mathcal{Q}^*$. Fluctuations due to conditioning on a specific X_i trajectory cause $\mathcal{Q}_{ij}$ to deviate from its mean $\mathcal{Q}^*$, resulting in information transfer. An additional source of fluctuations is the stochastic number of times the jump fires within δt, N_j. Over a small δt, when Q_{ij} stays temporally almost constant, N_j is Poisson-distributed with a mean $Q_{ij}\delta t$. The change in transfer entropy using Eq. 11 then is

$$\Delta\mathcal{T}^{i\to j}_{[k,k+1]} = \left\langle Q_{ij}\delta t \ln\frac{Q^{ij}}{Q^*} - (Q_{ij}-Q^*)\delta t \right\rangle \tag{26}$$

We see that the second term fluctuates around a mean of 0. By Taylor expanding the first term upto second order for small values of $(Q_{ij} - Q^*)$, we find

$$\begin{aligned}\Delta\mathcal{T}^{i\to j}_{[k,k+1]} &= \Big\langle (Q_{ij}-Q^*)\delta t \\ &\quad + \frac{\delta t(Q_{ij}-Q^*)^2}{2Q^*} + \mathcal{O}\big((Q_{ij}-Q^*)^3\big) \\ &\quad - (Q_{ij}-Q^*)\delta t \Big\rangle \qquad (27)\\ &= \left\langle \frac{\delta t(Q_{ij}-Q^*)^2}{2Q^*} + \mathcal{O}\big((Q_{ij}-Q^*)^3\big) \right\rangle \qquad (28)\end{aligned}$$

So fluctuations in the second term of Eq. 26 cancel a part of the fluctuations in the first term.

We numerically demonstrate this effect in a chemical reaction network of two species X and Y, consisting of reactions $\phi \to X, X \to \phi, X \to X + Y, Y \to \phi$, with rates 50, 1, 10 and 10 respectively. Plotted in Figs. 4a and b are the two different transfer entropy estimates and their errors as a function of increasing trajectory duration and increasing statistical averaging respectively. As there is no feedback, the transfer entropy is formally equal to the exact mutual information estimate from PWS,[13] against which we have compared our results. We find that though both estimates yield unbiased results, the reduced-variance estimate $\mathcal{T}^{(\mathrm{J+E})}_{X_i\to X_j}$ has an order of magnitude smaller error than the jump-based estimate $\mathcal{T}^{(\mathrm{J})}_{X_i\to X_j}$, for the same computational cost. On the other hand, the $\mathcal{O}(\delta t)$ error in $\mathcal{T}^{(\mathrm{J+E})}_{X_i\to X_j}$ from the quadrature in marginalization is negligible. $\mathcal{T}^{(\mathrm{J+E})}_{X_i\to X_j}$ is thus a more accurate estimate of the transfer entropy than $\mathcal{T}^{(\mathrm{J})}_{X_i\to X_j}$. We expect this theoretical result to be tested in experiments in the future, such as using data from neural spike trains.[34]

Variants of transfer entropy

A central advance in TE-PWS is the computation of jump propensities by marginalization over hidden variables. Here we show, in the context of discretized Langevin processes, how the same approach can be used to compute other trajectory-based metrics of directional information transmission beyond Schreiber's transfer entropy.

Directed information. An alternate measure of information transmission from the trajectory $X_{i,[0,N]}$ to $X_{j,[0,N]}$ is given by directed information,[17] defined as

$$\begin{aligned}&I[X_{i,[0,N]} \to X_{j,[0,N]}] \\ &= \sum_{k=0}^{N-1} I\left(X_j(k+1); X_{i,[0,k+1]} | X_{j,[0,k]}\right) \\ &= \left\langle \ln \frac{\prod_k P\left(X_j(k+1)|X_{i,[0,k+1]}, X_{j,[0,k]}\right)}{\prod_k P\left(X_j(k+1)|X_{j,[0,k]}\right)} \right\rangle \qquad (29)\end{aligned}$$

By comparing the above definition with that of the transfer entropy in Eq. 2, we see that the directed information incorporates the stepwise information transmission into X_j coming from the entire trajectory of X_i including the current value. We can bring it to a computable form by taking $X_i(k+1)$ out of the conditioning in the numerator,

$$\begin{aligned}
&I[X_{i,[0,N]} \to X_{j,[0,N]}] \\
&= \left\langle \ln \frac{\prod_k P\left(X_j(k+1)|X_{i,[0,k+1]}, X_{j,[0,k]}\right)}{\prod_k P\left(X_j(k+1)|X_{j,[0,k]}\right)} \right\rangle \\
&= \left\langle \ln \prod_k P\left(X_i(k+1), X_j(k+1)|X_{i,[0,k]}, X_{j,[0,k]}\right) \right\rangle \\
&- \left\langle \ln \prod_k P\left(X_j(k+1)|X_{j,[0,k]}\right) \right\rangle \\
&- \left\langle \ln \prod_k P\left(X_i(k+1)|X_{i,[0,k]}, X_{j,[0,k]}\right) \right\rangle \qquad (30)
\end{aligned}$$

Here we note that only two marginalization integrals are actually required for computing the three probabilities in Eq. 30. One is over the conditional distribution $P(X_{l,[0,k]}|X_{i,[0,k]}, X_{j,[0,k]})$ and the other over $P(X_{i,[0,k]}, X_{l,[0,k]}|X_{j,[0,k]})$, exactly the same as those sampled in TE-PWS for calculating $\mathcal{T}_{X_i \to X_j}$. These two distributions can give the probabilities in Eq. 30 as

$$\begin{aligned}
&P\left(X_i(k+1), X_j(k+1)|X_{i,[0,k]}, X_{j,[0,k]}\right) \\
&= \int D[X_{l,[0,k]}]\, P\left(X_{l,[0,k]}|X_{i,[0,k]}, X_{j,[0,k]}\right) \\
&.\, P\left(X_i(k+1), X_j(k+1)|X_{i,[0,k]}, X_{j,[0,k]}, X_{l,[0,k]}\right) \qquad (31) \\
&P\left(X_i(k+1)|X_{i,[0,k]}, X_{j,[0,k]}\right) \\
&= \int D[X_{l,[0,k]}]\, P\left(X_{l,[0,k]}|X_{i,[0,k]}, X_{j,[0,k]}\right) \\
&.\, P\left(X_i(k+1)|X_{i,[0,k]}, X_{j,[0,k]}, X_{l,[0,k]}\right) \qquad (32) \\
&P\left(X_j(k+1)|X_{j,[0,k]}\right) \\
&= \int\int D[X_{i,[0,k]}]\, D[X_{l,[0,k]}]\, P\left(X_{i,[0,k]}, X_{l,[0,k]}|X_{j,[0,k]}\right) \\
&.\, P\left(X_j(k+1)|X_{i,[0,k]}, X_{j,[0,k]}, X_{l,[0,k]}\right) \qquad (33)
\end{aligned}$$

where, aside from the conditional distributions $P(X_{l,[0,k]}|X_{i,[0,k]}, X_{j,[0,k]})$ and $P(X_{i,[0,k]}, X_{l,[0,k]}|X_{j,[0,k]})$, all the other probabilities in the integrals are analytically available. Thus, by keeping track of one additional average over the conditional distributions sampled through the RR scheme, TE-PWS can compute directed information with the same complexity as transfer entropy.

Conditional transfer entropy. The conventional transfer entropy can have a positive value even when there is no direct causal link from the input to the output variable, if information is being causally transmitted through intermediate variables. This motivated the definition of a conditional transfer entropy, also known as a causation entropy, that can measure direct causal links.[24,35,36] For any choice of a third variable X_m, the conditional transfer entropy is defined as

$$\begin{aligned}
\mathcal{T}_{X_i \to X_j|X_m} &= \sum_{k=0}^{N-1} I\left(X_j(k+1); X_{i,[0,k]}|X_{j,[0,k]}, X_{m,[0,k]}\right) \\
&= \left\langle \ln \frac{\prod_k P\left(X_j(k+1)|X_{i,[0,k]}, X_{j,[0,k]}, X_{m,[0,k]}\right)}{\prod_k P\left(X_j(k+1)|X_{j,[0,k]}, X_{m,[0,k]}\right)} \right\rangle \qquad (34)
\end{aligned}$$

This expectation can be computed similar to Eq. 4. The average is computed in a Monte-Carlo fashion over simulated trajectories of all variables. For each set of trajectories, the numerator and denominator are computed by marginalizing over all other hidden variables X_l which exclude X_m this time. The optimal reference dynamics should now be chosen to incorporate the effects of the X_m trajectory through a frozen field of dirft and diffusion, similar to how the X_i and X_j trajectories influence the reference dynamics as discussed before. Thus, calculation of each conditional transfer entropy with TE-PWS requires two marginalization integrals, similar to the ordinary transfer entropy.

Filtered transfer entropy. Recently, filtered transfer entropy has been proposed as a way to quantify information transfer in the spirit of filtering theory.[19] The filtered transfer entropy from X_i to X_j is defined as

$$\begin{aligned}
\widehat{\mathcal{T}}_{X_i \to X_j} &= \sum_{k=0}^{N-1} I\left(X_i(k+1); X_j(k+1)|X_{j,[0,k]}\right) \\
&= \sum_k \left\langle \ln \frac{P\left(X_i(k+1), X_j(k+1)|X_{j,[0,k]}\right)}{P\left(X_i(k+1)|X_{j,[0,k]}\right) P\left(X_j(k+1)|X_{j,[0,k]}\right)} \right\rangle \qquad (35)
\end{aligned}$$

which quantifies how much the prediction of $X_i(k+1)$ is improved by using $X_j(k+1)$ additional to the past trajectory $X_{j,[0,k]}$. Computation of $\widehat{\mathcal{T}}_{X_i \to X_j}$ requires marginalization over only one conditional distribution, $P(X_{i,[0,k]}, X_{l,[0,k]}|X_{j,[0,k]})$. Each of the probabilities in Eq. 35 can be computed by averaging analytically available transition probabilities over this conditional distribution $P(X_{i,[0,k]}, X_{l,[0,k]}|X_{j,[0,k]})$, which is provided by TE-PWS. Thus, TE-PWS can be used to compute filtered transfer entropy at half the computational cost as Schreiber's transfer entropy.

[1] T. Schreiber, Measuring information transfer, Physical review letters **85**, 461 (2000).

[2] R. E. Spinney, M. Prokopenko, and J. T. Lizier, Transfer entropy in continuous time, with applications to jump and neural spiking processes, Physical Review E **95**, 032319 (2017).

[3] R. Vicente, M. Wibral, M. Lindner, and G. Pipa, Transfer entropy—a model-free measure of effective connectivity for the neurosciences, Journal of computational neuroscience **30**, 45 (2011).

[4] J. Runge, Causal network reconstruction from time series: From theoretical assumptions to practical estimation, Chaos: An Interdisciplinary Journal of Nonlinear Science **28** (2018).

[5] K. Hlaváčková-Schindler, M. Paluš, M. Vejmelka, and J. Bhattacharya, Causality detection based on information-theoretic approaches in time series analysis, Physics Reports **441**, 1 (2007).

[6] H. Mattingly, K. Kamino, B. Machta, and T. Emonet, Escherichia coli chemotaxis is information limited, Nature physics **17**, 1426 (2021).

[7] M. Prokopenko and J. T. Lizier, Transfer entropy and transient limits of computation, Scientific reports **4**, 5394 (2014).

[8] J. M. Horowitz and H. Sandberg, Second-law-like inequalities with information and their interpretations, New Journal of Physics **16**, 125007 (2014).

[9] F. Tostevin and P. R. Ten Wolde, Mutual information between input and output trajectories of biochemical networks, Physical review letters **102**, 218101 (2009).

[10] A.-L. Moor and C. Zechner, Dynamic information transfer in stochastic biochemical networks, Physical Review Research **5**, 013032 (2023).

[11] S. P. Strong, R. Koberle, R. R. D. R. Van Steveninck, and W. Bialek, Entropy and information in neural spike trains, Physical review letters **80**, 197 (1998).

[12] J. Runge, J. Heitzig, V. Petoukhov, and J. Kurths, Escaping the curse of dimensionality in estimating multivariate transfer entropy, Physical review letters **108**, 258701 (2012).

[13] M. Reinhardt, G. Tkačik, and P. R. Ten Wolde, Path weight sampling: Exact monte carlo computation of the mutual information between stochastic trajectories, Physical Review X **13**, 041017 (2023).

[14] L. C. G. Rogers and D. Williams, *Diffusions, Markov processes and martingales: Volume 2, Itô calculus*, Vol. 2 (Cambridge university press, 2000).

[15] L. Onsager and S. Machlup, Fluctuations and irreversible processes, Phys. Rev. **91**, 1505 (1953).

[16] D. Frenkel and B. Smit, *Understanding molecular simulation: from algorithms to applications*, Vol. 1 (Elsevier, 2001).

[17] J. Massey *et al.*, Causality, feedback and directed information, in *Proc. Int. Symp. Inf. Theory Applic.(ISITA-90)*, Vol. 2 (1990).

[18] L. Novelli, P. Wollstadt, P. Mediano, M. Wibral, and J. T. Lizier, Large-scale directed network inference with multivariate transfer entropy and hierarchical statistical testing, Network Neuroscience **3**, 827 (2019).

[19] R. Chetrite, M. Rosinberg, T. Sagawa, and G. Tarjus, Information thermodynamics for interacting stochastic systems without bipartite structure, Journal of Statistical Mechanics: Theory and Experiment **2019**, 114002 (2019).

[20] D. J. Kiviet, P. Nghe, N. Walker, S. Boulineau, V. Sunderlikova, and S. J. Tans, Stochasticity of metabolism and growth at the single-cell level, Nature **514**, 376 (2014).

[21] N. G. Van Kampen, *Stochastic processes in physics and chemistry*, Vol. 1 (Elsevier, 1992).

[22] D. T. Gillespie, A general method for numerically simulating the stochastic time evolution of coupled chemical reactions, Journal of computational physics **22**, 403 (1976).

[23] T. M. Cover, *Elements of information theory* (John Wiley & Sons, 1999).

[24] R. G. James, N. Barnett, and J. P. Crutchfield, Information flows? a critique of transfer entropies, Physical review letters **116**, 238701 (2016).

[25] M. S. Derpich and J. Østergaard, Directed data-processing inequalities for systems with feedback, Entropy **23**, 533 (2021).

[26] P. B. Warren and P. R. Ten Wolde, Chemical models of genetic toggle switches, The Journal of Physical Chemistry B **109**, 6812 (2005).

[27] A. Das and P. R. Ten Wolde, Computing exact transfer entropy with path weight sampling, 10.5281/zenodo.13617365 (2024).

[28] S. Lahiri, P. Nghe, S. J. Tans, M. L. Rosinberg, and D. Lacoste, Information-theoretic analysis of the directional influence between cellular processes, PLoS One **12**, e0187431 (2017).

[29] A. Das and D. T. Limmer, Variational control forces for enhanced sampling of nonequilibrium molecular dynamics simulations, The Journal of chemical physics **151**, 244123 (2019).

[30] J. S. Lee, J.-M. Park, and H. Park, Thermodynamic uncertainty relation for underdamped langevin systems driven by a velocity-dependent force, Physical Review E **100**, 062132 (2019).

[31] G. Gundersen, https://gregorygundersen.com/blog/2020/02/09/log-sum-exp/ (2020).

[32] R. Douc and O. Cappé, Comparison of resampling schemes for particle filtering, in *ISPA 2005. Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis, 2005.* (Ieee, 2005) pp. 64–69.

[33] M. Gehri, N. Engelmann, and H. Koeppl, Mutual information of a class of poisson-type channels using markov renewal theory, arXiv preprint arXiv:2403.15221 (2024).

[34] D. P. Shorten, R. E. Spinney, and J. T. Lizier, Estimating transfer entropy in continuous time between neural spike trains or other event-based data, PLoS computational biology **17**, e1008054 (2021).

[35] J. Sun and E. M. Bollt, Causation entropy identifies indirect influences, dominance of neighbors and anticipatory couplings, Physica D: Nonlinear Phenomena **267**, 49 (2014).

[36] L. Faes, D. Marinazzo, G. Nollo, and A. Porta, An information-theoretic framework to map the spatiotemporal dynamics of the scalp electroencephalogram, IEEE Transactions on Biomedical Engineering **63**, 2488 (2016).