

# From Neural to Social Computation: Collective Intelligence in Networks of Complex Agents

Marie-Lou Laprise\*

May 5, 2024

“Think how hard physics would be if particles could think” –Murray Gell-Mann

## 1 Introduction: Seeking inspiration from neural computation to model social dynamics

A variety of complex systems across scales, from colonies of ants to neurons in the brain, exhibit the capacity to map noisy inputs to useful outputs in a decentralized, emergent manner: they perform distributed computation.<sup>1</sup> Social networks also do so; a prominent example being price discovery in financial markets. However, this remains a blind spot for both social science and theories of collective computation. Social scientists tend to study decision-making by looking at individual behaviors or aggregate outcomes. They rarely take a complex systems approach, and they have not thought of social networks in terms of computation. On the other hand, existing models and theories of collective computation do not provide satisfactory explanations for the social setting, where agents are complex and heterogeneous. In neuroscience or biophysics, models often feature homogeneous, memory-less agents. In control systems and opinion formation studies, many models rely on linear dynamics that cannot replicate phenomena like threshold effects or information saturation.

How do social networks of heterogeneous agents collect and exchange noisy information to collectively form opinions and make decisions? Can we extend models of neural computation to explore this setting?

Interacting adaptive agents face some collective challenges that resemble those of neurons in the brain or nodes in an artificial network: (i) implementing distributed computation robustly and flexibly, over a range of time scales, and (ii) building predictive models of their environment. Together, I call tackling those goals collective intelligence. If there exists common principles underlying collective intelligence across settings, then models of neural computation may help us understand social computation. Conversely, models of social computation between complex agents might in turn suggest new models for how groups of neurons or regions of the brain interact to create a richer dynamical repertoire.

For instance, Naomi Leonard and colleagues developed a model of multi-agent opinion formation that takes a form similar to a recurrent neural network (RNN) (Bizyaeva, Franci, and Leonard 2022, Leonard, Bizyaeva, and Franci 2024) (the “BFL” model). I describe it in Section 2 below. The BFL model introduced nonlinearity to better model information saturation, in contrast to the linear averaging models that dominated the field of opinion dynamics. The BFL model exhibits the ability to break indecision and rapidly converge to a steady state of consensus (all agents agree) or disagreement (agreement within subgroups but not between them) in a way that, its authors argue, parallels the observed behavior of bees choosing a nesting site or voters choosing a candidate.

---

\*Some of the ideas for this project were developed in collaboration with Kamesh Krishnamurthy (Princeton Physics & PNI)

<sup>1</sup>I loosely follow Crutchfield (Crutchfield 1994) by defining **computation** as a useful mapping from an input to an output; and **distributed** to mean that computation emerges from the time-varying state of a system of many interacting elements.

The BFL model offered an elegant solution to a long-standing problem in the field of opinion formation: how to craft a system that may generate stable disagreement as a function of inputs and initial conditions, rather than one that inevitably converges to consensus or avoids consensus only at the cost of a fixed, carefully hand-picked connectivity matrix (Ravazzi and Proskurnikov 2021, Bernardo et al. 2024). This problem is sometimes called Abelson’s diversity puzzle (Abelson 1964).

The BFL model, however, features simple, homogeneous agents, without memory or intelligence. For this project, I take a first step to improve this aspect of the BFL model by adding a unit-specific dynamical time constant to the model. I then examine the behavior of the modified model using simulations in Julia.

## 2 Methods

### 2.1 Previous model

While the BFL model allows for multi-opinion formation, here I focus on the case of two mutually exclusive opinions represented by a scalar value  $z$ . The agent favors a given option when  $z > 0$ , he disfavors it when  $z < 0$ , and he is neutral when  $z = 0$ .

In the BFL model, each agent is represented by coupled ODEs. The instantaneous rate of change of (1) the opinion  $z$  of agent  $i$  and (2) the susceptibility or excitability  $u$  of agent  $i$ , for a network of size  $N$ , are as follows:

$$\dot{z}_i(t) = -d_i z_i(t) + S \left( u_i(t) \cdot \sum_{j=1}^N J_{ij}^z z_j(t) \right) + I_i(t) \quad (1)$$

$$\tau^u \dot{u}_i(t) = -u_i(t) + u_0 + g^u \sum_{j=1}^N J_{ij}^u (z_j(t))^2 \quad (2)$$

In the opinion update equation, the first term corresponds to linear negative feedback or self-inhibition, the second term to nonlinear positive feedback or recurrent excitation, and the last term is a time-varying input (stimuli or bias).

Susceptibility  $u$  to network information controls the relative strength of the external input vs the recurrent excitation received from the network.<sup>2</sup> The susceptibility update is independent of the input; it only depends on the strength of the opinions of connected agents received through the network. The intuition here is that you will pay more attention to the opinion of your friends if they are highly convinced than if they are neutral. If all your friends are neutral, you may ignore their input, and only use external information to make a decision. Conversely, if all your friends are highly convinced, you may use their opinion more than external information when making a decision.

Terms are defined as follows:

- $d_i > 0$  is a damping or resistance coefficient
- $S : \mathbb{R} \rightarrow \mathbb{R}$  is a bounded, sufficiently smooth saturation function such that:  $S(0) = 0$ ;  $S'(0) = 1$ ;  $S'''(0) \neq 0$ , which we set to be  $\tanh(x)$
- $J^z \in \mathbb{R}^{N \times N}$  is the communication matrix and  $J^u \in \mathbb{R}^{N \times N}$  is the susceptibility matrix (they can be the same or different)
- $I_i(t)$  is a time-varying input that includes external input and internal bias
- $\tau^u \geq 0$  is a time-constant

---

<sup>2</sup>In the initial model, the  $u$  term is called attention. Here, we call it susceptibility to avoid any confusion with other uses of the word attention in machine learning and neuroscience.

- $u_0$  is a basal level of susceptibility
- $g^u \geq 0$  is a constant susceptibility gain

A few things should be noted about the assumptions involved in this model:

1. Agents are homogeneous between each other (only their state distinguishes them, with the exception of the damping term  $d_i$ ) and across time (they do not learn or remember, only their state changes)
  - Agents all share a susceptibility time constant of  $\tau^u$  and an opinion time constant of 1
  - Agents all share a constant basal susceptibility and constant susceptibility gain
  - Agents all share an identical, non-parameterized input-output function
2. The self-inhibition implies that the opinion activation of each agent tends to revert to zero. This is balanced out by the self-excitation represented by the diagonal term of the  $J^z$  matrix, but modulated by the susceptibility state. At a low susceptibility, self-inhibition dominates.
  - While this is realistic for integrate-and-fire neurons, it may not be appropriate for agents forming opinions. A more realistic assumption might be inertia, such that opinions tend to stay the same, not revert to zero.
3. It is not clear why opinion states must be squared in the susceptibility update. This assumes symmetry in the impact of negative and positive opinions, which is disproved by behavioral data. It is possible that the difference between the two is small enough that this does not impact results.
4. In the presence of large external inputs, the opinion state update can be arbitrarily large.
5. While the damping coefficient  $d_i$  is indexed as unit-specific, in most reported simulations and experiments, the authors set a homogeneous  $d_i = d$ .

It should also be emphasized that contrary to the RNN case (and the main difference with the RNN case) is that each agent  $i$  has only access to its own input  $I_i$ . The only way for an agent to access inputs that were distributed to other agents is by communication through the adjacency graph.

## 2.2 Modified model

In real social networks, agents individually and dynamically adjust their reaction times, as they receive information from other agents and assess the state of the social system in which they are embedded. As a first step in modeling more complex, heterogeneous agents, I add to the BFL model an adaptive time constant.

In addition to better reflecting real social networks, the modified model should exhibit a richer dynamical range. Research on the physics of neural computation suggests that the nature of multiplicative interactions (or gating) in a model crucially determine its computational capabilities (Krishnamurthy, Can, and Schwab 2022). This change should allow the network (i) to flexibly handle information on a range of time scales, and (ii) to enter regimes in which continuous attractors are formed.

I modify the model to include an input-dependent adaptive time-constant for the opinion update, in the form of a parameterized multiplicative gate that controls the rate of integration of the information received from the network:

$$\dot{z}_i(t) = \phi(x_i(t)) \cdot \left( -d_i z_i(t) + S \left( u_i(t) \cdot \sum_{j=1}^N J_{ij}^z z_j(t) \right) \right) + I_i(t) \quad (3)$$

$$\tau^u \dot{u}_i(t) = -u_i(t) + u_0 + g^u \sum_{j=1}^N J_{ij}^u (z_j(t))^2 \quad (4)$$

$$\tau^x \dot{x}_i(t) = -x_i(t) + \sum_{j=1}^N J_{ij}^x S(z_j(t)) + k_i I_i(t) \quad (5)$$

Where  $J^x$  is a coupling matrix for the gate,  $\phi(x) = (1 + e^{-\alpha x + \beta})^{-1}$ , and  $k_i I_i$  is a scaled version of  $I_i$ .

The number of parameters, if all networks are directed (all  $J$  asymmetric), is  $3N^2 + 2N + 6$  (three  $N \times N$  matrices, two unit-specific parameters  $d_i$  and  $k_i$ , and six global parameters  $\tau^u$ ,  $\tau^x$ ,  $u_0$ ,  $g^u$ ,  $\alpha$ ,  $\beta$ ). In this set-up, parameters are not learned through back-propagation; they must be set as hyper-parameters.

## 2.3 Goal of the simulations

With the initial model, the authors examined indecision-breaking bifurcations. In the BFL model, the dynamic susceptibility state represents a bifurcation parameter for opinion cascades.

With neutral initial opinions and in the absence of inputs, the state vector representing the opinions of all agents remains at the zero fixed point, the only stable state. When some agents receive short external inputs, the system may go back to the zero fixed point, or undergo a an indecision-breaking bifurcation toward a regime of multistability, in which case the agents will converge to some stable opinionated state that will outlast the external inputs.

The bifurcation parameter for this transition is the susceptibility state vector. This is because the susceptibility determines the relative importance, for each agent, of self-inhibition versus recurrent excitation from the network (including self-excitation, if any).

The authors also showed that at the bifurcation, indecision-breaking occurs very rapidly, in a switch-like manner. As the system nears bifurcation, it becomes “selectively ultrasensitive to inputs,” meaning that it will respond strongly to input vectors that align with the right eigenspace of the Jacobian at the bifurcation, but ignore input vectors that do not. Away from the bifurcation, opinion states remain stable even in the presence of small perturbations, but large perturbations can move the system to a different fixed point.

In the modified model

## 2.4 Simulations

### 2.4.1 Opinion, susceptibility and gating parameters

Some assumptions must be imposed on the hyper-parameters to reduce the number of simulations to be run to a feasible number. Some of these assumptions may be relaxed or modified in future work to explore the dynamics of the model in different regimes.

For the simplest case, I set  $\tau^u = 1$ ,  $g^u = 1$ ,  $u_0 = 0.01$ ,  $\tau^x = 1$ ,  $\alpha = 1$ , and  $\beta = 0$ . The damping and scaling factors are sampled from a Gaussian ( $d_i \sim \mathcal{N}(0.5, 0.25)$ ,  $k_i \sim \mathcal{N}(1, 0.5)$ ) to create heterogeneity across agents.

### 2.4.2 Connectivity matrices $J^a$ , $J^u$ and $J^x$

This leaves only the structure of the graphs to be determined. I set  $J^a = J^u$ , with the assumption that agents are both communicating with and paying attention to the same other agents (e.g. their friends). I set  $J^x$  to be the complete graph, to allow agents to have some perception of the state of the entire social system in which they are embedded, and adjust their integration time accordingly.

For now, I also assume that  $J^a$  is symmetric, which corresponds to the social graph being undirected (an assumption that should be relaxed in future work, to allow for unidirectional flow of information between agents).

I examine the dynamics associated with three different types of social graphs: Erdős-Rényi (ER), Barabási-Albert (BA), and Watts-Strogatz (WS). I set the parameters of each of the three generators such that the average ratio of the number of edges to the number of nodes is equal across all three. ER is a classic random graph model with a short average path length but without hubs or local clusters; BA is based on preferential attachment and has a power law degree distribution and hence hubs; WS has a small-world structure and hence local clusters. While none is a perfectly realistic depiction of social networks, together they cover an interesting range of network structures.

I set  $N = 1000$ , which should be sufficient to generate graphs that are structurally distinct while remaining computationally tractable.

### 2.4.3 Initial conditions

Agents are initialized at the basal susceptibility,  $u_i = 0.01$ , and at a random, almost neutral opinion state  $z_i \sim \mathcal{N}(0, 0.01)$ .

Gating states are initialized at zero.

### 2.4.4 External input patterns

The last thing to determine is the pattern of time and unit varying external input  $I_i(t)$ .

1. In the 1st condition, one third of the agents receive a unique burst of signal of low, medium, or high intensity.
2. In the 2nd condition, one third of the agents receive two successive bursts of signal with a pause in between; each burst is half of the magnitude of the single burst in the 1st condition.
3. In the 3rd condition, one fourth of the agents receive a burst of signal, and another fourth receive the inverse signal. Given the symmetry of negative or positive signals in this design, we test the combinations low-low, medium-medium, high-high, low-medium, medium-high, low-high.
4. The 4th condition is similar, but the second burst is delayed in time.
5. The 5th to 8th conditions mirror the first four, but in addition to the signals, all agents receive Gaussian noise at every step.
6. In the control condition, all agents receive Gaussian noise at every step.

I run each simulation for both the original and the gated models. I first simulate in discrete time using agent-based modeling and Euler's approximation with a step size  $h = 0.1$ , and then in continuous time using differential equations solvers.

The number of simulations is  $2 \times 2 \times 3 \times 37 = 444$  (with and without gating, discrete or continuous time, different social graphs, noise patterns).

Supposing that there is a decision to be made, and that only agents strongly in favor will make it, after they have accumulated sufficient evidence for it, I set a threshold  $c = 1.75$  and record, for each simulation, which agents make the decision, how many they are, and their location on the social graph.

### 3 Results

### 4 Discussion

### 5 Future directions

**Building predictive models of the environment from partial information.** Building predictive models of their environment is a crucial component collective intelligence. Here, I take model building to mean the discovery of structure and regularity in a dynamic environment so that a system can predict its next inputs better than at random; such models can be implicit and behavioral (Crutchfield 1994). In real social networks, the information acquired by agents in order to do so is not only noisy, it is also partial and heterogeneous across agents. Settings of partial and heterogeneous information have rarely been explicitly studied in the network dynamics literature.

The BFL model only examines subjective beliefs, not the ability to converge to correct or useful answers. In the future, I plan to change the task from belief propagation to the collective recovery of a model of the environment from partial and heterogeneous information. A ground truth will exist, and I will examine under what conditions the network is successful at recovering it from noisy, partial and asymmetric inputs, including varying the distribution of information, noise, and other system parameters. This will be formalized in two ways, first as a static matrix reconstruction problem, and then as a dynamic hidden Markov problem.

Ongoing research efforts (Malach 2023) suggest that in some settings, next-token prediction of structured sequences may enable a model to learn complex data-generating processes (DGP) accurately. However, if and how a DGP can be recovered by a network when different nodes each receive noisy and imperfect measurements of the sequences remains unknown.

**Hysteresis as a mechanism for the reproduction of social inequality.** ...

## References

- Abelson, Robert P. (1964). “Mathematical Models of the Distribution of Attitudes under Controversy”. In: *Contributions to mathematical psychology*. URL: <https://cir.nii.ac.jp/crid/1574231875271391104> (visited on 05/02/2024).
- Crutchfield, James P. (Aug. 1, 1994). “The Calculi of Emergence: Computation, Dynamics and Induction”. In: *Physica D: Nonlinear Phenomena* 75.1, pp. 11–54. ISSN: 0167-2789. DOI: 10.1016/0167-2789(94)90273-9. URL: <https://www.sciencedirect.com/science/article/pii/0167278994902739> (visited on 02/19/2024).
- Ravazzi, Chiara and Anton Proskurnikov (2021). “Dynamical Social Networks”. In: *Encyclopedia of Systems and Control*. Ed. by John Baillieul and Tariq Samad. Cham: Springer International Publishing, pp. 647–657. ISBN: 978-3-030-44183-8 978-3-030-44184-5. DOI: 10.1007/978-3-030-44184-5\_100129. URL: [https://link.springer.com/10.1007/978-3-030-44184-5\\_100129](https://link.springer.com/10.1007/978-3-030-44184-5_100129) (visited on 05/02/2024).
- Bizyaeva, Anastasia, Alessio Franci, and Naomi Ehrich Leonard (2022). “Nonlinear Opinion Dynamics With Tunable Sensitivity”. In: *IEEE Transactions on Automatic Control* 68.3, pp. 1415–1430. ISSN: 0018-9286, 1558-2523, 2334-3303. DOI: 10.1109/TAC.2022.3159527. URL: <https://ieeexplore.ieee.org/document/9736598/> (visited on 11/06/2023).

- Krishnamurthy, Kamesh, Tankut Can, and David J. Schwab (Jan. 18, 2022). “Theory of Gating in Recurrent Neural Networks”. In: *Physical Review X* 12.1, p. 011011. DOI: 10.1103/PhysRevX.12.011011. URL: <https://link.aps.org/doi/10.1103/PhysRevX.12.011011> (visited on 05/02/2024).
- Malach, Eran (Sept. 13, 2023). *Auto-Regressive Next-Token Predictors Are Universal Learners*. DOI: 10.48550/arXiv.2309.06979. arXiv: 2309.06979 [cs]. URL: <http://arxiv.org/abs/2309.06979> (visited on 03/31/2024). preprint.
- Bernardo, Carmela et al. (Jan. 1, 2024). “Bounded Confidence Opinion Dynamics: A Survey”. In: *Automatica* 159, p. 111302. ISSN: 0005-1098. DOI: 10.1016/j.automatica.2023.111302. URL: <https://www.sciencedirect.com/science/article/pii/S0005109823004661> (visited on 05/02/2024).
- Leonard, Naomi Ehrlich, Anastasia Bizyaeva, and Alessio Franci (2024). “Fast and Flexible Multi-Agent Decision-Making”. In: *Annual Review of Control, Robotics, and Autonomous Systems* 7.

## A Simulation results

### Contents