SYNCOFLY: An Interactive Platform for Collective Rhythm Synchronization

Ya-Ning Wu(✉), Gregory Hickok, and Jeffrey L. Krichmar

Department of Cognitive Sciences, University of California, Irvine, CA 92697, USA
{yaninw1,gshickok,jkrichma}@uci.edu

Abstract. Rhythmic coordination is central to collective behavior across biological and artificial systems, yet most synchronization models are evaluated at the level of isolated or dyadic agents under stable metronomic conditions. We introduce SYNCOFLY, an open-source, browser-based platform for studying collective rhythmic synchronization among multiple agents under configurable perturbation. SYNCOFLY supports non-isochronous rhythms, configurable predictive and reactive timing architectures, controllable signal-level and structural perturbations, real-time behavioral visualization, and exportable behavioral logs for reproducible analysis. Demonstration simulations show how SYNCOFLY can be used to compare rhythm synchronization architectures under beat omission, noise injection, and rhythm pattern switching. SYNCOFLY aims to provide an open and extensible platform for investigating collective temporal coordination in multi-agent systems.

Keywords: Rhythm synchronization · Multi-agent coordination · Collective timing

1 Introduction

Across biological and artificial systems, coordinated rhythmic behavior supports collective action including but not limited to music performance, conversation, animal swarms, and robotic coordination [7,14]. Robust coordination typically depends on predictive timing rather than reacting to each event in isolation, with individuals integrating information across temporal scales to form internal expectations and maintain synchronization when surface signals are incomplete or irregular [3,10].

Despite this, computational synchronization models are commonly evaluated using stable rhythmic stimuli with minimal environmental perturbation [5,9]. As a result, less is known about how a group's timing strategy shapes its ability to establish, maintain, or recover shared temporal structure when rhythmic signals or group composition change dynamically. A further methodological gap is the absence of a shared experimental environment. Existing open source codes typically implement a single synchronization strategy rather than providing a configurable sandbox in which multiple models, perturbation types, and group

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2027
O. Brock et al. (Eds.): SAB 2026, LNAI 16965, pp. 115–126, 2027.
https://doi.org/10.1007/978-3-032-39953-3_10

compositions can be systematically varied and compared. This makes it difficult to evaluate competing hypotheses under identical conditions or to assess whether a model's behavior generalizes across perturbation types.

Here we introduce SYNCOFLY, an interactive, browser-based platform for studying collective rhythmic synchronization under perturbation. SYNCOFLY lets users configure each agent's timing architecture, coupling weights to the rhythmic source and to peers, the type and magnitude of perturbation applied, and visualize synchronization behavior in real time. The platform includes one example predictive architecture (the Grid Model) and one reactive architecture to showcase how models can be implemented and compared under the same conditions. The main contributions of SYNCOFLY are:

1. An extensible, browser-based platform for configuring, visualizing, and comparing multi-agent rhythm synchronization experiments. It supports configurable rhythmic stimuli, agent architectures, and user-definable perturbations.
2. Demonstration experiments illustrating how the platform can be used to compare predictive and reactive synchronization strategies under shared experimental conditions.
3. An open-source codebase for supporting reproducible experimentation and future model development, available here: [20].

2 Related Work

Computational studies have proposed a range of models for sensorimotor synchronization, including adaptation-and-anticipation mechanisms (ADAM) [5,17] and Bayesian accounts of expectancy-based entrainment [1]. More recent work has extended these approaches through dynamical systems and neural resonance frameworks, including metastable self-other integration [6] and adaptive Hebbian learning with elasticity to account for synchronization across spontaneous motor tempos [15]. Collectively, these studies explain synchronization phenomena and motivate the predictive and reactive control mechanisms implemented as example agents in SYNCOFLY. However, existing models are typically evaluated within fixed paradigms, such as single-agent synchronization to an external rhythm or agent-to-agent entrainment, rather than within configurable multi-agent environments that systematically manipulate each agent's respective synchronization architectures, social influence, and their environmental conditions.

Meanwhile, empirical studies of rhythmic coordination have examined joint musical performance [7], leader-follower finger tapping [12], and piano duets under varying network latency [18]. More recently, larger-scale group experiments have shown that collective interactions can reduce temporal variability and stabilize tempo drift during group drumming [2], providing valuable behavioral benchmarks for computational models. Despite these advances, there lacks a general platform for comparing computational synchronization strategies and characterizing resulting behaviors. Consequently, a gap remains between theoretical synchronization models and observed human behaviors.

To sum up, several past studies support agent-based or neural synchronization modeling, but none integrates the features required for systematic study of collective rhythm or expandable purposes. Existing publicly available code primarily implements the specific architecture used in individual studies, rather than providing a general-purpose experimental platform. These implementations typically support a fixed synchronization model without allowing model swapping, focus on dyadic or small-N interaction without configurable group composition, or lack real-time visualization and perturbation control [1,9]. To our knowledge, no open-source framework currently combines configurable rhythmic stimuli, user definable timing architectures, systematic perturbation control, real-time visualization, and exportable behavioral logs within a unified experimental GUI designed for extensibility. Here, we introduce SYNCOFLY to address this gap.

3 Methods

3.1 Overview

SYNCOFLY is a browser-based experimental sandbox for investigating collective rhythm synchronization (Fig. 1). Each experimental run is first configured by users through the setup panel that controls pattern, agents, and perturbation, and then launched into a real-time canvas. The platform timestamps and uses the Web Audio API for sound generation [19]. Each experiment consists of five agents (B1-B5) synchronizing to a stationary rhythmic source (Source A). Synchronization is evaluated against Source A's beat timing and displayed in real time.

Experimental Setup. Users configure experiments through three setup tabs:

Pattern. Rhythms are specified with a polar clock editor supporting binary (16th-note) and ternary (24th-note) subdivisions. Presets include 4/4, 3/4, and polyrhythmic patterns. Users set cycle period T and a pattern-match tolerance window. When agents produce beats within the tolerance criteria, the goal will be marked as "unlocked" on the real-time canvas.

Agents. Each agent can be configured as predictive (Grid Model enabled) or reactive, with optional noise injection. Users control Source A and peer-coupling weights globally or per agent. Four preset group compositions are provided: Full Predictive, Perimeter-P, Mixed P/R, and Full Reactive.

Perturbations. Supported perturbations include Beat Omission (randomly silenced source beats) and Pattern Switch (A→B→A transitions). Users configure omission probability and epoch durations before launch.

Runtime Interface. The runtime interface (Fig. 1) displays agent movement, beat propagation, and real-time synchronization metrics. Live controls allow adjustment of hearing range and source reach during simulation.

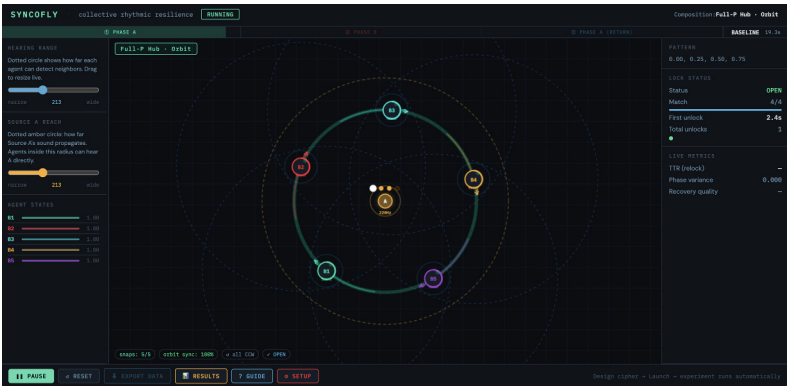


Fig. 1. SYNCOFLY platform interface screenshot. Source A emits a repeating rhythmic pattern. Five agents (B1-B5) orbit around it, stepping at each predicted downbeat. The left panel provides live controls; the right panel displays confidence bars and real-time outcome metrics. The epoch bar at the top tracks experimental phase.

3.2 Agent Architectures

Each agent is individually configured as either a predictive agent (Grid Model enabled) or a reactive agent (Grid Model disabled). All agents share identical signal-reception logic, social-coupling rules, and spatial dynamics, and differ only in how received events are processed internally. The setup aims to demonstrate a complete implementation end to end.

Predictive Agents: Grid Model.

Predictive agents maintain:

- cycle period estimate T,
- cycle anchor t_0,
- volatility estimate v,
- competing binary (L_2) and ternary (L_3) subdivision layers.

The Grid Model represents rhythm as competing metrical hypotheses organized into binary (L_2) and ternary (L_3) subdivision layers. Binary layers partition the cycle into duple groupings of evenly spaced beats (e.g., 2/2- or 4/4-like structure), whereas ternary layers partition the cycle into triple groupings (e.g., 3/4- or 6/8-like structure). These correspond to common forms of metrical organization in human rhythm perception [4]. Each layer defines a candidate temporal grid over the cycle period T. Incoming events are evaluated against both grids, and layer weights increase when events consistently align with a layer's predicted beat positions. Each layer $g \in \{2, 3\}$ generates tick positions:

$$\mathcal{T}_g = \left\{ \frac{i}{g \cdot 2^{r_g - 1}} \middle| i = 0, \ldots, g \cdot 2^{r_g - 1} - 1 \right\}$$

Internal phase is:

$$\phi(t) = \frac{(t - t_0) \bmod T}{T}$$

For each incoming event, the model computes the nearest-grid distance for both layers and updates the best-matching layer weight. Current-resolution matches increase weight by +0.4; resolution upgrades by +0.5. Non-matching layers decay proportionally:

$$w_{\text{other}} \leftarrow \max(0.05,\, w_{\text{other}} - 0.1 - 0.1 w_{\text{other}})$$

Downbeats update the cycle estimate:

$$T \leftarrow \begin{cases} \Delta & \text{if } |\Delta/T - 1| > 0.10 \text{ and not a missed-beat gap} \\ (1-\alpha)T + \alpha\Delta & \text{otherwise, with } \alpha = \min(0.5,\, 0.35 + 0.15\frac{T-0.6}{1.4}) \end{cases}$$

Missed predicted ticks reduce layer weight:

$$w_g \leftarrow \max(0, w_g - 0.3)$$

A layer is considered active when:

$$w_g > 0.5 \quad \text{and} \quad w_g \geq w_{\max} - 0.30$$

Active ticks generate motor emissions with a fixed 25 ms delay.

Reactive Agents.

Reactive agents carry no internal temporal model. Upon receiving an audio event at time t, the agent schedules an immediate tap at $t + \Delta_{\text{motor}}$, where $\Delta_{\text{motor}} = 25$ ms is a fixed motor delay matching empirically observed sensorimotor latency [13]. To prevent cascading echo loops between agents, a minimum inter-tap interval is enforced:

$$\tau_{\min} = \max\Big(80 \text{ ms},\ \alpha_{\text{rl}} \cdot \hat{T}_{\text{ibi}}\Big) \tag{1}$$

where $\hat{T}_{\text{ibi}}$ is a running exponential moving average (EMA) of observed inter-onset intervals (new-event weight 0.3; decay $\beta = 0.7$), and $\alpha\text{rl} = 0.40$ is the reactive rate-limit fraction, and the 80 ms floor prevents the interval from collapsing during startup or transient noise bursts before $\hat{T}_{\text{ibi}}$ has converged. The 0.40 fraction ensures $\tau_{\min}$ remains below the actual inter-beat interval, allowing the agent to respond to beats in the source pattern while suppressing echo amplification. Reactive agents cannot distinguish downbeats from sub-beats. They emit identical response regardless of incoming amplitude, and their visual display reflects this: every tap produces the same colored ring pulse, with no white downbeat indicator.

3.3 Shared Signal Reception Logic

Both predictive and reactive agents share the same signal-reception pipeline before the predictive/reactive process.

Refractory Filtering. Events arriving within:

$$\min(100\text{ ms}, 0.20T_{\text{eff}})$$

from the same source are discarded.

Distance Attenuation. Signal amplitude decays linearly with distance and is discarded beyond hearing range.

Source Weighting. Source A and peer events are scaled by user-defined trust weights.

Peer Conformity Gate. Peer events are evaluated against Source A's local grid estimate:

$$\text{conformity} = \max\left(0, 1 - \frac{\epsilon_\phi}{0.18}\right)$$

Peer influence is additionally boosted by local peer agreement:

$$w_{\text{peer_mod}} = 0.30 \times \text{conformity} \times \text{consensusBoost}$$

When the user explicitly configure a reactive agent to trust peers more than Source A, the reactive agent bypasses the conformity gate. Near-simultaneous peer events (<50 ms) are collapsed into a single update, and predictive peer-drive is rate-limited by:

$$\tau_{\min}^{\text{peer}} = \max(80\text{ ms}, 0.40 \times T/4)$$

3.4 Noisy Agent

Each agent can be individually marked as Noisy via the Agents tab. A Noisy agent receives spurious events on each simulation tick with probability set by the Noise Injection Rate slider (range 0.001–0.07; default 0.035). For predictive Noisy agents, the spurious event is injected directly into the Grid Model at amplitude 0.85 (above the downbeat threshold of 0.80), at a randomized time offset within $\pm 0.4 \cdot T$ of the current frame, potentially displacing the cycle anchor. For reactive Noisy agents, the spurious event is queued as an off-beat tap with no model to disturb.

3.5 Perturbations

SYNCOFLY supports perturbation categories that act at qualitatively different levels of the signal-structure hierarchy.

3.5.1 Beat Omission

During the shock epoch, each beat emitted by Source A is independently silenced with probability $p_{\text{miss}} \in [0, 1]$ (user-configurable, 0–100%). Silenced beats are never transmitted to agents; a red dropout ring is displayed on Source A when a beat is dropped.

3.5.2 Pattern Switch

The pattern switch replaces Source A's pattern mid-experiment with a user-defined alternate pattern (Pattern B), then reverts to Pattern A. The experiment follows an A→B→A structure, with independently configurable phase durations. Pattern B can be configured using an identical polar clock editor as Pattern A.

3.6 Orbit Task

Source A sits at the canvas center. Each agent orbits at a fixed canvas-relative radius $r_{\text{orbit}} = r_{\text{base}}$, advancing by a fixed angular arc on each step. Predictive agents step once per predicted downbeat (one large step per cycle); reactive agents step once per emitted beat, with the per-beat arc scaled by $\hat{T}_{\text{ibi}}/T$ so that the net per-cycle arc matches a predictive agent's. Orbit direction is coordinated through local confidence-weighted voting and collision-based confidence exchange between nearby agents.

3.7 Outcome Metrics

SYNCOFLY reports synchronization and recovery metrics at both agent and group levels. Pre-perturbation metrics include initial synchronization time and group lock success. Perturbation metrics include retention rate, corruption type, and relock time. Recovery metrics include per-agent recovery latency, group recovery time, recovery rate, and phase variance.

Five CSV file exports are available: (1) Time-Series Log: captures agent-level states including coordinates, circular phase, and internal confidence levels. (2) Epoch Summary Table: aggregate metrics (mean sync time, retention rate, recovery rate) for each experimental phase. (3) Beat-Event Log: records every tap emitted by Source A and all agents, including timestamps and downbeat markers. (4) Parameter Snapshot: a snapshot of every setup parameter (e.g., initial Source A weights, period length) at the moment of launch. (5) Parameter Change Ledger: records every manual adjustment to weights or parameters made by the user during the run.

4 Results

This section presents three example configurations implemented in SYNCOFLY to show how the platform can be used to observe and probe synchronization behaviors under perturbation. We evaluated SYNCOFLY under three perturbation conditions: (1) beat omission under a non-isochronous rhythm, (2) noise injection into a predictive agent, and (3) structural meter switching between ternary and binary patterns. These experiments are intended as demonstrations of the platform's functionality rather than evaluations of a synchronization model. Each configuration was run ten times, with their first runs visualized in Fig. 2 for illustrative purposes.

Across all experiments, group composition remained fixed. B1, B3, and B4 were predictive agents ($w_{\text{src}} = 1.0$, $w_{\text{peer}} = 0.5$), while B2 and B5 were reactive agents with different weighting profiles. B2 was configured as peer-dominant ($w_{\text{src}} = 0.10$, $w_{\text{peer}} = 1.50$), whereas B5 was source-dominant ($w_{\text{src}} = 1.00$, $w_{\text{peer}} = 0.50$). The weights determine how strongly agents respond to Source A versus neighboring agents. Cycle period was fixed at $T = 4\,\text{s}$.

In general, predictive agents maintained stable synchronization under perturbation, whereas reactive agents depended more heavily on surface input. Strong peer coupling enabled reactive agents to inherit predictive structure but also increased susceptibility to noise. Predictive agents showed higher extraneous rates because they structurally inferred beats which may occur without corresponding surface events. Per-agent accuracy rates are reported in Table 1.

4.1 Experiment 1: Beat Omission Under a Non-Isochronous Rhythm

Setup. Source A emitted a 6/8 pattern with the sixth subdivision silent, yielding a five-onset cycle. The run consisted of baseline, beat omission perturbation ($p_{\text{miss}} = 0.30$), and recovery epochs.

Results. Predictive agents maintained the full inferred 6-beat grid throughout most phases despite missing surface events (Fig. 2, Top). During perturbation, Source A output degraded to 2–5 beats per cycle, which B5 mirrored directly. In contrast, peer-dominant reactive agent B2 converged toward the predictive 6-beat grid, indicating socially inherited structural stabilization from its predictive peers.

4.2 Experiment 2: Peer-Inherited Structure Under a Noisy Predictive Neighbor

Setup. Source A emitted a 4/4 pattern with the fourth subdivision permanently silent. B1 was configured as a noisy predictive agent with phantom-event injection (0.02 probability per simulation frame). No external perturbation was applied.

Results. Predictive agents (B3, B4) maintained the inferred 4-beat grid including the silent subdivision (Fig. 2, Middle). Noisy predictive agent B1 produced dense off-grid emissions due to repeated disruption. Source-dominant reactive agent B5 mirrored Source A directly. Meanwhile, peer-dominant reactive agent B2 inherited the structurally inferred fourth beat but also propagated off-grid noise from B1. These results demonstrate that social coupling can transmit both stabilization and instability.

4.3 Experiment 3: Structural Meter Switch

Setup. Source A alternated between a binary 8/8 pattern and a ternary 6/8 pattern, each containing one structurally silent subdivision. The experiment tested whether predictive agents could adapt to changed patterns in real time.

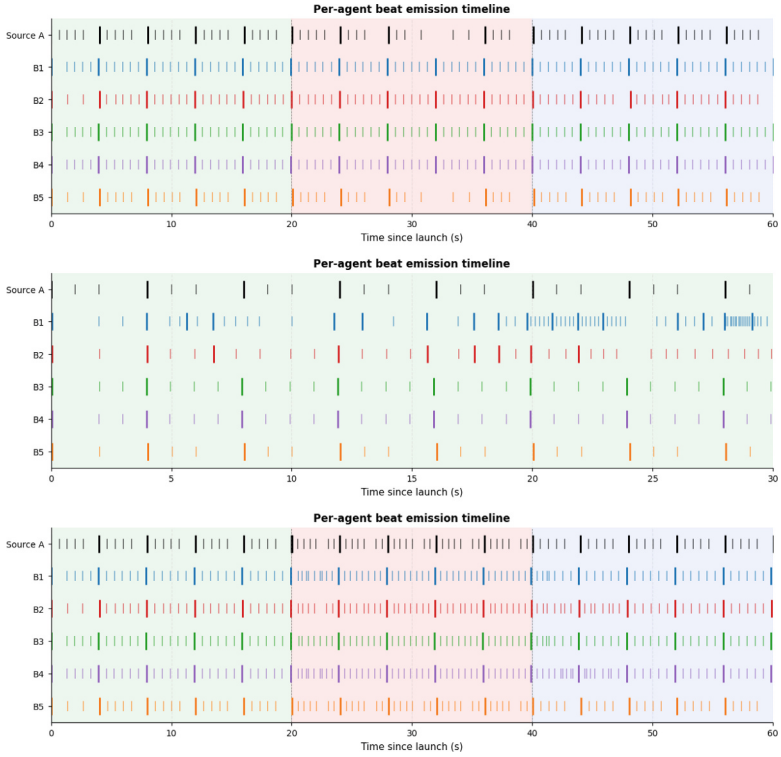


Fig. 2. Experiment 1–3 (panel ordered from top to down): *Experiment 1*: Beat omission. Vertical ticks indicate emitted beats; bold ticks mark downbeats. Shaded regions denote baseline (green), beat omission perturbation (red), and recovery (blue). Source A emits a 6/8 pattern with the final subdivision silent. Predictive agents predominantly maintain the full 6-beat grid, indicating successful inference of the underlying structure despite missing surface events. Source-dominant reactive agent B5 mirrors the degraded surface pattern, whereas peer-dominant reactive agent B2 converges toward the predictive 6-beat grid through social coupling. *Experiment 2*: Noise injection. Source A emits a 4/4 pattern with the fourth subdivision permanently silent. B1 is configured as a noisy predictive agent, producing dense off-grid emissions from phantom-event injections. Predictive agents B3 and B4 maintain a stable 4-beat grid including the silent slot. Source-dominant reactive agent B5 mirrors the surface pattern, whereas peer-dominant reactive agent B2 inherits both the structurally inferred fourth beat and off-grid noise from B1. *Experiment 3:* Structural pattern switch. Vertical ticks indicate emitted beats; bold ticks mark downbeats. Shaded regions denote baseline 6/8 pattern (green), perturbation switch to 8/8 (red), and recovery to 6/8 (blue). Both patterns contain one structurally silent subdivision. Predictive agents infer and emit the silent beat positions across patterns and successfully adapt to the pattern switch. Source-dominant reactive agent B5 primarily mirrors the surface pattern, whereas peer-dominant reactive agent B2 inherits structurally inferred beats from neighboring predictive agents.

Results. Predictive agents adapted to the meter switch within a cycle while continue emitting structurally inferred silent beats across both patterns (Fig. 2, Bottom). Source-dominant reactive agent B5 primarily mirrored the surface pattern and never emitted silent-slot beats. Peer-dominant reactive agent B2 inherited structurally inferred beats from predictive agents across both meter patterns.

Table 1. Per-agent beat accuracy across experiments. Values are reported as mean ± standard deviation across 10 independent runs for each experiment. On-target indicates when the beat fell within a ±0.15 cycle-fraction matching window (±600 ms; $T = 4.0$ s). Omissions rate indicate Source A beats without an agent response. Extraneous beats indicate agent emissions without a nearby Source A beat. When multiple emissions fall within the same matching window, only the closest emission is counted as on-target and the remaining emissions are categorized as extraneous.

			On-target (%)		Omissions (%)		Extraneous (%)	
Exp.	Agent	Model	Mean	SD	Mean	SD	Mean	SD
1	B1	grid	97.0	2.6	3.0	2.6	24.6	1.5
	B2	reactive	97.1	0.1	2.9	0.1	11.7	6.1
	B3	grid	96.4	2.3	3.6	2.3	24.5	1.4
	B4	grid	95.8	3.4	4.2	3.4	25.1	2.2
	B5	reactive	96.9	0.5	3.1	0.5	0.0	0.0
2	B1	grid	84.3	11.4	15.7	11.4	72.7	12.5
	B2	reactive	95.2	1.4	4.8	1.4	36.2	4.8
	B3	grid	94.8	1.8	5.2	1.8	26.3	0.7
	B4	grid	93.9	2.2	6.1	2.2	26.0	0.9
	B5	reactive	95.7	0.0	4.3	0.0	0.0	0.0
3	B1	grid	97.7	2.9	2.3	2.9	25.1	13.4
	B2	reactive	97.5	0.3	2.5	0.3	14.7	7.4
	B3	grid	97.1	2.9	2.9	2.9	28.3	17.5
	B4	grid	96.2	3.8	3.8	3.8	33.2	20.0
	B5	reactive	97.4	0.7	2.6	0.7	0.0	0.0

5 Discussion and Conclusions

SYNCOFLY is a configurable platform for investigating how temporal structure, environmental perturbations, and social interaction jointly shape collective synchronization under conditions that more closely resemble naturalistic rhythm entrainment than traditional metronomic paradigms. Through its interactive interface, researchers can configure agent and perturbation parameters, compare competing models visually without building simulation infrastructure from scratch.

The predictive architecture in SYNCOFLY, the Grid Model, demonstrates the types of hypotheses that SYNCOFLY can instantiate and evaluate. It highlights that robust synchronization can arise from selective, computationally lightweight representations updated primarily at high-confidence temporal anchors such as downbeats. This sparse-updating design allows predictive agents to sustain synchronization across missing or perturbed signals while remaining less sensitive to transient variability, consistent with evidence that rhythmic perception relies more strongly on structurally significant positions than on dense event-by-event tracking [4,11].

The demonstration experiments suggest several research directions now accessible through SYNCOFLY. Reactive agents reproduce surface timing effectively when reliable cues are available, but lack mechanisms for representing higher-order rhythmic structure. Such limitation becomes apparent for non-isochronous rhythm entrainment, where successive inter-onset intervals are not individually predictive to each other [8,16]. Meanwhile, social coupling can allow reactive agents to benefit from predictive neighbors, but also propagates disruption from noisy ones. This suggests that effectiveness of peer coupling depends on the balance between structured and unreliable information within the group. Future work could explore hybrid agents that dynamically balance predictive and reactive control, as well as adaptive coupling mechanisms under varying environmental conditions.

With SYNCOFLY lowering the technical barrier to evaluating and visualizing computational theories of synchronization, we hope its accessibility will facilitate more systematic and reproducible investigations of rhythmic synchronization across computational neuroscience, robotics, and related disciplines.

References

1. Cannon, J.: Expectancy-based rhythmic entrainment as continuous Bayesian inference. PLoS Comput. Biol. **17**(6), e1009025 (2021). https://doi.org/10.1371/journal.pcbi.1009025
2. Dotov, D., Delasanta, L., Cameron, D.J., Large, E.W., Trainor, L.: Collective dynamics support group drumming, reduce variability, and stabilize tempo drift. eLife **11**, e74816 (2022). https://doi.org/10.7554/eLife.74816
3. Fitch, W.T., Rosenfeld, A.J.: Perception and production of syncopated rhythms. Music Percept. Interdis. J. **25**(1), 43–58 (2007). https://doi.org/10.1525/mp.2007.25.1.43
4. Fujioka, T., Zendel, B.R., Ross, B.: Endogenous neuromagnetic activity for mental hierarchy of timing. J. Neurosci. **30**(9), 3458–3466 (2010). https://doi.org/10.1523/JNEUROSCI.3086-09.2010
5. Harry, B., Keller, P.E.: Tutorial and simulations with ADAM: an adaptation and anticipation model of sensorimotor synchronization. Biol. Cybern. **113**(4), 397–421 (2019). https://doi.org/10.1007/s00422-019-00798-6
6. Heggli, O.A., Konvalinka, I., Kringelbach, M.L., Vuust, P.: A metastable attractor model of self-other integration (MEAMSO) in rhythmic synchronization. Philos. Trans. Royal Soc. B Biol. Sci. **376**(1835), 20200332 (2021). https://doi.org/10.1098/rstb.2020.0332

7. Keller, P.E., Novembre, G., Hove, M.J.: Rhythm in joint action: psychological and neurophysiological mechanisms for real-time interpersonal coordination. Philos. Trans. Royal Soc. B Biol. Sci. **369**(1658), 20130394 (2014). https://doi.org/10.1098/rstb.2013.0394
8. Large, E.W., Jones, M.R.: The dynamics of attending: how people track time-varying events. Psychol. Rev. **106**(1), 119–159 (1999). https://doi.org/10.1037/0033-295X.106.1.119
9. Large, E.W., et al.: Dynamic models for musical rhythm perception and coordination. Front. Comput. Neurosci. **17**, 1151895 (2023). https://doi.org/10.3389/fncom.2023.1151895
10. Matthews, T.E., Witek, M.A.G., Lund, T., Vuust, P., Penhune, V.B.: The sensation of groove engages motor and reward networks. Neuroimage **214**, 116768 (2020). https://doi.org/10.1016/j.neuroimage.2020.116768
11. Nozaradan, S., Peretz, I., Missal, M., Mouraux, A.: Tagging the neuronal entrainment to beat and meter. J. Neurosci. **31**(28), 10234–10240 (2011). https://doi.org/10.1523/JNEUROSCI.0411-11.2011
12. Ogata, T., Katayama, T., Ota, J.: Cross-feedback with partner contributes to performance accuracy in finger-tapping rhythm synchronization between one leader and two followers. Sci. Rep. **9**, 7800 (2019). https://doi.org/10.1038/s41598-019-43352-x
13. Repp, B.H.: Sensorimotor synchronization: a review of the tapping literature. Psychon. Bull. Rev. **12**(6), 969–992 (2005). https://doi.org/10.3758/BF03206433
14. Repp, B.H., Su, Y.-H.: Sensorimotor synchronization: a review of recent research (2006–2012). Psychon. Bull. Rev. **20**(3), 403–452 (2013). https://doi.org/10.3758/s13423-012-0371-2
15. Roman, I.R., Roman, A.S., Kim, J.C., Large, E.W.: Hebbian learning with elasticity explains how the spontaneous motor tempo affects music performance synchronization. PLoS Comput. Biol. **19**(6), e1011154 (2023). https://doi.org/10.1371/journal.pcbi.1011154
16. Schreiber, H., Urbano, J., Müller, M.: Music tempo estimation: are we done yet? Trans. Int. Soc. Music Inf. Retrieval **3**(1), 111–125 (2020). https://doi.org/10.5334/tismir.43
17. van der Steen, M.C.M., Keller, P.E.: The adaptation and anticipation model (ADAM) of sensorimotor synchronization. Front. Hum. Neuroscience **7**, 253 (2013https://doi.org/10.3389/fnhum.2013.00253. https://doi.org/10.3389/fnhum.2013.00253
18. Washburn, A., Wright, M.J., Chafe, C., Fujioka, T.: Temporal coordination in piano duet networked music performance (NMP): interactions between acoustic transmission latency and musical role asymmetries. Front. Psychol. **12**, 707090 (2021). https://doi.org/10.3389/fpsyg.2021.707090
19. World Wide Web Consortium (W3C): Web audio API (2021). https://www.w3.org/TR/webaudio/. Accessed 07 May 2026
20. Wu, Y.N.: Syncofly (2026). https://github.com/yanni-yanni/syncofly