Summer 8-8-2023

Emergent Collective Dynamics with Applications in Bridge Engineering and Social Networks

Kevin Daley

EMERGENT COLLECTIVE DYNAMICS WITH APPLICATIONS IN BRIDGE ENGINEERING AND SOCIAL NETWORKS

by

KEVIN DALEY

Under the Direction of Igor Belykh, Ph.D.

A Dissertation Submitted in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

in the College of Arts and Sciences

Georgia State University

2023

ABSTRACT

This thesis presents several novel results on the nonlinear and emergent collective dynamics of crowds and populations in complex systems. Though, historically, the list of suspension bridges destabilized by pedestrian collective motion is long, the phenomenon still needs to be fully understood, especially regarding the effect of human-to-human interactions on the structure, and often incorrectly explained using synchronization theory. We present a simple general formula that quantifies the effect of pedestrian effective damping of a suspension bridge and illustrate it by simulating three mathematical models, including one with a strong propensity for synchronization. Despite the subtle effects of gait strategies in determining precise instability thresholds, our results show that average negative damping is always the trigger of pedestrian-induced high-amplitude lateral vibration of suspension bridges. Furthermore, we show that human-to-human interactions of heterogeneous pedestrians can trigger the instability of a bridge more effectively than crowds of identical pedestrians. We will also discuss the role of crowd heterogeneity in possible phase pulling between pedestrians and bridge motion. We also develop a model for the evolution of toxic memes on 4chan and report a significant influence on Twitter's anti-vaccine conspiracy discourse over a nine-year period. We show that 4chan topics evolve according to an emergent process mathematically similar to classic reinforcement learning methods, tending to maximize the expected toxicity of future discourse. We demonstrate that these topics can invade Twitter and persist in an endemic state corresponding to the associated spreading rate and initial distribution of post rates and coexisting with a higher-traffic regime of dynamics. We discuss the implication of

this result for preventing large-scale disinformation campaigns.

INDEX WORDS: Dynamical Systems, Synchronization, Stability, Instability, Structural Engineering, Complex Networks, Crowds, Behavioral Science, Social Media, Stochasticity

EMERGENT COLLECTIVE DYNAMICS WITH APPLICATIONS IN BRIDGE ENGINEERING AND SOCIAL NETWORKS

by

Kevin Michael Daley

Committee Chair:	Igor Belykh
Committee:	Michael Stewart
	Yaroslav Molkov
	Vladimir Bondarenko

Electronic Version Approved:

Office of Graduate Studies

College of Arts and Sciences

Georgia State University

August 2023

DEDICATION

For Paul.

ACKNOWLEDGMENTS

I would first like to express my sincerest gratitude to my advisor. Since my childhood, I always knew I wanted to be a Mathematician; it was Igor Belykh, more than anyone else, who understood that I could do it and taught me exactly how. His dedication, not only to Mathematics but to his students and their success, is rare today, and I can not imagine having arrived here without his help.

I also thank the Mathematics and Statistics department faculty and the dynamics community at Georgia State, who suffered my terrible handwriting for almost a decade. I extend special thanks to the dissertation committee— Vladimir Bondarenko, Yaroslav Molkov, and Michael Stewart— for their endurance and helpful feedback.

I thank my parents, who always trusted I was onto something even if they didn't understand any of it, and who helped and loved me every step of the way.

I thank the whole Belykh lab and all my friends for showing up to support me (even if they couldn't make it to the defense) and Kevin Slote, in particular, for helpful advice and collaboration. I thank our many collaborators on the works included in this dissertation—the late John Macdonald, Alan Champneys, Allan McRobie, Mateusz Bocian, and Maurizio Porfiri—for their obvious and significant contributions.

Finally, and most significantly, I thank the Lord Jesus Christ, without Whom, none of what we do here would be possible or have any significance.

This work was supported by the National Science Foundation through Grants DMS-1909924 and CMMI-1953135.

TABLE OF CONTENTS

LIST OF TABLES . . . . .

LIST OF FIGURES . . . . .

1 Introduction **1**

1.1 Collective Behavior in Complex Systems 1

2 Emergence of the London Millennium Bridge instability without synchronization **4**

2.0.0.1 Results. 6

2.1 Methods 18

2.1.1 Mathematical model 18

2.1.1.1 Model 1: Linearized inverted pendulum with step width control. 20

2.1.1.2 Model 2: Model 1 with step-timing adaptation. 21

2.1.1.3 Model 3: Rocking inverted pendulum. 22

2.1.1.4 Asymptotic derivation of negative damping criterion. 23

2.1.1.5 Model implementation. 30

2.2 Review of observational and experimental evidence 31

2.3 Video evidence of pedestrian-bridge interaction in Nepal 33

2.4 Numerical simulation algorithms 35

2.4.1 Procedure for adding pedestrians on the bridge 35

2.4.2 Implementation of negative damping criterion 37

2.4.2.1 To compute σ_1: 37

2.4.2.2 To compute σ_2. 39

2.4.2.3 To compute σ_3. 40

2.4.3 Calculations of the scatter plots and analytical curves in Fig. 2.4. 41

2.5 Further simulation results 44

2.5.1 Faster addition of pedestrians to the bridge 44

2.5.2 Extreme worst-case, complete resonance 46

2.6 Role of different foot placement strategies: negative vs. positive damping 46

3 Crowd Heterogeneity-Induced Instabilities of Footbridges 51

3.1 Introduction 51

3.2 Methods 51

3.3 Results 53

4 A data-driven model of invasive social pathogens from 4chan 60

4.1 Introduction 60

4.2 Data and Methods 62

4.2.1 Model and Validation 63

4.2.2 Analysis of metastable and stationary states 65

4.3 Optimality of the effect of "trolling" 71

4.4 Discussions 75

4.5 Conclusions 78

APPENDICES 80

A Appendix for Chapters 2 and 3 80

A Python source code for Chapter 2 simulations and damping coefficient calculations 80

B Julia source code listing: pedestrian/bridge and social force simulations 104

B Appendix for Chapter 4 121

A Python source code listing: toxicity detection and reinforcement learning 121

REFERENCES 141

LIST OF TABLES

Table 2.1 Default parameter values used in the simulations. Here, S.D. is the standard deviation of parameter mismatch among pedestrians, which follows a normal distribution in all cases. 22

Table 2.2 Reported cases of lateral bridge instability due to the action of walking pedestrians. The final column documents any evidence presented for pedestrian synchronization. 34

Table 2.3 Other reported instances of lateral pedestrian-induced bridge vibrations 35

LIST OF FIGURES

Figure 2.1 Explaining the fundamental mechanism underlying the negative damping 8

Figure 2.2 Outline of the mathematical model of pedestrian-induced lateral instability. (Left): Simulations are run for a coupled bridge-pedestrians system with pedestrians added sequentially at fixed time increments T_{add} apart. The addition of the nth pedestrian ($n = N_{\text{crit}}$) causes the overall damping coefficient to become negative hence the amplitude of motion to increase rather than diminish. (Right). Inverted pendulum model of bridge mode and pedestrian lateral motion. Here y is the lateral position of the pedestrian's center of mass, while p defines the lateral position of the center of pressure (CoP) of the foot, both relative to the bridge. L is the equivalent inverted pendulum length and m is the pedestrian mass. The displacement x of the bridge in a lateral vibration mode is represented by an equivalent platform with mass M, spring constant K, and damping coefficient C. $\tilde{H}$ is the lateral component of the pedestrian's foot force on the bridge deck. In return, the bridge motion causes an inertia force $-m\ddot{x}$ on the pedestrian's center of mass. 10

Figure 2.3 Example simulations showing the nature of the bridge instability for each of our three models. See Methods for model details and parameter values. (Top row): Bridge vibration amplitude as a function of the number of pedestrians N. The left-hand boundary of the pink shaded portion indicates the value N_{crit} where c_T crosses zero, and the blue-shaded portion is where a degree of synchrony is observed. Insets show illustrative bridge $x(t)$ (black) and a few representative pedestrian $y(t) - p(t)$ (colored) oscillations over three cycles. (Middle row): Computation of the total bridge damping c_T given by Eq. (2.1) and the Kuramoto order parameter r Eq. (2.3) calculated for the phases of pedestrians' CoP (Models 1,2) and CoM (Model 3). (Bottom row): instantaneous computed bridge and pedestrian foot placement frequencies. . 12

Figure 2.4 Average damping coefficient per pedestrian $\overline{\sigma}_1$ (top row) and the critical crowd size N_{crit} (bottom row) as a function of numerically calculated bridge and pedestrian frequencies ratio $[\Omega/\overline{\omega}]$. Simulations of Models 2 and 3 indicate the range of frequency ratio $[\Omega/\overline{\omega}]$ in which $\overline{\sigma}_1$ is negative so that a single pedestrian, on average, contributes to bridge instability. Each ratio of $[\Omega/\overline{\omega}]$ corresponds to different combinations of Ω and $\overline{\omega}$ (blue dots). Black dotted lines indicate the average of $\overline{\sigma}_1$ and N_{crit} for a given ratio. The red curve is the analytical expression (2.35) for $\overline{\sigma}_1$ (top plot) and analytical estimate (2.36) for N_{crit} (bottom plot), given in a subsequent section and calculated for Model 1 with identical pedestrians with fixed $\omega = 5.655$ and $S.D. = 0$. The pink dot corresponds to the initial ratio $[\Omega/\overline{\omega}]$ used in Fig. 2.3, the yellow dot corresponds to $\Omega/\overline{\omega} = 1$. 16

Figure 2.5 Snapshots from videos of trekkers crossing unstable bridges over the Dudh Kosi River in Nepal. The full videos are available in the repository at `https://doi.org/10.5281/zenodo.8132826` (84). A trekker places the feet in a highly irregular manner in response to lateral bridge movement (video 2). Notice the right foot traveling beyond the midline of the body; this leg crossing is an uncommon behavior, yet, it is accounted for in Models 1 and 2. (b). A trekker with a heavy backpack struggles to move forward on a laterally unstable bridge and makes significant foot placement adjustments to maintain his balance (video 3). (c). A group of trekkers walking across the same unstable bridge as in (b), with no visible signs of phase-locking in their stepping behavior (video 4). 36

Figure 2.6 Simulations as in Fig. 2.3 in which pedestrians are sequentially added at shorter intervals of $T_{\text{add}} = 10$ s. Notice the widening of the pink region corresponding to the onset of bridge oscillations without pedestrian phase locking. The plots for the damping per pedestrian terms $\sigma_{1,2,3}$ specify the contribution of each term to the onset of bridge instability. Other parameters are as in Fig. 2.3. 45

Figure 2.7 Simulations as in Fig. 2.3 but for the worst-case scenario of pedestrians with the same $\omega = 5.655$ rad/s ($S.D. = 0$) and the perfect resonance ratio $\Omega/\omega = 1$ corresponding to the yellow point in Fig. 2.4. Other parameters are as in Fig. 2.3 47

Figure 2.8 Upper figures show foot placement patterns (short black lines left foot, short blue lines right foot) for Model 1. The first is for a stationary platform, whilst the second and third are for a bridge oscillating at 6 mm amplitude at 0.4 Hz, with walkers adopting Hof's balance laws based on relative and absolute velocity respectively. The bridge motions induce quasiperiodic placement patterns. (The walker's center of mass and the bridge displacements are shown in red and green respectively.) The lower figures show the corresponding forces applied to the bridge. 48

Figure 2.9 Upper: the change in forces that are the result of the bridge motions for the walkers of Fig. 2.8. The bridge velocity is shown in red. Lower: the correlation between the bridge velocity and the induced forces. The red lines indicate the average effective damping coefficient $\overline{\sigma}_1$. 49

Figure 3.1 Dynamic social forcing term F_{dyn} on the pedestrian i from pedestrian j as a function of lateral (horizontal axis) and sagittal (vertical axis) displacement of the jth pedestrian from the ith one. The effect of the ith pedestrian's visual field is determined by the parameter λ. Here, as in (86), $A_d = 1.7, r = 0.31, B_d = 0.28, \alpha = 0.5, B_s = 0.1, A_s = 5$, and $\lambda = 0.31$; these parameters are used throughout this chapter. 54

Figure 3.2 (a) Bridge displacement amplitude distributions and (b) distribution of the Kuramoto order parameter r of the phase $\frac{t-t_s}{t_{s+1}-t_s}$ of each pedestrian step as a function of number of pedestrians N, each over ten trials. The initial stride frequency standard deviation $\sigma = 0.3$ and all other initial crowd parameters are identical to those used in Fig. 2.3 except that the simulations were reinitialized for each N. Note that while the instability of the bridge and the resulting phase coherence are now highly dependent on the exact initial configuration of pedestrian positions and velocities due to the crowd forcing, the bridge instability may occur at values of N as low as $N = 140$ and lead to phase coherence at values as low as $N = 150$, which was not observed previously from an initially still bridge. Furthermore, (c) depicts the bridge amplitude distribution and (d) the distribution of Kuramoto order parameters as a function of σ when $N = 160$. Note the significant increase in mean and maximum order and amplitude as a function of the heterogeneity of the crowd initial frequency even when compared to the case of pedestrians with identical initial stride frequencies (i.e., when $\sigma = 0$), in which case the phase coherence is significant but not as pronounced as when $\sigma = 0.4$, and high-amplitude bridge movement is not observed. 56

Figure 3.3 The general relationship described in Fig. 3.2a-d using a surface plot of the mean bridge displacement amplitude A_x (over each of ten sample initial conditions) as a function of σ, the standard deviation of the initial stride frequency distribution, and N, the number of pedestrians on the bridge. . . . 57

Figure 3.4 (a). Sample evolution of the distribution of bridge-to-pedestrian-stride frequency ratios Ω/ω sampled over time (vertical axis) using Gaussian kernel density estimation; blue dashed vertical lines correspond to roots and red dashed vertical line to the minimum of the curve in Fig. 2.4a. A shift in the frequency distribution occurs close to the instability due to the change in crowd velocity. (b). The corresponding bridge motion x. Parameters are $N = 155$, $\sigma = 0.5$. 58

Figure 3.5 The trails of pedestrian motion are plotted in color according to sagittal velocity, with dark blue corresponding to no forward motion and bright pink being the maximum forward velocity (3.2 m/s), each for a duration of 1 second from (a) 0-1 seconds, (b) 1-2 seconds, and (c) 2-3 seconds. Note the formation of several "traffic jams" in the sagittal direction (horizontal axis). 59

Figure 4.1 Comparison of limiting probability density histograms derived from the sample paths of (a) the model and (b) the time series data under the assumption of near-stationarity of the noise. (c). The plot of the corresponding probability masses for each binned value of ρ sampled from the time-series (vertical axis) vs. from the sample paths of the SDE (horizontal axis). Our model reproduces the distribution of ρ almost exactly. Note that the mean is neither 0 nor 1 but corresponds to the existence of a small subpopulation of conspiracy-believing users that represent an endemic state of the system. 66

Figure 4.2 Region for which the moment-stability conditions for the existence and local stability of stationary states are satisfied as a function of the mean and variance of ρ (vertical axis and rightmost horizontal axis) and (a) β (the spreading rate) and (b) κ (the coupling strength); all parameters other than those on the axes were those estimated from the model. The color of each surface represents the value on the z-axis and is used to orient the figure visually in three dimensions. 71

Figure 4.3 DistilBERT embeddings of a sample of 300,000 posts, projected onto two dimensions using the UMAP algorithm (113) and colored by topic (that is, by K-means cluster obtained from the unprojected embeddings). The visualization shows that separate topics are well separated by the projected embedding, even though much of the semantic information is lost. 73

Figure 4.4 (a). Log-probability associated with each detected topic in the 4chan dataset for 300,000 posts plotted as learned TD-value (expected discounted future toxicity of the discourse). The black dashed curve indicates the logarithm of a histogram-estimated PDF $P_{max}(V)$ determined from a random sample of TD values V. The analytic (red) curve indicates the value of $\hat{P}$. (b). Probability density curves P_{max} and $\hat{P}$ and sample topic frequencies (blue scatter plot points) corresponding to the values in the log-scale plot in (a). (c). Temporal difference value time-series plot corresponding to the interval of posts used in (a) and (b). Note that the expected TD value increases over time, as is typical of a reinforcement learning algorithm. (d). Equivalent scatterplot to (b) in the event that the order of the time series is permuted 100 times and the trial maximizing the expected TD-value is taken. The correlation is destroyed by the time-reordering. 76

CHAPTER 1
Introduction

1.1 Collective Behavior in Complex Systems

In 1908, in one of the founding works of sociology, Giddings (1) coined the term "collective behavior" to describe the kind of emergent human order that is not explained well by institutional norms, by conscience, or by rational decision-making, but instead by the emergence of self-organizing patterns or trends from the most basic interactions occurring on an individual level, which could be fully quantified only using a multi-disciplinary and scientific approach. This collective behavior is also widely observed in bacterial (2; 3), animal (4; 5; 6), political (7), and economic (8; 9) systems.

Mathematical models of this type of effect frequently, though not always, take the form of a continuous dynamical system on a complex network. Of these, we are particularly interested in those that apply nonlinear dynamical modeling to problems of emergent interaction of populations, thereby determining precisely what about their initial organization most influences their collective behavior. In particular, we study several questions in civil engineering and social science, which exhibit various modes of collective behavior in which simple changes to the initial distribution of population-level parameters lead to the coexistence of multiple dynamical regimes.

In Chapters 2 and 3, we study the problem of crowds of pedestrians on suspension bridges causing high-amplitude lateral vibrations due to the phase dynamics of their walking. It is demonstrated in Chapter 2, using a biophysical model of human lateral gait, that the

primary means of such instability is the independent interaction between the pedestrians and the structure; yet, in Chapter 3, we introduce the experimentally-informed model of human-human interaction and obstacle avoidance to the crowd dynamics and show that the instability is amplified when the crowd's movement speed is less coherent. In Chapter 4, we shift our attention to the data-driven study of social networks and show that the effect of small social networks prone to political extremism and conspiracy theory belief on mainstream networks that attempt to self-moderate is often quite large and mathematically difficult to mitigate. Informed by the literature on social contagion, we propose and fit a data-driven stochastic epidemic model to understand the different dynamical regimes of this invasive phenomenon in anti-vaccine conspiracy theory content collected from Twitter and 4chan. We also discover a self-organizing effect like natural selection, which increases the toxicity of 4chan posts over time, and discuss how a large-scale disinformation campaign could exploit it in an automated manner.

These chapters represent only a sample of the author's contributions to the scientific literature during the degree program, albeit very substantial. In all cases, the author of this dissertation was the primary contributor, having produced all experimental and numerical results and figures in Chapter 2 (except for Tables 1-2 and the simulations of Figures 5-6), conceived of the heterogeneity-related study in Chapter 3, and produced all final numerical results and figures based on technical input and literature suggestions from several co-authors; and conceived of and carried out the study in Chapter 4 with complete independence, deriving all results with occasional advice from a few collaborators. The results

contained in this thesis have been published in three journal articles (10; 11; 12), and two articles (one for Chapter 3 and one for Chapter 4) are at the final stage of preparation for submission. Additionally, my research in the Belykh lab resulted in three published journal articles (13; 14; 15).

CHAPTER 2

Emergence of the London Millennium Bridge instability without synchronization

Synchronization of coupled near-identical oscillators leads to emergent order in both natural and engineered complex systems (16; 17; 18; 19; 20; 21; 22; 23). The pedestrian-induced instability on the opening day of the London Millennium Bridge (24) is often used as the canonical example; a threshold number of walkers enabled them to synchronize their footsteps with each other at a bridge's natural vibration frequency (25). In this work, we dispel the synchronization myth and show that any synchronization of pedestrians' foot placement is a consequence of, not a cause of the instability, a result consistent with observations on over 28 bridges. Instead, uncorrelated pedestrians produce negative lateral damping on average to initiate significant bridge vibration over a range of bridge natural frequencies. We present a simple formula that quantifies the effective total negative damping per pedestrian and the contributions towards it from three distinct effects. We also show how this effect predicts the critical number of pedestrians in three distinct simulation models, including one that has a strong propensity for synchronization (26). The models also point to an almost universal frequency dependence of the instability criterion. More broadly than implications on design criteria for safe human-structure interaction, our work points to an alternative mechanism for the emergence of collective behavior in complex systems.

Kuramoto-like synchronization analysis has so far been unable to explain many of the instability features observed on the London Millennium Bridge and many other bridges. The main features of this instability are: bridges can exhibit large vibration amplitudes in

more than one mode of vibration simultaneously, which need not be tuned to a particular walking frequency (27; 28); a critical number of pedestrians is required in order to cause an instability (29; 30); evidence of pedestrian footstep synchronization (23; 31) is scant, with the most definitive study estimating only 20% of the crowd walked in time with the bridge motion (32); engineering consultants Arup, who re-engineered the Millennium Bridge, found that each pedestrian added, on average, effective *negative damping* (29); retrofitting additional dampers successfully cures the problem (33).

One of the first to call into question the synchronization explanation of the London Millennium Bridge instability was Nobel prize winner Brian Josephson, writing four days after the bridge's opening (34):

> *The Millennium Bridge problem has little to do with crowds walking in step: it is connected with what people do as they try to maintain balance if the surface on which they are walking starts to move, and is similar to what can happen if a number of people stand up at the same time in a small boat. It is possible in both cases that the movements that people make as they try to maintain their balance lead to an increase in whatever swaying is already present, so that the swaying goes on getting worse.*

Intuitive reasoning, underlying Josephson's argument and Arup's observations suggests that, to retain balance, each pedestrian should seek to lose angular momentum within their frontal plane. Further, Barker (35) identified a stepping mechanism whereby forces to the left and right do not necessarily average out. Therefore, on average, lateral vibration energy is

transferred from the pedestrian to the bridge vibration mode. In effect, each pedestrian applies negative damping to the bridge.

In fact, the situation is more subtle. The interaction force at the bridge vibration frequency can be decomposed into components in phase with the bridge's acceleration and in phase with the bridge's velocity. The former changes the effective inertia of the bridge motion, whereas the latter changes the bridge's effective damping (36; 37). Paradoxically, for some specific combinations of the bridge vibration and pedestrian walking frequencies, a theoretical argument suggests (38; 39) that the pedestrian can effectively extract energy from the bridge, which has been confirmed in laboratory treadmill tests (40; 41; 42).

Until now, it has been hard to quantify this negative damping effect in a model-independent way. Several theories have been proposed for its physical origin (35; 38; 39; 43); however, it is not clear whether negative damping can be a consequence of synchronization (44) or vice versa.

2.0.0.1 Results.

To solve this problem, we have established a general expression for the average contribution to the bridge damping of the interaction force of a single pedestrian over one gait cycle. We

have found that this increment can be written as the sum of three components (see Methods):

σ_1 coefficient of lateral bridge velocity-dependent component of pedestrian foot force on the bridge, ignoring gait timing adjustment,

σ_2 coefficient of lateral bridge velocity-dependent component of force due to adjustment of pedestrian lateral gait timing,

and σ_3 coefficient of lateral bridge velocity-dependent component of force due to adjustment to forward gait.

The terms σ_2 and σ_3 depend on the timing of the stepping behavior of pedestrians in response to the bridge motion. However, in all our simulations, we have found σ_1 to be the most important effect in triggering large-amplitude vibrations. This effect is perhaps counter-intuitive, since it may be imagined that, in the absence of phase synchrony between the bridge and pedestrian, the lateral foot force on the bridge would average to zero. However, this is not the case; see Fig. 2.1 for a detailed explanation.

The terms σ_1, σ_2, and σ_3 should be evaluated individually for each pedestrian i and will depend on that pedestrian's stride frequency ω_i as well as the vibration frequency Ω of the bridge in the mode in question. Thus, we can write the total effective damping coefficient c_T of the bridge with N pedestrians as

$$c_T = c_0 + N\overline{\sigma}(\overline{\omega}, \Omega) := c_0 + \sum_{i=1}^{N} \left(\sigma_1^{(i)}(\omega_i, \Omega) + \sigma_2^{(i)}(\omega_i, \Omega) + \sigma_3^{(i)}(\omega_i, \Omega) \right), \qquad (2.1)$$

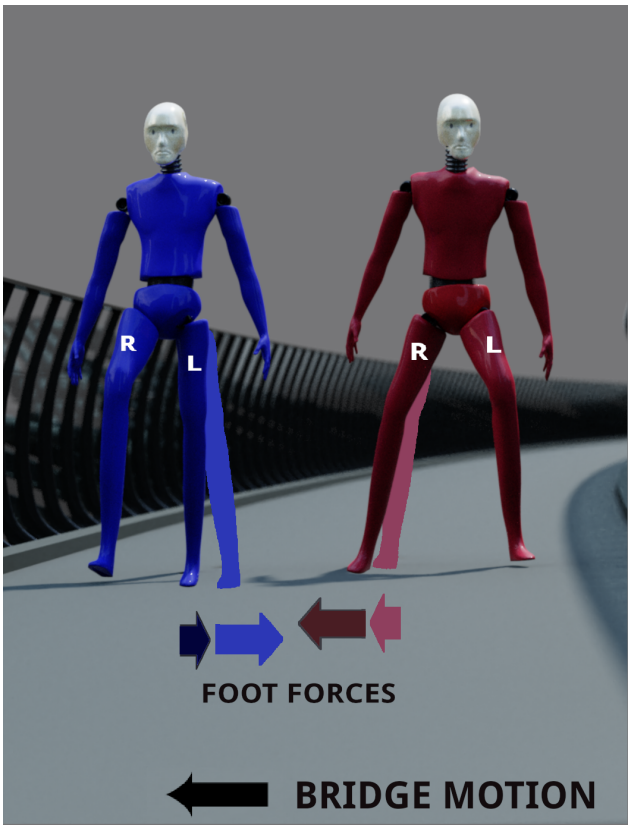

Figure 2.1 Explaining the fundamental mechanism underlying the negative damping owing to coefficient σ_1. The figure contrasts the force transmitted to the bridge by two identical pedestrians who, when they simultaneously place their stance foot on the bridge (at the dashed positions in an absolute co-ordinate frame), have equal and opposite gaits. As they place their feet, the lateral component of the foot force from each pedestrian is equal and opposite, so there is no net lateral force on the bridge. Suppose that during a time increment Δt the bridge moves to the left, so that the blue figure's leg decreases its angle to the vertical within the frontal plane, whereas the red figure's leg angle increases. Thus, during this bridge motion, the magnitude of the lateral component of the red figure's lateral foot force increases whereas that of the blue figure decreases. Thus there is, on average, a change in resultant force in the direction of the bridge's motion. Nevertheless, there can be large variations depending on a pedestrian's foot placement strategy.

where c_0 is the coefficient of natural (passive) damping of the bridge and $\overline{\omega}$ represents the mean pedestrian stride frequency.

We have found, over large ranges of pedestrian and bridge frequencies, that $\overline{\sigma} < 0$ on average. Imagine a thought experiment in which pedestrians are added to a bridge deck one by one, then when we reach a critical number

$$N = N_{\text{crit}} = -c_0/\overline{\sigma} \tag{2.2}$$

of pedestrians, the overall modal damping c_T of the bridge will become negative. Negative damping will cause the amplitude of the bridge vibration mode to grow exponentially.

To test this theory we have performed simulations on three different mathematical models describing a number of pedestrians coupled with a lateral bridge mode (see Methods for model descriptions). In each case, we make a parsimonious assumption, justified in the relevant literature, that walking is fundamentally a process in which the stance leg acts as a rigid strut, causing the body center of mass (CoM) to act like an inverted pendulum in the frontal plane (45; 46; 38; 39) during each footstep. Rather than fall over, the step ends when the other leg strikes the ground and, ignoring the brief double-stance phase seen in realistic gaits, the pedestrian switches to an inverted pendulum on that leg. We consider a single lateral vibration mode of the bridge, forced by the motion of N pedestrians walking in a direction perpendicular to this vibration. Any interaction between pedestrians other than indirectly through the bridge motion is ignored.

The modeling and simulation process are illustrated schematically in Fig. 2.2. We have

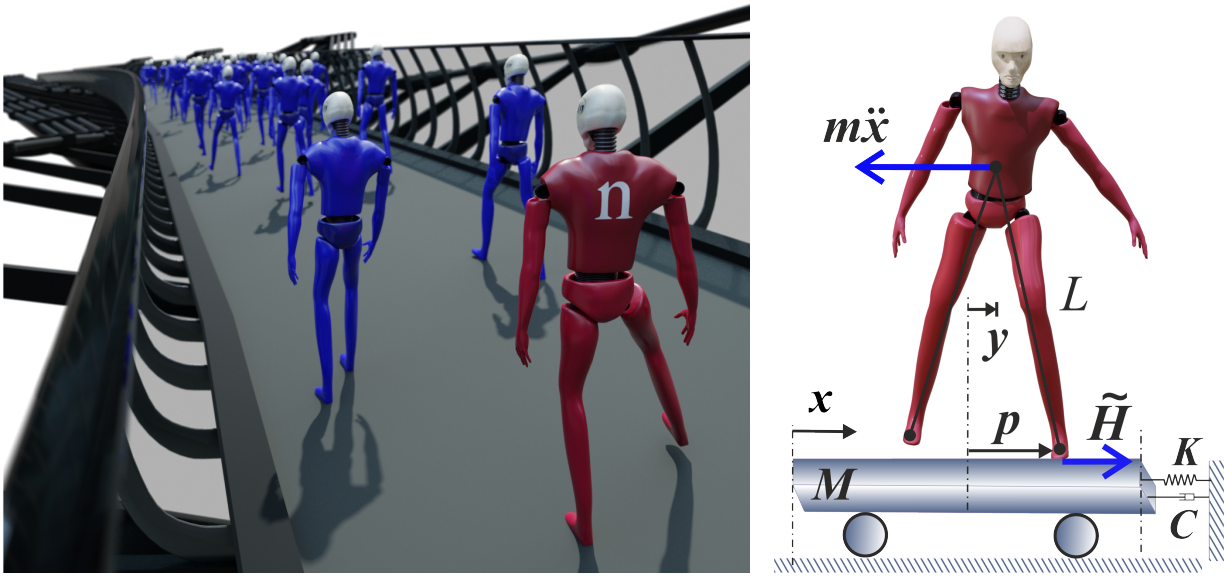

Figure 2.2 Outline of the mathematical model of pedestrian-induced lateral instability. (Left): Simulations are run for a coupled bridge-pedestrians system with pedestrians added sequentially at fixed time increments T_{add} apart. The addition of the nth pedestrian ($n = N_{\text{crit}}$) causes the overall damping coefficient to become negative hence the amplitude of motion to increase rather than diminish. (Right). Inverted pendulum model of bridge mode and pedestrian lateral motion. Here y is the lateral position of the pedestrian's center of mass, while p defines the lateral position of the center of pressure (CoP) of the foot, both relative to the bridge. L is the equivalent inverted pendulum length and m is the pedestrian mass. The displacement x of the bridge in a lateral vibration mode is represented by an equivalent platform with mass M, spring constant K, and damping coefficient C. $\tilde{H}$ is the lateral component of the pedestrian's foot force on the bridge deck. In return, the bridge motion causes an inertia force $-m\ddot{x}$ on the pedestrian's center of mass.

simulated three different variants of the pedestrian model. Model 1 (38; 39) is the most simple, based on linearizing the inverted pendulum in the frontal plane for small angles. It assumes the sagittal-plane dynamics is independent of the lateral foot position and that foot transitions occur at regularly spaced prescribed times. At each transition, the new lateral

foot position is governed by a biophysically-inspired control law (47) that enhances stability during horizontal ground motion. Model 2 is a new adaptation of Model 1, in which the timing of the foot placement alters as a kinematic consequence of the lateral bridge motion and foot placement. Finally, Model 3 (48; 26) assumes that the step timing is determined solely by the frontal-plane dynamics and that leg transition occurs each time the pedestrian CoM passes through a reference position defined as zero lateral displacement. A nonlinear feedback mechanism enables stable limit cycle motion in the absence of ground movement, and quasiperiodic motion on sinusoidally moving ground.

We choose parameters based on the set of controlled experiments on the London Millennium Bridge before reopening (29). Up to $N = 275$ pedestrians were added individually at equally spaced time intervals. It was found that significant lateral vibrations occurred for $N > 165$.

The pedestrian parameters are drawn from distributions (see Table 2.1) and multiple simulations are run for different bridge and mean pedestrian frequencies. The number of pedestrians at which the vibration amplitude begins to increase rapidly is noted for each simulation. Representative results are depicted in Fig. 2.3. For each simulation, in addition to numerical evaluation of c_T according to (2.1), we compute the well-established Kuramoto order parameter (49), r, defined using

$$re^{i\psi} = \frac{1}{N}\sum_{i=1}^{N}\langle e^{i\varphi_i}\rangle, \tag{2.3}$$

where φ_i is the numerically calculated phase of the ith pedestrian's CoM or CoP (the

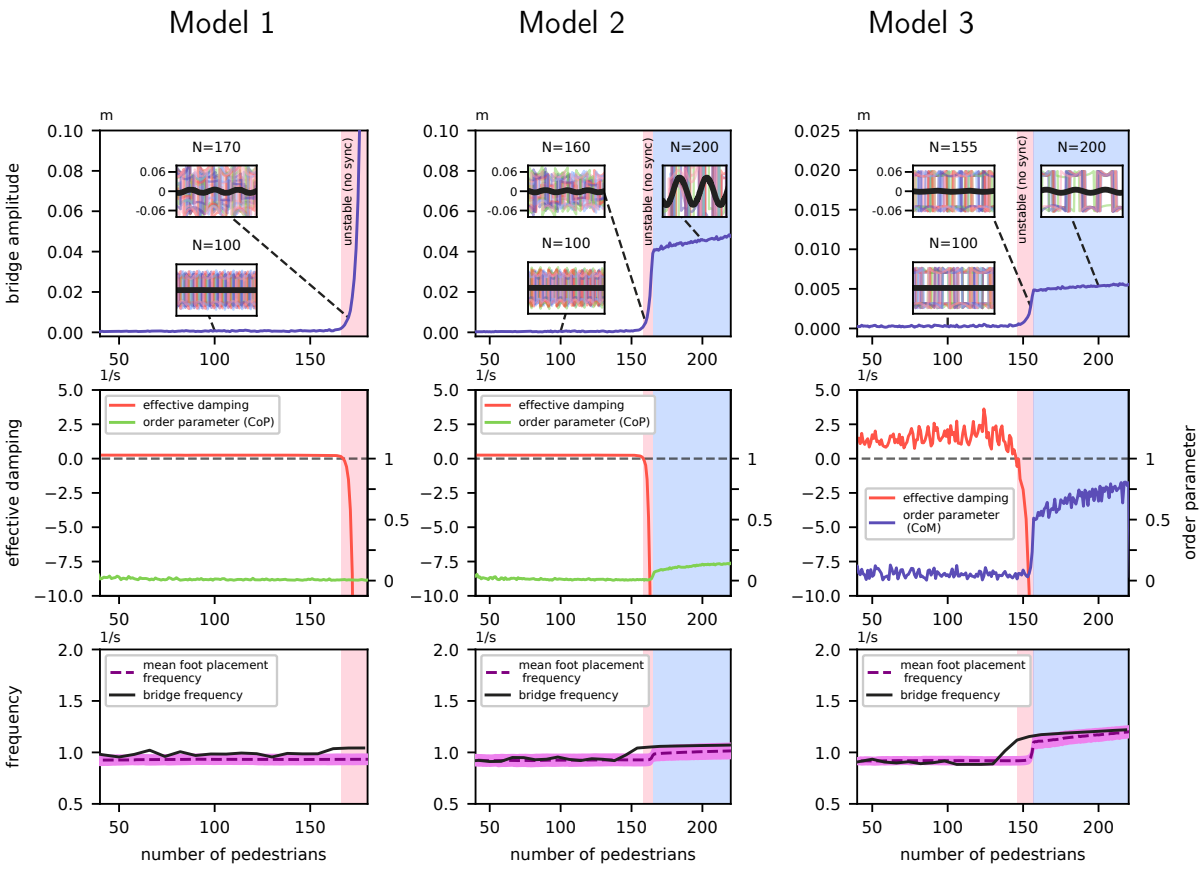

Figure 2.3 Example simulations showing the nature of the bridge instability for each of our three models. See Methods for model details and parameter values. (Top row): Bridge vibration amplitude as a function of the number of pedestrians N. The left-hand boundary of the pink shaded portion indicates the value N_{crit} where c_T crosses zero, and the blue-shaded portion is where a degree of synchrony is observed. Insets show illustrative bridge $x(t)$ (black) and a few representative pedestrian $y(t) - p(t)$ (colored) oscillations over three cycles. (Middle row): Computation of the total bridge damping c_T given by Eq. (2.1) and the Kuramoto order parameter r Eq. (2.3) calculated for the phases of pedestrians' CoP (Models 1,2) and CoM (Model 3). (Bottom row): instantaneous computed bridge and pedestrian foot placement frequencies.

distinction is made in Fig. 2.3), ψ is the average phase, and $\langle\cdot\rangle$ denotes time average. $r = 1$ implies complete synchrony, and $r = 0$ implies uncorrelated motion.

The simulations in Fig. 2.3 show how the onset of large amplitude bridge motion coincides with when the computed c_T becomes negative, at $N = N_{\text{crit}}$. For Model 1, in which

there is no adjustment to the gait frequency, the bridge's vibration amplitude grows unrealistically without bounds. In contrast, for Model 3, the onset of moderate amplitude motion starts a process of increased coherence (or phase pulling (42)) between the pedestrians' and bridge motion. The order parameter and inset sample solution traces indicate that increased synchrony then occurs between each pedestrian and the bridge. The amplitude of bridge vibrations then saturates. Model 2, which is a more realistic version of Model 1 for higher than moderate amplitude of bridge motion, shows similar amplitude saturation and coherence after instability occurs. Further simulations of Models 2 and 3 for different frequency parameters show that instability is at approximately $N = N_{\text{crit}}$ defined by (2.2), leading to a varying amount of synchrony as the amplitude grows. Thus, the negative-damping criterion can be understood as the cause of instability in all cases. Also, the varying degrees of synchrony are a consequence, not the cause of the instability.

The final question we have solved explains how the negative-damping coefficient depends on bridge and mean pedestrian stride frequencies Ω and $\overline{\omega}$, and whether it can be enhanced or suppressed by resonance effects. Figure 2.4 shows the results of many ensemble runs. For each model we show in an upper plot the computed value of $\overline{\sigma}_1$ as a function of the ratio $\Omega/\overline{\omega}$ of bridge to average pedestrian frequency.

Note that Models 1 and 2 are effectively identical prior to the onset of any large-amplitude bridge motion. For Model 1, McRobie (50) has derived an exact analytic expression for $\overline{\sigma}_1$ and N_{crit} (shown as the red curve in the left-hand panel of Fig. 2.4). The theory (50) predicts the much larger spread of σ-values we observe precisely at the resonance condition where

$\overline{\omega} = \Omega$ (represented by the yellow dot). The hypotheses behind our general calculation of $\overline{\sigma}$ fail precisely at this resonance. Also observe the paucity of data in certain regions of the lower right-hand panel of Fig. 2.4 and the apparent bi-modality of the data. This is because, for Model 3, limit cycle pedestrian motion is an emergent property of the simulations, rather than essentially an input parameter as it is for Models 1 and 2. Also, note this model is liable to hysteresis between limit cycles of different period (48).

For all three models, we find the value of the average value of $\overline{\sigma}$ to be mostly a function of the frequency ratio, being only a weak function of the pedestrian or bridge frequencies independently. Using this value in Eq. (2.2) gives the predicted critical number N_{crit} of pedestrians required to trigger an instability. The lower plots indicate the success of this prediction, by comparing with the value of N at which the vibration amplitude begins to increase rapidly in the simulations.

Also note the large spread of the model outputs for both $\overline{\sigma}_1$ and N_{crit}, especially for Model 2. Our theoretical calculations only consider the long-term averages of the effective damping coefficient c_T. This is only part of the story because true walking behavior is transient and involves changes to the trajectory of the walker's CoM and the foot placement strategy. On stationary ground, a walker's CoM will oscillate laterally with a dominant component at half the footfall frequency. Without changing the footfall frequency, the platform motion introduces a second frequency inducing the walker to adopt a two-frequency quasiperiodic pattern of footfall placement. Depending on the phase of this quasiperiodic pattern, we have found that pedestrians can show large deviations from the long-term average.

Nevertheless, for all three models, note that N_{crit} is minimised not when there is a frequency match between the pedestrian and bridge frequencies, $\Omega/\overline{\omega} = 1$, but when the pedestrian frequency is less than the bridge frequency, $\Omega/\overline{\omega} \approx 1.3$ for Models 1 and 2 and $\Omega/\overline{\omega} \approx 1.1$ for Model 3. Note, too, that there are some frequency ratios for which $\overline{\sigma}$ is positive. If pedestrians walked at those frequencies, then their motion would enhance that bridge mode's stability rather than reduce it.

An explanation of this frequency dependence can be summarised as being a question of timing. The argument in the caption of Fig. 2.1 implicitly assumes that the bridge is moving in a single direction during each step and that the bridge and pedestrian stride frequencies are similar. Particular tunings of this frequency ratio can in fact lead to a reversal of the effect in Fig. 2.1. Nevertheless, over the frequency range considered, both the size of the regions of pedestrian-induced negative damping and its average value greatly outweigh that of positive damping.

In conclusion, we have shown that the fundamental mechanism behind pedestrian-induced lateral instability of bridges is due to each pedestrian providing positive feedback to any lateral motion of the bridge. This has been dubbed a **negative-damping instability**. Mathematically, this is an example of a Hopf bifurcation and is characterized by a complex conjugate pair of eigenvalues of the bridge dynamics, crossing the imaginary axis (51). An analogous instability is well known in fluid-structure interaction, where it is called flutter.

Our results show that this negative-damping effect is intrinsic to humans trying to maintain balance without any requirement for synchronization of their footsteps. Any increased

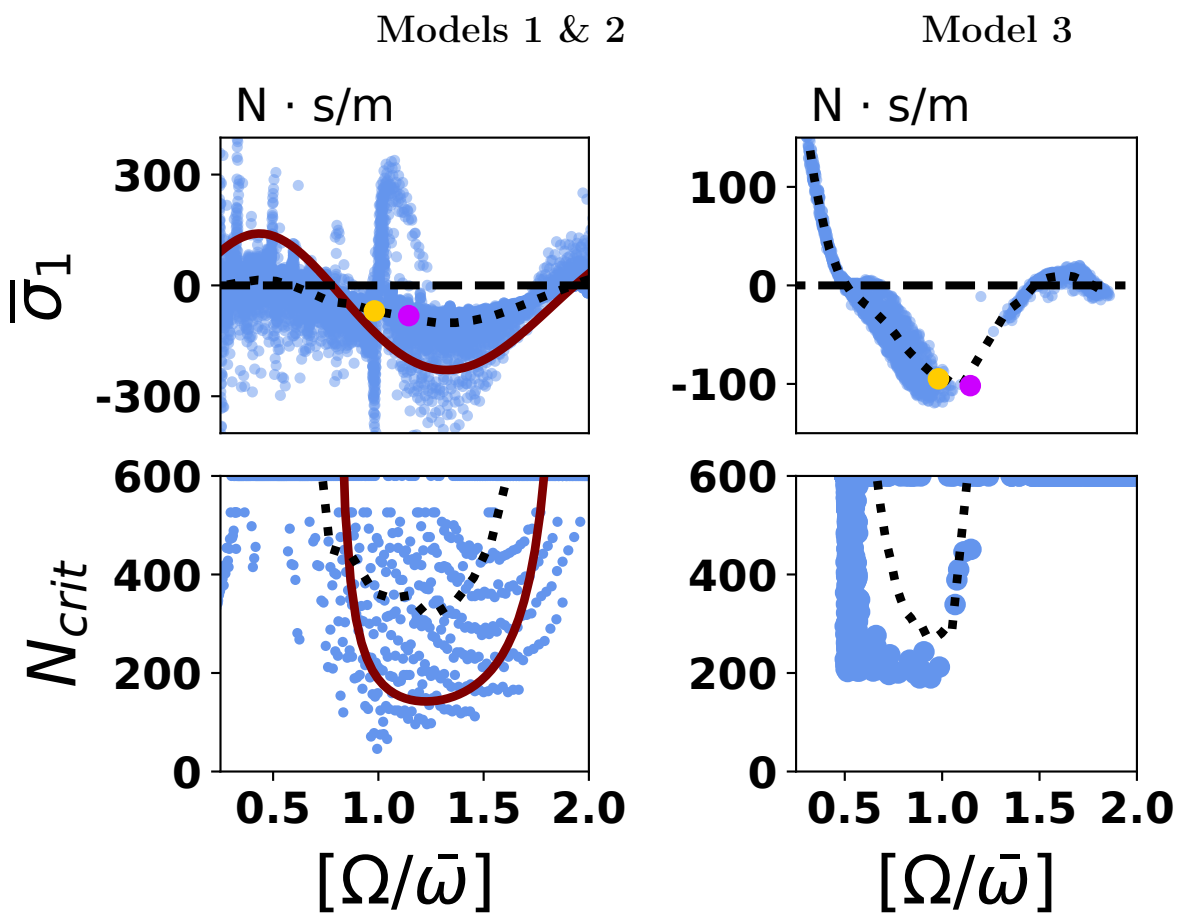

Figure 2.4 Average damping coefficient per pedestrian $\overline{\sigma}_1$ (top row) and the critical crowd size N_{crit} (bottom row) as a function of numerically calculated bridge and pedestrian frequencies ratio $[\Omega/\overline{\omega}]$. Simulations of Models 2 and 3 indicate the range of frequency ratio $[\Omega/\overline{\omega}]$ in which $\overline{\sigma}_1$ is negative so that a single pedestrian, on average, contributes to bridge instability. Each ratio of $[\Omega/\overline{\omega}]$ corresponds to different combinations of Ω and $\overline{\omega}$ (blue dots). Black dotted lines indicate the average of $\overline{\sigma}_1$ and N_{crit} for a given ratio. The red curve is the analytical expression (2.35) for $\overline{\sigma}_1$ (top plot) and analytical estimate (2.36) for N_{crit} (bottom plot), given in a subsequent section and calculated for Model 1 with identical pedestrians with fixed $\omega = 5.655$ and $S.D. = 0$. The pink dot corresponds to the initial ratio $[\Omega/\overline{\omega}]$ used in Fig. 2.3, the yellow dot corresponds to $\Omega/\overline{\omega} = 1$.

coherence of foot placement or **synchronization is a consequence of, not a cause of the instability** and is part of the process of the nonlinear adjustment to the amplitude of vibration after the instability has been initiated, which may cause saturation of the vibration amplitude or, in extreme cases, may further exacerbate the instability. These findings should enable bridge designers and other structural engineers to develop more accurate design criteria to avoid human-induced instability of a wide range of structures.

At a more general level, our results point to a new kind of emergent behavior among autonomous agents. The usual theory of synchronization distinguishes between cases where there is a master conductor that all other agents follow, and where synchrony emerges spontaneously without a leader. We have uncovered a third possibility, that there is an underlying, albeit nascent, collective frequency that does not become excited until the individual agents are sufficiently active. Each agent need not synchronize to the collective frequency nor to another agent. Each agent simply needs to display some positive feedback effect. An intuitive yet erroneous argument might suggest that in the absence of coherence, the feedback from all the agents would, on average, cancel each other out. But this is not how positive feedback works, it creates a bias that can lead to negative damping. This new kind of **emergent instability** may actually be more prevalent in nature and society than previously thought. For example, both in the mammalian (52) and insect (53) hearing systems, single-frequency instability of an active system can occur due to the beating of tiny incoherent neuro-mechanical oscillators. Also, macro-scale economic and financial systems tend to develop characteristic cycles (54) without there being obvious causal synchrony at the

microeconomic level.

2.1 Methods

2.1.1 Mathematical model

The displacement of the lateral bridge mode $x(t)$ is assumed to be governed by a simple second-order equation of motion

$$M\ddot{x} + C\dot{x} + Kx = \sum_{i=1}^{N} \tilde{H}^{(i)}(x, y^{(i)}), \tag{2.4}$$

where M, C, and K are the mass, damping, and stiffness coefficients, respectively, of the bridge mode. The forcing term $\tilde{H}^{(i)}$ is the lateral component of the ith pedestrian's foot force on the bridge deck. The lateral component of each single pedestrian's center of mass obeys the equation

$$m\ddot{y}^{(i)} + m\ddot{x} = -\tilde{H}^{(i)}(x, y^{(i)}), \quad i = 1, \ldots N. \tag{2.5}$$

In general $\tilde{H}^{(i)}$ is a function of exogenous variables associated with the pedestrian's gait, particularly the lateral motion, and will typically be a piecewise-smooth function with abrupt changes at foot transitions. Specifically, we assume that foot transitions occur at a sequence of times $\{t_s^{(i)}\}$, $s = 1, 2, 3, \ldots$, where $t_{s+1}^{(i)} > t_s^{(i)}$ for all s. By definition, the angular pedestrian stride frequency is $[\omega_i] = 2\pi/[(t_{s+2}^{(i)} - t_s^{(i)})]$, where $[\cdot]$ denotes possible adjustment due to bridge motion. For definiteness, we assume even s corresponds to the touchdown of the right foot and odd s to touchdown of the left.

Our analysis of negative damping is applicable to any model that can be written in the

form (2.4), (2.5). It is helpful to scale parameters and introduce dimensionless parameters ε and ζ measuring mass and damping ratios respectively:

$$H^{(i)} = \tilde{H}^{(i)}/m, \qquad \Omega = \sqrt{K/M}, \qquad \varepsilon^2 = m/M, \qquad \zeta = \frac{C}{2M\Omega\varepsilon}. \tag{2.6}$$

Then the equations of motion can be written in the form

$$\ddot{x} + 2\varepsilon\Omega\zeta\dot{x} + \Omega^2 x = \varepsilon^2 \sum_{i=1}^{N} H^{(i)}, \qquad \ddot{y}^{(i)} + H^{(i)} = -\ddot{x}, \quad i = 1, \ldots, N. \tag{2.7}$$

Note the modeling choice that the bridge's natural damping in (2.7) is assumed to be $\mathcal{O}(\varepsilon)$. This is consistent with values of bridge damping and number of pedestrians $N = \mathcal{O}(\varepsilon^{-1})$ required to trigger instability observed in practice.

Treating ε as a small parameter, a lengthy, but straightforward multiple-scale asymptotic expansion (see subsection Asymptotic derivation of negative damping criterion) can be used to evaluate the total bridge damping as the natural damping plus three additional terms:

$$c_T = 2\varepsilon\zeta\Omega + \varepsilon\nu(\overline{\sigma}_1 + \overline{\sigma}_2 + \overline{\sigma}_3) \;=\; 2\varepsilon\zeta\Omega + N\varepsilon^2 \sum_{i=1}^{N} (\sigma_1^{(i)} + \sigma_2^{(i)} + \sigma_3^{(i)}), \tag{2.8}$$

with

$$\sigma_1^{(i)} = \frac{1}{T_i} \int_0^{T_i} \frac{\partial H^{(i)}}{\partial \dot{x}} dt, \tag{2.9}$$

$$\sigma_2^{(i)} = \frac{1}{T_i\Omega} \left(\overline{y}_s^{(i)} \int_0^{T_i} \frac{\partial H^{(i)}}{\partial y} dt + \Omega\, \overline{y}_c^{(i)} \int_0^{T_i} \frac{\partial H^{(i)}}{\partial \dot{y}} dt \right), \tag{2.10}$$

$$\sigma_3^{(3)} = \frac{1}{T_i\Omega} \left(\overline{z}_s^{(i)} \int_0^{T_i} \frac{\partial H^{(i)}}{\partial z} dt + \Omega\, \overline{z}_c^{(i)} \int_0^{T_i} \frac{\partial H^{(i)}}{\partial \dot{z}} dt \right). \tag{2.11}$$

Here, a subscript c means component in phase with the bridge instantaneous displacement (c stands for cosine), and s means component in anti-phase with the bridge velocity (s stands for sine). Also, an overline means a time average over many steps. Furthermore, $z(t)$ is the perturbation, due to the lateral motion, of the pedestrian's forward position relative to a constant forward speed. Because each function $H^{(i)}$ is in general nonsmooth, partial derivatives should be interpreted in the distributional sense.

The particular pedestrian models we use in our simulations are distinguished only by their choice of the foot force function $H^{(i)}$, which we assume to take an identical form for each pedestrian, but to have parameters that can vary between pedestrians.

2.1.1.1 Model 1: Linearized inverted pendulum with step width control.

This model was developed by Macdonald, Bocian, and Burn (38; 39) and was shown to exhibit similar features to those observed in four independent experimental studies (55; 41; 42; 56; 40). Here

$$H^{(i)}(t) = \frac{g}{L}(p^{(i)}(t_s) - y^{(i)}), \tag{2.12}$$

with g being gravitational acceleration and L effective leg length, and $p^{(i)}(t_s)$ is the lateral center of pressure of the foot placed at time t_s. At the beginning of each step, $p^{(i)}(t_s)$ is adjusted according to the self-balancing control law determined theoretically and experimentally by Hof *et al.* (47; 57):

$$p^{(i)}(t_s) = y^{(i)}(t_s^-) + \sqrt{\frac{L}{g}}\dot{y}^{(i)}(t_s^-) + (-1)^s b_{\min}, \tag{2.13}$$

where t_s^- is the time immediately before foot transition, and $b_{\min} > 0$ is the margin of stability, proportional to the natural gait width in the absence of any bridge motion. In this model, the walking frequency that defines the switching times t_s is given by an external clock and is not adjusted due to bridge motion. Thus each ω_i remains constant throughout the simulation.

2.1.1.2 Model 2: Model 1 with step-timing adaptation.

We introduce adaptation to the step time t_s due to the geometric nonlinearity associated with the adjustment to the lateral gait width. Consider a rigid, three-dimensional inverted pendulum of length $L = \sqrt{X^2 + Y^2 + Z^2}$, where X, Y, and Z represent, respectively, displacements of the center of mass, relative to the center of pressure (CoP) of the stance foot, in longitudinal, transverse, and vertical pedestrian-centered coordinates. Suppose $X(t_s^-) = X_0 + \Delta X$, where (X_0, Y_0, Z_0) is the position of the center of mass at touchdown of the next foot for unperturbed steady-state walking. Assume that, with perturbations from the bridge motion, foot transition still occurs when $Z = Z_0$, then in the limit of small ΔX, we can write

$$\Delta X = \frac{1}{2X_0}(Y_0^2 - Y(t_s^-)^2), \tag{2.14}$$

where $Y(t_s^-) = y^{(i)}(t_s^-) - p^{(i)}(t_{s-1})$ is the transverse position of the center of mass, relative to the CoP, at touchdown, with $p^{(i)}(t_{s-1})$ from (2.13).

Introducing the mean forward velocity

parameter	meaning	units	default value	mismatch S.D.	source
a	auxiliary	m	0.047	0	Ref. (48; 26)
$b_{\min}$	margin of stability	m	0.0157	0.002	Ref. (38)
C	bridge damping	Ns/m	29,251		
g	acceleration of gravity	ms^{-2}	9.81		
L	effective leg length	m	1.17	0.092	Ref. (42)
m	pedestrian mass	kg	76.9	10	Ref. (42)
M	bridge mass	kg	113,000		Ref. (24)
p_c	auxiliary	m	0.063	0	Ref. (48; 26)
T_{add}	pedestrian addition time	s	20		
X_0	unperturbed half step length	m	0.36		Ref. (45; 58)
Y_0	unperturbed half step width	m	0.047		Ref. (38)
λ	damping due to walking	sm^{-2}	23.25	0	
ω	unperturbed angular stride frequency	rad/s	5.655	0.1	Ref. (26; 45)
Ω	angular bridge natural frequency	rad/s	6.503		Ref. (24; 25)

Table 2.1 Default parameter values used in the simulations. Here, S.D. is the standard deviation of parameter mismatch among pedestrians, which follows a normal distribution in all cases.

$$\chi = \frac{2X_0}{\pi/\omega_i} = \frac{2}{\pi} X_0 \omega_i, \tag{2.15}$$

the perturbation to the timing of the next step is approximately $\Delta t = \Delta X/\chi$, hence the time of the next step is given by

$$t_s = t_{s-1} + \frac{\pi}{\omega_i} + \frac{\Delta X}{\chi} = t_{s-1} + \frac{\pi}{\omega_i}\left[1 + \frac{Y_0^2 - \{y^{(i)}(t_s^-) - p^{(i)}(t_{s-1})\}^2}{2X_0^2}\right].$$

2.1.1.3 Model 3: Rocking inverted pendulum.

We have also implemented the autonomous walking model proposed and studied by Belykh *et al.* (48; 26) that displays stable limit cycle motion without the need for any control. Here

$$H = \lambda\left[\dot{y}^2 + \frac{g}{L}\left\{a^2 - (y - p_c sgn(y))^2\right\}\right]\dot{y} - \frac{g}{L}(y - p_c sgn(y)), \tag{2.16}$$

where, in contrast to Models 1 and 2, the lateral position of the CoP of the foot p is a fixed margin, denoted by constant p_c. Here, λ is a damping parameter, a is a parameter that controls the amplitude and the period of the limit cycle. In the absence of bridge motion, the amplitude and period of the limit cycle can be calculated explicitly. (48)

Unlike Models 1 and 2, the times at which the system with foot force (2.16) switches legs depends on the lateral motion of the center of mass, rather than the forward walking speed. That is, leg transition occurs whenever y crosses zero. Thus, the walking frequency adapts in the presence of bridge motion.

2.1.1.4 Asymptotic derivation of negative damping criterion.

Our aim is to derive a general expression for the total bridge damping for a general model of the form (2.7), as a function of the number of pedestrians. Hence we seek to find the number of pedestrians N_{crit} required for instability. In this section all frequencies are assumed to be angular frequencies in units of radians per second. We shall discover that $N_{\text{crit}} = \mathcal{O}(\varepsilon^{-1})$, hence it will be convenient in what follows to write

$$N = \nu\varepsilon^{-1}, \qquad \text{where} \quad \nu = \mathcal{O}(1). \tag{2.17}$$

We shall assume that the forward motion of the pedestrian's center of mass can also be described by a single degree of freedom $z^{(i)}$. Thus the general dimensionless model can be written in the form:

ith-pedestrian lateral motion:

$$\ddot{y}^{(i)} + H^{(i)}(x, \dot{x}, y^{(i)}, \dot{y}^{(i)}, z^{(i)}, \dot{z}^{(i)}) = -\ddot{x}, \tag{2.18}$$

ith-pedestrian forward motion:

$$\ddot{z}^{(i)} + G^{(i)}(y^{(i)}, \dot{y}^{(i)}, z^{(i)}, \dot{z}^{(i)}) = 0, \tag{2.19}$$

single lateral bridge mode:

$$\ddot{x} + \varepsilon 2\zeta\Omega\dot{x} + \Omega^2 x = \varepsilon^2 \sum_{i=1}^{N} H^{(i)}(x, \dot{x}, y^{(i)}, \dot{y}^{(i)}, z^{(i)}, \dot{z}^{(i)}). \tag{2.20}$$

Here $G^{(i)}$ is a general nonlinear function of its arguments and, like $H^{(i)}$, is typically nonsmooth.

In the absence of bridge motion, we assume that the pedestrian dynamics

$$\ddot{y}^{(i)} + H^{(i)}(0, 0, y^{(i)}, \dot{y}^{(i)}, z^{(i)}, \dot{z}^{(i)}) = 0, \qquad \ddot{z}^{(i)} + G^{(i)}(y^{(i)}, \dot{y}^{(i)}, z^{(i)}, \dot{z}^{(i)}) = 0$$

admits an asymptotically stable limit cycle with period $T_i = 2\pi/\omega_i$:

$$y^{(i)} = y_0^{(i)}(t), \quad y_0^{(i)}(t) = y_0^{(i)}(t + T_i), \quad z_0^{(i)}(t) = \chi t + z_0^{(i)}(t + T_i).$$

where y_0 and z_0 are periodic functions of time, and χ is the average forward velocity of the pedestrian's center of mass. Moreover, we suppose that

$$H^{(i)}(0, 0, y_0^{(i)}, \dot{y}_0^{(i)}, z_0^{(i)}, \dot{z}_0^{(i)}) = h_0^{(i)}(t), \quad \text{and} \quad G^{(i)}(y_0^{(i)}, \dot{y}_0^{(i)}, z_0^{(i)}, \dot{z}_0^{(i)}) = g_0^{(i)}(t)$$

are T_i-periodic functions.

We begin with a technical, detuning assumption that simplifies the analysis, namely that

each pedestrian has an independent frequency ω_i, and that there exists a constant $R > 0$ such that

$$\min_{i \neq j} |\omega_i - \omega_j| > R\varepsilon, \qquad \min_i |\omega_i - \Omega| > R\varepsilon. \tag{2.21}$$

We look for a coupled solution to the system (2.18)–(2.20) as an asymptotic expansion in $\varepsilon \ll 1$ of the form

$$x = \varepsilon x_1(t) + \varepsilon^2 x_2(t) + \ldots, \quad y^{(i)} = y_0^{(i)}(t) + \varepsilon y_1^{(i)}(t) + \ldots, \quad z^{(i)} = \chi t + z_0^{(i)}(t) + \varepsilon z_1^{(i)}(t) + \ldots.$$

For ease of notation, we shall let $\dot{x} = u$, $\dot{y} = v$, $\dot{z} = w$ and drop the superscript (i) in what follows, providing the meaning is clear. Also, let us define the vector $\xi = (x, u, y, v, z, w)$ and let $\xi_0 = (0, 0, y_0, v_0, \chi t + z_0, c + w_0)$ be the unperturbed limit-cycle motion. Now we can formally expand the functions H and G as power series

$$H = h_0 + \sum_{k=1}^{6} h_{\xi_k}(\xi_k - \xi_{k0}) + \mathcal{O}(|\xi - \xi_0|^2), \qquad G = g_0 + \sum_{k=1}^{6} g_{\xi_k}(\xi_k - \xi_{k0}) + \mathcal{O}(|\xi - \xi_0|^2). \tag{2.22}$$

Note that the coefficients h_{ξ_k}, and g_{ξ_k} represent partial derivatives of H and G with respect to their subscripted arguments, evaluated along the unperturbed solution $y = y_0(t)$, $z = \chi t + z_0(t)$. Hence each of these coefficients is a T-periodic function of time.

First-order solution. Substitution of the zeroth-order solution into (2.20) yields, to leading order,

$$\ddot{x}_1 + \Omega^2 x_1 = 0, \tag{2.23}$$

where we have used the assumption (2.21) that the pedestrians are uncorrelated to assume

$$\sum_{i=1}^{N} h_0^{(i)}(t) \ll \mathcal{O}(\varepsilon^{-1}).$$

The solution to (2.23) is the free vibration of the bridge, which can be written in the form

$$x_1(t) = X(\tau)\cos(\Omega t + \phi(\tau)), \tag{2.24}$$

where the amplitude X and phase ϕ are allowed to be functions of a slow time variable $\tau = \varepsilon t$.

Substitution of x_1 into (2.18) and (2.19) using (2.22) yields, to leading-order in ε,

$$\begin{pmatrix} \ddot{y}_1 \\ \ddot{z}_1 \end{pmatrix} + \begin{pmatrix} h_y & h_z \\ g_y & g_z \end{pmatrix} \begin{pmatrix} y_1 \\ z_1 \end{pmatrix} + \begin{pmatrix} h_v & h_w \\ g_v & g_w \end{pmatrix} \begin{pmatrix} \dot{y}_1 \\ \dot{z}_1 \end{pmatrix} = \begin{pmatrix} \Omega^2 - h_x & \Omega h_u \\ 0 & 0 \end{pmatrix} \begin{pmatrix} X(\tau)\cos[\Omega t + \phi(\tau)], \\ X(\tau)\sin[\Omega t + \phi(\tau)]. \end{pmatrix}. \tag{2.25}$$

This is a linear system with periodic coefficients and periodic forcing. It can be solved as the sum of free and forced vibration terms. Under the assumption that $\omega_i \neq \Omega$, and that the limit cycle in the absence of bridge motion is asymptotically stable, the free vibration part must decay to zero for large times. The only non-decaying part comes from the forced vibration. We find approximate expressions for this term by averaging the periodic functions h_ξ and g_ξ. Let an overbar represent the average of a quantity over each period T. That is

$$\overline{h}_{\xi_k}^{(i)} := \frac{1}{T_i}\int_0^{T_i} \left.\frac{\partial H^{(i)}}{\partial \xi_k}\right|_{x=0, y=y_0^{(i)}, z=\chi t + z_0^{(i)}} dt, \quad \overline{g}_{x_k}^{(i)} := \frac{1}{T_i}\int_0^{T_i} \left.\frac{\partial G^{(i)}}{\partial \xi_k}\right|_{x=0, y=y_0^{(i)}, z=\chi t + z_0^{(i)}} dt.$$

Then the solution for the forced vibration problem can be written in the form

$$y_1(t) = X(\tau)(y_c \cos[\Omega t + \phi(\tau)] + y_s \sin[\Omega t + \phi(\tau)] \ + y_r(t)), \tag{2.26}$$

$$z_1(t) = X(\tau)(z_c \cos[\Omega t + \phi(\tau)] + z_s \sin[\Omega t + \phi(\tau)] \ + z_r(t)) \tag{2.27}$$

where $y_{c,s}$, $z_{c,s}$ are constant amplitudes of cosines and sines of period T, and y_r and z_r are remainder terms that contain all other harmonics.

Expressions for $y_{c,s}$, $z_{c,s}$ can be written in closed form

$$y_c = \frac{D\Omega^2 - D\overline{h}_x - B\Omega\overline{h}_u}{AD - BC}, \qquad y_s = \frac{A\Omega\overline{h}_u - C\Omega^2 + C\overline{h}_x}{AD - BC},$$

$$z_c = -Z_2\, y_c + Z_1\, y_s, \qquad z_s = -Z_2\, y_c + Z_1\, y_s$$

where

$$A = \overline{h}_y - \Omega^2 - Z_1\overline{h}_z - Z_2\Omega\overline{h}_w, \qquad B = \Omega\overline{h}_v - Z_1\Omega\overline{h}_w + \overline{h}_z Z_2,$$

$$C = -\Omega\overline{h}_v + \overline{h}_w\Omega Z_1 - \overline{h}_z Z_2, \qquad D = \overline{h}_y - \Omega^2 - \overline{h}_w\Omega Z_2 - \overline{h}_z Z_1,$$

$$Z_1 = \frac{\Omega^2(\overline{g}_v\overline{g}_w - \overline{g}_y) + \overline{g}_y\overline{g}_z}{\Omega^4 + \omega^2\overline{g}_w^2 - 2\Omega^2\overline{g}_z + \overline{g}_z^2} \quad Z_2 = \frac{\Omega^2\overline{g}_v - \Omega(\overline{g}_y\overline{g}_w - \overline{g}_v\overline{g}_z)}{\Omega^4 + \omega^2\overline{g}_w^2 - 2\Omega^2\overline{g}_z + \overline{g}_z^2}\ .$$

Second-order solution. Substitution of the $\mathcal{O}(1)$ solution into bridge equation (2.20) at

second order yields

$$\ddot{x}_2 + \Omega^2 x_2 = \sum_{i=1}^{N} h_0^{(i)} + \varepsilon \sum_{i=1}^{N} h_1^{(i)} - [\dot{x}_1' + 2\zeta\Omega\dot{x}_1], \tag{2.28}$$

where $'$ means differentiation with respect to the slow time τ and

$$h_1^{(i)} = h_x^{(i)} x_1 + h_u^{(i)} \dot{x}_1 + h_y^{(i)} y_1^{(i)} + h_v^{(i)} \dot{y}_1^{(i)} + h_z^{(i)} z_1^{(i)} + h_w^{(i)} \dot{z}_1^{(i)}.$$

We can now substitute the form of x_1 from (2.24) and of $y_1^{(i)}$ and $z_1^{(i)}$ from (2.27) for each pedestrian i, and seek the general solution to the forced vibration problem.

In order to find a consistent asymptotic solution, under the formalism of the method of multiple scales (59), we must avoid secular terms on the right-hand side of (2.28). That is the components of $\cos(\Omega t + \phi)$ and $\sin(\Omega t + \phi)$ in the forcing term must vanish. Let us consider the three terms on the right-hand side (2.28) in turn.

Consider first the term $\sum_{i=1}^{N} h_0^{(i)}$. Here the assumption (2.21) about frequency separation means that there can be no contribution to the secular terms from this sum. Next consider the term $\varepsilon \sum_{i=1}^{N} h_1^{(i)}$. At first, this seems to be at lower order and so is unlikely to contribute. But, recalling the scaling (2.17) that $N = \mathcal{O}(1/\varepsilon)$ it may be that there is a cumulative contribution from each term in the sum that can contribute at the required order. By performing a spectral decomposition of the term $h_1^{(i)}$ we find the components of $\cos(\Omega t + \phi)$ and $\sin(\Omega t + \phi)$ to be

$$\cos(\Omega t + \phi): \quad X\left[\overline{h}_x^{(i)} + (\overline{h}_y^{(i)} y_c^{(i)} - \Omega \overline{h}_v^{(i)} y_s^{(i)}) + (\overline{h}_z^{(i)} z_c^{(i)} - \Omega \overline{h}_w^{(i)} z_s^{(i)})\right] = X\left[\overline{h}_x^{(i)} + \overline{\kappa}_y^{(i)} + \overline{\kappa}_z^{(i)}\right],$$
$$\sin(\Omega t + \phi): \; X\left[-\Omega \overline{h}_u^{(i)} + (\overline{h}_y^{(i)} y_s^{(i)} + \Omega \overline{h}_v^{(i)} y_c^{(i)}) + (\overline{h}_z^{(i)} z_c^{(i)} + \Omega \overline{h}_w^{(i)} z_c^{(i)})\right] = X\left[-\Omega \overline{h}_u^{(i)} + \overline{\sigma}_y^{(i)} + \overline{\sigma}_z^{(i)}\right].$$

Note that each of these coefficients is a constant for each pedestrian, because we already

averaged out the period-T_i components in the definition (2.27). Therefore we can sum each of these N terms individually so that

$$\varepsilon \sum_{i=1}^{N} \left[\overline{h}_x^{(i)} + \overline{\kappa}_y^{(i)} + \overline{\kappa}_z^{(i)} \right] = \varepsilon N \left[\hat{h}_x + \hat{\kappa}_y + \hat{\kappa}_z \right], \quad \varepsilon \sum_{i=1}^{N} \left[\overline{h}_u^{(i)} + \overline{\sigma}_y^{(i)} + \overline{\sigma}_z^{(i)} \right] = \varepsilon N \left[-\Omega \hat{h}_u + \hat{\sigma}_y + \hat{\sigma}_z \right],$$

where $\hat{\cdot}$ means averaging over all pedestrians

$$\hat{h}_x = \frac{1}{N} \sum_{i=1}^{N} \overline{h}_x^{(i)}, \quad \hat{h}_u = \frac{1}{N} \sum_{i=1}^{N} \overline{h}_u^{(i)}, \quad \text{etc.}$$

Finally, the third term on the right-hand side of (2.28) is

$$\dot{x}_1' + 2\zeta\Omega \dot{x}_1 = -\Omega X'(\tau) \sin(\Omega t + \phi(\tau)) - \phi' X \Omega \cos(\Omega t + \phi(\tau)) - 2\zeta X \Omega^2 \sin(\Omega t + \phi(\tau)).$$

Recalling that $N\varepsilon = \nu$, the vanishing of the secular terms on the right-hand side of (2.28) thus implies

$$\begin{aligned} &\text{component of } \cos(\Omega t + \phi): \qquad 0 = \Omega \phi' X + \nu X \left[\hat{h}_x + \hat{\kappa}_y + \hat{\kappa}_z \right], \\ &\text{component of } \sin(\Omega t + \phi): \qquad 0 = \Omega X'(\tau) + 2\zeta X \Omega^2 + \nu X \left[-\Omega \hat{h}_u + \hat{\sigma}_y + \hat{\sigma}_z \right]. \end{aligned} \tag{2.29}$$

We can rewrite (2.29) in the form

$$\phi' = -\frac{\nu}{\Omega} (\hat{h}_x + \hat{\kappa}_y + \hat{\kappa}_z), \tag{2.30}$$

$$\frac{X'(\tau)}{X} = -2\zeta\Omega - \frac{\nu}{\Omega} (-\Omega \hat{h}_u + \hat{\sigma}_y + \hat{\sigma}_z). \tag{2.31}$$

The right-hand sides of the equations (2.30) and (2.31) describe the slow adaptation to the frequency and damping of the bridge due to the presence of the pedestrians.

Each of these terms has three components. These represent respectively: (I) adaptation due to direct dependence of the foot force H on the bridge motion, neglecting any change in the timing of footsteps (the terms $\hat{h}_x$ and $\hat{h}_u$); (II) the component at the bridge frequency that is present in the adjustment to the pedestrian lateral foot placement (the terms $\hat{\kappa}_y$ and $\hat{\sigma}_y$); and (III) the component at the bridge frequency that is present in the adaptation to the pedestrian's forward motion (the terms $\hat{\kappa}_z$ and $\hat{\sigma}_z$).

Let us examine the damping equation (2.31). Note that the term of the right-hand side is the $\mathcal{O}(\varepsilon)$-component of the total negative damping of the bridge. That is, in the notation of (2.8)

$$\overline{\sigma}_1 = -\hat{h}_u, \qquad \overline{\sigma}_2 = \frac{\hat{\sigma}_y}{\Omega}, \qquad \overline{\sigma}_3 = \frac{\hat{\sigma}_z}{\Omega}.$$

Note that $\overline{\sigma}_1$ is identical to the condition derived in (38; 39) and expressed analytically in Ref. (50) for the negative damping contribution for Model 1, here. The terms $\overline{\sigma}_2$ and $\overline{\sigma}_3$ are other terms that should be considered at the same order for a general foot-force model.

2.1.1.5 Model implementation.

Parameters that characterize pedestrians' walking frequencies were chosen in a biomechanically realistic range. Bridge parameters were chosen close to those of the London Millennium Bridge. Table 2.1 gives the complete set of parameter values used in the simulations. Numerical simulations were performed using bespoke software written by us, mostly in Python, with some use of MATLAB and Java. Discretization was performed using the default 4th-order adaptive Runge-Kutta method of MATLAB and the LSODA implementation in scipy's

solve$_i vp$.

2.2 Review of observational and experimental evidence

When crossing a bridge, most people take for granted that the bridge will remain steady and support them, but history shows that this is not always the case. The first documented pedestrian bridge incident dates back to April 12, 1831 when one of Europe's first suspension bridges, England's Broughton Suspension Bridge, collapsed due to dynamical instability induced by marching troops. The prevailing wisdom since is that soldiers should avoid marching in step, in case their stepping frequency might resonate with a natural (vertical) vibration frequency of the bridge. It is now established practice that soldiers are given the command to "break step" upon crossing a bridge to avoid just such a phenomenon. Vertical vibrations of bridges due to random excitation from pedestrians are still of concern, but prior to the year 2000 lateral vibrations were given little attention. This was because, for normal walking, the lateral component of the ground reaction force is an order of magnitude smaller than the vertical component, and, in the absence of coherence between pedestrians, the resulting bridge responses were assumed to be negligible.

The London Millennium Bridge was designed as a collaboration between engineers, architects, and artists, as a very low-profile suspension bridge. Without visually intrusive vertical cables, the intention was that the structure would appear from the side to be like a mysterious long blade, spanning the river with little visible support. The slenderness of the span contributed to the bridge having greater flexibility than most bridges in the lateral

direction, giving natural frequencies similar to typical pedestrian stride frequencies, while its relatively low mass also made it susceptible to significant vibrations.

In fact, this same phenomenon of a *lateral* instability of pedestrian bridges had been seen before, and there is evidence going back to 1972. The complete list of pedestrian bridges that are known to have developed lateral oscillation due to pedestrian motion runs to at least 28 separate examples; see Table 2.2 for a list of those for which there are detailed scientific reports and Table 2.3 for others for which quantitative evidence is not available. Note in the final column of these tables the scant evidence for pedestrian synchronization being observed.

The geography of such crowd-induced instability events is truly worldwide. It includes the massive Bosphorus Bridge linking Asia and Europe (60) and an icon of Lower Manhattan, the Brooklyn Bridge which started swaying as a crowd of pedestrians trudged across during the 2003 blackout. When packed shoulder to shoulder with pedestrians, the bridge started vibrating making pedestrians lose balance and feel seasick (61). The Brooklyn Bridge repeatedly experienced crowd-induced instabilities during the 2011 protest and 2011 New Year's celebration (44) raising the concern that "Manhattans's emergency exit" - as the bridge is sometimes called – is not built for crowds.

Coincidentally, one of the more recent examples of lateral pedestrian instabilities is Squibb Park Bridge, also in Brooklyn (it is a city of bridges, after all) (62). Opened in 2013, this $3.9-million wooden park bridge was purposefully designed to bounce lightly but over time the increased bouncing and lateral swaying became a safety concern for pedestrians (63).

Three years after it was initially closed for $2.5-million repairs, the Squibb Park Bridge reopened in April 2017 (64) but was later demolished in 2019 amid concerns of its structural integrity.

While the evidence of bridge instabilities is often anecdotal, some direct measurements of bridge response characteristics are available for recent crowd-induced instability events involving the Toda Park Bridge in Japan (66), Solférino Bridge in Paris (76), the London Millennium Bridge (29), the Maple Valley Great Suspension Bridge in Japan (32), Singapore Airport's Changi Mezzanine Bridge (28), the Clifton Suspension Bridge in Bristol, UK (27), and the Pedro e Inês Footbridge in Portugal (30).

A particularly notable observation was the instability due to crowds returning from an annual hot-air balloon festival across Bristol's iconic Clifton Suspension Bridge (27). Since vibrations of the bridge had been observed during previous crowd events, Macdonald was commissioned by the bridge's operating trust to fit accelerometers to record the vibrations as the instability occurred. Observations showed that two lateral modes of vibration were excited simultaneously by the large pedestrian crowd, neither of which was tuned to the average walking frequency. Since then, the trust has stipulated that the bridge must remain closed to all pedestrians and other traffic at peak times during the balloon festival.

2.3 Video evidence of pedestrian-bridge interaction in Nepal

Figure 2.5 provides evidence from the personal experience of Prof. Belykh while crossing Dudh Kosi River bridges on the way to Mount Everest in Nepal. These unstable bridges

Bridge	Country	Year reported	Bridge type	Length (m)	Frequency (Hz)	Observation	Sync evidence
Erlach Footbridge (65)	Germany	1972	several span continuous girder; main span supported by arch	110	1.12	strong response with 300-400 crossing pedestrians	no evidence
Toda Park Bridge (66)	Japan	1993	cable stayed; steel box-girder deck	179	0.9	$\leq$2000 pedestrians (2.1 ped/m^2); amplitude in excess of 0.01m; increase of vibration frequency during moderate occupancy	$\leq$ 20% synchronized pedestrians estimated from video analysis
Léopold-Sédar-Senghor Footbridge (67)	France	1999	shallow steel arch	140	0.81	exponential growth once amplitude reached 0.1 to 0.15 m/s^2	no evidence
London Millennium Bridge (29; 24)	UK	2000	shallow suspension	325	0.5, 0.8, 1.0	1.3 to 1.5 ped/m^2; 1.86 to 2.45 m/s^2 max acceleration; pedestrians alternately tuned and detuned their pace with lateral bridge motion	no direct evidence; vertical pedestrian force random while lateral force correlated with bridge motion
Lardal Footbridge (68)	Norway	2001	shallow glue-laminated timber arch	91 and two approach spans of 13	0.83	$>$ 1m/s^2 for 40 pedestrians	no evidence; evidence of saturation (self-limiting) effect
Maple Valley Great Suspension Bridge (69)	Japan	2002	suspension	440	0.88, 1.02	0.045m max displacement (1.35m/s^2); $0.7 - 1.3$ped/m^2	frequency synchronization and "*tuned and not tuned*" effect from accelerometers on pedestrians' waist
Geneva Airport Footbridge (70)	Switzerland	2002	reinforced concrete multi-span	94.5	1.0	one-directional traffic; "bordered on panic" while rapidly evacuating bridge	no evidence
Changi Mezzanine Bridge (28)	Singapore	2002	shallow steel arch	140	0.9	0.055m (0.17m/s^2)	no evidence
Clifton Suspension Bridge (27)	UK	2003	suspension	214	0.53, 0.77	1.1 ped/m^2; max 0.2 m/s^2 = 0.011m	evidence of a lack of synchronization
Pedro and Ines Footbridge (30)	Protugal	2006	multispan with shall steel main arch	275	0.91	abrupt amplitude increase once critical number of pedestrian reached; max 0.2 m/s^2 for 73 ped and 1.2 m/s$^=$ 0.04m for 145 ped	no evidence
Simone de Beauvoir Footbridge (71)	France	2006	shallow arch with tension links	304	0.56, 1.12	0.03m for 80-100 pedestrians with 20 synced; 0.06m for 60 synced pedestrians	tests with imposed synchrony showed saturation effect
Cragside Bridge (72)	UK	2006	wrought iron arch	69	2.8	increase of vibration frequency during pedestrian loading; max. acceleration amplitude 13.9 m/s^2 for 9 pedestrians walking at 110 steps/min	tested under intentional synchronization
Weil-am-Rhein Footbridge (73)	Switzerland	2007	arch	230	0.95	1.7m/s^2 = approx. 0.08m peak-to-peak with 800 people	limited tuning effect during crowd load testing and argued to propagate in the crowd
Squibb Park Bridge (63; 74)	USA	2013	underslung suspension	122	0.84	N/A	N/A
Luiz I Bridge (75)	Portugal	2020	double-deck metallic truss incorporating parabolic arch	391.5, 172	0.73, 0.95	instability can be triggered independently at two vibration modes	no evidence

Table 2.2 Reported cases of lateral bridge instability due to the action of walking pedestrians. The final column documents any evidence presented for pedestrian synchronization.

Bridge	Country	Year	Observation
Angers Bridge (77)	France	1850	collapsed while a battalion of soldiers was marching across the bridge, killing 226 of them; the bridge movement "involuntarily gave the soldier a certain cadence"
Brooklyn Bridge (78)	USA	1880	swaying of catwalks during construction
Wuhan Yangtze Bridge (79)	China	1957	
Kiev suspension bridge (80)	Ukraine	1958	
Bosporus Bridge, Istanbul (60)	Turkey	1973	100,000 pedestrians on opening day caused it to sway
Auckland Harbour Bridge (29)	New Zealand	1975	0.67Hz oscillation during public demonstration
Groves Bridge, Chester (29)	UK	1977	100m suspension bridge filled with rowing regatta spectators
NEC, Birmingham (29)	UK	1990	0.7Hz oscillations of 45m bridge linking exhibition center to railway station after major events
Expo 1998 footbridges, Lisbon (81)	Portugal	1998	"acceleration in horizontal vibrations can go over adequate limits with just a few pedestrians."
Alexandra Bridge, Ottawa (24)	Canada	2000	crowd due to firework display
Brooklyn Bridge (61)	USA	2003	"Packed shoulder to shoulder with pedestrians" during blackout; "feeling seasick, having to weave as they walked", couldn't keep balance if stood still.
Bosphorus Bridge, Istambul (82)	Turkey	2010	
Bassac River Bridge (83)	Cambodia	2010	456 people died in stampede after panic caused by swaying of bridge filled with over 7000 pedestrians trying to reach popular water festival
Westminster Bridge, London (44)	UK	2010	
Brooklyn Bridge (44)	USA	2011	

Table 2.3 Other reported instances of lateral pedestrian-induced bridge vibrations

are prone to significant lateral swaying which forces trekkers to adjust their footsteps, often in an unusual way (see the videos corresponding to Fig. 2.5 a,b; yet, showing no signs of trekkers' phase locking. (see the video corresponding to Fig. 2.5 c).

2.4 Numerical simulation algorithms

2.4.1 Procedure for adding pedestrians on the bridge

Initially, the bridge-pedestrian system is simulated with a crowd size of two pedestrians. Every T_{add} of simulation time, we insert another pedestrian with a uniform random phase initialized to the limit cycle solution of the model in the absence of bridge motion. Subsequently, we advance the simulation T_{add} seconds, compute the maximum amplitude of the bridge and mean order parameter of the pedestrians over the current interval, and save the entire state.

Figure 2.5 Snapshots from videos of trekkers crossing unstable bridges over the Dudh Kosi River in Nepal. The full videos are available in the repository at `https://doi.org/10.5281/zenodo.8132826` (84). A trekker places the feet in a highly irregular manner in response to lateral bridge movement (video 2). Notice the right foot traveling beyond the midline of the body; this leg crossing is an uncommon behavior, yet, it is accounted for in Models 1 and 2. (b). A trekker with a heavy backpack struggles to move forward on a laterally unstable bridge and makes significant foot placement adjustments to maintain his balance (video 3). (c). A group of trekkers walking across the same unstable bridge as in (b), with no visible signs of phase-locking in their stepping behavior (video 4).

2.4.2 Implementation of negative damping criterion

2.4.2.1 To compute σ_1:

According to the formula (2.9), for each pedestrian, we need to compute

$$\overline{h}_u = \frac{1}{T}\int_0^T \frac{\partial H}{\partial u} dt,$$

where $u = \dot{x}$. We consider specifically the case of Models 1 and 3 in which there is a jump in the force H. Specifically, for $t \in (t_{s-1}, t_{s+1})$ we can write

$$H(y,t) = H_s(y) + \Theta(t - t_s)J(y; t_s, t_{s-1}),$$

where Θ is the Heaviside step function,

$$H_s(y) = \sqrt{\frac{g}{L}}\left(p(t_{s-1}) - y(t)\right), \quad J(y(t); y(t_s), \dot{y}(t_s)) = \sqrt{\frac{g}{L}}\left(p(t_s) - p(t_{s-1})\right)$$

and $p(t_s)$ is given by (2.13). The principle is easily generalized to any function $H(y)$ with jumps at $t = t_s$. That is H during step s is given by H_s and $J = H_{s+1} - H_s$. For the specific case of Models 1 and 2, we can write

$$\frac{\partial H}{\partial u} = \frac{\partial H}{\partial y}\frac{\partial y}{\partial u} + \frac{\partial H}{\partial t}\ddot{x} = -\sqrt{\frac{g}{L}}\frac{\partial y}{\partial u} \quad + \delta(t - t_s)J(t_s)\ddot{x}.$$

Hence

$$\int_{t_{s-1}^+}^{t_{s+1}^-} \frac{\partial H}{\partial u} dt = J(y(t_s)) - \int_{t_{s-1}^+}^{t_{s+1}^-} \sqrt{\frac{g}{L}}\frac{\partial y}{\partial u}(t)\, dt\,.$$

That is, we compute the integral as if the singularity were absent, provided we add an extra jump term every time $t = t_s$ for $s = 1, 2, 3, \ldots$

It remains to show how to compute

$$\eta(t) := \frac{\partial y}{\partial u}(t) \qquad \text{from } t_{s-1} \text{ to } t_s.$$

Note that

$$\begin{aligned}\frac{d^2\eta}{dt^2} &= \frac{\partial}{\partial u}\left(\frac{\partial^2 y}{\partial t^2}\right) \\ &= \frac{\partial}{\partial u}\left(-\ddot{x} - H(y)\right) = -\dot{u}\,\ddot{u} - \frac{\partial H}{\partial y}\eta\end{aligned}$$

Hence $\eta(t)$ satisfies the variational differential equation

$$\ddot{\eta} = \sqrt{\frac{g}{L}}\eta - \ddot{u}\,\dot{u} \qquad \text{subject to} \quad \eta(t_{s-1}) = \dot{\eta}(t_{s-1}) = 0. \tag{2.32}$$

To compute σ_1 for the particular case of Model 3, we proceed similarly; in this case, note that the jump term

$$\begin{aligned}\delta(y)J(y,\dot{y},t) = \delta(y)\lim_{\varepsilon\to 0}[&-\lambda^2\nu^2\dot{y}(y(t+\varepsilon) - p_c\text{sgn}(y(t+\varepsilon)))^2 + \frac{g}{L}(y(t+\varepsilon) - p_c\text{sgn}(y(t+\varepsilon))) \\ &+ \lambda^2\nu^2(y - p_c\text{sgn}(y(t))^2 - \frac{g}{L}(y(t) - p_c\text{sgn}(y(t)))] \\ &= \delta(y)\left[4\lambda^2\nu^2 p_c y\text{sgn}(\dot{y}) + 2\frac{g}{L}\left(p_c\text{sgn}(\dot{y})\right)\right] = \delta(y)\left[2\frac{g}{L}\left(p_c\text{sgn}(\dot{y})\right)\right]\end{aligned}$$

is due to the presence of the discontinuity in $\text{sgn}(y)$ at each step and takes a similar form to that of the jump term of Model 1 in the case of a fixed step width p_c.

Likewise, we may formulate a second-order variational equation, akin to (2.32), for the continuous part of $H_{\dot{x}}$ using the multivariate chain rule:

$$\dot{\eta}(t) - \dot{u}\ddot{u} - \eta H_y - \dot{\eta} H_{\dot{y}}, \quad \text{where} \quad \eta(t) = \partial_u y(t).$$

Within the pedestrian step, $H_{\dot{y}}$ and H_y have closed form and may be used to compute σ_2 (see next section).

2.4.2.2 To compute σ_2.

Here, according to (2.10), for each pedestrian we need to compute y_c y_s, $\overline{h}_y$ and $\overline{h}_v$. By definition, y_c and y_s would be the Fourier cosine and since coefficients, respectively, of the $O(\varepsilon)$ components of $y(t)$, assuming that the dominant bridge motion is $\cos(\Omega t)$. Thus, given bridge motion $x(t)$, we have

$$y_c = \frac{1}{AT}\int_0^T x(t)y(t)dt, \qquad y_s = -\frac{1}{\Omega AT}\int_0^T \dot{x}(t)y(t)dt, \tag{2.33}$$

where

$$A^2 = \int_0^{2\pi/\Omega} x(t)^2 dt.$$

The integrals $\overline{h}_y$ and $\overline{h}_v$ can be computed similarly to $\overline{h}_u$. In particular, for Models 1 and 3 we get

$$\frac{\partial H}{\partial y} = \frac{dH}{dy} + \frac{\partial H}{\partial t}\ddot{y} = -\sqrt{\frac{g}{L}} + \delta(t - t_s)J(t_s)\dot{y}.$$

and

$$\frac{\partial H}{\partial v} = \frac{\partial H}{\partial y}\frac{\partial y}{\partial v} \; + \; \frac{\partial H}{\partial t}\ddot{y} = \; 0 \; + \; \delta(t - t_s)J(t_s)\ddot{y}.$$

Hence

$$\overline{h}(y) = -\sqrt{\frac{g}{L}} + \frac{1}{t_s - t_{s-1}}J(t_s)\dot{y}(t_s)$$

and

$$\overline{h}u = \frac{1}{t_s - t_{s-1}}J(t_s)\ddot{y}(t_s).$$

2.4.2.3 To compute σ_3.

The formula (2.11) requires evaluation of the forward dynamics, specifically a function $z^{(i)}(t)$ which is the fluctuating component of the pedestrian's progress $Z^{(i)}(t)$:

$$Z^{(i)}(t) = \chi^{(i)}t + z^{(i)}(t),$$

where $\chi^{(i)}$ is their average forward velocity. However, none of the models we consider have an equation for the forward dynamics $z(t)$.

Instead, Models 2 and 3 have an update rule that generates the time t_{s+1} as a function of t_s. Because everything is averaged over a cycle, we can assume that $z(t)$ is a continuous variable that at $t = t_s$ gives the perturbation to the position along the bridge span of the pedestrian's center of mass from its position if it were walking at a constant velocity.

That is, we can assume that for $t \in [t_s, t_{s+1})$

$$\dot{z} = w_s, \qquad \text{where} \quad w_s = \frac{t_{s-1} - 2t_s + t_{s+1}}{t_{s+1} - t_s}$$

So that z becomes a piecewise-linear function of time

$$z(t) = z(t_s) + w_s(t - t_s), \qquad \text{for } t_s \le t < t_s + 1.$$

Then the functions z_c and z_s can be defined similarly to y_c and y_s above. Also

$$\frac{\partial H}{\partial z} = \frac{\partial H}{\partial t}\frac{\partial z}{\partial t} = \delta(t - t_s)J(t_s)w \quad \text{for} \quad t = t_s$$

and

$$\frac{\partial H}{\partial w} = \frac{\partial H}{\partial t}\frac{\partial w}{\partial t} = 0.$$

Hence we need only consider the first term in (2.11), which necessitates computation of z_s. Analogously with (2.33), we have

$$\overline{z_s} = -\frac{1}{\Omega A T}\int_{t_{s-1}}^{t_s} w\dot{x}(t)t dt.$$

2.4.3 Calculations of the scatter plots and analytical curves in Fig. 2.4.

To generate the scatter plots Fig. 2.4 (top) and estimate average damping coefficient per pedestrian $\bar{\sigma}_1$, we simulate the pedestrian system (2.5) with imposed bridge motion, wherein the bridge is taken to be sinusoidal at its natural frequency, i.e.,

$$x(t) = X_b \sin \Omega t,$$

where $X_b = 0.006$m is the constant amplitude of the imposed bridge motion. Only a single pedestrian is placed on the bridge and the bridge's acceleration is fixed to the second derivative of the sinusoid: $\ddot{x} = -X_b\Omega^2 \sin \Omega t$, regardless of the pedestrian's foot force. This scenario accurately describes the case for a low mass ratio, such that the force from an

individual pedestrian has a negligible effect on the structure. We perform this simulation over a uniform 1200-point grid of the pedestrian and bridge natural frequencies $\overline{\omega}$ and Ω, where Ω ranges from 0.25 to 3 Hz and $\overline{\omega}$ ranges from 0.6 to 1.2 Hz. At each point of the grid, we compute the mean frequency of the pedestrian and bridge movement. This numerically calculated frequency ratio $[\Omega/\overline{\omega}]$ may differ from the ratio of natural frequencies $\Omega/\overline{\omega}$ due to the adaptation of the pedestrian stride to the bridge motion. This yields the non-uniform distribution of the blue points across parameter range $[\Omega/\overline{\omega}]$ in Fig. 2.4 as the pedestrian frequency adaptation promotes specific frequency ratios of $[\Omega/\overline{\omega}]$.

For each frequency ratio $[\Omega/\overline{\omega}]$, we numerically compute the force $H^{(1)}$ by simulating equation (2.5) for $\ddot{y_1}$ over 50 footsteps for Model 2 and over 10,000 steps for Model 3. To calculate the component of $H^{(1)}$ in phase with the bridge velocity, we modify the formula (9) from Ref. (39) so that:

$$\hat{H}_u = \frac{2}{t_s - t_{s-1}} \int_0^{t_s - t_{s-1}} H^{(1)} \cos \Omega t \, dt.$$

We then compute a histogram of $\hat{H}_u$, parametrized by a discrete set of numerically computed phase offsets φ of the pedestrian step time. To calculate the damping coefficient of the pedestrian, we use the histogram to calculate the expected value $\overline{H}_u$ of $\hat{H}_u$ for each φ. We then apply the scaling in equation (11) from Ref. (39) to the resulting average. Thus, we obtain the following formula:

$$\overline{\sigma}_1 = -\frac{\overline{H}_u}{X_b \Omega} = -\frac{1}{\pi X_b \Omega (t_s - t_{s-1})} \int_0^{2\pi} \int_0^{t_s - t_{s-1}} H^{(1)} \cos \Omega t \, dt d\varphi. \tag{2.34}$$

This formula is used to generate the blue points in the top panels of Fig. 2.4 for different frequency ratios $[\Omega/\overline{\omega}]$.

The analytical curve for $\overline{\sigma}_1$ depicted by the red solid line in the top plot of Fig. 2.4 is calculated via

$$\overline{\sigma}_1 = \frac{mg}{L} A^2 \frac{\overline{\omega}}{\pi\Omega^2} \left(\Omega \sqrt{\frac{L}{g}} a_2 - a_1 \right), \tag{2.35}$$

where $A = \frac{1}{1+\frac{g}{L\Omega^2}}$, $a_1 = 1 - e^{\frac{\pi}{\overline{\omega}}\sqrt{\frac{g}{L}}} \cos\frac{\pi\Omega}{\overline{\omega}}$, and $a_2 = e^{\frac{\pi}{\overline{\omega}}\sqrt{\frac{g}{L}}} \sin\frac{\pi\Omega}{\overline{\omega}}$.

Formula (2.35), derived in Ref. (50), estimates the negative damping contribution of the pedestrian described by Model 1. As a result, it does not account for the effect of step timing adaptation and, therefore, it yields estimates that differ from the numerical results for Model 2 that does allow for pedestrian step timing adaptation.

We also use formula (2.35) to estimate the critical crowd N_{crit} as a function of the frequency ratio Ω/ω (the red solid line in the bottom plot of Fig. 2.4). This is done by setting the total bridge damping $c_T = 0$ in (2.8) and neglecting the other damping coefficients $\sigma_2^{(i)}$ and $\sigma_3^{(i)}$. Therefore, we obtain the condition

$$2\varepsilon\zeta\Omega + N_{crit}\overline{\sigma}_1/M = 0$$

which yields the following estimate for the critical crowd size

$$N_{crit} = -\frac{2\varepsilon\zeta\Omega M}{\overline{\sigma_1}}, \tag{2.36}$$

where $\overline{\sigma_1}$ is estimated via (2.35). The analytical expression (2.36) estimates the critical number of pedestrians N_{crit} described by Model 1 rather precisely. However, it becomes less accurate for pedestrians described by Model 2, and especially by Model 3 in which $\sigma_2^{(i)}$ plays a significant role (see Fig. 2.6 for the comparison of the damping terms $\sigma_1^{(i)}$ and $\sigma_2^{(i)}$ in Models 2 and 3). For this reason, the curve (2.36) is plotted in the bottom plot of Fig. 2.4 for Model 2 only.

2.5 Further simulation results

2.5.1 Faster addition of pedestrians to the bridge

To better understand the role of time interval T_{add} at which pedestrians are added sequentially to the bridge, we perform numerical simulations similar to those reported in Fig. 2.3 but with shorter time interval $T_{\text{add}} = 10$ s. In this case, the pedestrian-bridge system has a narrower time window for transient effects before the addition of the next pedestrian. As a result, one can expect that the crowd will have grown larger by the time the vibrations have increased in amplitude significantly. The simulations displayed in Fig. 2.6 confirm this intuition and indicate the widening of the instability region (pink) preceding the onset of weak (Model 2) and strong synchronization (Model 3).

To elucidate the contributions of the damping per pedestrian terms σ_1, σ_2, σ_3, we also include the additional plots (third row) in Fig. 2.6. These plots indicate that while σ_1 is the only factor that matters for the onset of instability for non-adaptive Model 1, σ_2, that accounts for the adjustment of pedestrian lateral gait timing contributes to the overall

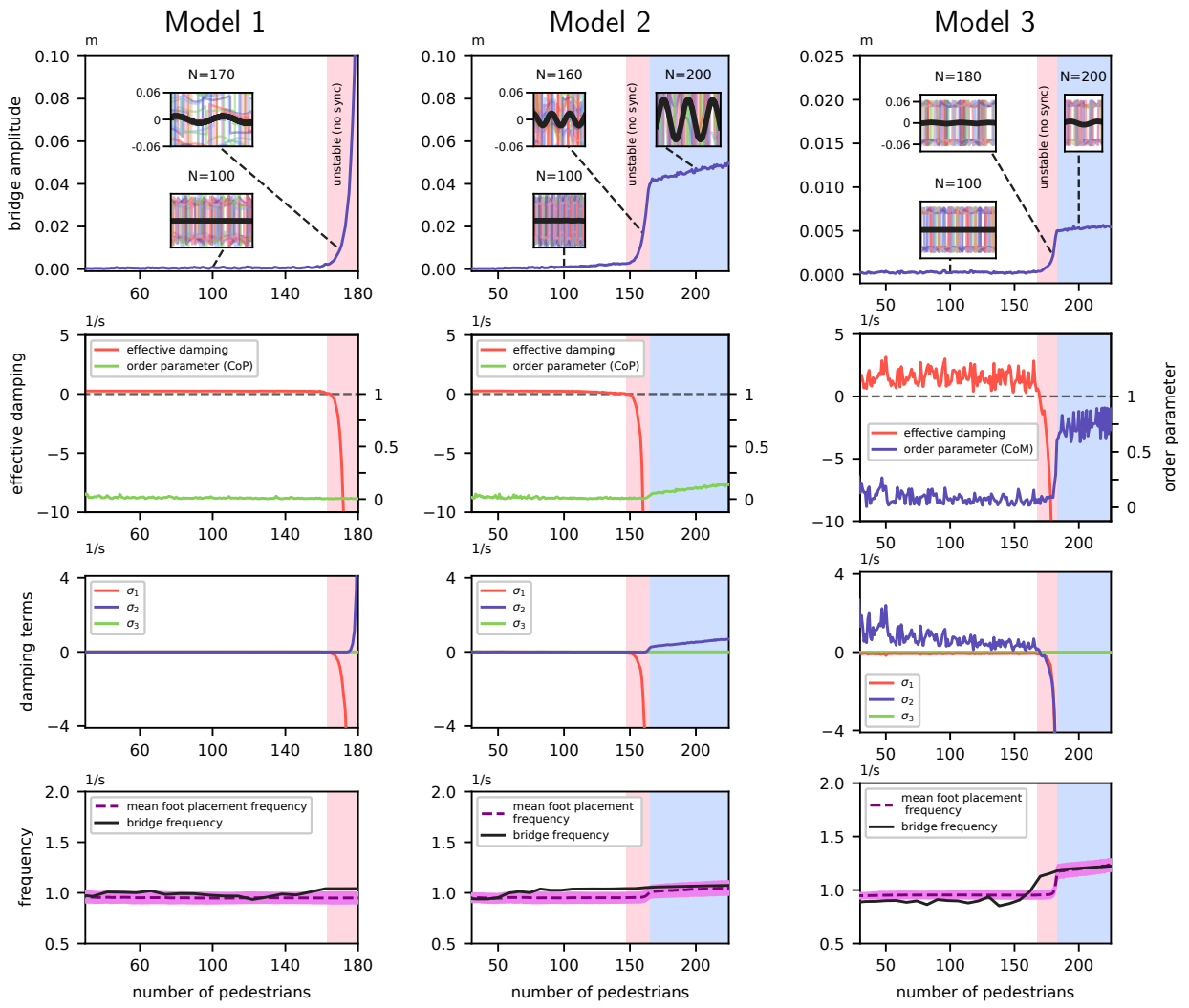

Figure 2.6 Simulations as in Fig. 2.3 in which pedestrians are sequentially added at shorter intervals of $T_{\text{add}} = 10$ s. Notice the widening of the pink region corresponding to the onset of bridge oscillations without pedestrian phase locking. The plots for the damping per pedestrian terms $\sigma_{1,2,3}$ specify the contribution of each term to the onset of bridge instability. Other parameters are as in Fig. 2.3.

negative damping to a lesser (Model 2) and greater (Model 3) degree. In all cases, we find that σ_3 makes little contribution.

2.5.2 Extreme worst-case, complete resonance

To provide further evidence that pedestrian synchronization is not necessary for bridge instability, we consider the worst-case scenario in which the pedestrians have identical natural stride frequency ω (but initially random phase) which also coincides with the bridge natural frequency Ω. It seems to be the ideal resonance scenario for the emergence of synchronization among the pedestrians and with the bridge, and therefore, we could expect synchronization to emerge at smaller crowd sizes and coincide with the onset of bridge instability. We also note that this case violates the central assumption (2.21) that underlies the above asymptotic derivation of the coefficients $\sigma_{1,2,3}$.

The results are shown in Fig. 2.7. Despite violating the above assumption, observe that the negative damping criterion still predicts the onset of bridge instability for Models 1 and 2. In particular, note how significant synchronization now occurs, after the onset of large vibrations, for Model 2 with an order parameter greater than 0.5. For Model 3, the negative damping criterion no longer provides accurate information, but note that the large increase in negative damping at around 50-100 pedestrians precedes the onset of significant instability, which in this case leads to complete synchrony (order parameter $r = 1$).

2.6 Role of different foot placement strategies: negative vs. positive damping

Figure 2.1 explains how bridge motion breaks the symmetry of the loading applied by mirror-imaged walkers such that long-term averages need not equal zero. That is only part of the explanation, as it does not consider the motion of the walkers' centers of mass nor the

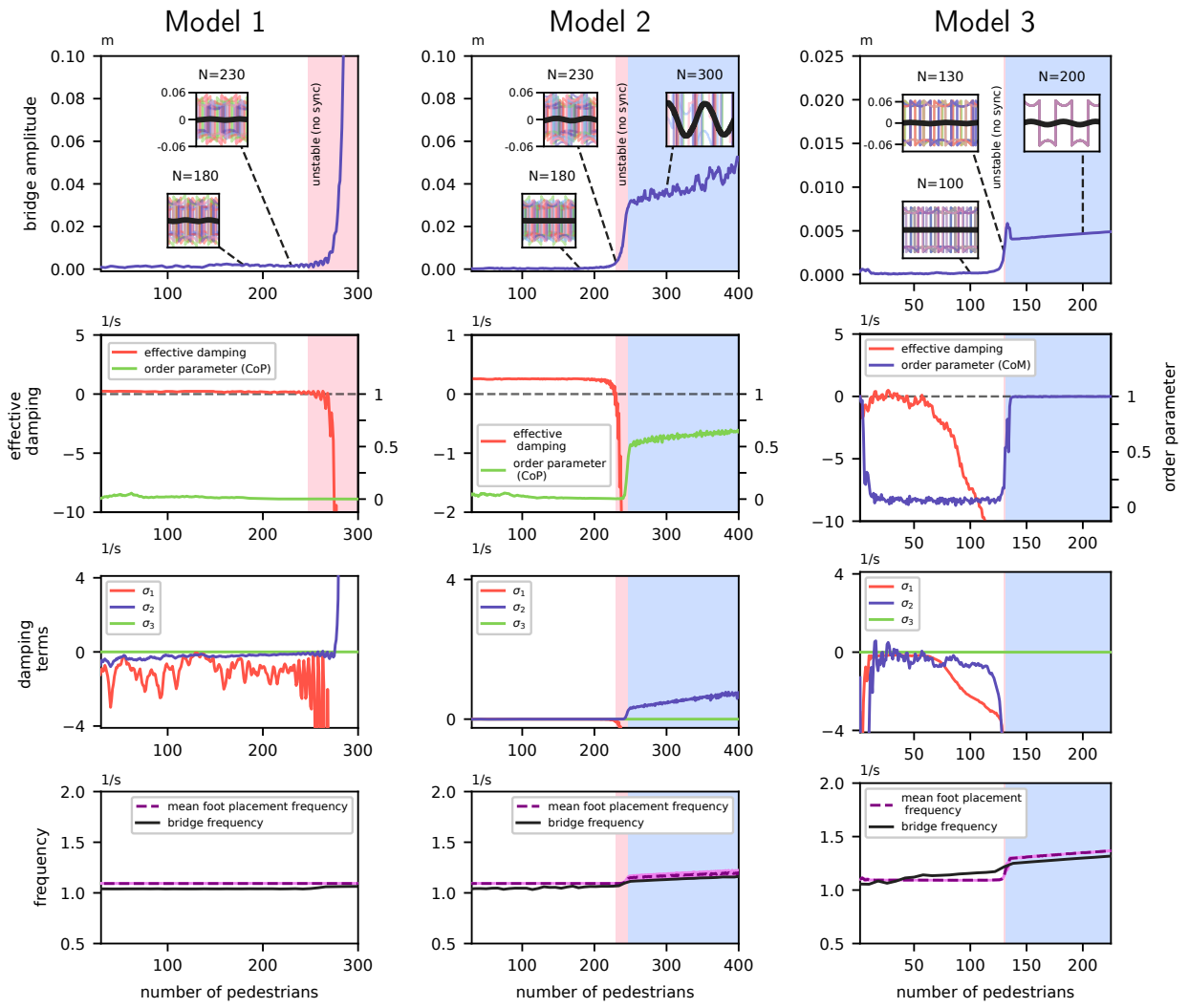

Figure 2.7 Simulations as in Fig. 2.3 but for the worst-case scenario of pedestrians with the same $\omega = 5.655$ rad/s ($S.D. = 0$) and the perfect resonance ratio $\Omega/\omega = 1$ corresponding to the yellow point in Fig. 2.4. Other parameters are as in Fig. 2.3 .

various foot placement strategies that may be adopted to maintain balance. On a stationary platform, a walker's center of mass will oscillate laterally with a dominant component at half the footfall frequency. Without changing the footfall frequency, the platform motion introduces a second frequency inducing the walker to adopt a two-frequency quasiperiodic pattern of footfall placement (Fig. 2.8).

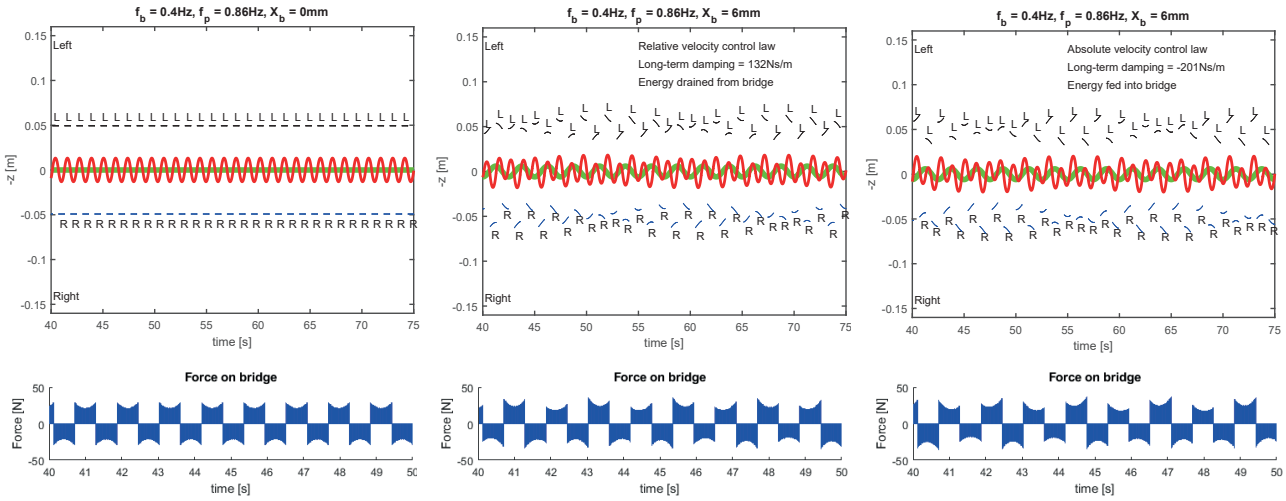

Figure 2.8 Upper figures show foot placement patterns (short black lines left foot, short blue lines right foot) for Model 1. The first is for a stationary platform, whilst the second and third are for a bridge oscillating at 6 mm amplitude at 0.4 Hz, with walkers adopting Hof's balance laws based on relative and absolute velocity respectively. The bridge motions induce quasiperiodic placement patterns. (The walker's center of mass and the bridge displacements are shown in red and green respectively.) The lower figures show the corresponding forces applied to the bridge.

The corresponding forces applied to the bridge in these three cases are also shown in Fig. 2.8. Since bridge motions are small, the forces are similar in all three cases. By taking the difference in forces, Fig. 2.9 highlights the small change in the applied forces that are the result of the bridge motion and correlates these with bridge velocity. The walker adopting the relative velocity control law creates forces that are negatively correlated with bridge velocity, leading to a positive damping effect. By contrast, the additional forces generated by the walker adopting the absolute velocity balance law are positively correlated with the bridge velocity, leading to the negative damping effect which feeds energy into the bridge.

In summary, the bridge motions cause the walkers to adjust their foot placements which induces small quasiperiodic forces which have a component at the bridge frequency. Depend-

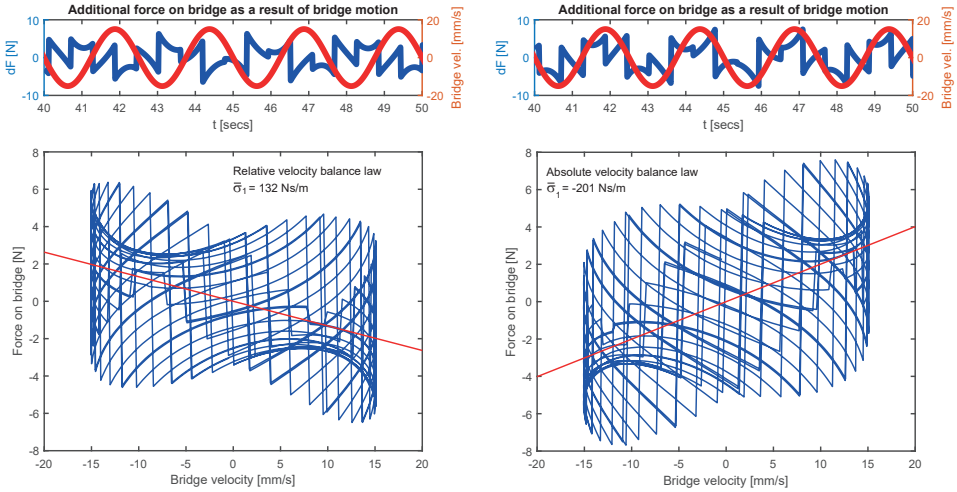

Figure 2.9 Upper: the change in forces that are the result of the bridge motions for the walkers of Fig. 2.8. The bridge velocity is shown in red. Lower: the correlation between the bridge velocity and the induced forces. The red lines indicate the average effective damping coefficient $\overline{\sigma}_1$.

ing on the balance law adopted (and the frequency of bridge motion and other parameters), the phases of these additional forces can either add or extract energy to/from the bridge. Experimental evidence is limited as to which balance law is more realistic for a walker on a moving platform, but the laboratory experiments augmented with Virtual Reality by Bocian *et al.* (42) provide some evidence for the absolute velocity control law. Walkers following either law could be present on the bridge. Also, the energy flows vary within different regimes of the quasiperiodic motions, such that the short-term effective damping may vary markedly from its theoretical long-term average value. Bridge designers should thus be aware that there could be dangerous instances of the negative damping effect at any bridge frequency.

This is the underlying cause of the instability of footbridges and does not entail walkers making any change to the frequency of their footsteps. Instead, gait widths are amplitude

modulated, introducing complicated phase relationships between foot placements and bridge motions, many of which have the effect of negative damping and feed energy into the bridge.

As bridge amplitudes grow, adjustment of footfall timing is an additional possibility and this is included in Models 2 and 3. Potential outcomes include the now-classical Kuramoto transition to synchronization, as well as phase-pulling phenomena where footfalls do not fully synchronize to the bridge motions but spend proportionally longer at some relative phase offsets (42; 44). Walkers who synchronize or exhibit phase pulling can add differing amounts of energy to the bridge, depending on how their footfall phases relate to that of the bridge's velocity. Phase synchronization can be triggered by bridge motions excited by the more fundamental mechanism of amplitude-modulated gait width, and this can lead to dangerous amplification of the bridge motions. It may also be noted that there exist parameter regimes where walkers synchronize at phases that lead to energy absorption or where they synchronize with a certain phase but the amplitude of the forcing does not grow indefinitely with the bridge amplitude, thereby limiting the bridge response. However, there is insufficient evidence for this to be relied upon in bridge design.

CHAPTER 3

Crowd Heterogeneity-Induced Instabilities of Footbridges

3.1 Introduction

The effects of human-to-human interaction on the onset of instability described in the previous chapter need to be better understood. Here, we demonstrate, using the model introduced in (85) and modified to fit actual crowd data in (86), that, in the presence of such interactions, increased heterogeneity of the footfall frequencies of a crowd of pedestrians on the bridge (described via (2.12)) both significantly decreases the threshold crowd size required for the onset of high-amplitude lateral vibration and leads to more significant partial synchronization of the pedestrians' step.

3.2 Methods

We further augment the inverted pendulum model introduced in Chapter 2, specifically (2.12) with the step timing adaptation in (2.14): first, we introduce the variable $z = [z_x, z_y]^T$ of the pedestrian with sagittal COM position z_x and lateral position z_y. The base stride frequency ω of each pedestrian then becomes time-dependent and is given by the polynomial relationship observed in (87):

$$\omega = f_v(|| \dot{z} ||) = \pi \left(0.35 || \dot{z} ||^3 - 1.59 || \dot{z} ||^2 + 2.93 || \dot{z} ||\right). \tag{3.1}$$

Then in (2.14) the unperturbed step width

$$Y_0 = \frac{| b_{min} |}{1 - \tanh(\frac{\sqrt{\frac{g}{L}}}{\frac{2}{\pi}\omega})}$$

and length

$$X_0 = \frac{0.36}{f^{-1}(E[\omega_0])}\dot{z}_x$$

are computed from the frequency and velocity. Exactly as in (86), z_i, the position of the ith pedestrian, evolves according to the second-order system:

$$\ddot{z}_i = \alpha(v_{d,i}[0,1]^T - \dot{z}_i) + \sum_{j=1, i\neq j}^{N} F_{dyn}(z_j - z_i)[z_i - z_j] + \sum_{w\in\{1,-1\}} F_{stat}(z_{x,i} + wz_{wall})[w, 0]^T,$$

where α is a relaxation coefficient, $v_{d,i}$ denotes the desired forward velocity of the ith pedestrian, z_{wall} is equal to half the bridge's span width (2 meters in the case of the London Millennium Bridge), and

$$F_{dyn}(\Delta) = A_d e^{\frac{2r-||\Delta||}{B_d}}\left(\lambda + \frac{1-\lambda}{2}(1 + \Delta\cdot[0,1]^T)\right) + Ae^{\frac{2r-||\Delta||}{B_d}} \tag{3.2}$$

denotes the repulsive force on each pedestrian due to close interactions with other visible pedestrians, and finally

$$F_{stat}(d) = A_s e^{\frac{r-|d|}{B_s}} \tag{3.3}$$

denotes the repulsive force due to the bridge railings as a function of their lateral displacement d from the pedestrian COM (whose position is given by $[z_x, z_y + y]$). The parameter r determines the desired avoidance radius of each pedestrian. The first term of F_{dyn} corresponds to the repulsion of the pedestrian COM due to other pedestrians in its visual field,

while the second term is due to very short-range interactions such as shoving. The remaining system is unchanged from Chapter 2. Figure 3.1 depicts the magnitude of F_{dyn} as a function of the position offset of pedestrian j from pedestrian i. The simulated bridge and crowd were initialized using the same parameters and distributions as in Table 2.1. The Newton-Raphson method was used to determine $v_{d,i} = f_v^{-1}(\omega_i)$ from the initial pedestrian frequency determined as in the previous chapter, which was also taken to be the initial forward velocity of each pedestrian. For a range of crowd sizes from 80 to 190 pedestrians and for a range of standard deviations σ of the pedestrian frequency in Hz from 0 (identical initial step frequencies) to 1 (extreme heterogeneity), the pedestrian-bridge system with social force was simulated for 20 seconds for each of 10 trials in which the initial crowd density, frequency, state, and other parameters were randomly sampled from the distributions in Table 2.1. The resulting bridge amplitudes and pedestrian step phase coherence were taken as distributions over the 10 trials.

3.3 Results

Figure 3.2 shows the dependence of the structure's lateral vibration amplitude and the Kuramoto order parameter of the pedestrian step phase on both the number N of pedestrians and the heterogeneity σ of the initial crowd stepping frequency. Figure 3.3 displays the simultaneous dependence of the sample mean amplitude (over each trial) on both crowd size and σ. It can be observed that the threshold effect for N reported in the previous chapter and similarly in (26) exhibits a bistability that, in the presence of social force terms,

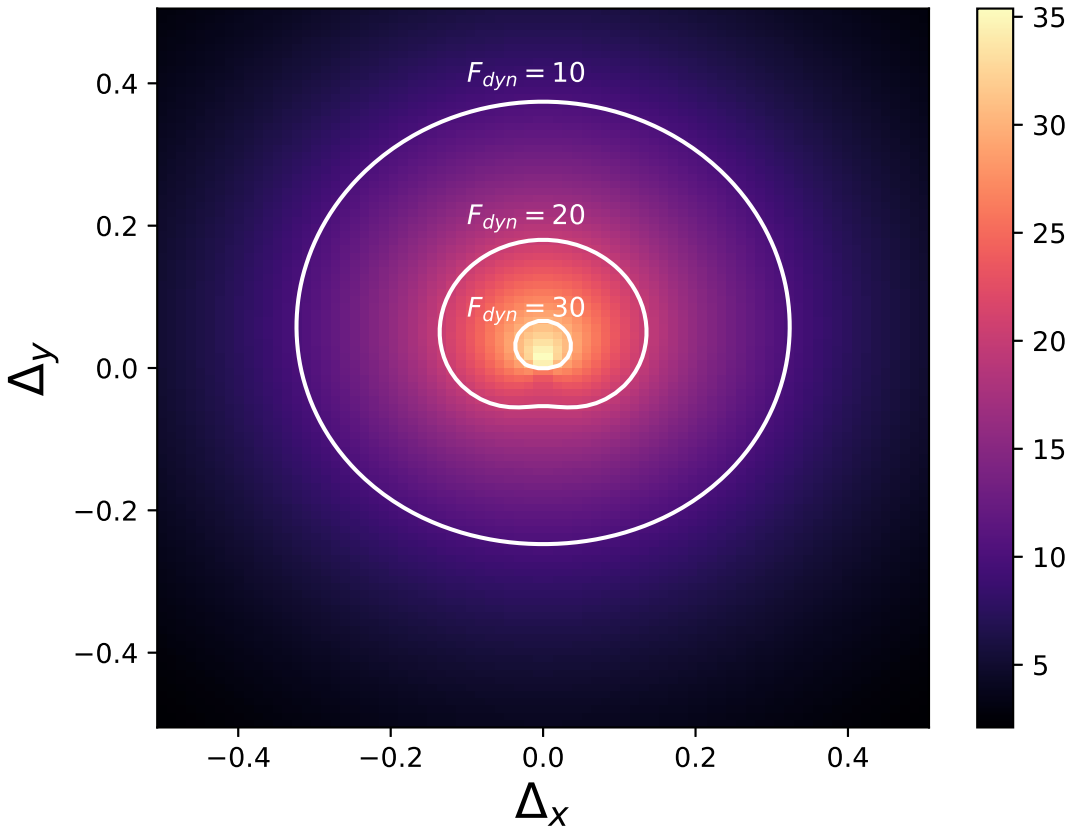

Figure 3.1 Dynamic social forcing term F_{dyn} on the pedestrian i from pedestrian j as a function of lateral (horizontal axis) and sagittal (vertical axis) displacement of the jth pedestrian from the ith one. The effect of the ith pedestrian's visual field is determined by the parameter λ. Here, as in (86), $A_d = 1.7, r = 0.31, B_d = 0.28, \alpha = 0.5, B_s = 0.1, A_s = 5$, and $\lambda = 0.31$; these parameters are used throughout this chapter.

becomes highly sensitive to the precise initial configuration of the crowd. Furthermore, the instability is more frequent and more pronounced when the pedestrian frequencies are nonidentical; that is, the median amplitude and phase coherence of the bridge and crowd grow directly as σ. This counter-intuitive phenomenon, an instance of disorder-promoted synchrony, can be explained using the frequency-damping relationship depicted in Fig. 2.4.

The degree of tuning of pedestrians to the bridge frequency is modified by the presence of slow pedestrians, which inhibit the velocity of the pedestrians behind them in a cascade, leading to a transient regime of "traffic jams" resulting in an adjustment of the pedestrian velocity, and thus (by the relationship in (3.1)) a tuning of the pedestrian frequency to one at which the pedestrians give more energy to the bridge. The precise temporal evolution of the bridge-pedestrian frequency ratio distribution for a set of sample trajectories that exhibit lateral vibration and its relationship to the curve in Fig. 2.4, is depicted in Fig. 3.4a as a ridge plot. A noticeable shift occurs directly before the onset of high-amplitude oscillation of the bridge displacement, depicted in Fig. 3.4b over the same duration. This shift causes most pedestrians to be negative dampers (i.e., having tuning ratios between the two vertical lines). It places the most frequent pedestrian tuning ratio at almost exactly the red dashed line, which corresponds to the most negative damping per pedestrian. After the bridge becomes sufficiently unstable, the step timing adaptation shifts the tuning ratio distribution to the left but still well within the regime of negative damping. The distribution of velocities due to the social force determines the frequency shift and thus the instability; these dynamics and the resulting traffic jams are depicted as snapshots in Fig. 3.5.

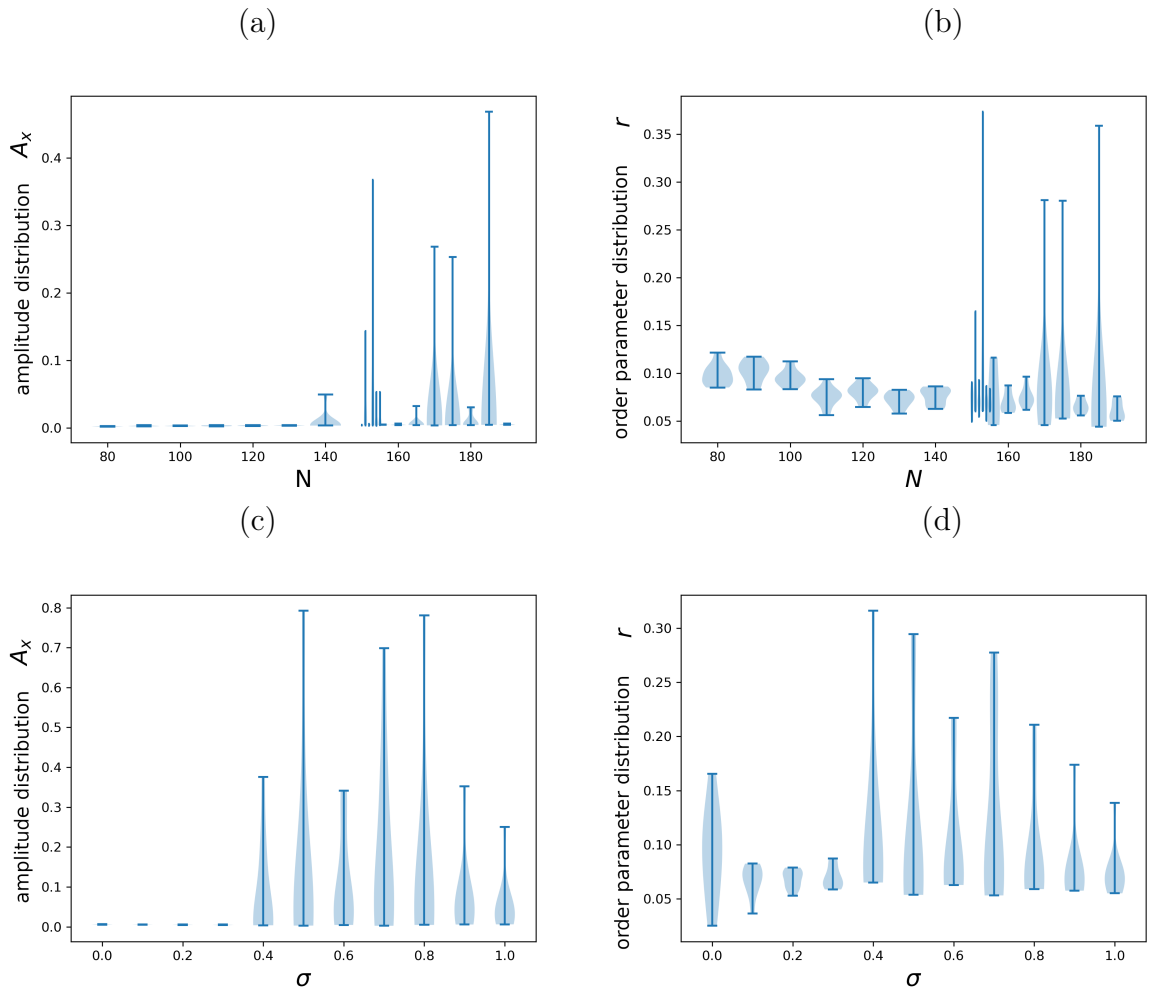

Figure 3.2 (a) Bridge displacement amplitude distributions and (b) distribution of the Kuramoto order parameter r of the phase $\frac{t-t_s}{t_{s+1}-t_s}$ of each pedestrian step as a function of number of pedestrians N, each over ten trials. The initial stride frequency standard deviation $\sigma = 0.3$ and all other initial crowd parameters are identical to those used in Fig. 2.3 except that the simulations were reinitialized for each N. Note that while the instability of the bridge and the resulting phase coherence are now highly dependent on the exact initial configuration of pedestrian positions and velocities due to the crowd forcing, the bridge instability may occur at values of N as low as $N = 140$ and lead to phase coherence at values as low as $N = 150$, which was not observed previously from an initially still bridge. Furthermore, (c) depicts the bridge amplitude distribution and (d) the distribution of Kuramoto order parameters as a function of σ when $N = 160$. Note the significant increase in mean and maximum order and amplitude as a function of the heterogeneity of the crowd initial frequency even when compared to the case of pedestrians with identical initial stride frequencies (i.e., when $\sigma = 0$), in which case the phase coherence is significant but not as pronounced as when $\sigma = 0.4$, and high-amplitude bridge movement is not observed.

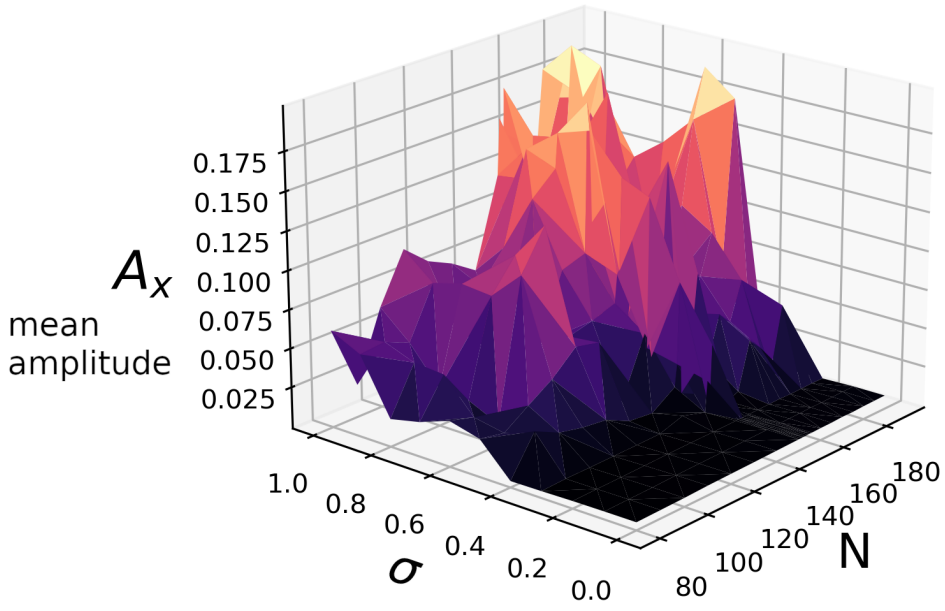

Figure 3.3 The general relationship described in Fig. 3.2a-d using a surface plot of the mean bridge displacement amplitude A_x (over each of ten sample initial conditions) as a function of σ, the standard deviation of the initial stride frequency distribution, and N, the number of pedestrians on the bridge.

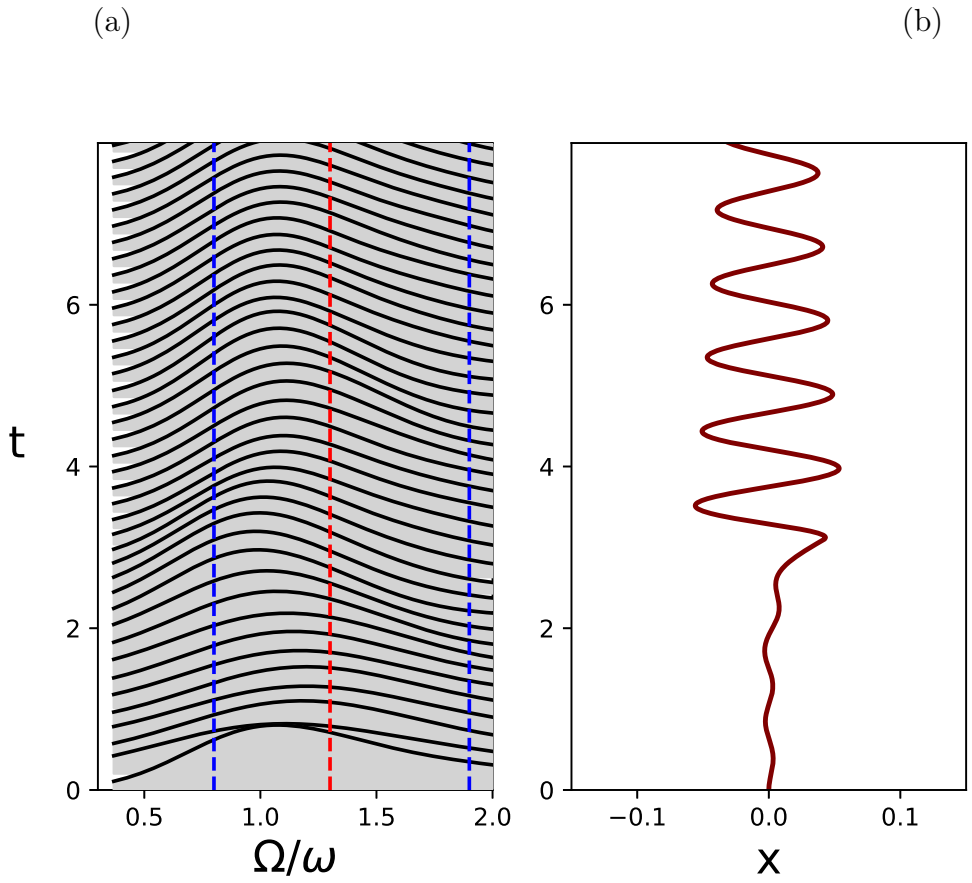

Figure 3.4 (a). Sample evolution of the distribution of bridge-to-pedestrian-stride frequency ratios Ω/ω sampled over time (vertical axis) using Gaussian kernel density estimation; blue dashed vertical lines correspond to roots and red dashed vertical line to the minimum of the curve in Fig. 2.4a. A shift in the frequency distribution occurs close to the instability due to the change in crowd velocity. (b). The corresponding bridge motion x. Parameters are $N = 155$, $\sigma = 0.5$.

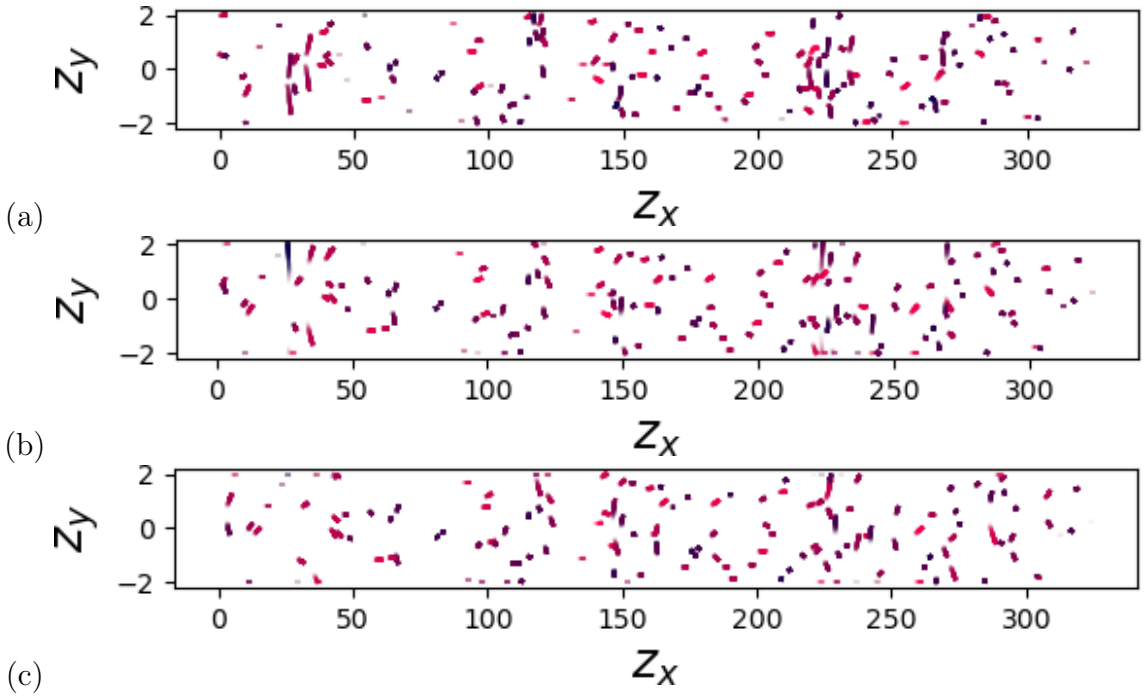

Figure 3.5 The trails of pedestrian motion are plotted in color according to sagittal velocity, with dark blue corresponding to no forward motion and bright pink being the maximum forward velocity (3.2 m/s), each for a duration of 1 second from (a) 0-1 seconds, (b) 1-2 seconds, and (c) 2-3 seconds. Note the formation of several "traffic jams" in the sagittal direction (horizontal axis).

CHAPTER 4

A data-driven model of invasive social pathogens from 4chan

4.1 Introduction

Online imageboards such as 4chan, 8kun, and others host various conspiracy theories and extremist far-right content (88; 89) and have outsized influence on mainstream social media and political opinion despite their relatively low traffic. Such imageboards are characterized by the ability to post anonymously and engage in discussion with other users in reply to their posts in the form of an extended comment thread beginning with an initial post; content is generally uncensored and promotes topics such as hate speech, misinformation, and harassment. The study of the viral spreading of memes on social media has primarily considered more mainstream networks such as Twitter and Facebook as isolated systems. Previous work suggests they are extremely prone to social contagion and epidemic; in (90), Kucharski estimates the basic reproduction number R_0 of Facebook to be between 1.9 and 2.5 for widely disseminated memes, while Skaza *et al.* have estimated that of Twitter to be between one and $1,000$ (91). In both instances, extreme viral spreading was found to occur. The basic reproduction number is a property of populations with uniform mixing and recovery; such epidemiological models may also be refined to incorporate network effects. For example, Jiang *et al.* (92; 93) and Zhang *et al.* (94) modeled social spreaders as replicator agents (95) playing a two-strategy evolutionary game on a network with non-uniform degree distribution and derived a mean-field ordinary differential equation for the evolution of the ratio of spreaders to the total population. In fact, for a particular form of the payoff function,

this viewpoint is dynamically similar to the epidemiological one since the dynamics of both the replicator model with uniform mixing and constant payoff and the classic SI model reduce to a low-order polynomial ODE with no constant term when, respectively, expanded and normalized to the population size (96). Conspiracy theories are known empirically to be highly intrinsically contagious phenomena; Jolley *et al.* (97) report that single exposure to a racial conspiracy theory promoted general bias towards multiple out-groups, while Van Der Linden *et al.* (98) found a similar effect with anti-climate-change conspiracy theories; further significant results were confirmed by Balafoutas *et al.* (99) though, in this case, they were not central to the results of the study. Cinelli *et al.* (100) have found COVID-19 misinformation highly contagious in social networks. The dynamical evolution of misinformation has been documented in (101), who likens its spread to a game of telephone, in which every message slightly differs from the previous.

Using time-series data of the occurrence of anti-vaccine keywords on /pol/ and Twitter, which we determine to be causally linked, we hypothesize an invasive epidemiological mechanism for spreading conspiracy theories from online imageboards to mainstream social networks. We formulate this system as a one-dimensional stochastic differential equation (SDE) in contrast to Hindes *et al.* (102), who stochastically vary the transmission rate parameters; this approach is taken due to the significant measurement error associated with the approximation of the infection ratio, measured in scaled number of users, of relevant conspiracy theories by the short-time frequency, measured in tweets per hour, of each key phrase. We assume that this measurement error is roughly stationary and contains no in-

formation useful to the model (i.e., it is entirely noisy). We simultaneously fit the epidemic and noise parameters using maximum likelihood estimation from the time series data. We validate this model on the data and show analytically and numerically that the probability measure associated with a fixed initial distribution is an invariant measure under the evolution of the Fokker-Planck equation corresponding to the SDE model when the transmission rate of the Twitter network exceeds a value directly proportional to the external coupling. We further attempt to characterize the deterministic evolution of the 4chan topic-frequency parameter (the input current) as an optimization process. We use the pre-trained language model DistilBERT (103) and K-means clustering to detect and distinguish topics in the full archive of /pol/posts and regression with a labeled subset of the data to determine the toxicity. The fitness of each topic is observed to follow an emergent optimization process like temporal-difference learning with the toxicity of the content as the primary reward mechanism. This mirrors the reported behavior of individual 4chan users, termed "trolling," in which incendiary content is posted to elicit a strong reaction (104).

4.2 Data and Methods

We have obtained an archive of all 4chan posts to the board /pol/ between January 1, 2013 and January 23, 2022. From these, we filter out posts before November 20, 2019, due to the sparsity of anti-vaccine sentiment in prior years. We derive the full set of sparse orthogonal bigrams (105) collocated with the word "vaccine" at a maximum distance of three words. We order these sparse bigrams by frequency and remove stop phrases and skips to determine a set

of search key phrases, which we filter to include the most frequent conspiracy-theory-related key phrases. These phrases were as follows:

big pharma, *refuse get*, *trying force*,

people shilling, *bill gates*, *forced take*,

cause autism, *side effects*, and *deaths swine*.

These keywords, several of which refer to specific well-known conspiracy theories (106; 107; 108), are used to quantify network traffic in similar conspiracy theories on other social networks. In particular, we search for each key phrase (prepending the word "vaccine" for specificity) over the same duration using the Twitter v2 counts API and use the hourly frequency of key phrase occurrence as a proxy for the infection ratio $\beta I/N$. To model this quantity, we construct a population-based nonlinear drift-diffusion system describing cross-social-network disinformation spreading, taking the form of a normalized stochastic SIS model with 4chan as a noisy driving current.

4.2.1 Model and Validation

First, we determine the causal relationship between the occurrences of the anti-vaccine keywords in the Twitter and 4chan datasets. The corresponding multi-year hourly time series were deseasonalized with the R *msts* package using periods of 365.25 days (the mean duration of one year), 30.4375 days (1/12 the mean duration of one year), 7 days, and 24 hours. After extracting the seasonal periodicity and the trends, the remaining time series for Twitter and 4chan were evaluated for significant transfer entropy in either direction. Transfer

entropy, introduced in (109), is an information-theoretic measure that describes the degree to which a source time series (in this case, 4chan anti-vaccine content in posts per hour, which we denote Y) reduces the future uncertainty associated with a target time series (in this case, Twitter anti-vaccine content in posts per hour, which we denote X) when conditioned on the history of the target. In this case, using the symbolization of each time series by quantile function into three equally-sized partitions of the unit interval, $T_{Y \to X} = 0.0058$ is the transfer entropy value. It corresponds to the significance value $\mathbf{p \leq 10^{-3}}$ when a shuffle test is performed with respect to Y, with a characteristic delay of 3 hours sufficient to detect causality. No direct linear correlation was found between the two time series. Still, the reciprocal of Y was found to be highly inversely correlated to X, which is considered in the subsequent model.

We consider the following one-dimensional stochastic differential equation of an Itô process:

$$d\rho = \left[\beta\rho(1-\rho) - \gamma\rho - \kappa\frac{Y_{max}}{Y}\right]dt + \sigma dB 0 < \rho \leq 1. \tag{4.1}$$

Here, ρ represents the mean infection density of Twitter I/N according to the formalism of a susceptible-infected-susceptible (SIS) model; γ, β are the network-adjusted mean decay and transmission rates, and κ is a coupling coefficient describing in a nonlinear way the degree of effect that the hourly 4chan conspiracy post rate Y has on ρ (the exact nature of this nonlinear relationship was determined empirically by ranking a set of power transforms of Y by Pearson correlation with ρ). Y_{max} represents the maximum value of Y over the entire time-series and is used as a normalizing factor. B is a standard Brownian motion process, of which

dB is the differential in the usual formalation of a Wiener process. The diffusion parameter σ characterizes the shape of the estimation noise and nondeterminism of the model, which is subject to an absorbing boundary condition at $\rho = 0$ and a reflecting condition at $\rho = 1$. The drift term of the SDE (4.1) is derived from the mean-field SIS model for the infection density in a homogeneous network (110). Pseudo-maximum likelihood estimation of the parameters using the Euler approximation of the infinitesimal transition density via the R programming language's "Sim.DiffProc" package yields $\beta = 0.3119747$,$\gamma = 0.3223287$, $\kappa = 2.843189 \times 10^{-}5$, and $\sigma = 0.004616367$. We numerically verify (Fig. 4.1) that the distribution of points along sample paths of the evolution of this SDE (generated by numerical integration using the method in (111)) fits the final distribution of the data with minimal error. Note that the mean is neither 0 nor 1 but corresponds to the existence of a small subpopulation of conspiracy-believing users that represent an endemic state (actually, a set of states) of the system. This phenomenon is unique to the stochastic, driven form of the model, and we will show that this regime of the dynamics represents a stationary state of the system.

4.2.2 Analysis of metastable and stationary states

For this analysis, we restrict our attention to the case where $\rho > 0$; $\rho = 0$ is not an equilibrium of the system due to the nonzero y and diffusion term; we also fix Y. Then, the Fokker-Planck equation corresponding to (4.1) reads:

$$\partial_t p(t,\rho) = -\partial_\rho \left[\left((\gamma - \beta)\rho + (\beta - \frac{\kappa Y_{max}}{Y}) \right) p(t,\rho) \right] + \sigma \partial_\rho^2 p(t,\rho) = -\partial_\rho [\mu(\rho,t)\rho] + \sigma \partial_\rho^2 p(t,\rho). \tag{4.2}$$

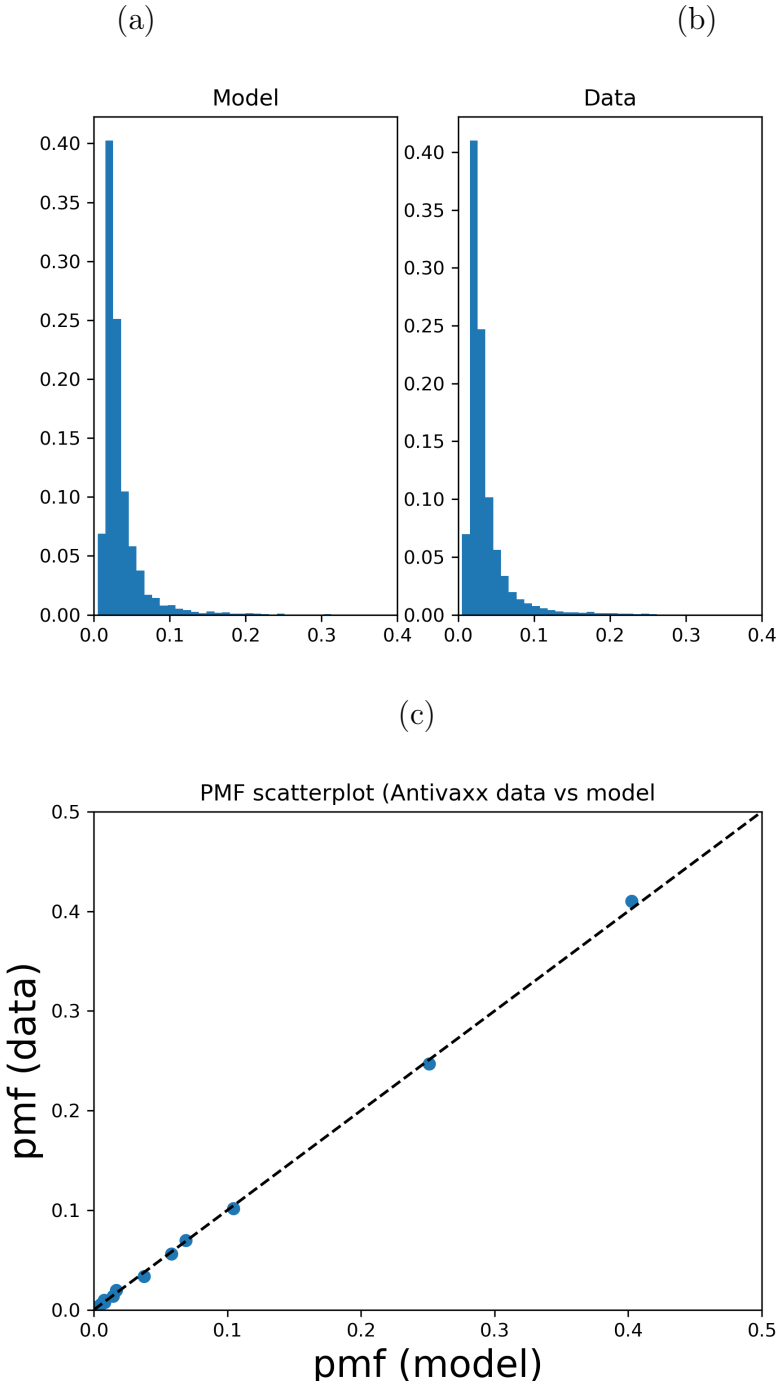

Figure 4.1 Comparison of limiting probability density histograms derived from the sample paths of (a) the model and (b) the time series data under the assumption of near-stationarity of the noise. (c). The plot of the corresponding probability masses for each binned value of ρ sampled from the time-series (vertical axis) vs. from the sample paths of the SDE (horizontal axis). Our model reproduces the distribution of ρ almost exactly. Note that the mean is neither 0 nor 1 but corresponds to the existence of a small subpopulation of conspiracy-believing users that represent an endemic state of the system.

Here, $p(t,\rho)$ is the time-evolving probability density function of ρ with support on $(0,1]$. The moment generating function, $M(s) = E[e^{\rho s}]$, is precisely the scaled inverse Laplace transform of the PDF of ρ, to which we apply the Fokker-Planck equation in (4.2):

$$\begin{aligned}
\dot{E}[e^{s\rho}] &= -\int_0^1 \partial_\rho[\mu(\rho)p]e^{s\rho}d\rho + \sigma\int_0^1 \partial_\rho^2[pe^{s\rho}]d\rho \\
&= s\int_0^1 [\mu(\rho)p(\rho)e^{s\rho}]d\rho + \sigma s^2\int_0^1 p(\rho)e^{s\rho}d\rho \\
&= s(\beta-\gamma)E[\rho e^{s\rho}] - \beta s E[\rho^2 e^{s\rho}] - (s\kappa\frac{Y_{max}}{Y} + s^2\sigma)E[e^{s\rho}].
\end{aligned}$$

So by Leibniz's rule

$$\begin{aligned}
\partial_t\partial_s E[e^{s\rho}] &= (\beta-\gamma)E[\rho e^{s\rho}] + s(\beta-\gamma)E[\rho^2 e^{s\rho}] \\
&\quad - \beta E[\rho^2 e^{s\rho}] - s\beta E[\rho^3 e^{s\rho}] - (s\kappa\frac{Y_{max}}{Y} - s^2\sigma)E[\rho e^{s\rho}] - (\kappa\frac{Y_{max}}{Y} - 2s\sigma)E[e^{s\rho}] \\
\partial_t\partial_s^2 E[e^{s\rho}] &= (\beta-\gamma)E[\rho^2 e^{s\rho}] - (\beta-\gamma)E[\rho^2 e^{s\rho}] + s(\beta-\gamma)E[\rho^3 e^{s\rho}] - \beta E[\rho^3 e^{s\rho}] \\
&\quad - s\beta E[\rho^4 e^{s\rho}] - (\kappa\frac{Y_{max}}{Y} - 2s\sigma)E[\rho e^{s\rho}] + (2\sigma)E[\rho^2 e^{s\rho}] \\
&\quad - (\kappa\frac{Y_{max}}{Y} - 2s\sigma)E[\rho e^{s\rho}] + 2\sigma)E[e^{s\rho}] \\
\partial_t\partial_s^3 E[e^{s\rho}] &= 3(\beta-\gamma)E[\rho^3 e^{s\rho}] - 2\beta E[\rho^4 e^{s\rho}] \\
&\quad - 2\kappa\frac{Y_{max}}{Y}E[\rho^2 e^{s\rho}] + 4\sigma E[\rho^2 e^{s\rho}] - 4\sigma E[\rho e^{s\rho}] \\
\partial_t\partial_s^n E[e^{s\rho}] &= 3(\beta-\gamma)E[\rho^n e^{s\rho}] - 2\beta E[\rho^{n+1} e^{s\rho}] \\
&\quad - 2\kappa\frac{Y_{max}}{Y}E[\rho^{n-1} e^{s\rho}] + 4\sigma E[\rho^{n-1} e^{s\rho}] - 4\sigma E[\rho^{n-2} e^{s\rho}], n > 3,
\end{aligned}$$

and setting $s = 0$, we obtain that

$$\begin{aligned}
\dot{E}[\rho] &= (\beta-\gamma)E[\rho] - \beta E[\rho^2] - (\kappa\frac{Y_{max}}{Y} + 2\sigma) \\
\dot{E}[\rho^2] &= 2(\beta-\gamma)E[\rho^2] - \beta E[\rho^3] - 2(\kappa\frac{Y_{max}}{Y})E[\rho] + 2\sigma(E[\rho^2]+1) \\
\dot{E}[\rho^3] &= 3(\beta-\gamma)E[\rho^3] - 2\beta E[\rho^4] - 2\kappa\frac{Y_{max}}{Y}E[\rho^2] + 4\sigma E[\rho^2] - 4\sigma E[\rho] \\
\dot{E}[\rho^n] &= 3(\beta-\gamma)E[\rho^n] - 2\beta E[\rho^{n+1}] - 2\kappa\frac{Y_{max}}{Y}E[\rho^{n-1}] + 4\sigma E[\rho^{n-1}] \\
&\quad - 4\sigma E[\rho^{n-2}], n > 3.
\end{aligned}$$

The existence of an equilibrium is determined by the norm

$$\Delta = \sum_{i=1}^{\infty} \dot{E}[\rho^n]^2 = \dot{E}[\rho]^2 + \dot{E}[\rho^2]^2 + \sum_{i=1}^{\infty}\left[3(\beta-\gamma)E[\rho^{3+n}] - 2\beta E[\rho^{n+4}] - 2(\kappa\frac{Y_{max}}{Y} - 2\sigma)E[\rho^{n+2}]\right]^2. \tag{4.3}$$

Expanding and using Jensen's inequality, the infinite series term in (4.3) is bounded below by zero and above by

$$\begin{aligned}
&(9\beta^2 - 18\gamma\beta + 9\gamma^2)E[\sum_{n=3}^{\infty}\rho^{2n}] \\
-(6\beta^2 - 6\gamma\beta)E[\rho\sum_{n=3}^{\infty}\rho^{2n}] &+ (24\beta\sigma - 24\gamma\sigma - 12(\beta-\gamma)\kappa\frac{Y_{max}^2}{Y^2})E[\frac{1}{\rho}\sum_{n=4}^{\infty}\rho^{2n}] \\
&+\beta^2 E[\sum_{n=4}^{\infty}\rho^{2n}] - (8\beta\gamma + 4\beta\kappa\frac{Y_{max}^2}{Y^2})E[\sum_{n=2}^{\infty}\rho^{2n}] \\
&+[16\sigma(\sigma - \kappa\frac{Y_{max}^2}{Y^2}) + 4\kappa^2\frac{Y_{max}^4}{Y^4}]E[\frac{1}{\rho}\sum_{n=1}^{\infty}\rho^{2n}]
\end{aligned}$$

which can be further simplified using the well-known formula for the sum of an infinite

geometric series in ρ^2.

We next choose a Lyapunov candidate function for the first two moments of the system: let $\delta = \frac{1}{2-m_1}(E[\rho] - m_1)^2 + \frac{1}{2-m_2}(E[\rho^2] - m_2)^2$ for initial first and second moments m_1, m_2. Then $\delta > 0$ and

$$\dot{\delta} = \dot{E}[\rho]E[\rho] + \dot{E}[\rho^2]E[\rho^2].$$

Here, we can again use Jensen's inequality for the moments of order higher than two. The mean and variance of ρ are locally Lyapunov-stable where $\dot{\delta} < 0$ holds in a neighborhood of an equilibrium. The terms of the Lyapunov function can be written for higher-order moments ($n \geq 3$) and generally take the form

$$\dot{\delta}_3 = \sum_{n=3}^{\infty} E[\rho^n]\dot{E}[\rho^n].$$

In a neighborhood of the equilibrium, $\dot{E}[\rho^n]$ can be kept smaller than some arbitrary constant ε due to continuity, so that

$$\dot{\delta}_3 \leq \varepsilon E[\frac{\rho^2}{1-\rho}].$$

Hence, the destabilizing effect of the higher-order moments can be accounted for locally by increasing the threshold slightly below 0.

The boundary of the intersection of the existence and stability conditions as a function of $E[\rho]$, $E[\rho^2]$, and β is displayed as an isosurface in Fig. 4.2a; and, with respect to $E[\rho]$, $E[\rho^2]$, and κ, in Fig. 4.2b. In both instances, we note the coexistence of a metastable epidemic and a stable endemic state for the same value of the parameters, which is mathematically

counter-intuitive and unique to the stochastic, driven version of the model. The upper, metastable branch of the surface represents the spread of the conspiracy theory through the entire susceptible population; even though it is not a permanent regime of the system, its effect can be politically significant, as we will discuss. For some values of the moments there are also multiple metastable equilibria as a function of β; however, this effect is not present when we consider the dependence on κ, though the dependence is qualitatively similar. This surface represents a subset of the equilibria, which are locally asymptotically stable in some small neighborhood of the isosurface; larger amplitude noise may destabilize the top branches. The resulting diagram depicts multiple states, such as an epidemic state, which is metastable (a sufficiently large perturbation to the distribution can result in immediate loss of stability), and an endemic state of smaller magnitude; the two states collide when the variance is sufficiently high. We emphasize that this diagram potentially does not represent an exhaustive enumeration of the stable regimes of the system but only a sufficient condition for (meta)stability and existence. Note that we have also restricted ourselves to the case where the parameters are all constant, and the input from /pol/ itself is either constant or stationary. As we will discuss in the next section, this situation does not represent reality even on shorter timescales; instead, 4chan content is continually optimizing itself to become more toxic, which has implications for both the design and prevention of large-scale misinformation campaigns.

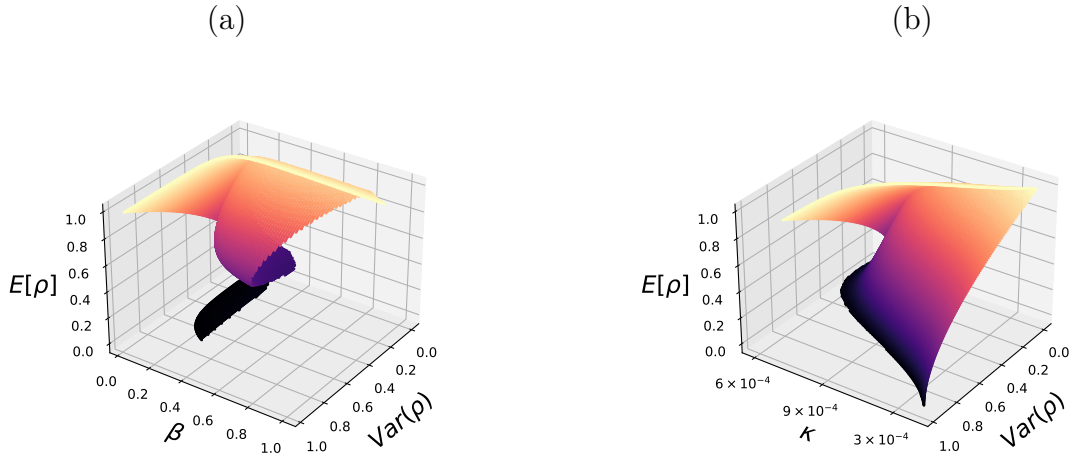

Figure 4.2 Region for which the moment-stability conditions for the existence and local stability of stationary states are satisfied as a function of the mean and variance of ρ (vertical axis and rightmost horizontal axis) and (a) β (the spreading rate) and (b) κ (the coupling strength); all parameters other than those on the axes were those estimated from the model. The color of each surface represents the value on the z-axis and is used to orient the figure visually in three dimensions.

4.3 Optimality of the effect of "trolling"

4chan users have been observed to engage in hyperbole and controversial posting to elicit a reaction (104). Using recently developed techniques in natural language embedding and classification (112; 113), we discover that this behavior significantly impacts the frequency of topics due to the spontaneous and collective emergence of a temporal difference learning process (114). We demonstrate the result as follows: beginning with the full archive of text posts to /pol/ from January 1, 2013 to January 23, 2022, we tokenize each post using the NLTK python package's Casual tokenizer (115). Subsequently, we fine-tune the transformer-type natural language model DistilBERT (103), training it to predict a partially-masked copy of a random sample of posts (359,609 posts were chosen uniformly for the training set from the nearly one billion posts in the overall dataset). This class of natural language

model, which works by learning to encode distributions of words in a recurrent model using a neural network, has achieved remarkable accuracy in natural-language classification tasks (116; 117) from a few training examples; for precise implementation details, we refer the reader to (103; 112; 118). The fine-tuned transformer model was then combined with a two-layer neural network used as a regression head; the resulting model was trained on a balanced random training split of the Jigsaw Unintended Bias in Toxicity Classification dataset to detect toxicity and then evaluated on the testing split, and the automatically generated labels in the dataset reported in (119). Individually, each of these resulted in an F1-score consistently greater than 0.8 (and a thresholded classification accuracy of at least 80%), which we deemed sufficient for our purposes since we are primarily concerned with trends in toxicity aggregated by topic. To aggregate posts by topic, the 784-dimensional embeddings given by the fine-tuned DistilBERT without the classification or prediction heads were clustered using K-Means with 1,000 centroids, and each cluster was taken to be a topic. A sample set of topic vectors colored by cluster and visualized using the UMAP algorithm in two dimensions is plotted in Fig. 4.3; note that, to a high degree of accuracy, each topic cluster occupies a distinct and connected region of space; this property is desirable for the remainder of our analysis.

The topic clustering and the classification were applied to the chronological series of 4chan posts over a sliding window of 300,000 posts, and the toxicity of each post in the window was estimated using the DistilBERT classifier. The separability of topics projected using the UMAP embedding algorithm (113) is depicted in Fig. 4.3. Subsequently, a value

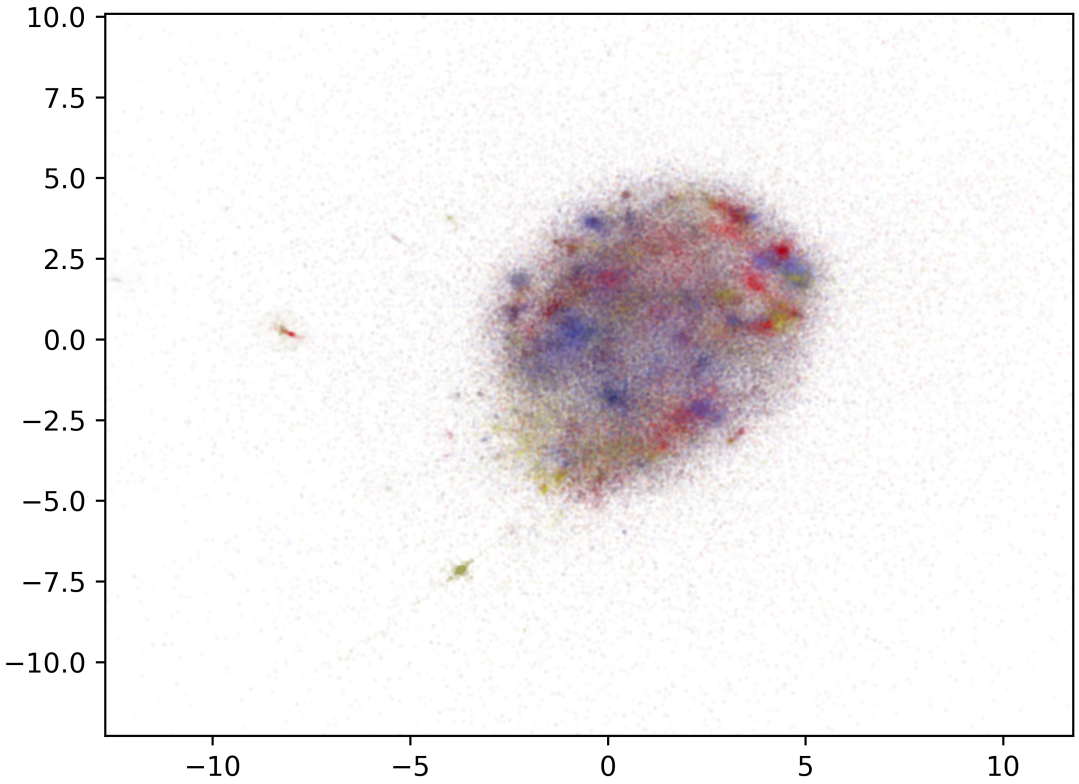

Figure 4.3 DistilBERT embeddings of a sample of 300,000 posts, projected onto two dimensions using the UMAP algorithm (113) and colored by topic (that is, by K-means cluster obtained from the unprojected embeddings). The visualization shows that separate topics are well separated by the projected embedding, even though much of the semantic information is lost.

function tabulated over the set of topics was trained on the current window using the TD(0) algorithm introduced by Sutton (114) with the estimated toxicity of the next post as the reward. TD(0) is a temporal difference reinforcement learning algorithm hypothesized to be similar to the effect of dopaminergic reinforcement in the animal brain. Given a Markov decision process over discrete time t with state sequence s_t and instantaneous reward per

state $r(s)$, the algorithm iteratively estimates the expected state-value function (hereafter referred to as the TD-value)

$$V(s) = E\left[\sum_{\tau=t}^{\infty} r(s_\tau)\gamma^\tau\right] \tag{4.4}$$

representing the expected future reward resulting from the transition to state s discounted over time exponentially by a factor of γ. The estimate to (4.4) is obtained iteratively from a time series of visited states and observed rewards using the Bellman optimality principle as

$$V_{i+1}(s_t) = V_i(s) + \lambda\left((r(s_t, t) + \gamma V(s_{t+1})) - V(s_t)\right), \tag{4.5}$$

where i is the current iteration of the update algorithm and λ is a learning rate, usually less than one. Note that the effect of the resulting TD-value function is invariant under a uniform scaling or other monotone transformation since it is only used to rank topics. Due to the mechanism of temporal difference learning, this value function informs the choice of topic, which would maximize the time-discounted expected future toxicity of every subsequent post. We determine that, for every window analyzed, the probability distribution over the mentioned topics in the dataset faithfully reproduces the theoretical probability distribution that would maximize the obtained value function (see Fig. 4.4 and the caption for a detailed explanation of the evaluation used). This theoretical distribution, corresponding to the black dashed curve in Fig. 4.4a-b, was generated as follows: first, 40 uniformly distributed random values were sampled from the real numbers between 0 and 21, and second, the maximum of these 40 numbers was chosen to be the result of the trial. This process roughly corresponds to the selection of the one with the highest toxicity TD-value from a list of relevant topics.

The resulting curve was a close match to the scatter-plot representing each topic's empirical normalized frequency of occurrence by TD value (except at the extreme tail, where they differed by small orders of magnitude). A closed-form curve of the form

$$\hat{P}(V) = \left[e^{ka_1V+kb_1} + e^{ka_2V+kb_2}\right]^{\frac{1}{k}}$$

was fit using maximum-likelihood estimation to the points of the scatter-plot and retains the shape of the tail. This striking result indicates that posts to the anonymous image-board /pol/can self-organize to increase the degree of toxicity of the resulting discourse. The method was tested over a large number of intervals spanning almost eight years, and all other results were consistent with what is reported in this figure except for slight scaling of the TD-Value function and re-embedding of the topics. Moreover, as observed in Fig. 4.4d, the relationship between topic occurrence frequency and the learned value function is destroyed when the value function is learned from a random time-reordering of the posts. Shuffling the order of the posts 100 times, reproducing the entire optimization method for the shuffled time series of DistilBERT embeddings, and selecting the trial which produces the highest expected TD-value per post destroys the significant relationship between topic frequency and TD-value; the observed self-optimization is a dynamical process and cannot be observed if the posts do not exist in the correct chronological order.

4.4 Discussions

The existence of misinformation as both an organic and externally-organized phenomenon poses a grave risk to the political sphere, and our model, though by no means complete, has

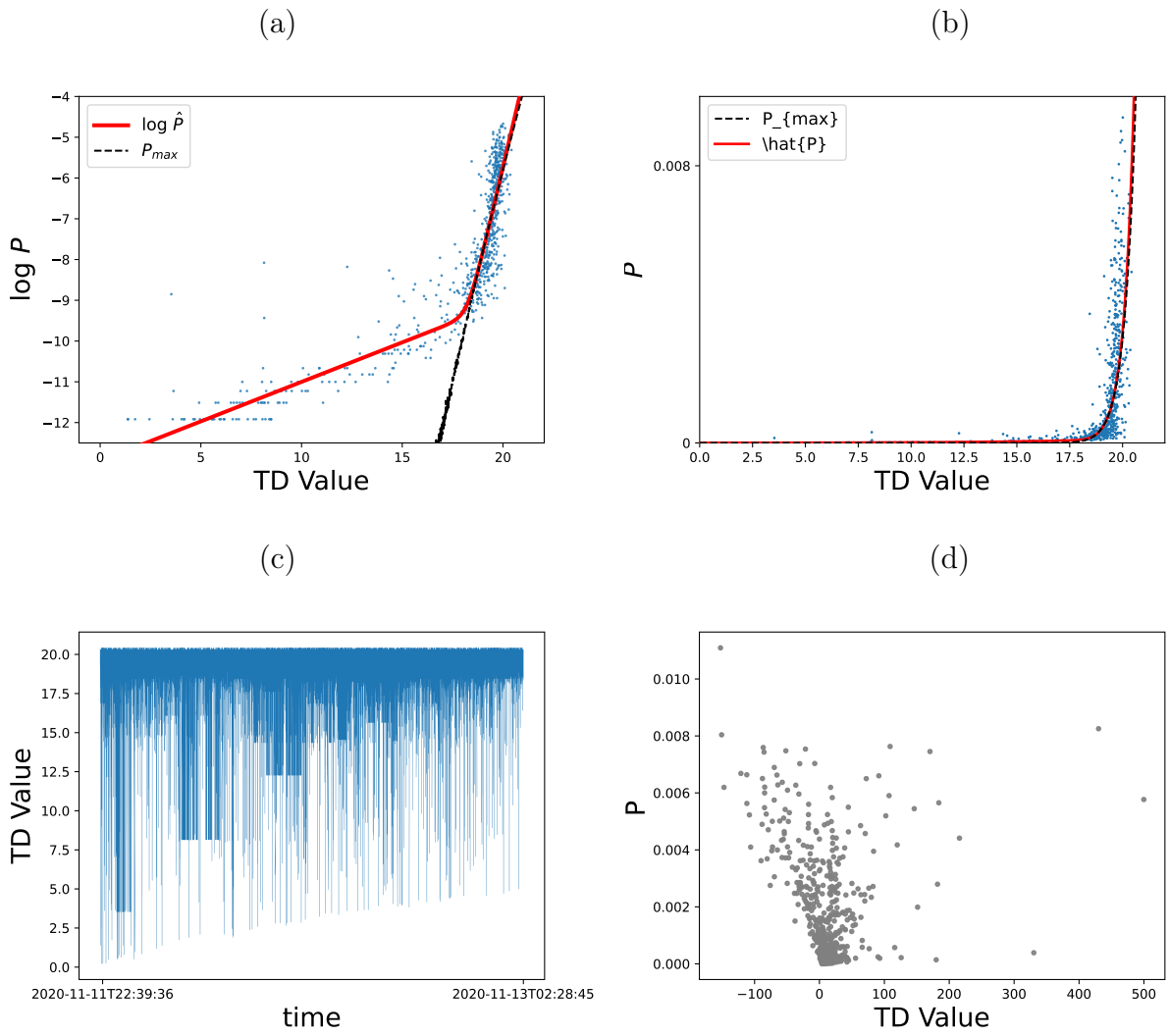

Figure 4.4 (a). Log-probability associated with each detected topic in the 4chan dataset for 300,000 posts plotted as learned TD-value (expected discounted future toxicity of the discourse). The black dashed curve indicates the logarithm of a histogram-estimated PDF $P_{max}(V)$ determined from a random sample of TD values V. The analytic (red) curve indicates the value of $\hat{P}$. (b). Probability density curves P_{max} and $\hat{P}$ and sample topic frequencies (blue scatter plot points) corresponding to the values in the log-scale plot in (a). (c). Temporal difference value time-series plot corresponding to the interval of posts used in (a) and (b). Note that the expected TD value increases over time, as is typical of a reinforcement learning algorithm. (d). Equivalent scatterplot to (b) in the event that the order of the time series is permuted 100 times and the trial maximizing the expected TD-value is taken. The correlation is destroyed by the time-reordering.

several implications for its prevention. First, there is the matter of preventing the passive spread of misinformation once it has reached the endemic stage. Note that the stability of the endemic and epidemic states and their magnitudes all depend on the variance of the number of infected individuals and the spreading parameter β. This suggests the feasibility of varying the connections of the network and, therefore β in a stochastic manner to limit the instantaneous reach of users may work to mitigate the spread of misinformation, particularly during a "storm" when the influx of users or content from the external source (/pol/, for instance) is strong.

We now focus on how such a misinformation storm may be optimized for maximum effect. In addition to the classification of text, large language models may be used to generate large quantities of text with remarkable success predictively (120; 121; 122); for ethical reasons, we did not use such a model, as it would need to be trained on 4chan data to work, and would thus be capable of generating toxic content and hate speech. However, the controversial GPT-4chan generative language model was trained using an open-source natural language model with only 6 billion parameters on the data of (119) and then deployed on 4chan to imitate an actual user. The 4chan site allows users to bypass CAPTCHA verification for a small fee, and it was observed that users on 4chan could not immediately distinguish GPT-4chan, whose generative capability has now long since been superseded by newer models, from a human user (123; 124). Notably, many newer generative language models employ reinforcement learning to target their generation to a reward function (125); primarily, this method is used to increase the realism of the output, but in principle, any target could be used. It

is, therefore, conceivable that a hostile actor or organization could launch a disinformation campaign using 4chan as a "petri dish" to increase the spread of conspiracy theory content in a purely automated fashion, exploiting the emergent optimization mechanism on 4chan that results from a frequent and strong reaction to a particular category of post, and then time the release of the resulting content into the mainstream where it could influence elections or destabilize the geopolitical landscape. To provide insight into how to mitigate this type of threat, our model requires augmentation in multiple ways. For example, first, we need to consider the effect of the intentional recruitment of users by promoters of a conspiracy theory, particularly those with a high degree centrality in the target network. Second, our model must be updated to quantitatively reflect the optimization process that could drive both the spread rate of the content and its influx from /pol/. Third, to estimate the parameters of the augmented model, we must experimentally or numerically determine the maximum efficiency and speed of such an optimization process, which we expect to depend heavily on the maximum post rate that allows a bot to evade detection, but which may also depend on the sparsity of the reward due to the interactions and therefore on the volume of engagement on /pol/.

4.5 Conclusions

We have introduced a data-driven epidemiological model of the spread of conspiracy theories from /pol/ to Twitter. The model fits the distribution of time series data of Twitter conspiracy theory content well and can be analytically shown to exhibit multiple stationary

states for each value of the parameters, especially those related to the spreading rate of the content and its prevalence on 4chan, the latter of which we find is related in a counter-intuitive and striking way to the evolution of the toxic discourse on 4chan. This result and the corresponding method have novel implications for protecting against political misinformation campaigns and radicalization. As future work, we propose to consider the effect of intentional recruitment and organized misinformation campaigns, to extend our results on toxicity to a mathematical model of the evolution of the rate parameters, to incorporate the degree distribution of the network, to experimentally validate the scenario described in the previous section in a controlled and ethical setting, and to extend our analysis to multiple types of conspiracy theories in various social networks and compare their effectiveness.

CHAPTER A

Appendix for Chapters 2 and 3

A Python source code for Chapter 2 simulations and damping coefficient calculations

```
import numpy as np, sys, os

from autograd import numpy as jnp, make_jvp

import random

from scipy.integrate import odeint as ode, solve_ivp as ode2



#commandline arguments

#usage is python negative_damping.py MODEL INTERVAL_TO_ADD_PEDESTRIANS [
    ↪ WHETHER_TO_GENERATE_FOR_SUPPLEMENT]

#this code generates the data for plotting the multi-panel figures (4,5,6)

#the scatter plots (fig 3) can be generated by running this code for n=1 and
    ↪  a sinusoidally forced bridge (see comments in the integrand function
    ↪ )

#and then sweeping over bridge and pedestrian frequencies

#with long interval while binary searching over the final amplitude to find
    ↪ the max_n_walkers value for which max_amplitude exceeds a particular
    ↪ value.
```

```
#sample traces are included for a single pedestrian and sinusoidally forced
   ↪ bridge (with slightly a different output format), which were used to
   ↪ generate the
which_model=int(sys.argv[1]) #1,2,3
walker_addition_interval=int(sys.argv[2]) #try 20 (seconds)
if len(sys.argv)>3:
       use_supplemental_fig_params=int(sys.argv[3])!=0
else:
       use_supplemental_fig_params=False



#default params
max_n_walkers=600
walker_masses=np.random.randn(max_n_walkers)*10.0+76.9
normalized_leg_lengths=np.random.randn(max_n_walkers)*0.092+1.17
stability_margin_hof=np.random.randn(max_n_walkers)*0.002+0.0157
stability_margin_hof*=1.0-2.0*(np.random.rand(max_n_walkers)>0.5) #pick a
   ↪ random foot to be down
if which_model!=3:
       step_frequency_hz=np.abs((1.04 if use_supplemental_fig_params else
          ↪ 0.9)+np.random.randn(max_n_walkers)*0.0)
```

```
else:
        step_frequency_hz=(1.04 if use_supplemental_fig_params else 0.9)*np.
            ↪ ones(max_n_walkers)
COP_lateral_offsets=stability_margin_hof*(1.0-np.tanh(0.25*np.sqrt(9.8/
    ↪ normalized_leg_lengths)/step_frequency_hz))
inverted_pendulum_frequency_unadjusted_radians=np.sqrt(9.8/1.17)*(1.0 if not
    ↪  use_supplemental_fig_params else 1.04/0.9)+0.0*np.random.randn(
    ↪ max_n_walkers)
stability_margin_hof*=-1
t_next_step=(np.random.rand(max_n_walkers)*0.5/step_frequency_hz)
bridge_frequency_hz=1.03
bridge_frequency_radians=bridge_frequency_hz*2*np.pi
bridge_mass=113e3
COP_offset_fixed=.63
limit_cycle_amplitude_parameter=0.47
bridge_damping_ratio=0.04
bridge_damping_coefficient=bridge_frequency_radians*bridge_damping_ratio
bridge_frequency_squared=bridge_frequency_radians*bridge_frequency_radians
limit_cycle_damping_parameter=23.25
break_on_crit=False
```

```
t=0
def jacobian(F,N,x):
        identity=np.eye(N)
        Jcols=[]
        jvp=make_jvp(F)(x)
        for i in range(N):
          Jcols.append(jvp(identity[i])[1])
        return np.concatenate(Jcols).reshape(identity.shape)
def integrand_model1_2(state,n_walkers,COP_lateral_offsets,
    ↪ sagittal_velocity_perturbed,normalized_leg_lengths,walker_masses):
    y=state[2:n_walkers+2]
    ydot=state[n_walkers+2:2*n_walkers+2]
    foot_force=9.8*(COP_lateral_offsets[:n_walkers]-y[:n_walkers])/
        ↪ normalized_leg_lengths[:n_walkers]
    Fdot=-9.8*ydot/normalized_leg_lengths[:n_walkers]
    xdotdot=(jnp.sum(walker_masses[:n_walkers]/bridge_mass*foot_force)-
        ↪ bridge_damping_coefficient*state[1]-state[0]*
        ↪ bridge_frequency_squared)
```

```
    xdotdotdot=(jnp.sum(walker_masses[:n_walkers]/bridge_mass*Fdot)-
       ↪ bridge_damping_coefficient*xdotdot-state[1]*
       ↪ bridge_frequency_squared)
    yddot=-foot_force-xdotdot
    eta=state[3*n_walkers+2:4*n_walkers+2]
    sagittal=state[9*n_walkers+2:10*n_walkers+2]
    deriv=jnp.concatenate([jnp.array([state[1],xdotdot]),ydot,yddot, jnp.
       ↪ zeros(n_walkers),9.81/normalized_leg_lengths[:n_walkers]*eta-
       ↪ xdotdotdot*xdotdot,y*state[1], y*state[0],eta,
       ↪ sagittal_velocity_perturbed,sagittal*state[1],sagittal*state[0],
       ↪ state[:1]**2])
    return deriv
def sgn(x):#not really limit_cycle_amplitude_parameter signum function
   ↪ because it has sgn(0)!=0
    return 1-2*(x<0)


def integrand_model3(state,N,omegas,Ms):
        n_walkers=N
        w0=omegas[:n_walkers]
        walker_masses=Ms[:n_walkers]
        nu=w0
```

```
y=state[2:n_walkers+2]
ydot=state[n_walkers+2:2*n_walkers+2]
p=COP_offset_fixed
r=walker_masses/(bridge_mass+jnp.sum(walker_masses))
foot_force=limit_cycle_damping_parameter*(ydot**2+nu**2*(
    ↪ limit_cycle_amplitude_parameter**2-(y-p*sgn(y))**2))*ydot-w0
    ↪ **2*(y-p*sgn(y))
xdotdot=1/(1-jnp.sum(r))*(jnp.sum(foot_force*r)-
    ↪ bridge_damping_coefficient*state[1]-state[0]*
    ↪ bridge_frequency_squared)
yddot=(-foot_force-xdotdot)
dHdydot=limit_cycle_damping_parameter*(3*ydot**2+nu*nu*(
    ↪ limit_cycle_amplitude_parameter**2-(y-sgn(y)*p)**2))
dHdy=limit_cycle_damping_parameter*ydot*(-nu**2*(2*y-2*p*sgn(y)))-w0
    ↪ **2
Fdot=dHdy*ydot+dHdydot*yddot
mu=state[2*n_walkers+2:3*n_walkers+2]
eta=state[3*n_walkers+2:4*n_walkers+2]
hu=dHdy*eta+dHdydot*mu
xdotdotdot=1/(1-jnp.sum(r))*(jnp.sum(Fdot*r)-
    ↪ bridge_damping_coefficient*xdotdot-state[1]*
```

```
            ↪ bridge_frequency_squared)
        deriv=jnp.concatenate([jnp.array([state[1],xdotdot]),ydot,yddot, -
            ↪ xdotdot*xdotdotdotdot-hu,mu,y*state[1],y*state[0],hu,jnp.zeros(
            ↪ n_walkers),dHdy,dHdydot,state[:1]**2])
# dx=jnp.concatenate([jnp.array([state[1],xdotdot]),ydot,yddot, jnp.zeros(8*
    ↪ n_walkers),state[:1]**2])
        return deriv


bridge_displacement=np.zeros(2)
inverted_pendulum_frequency_radians=np.sqrt(9.81/normalized_leg_lengths)
if which_model!=3:
        COM_displacement=np.zeros(1)
        COM_velocity=COP_lateral_offsets[:1]*
            ↪ inverted_pendulum_frequency_radians[:1]*np.tanh(
            ↪ inverted_pendulum_frequency_radians[:1]*0.5/step_frequency_hz
            ↪ [:1])
else:
        COM_velocity=np.zeros(1)
        COM_displacement=(COP_offset_fixed-limit_cycle_amplitude_parameter)*
            ↪ np.ones(1)
```

```
#forward_speed=(step_frequency_hz*normalized_leg_lengths/1.34/1.352)**2*
    ↪ step_frequency_hz
forward_speed=0.36*np.ones(max_n_walkers)
t=0
t_prev_step=t_next_step-0.5/step_frequency_hz
tprevL=(stability_margin_hof<0)*t_prev_step+(stability_margin_hof>=0)*(
    ↪ t_prev_step-0.5/step_frequency_hz)
COM_displacement_prev=np.zeros(max_n_walkers)
t_prev_COM_period=np.zeros(max_n_walkers)
t_prev_prev_COM_period=np.zeros(max_n_walkers)
t_prev_prev_step=np.zeros(max_n_walkers)
t_prev_prev_prev_step=np.zeros(max_n_walkers)
for n_walkers in range(2,max_n_walkers):
        t_prev_prev_step[n_walkers-1]=t
        n_periods_COM=0
        t_prev_prev_prev_step[n_walkers-1]=t-t/0.9
        step_last=np.zeros(n_walkers)
        step_first=np.ones(n_walkers)*1.0e9
        step_interval=0.5/step_frequency_hz[:n_walkers]
        time_average_d_foot_force_d_bridge_velocity=np.zeros(n_walkers)
        COM_sin_component=np.zeros(n_walkers)
```

```
COM_cos_component=np.zeros(n_walkers)

sagittal_sin_component=np.zeros(n_walkers)

sagittal_cos_component=np.zeros(n_walkers)

mu=np.zeros(n_walkers)

eta=np.zeros(n_walkers)

d_foot_force_d_COM_velocity=np.zeros(n_walkers)

d_foot_force_d_COM=np.zeros(n_walkers)

bridge_mean_amplitude=0

inverted_pendulum_freq_squared=(9.81/normalized_leg_lengths[
    ↪ n_walkers-1])

t_next_step[n_walkers-1]+=t #account for the fact that the current
    ↪ most recently added walker has his next footfall way in the
    ↪ past, since he was initialized at the beginning of the
    ↪ simulation.

t_prev_step[n_walkers-1]+=t

if which_model!=3:

        COM_velocity=np.concatenate([COM_velocity,
            ↪ COP_lateral_offsets[n_walkers-1:n_walkers]*
            ↪ inverted_pendulum_freq_squared*np.tanh(
            ↪ inverted_pendulum_freq_squared*0.5/step_frequency_hz[
            ↪ n_walkers-1])])
```

```
        COM_displacement=np.concatenate([COM_displacement,np.zeros
            ↪ (1)])
else:
        random_side=random.choice([-1,1])
        COM_velocity=np.concatenate([COM_velocity,np.zeros(1)])
        COM_displacement=np.concatenate([COM_displacement,
            ↪ random_side*(COP_offset_fixed-
            ↪ limit_cycle_amplitude_parameter)*np.ones(1)])
d_foot_force_d_bridge_velocity=np.zeros(n_walkers)
jump_term_1=np.zeros(n_walkers)
jump_term_2=np.zeros(n_walkers)
jump_term_3=np.zeros(n_walkers)
jump_term_4=np.zeros(n_walkers)
sagittal_velocity_perturbed=np.zeros(n_walkers)
num_steps=np.zeros(n_walkers)
t_prev_addition=t
has_zero_cross=False
n_footfalls=0
tfirstp=t_prev_addition
sagittal=np.zeros(n_walkers)
saggital_prev=np.zeros(n_walkers)
```

```
t_first_bridge_zero=walker_addition_interval+t_prev_addition

t_final_bridge_zero=t_prev_addition

bridge_max_amplitude=0

num_zero_cross=0

order_param_cos_component_step=0

order_param_sin_component_step=0

order_param_cos_component_COM=0

order_param_sin_component_COM=0

while t<t_prev_addition+walker_addition_interval:

  if which_model!=3:

      integrand_autonomous=lambda x: integrand_model1_2(x,n_walkers
         ↪ ,COP_lateral_offsets[:n_walkers],
         ↪ sagittal_velocity_perturbed[:n_walkers],
         ↪ normalized_leg_lengths[:n_walkers],walker_masses[:
         ↪ n_walkers])

      t_next_footfall_all_walkers=max(t,np.min(t_next_step[:
         ↪ n_walkers]))

  else:

      integrand_autonomous=lambda x: integrand_model3(x,n_walkers,
         ↪ inverted_pendulum_frequency_unadjusted_radians[:
         ↪ n_walkers], walker_masses[:n_walkers])
```

```
    t_next_footfall_all_walkers=min(t+0.002,t_prev_addition+
        ↪ walker_addition_interval)
#Fn_=jit(Fn__)
integrand=lambda t,x: integrand_autonomous(x)
ts=np.linspace(0,t_next_footfall_all_walkers-t,15 if which_model
    ↪ ==3 else 10)
if which_model==3:
    state=np.concatenate([bridge_displacement[:2],
        ↪ COM_displacement[:n_walkers],COM_velocity[:n_walkers],
        ↪ mu,eta, COM_sin_component,COM_cos_component,
        ↪ d_foot_force_d_bridge_velocity,np.zeros(n_walkers),
        ↪ d_foot_force_d_COM,d_foot_force_d_COM_velocity,
        ↪ bridge_mean_amplitude*np.ones(1)])
else:
    state=np.concatenate([bridge_displacement[:2],
        ↪ COM_displacement[:n_walkers],COM_velocity[:n_walkers],
        ↪ mu,eta, COM_sin_component,COM_cos_component,
        ↪ d_foot_force_d_bridge_velocity,sagittal,
        ↪ sagittal_sin_component,sagittal_cos_component,
        ↪ bridge_mean_amplitude*np.ones(1)])
state_prev_sample=state.copy()
```

```
J_=lambda t,x: jacobian(integrand_autonomous,len(state),x)
trajectory_current_sample=ode2(integrand, [0, ts[-1]],state,
    ↪ t_eval=ts, method='LSODA', jac=J_, rtol=1e-6, atol=1e-8).y.
    ↪ transpose()
state=trajectory_current_sample[-1]
for i,trajectory_current_subsample in enumerate(
    ↪ trajectory_current_sample[:]):
    if i<2:
            continue
    tsi=ts[i]
# if which_model!=3:
# sys.stderr.write('%i %f %f %f '%(n_walkers,ts[i]+t,
    ↪ trajectory_current_subsample[0],
    ↪ trajectory_current_subsample[1])+('%f '*n_walkers)%tuple((
    ↪ COP_lateral_offsets[:n_walkers]-
    ↪ trajectory_current_subsample[2:n_walkers+2]).tolist())+'\
    ↪ n_walkers')
# else:
# sys.stderr.write('%i %f %f %f '%(n_walkers,ts[i]+t,
    ↪ trajectory_current_subsample[0],
    ↪ trajectory_current_subsample[1])+('%f '*n_walkers)%tuple((
```

```
↪ trajectory_current_subsample[2:n_walkers+2]).tolist())+'\
↪ n_walkers')
 if trajectory_current_sample[i-2,0]<trajectory_current_sample
    ↪ [i-1,0] and trajectory_current_sample[i-1,0]>
    ↪ trajectory_current_sample[i,0]:
     if has_zero_cross:
            COM_sin_component_full_period=
               ↪ trajectory_current_sample[i-1,4*n_walkers
               ↪ +2:5*n_walkers+2]
            COM_cos_component_full_period=
               ↪ trajectory_current_sample[i-1,5*n_walkers
               ↪ +2:6*n_walkers+2]
            if which_model==3:
                d_foot_force_d_COM_full_period=np.array(
                   ↪ trajectory_current_sample[i-1,-2*
                   ↪ n_walkers-1:-n_walkers-1])
                d_foot_force_d_COM_velocity_full_period=np.
                   ↪ array(trajectory_current_sample[i-1,-
                   ↪ n_walkers-1:-1])
            else:
```

```
                    sagittal_sin_component_full_period=
                        ↪ trajectory_current_sample[i-1,-2*
                        ↪ n_walkers-1:-n_walkers-1]
                    sagittal_cos_component_full_period=
                        ↪ trajectory_current_sample[i-1,-n_walkers
                        ↪ -1:-1]
                mean_amplitude=trajectory_current_subsample[-1]
        num_zero_cross+=1
        has_zero_cross=True
        t_final_bridge_zero=max(tsi+t,t_final_bridge_zero)
        t_first_bridge_zero=min(tsi+t,t_first_bridge_zero)
if which_model!=3:
        sys.stderr.write('%i %f %f %f '%(n_walkers,t,state[0],
            ↪ state[1])+('%f '*n_walkers + '0.0 '*(max_n_walkers-
            ↪ n_walkers))%tuple((COP_lateral_offsets[:n_walkers]-
            ↪ state[2:n_walkers+2]).tolist())+'\n')
else:
        sys.stderr.write('%i %f %f %f '%(n_walkers,t,state[0],
            ↪ state[1])+('%f '*n_walkers+ ' 0.0 '*(max_n_walkers-
            ↪ n_walkers))%tuple(state[2:n_walkers+2].tolist())+'\
            ↪ n')
```

```
sys.stderr.flush()
sagittal=state[7*n_walkers+2:8*n_walkers+2]
bridge_displacement=state[:2]


if has_zero_cross:
  COM_sin_component=state[4*n_walkers+2:5*n_walkers+2]
  COM_cos_component=state[5*n_walkers+2:6*n_walkers+2]
  d_foot_force_d_bridge_velocity=state[6*n_walkers+2:7*n_walkers
     ↪ +2]
  if which_model!=3:
         sagittal_sin_component=state[-2*n_walkers-1:-n_walkers
            ↪ -1]
         sagittal_cos_component=state[-n_walkers-1:-1]
  else:
        d_foot_force_d_COM=state[-2*n_walkers-1:-n_walkers-1]
        d_foot_force_d_COM_velocity=state[-n_walkers-1:-1]
  mu=state[2*n_walkers+2:3*n_walkers+2]
  eta=state[3*n_walkers+2:4*n_walkers+2]
  bridge_mean_amplitude=state[-1]
k=integrand_autonomous(state)
t=t_next_footfall_all_walkers
```

```
bridge_displacement=state[:2]
COM_displacement=state[2:n_walkers+2]
COM_velocity=state[n_walkers+2:2*n_walkers+2]
if which_model!=3:
        indices_foot_down=np.where(t_next_step[:n_walkers]<=t+1.0e
            ↪ -10)[0]
else:
        indices_foot_down=np.where(sgn(state_prev_sample[2:
            ↪ n_walkers+2])!=sgn(state[2:n_walkers+2]))[0]
time_average_d_foot_force_d_bridge_velocity[indices_foot_down]+=
    ↪ d_foot_force_d_bridge_velocity[indices_foot_down]
t_prev_step[indices_foot_down]=t
if which_model!=3:
        for i in indices_foot_down:
           if stability_margin_hof[i]<0:
             tprevL[i]=t_prev_step[i]
        u2=COM_displacement[indices_foot_down]+COM_velocity[
            ↪ indices_foot_down]*np.sqrt(normalized_leg_lengths[
            ↪ indices_foot_down]/9.8)+stability_margin_hof[
            ↪ indices_foot_down]
```

```
        width=stability_margin_hof/(1-np.tanh(np.sqrt(9.81/
            ↪ normalized_leg_lengths)/4.0/step_frequency_hz))
        adapt = np.maximum(-0.5,(width[indices_foot_down]**2-(
            ↪ COM_displacement[indices_foot_down]-u2)**2)/(4*
            ↪ forward_speed[indices_foot_down]**2)) if
            ↪ which_model==2 else 0
        t_next_step[indices_foot_down]=t+0.5/step_frequency_hz[
            ↪ indices_foot_down]*(1+adapt)
step_last[indices_foot_down]=np.maximum(step_last[
    ↪ indices_foot_down],t)
step_first[indices_foot_down]=np.minimum(step_first[
    ↪ indices_foot_down],t)
num_steps[indices_foot_down]+=1
if which_model!=3:
        saggital_prev[indices_foot_down]=sagittal[
            ↪ indices_foot_down]
COM_displacement_prev[indices_foot_down]=COM_displacement[
    ↪ indices_foot_down]
indices_COM_period=np.where(np.signbit(state_prev_sample[2:
    ↪ n_walkers+2])!=np.signbit(state[2:n_walkers+2]))[0]
```

```
t_prev_prev_COM_period[indices_COM_period]=t_prev_COM_period[
    ↪ indices_COM_period]
t_prev_COM_period[indices_COM_period]=t
step_interval_prev=np.zeros(len(step_interval))
step_interval_prev[:]=step_interval[:]
if which_model==3:
        com_offset_current=state[2:n_walkers+2]
        com_offset_prev=state_prev_sample[2:n_walkers+2]
        force_change_over_step_discontinuity=2*
            ↪ inverted_pendulum_frequency_unadjusted_radians[
            ↪ indices_foot_down]**2*np.array(COP_offset_fixed*sgn
            ↪ (com_offset_current[indices_foot_down]-
            ↪ com_offset_prev[indices_foot_down]))
        sagittal_velocity_perturbed[indices_foot_down]=0.0
else:
        force_change_over_step_discontinuity=(u2-
            ↪ COP_lateral_offsets[indices_foot_down])*9.8/
            ↪ normalized_leg_lengths[indices_foot_down]
        step_interval[indices_foot_down]=t_next_step[
            ↪ indices_foot_down]-t_prev_step[indices_foot_down]
```

```
        sagittal_velocity_perturbed[indices_foot_down]=1-
            ↪ step_interval_prev[indices_foot_down]/step_interval
            ↪ [indices_foot_down]
        COP_lateral_offsets[indices_foot_down]=u2
jump_term_1[indices_foot_down]+=(
    ↪ force_change_over_step_discontinuity)*k[1]
jump_term_2[indices_foot_down]+=(
    ↪ force_change_over_step_discontinuity)*np.array(k[2:
    ↪ n_walkers+2][indices_foot_down])
jump_term_3[indices_foot_down]+=(
    ↪ force_change_over_step_discontinuity)*
    ↪ sagittal_velocity_perturbed[indices_foot_down]
jump_term_4[indices_foot_down]+=(
    ↪ force_change_over_step_discontinuity)*np.array(k[2+
    ↪ n_walkers:2+2*n_walkers][indices_foot_down])
t_prev_prev_prev_step[indices_foot_down]=t_prev_prev_step[
    ↪ indices_foot_down]
t_prev_prev_step[indices_foot_down]=t_prev_step[indices_foot_down
    ↪ ]
t_prev_step[indices_foot_down]=t
```

```
stability_margin_hof[indices_foot_down]*=-1
com_offset_current=state[2:n_walkers+2]
com_offset_prev=state_prev_sample[2:n_walkers+2]
if which_model!=3:
        phZ=(t-t_prev_step[:n_walkers])/(t_next_step[:n_walkers]-
            ↪ t_prev_step[:n_walkers])
        order_param_cos_component_step+=np.sum(np.cos(2*np.pi*phZ)
            ↪ )
        order_param_sin_component_step+=np.sum(np.sin(2*np.pi*phZ)
            ↪ )
phase_COM=(t-t_prev_COM_period[:n_walkers])*2*np.pi/(
    ↪ t_prev_COM_period[:n_walkers]-t_prev_prev_COM_period[:
    ↪ n_walkers])
phase_COM=phase_COM[np.where(t_prev_COM_period>
    ↪ t_prev_prev_COM_period)[0]]
order_param_cos_component_COM+=np.sum(np.cos(2*np.pi*phase_COM))
order_param_sin_component_COM+=np.sum(np.sin(2*np.pi*phase_COM))
n_footfalls+=1
n_periods_COM+=len(phase_COM)
mu[indices_foot_down]=0
eta[indices_foot_down]=0
```

```
  d_foot_force_d_bridge_velocity[indices_foot_down]=0
  bridge_max_amplitude=max(bridge_max_amplitude,state[0])
time_from_first_to_last_footfall=(step_last[:n_walkers]-step_first
    ↪ [:n_walkers])
mean_step_frequency=np.mean(0.5*num_steps[:n_walkers]/
    ↪ time_from_first_to_last_footfall)
stdev_step_frequency=np.std(0.5*num_steps[:n_walkers]/
    ↪ time_from_first_to_last_footfall)
total_integration_time=t-t_prev_addition
time_from_first_to_last_zero_cross_bridge=t_final_bridge_zero-
    ↪ t_first_bridge_zero
COM_sin_component_full_period/=(
    ↪ time_from_first_to_last_zero_cross_bridge*np.sqrt(
    ↪ mean_amplitude/time_from_first_to_last_zero_cross_bridge))
COM_cos_component_full_period/=(
    ↪ time_from_first_to_last_zero_cross_bridge*np.sqrt(
    ↪ mean_amplitude/time_from_first_to_last_zero_cross_bridge))
if which_model!=3:
        sagittal_sin_component_full_period/=(
            ↪ time_from_first_to_last_zero_cross_bridge*np.sqrt(
            ↪ mean_amplitude/
```

```
            ↪ time_from_first_to_last_zero_cross_bridge))
        sagittal_cos_component_full_period/=(
            ↪ time_from_first_to_last_zero_cross_bridge*np.sqrt(
            ↪ mean_amplitude/
            ↪ time_from_first_to_last_zero_cross_bridge))
        d_foot_force_d_COM_full_period=-
            ↪ inverted_pendulum_frequency_radians[:n_walkers]**2*
            ↪ time_from_first_to_last_footfall
        d_foot_force_d_COM_velocity_full_period=0
        time_average_d_foot_force_d_bridge_velocity*=9.81/
            ↪ normalized_leg_lengths[:n_walkers]
sigma_1=np.sum((time_average_d_foot_force_d_bridge_velocity+
    ↪ jump_term_1)/time_from_first_to_last_footfall)
sigma_2=np.sum(-(jump_term_2+d_foot_force_d_COM_full_period)*
    ↪ COM_sin_component_full_period/bridge_frequency_radians/
    ↪ time_from_first_to_last_footfall+(jump_term_4+
    ↪ d_foot_force_d_COM_velocity_full_period)*
    ↪ COM_cos_component_full_period/
    ↪ time_from_first_to_last_footfall)
sigma_3=np.sum(sagittal_sin_component_full_period*jump_term_3/
    ↪ time_from_first_to_last_footfall)/bridge_frequency_radians
```

```
    ↪ if which_model!=3 else 0
order_footsteps=np.sqrt(order_param_cos_component_step*
    ↪ order_param_cos_component_step+
    ↪ order_param_sin_component_step*
    ↪ order_param_sin_component_step)/(n_walkers*n_footfalls)
order_COM=np.sqrt(order_param_cos_component_COM*
    ↪ order_param_cos_component_COM+order_param_sin_component_COM*
    ↪ order_param_sin_component_COM)/(n_periods_COM)
print(n_walkers,
                      bridge_damping_coefficient-n_walkers*np.mean(
                          ↪ walker_masses[:n_walkers])/bridge_mass*(
                          ↪ sigma_1+sigma_2+sigma_3),
                      bridge_max_amplitude,
                      order_footsteps,
                      order_COM,
                      mean_step_frequency,
                      stdev_step_frequency,
                      np.mean(forward_speed),
                      np.sqrt(mean_amplitude/
                          ↪ time_from_first_to_last_zero_cross_bridge
                          ↪ ),
```

```
                                num_zero_cross/(
                                    ↪ time_from_first_to_last_zero_cross_bridge
                                    ↪ ),
                                sigma_1, sigma_2, sigma_3
                                )
# if bridge_damping_coefficient-n_walkers*np.mean(walker_masses[:n_walkers])
    ↪ /bridge_mass*(sigma_1+sigma_2+sigma_3)<0 and break_on_crit:
          if bridge_max_amplitude>0.015 and break_on_crit:
               sys.exit(n_walkers)
          sys.stdout.flush()
sys.exit(max_n_walkers)
```

B Julia source code listing: pedestrian/bridge and social force simulations

```
using Pkg;
Pkg.add("Plots")
Pkg.add("CUDA")
Pkg.add("ForwardDiff")
Pkg.add("DifferentialEquations")
using CUDA
using ForwardDiff
CUDA.versioninfo()
```

```
using Plots

using LinearAlgebra

using DifferentialEquations

using Random

bridge_half_width=2

bridge_length=325

ra = 0.31

la = 0.31

A1=1.7

A2 = 1.7

B1 = 0.28

B2=0.28

tau=0.5

Ai=5

Bi =0.1

M=113e3

bridge_damp=0.043

Omega=2*pi*1.03



function social_force_term(diffzx,diffzy)

    distz=sqrt.(diffzx^2+diffzy^2)
```

```
    d=max.(distz,1e-8)
    nx=diffzx/d
    ny=diffzy/d
    cosphi=-ny
    F1 = 1.7*exp((2*0.31 - d) / 0.28) * (0.31 + (1 - 0.31) * ((1 + cosphi) /
        ↪  2)) + 1.7 * exp((2*0.31 - d) / 0.28)
    F1*nx,F1*ny
end
function wall_avoidance_term(z)
    bound=2
    leftwall=min.(z+bound,-1e-7)
    rightwall=max.(z-bound,1e-7)
    left_repel=5*exp.((0.31-abs.(leftwall))./0.1).*leftwall
    right_repel=5*exp.((0.31-abs.(rightwall))./0.1).*rightwall
    left_repel.+right_repel
end
function social_force_frontiers(F::CuDeviceVector{Float64},N::Int64,Z::
    ↪ CuDeviceMatrix{Float64},Zdot::CuDeviceMatrix{Float64},Vd::
    ↪ CuDeviceVector{Float64})

    Zshared = CUDA.@cuDynamicSharedMem(Float64, N*2)
```

```
i=threadIdx().x
z1::Float64=convert(Float64,0)
z2::Float64=convert(Float64,0)
if i<=N
    z1=Z[i,1]
    z2=Z[i,2]
    Zshared[(i-1)*2+1]=z1
    Zshared[(i-1)*2+2]=z2
end
    sync_threads()
if i<=N
    Fix::Float64=0
    Fiy::Float64=0
    for j in range(1,N)
        zj1=Zshared[(j-1)*2+1]
        zj2=Zshared[(j-1)*2+2]
        fx,fy=social_force_term(z1-zj1,z2-zj2)
        Fix+=(i==j) ? 0 : fx
        Fiy+=(i==j) ? 0 : fy
    end
    Fix+=wall_avoidance_term(z1)
```

```
        F[i]=0.5*(-Zdot[i,1]).+Fix
        F[i+N]=0.5*(Vd[i]-Zdot[i,2]).+Fiy
    end
    return nothing
end


function test_force(x,y)
  norm(social_force_term(
               x, y
                )
    )
end




g=9.81
function H_1(y,p,L)
    g./L.*(p-y)
end
function bridge_ode(deriv::CuDeviceVector{Float64},
                    N::Int64,
                    L::CuDeviceVector{Float64},
```

```
                p::CuDeviceVector{Float64},
                rsum::Float64,
                msum::Float64,
                m::CuDeviceVector{Float64},
                state::CuDeviceVector{Float64},
                M::Float64,
                Omega::Float64,
                bridge_damp::Float64)
i=convert(Int32,threadIdx().x)
foot_force=convert(Float64,0)
g_=convert(Float64,9.81)
deriv_shared = CUDA.@cuDynamicSharedMem(eltype(deriv), 1)
if i==1
    deriv_shared[1]=0
end
sync_threads()
if i<=N
    r=m[i]/(M+msum)
    y0dot=state[2+5*N+i]
    zdot=state[2+4*N+i]
    vnorm=sqrt(zdot^2+y0dot^2)
```

```
        foot_force=g_/L[i]*(p[i]-state[i+2])*(carroll(vnorm)>=0.3)
        deriv_term=1/(1-rsum)*(foot_force*r)
        CUDA.atomic_add!(CUDA.pointer(deriv_shared),deriv_term)
    end
    sync_threads()
    if i==1
            xdot=state[2]
            x=state[1]
            deriv[i]=xdot
            xdotdot=1/(1-rsum)*(-bridge_damp*Omega*xdot-Omega*Omega*x)
            deriv_shared[1]+=xdotdot
            deriv[i+1]=deriv_shared[1]
    end
    sync_threads()
    if i<=N
        deriv[i+2+3*N]=-foot_force-deriv_shared[1]
    end
    return nothing
end


function bridge_pedestrian_ode_model2(deriv,state,param,t)
```

```
N,L,m,p,vd,msum,rsum=param

y0=state[3+2*N:2+3*N]

y1=state[3:2+N]

y=min.(max.(y1+y0,-bridge_half_width),bridge_half_width)

z=state[3+N:2+2*N] .%bridge_length

ydot=state[3+3*N:2+4*N]

zdot=state[3+4*N:2+5*N]

y0dot=state[3+5*N:2+6*N]

M_=convert(Float64,M)

Omega_=convert(Float64,Omega)

bridge_damp_=convert(Float64,bridge_damp)

@sync @cuda blocks=1 threads=N shmem=sizeof(Float64) bridge_ode(deriv,N,
    ↪ L,p,rsum,msum,m,state,M_,Omega_,bridge_damp_)

Z=CuArray{Float64}(CUDA.zeros(N,2))

Zdot=CuArray{Float64}(CUDA.zeros(N,2))

Z[:,1].=y

Z[:,2].=z

Zdot[:,1].=y0dot

Zdot[:,2].=zdot

crowd=CuArray{Float64}(CUDA.zeros(N*2))
```

```
    @sync @cuda blocks=1 threads=N shmem=2*N*sizeof(Float64)
        ↪ social_force_frontiers(crowd,N,Z,Zdot,vd)
    vnorm=sqrt.(y0dot.^2+zdot.^2)
    deriv[3:N+2].=ydot.*(carroll(vnorm).>=0.3)
    deriv[3+N:2+2*N].=zdot
    deriv[3+2*N:2+3*N].=y0dot
    deriv[3+4*N:2+5*N].=crowd[N+1:2*N]
    deriv[3+5*N:2+6*N].=crowd[1:N]
end


function carroll(v)
    0.5*(0.35*v.^3 - 1.59*v.^2 + 2.93*v)
end
function deriv_carroll(v)
    0.5*(0.35*3*v.^2-1.59*2*v .+2.93)
end
function inverse_carroll(v0,fp)
    v=v0
    for it in range(1,100)
        v.-=0.99*(carroll(v)-fp)./deriv_carroll(v)
    end
```

```
    v
end


function generate_crowd(N,mean_freq,std_freq)
    L=randn(N)*0.092.+1.17
    bmin=0.0157.+0.002*randn(N)
    bmin.*=1 .-2*rand(N)
    y0=rand(N)*4 .-2
    y0dot=zeros(N)
    y=rand(N)*0.1-0.2
    z=rand(N)*325
    p=bmin.*(1.0-tanh.(0.25*sqrt.(9.8./L) ./ fp))
    bmin.*-1
    x=0.
    xdot=0.
    m=76.9 .+10*randn(N)
    vd=inverse_carroll(ones(N),abs.(randn(N)*std_freq .+mean_freq))
    vd=max.(0.1,vd)
    fp=carroll(vd)
    ydot= p .* sqrt.(9.81./L) .* tanh.(sqrt.(9.81./L) .* 0.5./fp)
    tnext=rand(N)*0.5 ./fp
```

```
    tprev=tnext-0.5 ./fp
    msum=convert(Float64,sum(m))
    r=m./(M+msum)
    rsum=convert(Float64,sum(r))
    cat([x,xdot],y,z,y0,ydot,vd,y0dot,dims=1), (L,m,p,tnext,tprev,bmin,vd,
        ↪ msum,rsum)
end




function crowd_loop(Tmax,N,crowd_state,crowd_param, tracefile)
    t=0
    t_trace_last=-0.1
    L,m,p,tnext,tprev,bmin,vd,msum,rsum=crowd_param
    L_d=CuArray{Float64}(L)
    vd_d=CuArray{Float64}(vd)
    m_d=CuArray{Float64}(m)
    while t<Tmax
        #solve the ODE until the next pedestrian footfall
        L,m,p,tnext,tprev,bmin,vd,msum,rsum=crowd_param
```

```
p_d=CuArray{Float64}(p)

crowd_param_gpu=N,L_d,m_d,p_d,vd_d,msum,rsum

t1=minimum(tnext)

if t1-t>1e-10

  prob=ODEProblem(bridge_pedestrian_ode_model2,CuArray{Float64}(
     ↪ crowd_state), (0,t1-t), crowd_param_gpu)

  sol=solve(prob, AutoVern7(AutoTsit5(Rosenbrock23()),
        abstol=1e-13,reltol=1e-10,save_everystep = false)

  crowd_state=Array(sol.u[size(sol.u)[1]])

end

#update the step timing and xCOP for the pedestrians that stepped...

ii=(tnext .<=t1+1e-10)

zdot=crowd_state[3+4*N:2+5*N]

y0=crowd_state[3+2*N:2+3*N]

y0dot=crowd_state[3+5*N:2+6*N]

vnorm=sqrt.(zdot.^2+y0dot.^2)

crowd_state[3+3*N:2+4*N].*=(carroll(vnorm).>=0.3)

ydot=crowd_state[3+3*N:2+4*N]

y1=crowd_state[3:2+N]

bnd=bridge_half_width
```

```
p[ii].=y1[ii]+(ydot[ii].*sqrt.(L[ii]./9.81).*(carroll(vnorm[ii])
   ↪ .>=0.3))+bmin[ii]
bmin[ii].=-bmin[ii]
tprev[ii].=tnext[ii]
fp=max.(0.3,carroll(vnorm))
step_width=abs.(bmin)./(1 .-tanh.(sqrt.(0.91 ./L).*0.25./fp))
step_length=max.(1e-1,0.36*(zdot/1.151466))
tnext[ii].=tnext[ii].+max.(0.1,0.5./fp[ii].*
(1 .+(carroll(vnorm[ii]).>=0.3).*(step_width[ii].^2-(y1[ii]-p[ii])
   ↪ .^2)./(4*step_length[ii].^2)))
t=t1
#increase the time and, at a certain interval, write out the traces
   ↪ to the file
if t_trace_last+0.01<=t
    z=crowd_state[3+N:2+2*N]
    x,xdot=crowd_state[1],crowd_state[2]
    t_trace_last=t
    line="$(N) $(t) $(x) $(xdot) "
    for i in range(1,N)
        line=line*" $(g/L[i]*(p[i]-y1[i])) "
    end
```

```
for i in range(1,N)
    line=line*" $(p[i]-y1[i]) "
end
for i in range(1,N)
    line=line*" $(y1[i]) "
end
for i in range(1,N)
    line=line*" $(y0[i]) "
end
for i in range(1,N)
    line=line*" $(z[i]) "
end
for i in range(1,N)
    line=line*" $(ydot[i]) "
end
for i in range(1,N)
    line=line*" $(y0dot[i]) "
end
for i in range(1,N)
    line=line*" $(zdot[i]) "
end
```

```
    for i in range(1,N)
        line=line*" $(bmin[i]) "
    end
    for i in range(1,N)
        line=line*" $(tnext[i]) "
    end
    for i in range(1,N)
        line=line*" $(tprev[i]) "
    end
    for i in range(1,N)
        line=line*" $(fp[i]) "
    end



    for i in range(1,N)
        line=line*" $(vd[i]) "
    end
    write(tracefile,line*"\n")
    flush(tracefile)
end #if
crowd_param=(L,m,p,tnext,tprev,bmin,vd,msum,rsum)
```

```
    end #while
end #fn
function compute_traces!(N, std_freq, sample)
    Tmax=20
    mean_freq=0.95
    CURAND.seed!(sample)
    Random.seed!(sample)
    tracefile=open("crowd-traces/$(N)_$(mean_freq)_$(std_freq)_$(sample).txt
        ↪ ","w+")

                #initial conditions
    crowd_state, crowd_param=generate_crowd(N,mean_freq,std_freq)
    L,m,p,tnext,tprev,bmin,vd=crowd_param
    crowd_loop(Tmax,N,crowd_state,crowd_param,tracefile)
    nothing
end
for N in range(160,250,step=5)
    for sample in range(1,10)
      @sync begin
                for sigma in range(0,1.0,step=0.1)
                    @async begin
```

```
                        compute_traces!(N,sigma,sample)
                    end
                end
            end
    end
end
```

CHAPTER B

Appendix for Chapter 4

A Python source code listing: toxicity detection and reinforcement learning

```
import pickle

import matplotlib.pyplot as plt

import sys, numpy as np

import glob

import scipy.signal

from sklearn.cluster import MiniBatchKMeans as KMeans

import sklearn

from sklearn.svm import SVR

from sklearn.preprocessing import StandardScaler

from sentence_transformers import SentenceTransformer

from sklearn.metrics import r2_score, mean_absolute_percentage_error,
    ↪ accuracy_score

from torch.optim import Adam

from transformers import AdamW

import pandas as pd

import torch.nn as nn

import torch

TWT=False
```

```
from transformers import DistilBertModel

import open_clip

# Instantiate DistilBERT tokenizer...we use the Fast version to optimize

    ↪ runtime

Load=True

Train=False

RELABEL=False

from transformers import AutoTokenizer, AutoModel, AutoModelForMaskedLM

from sentence_transformers import SentenceTransformer

from sentence_transformers import evaluation

import functools

from torch.utils.data import Dataset, DataLoader


if RELABEL:

    labels=pd.read_csv('pol_train0.csv').sample(200)

    keys=np.arange(200).tolist()

    text=labels['tokens'].tolist()

    comparisons=0

    def compare(i,j):

        while True:

          try:
```

```
            global comparisons
            prompt='1:'+str(text[i])+'\n\n2:'+str(text[j])+'\n'
            print(comparisons,prompt)
            comparisons+=1
            val=int(input())
            if val==0:
                return 0
            elif val==1:
                return -1
            else:
                return 1
          except:
             continue
    keys=sorted(keys,key=functools.cmp_to_key(compare))
    labels['TOXICITY']=np.linspace(0,1,len(labels))[keys]
    labels['text']=text
    labels.to_csv('pol_train.csv')



def mean_pooling(model_output, attention_mask):
```

```
    token_embeddings = model_output[0] #First element of model_output
        ↪ contains all token embeddings
    input_mask_expanded = attention_mask.unsqueeze(-1).expand(
        ↪ token_embeddings.size()).float()
    return torch.sum(token_embeddings * input_mask_expanded, 1) / torch.
        ↪ clamp(input_mask_expanded.sum(1), min=1e-9)


class BertRegressor(nn.Module):


    def __init__(self, model_str='distilbert-base-4ch', drop_rate=0.2,
        ↪ device='cuda'):
        self.model_device=device
        self.regressor_device=device
        super(BertRegressor, self).__init__()
        self.model = AutoModelForMaskedLM.from_pretrained(model_str).to(
            ↪ device)
        self.model.train()
        self.tokenizer=AutoTokenizer.from_pretrained('distilbert-base-
            ↪ uncased', use_fast=True)
        self.regressor1 = nn.Sequential(
            nn.Dropout(drop_rate),
```

```
        nn.Linear(768,256),
        nn.ReLU(),
        nn.Dropout(drop_rate),
        nn.Linear(256,1)).to(device)
    self.params=list(self.regressor1.parameters())+list(self.model.
        ↪ parameters())
    self.freeze=False
def freeze_bert(self):
    self.model.save_pretrained('distilbert-base-4ch2')
    self.model=AutoModel.from_pretrained('distilbert-base-4ch2').cuda()
    self.freeze=False
def unfreeze_bert(self):
    self.model.train()
    self.freeze=False
def forward(self, x, key='text', device='cuda'):
    if self.regressor_device!=device:
        self.regressor1=self.regressor1.to(device)
        self.regressor_device=device
    return self.predict(self.encode(x,key=key,device=device))
def encode(self, x, key='text', device='cuda'):
    if self.model_device!=device:
```

```
            self.model=self.model.to(device)
            self.model_device=device
        inputs=self.tokenizer(x[key].apply(str).tolist(), truncation=True,
            ↪ padding=True, return_tensors="pt")
        if self.freeze:
            with torch.no_grad():
                model_output = self.model(inputs['input_ids'].to(device),
                    ↪ inputs['attention_mask'].to(device))
        else:
                model_output = self.model(inputs['input_ids'].to(device),
                    ↪ inputs['attention_mask'].to(device))
        sentence_embeddings = mean_pooling(model_output, inputs['
            ↪ attention_mask'].to(device))
        return sentence_embeddings
    def predict(self,x):
        y = torch.sigmoid(self.regressor1(x))
        return y
def encode_in_batches(model,df,key='text'):
        batches_out=[]
        for i in range(0,int(np.ceil(len(df)/10))):
           batch= df.iloc[i*10:min(i*10+10,len(df))]
```

```
            batches_out.append(model.encode(batch,key=key).detach().cpu().
                ↪ numpy())
            print(i)
        return np.concatenate(batches_out,axis=0)
def regressor_predict(x):
     reward_model.regressor1=reward_model.regressor1.to('cpu')
     return reward_model.predict(torch.tensor(x)).detach().cpu().numpy()


import sys
from torch.nn.utils.clip_grad import clip_grad_norm_


from transformers import get_linear_schedule_with_warmup
def bert_score(model, tokenizer, tensor_input):
            mask=(torch.rand(tensor_input.shape)<0.15)
            masked_input = tensor_input+0
            masked_input[mask]=tokenizer.mask_token_id
            loss = model(masked_input.cuda(), labels=tensor_input.cuda()).loss
            return loss
def train(model):
    df20=pd.read_csv('pol_val.csv')
    optimizer1 = Adam(model.params,
```

```
            lr=0.001)
optimizer2 = AdamW(model.params,
            lr=0.0001)
model=model
model.train()
lossfn=torch.nn.BCELoss()
chunker0=pd.read_csv('pol_train.csv', chunksize=5000)
model.unfreeze_bert()
bz=512
bs=4
val_loss_avg=[]
for epoch in range(0):
    for batch,df0 in enumerate(chunker0):
        toks=model.tokenizer.encode(' '.join(df0['text'].apply(str).
            ↪ sample(frac=0.2)),return_tensors='pt')
        toks_train=toks[0][:-bs*bz]
        toks_val=toks[0][-bs*bz:]
        loss_avg=[]
        for i in range(0,int((len(toks_train)/bz/bs))):
            optimizer2.zero_grad()
            sentence=toks_train[i*bs*bz:i*bs*bz+bs*bz].reshape([bs,-1])
```

```
                loss = bert_score(model.model,model.tokenizer,sentence)
                loss.backward()
                loss_avg.append(loss.item())
# clip_grad_norm_(model.model.parameters(),2)
                optimizer2.step()
                sys.stdout.flush()
            with torch.no_grad():
                val_loss=bert_score(model.model, model.tokenizer, toks_val.
                    ↪ reshape([bs,-1]))
                val_loss_avg.append(val_loss.item())
            print(epoch,batch,sum(loss_avg)/len(loss_avg), val_loss.item())
            sys.stdout.flush()
            model.model.save_pretrained('distilbert-4ch-finetuned2')
    model.freeze_bert()
    chunker=pd.read_csv('all_data.csv')
    for epoch in range(100):
        best_loss = 1e10
        for batch in range(100):
                df0=chunker[chunker['split']=='train']
                df2=df20.sample(n=25)
                df=pd.concat(
```

```
                [df0[df0['toxicity']>0.5].sample(n=10),
                df0[df0['toxicity']<=0.5].sample(n=10)]
                )

            y=torch.tensor(np.array(df['toxicity'])).float().cuda()
            y2=torch.tensor(np.array(df2['TOXICITY'])).float().cuda()
            optimizer1.zero_grad()
            outputs = model(df,key='comment_text')
            loss0 = lossfn(outputs.squeeze(),
                           y.squeeze())
            loss0.backward()
            optimizer1.step()
            with torch.no_grad():
                outputs2=model(df2)
                val_loss=lossfn(outputs2.squeeze(), y2.squeeze())
            print(epoch,batch,loss0.item(), val_loss.item(), sklearn.
                ↪ metrics.f1_score(y2.detach().cpu().numpy()>0.5,outputs2
                ↪ .detach().cpu().numpy()>0.5))
            sys.stdout.flush()
            torch.save(model,'bert_toxicity2.pt')
    return model
```

```
if Load:
    reward_model=torch.load('bert_toxicity2.pt')
    print('load model')
else:
    reward_model = BertRegressor(drop_rate=0.2)
if Train:
    reward_model=train(reward_model)
    print('train')
reward_model.eval()
dft=pd.read_csv('~/Downloads/test_private_expanded.csv').sample(n=1500)
dft2=pd.read_csv('pol_test.csv').sample(n=1500)
regressor=reward_model
pred=regressor_predict(encode_in_batches(reward_model,dft,key='comment_text
    ↪ '))
pred2=regressor_predict(encode_in_batches(reward_model,dft2,key='text'))
R1t=np.array(dft['toxicity'])
R2t=np.array(dft2['TOXICITY'])
print('r2=',r2_score(R1t,pred),', mean_abs_%_error=',
    ↪ mean_absolute_percentage_error(R1t,pred), accuracy_score(R1t>0.4,pred
    ↪ >0.4))
```

```
print('r2=',r2_score(R2t,pred2),', mean_abs_%_error=',
    ↪ mean_absolute_percentage_error(R2t,pred2), accuracy_score(R2t>0.4,
    ↪ pred2>0.4))
print(sklearn.metrics.classification_report(R1t>0.4, pred>0.4))
print(sklearn.metrics.classification_report(R2t>0.4, pred2>0.4))
print('toxicity_train')
from cuml.manifold import UMAP
if TWT:
    bertmodel=reward_model
    twt=pd.read_csv('twt_firearm_min.csv').iloc[::-1]
pickle.dump(reward_model,open('KNN_reward.pkl','wb+'))


reducer = UMAP(n_components=2)
A0=np.loadtxt('4plebs/training2.np')
B0=reducer.fit_transform(A0[::1])
print('umap_train')
pickle.dump(reducer,open('umap.pkl','wb+'))
infer_files=[f'4plebs/pol_embed.np.{index}' for index in range(26600,26643)]
xl=[np.mean(B0[:,0])-np.std(B0[:,0]),np.mean(B0[:,0])+np.std(B0[:,0])*0.8]
yl=[np.mean(B0[:,1])-np.std(B0[:,1]),np.mean(B0[:,1])+np.std(B0[:,1])*0.8]
i=26599
```

```
indexof=lambda x: int(x.split('.')[-1])

infer_files=sorted(infer_files, key=indexof)

#A2=np.loadtxt(infer_files[1])

#A1=np.loadtxt(infer_files[0])

import scipy.stats

import pandas as pd

import dateparser

Nc=1000

N=30

d0=dateparser.parse('2013-01-01T00:00:00')

d0z=dateparser.parse('2013-01-01T00:00:00.000Z')

d0zz=pd.Timestamp('2013-01-01T00:00:00.000Z')

timediff = lambda x: (dateparser.parse(str(x))-d0).total_seconds()

timediffz= lambda x: (dateparser.parse(str(x))-d0z).total_seconds()

file2s=[pd.read_csv(f'4plebs/pol_tok.csv.{indexof(infer_files[k])}') for k
    ↪ in range(N)]

from nltk.tokenize.casual import casual_tokenize

#As=[np.loadtxt(infer_files[k]) for k in range(N)]

if TWT:

    twtdate=np.array((pd.to_datetime(twt['created_at'])-d0zz).dt.
        ↪ total_seconds())
```

```
print('load')
def get_embeddings(x):
       return ' '.join(casual_tokenize(str(x)))
for j,file in enumerate(infer_files[1:-N]):
  ix=0
  file2s=file2s[1:]+[pd.read_csv(f'4plebs/pol_tok.csv.{indexof(infer_files[j
     ↪ +N-1])}')]
  with torch.no_grad():
     As=[encode_in_batches(reward_model,x,key='tokens') for x in file2s]
  if i%10!=0:
     i+=1
     continue
  i+=1
  times=[np.array(file2['date'].apply(timediff)) for file2 in file2s]
  time=np.concatenate(times)
  T0=time[0]
  T1=time[-1]
  if TWT:
    text=[file2['tokens'] for file2 in file2s]
    txt0=twt['text'][np.logical_and(twtdate>=T0, twtdate<=T1)].apply(
       ↪ get_embeddings)
```

```
  print('tokenize')
  txt=bertmodel.encode(txt0.tolist(), convert_to_numpy=True).astype(float)
  print('embed')
  print(txt.shape)
  rtc=np.array(twt['public_metrics.retweet_count'][np.logical_and(twtdate
      ↪ >=T0,twtdate<=T1)]).astype(int)
A4=np.concatenate(As,axis=0)
file3=pd.concat(file2s)
KM=KMeans(Nc)
embedding=reducer.transform(A4)
label=np.array(KM.fit_predict(A4))
file3['label']=label
P=np.bincount(label).astype(np.float32)
P/=P.sum()
Vk=np.random.rand(Nc)/Nc
Nk=np.ones(Nc)
Vshuf=np.random.rand(100,Nc)/Nc
Nshuf=np.ones([100,Nc])
if TWT:
    jj=np.zeros(len(txt),dtype=int)-1
    dist=np.ones(len(txt))*1e-2
```

```
    for ii,cc in enumerate(KM.cluster_centers_):
        dist2=np.linalg.norm(cc[None,:]-txt,axis=-1)
        jj=ii*(dist2<dist)*1+jj*(dist<=dist2)*1
        dist=np.minimum(dist,dist2)
    kk=np.where(jj>-1)[0]
    jj=np.array(jj)[kk]
    reward_twt=reward[jj.astype(int)]
    DD=np.stack([reward_twt,rtc[kk]],axis=-1)
    np.savetxt('frames/%07d.rtc.np'%i,DD)
    plt.scatter(reward_twt,rtc[kk])
    plt.xlabel('Toxicity',fontsize=20)
    plt.ylabel('RT count',fontsize=20)
    plt.savefig('frames/%07d.rtc.png'%i,dpi=400)
L=np.unique(label)
for it in range(1):
     rewards=[]
     for t in range(A4.shape[0]-1):
         rewards.append(regressor_predict(A4[t]))
         Vk[label[t]]+=0.25*(rewards[-1]+0.99*Vk[label[t+1]]-Vk[label[t]])
print('value_iter')
logitP=np.nan_to_num(np.log(P))
```

```
 P0=P*1.0
 plt.scatter(Vk[L]/Nk[L],logitP[L],s=10,alpha=0.9,c='cornflowerblue')
 coef=np.linalg.lstsq(np.stack([Vk[L],np.ones(len(L))],axis=-1),logitP[L])
    ↪ [0]
#plt.plot(Vk[L],coef[0]*Vk[L]+coef[1],'k--')
 plt.xlim([Vk[L].min(), Vk[L].max()])
 plt.ylim([logitP[L].min(),0])
 plt.xlabel('TD Value',fontsize=20)
 plt.ylabel('$S^{-1}(P)$',fontsize=20)
 plt.savefig('frames/%07d.td_value_logit.png'%i, dpi=1000)
 file3['value']=Vk[label]
 file3['P']=P[label]
 file3.to_csv(f'4plebs/pol_labeled_VF.csv.{i}')
 np.savetxt('frames/%07d.td_value.np'%i,np.stack([Vk[L],P[L]],axis=-1))
 plt.clf()
 if True:
   A=A4[:]
   np.savetxt(f'frames/centroids.{i}.np',KM.cluster_centers_)
   print(i)
   plt.scatter(embedding[:,0], embedding[:,1], c=label, s=0.5,alpha=np.
      ↪ minimum(6*P[label],1), cmap='gist_stern')
```

```
    plt.xlim(xl)
    plt.ylim(yl)
    plt.savefig('frames/%07d.cluster.png'%i, dpi=600)
    plt.clf()
    h,_,_,_=plt.hist2d(embedding[:,0], embedding[:,1], bins=200, range=[xl,
        ↪ yl], density=True, weights=Vk[label], cmap='magma')
    plt.xlim(xl)
    plt.ylim(yl)
    plt.clf()
    plt.pcolor(h.T, cmap='magma')
    cb=plt.colorbar()
    plt.savefig('frames/%07d.utility.png'%i, dpi=600)
    plt.clf()
# for shuf in range(100):
# i_shuf=np.random.permutation(np.arange(len(A4)-1))
# label_shuf=label[i_shuf]
# reward_shuf=np.array(rewards)[i_shuf]
# for it in range(1):
# for t in range(reward_shuf.shape[0]-1):
# Vshuf[shuf,label_shuf[t]]+=0.25*(reward_shuf[t]+0.99*Vk[label_shuf[t+1]]-
    ↪ Vk[label_shuf[t]])
```

```
#cshuf=np.array([scipy.stats.kendalltau(Vshuf[i,L]/Nshuf[i,L],logitP[L])
    ↪ [0] for i in range(100)])
#r=scipy.stats.kendalltau(Vk[L]/Nk[L],logitP[L])[0]
#print('shuf_test')
#pk=1-(cshuf<r).mean()
#cutoff=np.argsort(cshuf)[-1]
#cutoff2=np.argsort(cshuf)[0]
#r2=scipy.stats.kendalltau(Vshuf[cutoff,L]/Nshuf[cutoff,L],logitP[L])[0]
#r3=scipy.stats.kendalltau(Vshuf[cutoff2,L]/Nshuf[cutoff2,L],logitP[L])
    ↪ [0]
#print(r,pk,r2,r3)
#plt.scatter(Vshuf[cutoff,L]/Nshuf[cutoff,L],P[L],s=10,alpha=0.9,c='gray
    ↪ ')
#plt.xlabel('TD Value',fontsize=20)
#plt.ylabel('P',fontsize=20)
#plt.savefig('frames/%07d.td_value_shuf_max.png'%i, dpi=1000)
plt.clf()
plt.scatter(Vk[L]/Nk[L],P[L],s=10,alpha=0.9,c='cornflowerblue')
plt.xlabel('TD Value',fontsize=20)
plt.ylabel('P',fontsize=20)
plt.savefig('frames/%07d.td_value.png'%i, dpi=1000)
```

```
plt.clf()

plt.plot(file3['date'][:len(label)],Vk[label],linewidth=0.25)

plt.xticks([file3['date'].min(),file3['date'].max()])

plt.xlabel('time',fontsize=20)

plt.ylabel('TD Value',fontsize=20)

plt.savefig('frames/%07d.td_value_t.png'%i, dpi=1000)

plt.clf()

np.savetxt('frames/%07d.td_value_t.np'%i,np.stack([label,Vk[label],P[
    ↪ label]],axis=-1))

plt.scatter(embedding[:,0], embedding[:,1], c=Vk[label]/Nk[label], alpha
    ↪ =0.4, cmap='magma', s=1)

plt.xlim(xl)

plt.ylim(yl)

cb=plt.colorbar()

plt.savefig('frames/%07d.V.png'%i, dpi=600)

plt.clf()

np.savetxt(f'frames/%07d.umap.np'%i,embedding[:,:])

np.savetxt(f'frames/%07d.umap_centroids.np'%i,embedding[:,:])

np.savetxt(f'frames/%07d.clust.np'%i,label[:])

np.savetxt(f'frames/%07d.toxic.np'%i,rewards[:])

print('saved')
```

REFERENCES

[1] Franklin Henry Giddings. *Sociology (a lecture delivered at Columbia university in the series on science, philosophy and art, February 26, 1908)*. Columbia University Press, 1908.

[2] Vincent Hakim and Pascal Silberzan. Collective cell migration: a physics perspective. *Reports on Progress in Physics*, 80(7):076601, 2017.

[3] J Tailleur and ME Cates. Statistical mechanics of interacting run-and-tumble bacteria. *Physical review letters*, 100(21):218103, 2008.

[4] Michele Ballerini, Nicola Cabibbo, Raphael Candelier, Andrea Cavagna, Evaristo Cisbani, Irene Giardina, Vivien Lecomte, Alberto Orlandi, Giorgio Parisi, Andrea Procaccini, et al. Interaction ruling animal collective behavior depends on topological rather than metric distance: Evidence from a field study. *Proceedings of the national academy of sciences*, 105(4):1232–1237, 2008.

[5] Bard Ermentrout. An adaptive model for synchrony in the firefly pteroptyx malaccae. *Journal of Mathematical Biology*, 29(6):571–585, 1991.

[6] M. Aureli, F. Fiorilli, and M. Porfiri. Portraits of self-organization in fish schools interacting with robots. *Physica D: Nonlinear Phenomena*, 241(9):908–920, 2012.

[7] Philipp Bagus, José Antonio Peña-Ramos, and Antonio Sánchez-Bayón. Covid-19 and the political economy of mass hysteria. *International Journal of Environmental Research and Public Health*, 18(4):1376, 2021.

[8] M. Friedman. *Essays in Positive Economics*. A Phoenix book. Business economics. University of Chicago Press, 1953.

[9] L. Ljungqvist and University of Wisconsin-Madison. Social Systems Research Institute. *Destabilizing Exchange Rate Speculation: A Counterexample to Milton Friedman*. Institutet för Internationell Ekonomi Stockholm: Seminar paper. Social Systems Research Institute, University of Wisconsin, 1993.

[10] Igor Belykh, Mateusz Bocian, Alan R Champneys, Kevin Daley, Russell Jeter, John HG Macdonald, and Allan McRobie. Emergence of the london millennium bridge instability without synchronisation. *Nature Communications*, 12(1):7223, 2021.

[11] IV Belykh, KM Daley, and VN Belykh. Pedestrian-induced bridge instability: The role of frequency ratios. *Radiophysics and Quantum Electronics*, 64(10):700–708, 2022.

[12] Igor Belykh, Mateusz Bocian, Alan Champneys, Kevin Daley, Russell Jeter, John Macdonald, and Allan McRobie. The london millennium footbridge revisited: Emergent instability without synchronization. *SIAM News*, 55(3), 2022.

[13] Reimbay Reimbayev, Kevin Daley, and Igor Belykh. When two wrongs make a right: synchronized neuronal bursting from combined electrical and inhibitory coupling. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 375(2096):20160282, 2017.

[14] Longkun Tang, Kelley Smith, Kevin Daley, and Igor Belykh. When multilayer links exchange their roles in synchronization. *Physical Review E*, 106(2):024214, 2022.

[15] Kevin Daley, Kun Zhao, and Igor V Belykh. Synchronizability of directed networks:

The power of non-existent ties. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 30(4), 2020.

[16] W. Ditto. Synchronization: A universal concept in nonlinear sciences. *Nature*, 415:736–737, 2002.

[17] S.H. Strogatz. Exploring complex networks. *Nature*, 410(6825):268–276, 2001.

[18] J. Yan, M. Bloom, C. B., Sung, E. Luijten, and S. Granick. Linking synchronization to self-assembly using magnetic Janus colloids. *Nature*, 491(7425):578, NOV 22 2012.

[19] L. Aron and B.A. Yankner. Neural synchronization in Alzheimer's disease. *Nature*, 540(7632):207–208, 2016.

[20] A.E. Motter, S.A. Myers, M. Anghel, and T. Nishikawa. Spontaneous synchrony in power-grid networks. *Nature Physics*, 9:191–197, 2013.

[21] A. Aubret, M. Youssef, S. Sacanna, and J. Palacci. Targeted assembly and synchronization of self-spinning microgears. *Nature Physics*, 14(11):1114–1118, 2018.

[22] A.C. Barato. The cost of synchronization. *Nature Physics*, 16:5, 2020.

[23] Y. Ma, E.W.M. Lee, M. Shi, and R.K.K. Yuen. Spontaneous synchronization of motion in pedestrian crowds of different densities. *Nature Human Behaviour*, 2021. doi: 10.1038/s41562-020-00997-3.

[24] P. Dallard, A.J. Fitzpatrick, A. Flint, A. Low, R. Ridsdill Smith, M. Willford, and M. Roche. London Millennium Bridge: pedestrian-induced lateral vibration. *Journal of Bridge Engineering*, 6(6):412–417, 2001.

[25] S.H. Strogatz, D.M. Abrams, A. McRobie, B. Eckhardt, and E. Ott. Crowd synchrony

on the Millennium Bridge. *Nature*, 438(7064):43–44, 2005.

[26] I.V. Belykh, R. Jeter, and V.N. Belykh. Foot force models of crowd dynamics on a wobbly bridge. *Science Advances*, 3:e1701512, 2017.

[27] J.H.G. Macdonald. Pedestrian-induced vibrations of the Clifton Suspension Bridge. *Proceedings of the Institution of Civil Engineers-Bridge Engineering*, 161(2):69–77, 2008.

[28] J.M.W. Brownjohn, P. Fok, M. Roche, and P. Omenzetter. Long span steel pedestrian bridge at Singapore Changi Airport — Part 2: crowd loading tests and vibration mitigation measures. *The Structural Engineer*, 82(16):21–27, 2004.

[29] P. Dallard, A.J. Fitzpatrick, A. Flint, S. Le Bourva, A. Low, R.M. Ridsdill Smith, and M. Willford. The London Millennium Footbridge. *The Structural Engineer*, 79(22):17–21, 2001.

[30] E. Caetano, A. Cunha, F. Magalhães, and C. Moutinho. Studies for controlling human-induced vibration of the Pedro e Inês footbridge, Portugal. Part 1: Assessment of dynamic behaviour. *Engineering Structures*, 32(4):1069–1081, 2010.

[31] M. Bocian, J.M.W. Brownjohn, V. Racic, D. Hester, A. Quattrone, L. Gilbert, and R. Beasley. Time-dependent spectral analysis of interactions within groups of walking pedestrians and vertical structural motion using wavelets. *Mechanical Systems and Signal Processing*, 105:502–523, 2018.

[32] S.I. Nakamura. Field measurements of lateral vibration on a pedestrian suspension bridge. *The Structural Engineer*, 81(22):22–26, 2003.

[33] C. Meinhardt, D. Newland, J. Talbot, and D. Taylor. Vibration performance of London's Millennium Footbridge. *24th International Congress on Sound and Vibration, London, UK, 23-27 July*, 2017.

[34] B. Josephson. Out of step on the bridge, 14th June, 2000. Letter to the Editor, The Guardian, UK.

[35] C. Barker. Some observations on the nature of the mechanism that drives the self-excited lateral response of footbridges. In *Proc. 1st Int. Conf. Design & Dynamic Behaviour of Footbridges, Paris, 20–22 November*, 2002.

[36] A.D. Pizzimenti and F. Ricciardelli. Experimental evaluation of the dynamic lateral loading of footbridges by walking pedestrians. *Proceedings of Eurodyn 2005 - 6th International Conference on Structural Dynamics, Paris, France*, 2005.

[37] F. Ricciardelli and A.D. Pizzimenti. Lateral walking-induced forces on footbridges. *Journal of Bridge Engineering*, 330(6):1265–1284, 2007.

[38] J.H.G. Macdonald. Lateral excitation of bridges by balancing pedestrians. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 465:1055–1073, 2009.

[39] M. Bocian, J.H.G. Macdonald, and J.F. Burn. Biomechanically inspired modelling of pedestrian-induced forces on laterally oscillating structures. *Journal of Sound and Vibration*, 331(16):3914–3929, 2012.

[40] E.T. Ingolfsson, C.T. Georgakis, F. Ricciardelli, and J. Jonsson. Experimental identification of pedestrian-induced forces on footbridges. *Journal of Sound and Vibration*,

330(6):1265–1284, 2011.

[41] M. Bocian, J.H.G. Macdonald, J.F. Burn, and D. Redmill. Experimental identification of the behaviour of and lateral forces from freely-walking pedestrians on laterally oscillating structures in a virtual reality environment. *Engineering Structures*, 105:62–76, 2015.

[42] M. Bocian, J.F. Burn, J.H.G. Macdonald, and J.M.W. Brownjohn. From phase drift to synchronisation - pedestrian stepping behaviour on laterally oscillating structures and consequences for dynamic stability. *Journal of Sound and Vibration*, 392:382–399, 2017.

[43] V. Joshi and M. Srinivasan. Walking crowds on a shaky surface: stable walkers discover Millennium Bridge oscillations with and without pedestrian synchrony. *Biology Letters*, 14(10):20180564, 2018.

[44] A. McRobie, G. Morgenthal, D. Abrams, and J. Prendergast. Parallels between wind and crowd loading of bridges. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 371(1993):20120430, 2013.

[45] M.A. Townsend. Biped gait stabilization via foot placement. *Journal of Biomechanics*, 18:21–38, 1985.

[46] S.P. Carroll, J.S. Owen, and M.F.M. Hussein. A coupled biomechanical/discrete element crowd model of crowd–bridge dynamic interaction and application to the Clifton Suspension Bridge. *Engineering Structures*, 49:58–75, 2013.

[47] A.L. Hof, R.M. van Bockel, T. Schoppen, and K. Postema. Control of lateral balance

in walking: experimental findings in normal subjects and above-knee amputees. *Gait & Posture*, 25(2):250–258, 2007.

[48] I.V. Belykh, R. Jeter, and V.N. Belykh. Bistable gaits and wobbling induced by pedestrian-bridge interactions. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 26(11):116314, 2016.

[49] Y. Kuramoto. Self-entrainment of a population of coupled non-linear oscillators. In H. Araki, editor, *International Symposium on Mathematical Problems in Theoretical Physics*, volume 39 of *Lecture Notes in Physics*. Springer-Verlag, 1975.

[50] A. McRobie. Long-term solutions of Macdonald's model for pedestrian-induced lateral forces. *Journal of Sound and Vibration*, 332:2846–2855, 2013.

[51] S.H. Strogatz. *Nonlinear Dynamics and Chaos*. CRC Press, New York, second edition, 2014.

[52] J. Ashmore, P. Avan, and W.E. Brownwell et al. The remarkable cochlear amplifier. *Hearing Research*, 266:1–17, 2010.

[53] D. Avitabile, M. Homer, A.R. Champneys, J.C. Jackson, and D. Robert. Mathematical modelling of the active hearing process in mosquitoes. *J. Royal Soc. Interface*, 7:105–122, 2010.

[54] J. Heathcote and F. Perri. Financial autarky and international business cycles. *Journal of Monetary Economics*, 49:601–627, 2003.

[55] S.P. Carroll, J.S. Owen, and M.F.M. Hussein. Experimental identification of the lateral human-structure interaction mechanism and assessment of the inverted-pendulum

biomechanical model. *Journal of Sound and Vibration*, 333:5865–5884, 2014.

[56] D. Claff, M.S. Williams, and A. Blakeborough. The kinematics and kinetics of pedestrians on a laterally swaying footbridge. *Journal of Sound and Vibration*, 407:286–308, 2017.

[57] A.L. Hof, S.M. Vermerris, and W.A. Gjaltema. Balance responses to lateral perturbations in human treadmill walking. *The Journal of Experimental Biology*, 213(15):2655–2664, 2010.

[58] A. Pachi and T. Ji. Frequency and velocity of people walking. *Structural Engineer*, 84(3):36–40, 2005.

[59] A.H. Nayfeh. *Perturbation Methods*. Wiley, Chichester, UK, 2000.

[60] B. Wolmuth and J. Surtees. Crowd-related failure of bridges. *Proceedings of ICE: Civil Engineering*, 156(paper no. 12844):116–123, 2003.

[61] R. Julavits. Strike coverage: Brooklyn Bridge not built for crowds, December 2005. The Village Voice.

[62] R. Woodward and T. Zoli. Two bridges built using black locust wood. Las Vegas, Nevada, USA, 2013. Proceedings of the International Conference on Timber Bridges.

[63] L.W. Foderaro. A new bridge bounces too far and is closed until the spring. New York Times, October 2014.

[64] A. Plitt. Brooklyn Bridge Park's Squibb Park Bridge reopens with slightly less bounce. Curbed NY, April 2017.

[65] H. Bachmann. Case studies of structures with man-induced vibrations. *Journal of*

Structural Engineering, 113:621–647, 1992.

[66] Y. Fujino, B.M. Pacheco, S.-I. Nakamura, and P. Warnitchai. Synchronization of human walking observed during lateral vibration of a congested pedestrian bridge. *Earthquake Engineering & Structural Dynamics*, 22(9):741–758, 1993.

[67] Sétra. Footbridges - assessment of vibrational behaviour of footbridges under pedestrian loading. *Sétra Technical Guide*, 2006.

[68] A. Rönnquist and E. Strømmen. Dynamic properties from full scale recordings and FE-modelling of a slender footbridge with flexible connections. *Structural Engineering International*, 4:421–426, 2008.

[69] S.-I. Nakamura. Model for lateral excitation of footbridges by synchronous walking. *Journal of Structural Engineering*, 130(1):32–37, 2004.

[70] H. Bachmann. "Lively" footbridges – a real challenge. *Footbridge – 1st International Conference, Paris, France*, 2002.

[71] W. Hoorpah, O. Flamand, and X. Cespedes. The Simone de Beauvoir Footbridge in Paris. Experimental verification of the dynamic behaviour under pedestrian loads and discussion of corrective modifications. *Footbridge 2008 – Third International Conference, Porto, Portugal*, 2008.

[72] S.M. Wilkinson and J. Knapton. Analysis and solution to human-induced lateral vibrations on a historic footbridge. *Journal of Bridge Engineering*, 11(1), 2006.

[73] M. Mistler and D. Heiland. Lock-in-effekt bei brücken infolge fussgängeranregung – schwingungstest der weltlängsten fussgänger- und velobrücke – dreiländerbrücke (Lock-

in effect due to pedestrian excitation – vibration test of the world's longest pedestrian and bicycle bridge "Dreiländerbrücke"). *D-A-CH Tagung 2007, Vienna, Austria*, 2007.

[74] L.W. Foderaro. Brooklyn walkway to reopen, with less bounce in your steps. New York Times, April 2017.

[75] C. Moutinho, S. Pereira, and Á. Cunha. Continuous dynamic monitoring of human-induced vibrations at the Luiz I Bridge. *Journal of Bridge Engineering*, 25(8):05020006, 2020.

[76] F. Danbon and G. Grillaud. Dynamic behaviour of a steel footbridge. Characterization and modelling of the dynamic loading induced by a moving crowd on the Solferino Footbridge in Paris. *Proceedings of Footbridge 2005*, 2005.

[77] M.M. Dupuit, M. de Contades, Ilougan, Roland, and Mahyer. Rapport de la commission d'enquéte (*) nommée par arrété de m. le préfet de maine-et-loire, en date du 20 avril 1850, pour rechercher les causes et les circinstances qui ont amene la chute du pont suspendu de la basse-chaine. *Annales des Ponts et Chaussés*, XX(237):394–411, 1850.

[78] Y. Fujino and B.M. Pacheco. Discussion of "Case studies of structures with man-induced vibrations" by H. Bachmann (J. Struct. Eng. 1992, 118(3) paper no.26567). *Journal of Structural Engineering*, 19:3682–3683, 1993.

[79] L. Sun and X. Yuan. Study on pedestrian-induced vibration of footbridge, 2008. Footbridge — 3rd International Conference, Porto, Portugal.

[80] A.N. Blekherman. Swaying of pedestrian bridges. *Journal of Bridge Engineering*,

10(2):142–150, 2005.

[81] A. Adão da Fonseca. Footbridges in portugal. *Proceedings of Footbridge 2008 - 3rd International Conference, Porto, Portugal, 2–4 July*, 2008.

[82] Hurriyet Daily News. Swaying causes running wariness over Bosphorus Bridge, 2010. 18 October 2010.

[83] BBC News. Swaying footbridge triggered deadly Cambodia stampede, 24th November 2010. `https://www.bbc.co.uk/news/world-asia-pacific-11827313` date accessed 17/07/2020.

[84] Kevin M Daley. Supplementary Videos for Doctoral Dissertation of Kevin Daley, July 2023. Available at `https://doi.org/10.5281/zenodo.8132826`.

[85] Dirk Helbing and Peter Molnar. Social force model for pedestrian dynamics. *Physical review E*, 51(5):4282, 1995.

[86] Elisa Bassoli and Loris Vincenzi. Parameter calibration of a social force model for the crowd-induced vibrations of footbridges. *Frontiers in Built Environment*, 7:656799, 2021.

[87] Luca Bruno and Fiammetta Venuti. Crowd–structure interaction in footbridges: modelling, application to a real case-study and sensitivity analyses. *Journal of Sound and Vibration*, 323(1-2):475–493, 2009.

[88] Gabriel Hine, Jeremiah Onaolapo, Emiliano De Cristofaro, Nicolas Kourtellis, Ilias Leontiadis, Riginos Samaras, Gianluca Stringhini, and Jeremy Blackburn. Kek, cucks, and god emperor trump: A measurement study of 4chan's politically incorrect forum

and its effects on the web. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 11, pages 92–101, 2017.

[89] Kevin Rose. What is qanon, the viral pro-trump conspiracy theory? *The New York Times*, Sep 2021.

[90] Adam J. Kucharski. Modelling the transmission dynamics of online social contagion, 2016.

[91] Jonathan Skaza and Brian Blais. Modeling the infectiousness of twitter hashtags. *Physica A: Statistical Mechanics and its Applications*, 465:289–296, 2017.

[92] Chunxiao Jiang, Yan Chen, and K. J. Ray Liu. Evolutionary dynamics of information diffusion over social networks. *IEEE Transactions on Signal Processing*, 62(17):4573–4586, 2014.

[93] Xuanyu Cao, Yan Chen, Chunxiao Jiang, and KJ Ray Liu. Evolutionary information diffusion over heterogeneous social networks. *IEEE Transactions on Signal and Information Processing over Networks*, 2(4):595–610, 2016.

[94] Zi-Ke Zhang, Chuang Liu, Xiu-Xiu Zhan, Xin Lu, Chu-Xu Zhang, and Yi-Cheng Zhang. Dynamics of information diffusion and its applications on complex networks. *Physics Reports*, 651:1–34, 2016. Dynamics of information diffusion and its applications on complex networks.

[95] Peter Schuster and Karl Sigmund. Replicator dynamics. *Journal of Theoretical Biology*, 100(3):533–538, 1983.

[96] Jörgen W Weibull. *Evolutionary game theory*. MIT press, 1997.

[97] Daniel Jolley, Rose Meleady, and Karen M. Douglas. Exposure to intergroup conspiracy theories promotes prejudice which spreads across groups. *British Journal of Psychology*, 111(1):17–35, 2020.

[98] Sander Van der Linden. The conspiracy-effect: Exposure to conspiracy theories (about global warming) decreases pro-social behavior and science acceptance. *Personality and Individual Differences*, 87:171–173, 2015.

[99] Loukas Balafoutas, Alexander Libman, Vasileios Selamis, and Björn Vollan. Exposure to conspiracy theories in the lab. *Economic and Political Studies*, 9(1):90–112, 2021.

[100] Matteo Cinelli, Walter Quattrociocchi, Alessandro Galeazzi, Carlo Michele Valensise, Emanuele Brugnoli, Ana Lucia Schmidt, Paola Zola, Fabiana Zollo, and Antonio Scala. The covid-19 social media infodemic. *Scientific reports*, 10(1):1–10, 2020.

[101] Mingfei Guo, Xiuying Chen, Juntao Li, Dongyan Zhao, and Rui Yan. How does truth evolve into fake news? an empirical study of fake news evolution. In *Companion Proceedings of the Web Conference 2021*, pages 407–411, 2021.

[102] Jason Hindes, Michael Assaf, and Ira B. Schwartz. Outbreak size distribution in stochastic epidemic models. *Phys. Rev. Lett.*, 128:078301, Feb 2022.

[103] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. In *5th Workshop on Energy Efficient Machine Learning and Cognitive Computing @ NeurIPS 2019*, 2019.

[104] Angela Nagle. *Kill all normies: Online culture wars from 4chan and Tumblr to Trump and the alt-right.* John Hunt Publishing, 2017.

[105] Christian Siefkes, Fidelis Assis, Shalendra Chhabra, and William S Yerazunis. Combining winnow and orthogonal sparse bigrams for incremental spam filtering. In *Knowledge Discovery in Databases: PKDD 2004: 8th European Conference on Principles and Practice of Knowledge Discovery in Databases, Pisa, Italy, September 20-24, 2004. Proceedings 8*, pages 410–421. Springer, 2004.

[106] J.J. Collins and I.N. Stewart. Coupled nonlinear oscillators and the symmetries of animal gaits. *Journal of Nonlinear Science*, 3(1):349–392, 1993.

[107] Frank DeStefano. Vaccines and autism: evidence does not support a causal association. *Clinical Pharmacology & Therapeutics*, 82(6):756–759, 2007.

[108] Jonathan E Fielding. Managing public health risks: the swine flu immunization program revisited. *American Journal of Law & Medicine*, 4(1):35–43, 1978.

[109] Thomas Schreiber. Measuring information transfer. *Physical review letters*, 85(2):461, 2000.

[110] Roy M Anderson and Robert M May. *Infectious diseases of humans: dynamics and control*. Oxford university press, 1991.

[111] Andreas Rößler. Runge–kutta methods for the strong approximation of solutions of stochastic differential equations. *SIAM Journal on Numerical Analysis*, 48(3):922–952, 2010.

[112] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[113] Leland McInnes, John Healy, Nathaniel Saul, and Lukas Großberger. Umap: Uniform manifold approximation and projection. *Journal of Open Source Software*, 3(29):861, 2018.

[114] Richard S Sutton. Learning to predict by the methods of temporal differences. *Machine learning*, 3:9–44, 1988.

[115] Edward Loper and Steven Bird. Nltk: The natural language toolkit. *arXiv preprint cs/0205028*, 2002.

[116] Hyunjin Choi, Judong Kim, Seongho Joe, and Youngjune Gwon. Evaluation of bert and albert sentence embedding performance on downstream nlp tasks. In *2020 25th International conference on pattern recognition (ICPR)*, pages 5482–5487. IEEE, 2021.

[117] Zhengjie Gao, Ao Feng, Xinyu Song, and Xi Wu. Target-dependent sentiment classification with bert. *Ieee Access*, 7:154290–154299, 2019.

[118] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[119] Antonis Papasavva, Savvas Zannettou, Emiliano De Cristofaro, Gianluca Stringhini, and Jeremy Blackburn. Dataset: Raiders of the Lost Kek: 3.5 Years of Augmented 4chan Posts from the Politically Incorrect Board, January 2020.

[120] Katharine Sanderson. Gpt-4 is here: what scientists think. *Nature*, 615(7954):773, 2023.

[121] Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hess-

low, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, et al. Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*, 2022.

[122] Margaret Ryznar. Exams in the time of chatgpt. *Washington and Lee Law Review Online*, 80(5):305, 2023.

[123] Deep Ganguli, Liane Lovitt, Jackson Kernion, Amanda Askell, Yuntao Bai, Saurav Kadavath, Ben Mann, Ethan Perez, Nicholas Schiefer, Kamal Ndousse, et al. Red teaming language models to reduce harms: Methods, scaling behaviors, and lessons learned. *arXiv preprint arXiv:2209.07858*, 2022.

[124] Andrey Kurenkov. Lessons from the gpt-4chan controversy. *The Gradient*, 2022.

[125] Yuqing Du, Olivia Watkins, Zihan Wang, Cédric Colas, Trevor Darrell, Pieter Abbeel, Abhishek Gupta, and Jacob Andreas. Guiding pretraining in reinforcement learning with large language models. *arXiv preprint arXiv:2302.06692*, 2023.