# Modeling synchronization utilizing reinforcement learning in *Photuris frontalis* fireflies

Nolan R. Bonnie
*Department of Computer Science*
*BioFrontiers Institute*
*University of Colorado Boulder*
nolan.bonnie@colorado.edu

Anna Rahn
*Department of Computer Science*
*University of Colorado Boulder*
anra8571@colorado.edu

*Abstract*—**Photuris frontalis is a firefly species known to exhibit highly synchronous bioluminescent tendencies in male swarms. This phenomenon has been observed extensively in literature, but oftentimes simple synchronization models can fail to capture the complexity of synchronous fireflies. In this paper we examine the performance of a traditional oscillatory model (Kuramoto) and a more modern deep Q-learning algorithm. We trained both models to synchronize with the *P. frontalis* swarm data, and found that both achieved high levels of accuracy. Moving forward, we recommend these experiments be repeated on different firefly species that are less synchronous and periodic.**

## I. INTRODUCTION

Modeling synchronous swarms of fireflies is an important challenge in computational ecology that is typically calculated through the use of a synchronization model, like the Kuramoto model. However, this process is not always effective given the substantial variability and noise in the field data of synchronous fireflies. There has never been an attempt at trying to train a reinforcement learning algorithm to describe the synchrony. Moreover, synchronized phenomena have rarely been modeled with reinforcement learning. This project aims to understand the ability for reinforcement learning to operate in a synchronous environment, and compare the produced model to standard synchronization models in terms of accuracy and convergence rate.

## II. BACKGROUND

### A. Previous Work

Certain species of fireflies have been found to synchronize their flashes within swarms, and the temporal nature of this phenomenon has been studied extensively [1], [2]. One firefly species that is particularly synchronous is *Photuris frontalis*, where firefly swarms have been observed to be very periodic with simple flashing patterns [1], [3]. Every firefly species has a different flashing pattern, and for a while scientists have attempted to record species flash patterns through field observations [3], [4]. However, with the advancement of imaging techniques, data collection methods, and computational power, there has been a shift in the field towards developing computational models to describe firefly flash patterns [2], [5], [6]. Because we are studying a living system *P. frontalis* is not perfectly synchronous, meaning that quantifying and modeling synchrony can be difficult. Currently, tools like the Kuramoto model are applied towards synchronous firefly data to describe the system, however due to the variability found in the data these models often fail to a produce accurate representations [6]. This is mainly due to the fact that the Kuramoto model is calculated off different oscillatory periods, and better represents theoretical rather than living synchrony [7]. In order to develop accurate computational models of complex synchronous living systems, we need a better way to quantify and understand the imperfect synchronization that occurs. Reinforcement learning (RL) has never been applied to field data of any firefly species as a means of describing their synchronous flashing behavior. Due to the ability of RL to find optimal policies, it is worth exploring its application to our data.

### B. Contributions

Our lab has collected temporal flash data of *P. frontalis* swarms from Congaree National Park. Understanding communication mechanisms found in nature is an important challenge which could greatly benefit from modern computational tools [8], [9]. According to the International Union for Conservation of Nature, one in three of all Northern American firefly species are at risk for

extinction, and half of the species found in the region are at an unknown threat status due to insufficient data [10]. By running computational simulations of firefly systems, we can accurately assess the impact of certain environmental threats. However, there are several factors preventing an accurate firefly simulation. One of these challenges is developing a synchronization model that accurately replicates the synchronous flash patterns that we observe in nature. The best implementation at the moment is the Kuramoto model, which utilizes mathematical theories of coupled oscillators. The Kuramoto model was never intended to describe living systems, and oftentimes fails to accurately model synchronous fireflies where high amounts of variability exists. It is not entirely understood why fireflies synchronize, or what environmental factors impact individual fireflies to make them not flash with the swarm. However, it is clear that a model is needed that can adapt to the imperfect data recorded from synchronous firefly swarms.

For the purposes of this experiment, we wish to find a computational model that can learn the correct flashing policy and synchronize to the experimental data we captured in the field. We believe this scenario is a strong application for reinforcement learning, since the data provides a very clear environment and response function that will help our reinforcement learning agent to converge to an optimal policy.

## III. METHODOLOGY

### A. Data

The data used in our computational experiments were collected by the Peleg Lab in Congaree National Park during the summer of 2020. Congaree is commonly known for its large populations of *Photuris frontalis* fireflies. *P. frontalis* is a highly synchronous firefly species, meaning that compared to other synchronous fireflies like *Photinus carolinus* or *Photinus knulli*, *P. frontalis* swarm flashes are precisely spaced apart, with less background noise. Data was collected over several hours at a rate of 30 frames per second, and subsequent post-processing was done to determine how many fireflies are flashing in each individual frame.

When viewing the data across several seconds, the periodic and synchronous nature of *P. frontalis* becomes clear. In order to reduce background noise of the system, all frames where 1 or 2 fireflies are flashing were considered to be frames where 0 fireflies are flashing. The full extent of individual firefly flashes is not completely understood, and seemingly random asynchronous flashes are not uncommon. Since we aim to train a reinforcement
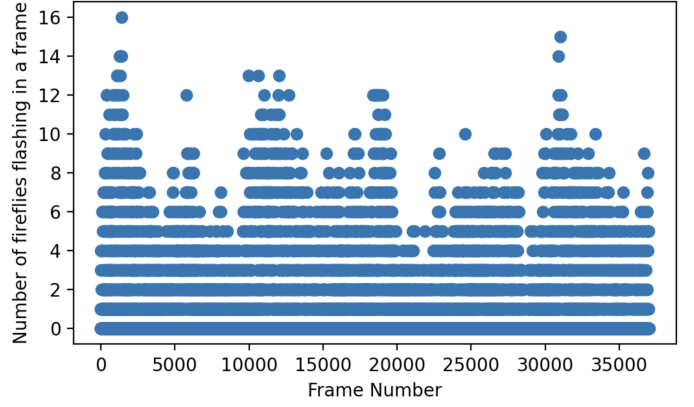


Fig. 1. *P. frontalis* flash data across several hours. Every data point represents how many fireflies were flashing in the video frame. A value of 12 implies that in that specific frame, there were 12 fireflies in bioluminescence.
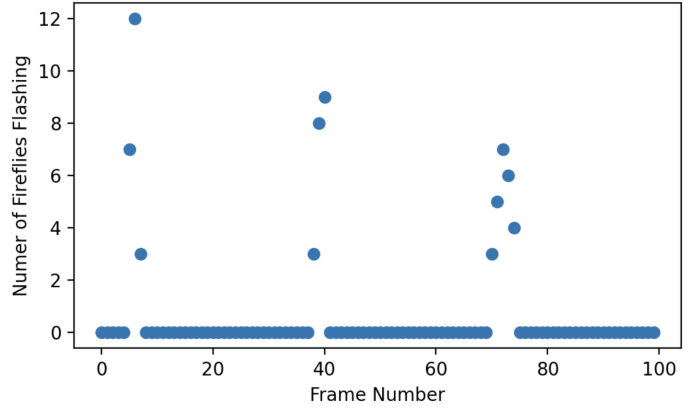


Fig. 2. A zoomed in plot showing the first 100 frames in our cleaned data set which omits frames where only 1 or 2 fireflies were flashing.

learning model that can learn a synchronous policy, removing these outlier data points should have a minimal impact on our ability to answer this question.

### B. Validation Data & Model Accuracy

A reinforcement learning model requires a reward function to inform the agent on how successful a particular action was given a state. In context, our reward function needs to be able to evaluate the agent's choice to either flash or not flash given the current and past N frames, where each frame describes how many fireflies were flashing in that given frame. In order to evaluate accuracy, we need a way to determine when the model should be flashing, and compare that with when the model actually chooses to flash. In order to calculate when a new firefly objectively should flash in order to synchronize with the swarm, we employed an algorithm

to approximate all the local maxima of the firefly flashing data. Given the extremely synchronous and periodic nature of *P. frontalis*, a moving window was used to locate all local maxima.
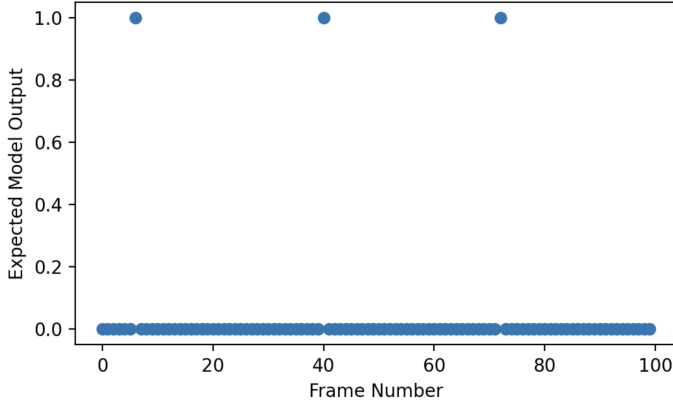


Fig. 3. Similar to Figure 2, this plot shows the first 100 frames. Each point represents the expected value our models should produce in order to flash in sync with the *P. frontalis* swarm.

With the local maxima calculated, assessing accuracy is a trivial task. We consider perfect accuracy to be when the model predicted flashes align perfectly with the "true" flash values. Any amount of phase deviation from true value linearly increases the penalty the model receives. This same process can also be applied to the whole data set as well. Normalizing the minimum error to be 1, and the maximum error to be 0, we can effectively calculate model accuracy.

### C. Kuramoto Model

The Kuramoto model, developed by Yoshiki Kuramoto, is a model used to describe synchronization when given a relatively large network of coupled oscillators. Originally developed to model chemical and biological systems, such as semiconductor laser arrays [12] and cardiac pacemaker cells [13], we have now found it to be applicable to areas such as neural synchronization, yeast cell metabolism [14], Josephson junctions [15], a physical system, and firefly flashes.

Each oscillator begins with a randomly distributed starting position and intrinsic frequency and is adjacent to every other oscillator in the system. The intrinsic frequencies are randomly distributed according to the probability density function g($\omega$), which is usually a gaussian-like symmetric function [17]. As training progresses, the model slows down and speeds up individuals

until all oscillators move in sync. The most common form, and the one we are using here, is as follows:

$$\Theta_i = \omega_i + \frac{K}{N}\Sigma_{j=1}^{N}\sin(\Theta_j - \Theta_i) \quad i = 1, ..., N$$

Here $\Theta_i$ is the phase of each oscillator, $\omega_i$ is the natural frequency of each oscillator (so the frequency it would run at with no outside influence), K is the coupling constant (which dictates the strength of interactions), and N is the number of oscillators. Pairs of oscillators are coupled according to the sine of their phase differences, and K must be set above some threshold in order to see good synchronization.

The degree of synchronization success can be determined by the complex-valued order parameter, which is essentially the average position of the system. The value is calculated as follows:

$$Z = \frac{1}{N}\Sigma_{j=1}^{N}\exp(i\Theta_j)$$

where $\exp(i\Theta_j)$ is the state of the jth oscillator. Z ranges from zero to one; the closer to one, the more synchronized the state is.

There are only a handful of results this model can produce. If the coupling is too weak or the training period too short, synchronization may not occur at all. In some cases, the oscillators may partially synchronize; most of the time, this means that two groups form, each moving at the same frequency, but their position a half-period off from one another. Here, the interaction strength between oscillators is usually too weak to bring the system to full synchrony. Less commonly, three of more groups may form. In all partially synchronous cases, the order parameter will plateau at some fraction between zero and one. Ideally (in our case), the model will fully synchronize. If this happens, the order parameter should be very close to one and the oscillators will be moving together in a single group.

To adapt this model to our purposes, we set ten oscillators (that each represent individual fireflies in the environment) to natural frequencies sampled from the data. Each begins at a random position as normal. We then create an all-to-all connectivity graph and pass in the graph, K=3, dt=0.01, and the frequencies as parameters to the model. After synchronization, we can reconstruct the collective behavior by defining a flash as every local maxima. By taking the median of all ten individuals after synchrony has occurred, we can determine when the Kuramoto swarm as a whole is flashing and compare this to the expected results as described in the Model Accuracy section. This whole process is then repeated

for varying numbers of oscillators. Though in many cases the Kuramoto model fails to represent the variance in biological systems, we anticipate in our experiment that it will synchronize fairly well because the oscillators (just like Photuris frontalis fireflies) all start with similar intrinsic frequencies.

Our model also aligns with each assumption made by Kuramoto theory - our oscillators are identical, the order parameter and synchronization frequency as both constants in time [16], and interaction strength between two individuals depends on the sinusoidal phase difference.

### D. Reinforcement Learning Model

For the reinforcement learning model, we chose to follow the deep Q-network architecture introduced by DeepMind in 2013. This operates in a very similar way to tabular Q-learning methods, except instead of explicitly calculating and storing every $Q(S,a)$ value for all possible states ($S$) and actions ($a$), we approximate the $Q(S,a)$ values utilizing a deep neural network. The architecture we choose utilizes three hidden dense layers, all with rectified linear unit (ReLU) activation functions. The deep neural network (DNN) architecture was chosen through experimentation.
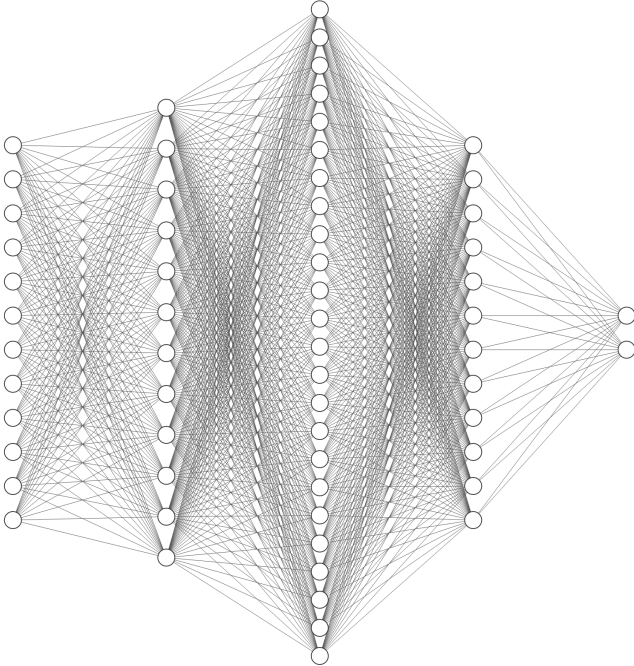


Fig. 4. Fully connected neural network visualization of our DNN architecture.

The deep Q-network (DQN) works by taking in the current state as an input, which for our data set is a 12-element array slice, where each element is the number

of fireflies flashing in the given frame. These 12 frames are all chronologically consecutive. Multiple of these 12-frame slices are randomly aggregated into a batch, which is then passed through the DQN. For each slice in the batch, the DQN will approximate the appropriate $Q(S,a)$ value for each possible action (flash, not flash), and will greedily pick the highest Q value. This action will then be checked against the expected output (as described in III.B.), and the DQN will receive appropriate feedback on its performance. After the batch has completed, the DQN will feed-forward updates to the node weights. This process will repeat for a maximum of 40 epochs, or until the model loss is small enough. Within batches, slices are randomly pulled from the training data pool in order to prevent any unwanted correlation in training. Each batch includes ten 12-frame slices which are passed through the DQN before weights are updated. Every epoch contains 200 batches. Model epochs are repeated until model loss falls below a small threshold.

## IV. RESULTS

### A. Kuramoto Performance

We see in Figure 5 that all the oscillators start out in a random position, but by the end of the training period, converge to a regular sinusoidal function.
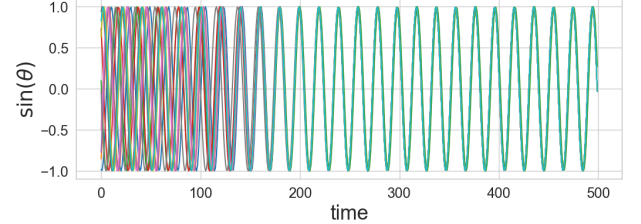


Fig. 5. Kuramoto model graph of nodes vs. time. Each line represents one oscillator's position over the 500 second training period

By plotting the order parameter (Figure 6), we can confirm the convergence. Z increases to 1 at about 150 seconds, which is at the same time we can visually see that the oscillators converge.

We then compare the reconstructed Kuramoto swarm's flashing behavior after synchrony to the expected data and find that it has an average accuracy of 96.67%. The accuracy represents how well the Kuramoto model synchronizes with the test data with 100% meaning that all flashes perfectly align. Both our 25-oscillator and 40-oscillator models have an average accuracy of 97.23%.

### B. Reinforcement Learning Performance

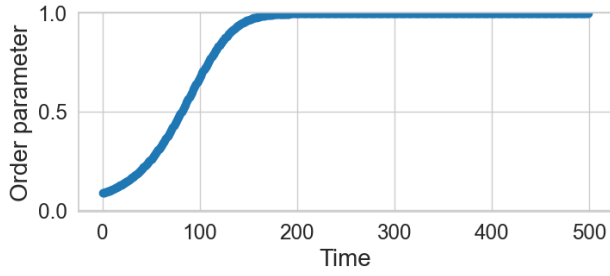Since there is some random selection in terms of which slices are formed into a batch for the DQN to

Fig. 6. Order parameter of Kuramoto model over 500 second training period

train on, there is of course no single statistically robust metric to quantify the model accuracy. However, in every example we ran, the DQN was able to converge in less than 40 epochs, with an accuracy from 96% to 98%. An accuracy of 96% implies that when the model was run on our testing data, the reinforcement learning agent was usually able to correctly flash in order to synchronize with the swarm. The same model was trained with different random seeds over twenty times, and in every occurrence, the model consistently achieved accuracy of over 96%. Figure 7 shows the loss function for one of our DQN training runs. As you can see, as the epochs progress, the model loss quickly decreases to small values, which implies high model training accuracy. 30% of the entire data set was set aside for testing the model accuracy, and model performance on the testing data is what we reported as model accuracy.
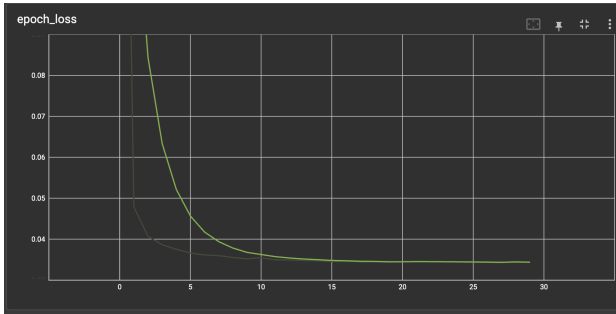


Fig. 7. Loss graph over epoch for our DQN.

## V. CONCLUSION

### A. Discussion

As predicted, the Kuramoto model performed fairly well. The oscillators, which were set to very similar natural frequencies, did not need to deviate much from their intrinsic behaviour to synchronize. We also saw small accuracy increases by using more oscillators, which is

likely a result of the model improving after having sampled from more data. Due to the natural variance in biological systems, we will always expect to see some amount of error, but found that the Kuramoto model was able to synchronize to *P. frontalis* flashes very well.
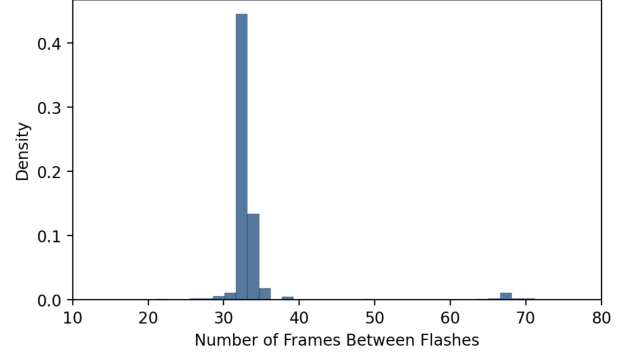


Fig. 8. Density of the number of frames in between flashes in *P. frontalis* swarm data.

As discussed earlier, *P. frontalis* has a highly synchronous and periodic flash pattern, which is an attribute unique to this specific firefly species. We credit a lot of success with the Kuramoto model to this. Within *P. frontalis* swarms, there is not a lot of variation within the time between flashes, which can be seen in Figure 8. Oscillatory motion is exactly what the Kuramoto model was developed to describe, so the success we see in our experiments is not surprising. We would expect these results to dramatically change if we introduced the Kuramoto model in a different firefly species environment, like *P. carolinus* or *P. knulli*, whose flash patterns are much more complex and asynchronous.

Given the synchronous nature of the data, it is not surprising that our DQN was able to discover an optimal policy. A surface level review of both models would suggest that both are equally capable of modeling firefly synchronization. However, we believe that the DQN was not given the right environment to demonstrate its true potential. We originally picked *P. frontalis* as our model organism because we believed that the reinforcement learning agent would need a simple problem in order to demonstrate synchronous properties. However, after seeing the successful results of this experiment, it's clear that we should introduce this same RL agent to a much more complex environment. It was discovered that reinforcement learning could learn to synchronize quite effectively, which was not known before these experiments. Going forward, we believe that more experimentation needs to be done on other firefly species

in order to fully evaluate the potential of reinforcement learning over the standard Kuramoto model.

*B. Limitations*

One possible limitation is that the data we are using to train and test these was collected on a relatively small swarm of *P. frontalis* fireflies. This limits the scope of our results, as we can't be certain these models would generalize well if we applied them to data of a larger *P. frontalis* swarm. Intuitively though, by having data from a smaller swarm of fireflies, we are reducing the variance and noise. Moreover, the data was collected across a few hours of a single night in Congaree National Park. Firefly flashing patterns are known to vary depending on environmental conditions like heat, humidity, and light pollution. It's entirely possible that *P. frontalis* firefly swarms across different days and locations would produce slightly different flashing patterns, which means that the models we trained for this experiment will likely not generalize to all *P. frontalis* swarms. In order to solve that issue, more data would need to be collected of *P. frontalis* flashing patterns from different geological locations and across different environmental conditions.

Another possible limitation is that single-estimator Q-learning is known to overestimate action values [11], an error that propagates to single-estimator DQN models. Given more time, we might also experiment with double DQN methods in order to reduce or eliminate the positive bias, which would be expected to improve the stability and convergence times of our model.

### REFERENCES

[1] Lewis, Sara M. "Silent Sparks." Silent Sparks. Princeton University Press, 2016.

[2] Sarfati, Raphaël, Julie C. Hayes, and Orit Peleg. "Self-organization in natural swarms of Photinus carolinus synchronous fireflies." Science Advances 7.28 (2021).

[3] Stanger-Hall, Kathrin F., and James E. Lloyd. "Flash signal evolution in Photinus fireflies: character displacement and signal exploitation in a visual communication system." Evolution 69.3 (2015): 666-682.

[4] Lewis, Sara M., Christopher K. Cratsley, and Kristian Demary. "Mate recognition and choice in Photinus fireflies." Annales Zoologici Fennici. Finnish Zoological and Botanical Publishing Board, 2004.

[5] Sarfati, Raphaël, et al. "Spatio-temporal reconstruction of emergent flash synchronization in firefly swarms via stereoscopic 360-degree cameras." Journal of The Royal Society Interface 17.170 (2020).

[6] Sarfati, Raphaël, et al. "Statistical analysis reveals the onset of synchrony in sparse swarms of Photinus knulli fireflies." Journal of the Royal Society Interface 19.188 (2022).

[7] Acebrón, Juan A., et al. "The Kuramoto model: A simple paradigm for synchronization phenomena." Reviews of modern physics 77.1 (2005): 137.

[8] G. De Croon et al., "Insect-inspired AI for autonomous robots." Science Robotics 7.67 (2022).

[9] T. Li et al., "Applications of Multi-Agent Reinforcement Learning in Future Internet: A Comprehensive Survey" IEEE Communications 24.2 (2022).

[10] Candace E. Fallon et al., "Evaluating firefly extinction risk: Initial red list assessments for North America." PLoS ONE 16.11 (2021).

[11] Hasselt, H., "Double Q-learning". Advances in neural information processing systems, 23 (2010).

[12] Kozyreff G, Vladimirov A G and Mandel P 2000 Phys. Rev. Lett. 85 3809

[13] Winfree A T 1980 The Geometry of Biological Times (New York: Springer)

[14] Bier M, Bakker B M and Westerhoff H V 2000 Biophys. J. 78 1087

[15] Wiesenfeld K, Colet P and Strogatz S H 1996 Phys. Rev. Lett. 76 404

[16] J D da Fonseca and C V Abud J. Stat. Mech. (2018) 103204

[17] Damicelli, Fabrizio, "Python Implementation of the Kuramoto model". Github (2019).