Police Patrol Optimization With Geospatial Deep Reinforcement Learning

Presenter: Daniel Wilson

Other Contributors:

Orhun Aydin

Omar Maher

Mansour Raad

Before we begin…

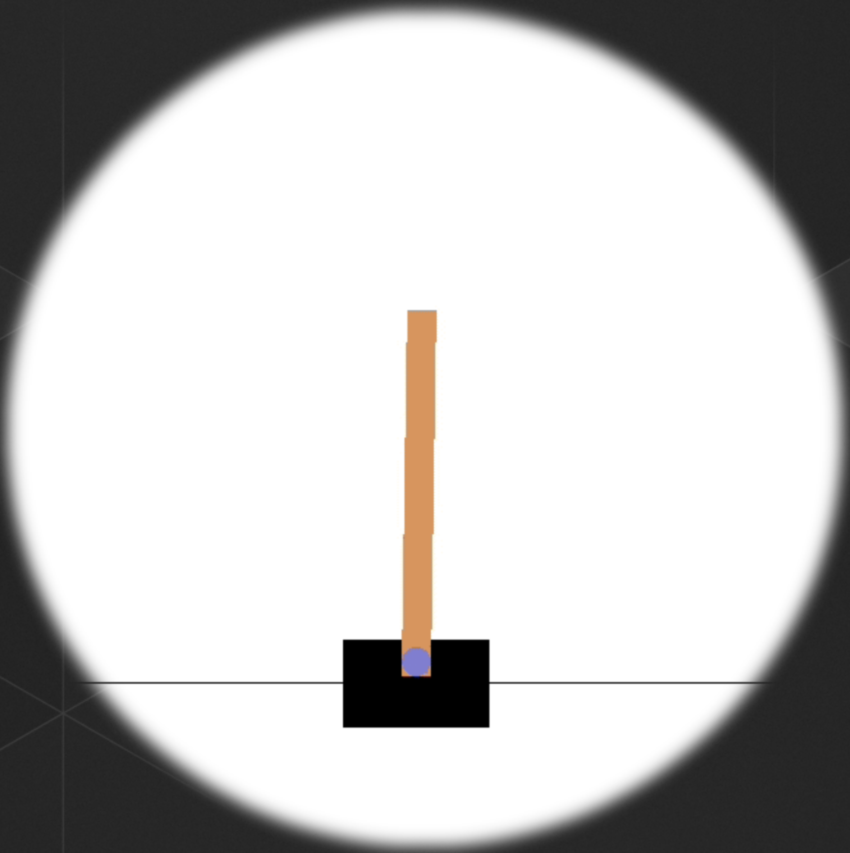
I am <u>not</u> a criminologist! We are working with a police department to help solve their resource allocation challenges.

This will never supersede human expertise, but rather inform decision makers.

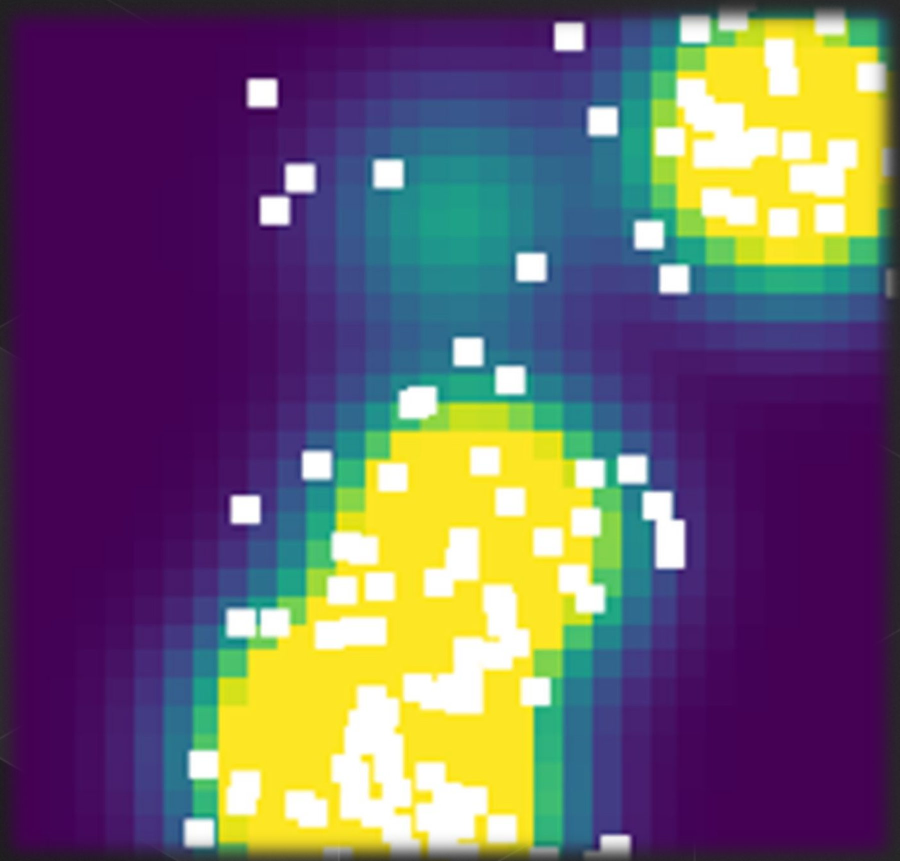
This is not Skynet; but we think it is pretty cool

We want your feedback, ask questions!

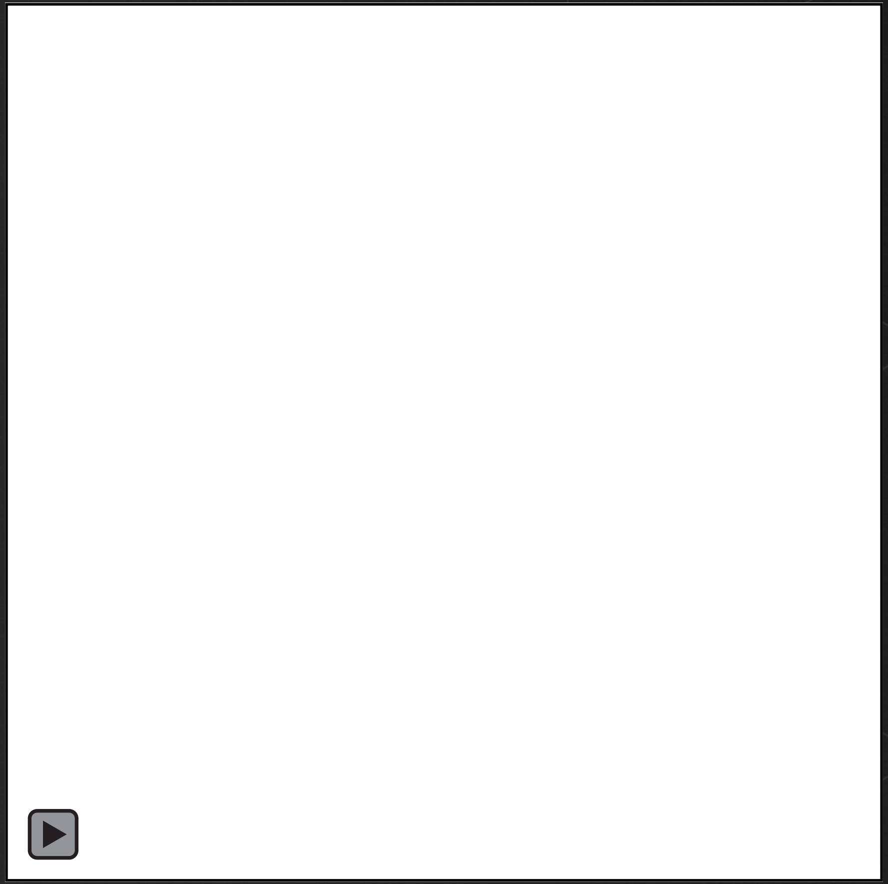
CartPole – the “Hello World” of Reinforcement Learning



Our first “CartPole”

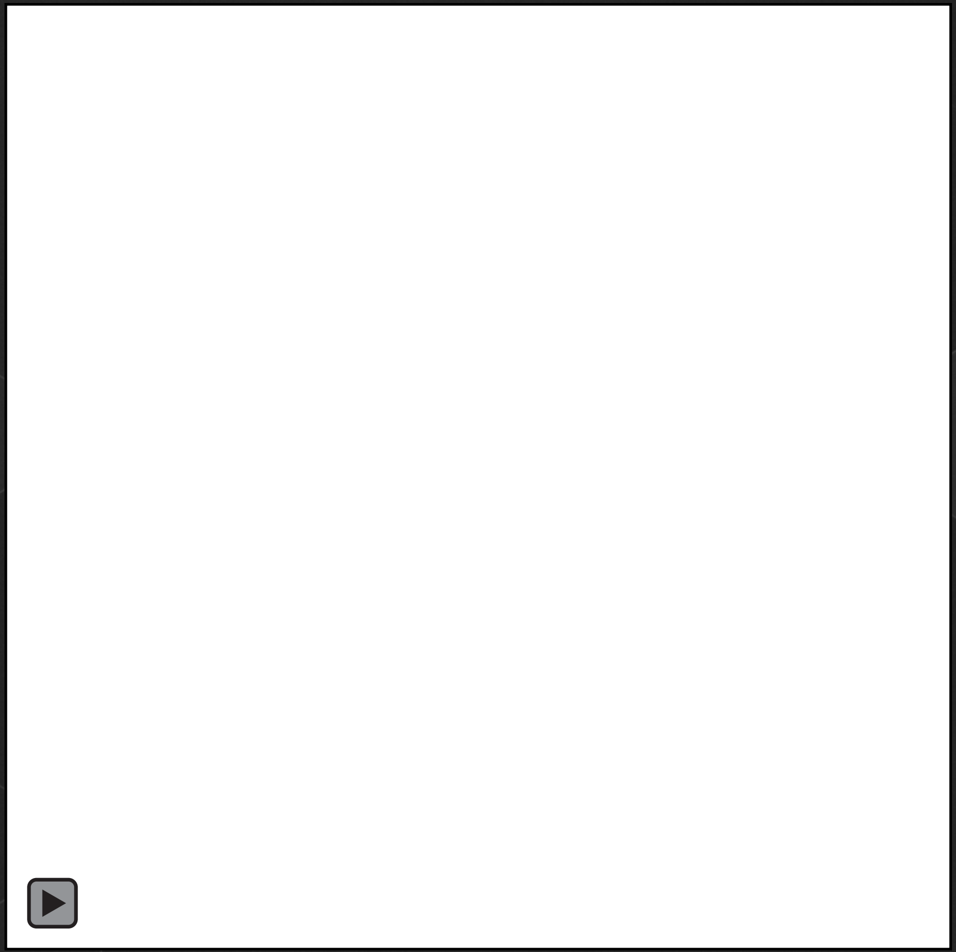


Inhomogeneous Poisson Process
to simulate toy crime hotspots

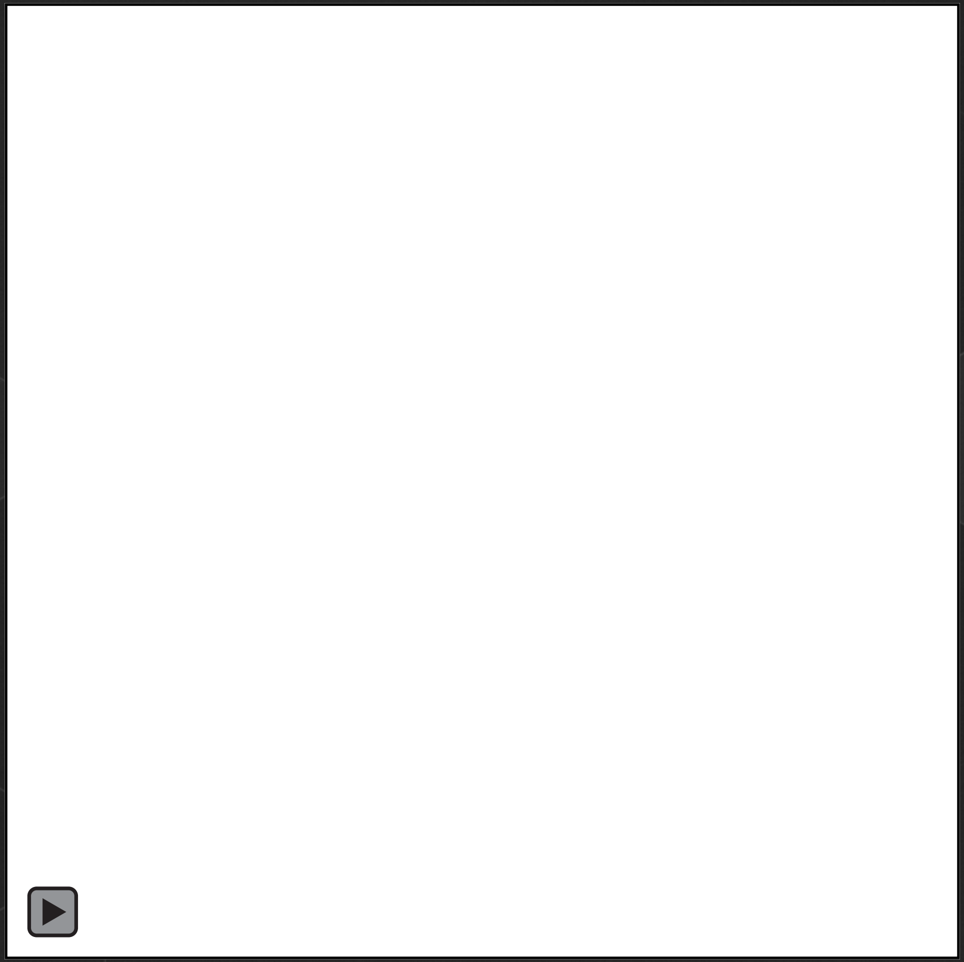


Proof of concept: Simple agent learns to
approximate spatial distribution from discrete
observations

Baby steps…

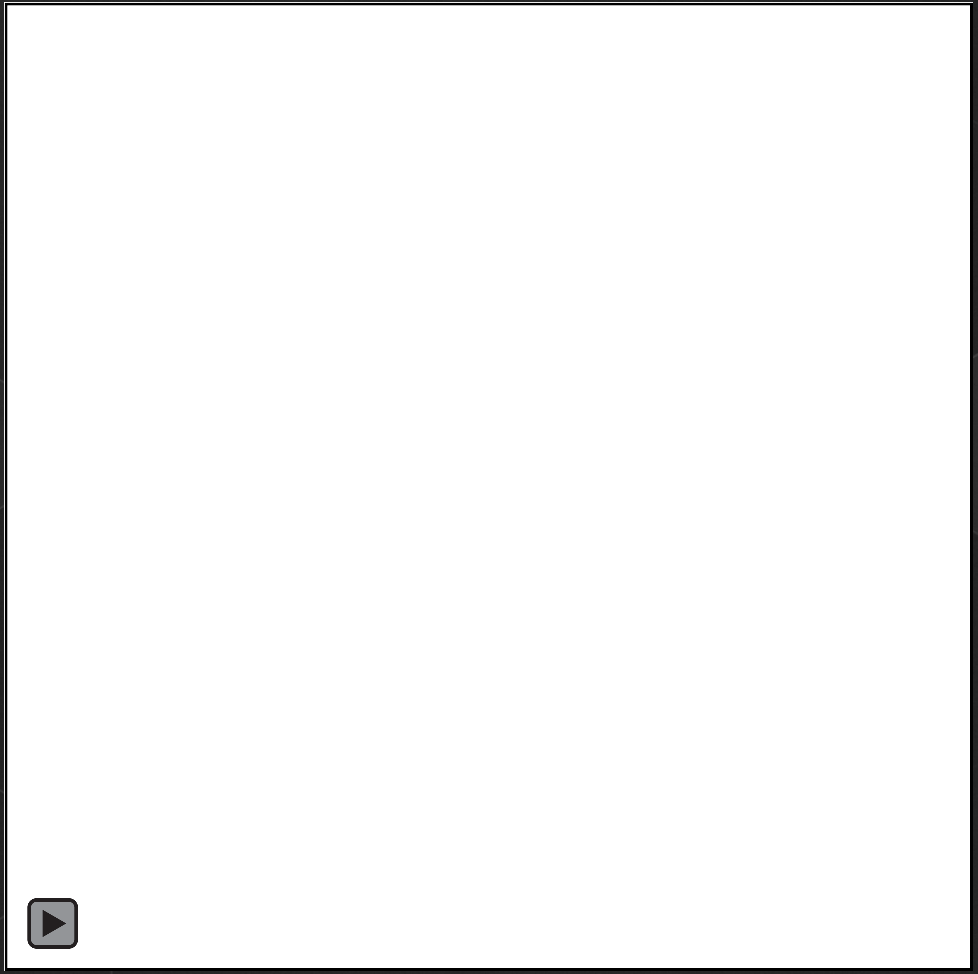


Lessons Learned



Discount too small…

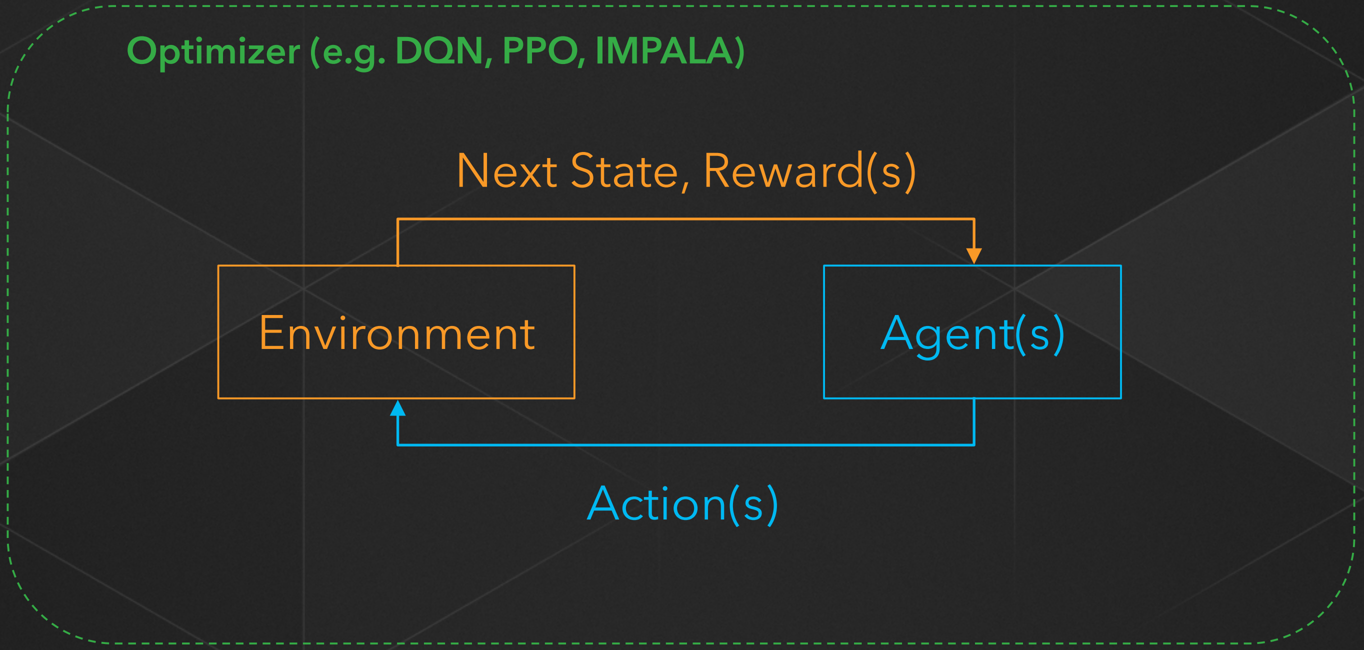
Lessons Learned



Poor state space representation – can’t learn individual actions

Where we are today.

Let's back up… what is Reinforcement Learning?



What can it do?

Google DeepMind DQN Playing Atari

OpenAI Five playing DOTA2

Google DeepMind AlphaGo vs Lee Sedol

Google DeepMind AlphaStar
playing Starcraft 2

Police Patrol Allocation

How to cast this as reinforcement learning?

We need to define an environment:

State

Actions

Reward

Reinforcement learning is sample inefficient - focus on modeling

Real world actions are complex – simplify, but don’t make it trivial

Many tradeoffs – sensible reward shaping to control strategies

The All Important GIS

- Police patrols act in a city, subject to all the constraints of a city
- Agent must learn to act in a simulated city environment to be applicable
- Crime/calls simulated from past data
- Crime deterrence modeled through spatial statistics

State

A lot of data to consider; agent needs a compact state representation.

For every time step (one minute)

Patrol location, state, action, availability

Crime location, type, age

Call location, type, age, status

Patrol-crime distance

Patrol-call distance

Crime/call statistics

… more

Our agent processes all of these features to determine optimal actions

Actions

Police patrols deter crime, but police precincts have limited resources.

Focus on simple actions to deter crime:

Patroling

Loitering

Responding to Calls

Our agent learns high level strategies from these low level actions

Reward

The goal is complex, and there are trade offs:

Minimize crime: penalty for each crime

Minimize call response time: penalty for every minute call unaddressed

Maximize security/safety: penalty every time security status for patrol area drops

Maximize traffic safety: penalty for every minute patrols use siren

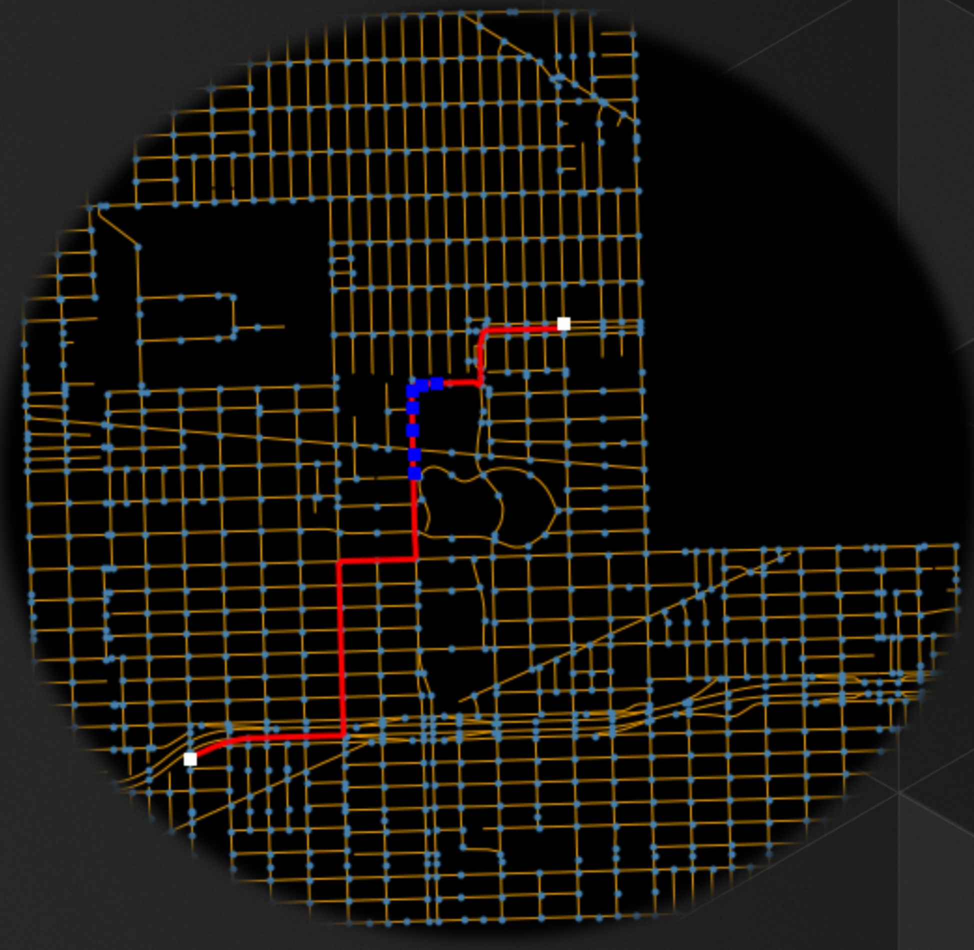
We can see different behaviors and strategies emerge based on reward shaping – more on this later!

Modeling the Environment

- We can't model everything, but we can learn strategies for what we can:
  - Model patrol paths/arrival times using graph/network analysis
  - Model security level with survival analysis
  - Model calls/crimes using spatial point processes
  - Model call resolution times using distribution statistics

Patrol Routing

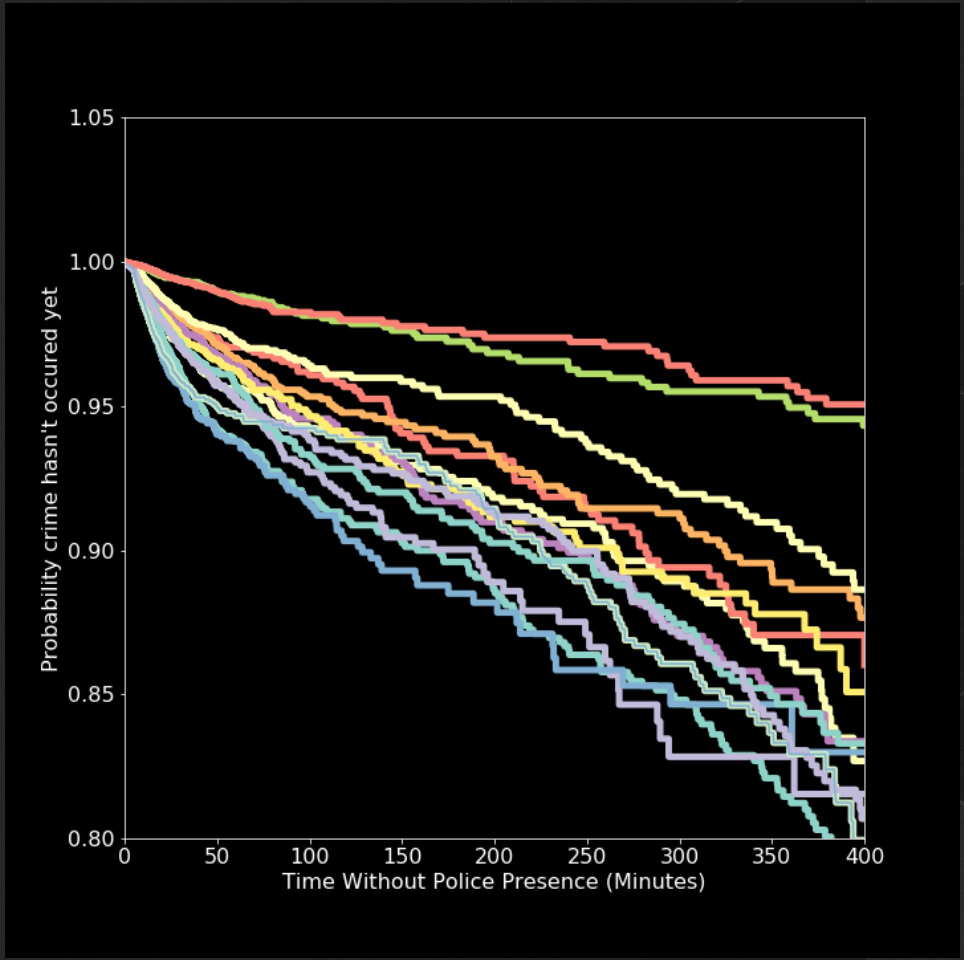
- Use actual road network for the police district
- Movement of patrols constrained by the road and speed constraints
- Different impedance values for siren on/off
- A* algorithm performs shortest path calculations
- Simulated trajectory along shortest path



Simulated route (red), GPS simulated points along the route spaced by 30 seconds

Security Level

- Model distribution of failure times
- Failure, in this case, is violent crime
- Each beat has a different distribution
- Acts as a dense reward signal, updated every timestep based on time beat has been without police presence
- Kaplan-Meier estimator used for now
- Other models could capture more complex patrol behaviors



Call / Crime Simulation

We are using three different models with different properties. Each has strengths and weaknesses

Homogeneous Poisson Process:

Uniformly sample across region, reject points based on patrol locations

Inhomogeneous Poisson Process:

Sample according to historical density, reject points based on patrol locations

Strauss Marked Point Process:

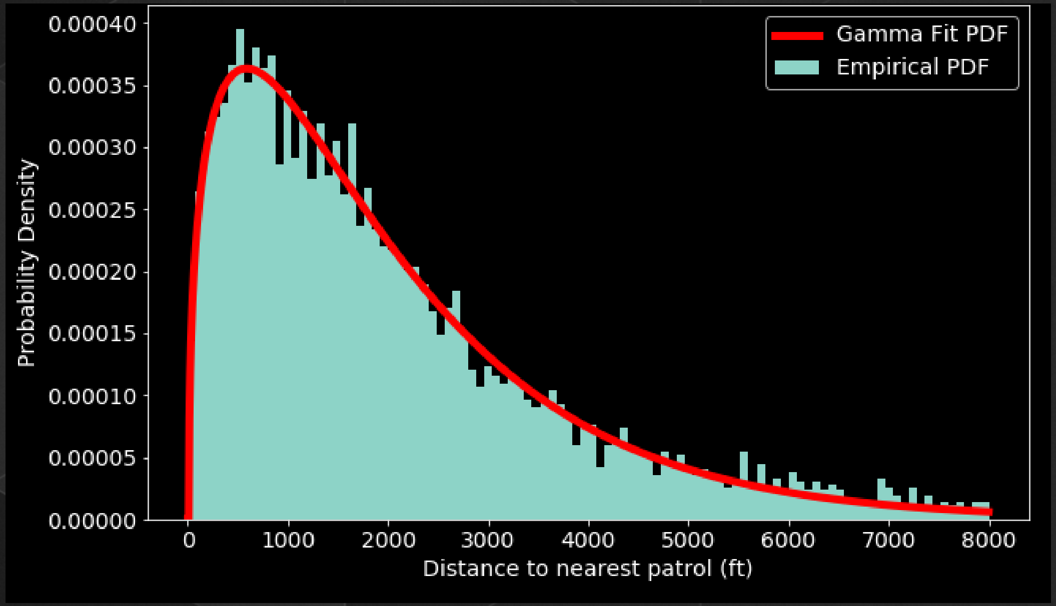
Model attraction/repulsion characteristics between crimes/calls/police

Call / Crime Simulation

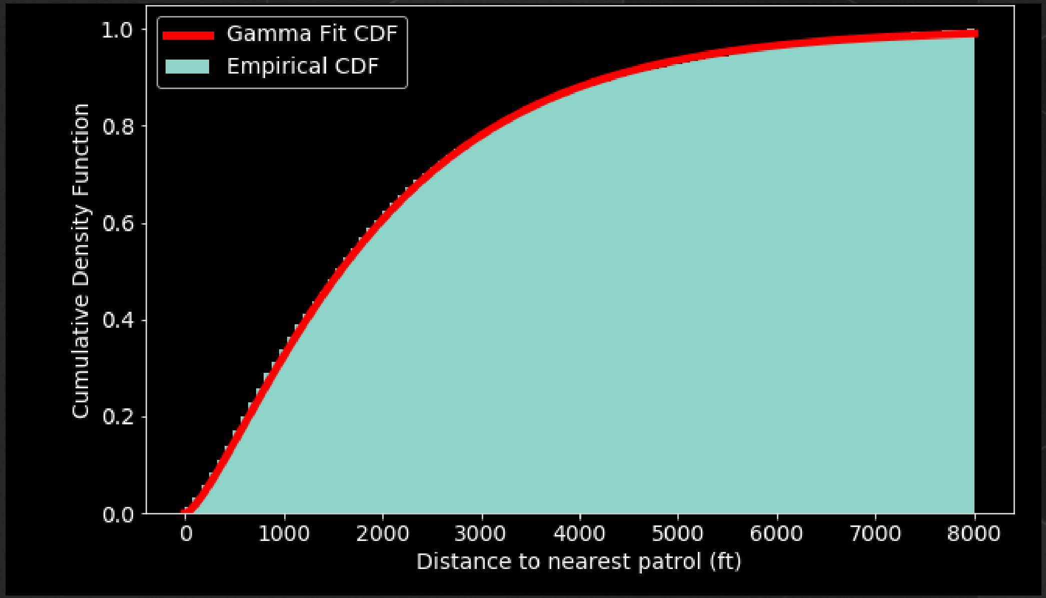
Rejection Region:

Police patrols have a deterrent effect on crimes

We calculate for every crime the distance to closest patrol prior to the crime



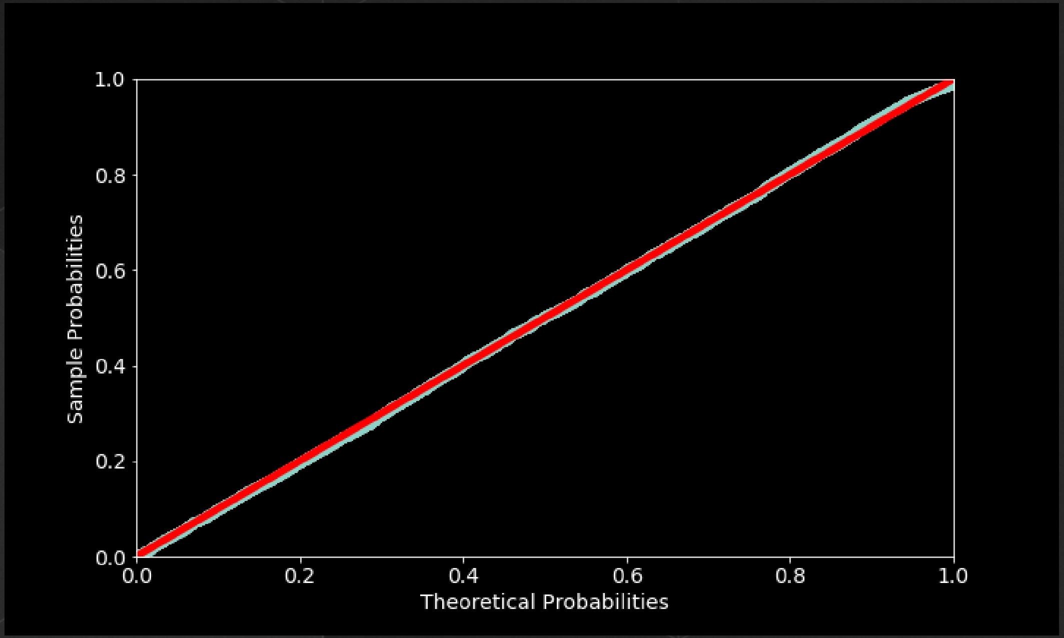
Patrol-Crime Distances PDF, Gamma Fit



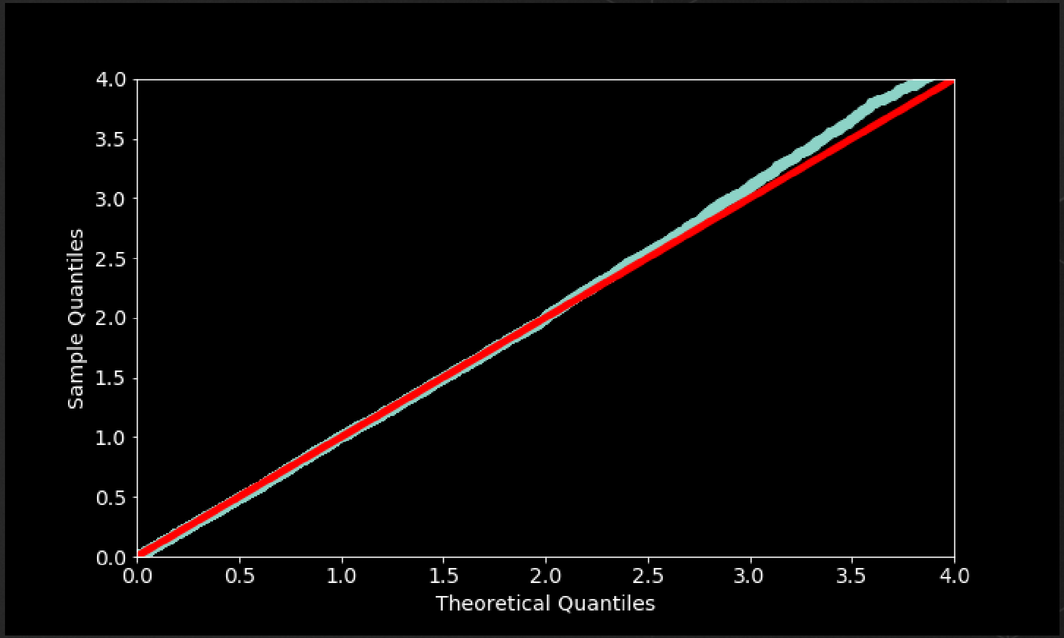
Patrol-Crime Distances CDF, Gamma Fit

Call / Crime Simulation

The fit is very good...



Patrol-Crime Distances PP-Plot

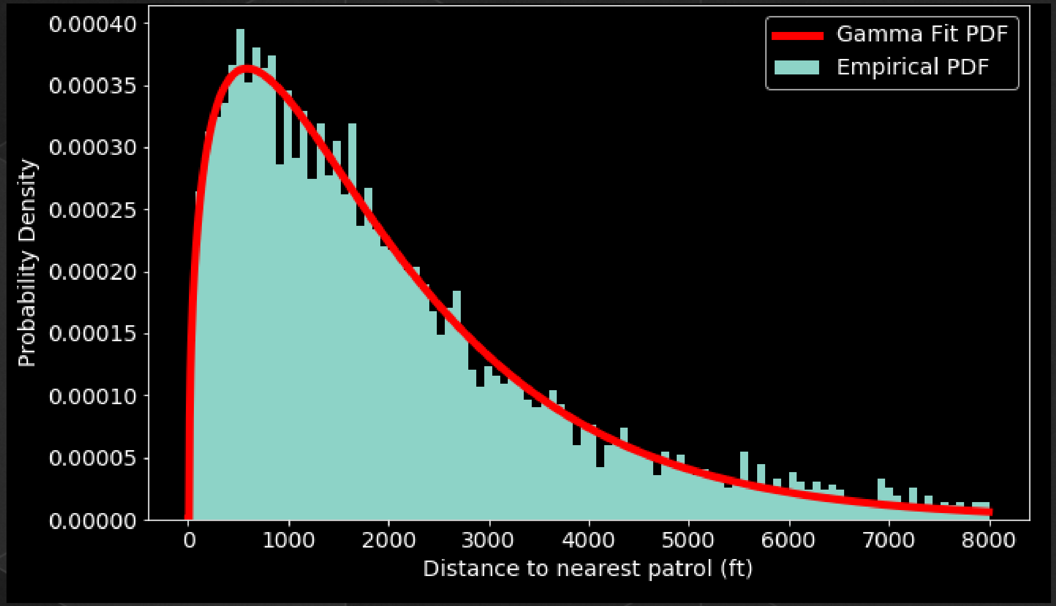


Patrol-Crime Distances QQ-Plot

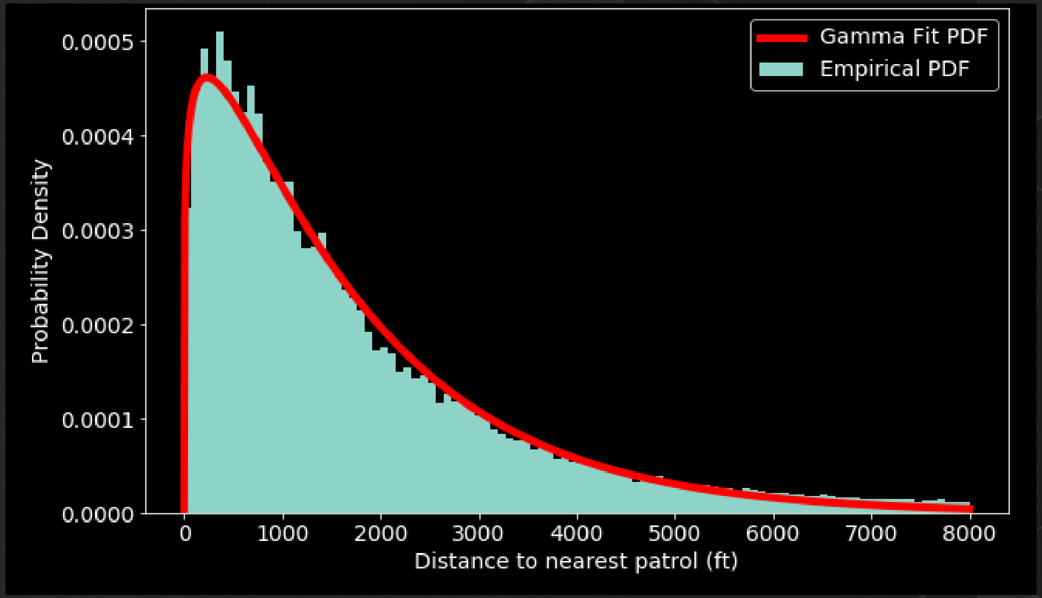
Call / Crime Simulation

Similarly for calls

Calls tend to occur closer to patrols than crimes



Patrol-Crime Distances PDF, Gamma Fit



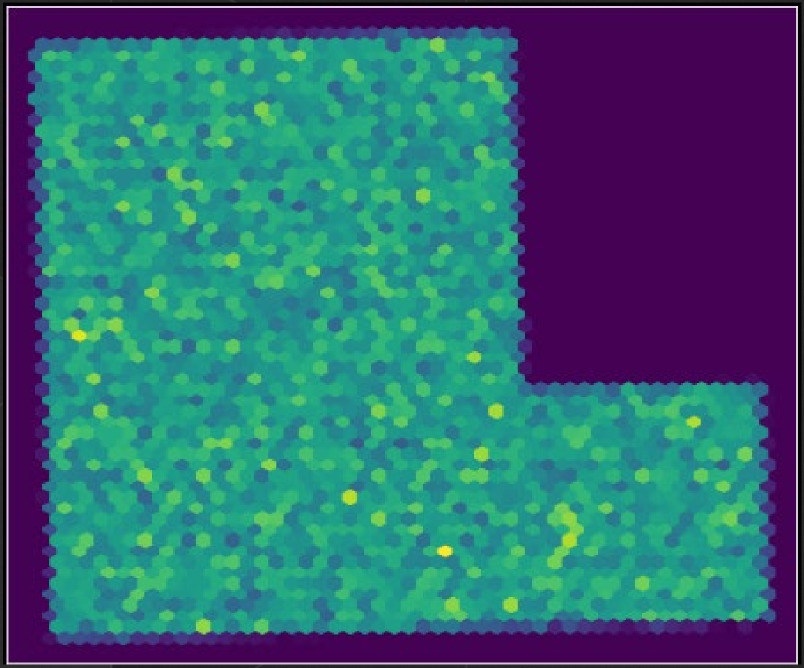
Patrol-Call Distances PDF, Gamma Fit

Call / Crime Simulation

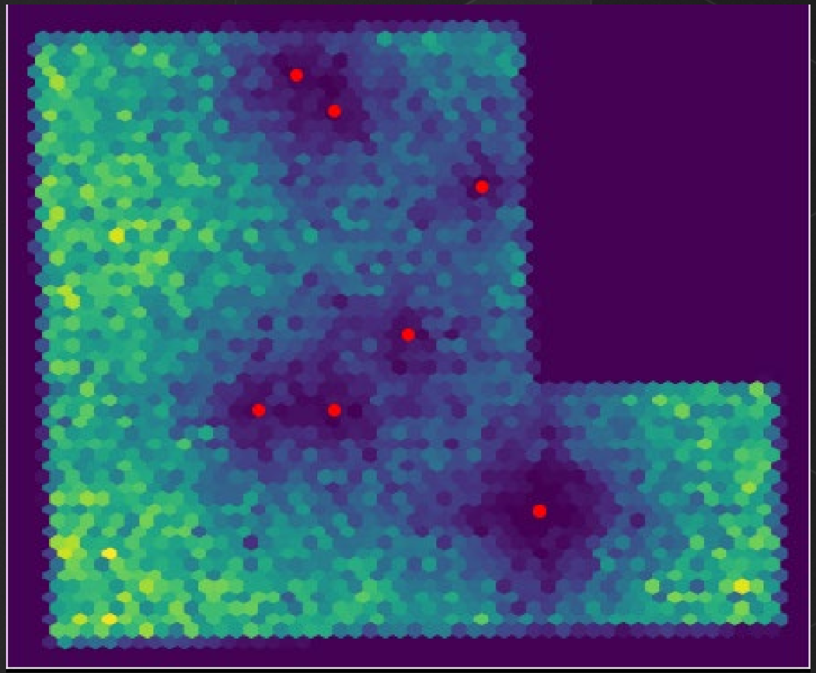
Homogeneous Poisson Process:

Sample according to 2D Poisson process, reject points based on patrol locations

Subject to no bias, but does not reflect expected crime distribution



Sampling from Poisson process, no patrols



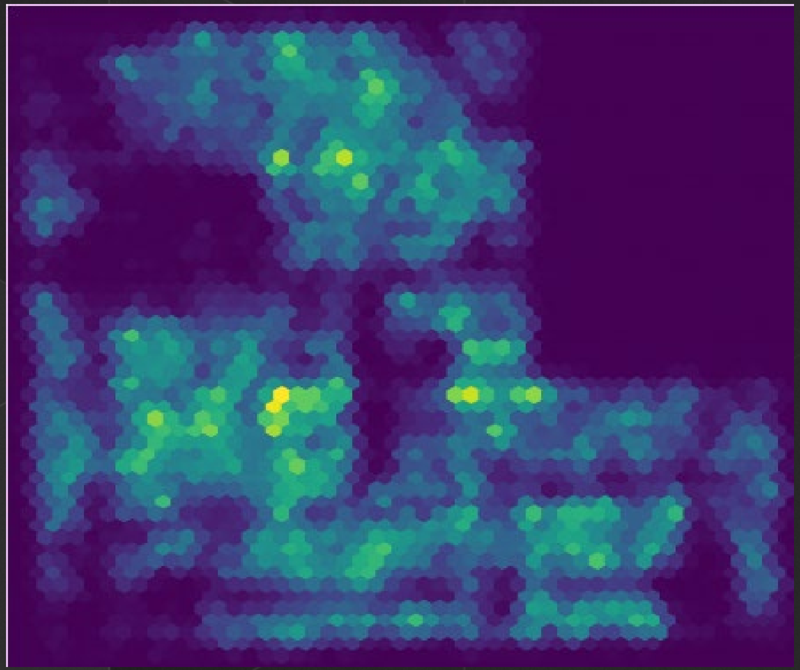
Using patrols for rejection sampling from distribution

Call / Crime Simulation

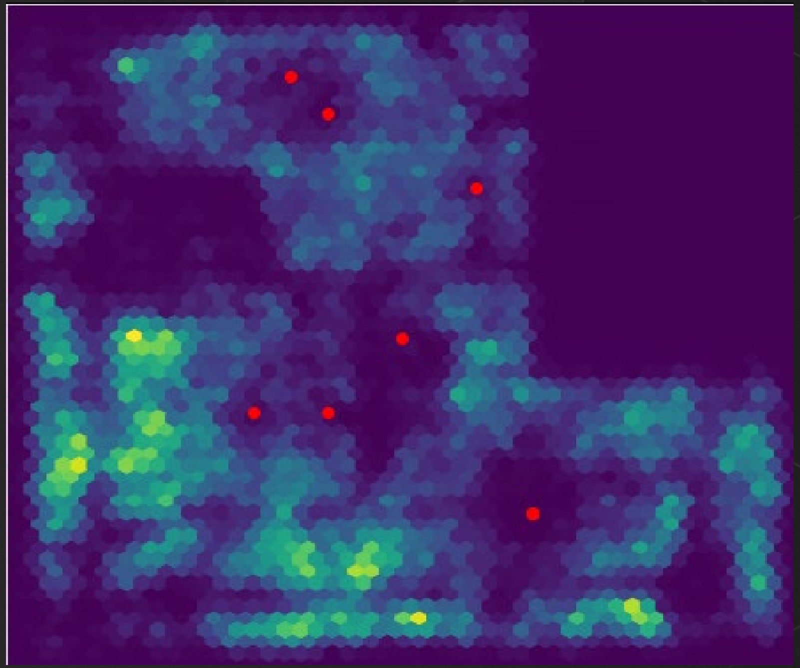
Innomogeneous Poisson Process:

Sample according to historical density, reject points based on patrol locations

Subject to historical bias, but reflects persistent crime hotspots



Sampling from historical distribution, no patrols



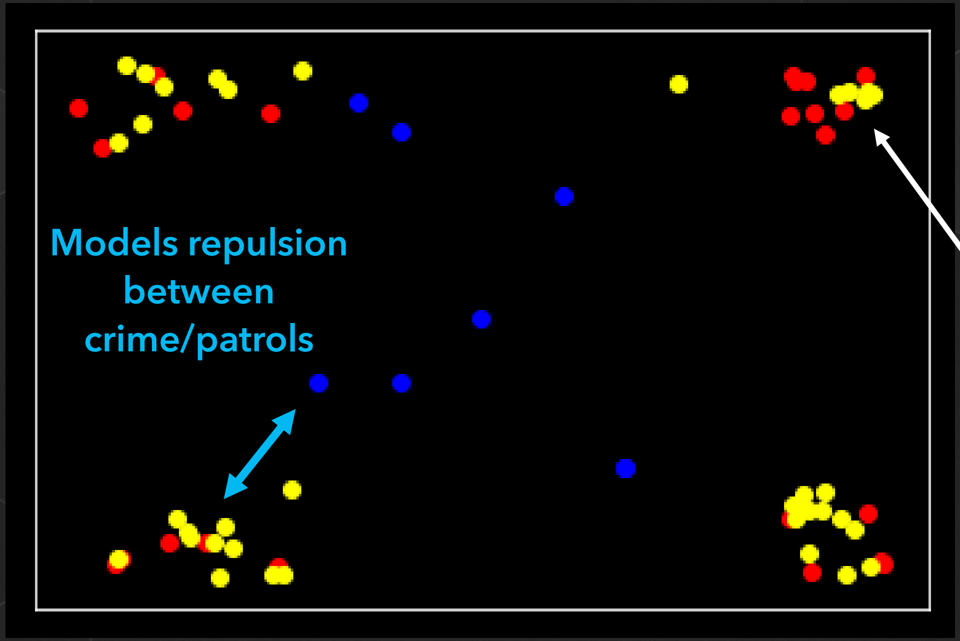
Using patrols for rejection sampling from historical distribution

Call / Crime Simulation

Strauss Marked Point Process:

Strauss Marked Point Process models attraction and repulsion between crimes, calls, and patrols.

No historical bias, more accurate than homogeneous process, doesn't reflect real hotspots



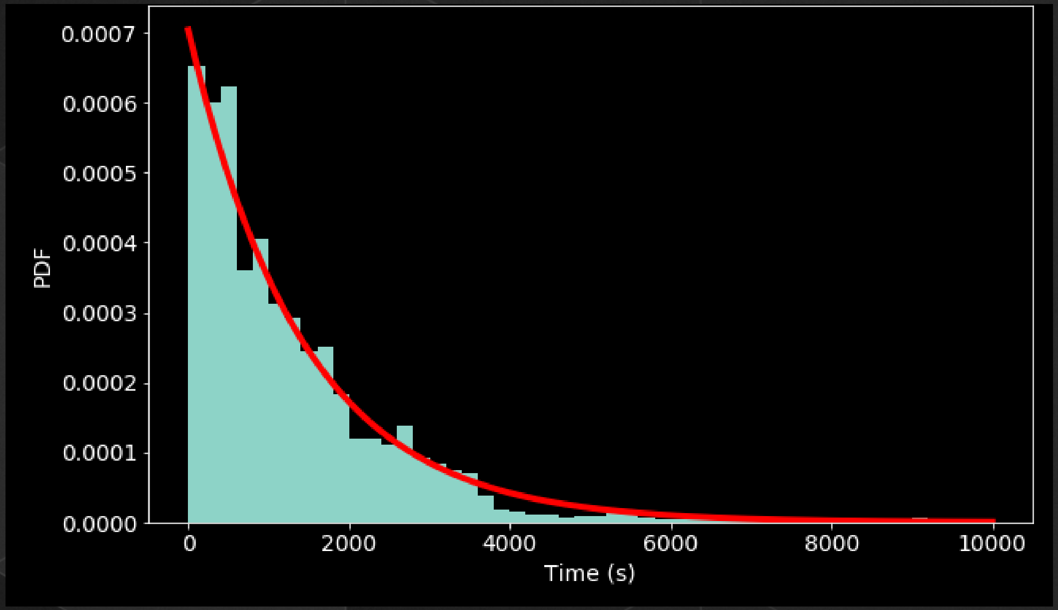
Clustering and self excitation modeled.

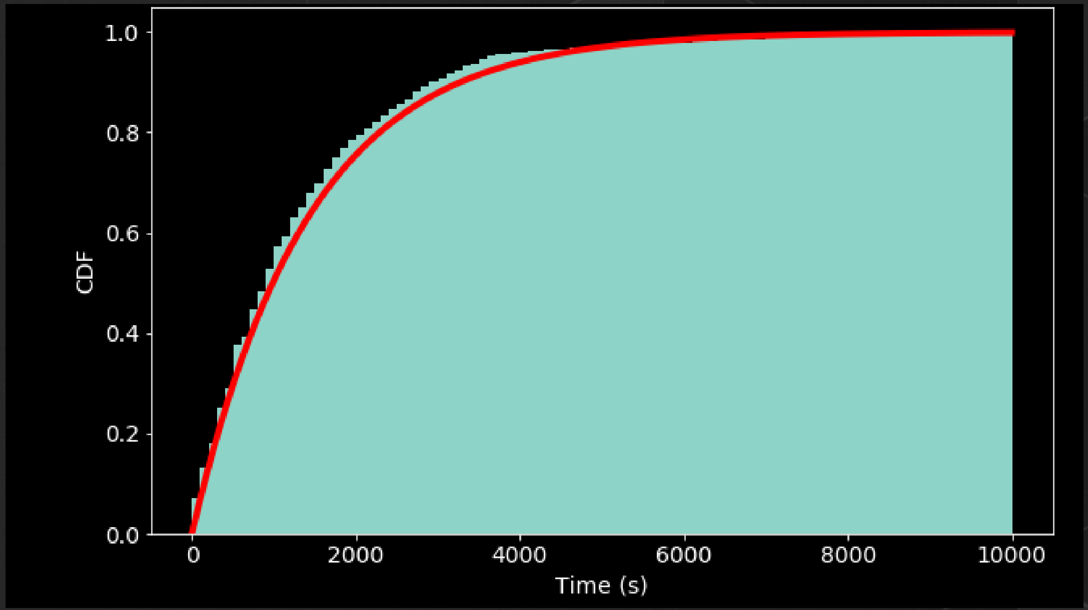
Police (blue) repel certain crime times that attract each other. (Exaggerated)

Call Resolution Simulation

Calls take time to be resolved

Look at distribution of call resolution times. Simulate calls from this distribution





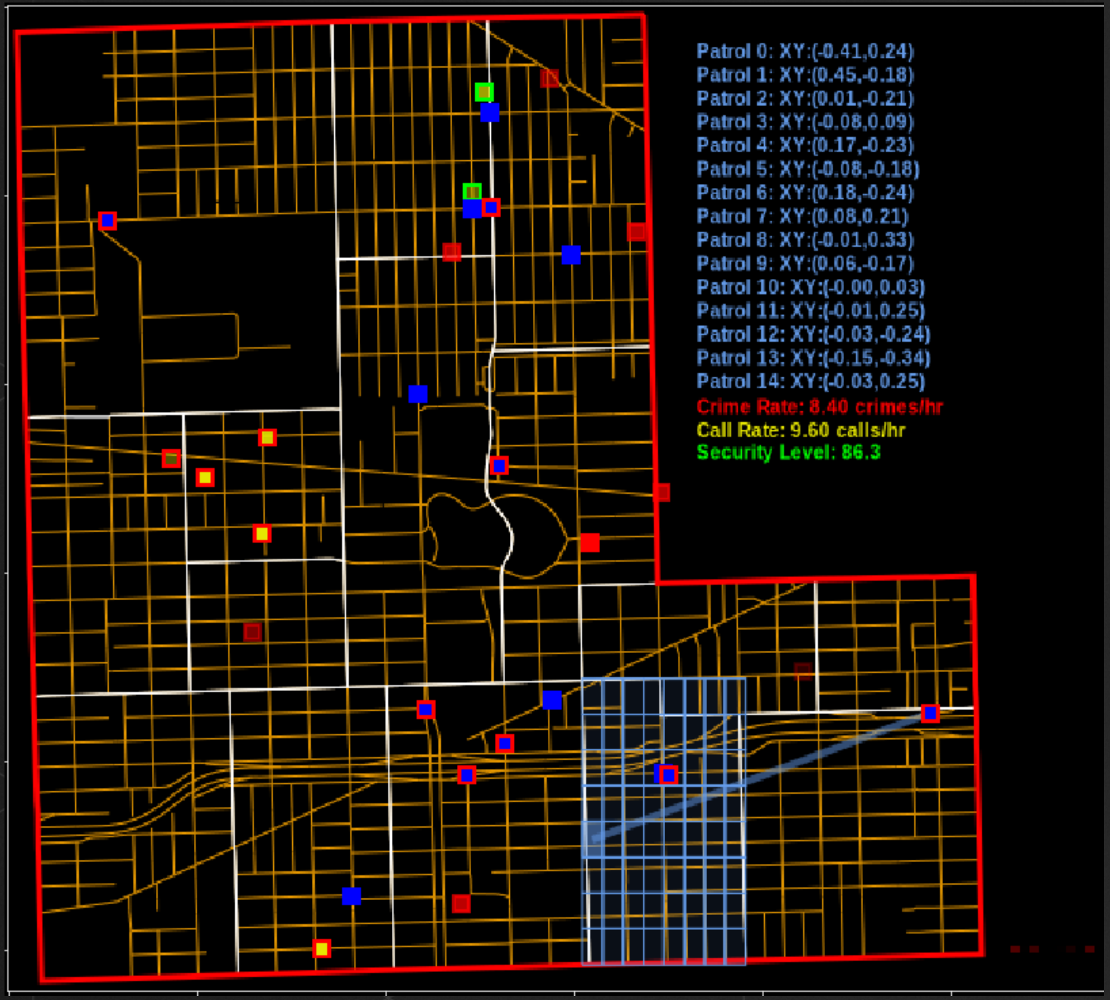
Call Resolution Times, Exponential Fit

Other Environment Details

- Patrols are assigned missions:
  - Respond to call
  - Random patrol in area
  - Loiter in area
  - Return to station
- Each mission has a time duration; patrols cannot be reassigned during a mission (except to respond to a call)
- Patrol missions address areas with high crime through deterrence and keeping security level maximal.

- At each timestep, patrols advanced.
- Agent can optionally assign patrol mission
- Impact on area is modeled and new crimes/calls are sampled
- This process repeats until max timesteps reached

Rendering



We render all the state information* the agent gets into a visual representation

Patrol - Siren On

Patrol - Siren Off

Call - Unanswered

Call - Answered

Crime

Patrol action assignment

Transparency of call/crime reflect age

Gym

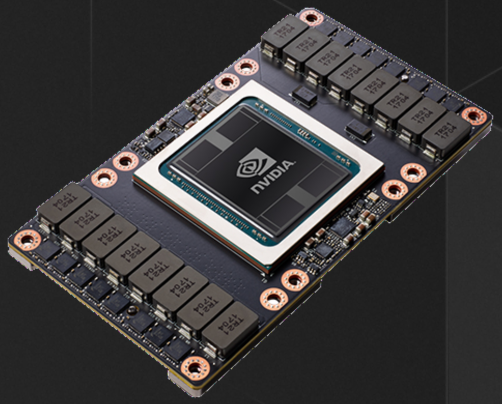
*agent doesn't get road network, beat boundaries, or district boundaries

Distributed Reinforcement Learning

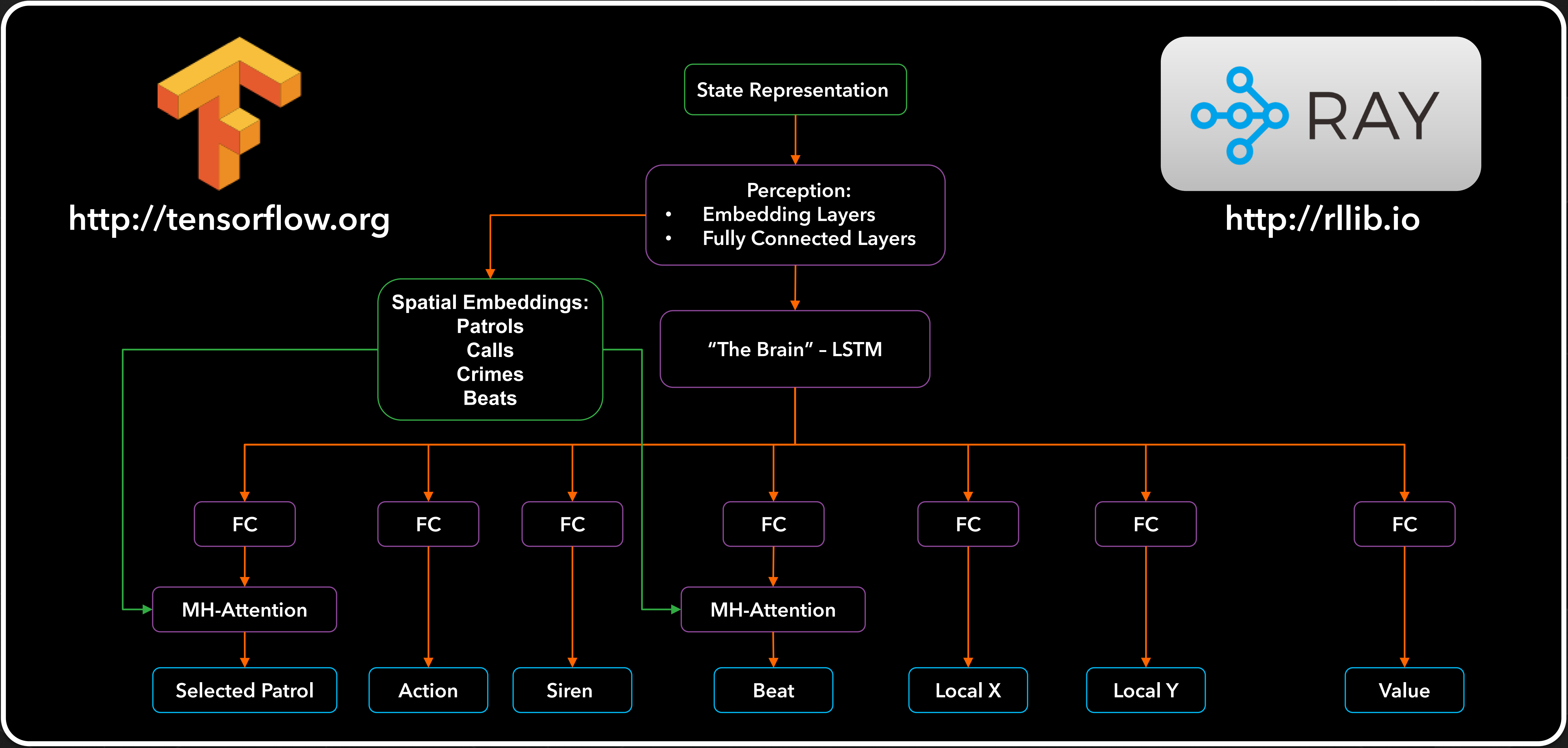
- Distributed, Multi-GPU Learning managed by Ray/Rllib (arXiv:1712.05889)
  - Ray is distributed execution framework
  - Simple use pattern, simple to scale
- Custom Tensorflow Policy
- Proximal Policy Optimization (arXiv:1707.06347)
  - Scales well
  - Simple to tune
  - Flexible
- Training can be scaled up to as many GPU/CPU needed
- Quick updates to policy, explore different strategies
- Utilizing NVIDIA GPU on Microsoft Azure

https://ray.readthedocs.io

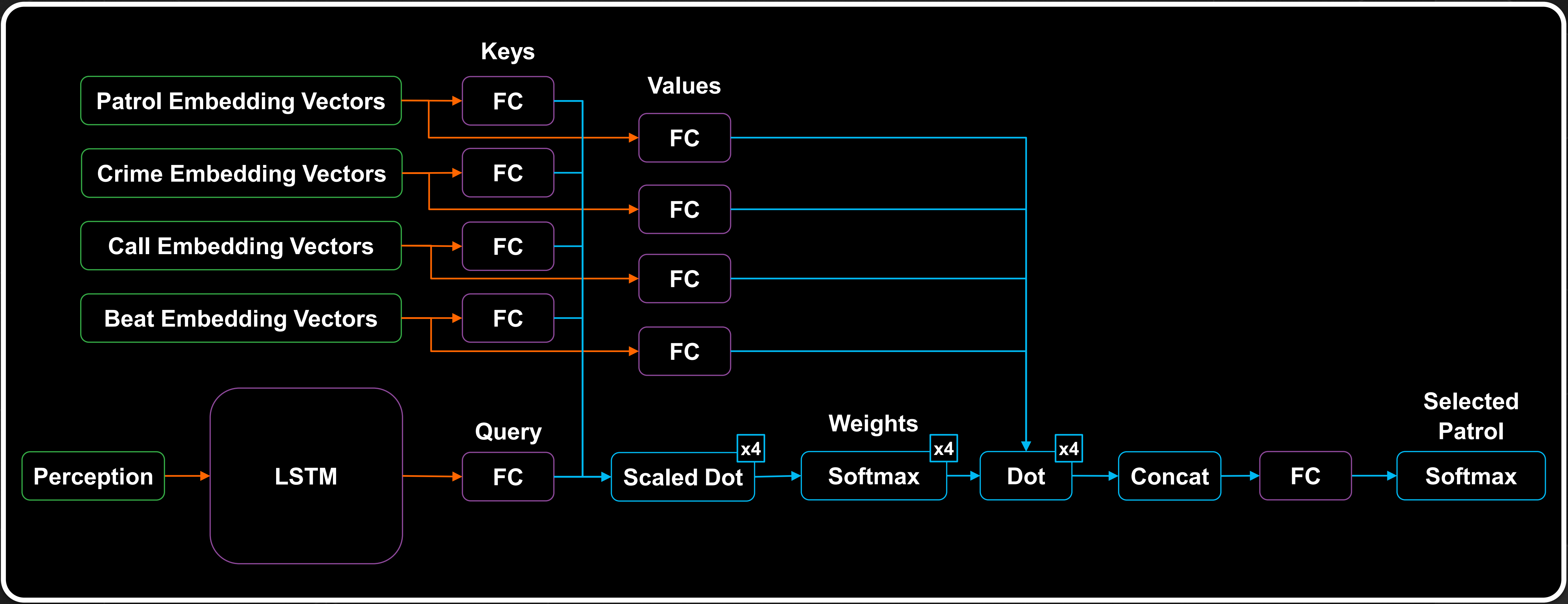
http://rllib.io



Policy / Agent

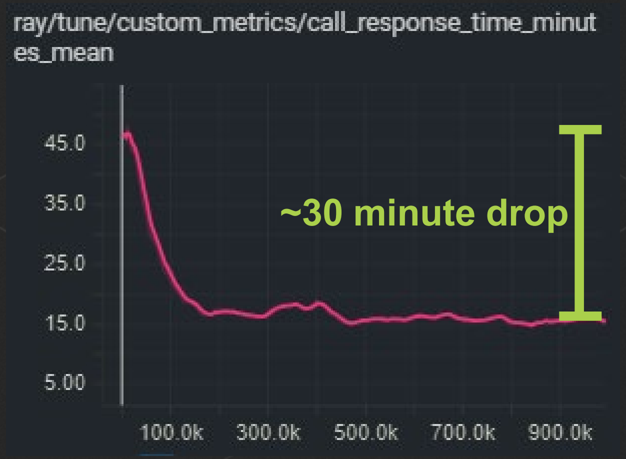


Attention Example: Patrol Selection



Results

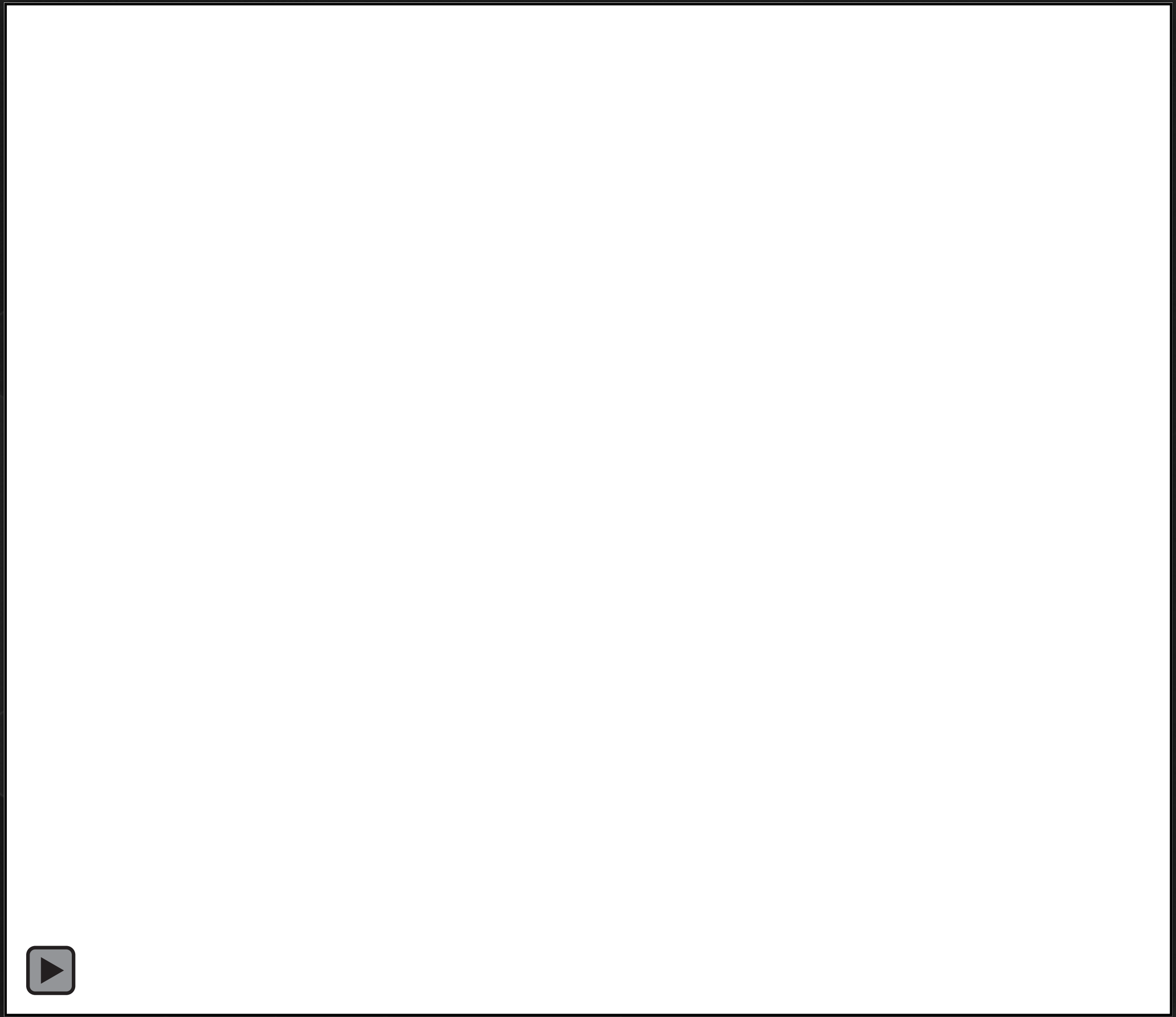
Reward shows convergence after less than 1 million steps



Reward shape emphasizes call response

Penalty for crimes drives crime rate down by a few percent while still swiftly answering calls

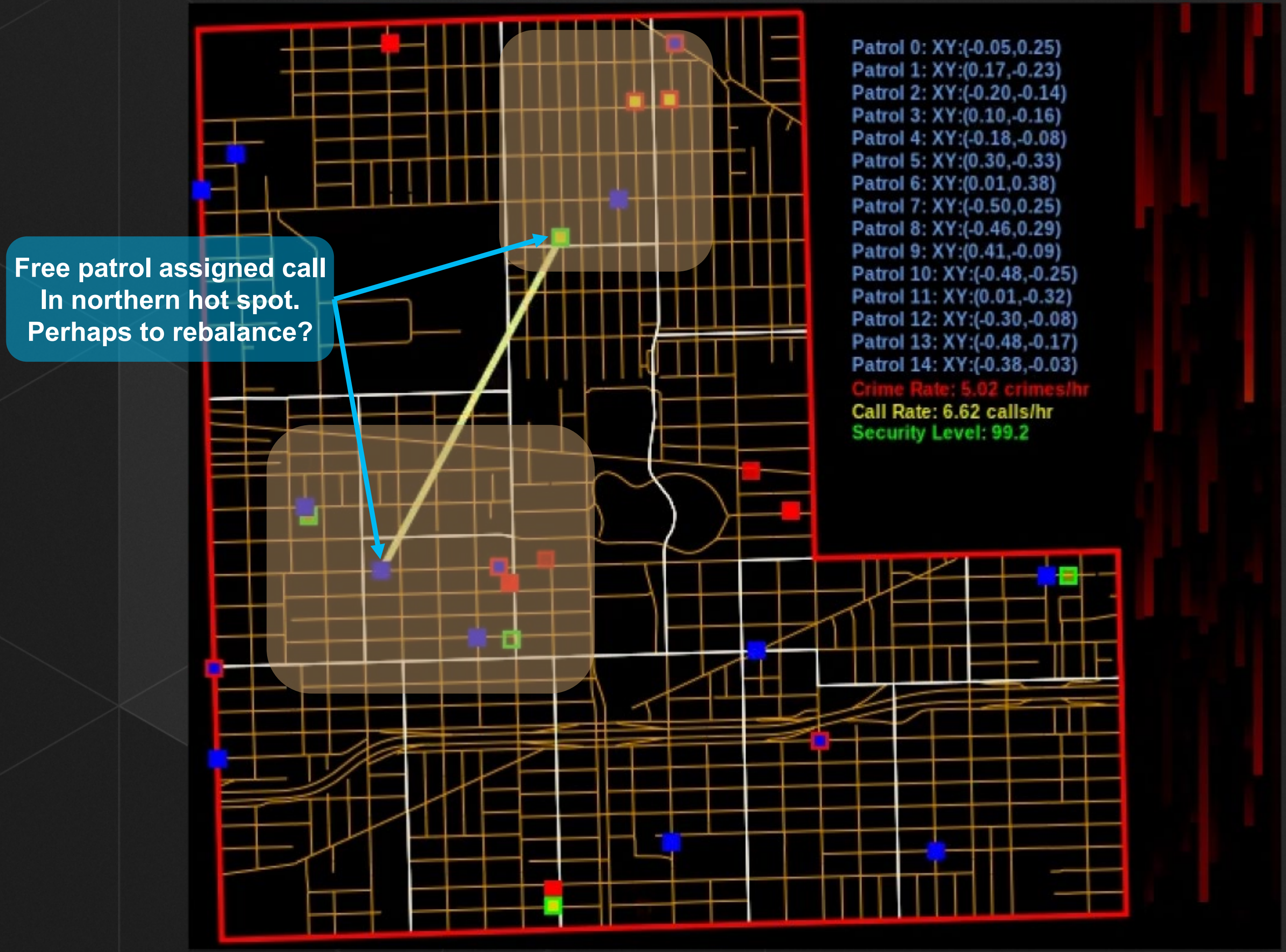
Results



Patrols kept around high risk areas, but broad coverage keeps security levels high.

Calls are promptly responded to by the patrol that maximizes reward

Results



Patrols kept around high risk areas, but broad coverage keeps security levels high.

Calls are promptly responded to by the patrol that maximizes reward

Next Steps

- Further analysis/study
  - Create more expert baseline agents (how do hand crafted rules compare?)
  - Any systematic biases that are unwanted?
  - Additional reward/penalty signals?
  - Best reward shaping to achieve desired behavior?
  - More informative state representations
  - Better Safety/Security Models (Account for covariates)
  - WGAN Generative point process (arXiv: 1705.08051)

- Multi-agent:
  - Agent to district: Multiple agents optimize city strategy from their individual jurisdiction
  - Agent per patrol: Each patrol has its own decision making strategy

- Pilot deployment
  - What's the best way to apply in a noninvasive, safe manner?
  - Identify improvements
  - Discover new policing insights
  - Feedback from the experts

Questions?

My Contact:

Email: dwilson@esri.com

LinkedIn: https://www.linkedin.com/in/daniel-wilson-a274b218/

esri

THE
SCIENCE
OF
WHERE