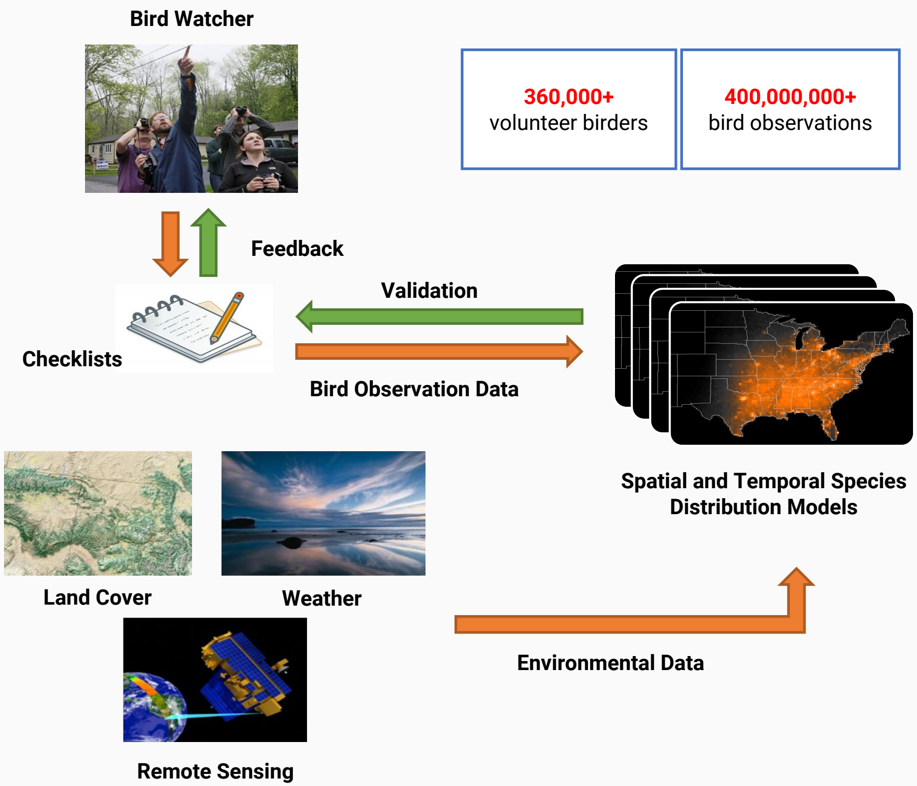
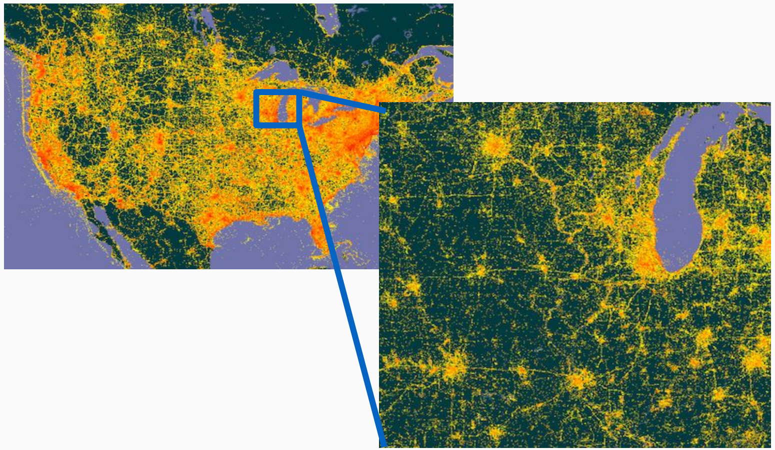
Avicaching

GPU-accelerated Bias Reduction in Citizen Science

Anmol Kabra[1], Yexiang Xue[2], Carla P. Gomes[1]
[1]Computer Science Dept., Cornell University; [2]Computer Science Dept., Purdue University

Spatial bias in citizen science projects, esp. eBird

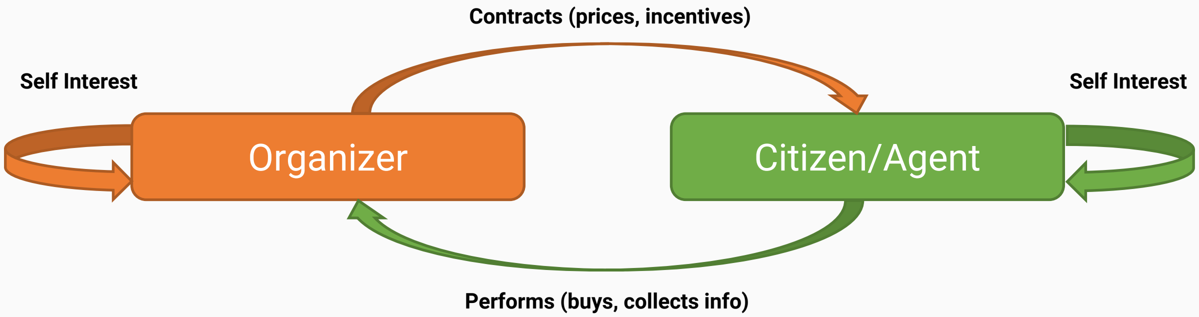




Spatially concentrated distribution of eBird observations in the US
(coinciding with urban areas)

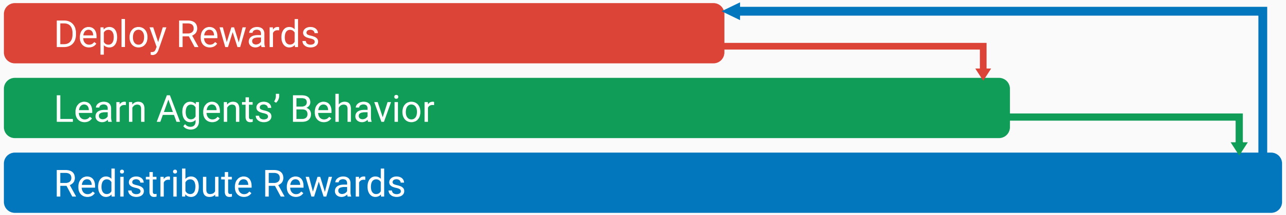
Incentivizing citizens for favorable actions

- Reduce spatial bias by incentivizing visits to undersampled locations
- A 2-stage method to determine incentives' placement and quantity:



Identification Problem

Pricing Problem



Identification Problem

Learn how citizens behave on rewards.
That is, learn a map from old to new visit densities.

1

$$\begin{bmatrix} x_1 \\ x_2 \\ . \\ . \\ . \\ x_J \end{bmatrix} \xrightarrow{P(w,\, r)} \begin{bmatrix} y_1 \\ y_2 \\ . \\ . \\ . \\ y_J \end{bmatrix}$$

Before r_j placed   After r_j placed

$$w^* = \text{argmin}_w\, \Sigma_t\, \omega_t\, || y_t - P(w, r) x_t ||_2$$

Pricing Problem

Change rewards to minimize variance in predicted visit densities, based on learned map.

3

$$r^* = \text{argmin}_r\, || y - y_{\text{mean}} ||_2$$

2

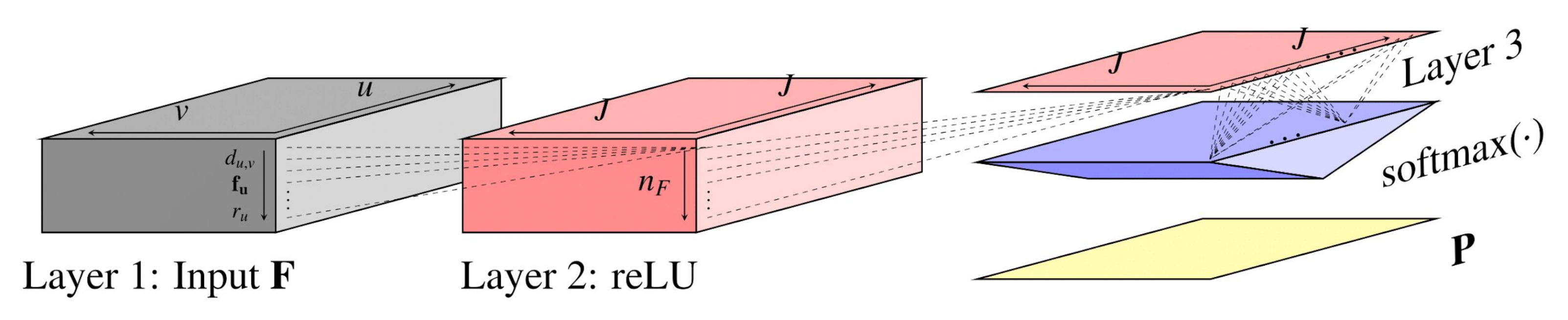
subject to:

- $y = P(w^*, r)x$
- $C_I r \leq 0$
- $C_E r = 0$

where $x = \text{average}(x_t)$

Rewards are subject to constraints, such as non-negativity, total maximum rewards ($1^{\text{T}} r \leq R$), rewards only being placed on qualified locations etc.

Solving both problems with Machine Learning techniques on GPUs



Since P is a map from location space to location space, $p_{u,v}$ represents the likelihood of citizens moving from location v to location u. We find this likelihood with a multi-layer neural network with rewards, location distances, and environmental features as input features.

- The map $P(w, r)$ can be modeled as a neural network, identifying weights for different factors that might change x_t to y_t.
- Both w^* and r^* optimized using gradient-descent using parallel computation on GPUs.
- Obtained a **72x** speedup in runtime compared to MIP models used previously, demonstrating our model's scalable infrastructure (runtime for model over 30 locations is ~20 min compared to ~1 day).