GPU-accelerated Principal-Agent Game for Scalable Citizen Science

Anmol Kabra[1], Yexiang Xue[2], Carla P. Gomes[1]

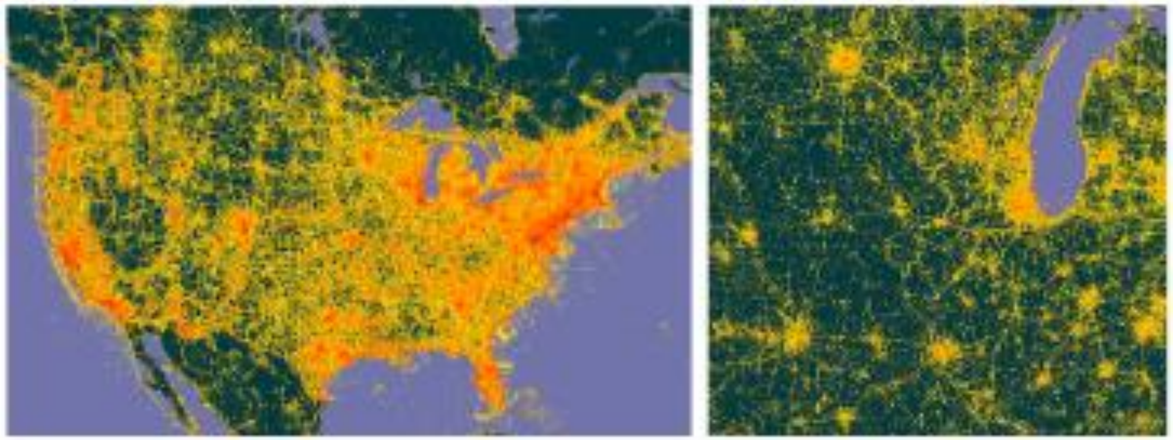
ak2426@cornell.edu yexiang@purdue.edu gomes@cs.cornell.edu

[1]Cornell University, [2]Purdue University

ACM Computing & Sustainable Societies 2019

Sampling Bias in Citizen Science

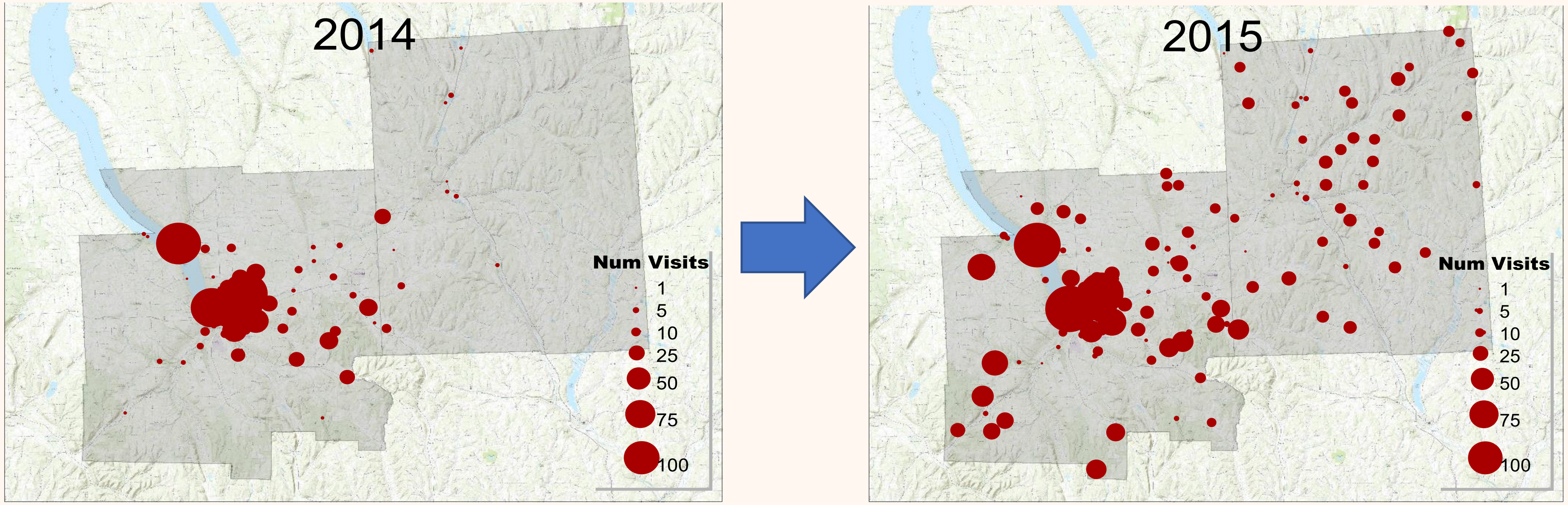
- Crowdsourcing/Citizen Science programs (*eBird*, *Zooniverse*, *CoralWatch*) engage public in collecting data for research problems
- Data used for policy making, environmental conservation etc.
- Citizens' motivations for tasks ➔ Sampling bias ➔ Spatial clustering



Spatial clustering in Mainland and Midwest US in *eBird* before 2014 (Xue, 2016a)

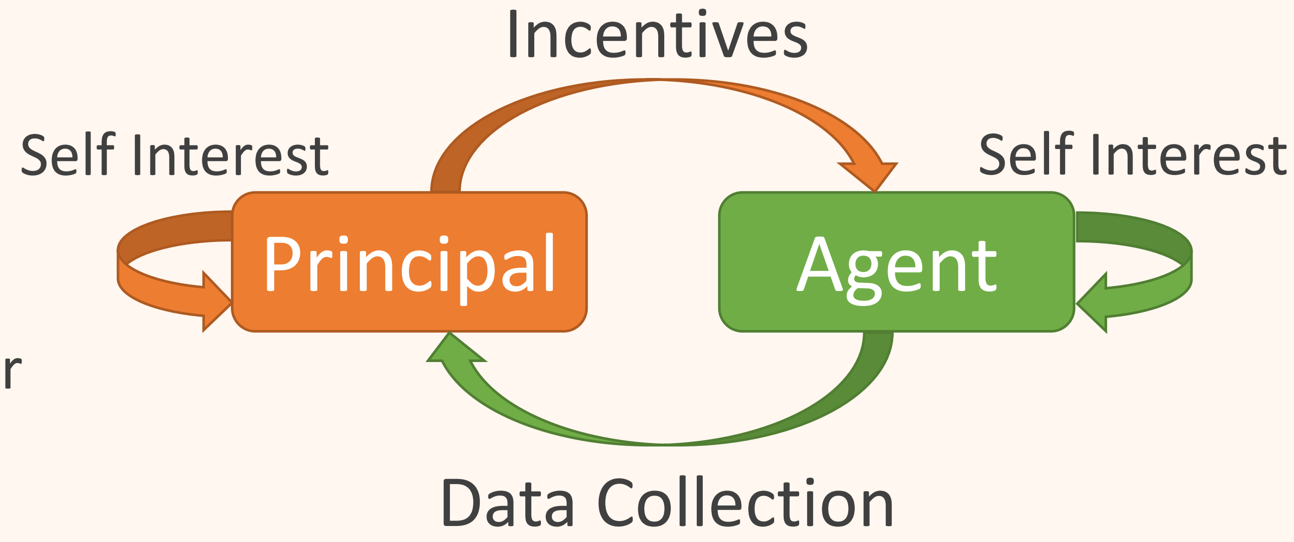
Previous Approaches

- Misaligned motivations of program (principal) and citizens (agent)
- *Avicaching*: incentivize citizens to visit under-sampled locations
    - 20% shift in *eBird* submissions after *Avicaching* (Xue, 2016a)



Previous Approaches

- A Principal-Agent game: model citizen behavior to distribute effective rewards
- Two subproblems:
  - Identification: learn agent behavior
  - Pricing: redistribute rewards
- MIP solves pricing, identification embedded
  - 3 hours for ≈30 locations



MIP = Mixed-Integer-Programming

Formalizing the Problem: Identification

$$\underbrace{\begin{pmatrix} x_{t,1} \\ \vdots \\ x_{t,n} \end{pmatrix}}_{\text{Before } \mathbf{r_t}} \xrightarrow{\mathbf{P}_{\mathbf{f},\mathbf{r_t};\mathbf{w}}} \underbrace{\begin{pmatrix} y_{t,1} \\ \vdots \\ y_{t,n} \end{pmatrix}}_{\text{After } \mathbf{r_t}}$$

- For time period t, $\mathbf{x_t} \in \mathbb{R}^n$ are visit densities of n locations before rewards $\mathbf{r_t} \in \mathbb{R}^n$ were placed; $\mathbf{y_t}$ are visit densities after placement
- Goal: learn matrix $\mathbf{P}$ s.t. $\mathbf{P}\mathbf{x_t} \approx \mathbf{y_t}$
  - $\mathbf{P}$ depends on features of locations $\mathbf{f}$, rewards $\mathbf{r_t}$, with parameters $\mathbf{w}$
  - $p_{u,v} = \Pr(\text{shift of submissions from location } v \text{ to } u)$

$$\mathbf{w}^* = \operatorname*{argmin}_{\mathbf{w}} \sum_t \|\mathbf{y_t} - \mathbf{P}\mathbf{x_t}\|_2^2$$

Formalizing the Problem: Pricing

With $\mathbf{w}^*$ learned from Identification

$$\mathbf{P}_{\mathbf{f},\mathbf{r};\mathbf{w}^*}$$

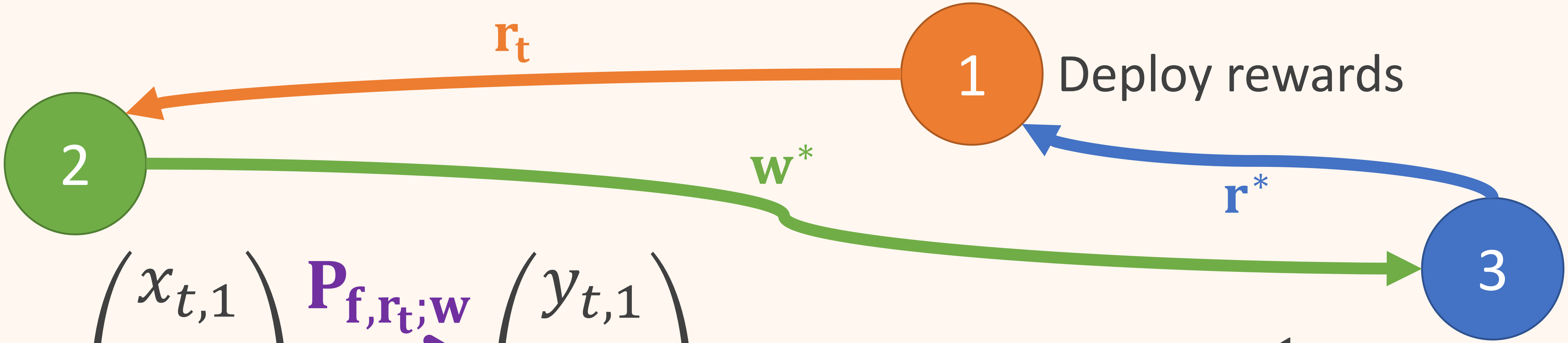
$$\underbrace{\begin{pmatrix} x_{t,1} \\ \vdots \\ x_{t,n} \end{pmatrix}}_{\text{Before } \mathbf{r_t}} \xrightarrow{\mathbf{P}_{\mathbf{f},\mathbf{r};\mathbf{w}^*}} \underbrace{\begin{pmatrix} y_{t,1} \\ \vdots \\ y_{t,n} \end{pmatrix}}_{\text{After } \mathbf{r_t}}$$

- Goal: distribute rewards s.t. future visit density is uniform
- With $\mathbf{x} = \sum_t \mathbf{x_t}$, reduce variance of $\mathbf{y} = \mathbf{P}_{\mathbf{f},\mathbf{r};\mathbf{w}^*}\mathbf{x}$

$$\mathbf{r}^* = \underset{\mathbf{r}}{\operatorname{argmin}} \frac{1}{n} \|\mathbf{y} - \bar{\mathbf{y}}\|_1$$

- Constraints on $\mathbf{r}$:
  - Sum up to budget R, i.e., $\|\mathbf{r}\|_1 \leq R$
  - Non-negative, i.e., $\forall\, u, r_u \geq 0$

Scaling up the Game with Machine Learning



$$\begin{pmatrix} x_{t,1} \\ \vdots \\ x_{t,n} \end{pmatrix} \xrightarrow{\mathbf{P}_{\mathbf{f},\mathbf{r_t};\mathbf{w}}} \begin{pmatrix} y_{t,1} \\ \vdots \\ y_{t,n} \end{pmatrix}$$

Before $\mathbf{r_t}$ After $\mathbf{r_t}$

$$\mathbf{w}^* = \underset{\mathbf{w}}{\operatorname{argmin}} \sum_t \|\mathbf{y_t} - \mathbf{P}\mathbf{x_t}\|_2^2$$

$$\mathbf{r}^* = \underset{\mathbf{r}}{\operatorname{argmin}} \frac{1}{n} \|\mathbf{y} - \bar{\mathbf{y}}\|_1$$

subject to:

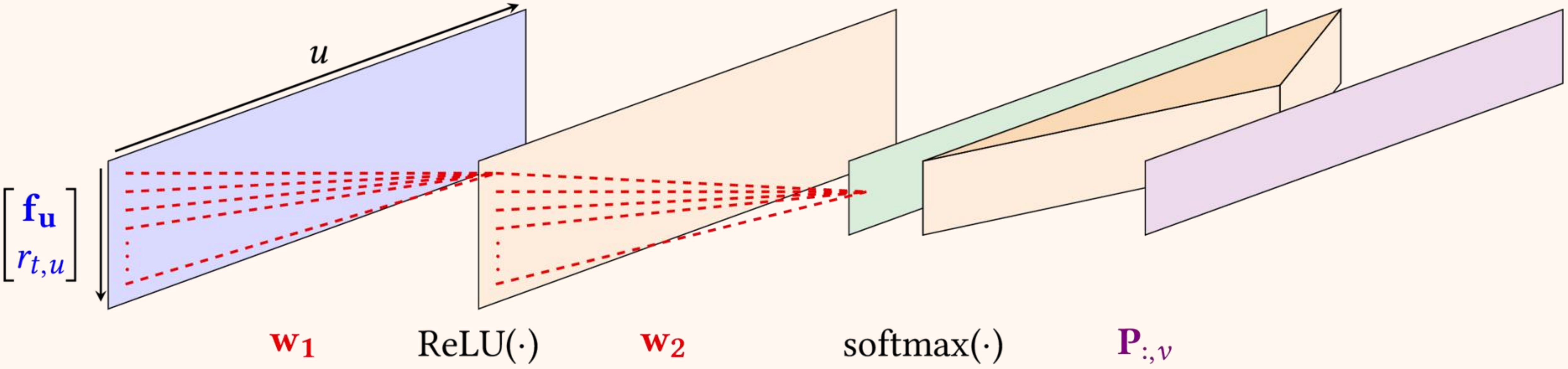
- $\mathbf{y} = \mathbf{P}_{\mathbf{f},\mathbf{r};\,\mathbf{w}^*}\,\mathbf{x}$
- $\|\mathbf{r}\|_1 \leq R$
- $\forall\, u, r_u \geq 0$

Scaling up the Game with Machine Learning

- Recall: MIP-based approach embedded Identification as linear constraints in Pricing
  - Optimal for Pricing, but not scalable or fast (standard CPU hardware)
  - Identification embedded as linear constraints

  ➔ Model can't capture non-linear behavior
- Our work:
  - $p_{u,v}$ can be non-linear, result of a sequence of non-linearities
  - Parallelizable on GPUs: fast and scalable
  - Rewards can be non-integers

Scaling up the Game with Machine Learning

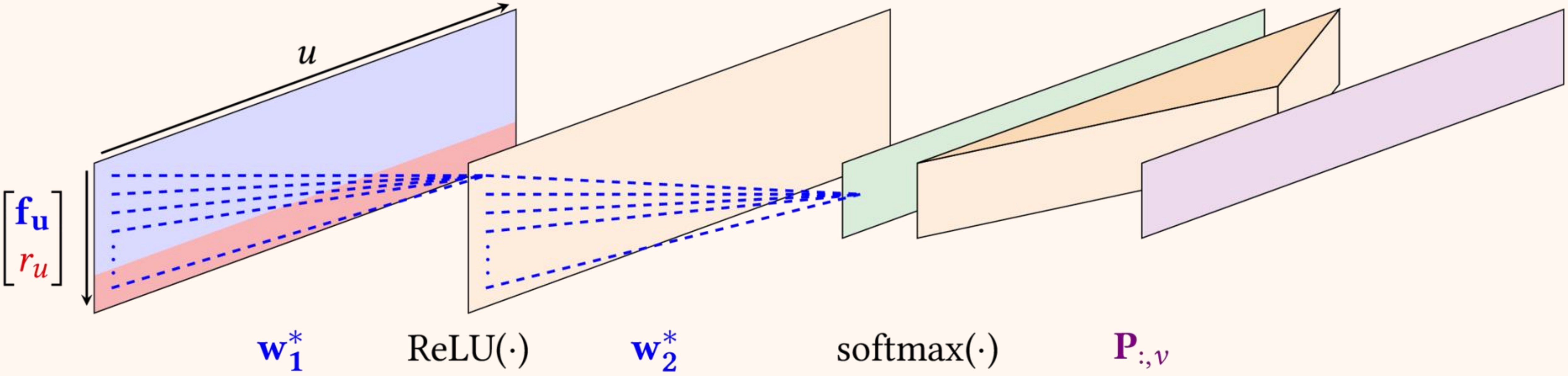


A 3-layer neural network for Identification Problem

For a location v, each vertical slice of the network weighs features of all locations u to get $\Pr$(shift of submissions from v to other locations)

Red variables are optimized, blue do not change

Scaling up the Game with Machine Learning



Same network as before for Pricing Problem, only optimizing $\mathbf{r}$

For a location v, each vertical slice of the network adjusts r_u to minimize variance of predicted visit densities, $\mathbf{y}$

Red variables are optimized, blue do not change

Experiments

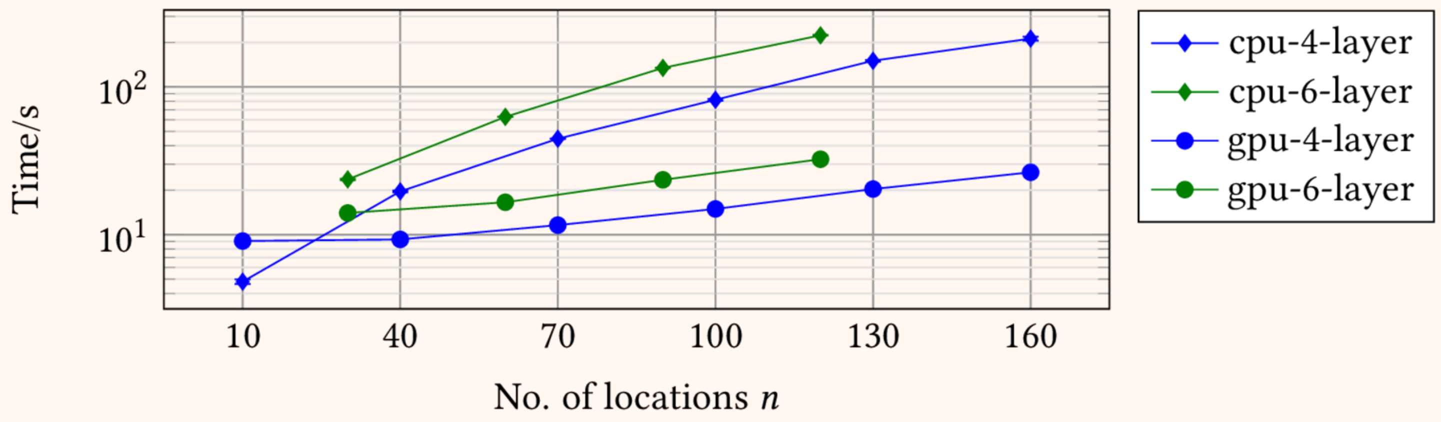
- Goals:
  - Improve speed and scalability
  - Not lose performance on objective

$$\min_{\mathbf{w}} \sum_t \|\mathbf{y_t} - \mathbf{P}\mathbf{x_t}\|_2^2$$

Identification Problem

$\# \, t = 182 \cdot n = 116 \cdot$ # features $= 34 \cdot$ 75-5-20 split · Adam algorithm for gradient descent

Model	Loss	Runtime (s)
Random	1.014	—
Random Forest	0.491	26.4
BFGS (Xue, 2016b)	0.374	507.3
2-layer	**0.366**	48.0
6-layer	**0.358**	647.8



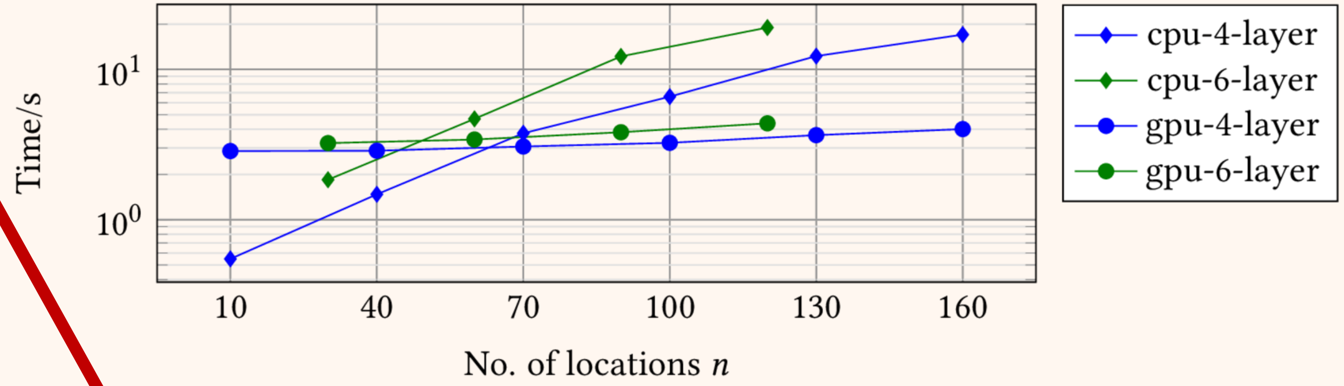
Experiments

$$\min_{\mathbf{r}} \frac{1}{n}\|\mathbf{y}-\bar{\mathbf{y}}\|_1$$

Pricing Problem

$R = 365 \cdot n = 116 \cdot$ Adam algorithm for optimization

Model	Objective	Runtime (s)
MIP (Xue, 2016b)	1.110	≥ 36,000
2-layer	**1.073**	9.65
4-layer	1.236	26.80
6-layer	**1.025**	44.45



6-layer network 800x faster than MIP

Conclusion

- A novel approach to solve Principal-Agent game for reducing sampling bias in large-scale citizen science programs
- Compared to the previous state-of-the-art MIP, our neural-network-based approach delivers slightly better performance and orders of magnitude speedup with GPUs
- Future areas of study:
    - Memory-efficient networks
    - End-to-end learning framework for convenient deployment

Thanks!

GPU-accelerated Principal-Agent Game for Scalable Citizen Science

Contact:

- Anmol Kabra: ak2426@cornell.edu, @anmolkabra, anmolkabra.com
- Yexiang Xue: yexiang@purdue.edu
- Carla Gomes: gomes@cs.cornell.edu

References

- (Xue, 2016a) Yexiang Xue, Ian Davies, Daniel Fink, Christopher Wood, and Carla P. Gomes. 2016. Avicaching: A Two Stage Game for Bias Reduction in Citizen Science. In *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems (AAMAS)*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 776–785. https://dl.acm.org/citation.cfm?id=2936924.2937038
- (Xue, 2016b) Yexiang Xue, Ian Davies, Daniel Fink, Christopher Wood, and Carla P. Gomes. 2016. Behavior Identification in Two-Stage Games for Incentivizing Citizen Science Exploration. In *Principles and Practice of Constraint Programming*, Michel Rueher (Ed.). Springer International Publishing, Cham, 701–717. https://doi.org/10.1007/978-3-319-44953-1_44