Can we emulate the LSST detection pipeline using neural nets?

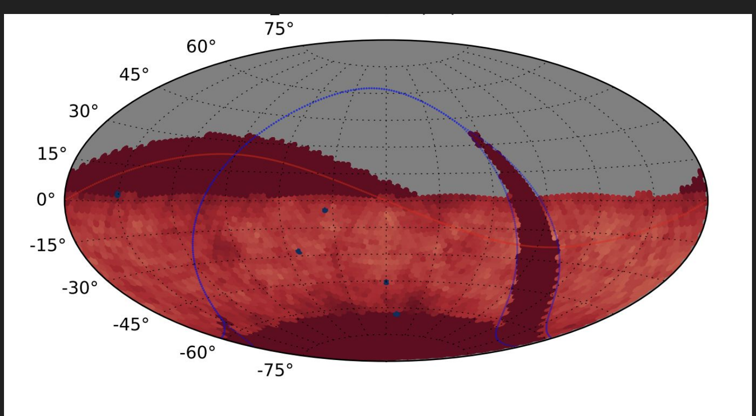
Paxton Scott

Outline

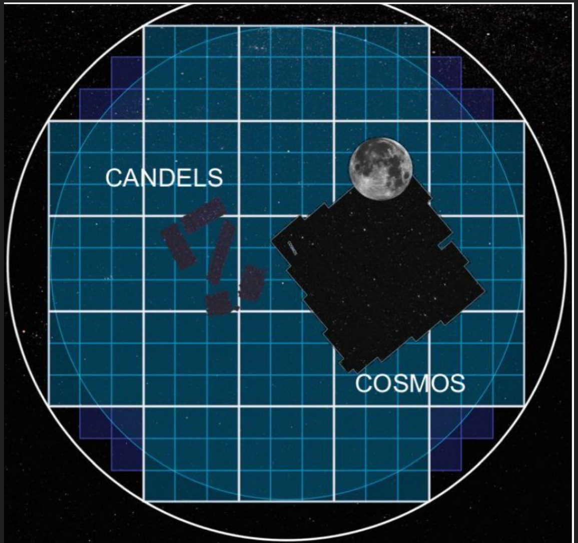
1. **Rubin Observatory/LSST overview**
2. Emulation project outline
3. Data and model results
4. Next Steps
5. Questions

Rubin Observatory/LSST

- Rubin Observatory will conduct 10 year Legacy Survey of Space and Time (LSST)
- Two of the dark energy probes:
    - Weak lensing
    - large scale structure



LSST Observing Strategy (2018)



The problem: overlapping galaxy images (blending)

- LSST looks deeper (further back in time)
- More overlapping galaxies images

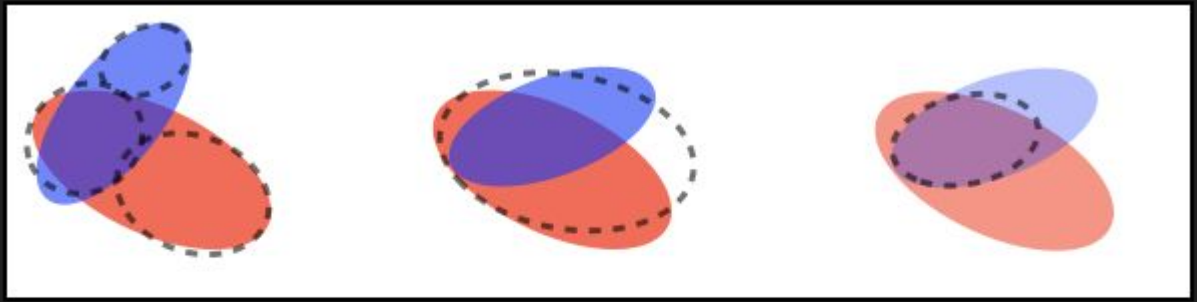


Image from Sowmya Kamath’s thesis defense

Shredded blends

Unrecognized blends

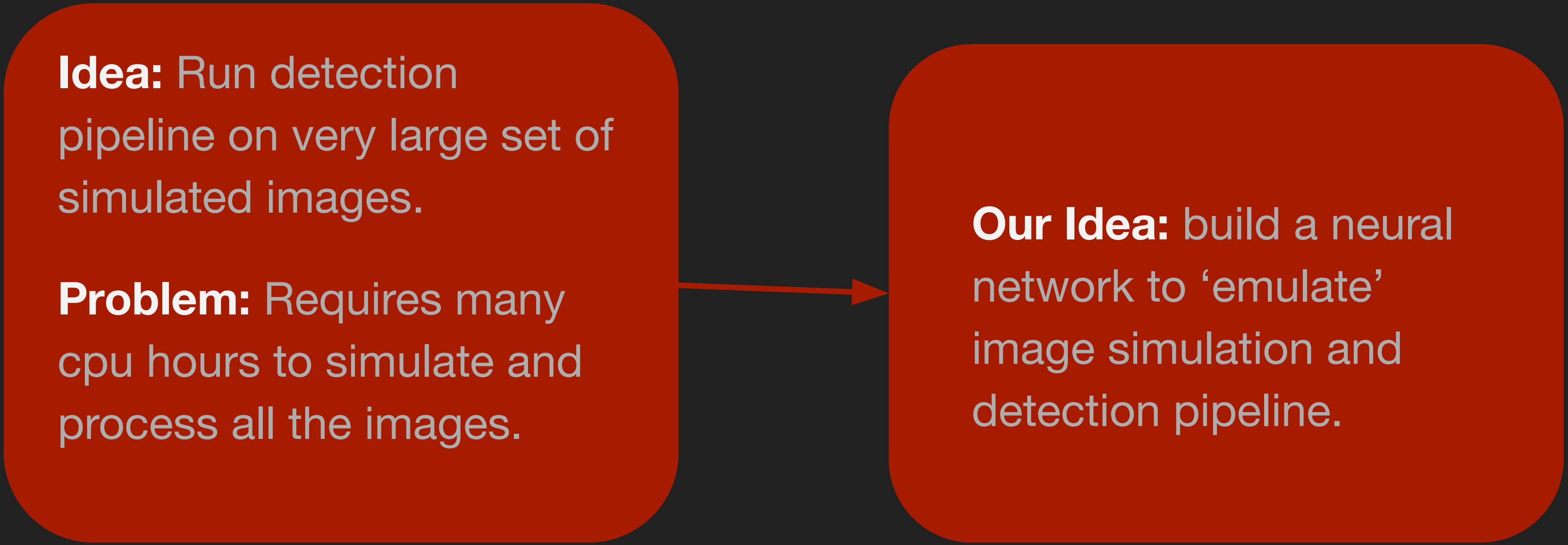
Blending induced detections

- Blending can introduce bias in measurements for weak lensing and large scale structure

Outline

1. Rubin Observatory/LSST overview
2. **“Emulation” project outline**
3. Data and model results
4. Next Steps
5. Questions

Question: How to estimate bias introduced by blending?



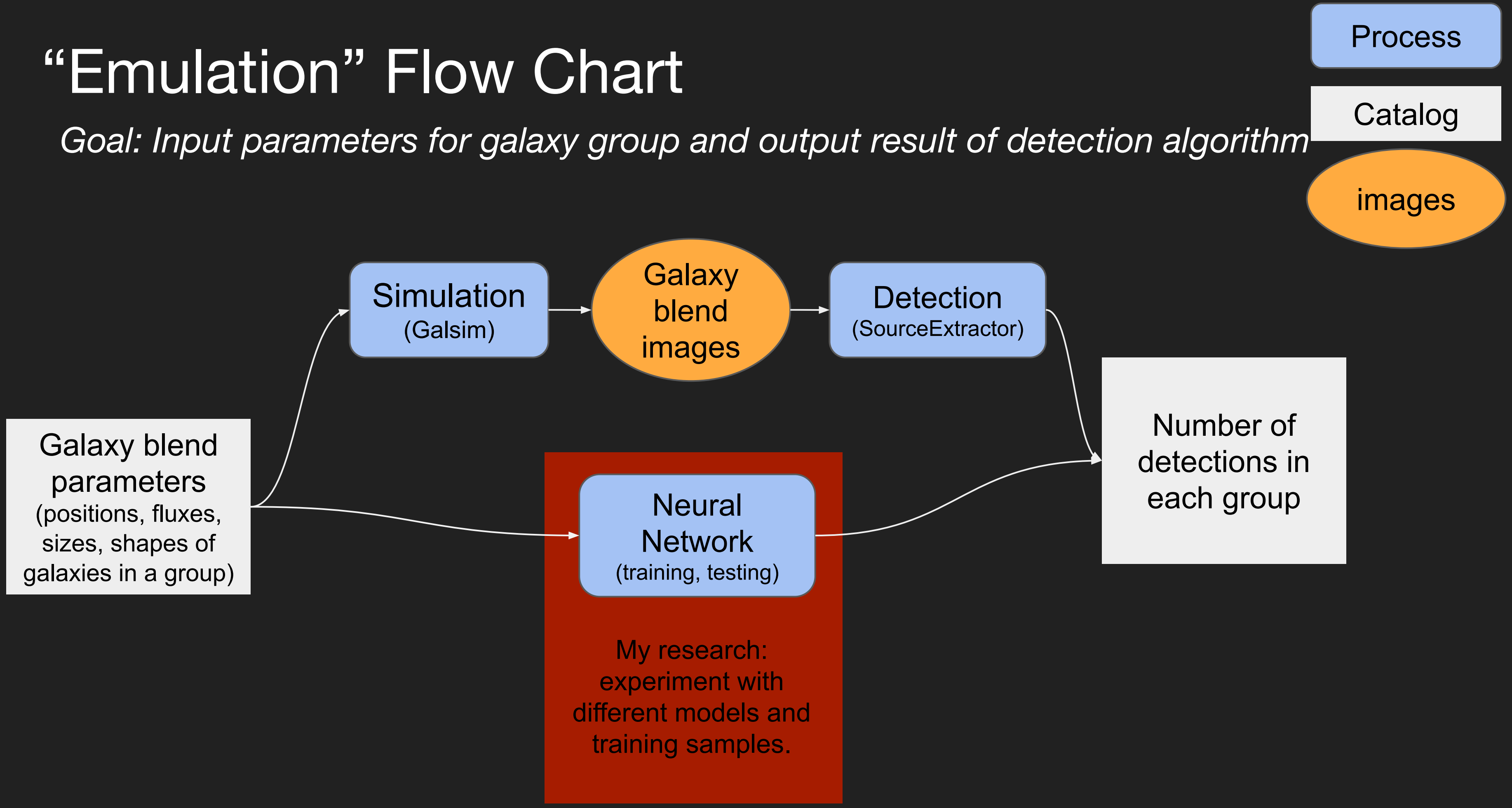
Idea: Run detection pipeline on very large set of simulated images.

Problem: Requires many cpu hours to simulate and process all the images.

Our Idea: build a neural network to ‘emulate’ image simulation and detection pipeline.

“Emulation” Flow Chart

Goal: Input parameters for galaxy group and output result of detection algorithm

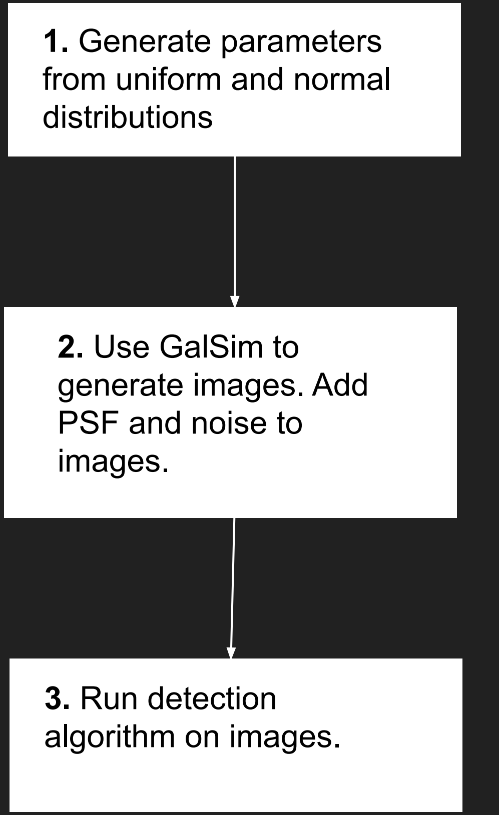

Process
Catalog
images
Galaxy blend parameters (positions, fluxes, sizes, shapes of galaxies in a group)
Simulation (Galsim)
Galaxy blend images
Detection (SourceExtractor)
Number of detections in each group
Neural Network (training, testing)

My research: experiment with different models and training samples.

Outline

1. Rubin Observatory/LSST overview
2. “Emulation” project outline
3. **Data and model results**
4. Next Steps
5. Questions

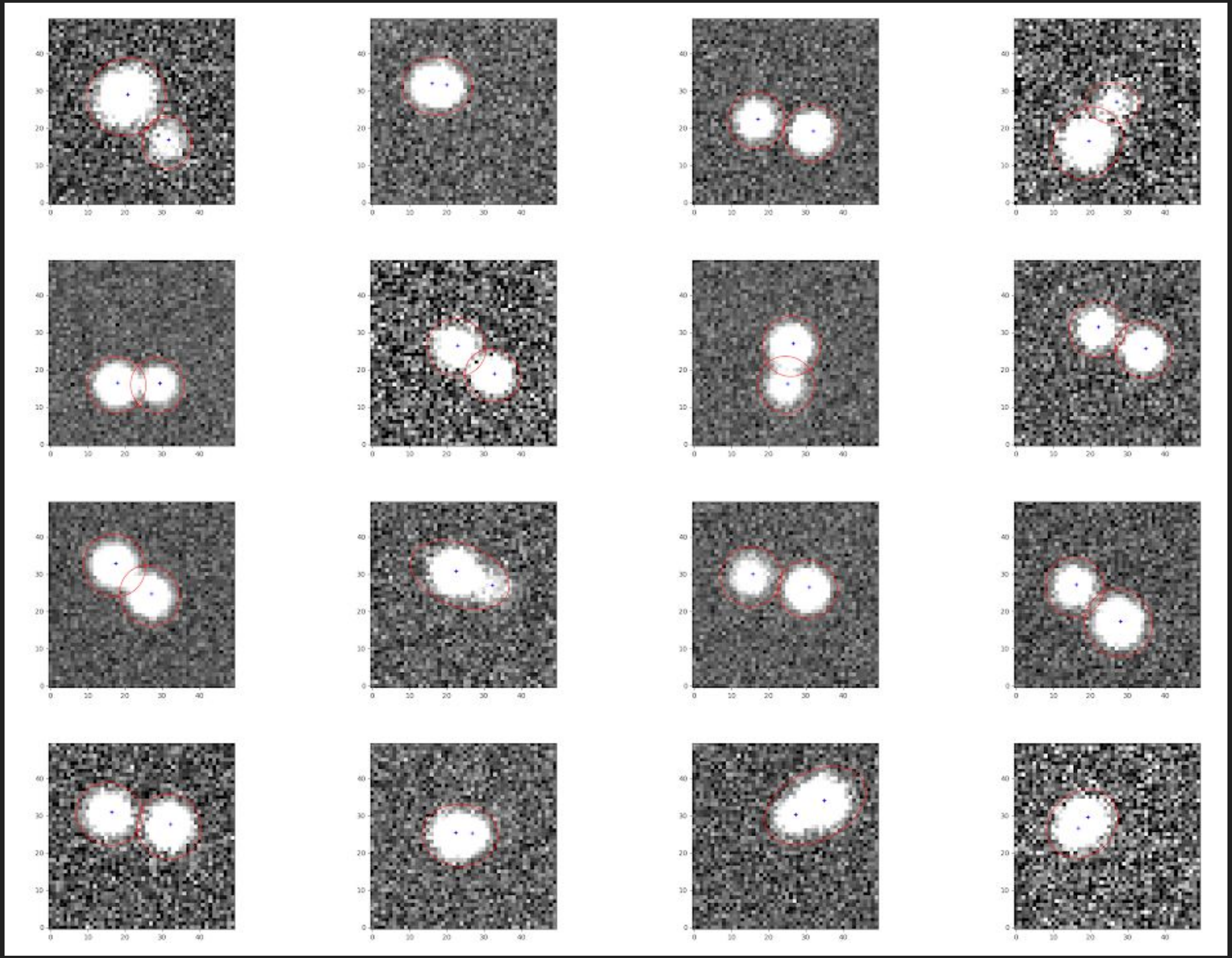
Training Data



1. Generate parameters from uniform and normal distributions

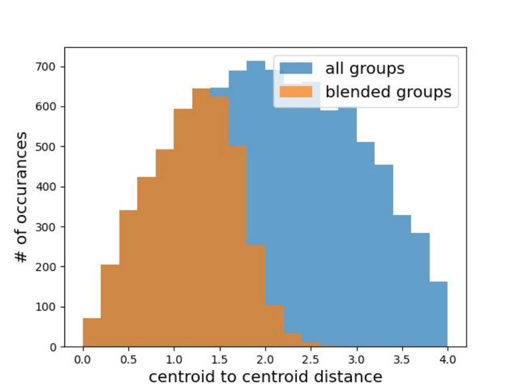
2. Use GalSim to generate images. Add PSF and noise to images.

3. Run detection algorithm on images.

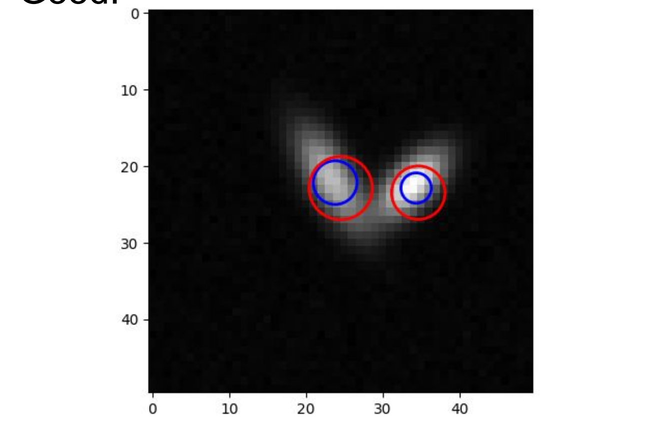


Binary classification

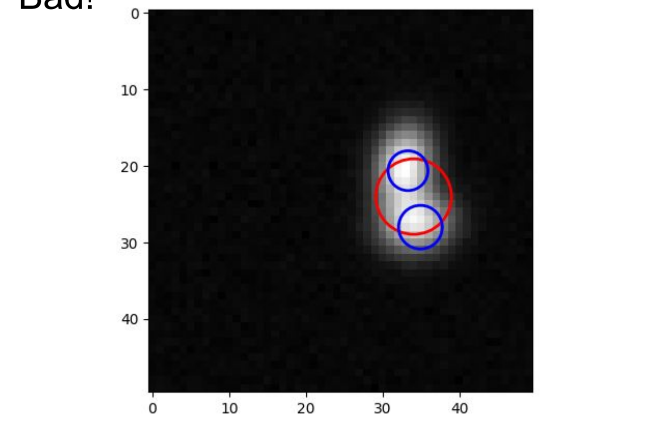
- Simple problem
- familiarize myself with PyTorch
- Output deblended (good) or unrecognized blend (bad)
- **97%** accuracy vanilla neural net
- **94%** accuracy using only euclidean distance



Good!



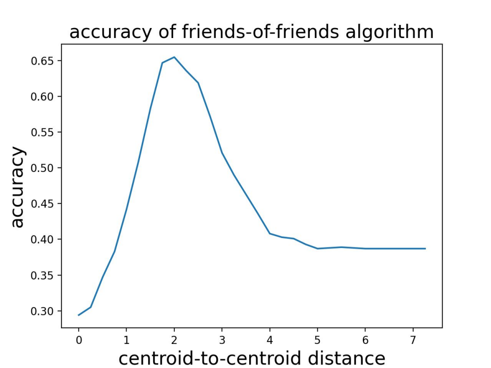
Bad!



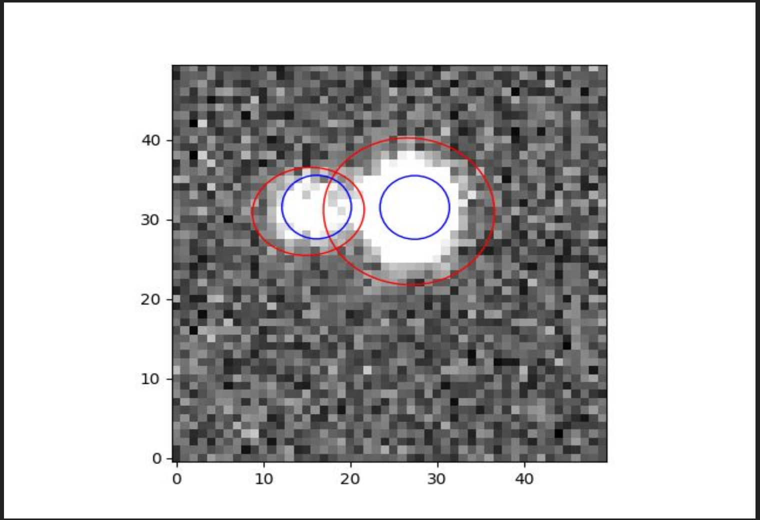
Multiclass classification

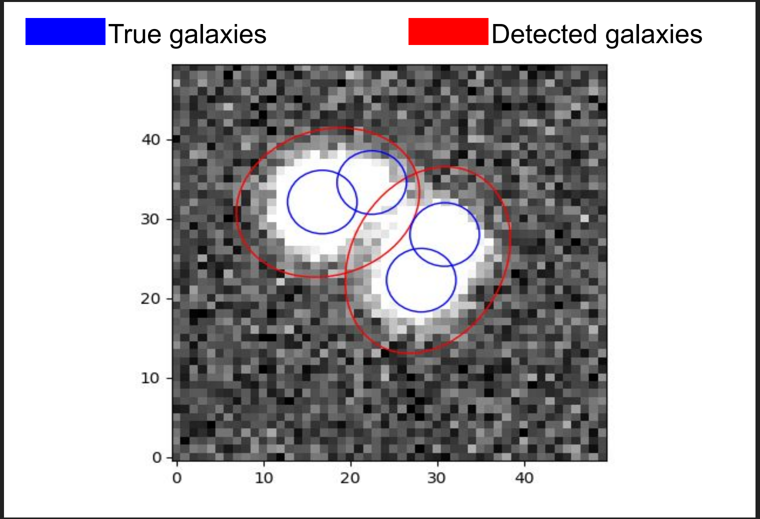
- 1-5 possible galaxies
- Predicts number detected
- Simple architecture
- More complex problem
- **85%** accuracy

65%
Accuracy (classical distance based algorithm)



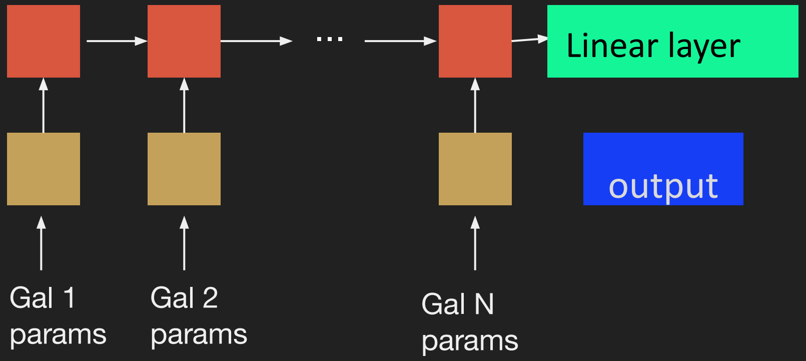
- Padding concern (neural net takes in constant dimension input)



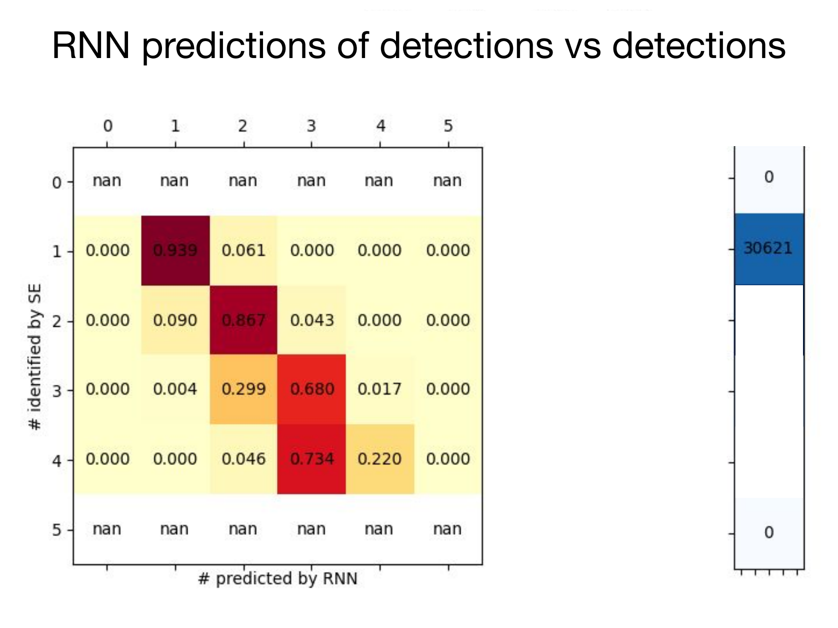


Multiclass classification - recurrent neural net

- RNN solved padding concern
- Variable sized input
- 87% accuracy
- Model assumes sequential order

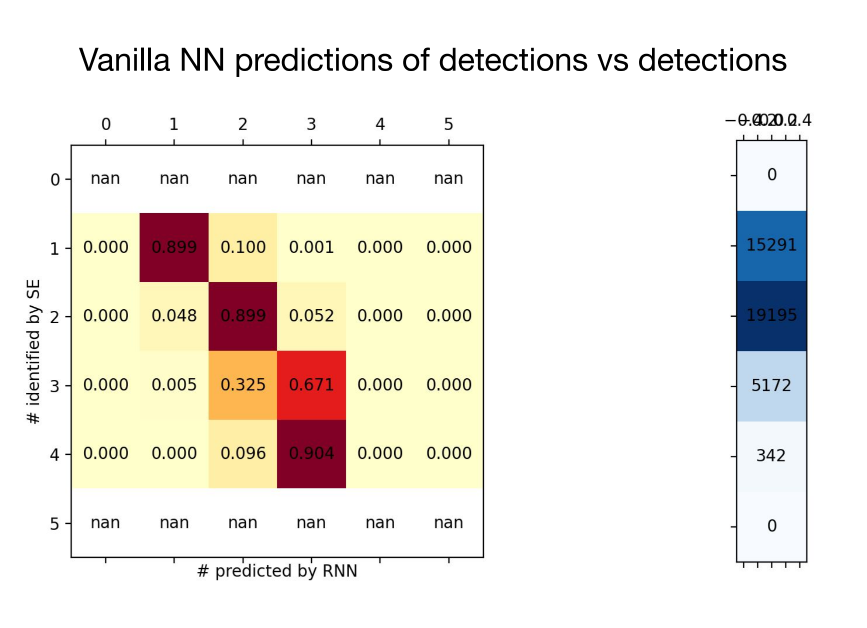


Accuracy matrices



RNN predictions of detections vs detections

# identified by SE \ # predicted by RNN	0	1	2	3	4	5
0	nan	nan	nan	nan	nan	nan
1	0.000	0.939	0.061	0.000	0.000	0.000
2	0.000	0.090	0.867	0.043	0.000	0.000
3	0.000	0.004	0.299	0.680	0.017	0.000
4	0.000	0.000	0.046	0.734	0.220	0.000
5	nan	nan	nan	nan	nan	nan



Vanilla NN predictions of detections vs detections

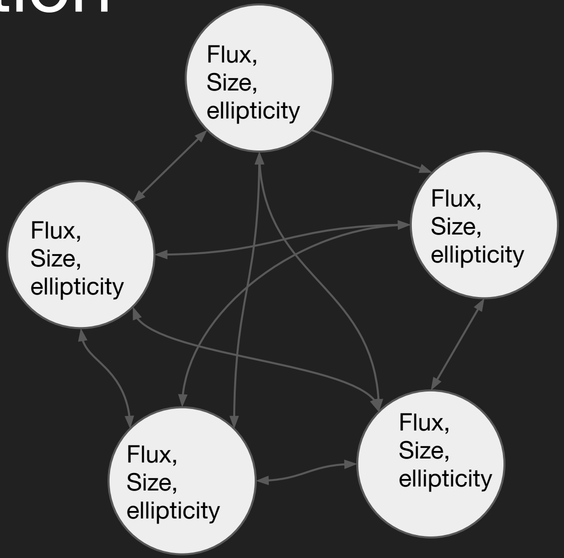
# identified by SE \ # predicted by RNN	0	1	2	3	4	5
0	nan	nan	nan	nan	nan	nan
1	0.000	0.899	0.100	0.001	0.000	0.000
2	0.000	0.048	0.899	0.052	0.000	0.000
3	0.000	0.005	0.325	0.671	0.000	0.000
4	0.000	0.000	0.096	0.904	0.000	0.000
5	nan	nan	nan	nan	nan	nan

RNN: 87%

Vanilla NN: 85%

Graph Neural Network - Classification

- Idea: model for neural net should reflect our knowledge about the problem
- Assumes permutation invariance



↕ weight=distance
between centroids

Roadmap

1. Rubin Observatory/LSST overview
2. “Emulation” project outline
3. Data and model results
4. **Next Steps**
5. Questions

Next steps

- Continue work on the graph based neural network
- Ultimately use neural net to measure the bias of unrecognized blends on two point correlation functions

Roadmap

1. Rubin Observatory/LSST overview
2. “Emulation” project outline
3. Data and model results
4. Next Steps
5. **Questions**