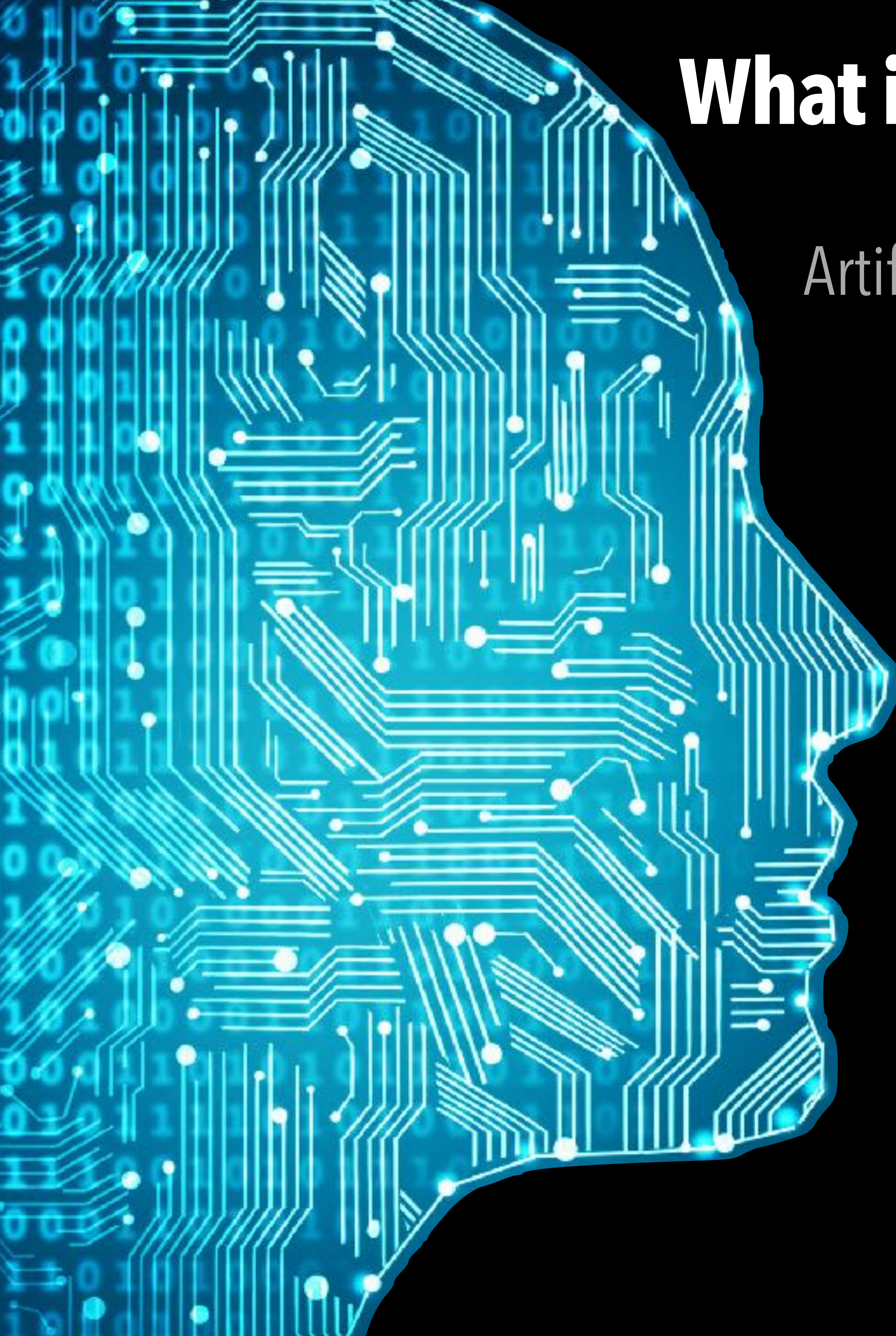
Harnessing Machine Learning for the Classification, Identification and Modeling of Astronomical Data

Stella Offner
The University of Texas at Austin
Center for Scientific Machine Learning, Co-Director

Work is supported by:

Duo Xu (UVA), Rob Gutermuth (UMass), Josh Taylor (UT Austin), Keith Poletti (UT Austin), Rachel Ward (UT Austin/Microsoft)

What is artificial intelligence?



Artificial Intelligence:: The science and engineering of making intelligent machines.

– John McCarthy, 1955

Machine Learning:: Field of computer science that gives computers the ability to learn without being explicitly programmed.

– Arthur Samuel, 1959

Why machine learning?

MACHINE LEARNING IS THE FUTURE

Big Questions

1. How can we accelerate the adoption of machine learning methods to *effectively* address SF problems?
2. What ML techniques are the most effective for analysis tasks such as classifying objects, identifying structures, and predicting physical quantities?

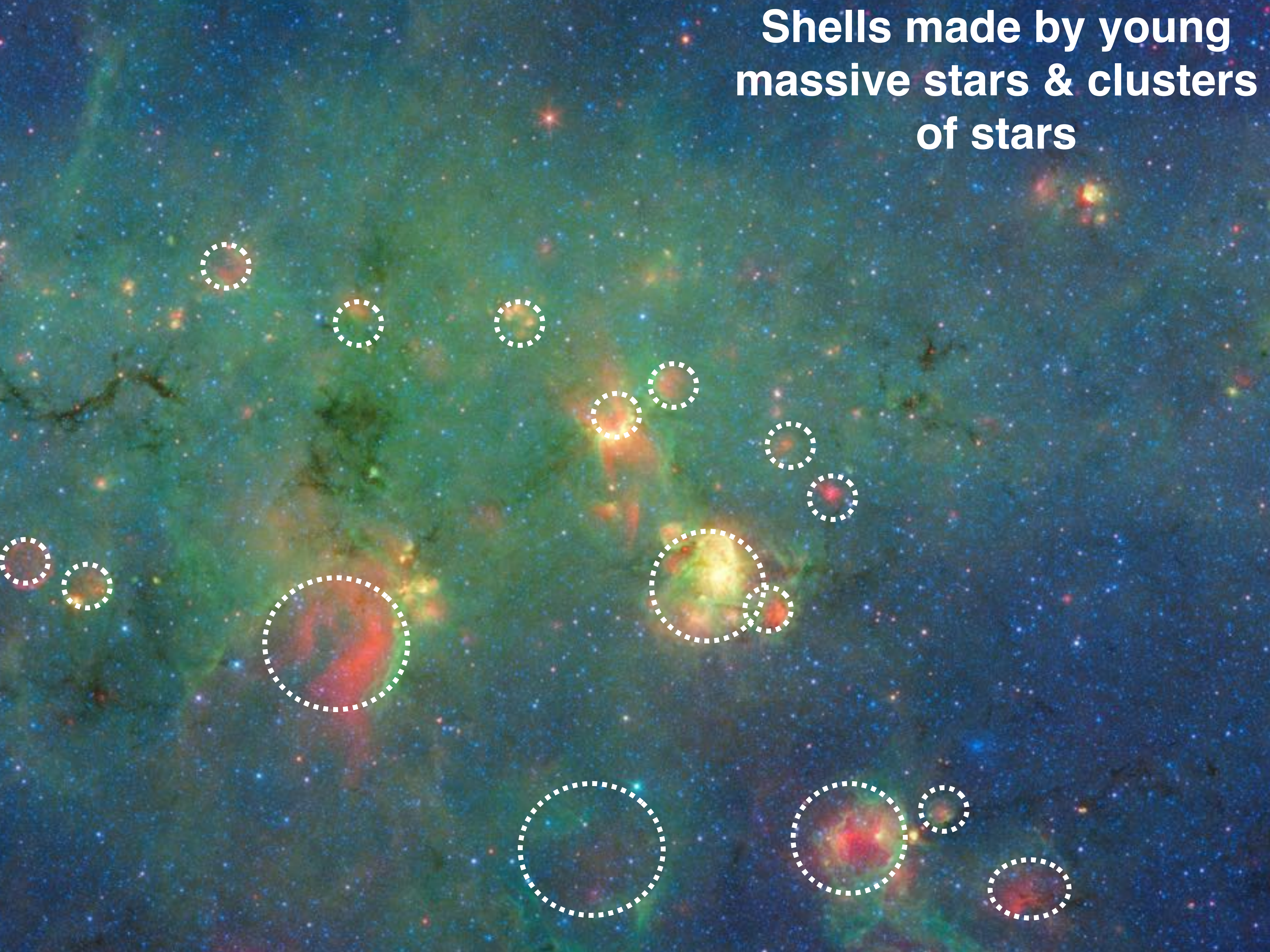
- **In low-mass SF regions … which feedback process dominates (e.g., jets, stellar winds, radiation…)?**
- **How do the radiation, winds, flows …. produced during the SF process affect the SF in the region?**
- **How much turbulence is injected by the jets and outflows into the SF region?**
- **Are there cores that have not formed in a filament?**
- **How important is core collision in the overall math of SF?**
- **Is it possible to destroy cores (e.g. by large-scale shearing motions) before they can create a star?**
- **What causes the core collapse? Which processes stabilize the core?**
- **When & how do gravity, B fields, radiation, and turbulence impact the formation & evolution of MCs?**
- **What is the fraction of MCs that undergo gravitational collapse?**

Big Problems

1. There is a lot of data!
2. The data is high-dimensional!
3. The data is messy, noisy, and complex!
4. We observe photons … not physical variables!
5. Evolutionary timescales are long (simulations are slow)!

Problem 1: There is a lot of data ....

Goal: to classify and identify features in data

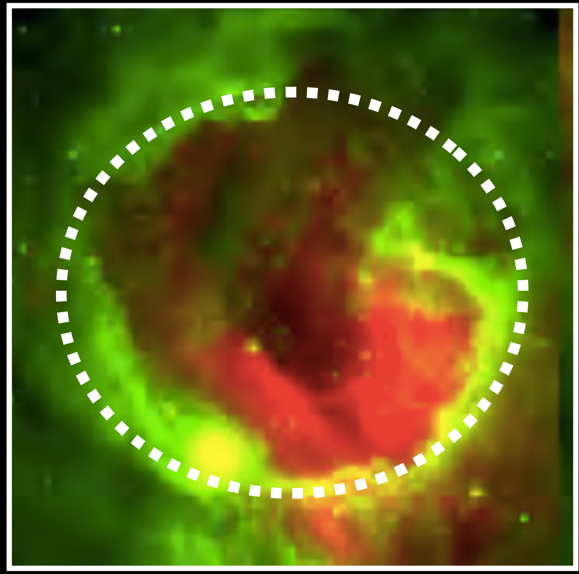

Shells made by young
massive stars & clusters
of stars

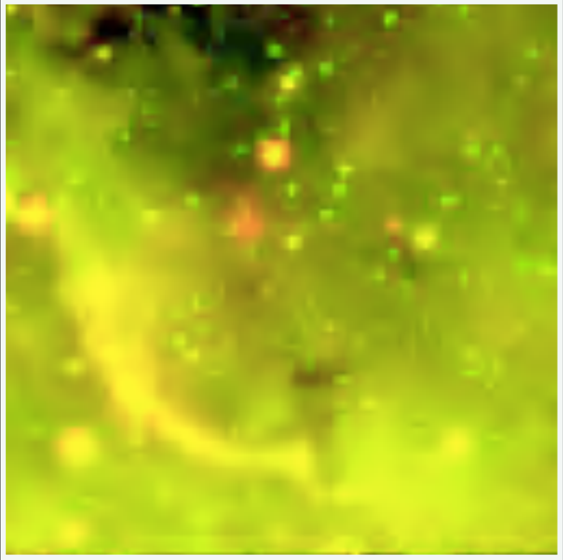
Classification

Sorting Data

Citizen science is powerful …

✔





?

DR2: Jayasinghe + 2019

Citizen science is powerful …

Benefits

1. Engage the public in Science!
2. Numerous!
3. Free!
4. Can identify atypical cases!

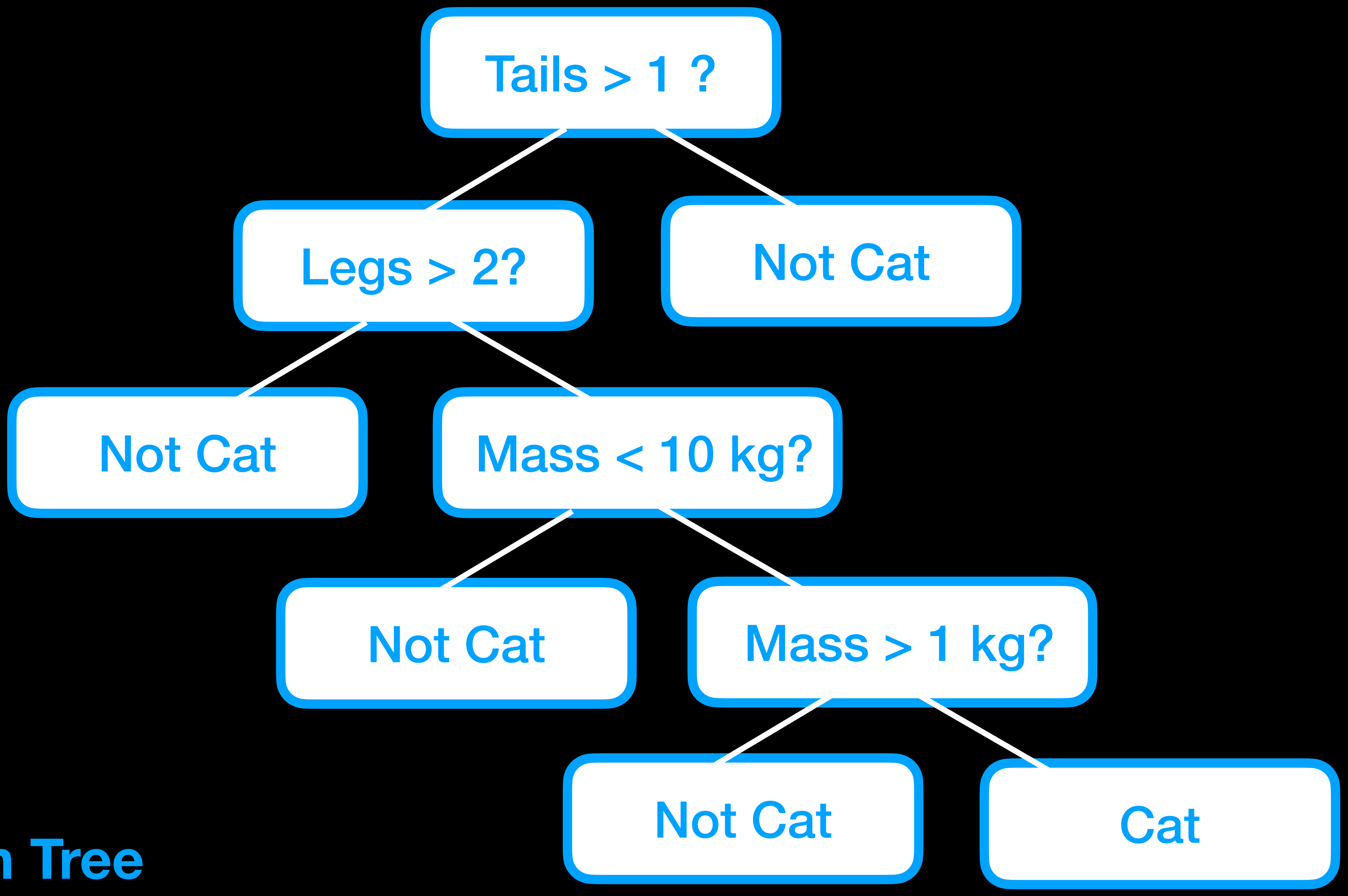
Problems:

1. Different opinions.
2. Need simple instructions.
3. Can be hangry.

Even experts don’t know the “right” answer (“Ground Truth”).

Object Classification: Random Forest

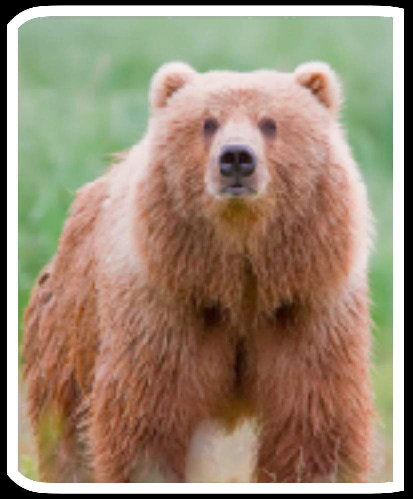
- Method to identify and sort objects in a sample (introduced in 1995!)
- Works well on vectorized data/images

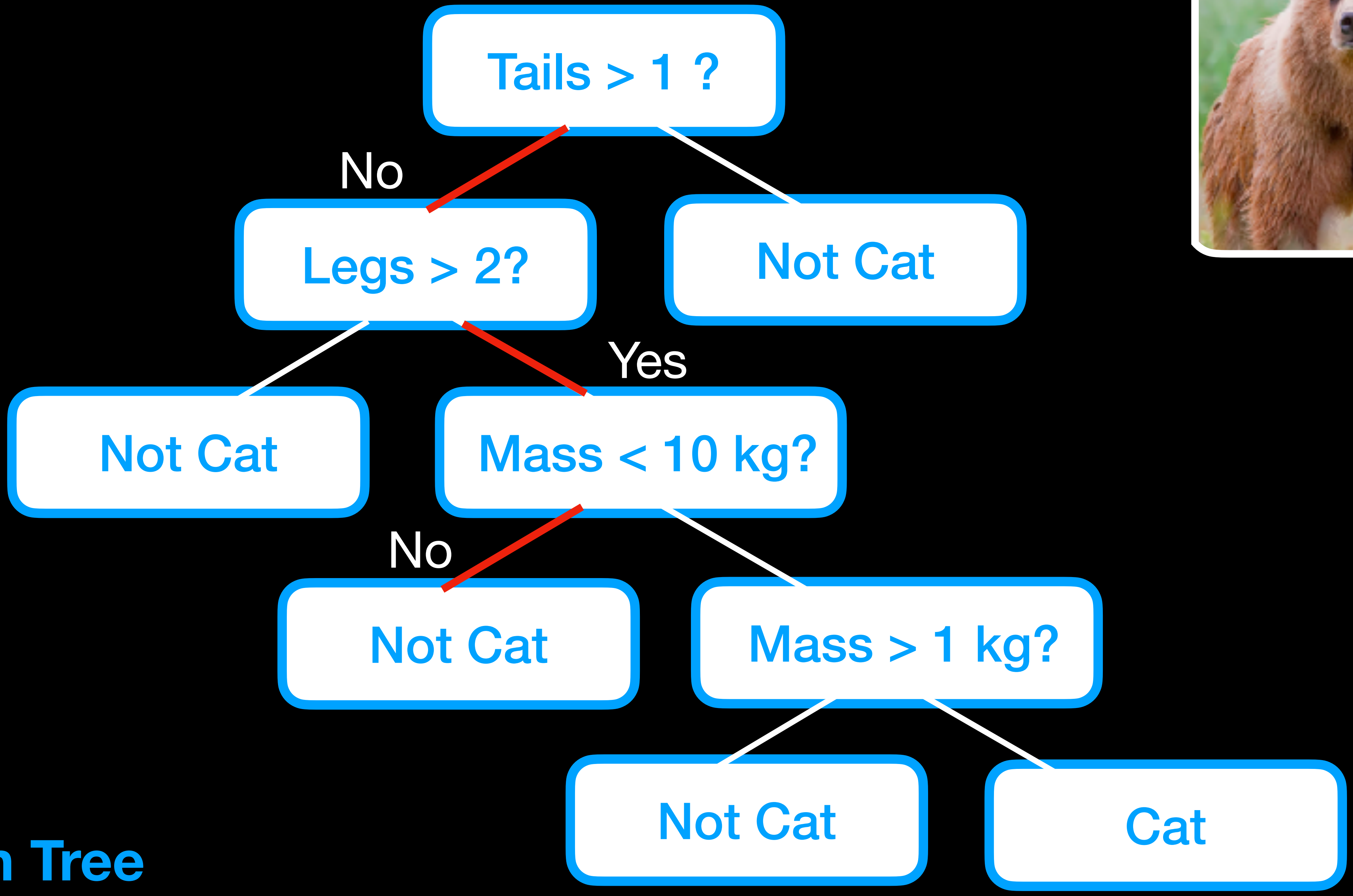


Decision Tree

Object Classification: Random Forest

- Method to identify and sort objects in a sample
- Works well on vectorized data/images

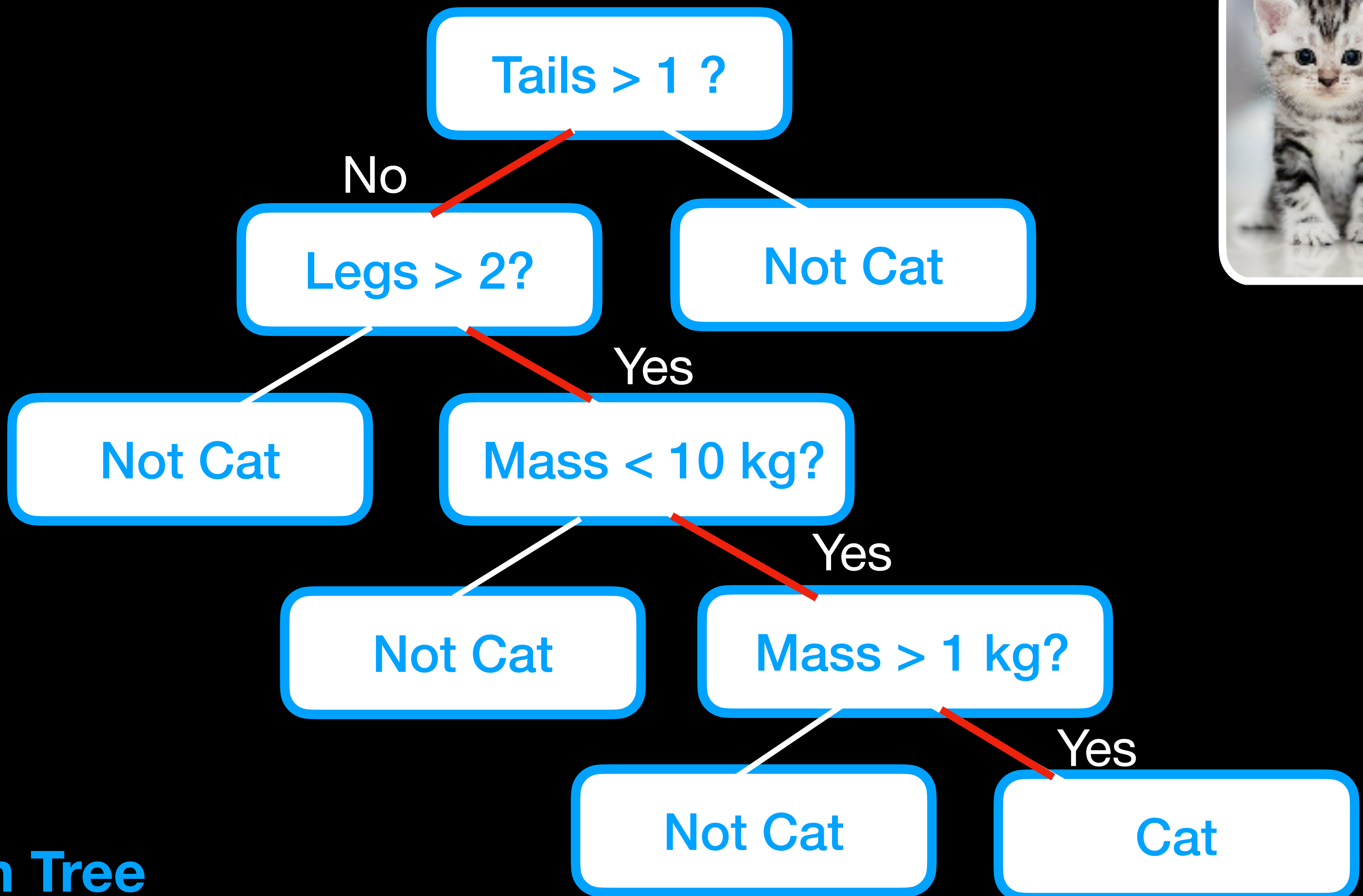




Decision Tree

Object Classification: Random Forest

- Method to identify and sort objects in a sample
- Works well on vectorized data/images

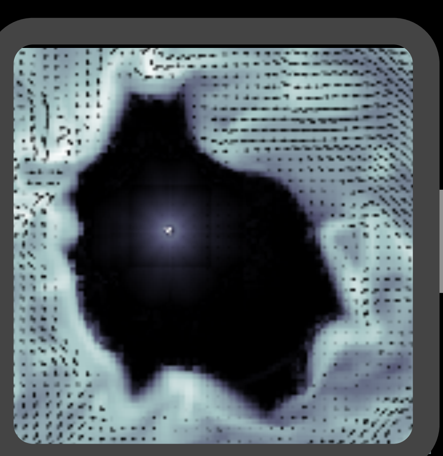


Decision Tree

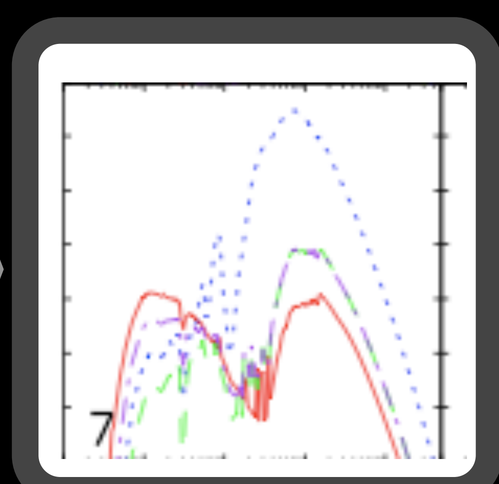
Training Astronomy ML Methods

- Learning the classification requires training data – TRUE answer is known
- This data is used to set the free parameters of the method
- Thousands or millions of examples are often required
- Size of training set needed depends on problem complexity

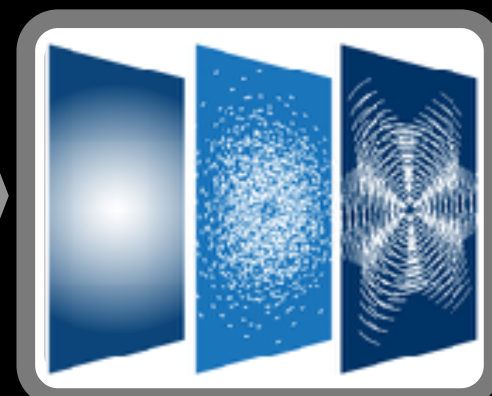
Simulated bubbles from winds



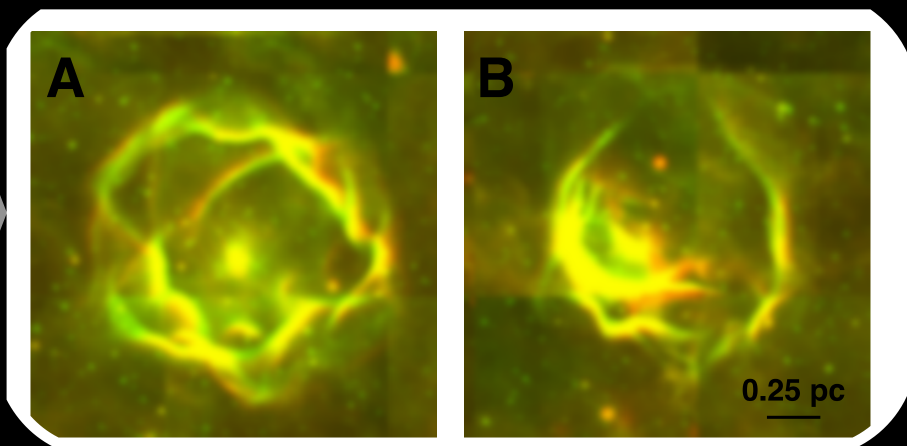
Model light emitted



"Observe" assuming a distance + noise



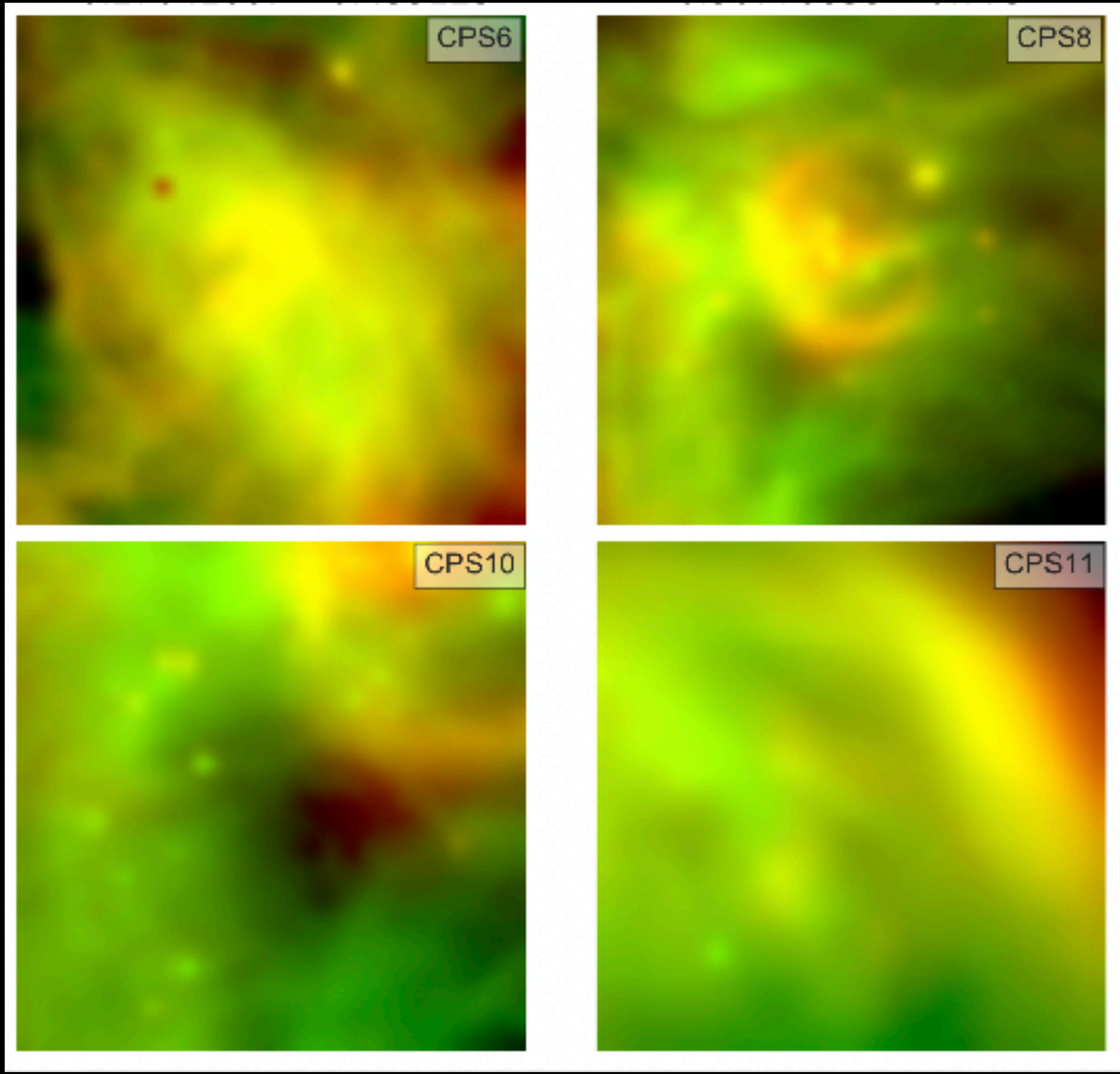
Training Set "Mock Observations"



New Bubbles!

- Training on simulations increases ability to detect some types of bubbles

Xu & Offner 2017



Bubbles previously missed when training data uses only "by-eye" detections

Scikit-learn Example

```
>>> from sklearn.ensemble import RandomForestClassifier
>>> from sklearn.datasets import make_classification
>>> X, y = make_classification(n_samples=1000, n_features=4,
...                            n_informative=2, n_redundant=0,
...                            random_state=0, shuffle=False)
>>> clf = RandomForestClassifier(max_depth=2, random_state=0)
>>> clf.fit(X, y)
RandomForestClassifier(...)
>>> print(clf.predict([[0, 0, 0, 0]]))
[1]
```

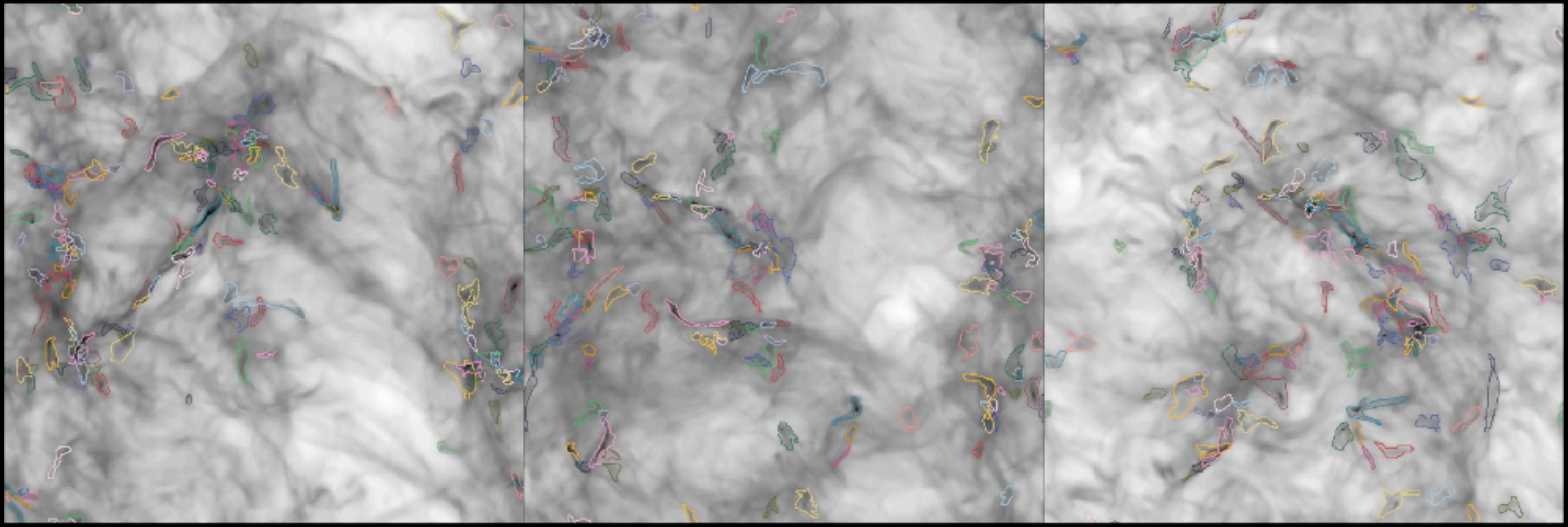
https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html#sklearn.ensemble.RandomForestClassifier

Other examples: Beaumont et al. 2014 (Bubbles); Gomez et al. 2020, 2023 (SNe)

Summary Problem I: Big Data

- “Classic” ML technique, Random Forests, provides a **reliable, fast** way to classify astronomical data.
- Can be used to classify data vectors (e.g., photometric and spectroscopic data of bubbles, SNe)
- Relatively **easy to implement** in python: scikit-learn

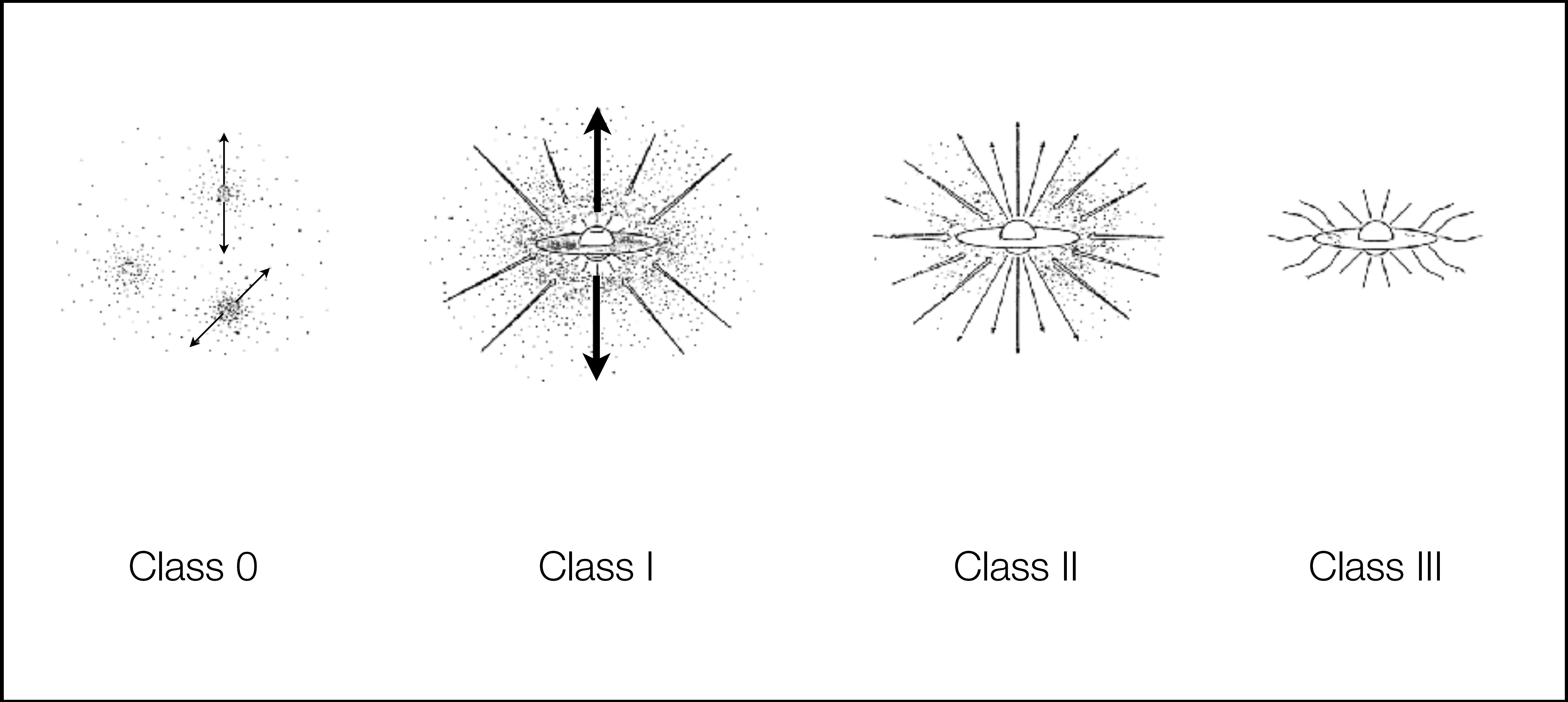
Problem 2: Data is high-dimensional



Gas properties are complex.
Simple descriptions (mass, viral parameter) miss the big picture.

Protostellar Evolution

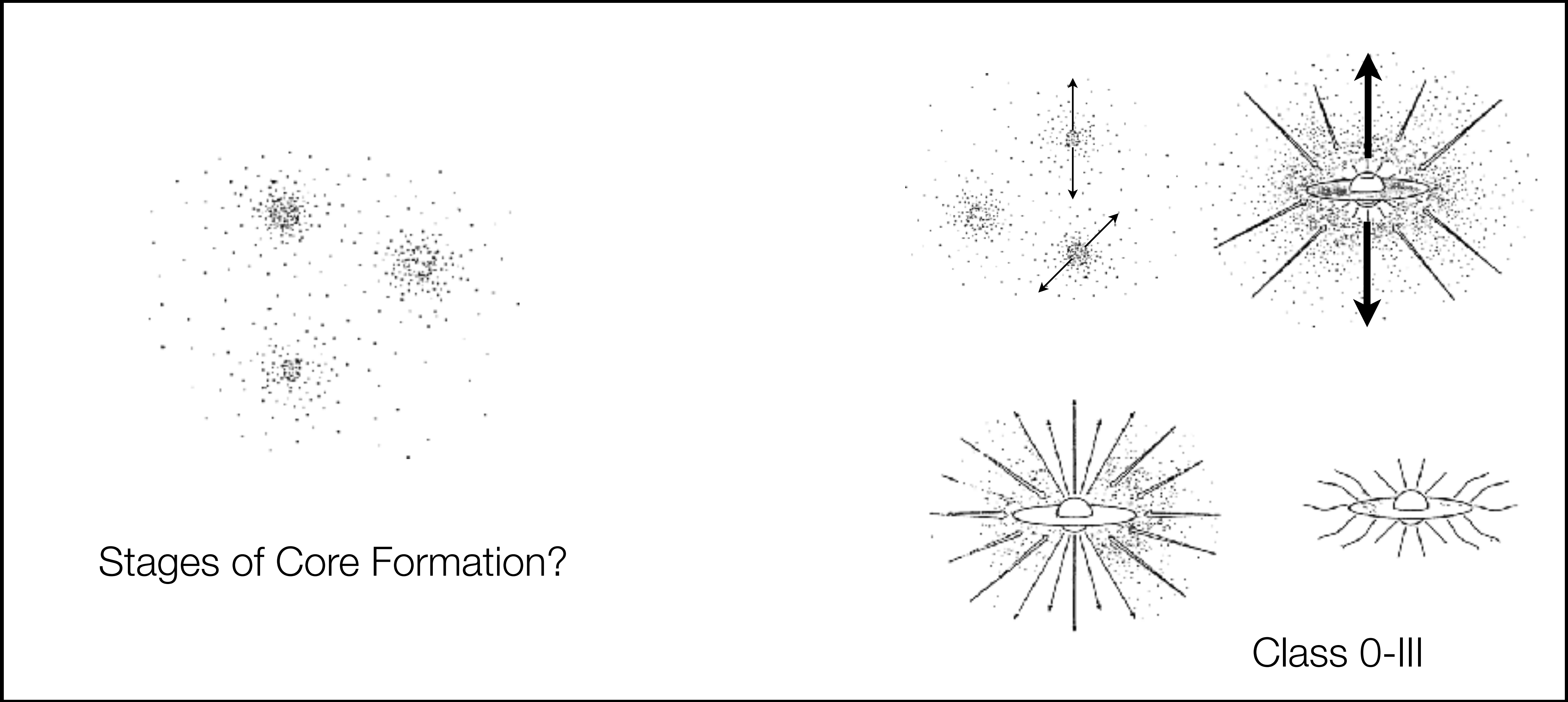
Classic Stages of *Star* Formation



Shu ea 1987

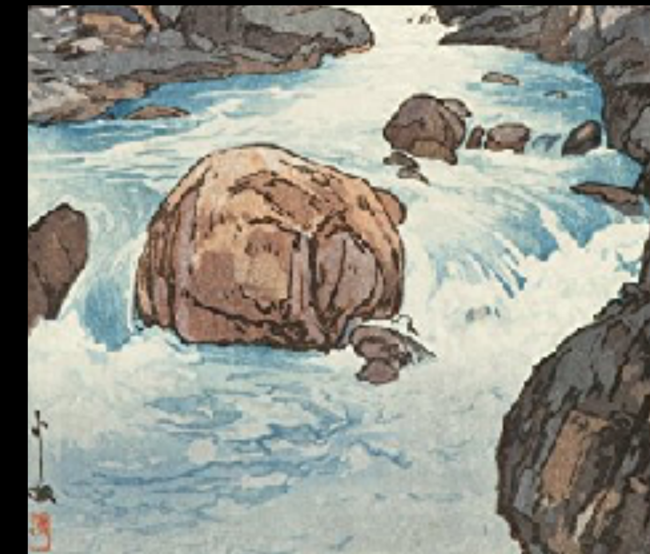
Prestellar Evolution?

Classic Stages of *Star* Formation

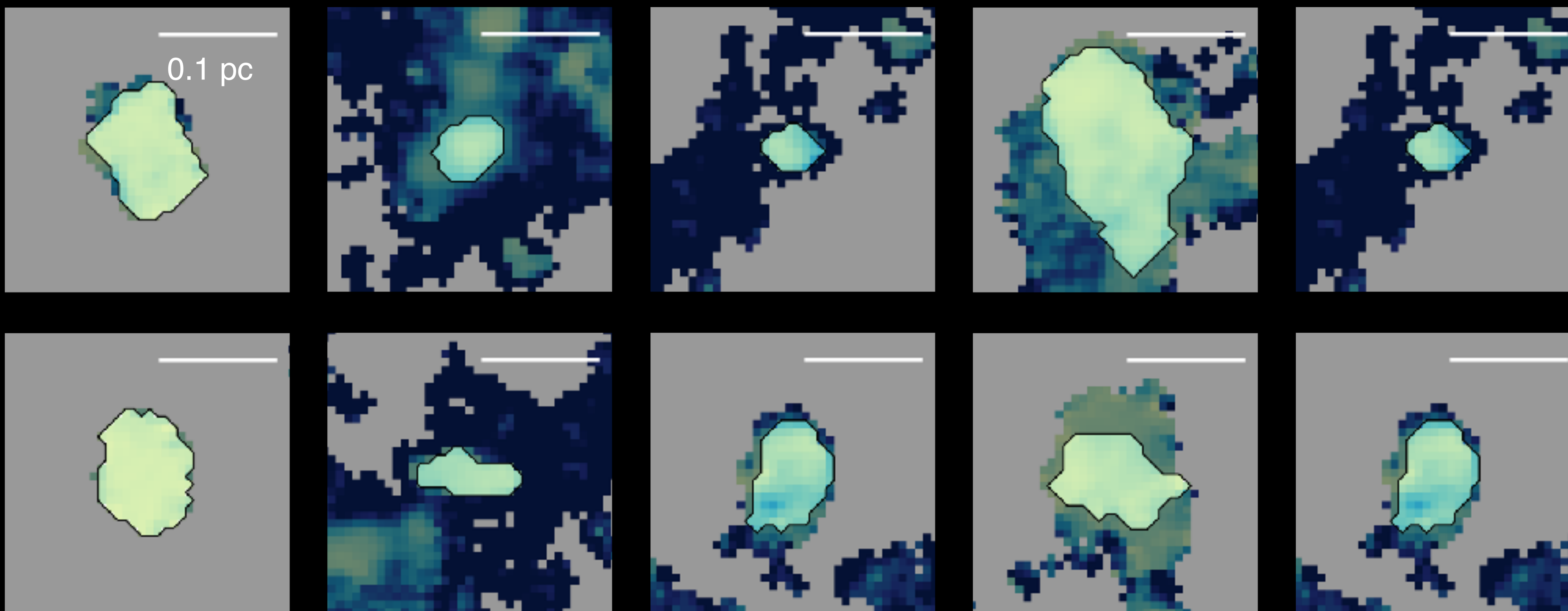


Terminology: Dense Core, Starless Core, Prestellar Core, Protostellar Core, Droplet, Gravitationally-Bound, Pressure-Confined, Coherent Core

Droplets: A new type of core



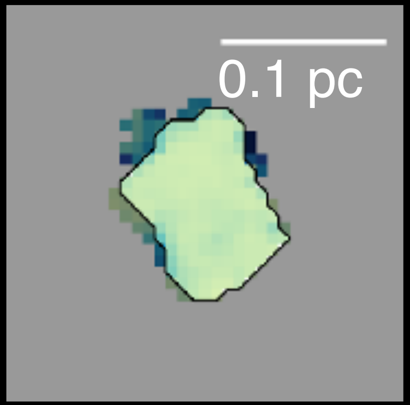
NH3 Velocity dispersion

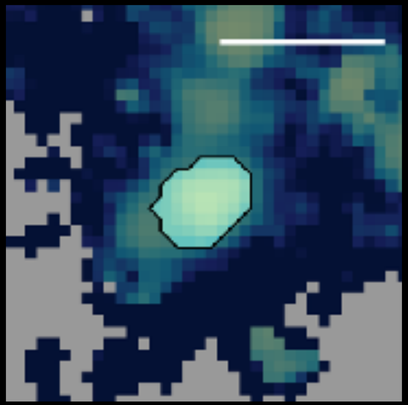


H. Chen, Goodman et al. (2019)

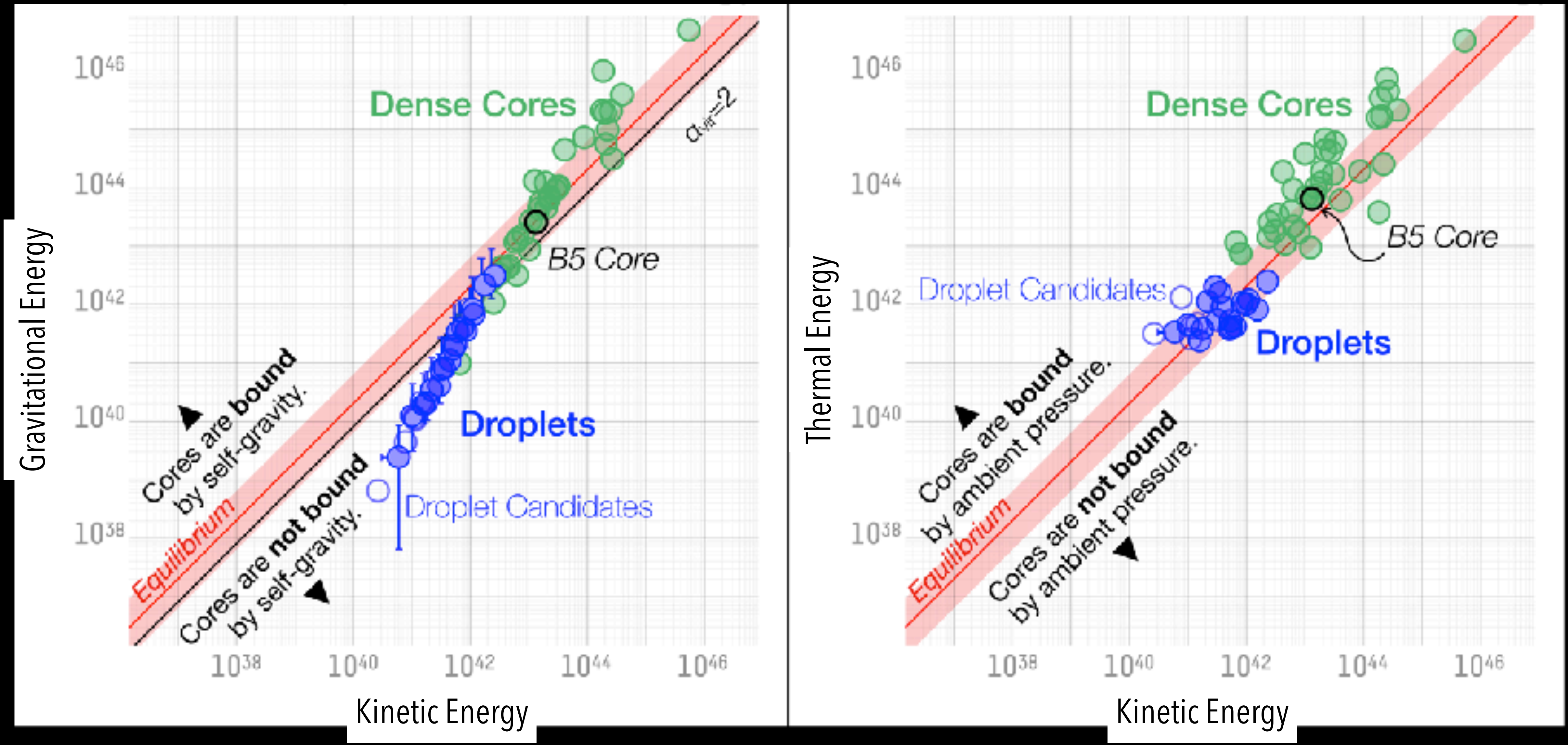
Small starless quiescent structures, likely bound by external pressure.

Droplets: A new type of core





H. Chen, Goodman et al. (2019)

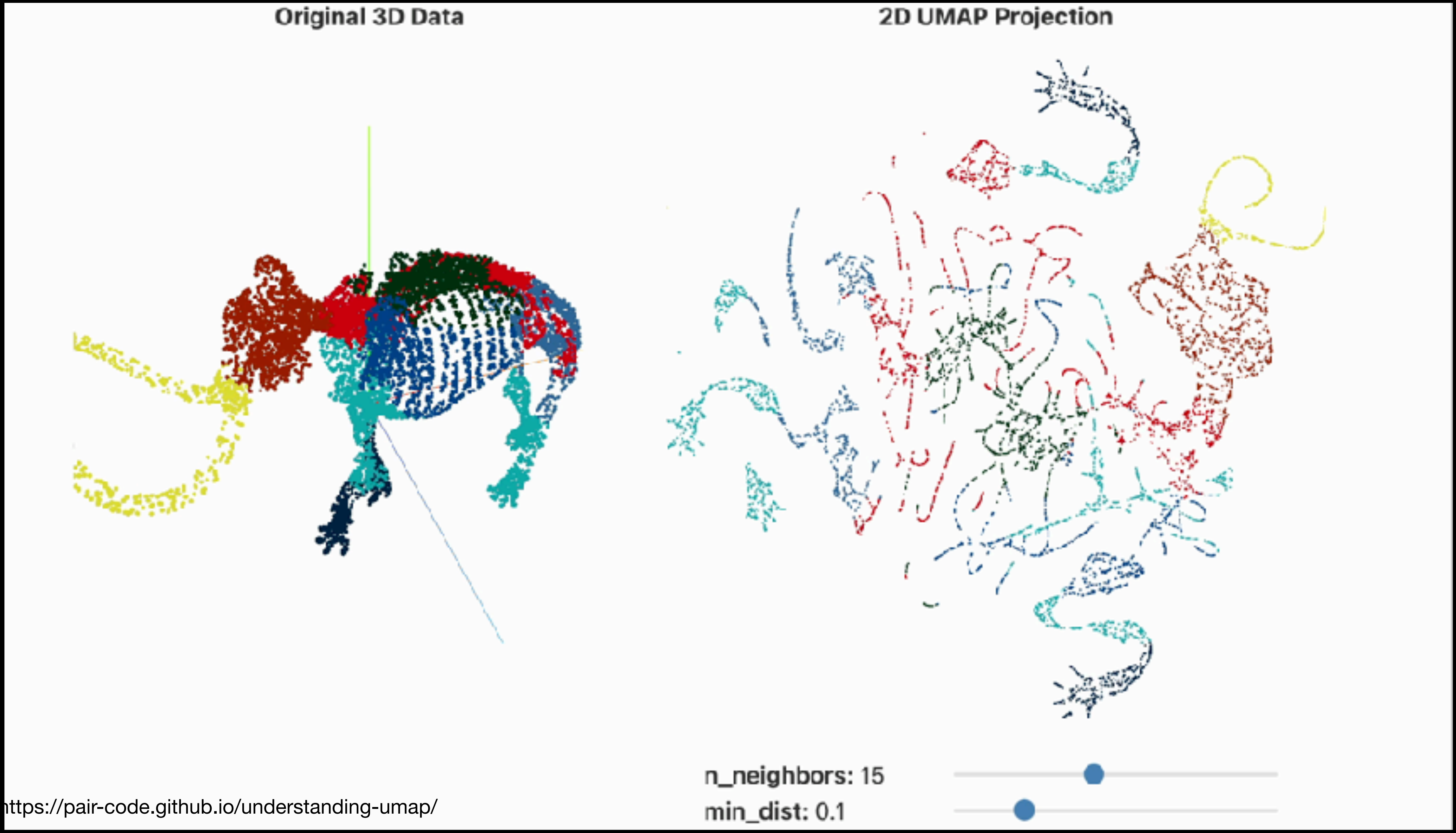


Small starless quiescent structures, likely bound by external pressure.

Classification

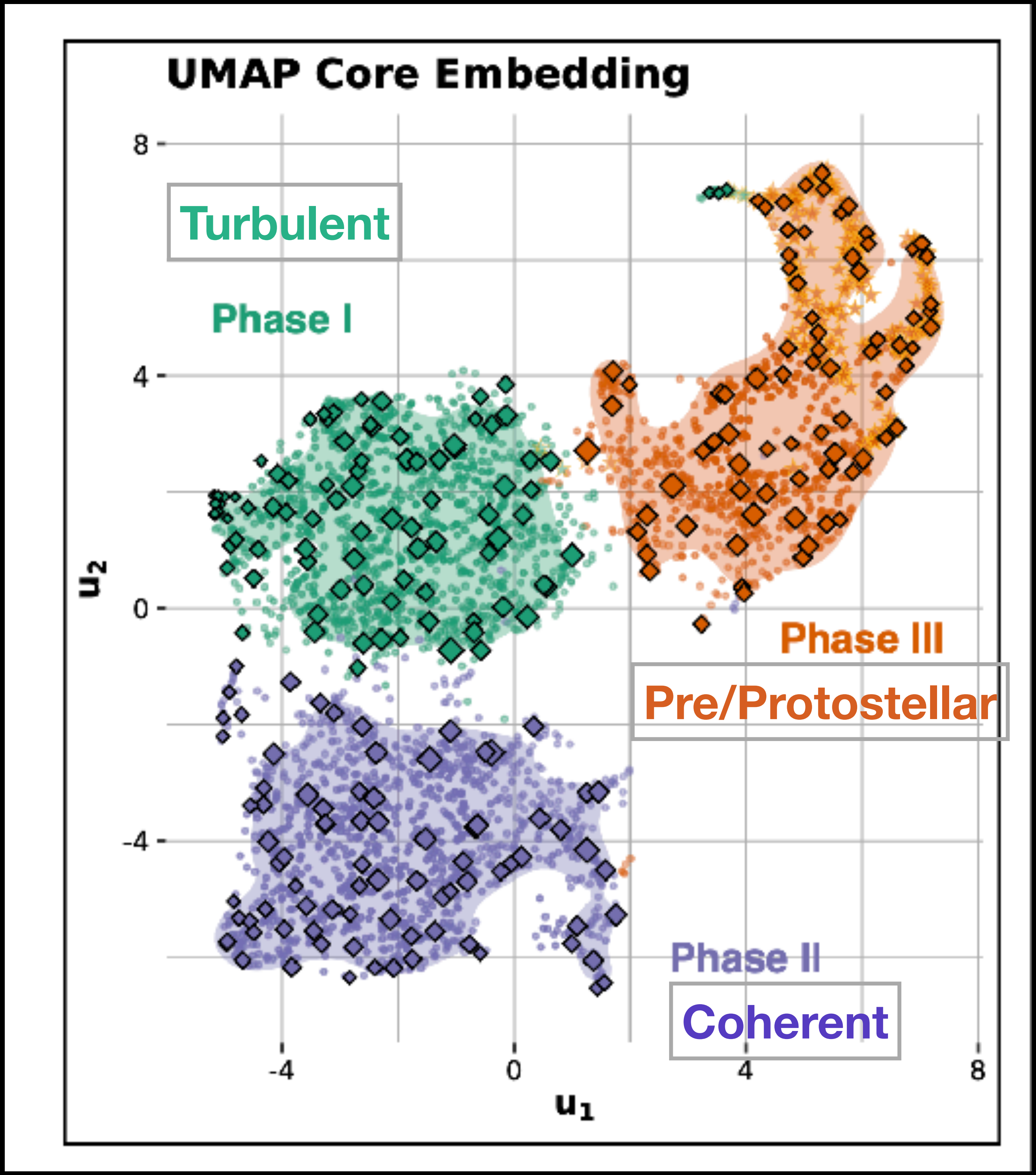
Via Clustering and Data Exploration

UMAP Uniform Manifold Approximation for Dimension Reduction (McInnes et al. 2018)

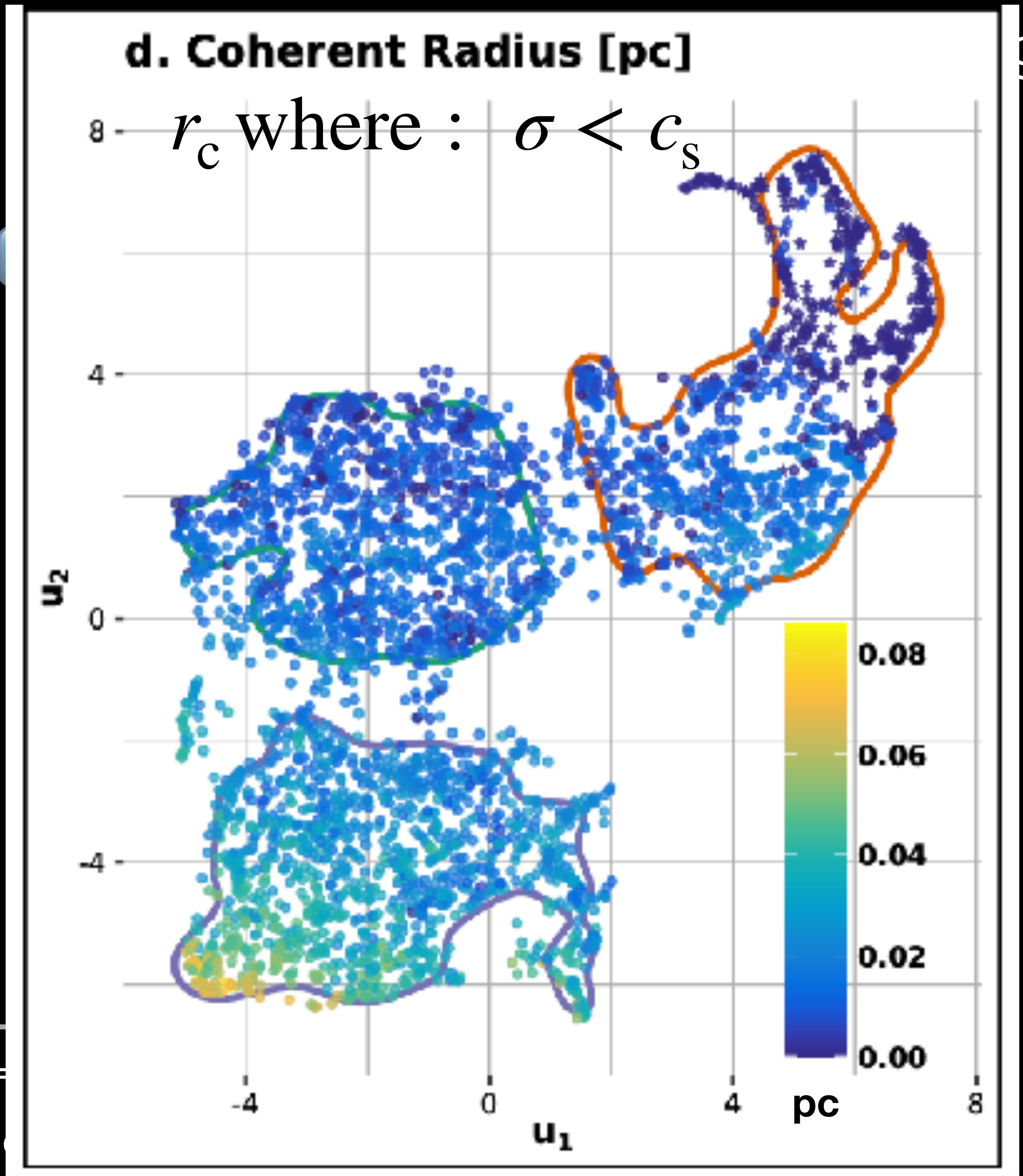


- Builds a graph representation in high-d space and optimizes a low-d graph to be as similar as possible
- Like t-SNE (t-stochastic neighbor embedding) but more computationally efficient for high-d, better at preserving distances in low-d

Three core stages:
turbulence, coherence, collapse

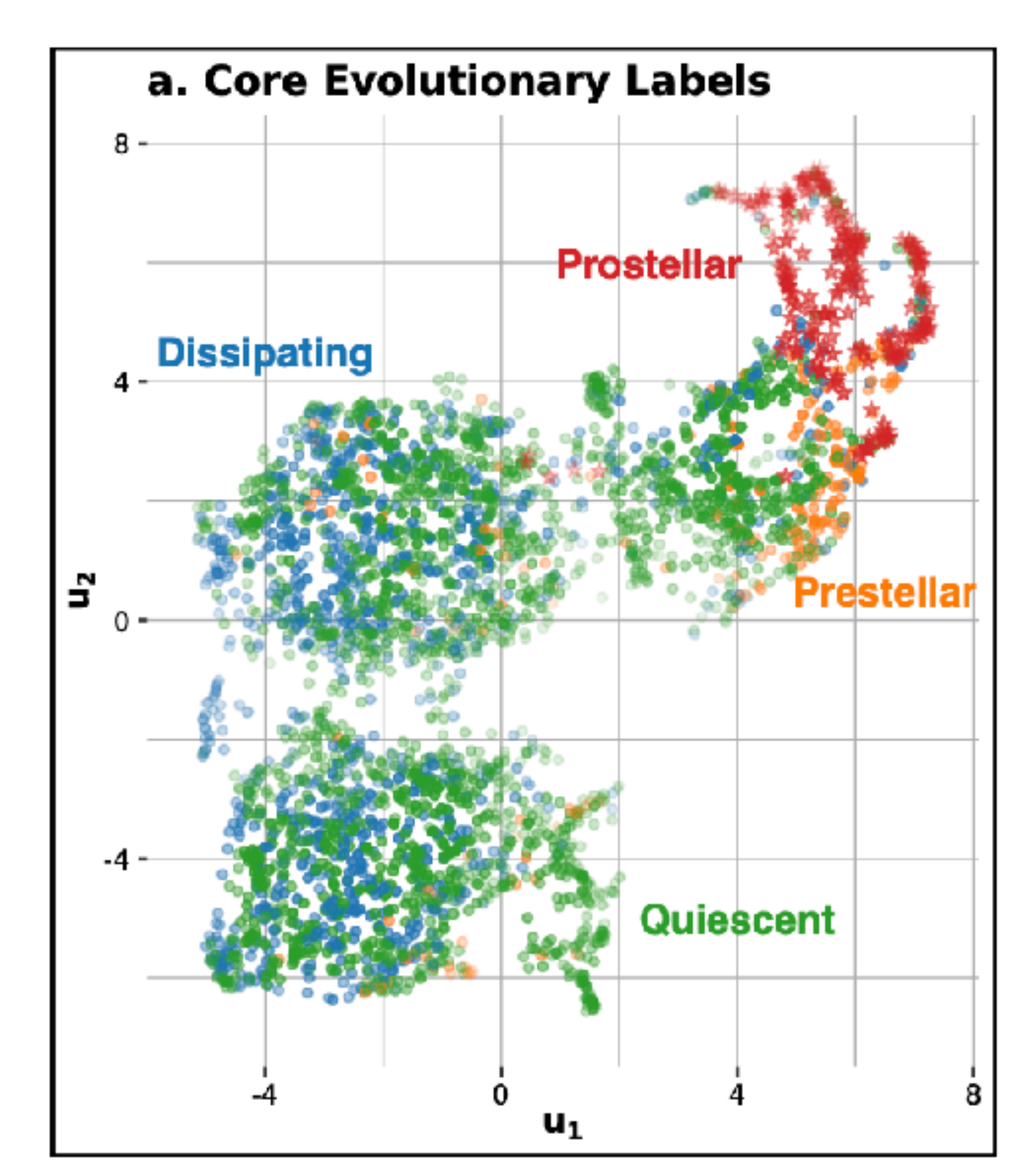


- Data vector: density + velocity profiles, core mass, vel. Dispersion, radius of coherence, radius, viral parameter
-
-



Offner et al. 2022

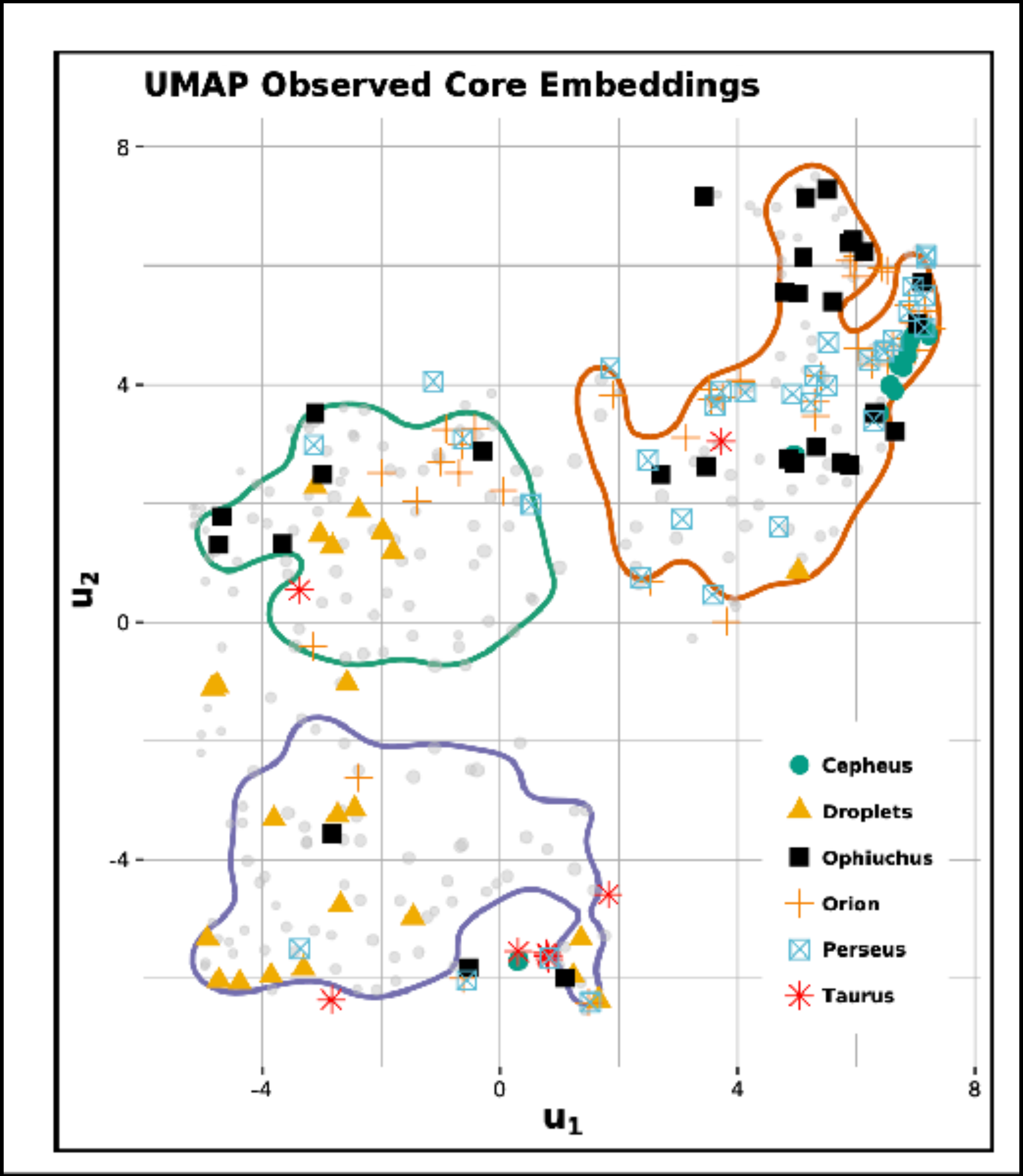
Three core outcomes:
dispersing, quiescent, pre/protostellar



- Color UMAP by evolution: those that disperse, long-lived (quiescent) and pre/protostellar
- 55% belong to 2 or more phases
- **Cores are stochastic:** Evolutionary properties do not predict well the final outcome

Offner et al. 2022

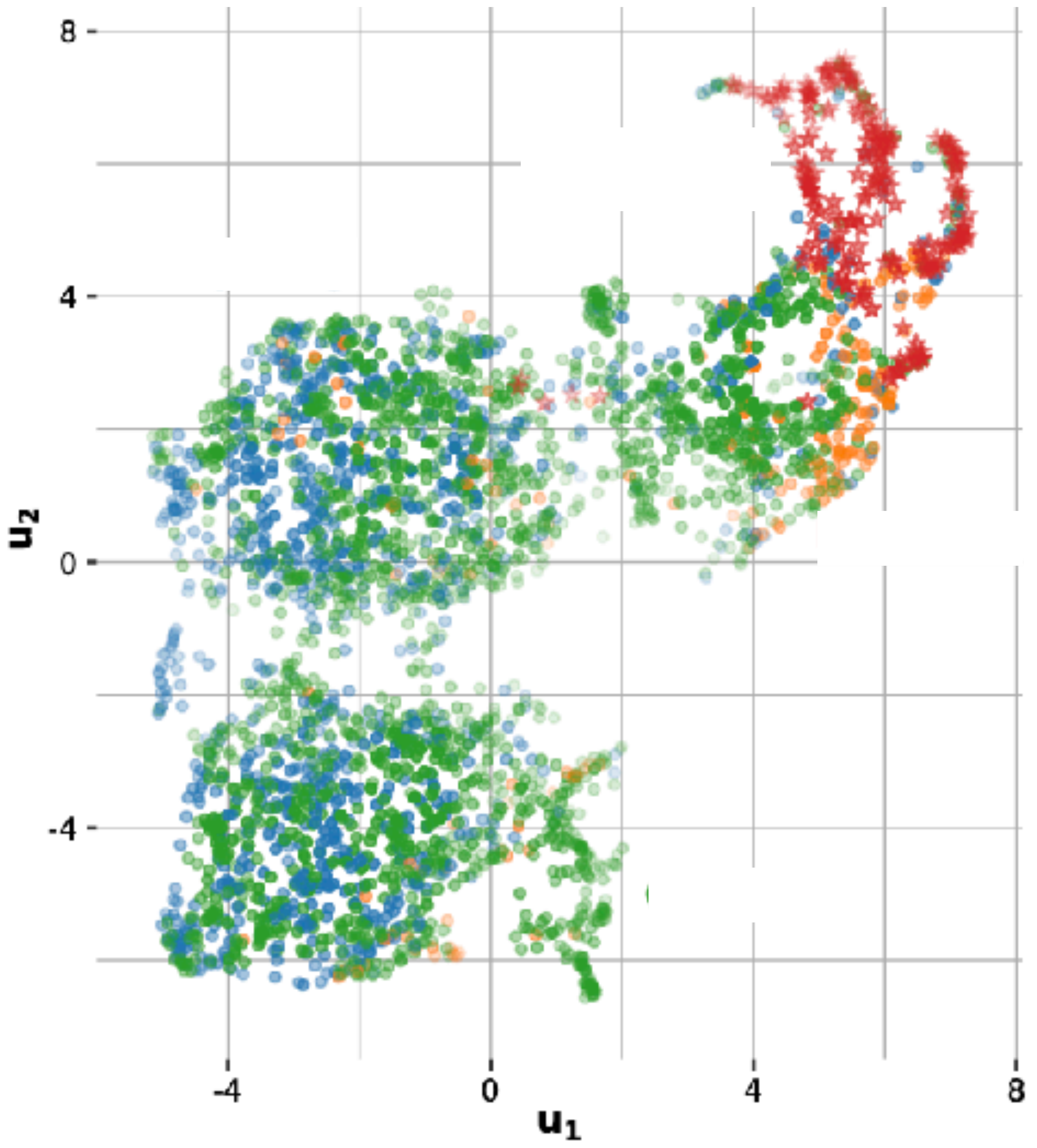
Predict Observed Core Outcomes



- Map 159 GAS cores into the UMAP (Kirk et al 2017, Keown et al 2017, Kerr et al 2019, Chen et al 2019)
- >20% are likely dispersing, >50% likely star-forming
- Single properties (like α) insufficient to classify cores: predictions can be made with machine learning!

Offner et al. 2022

Core Dictionary



Terminology: Dense Core

Starless **Protostellar**

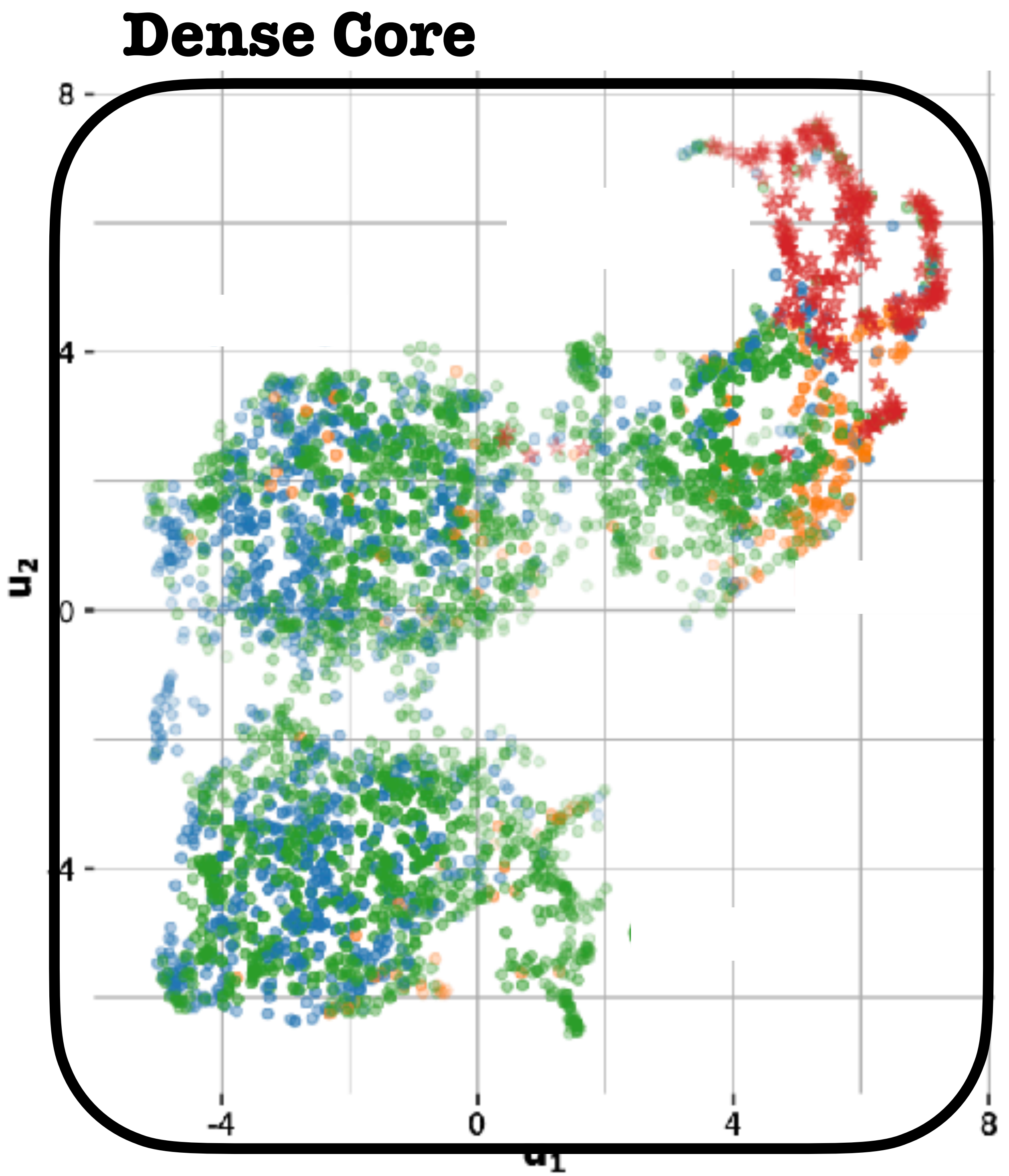
Prestellar

Coherent Core, Droplet

Gravitationally Bound

Pressure Confined

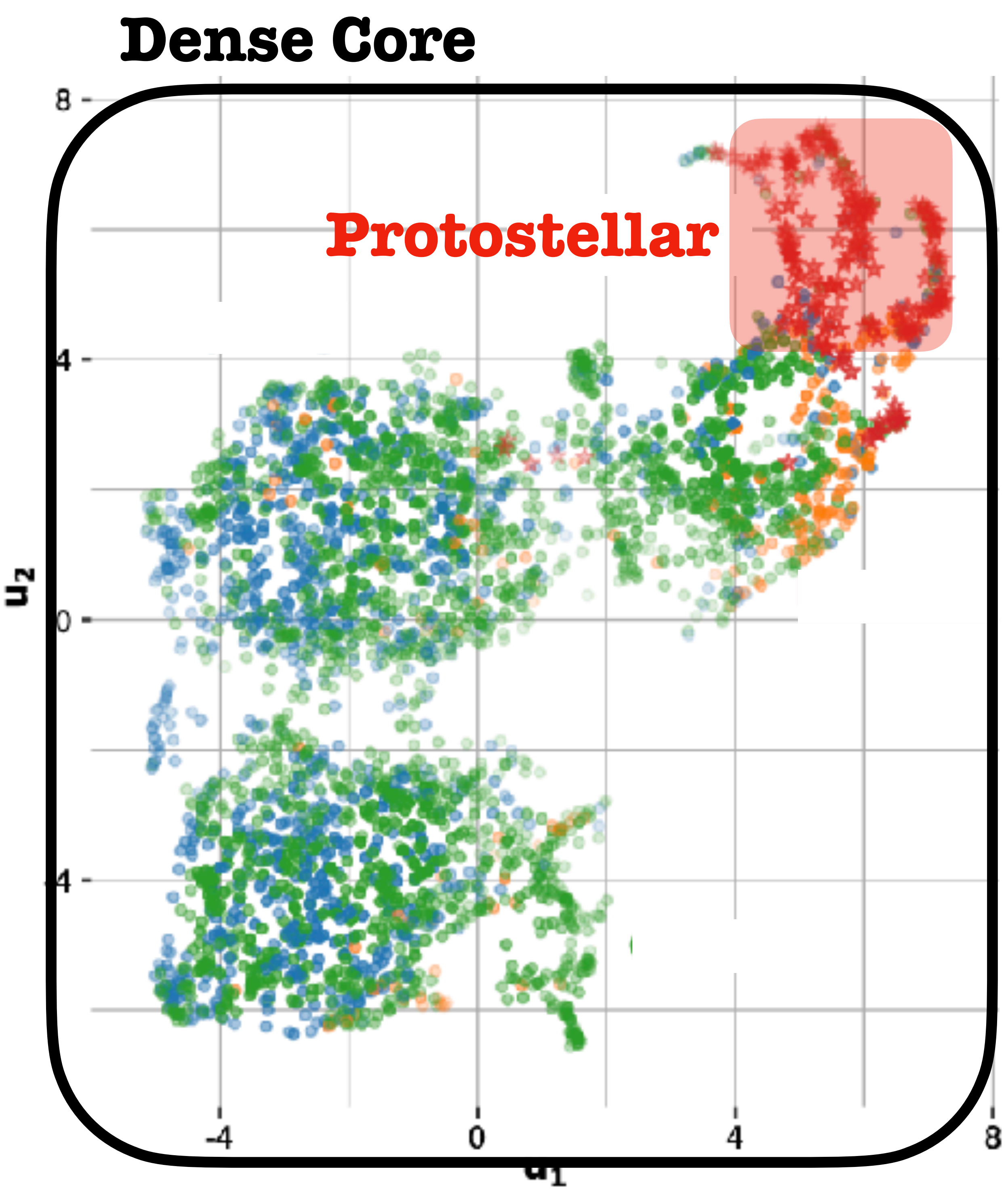
Core Dictionary



Terminology:

Starless **Protostellar**
Prestellar
Coherent Core, Droplet
Gravitationally Bound
Pressure Confined

Core Dictionary



Terminology:

Starless

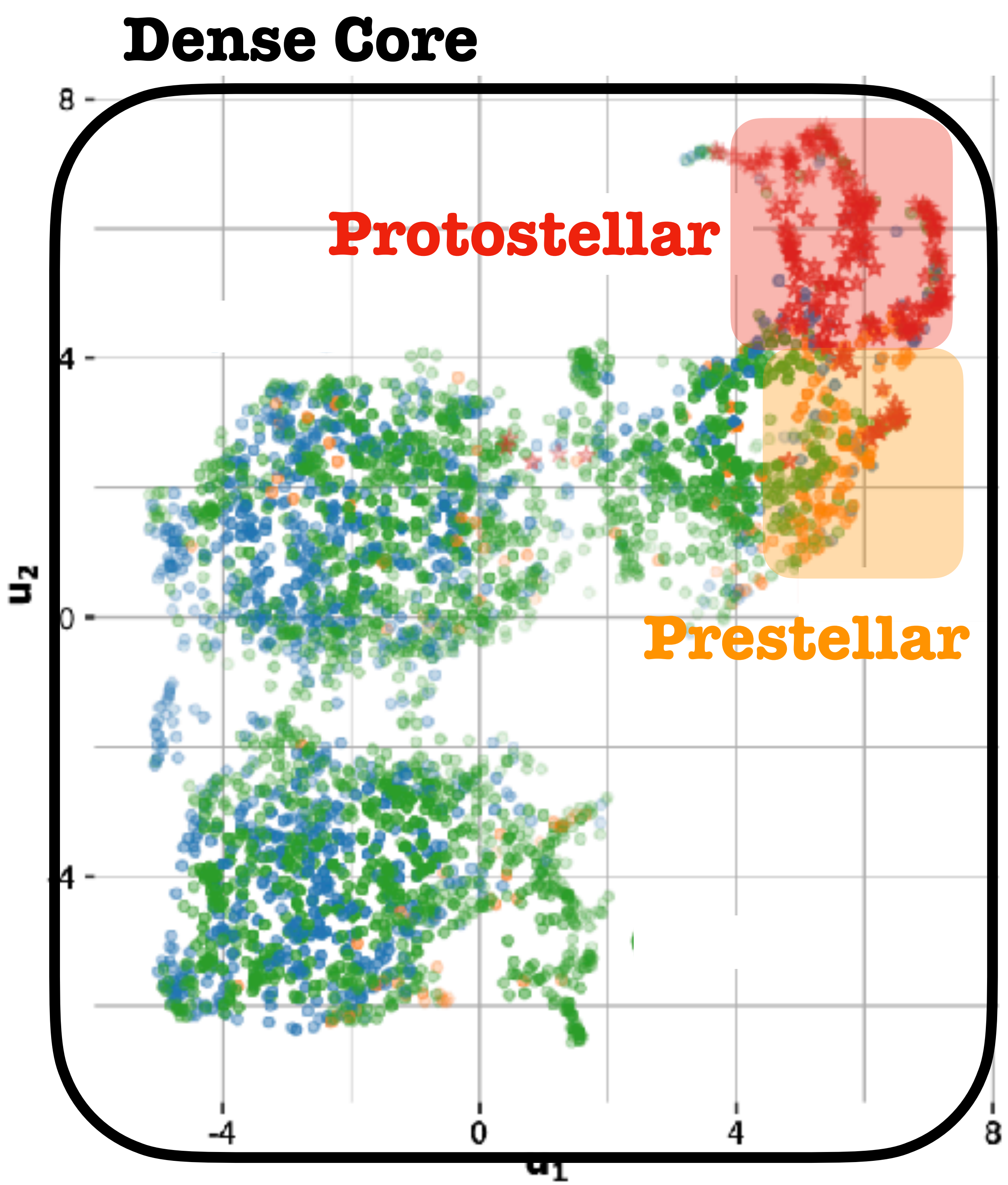
Prestellar

Coherent Core, Droplet

Gravitationally Bound

Pressure Confined

Core Dictionary



Terminology:

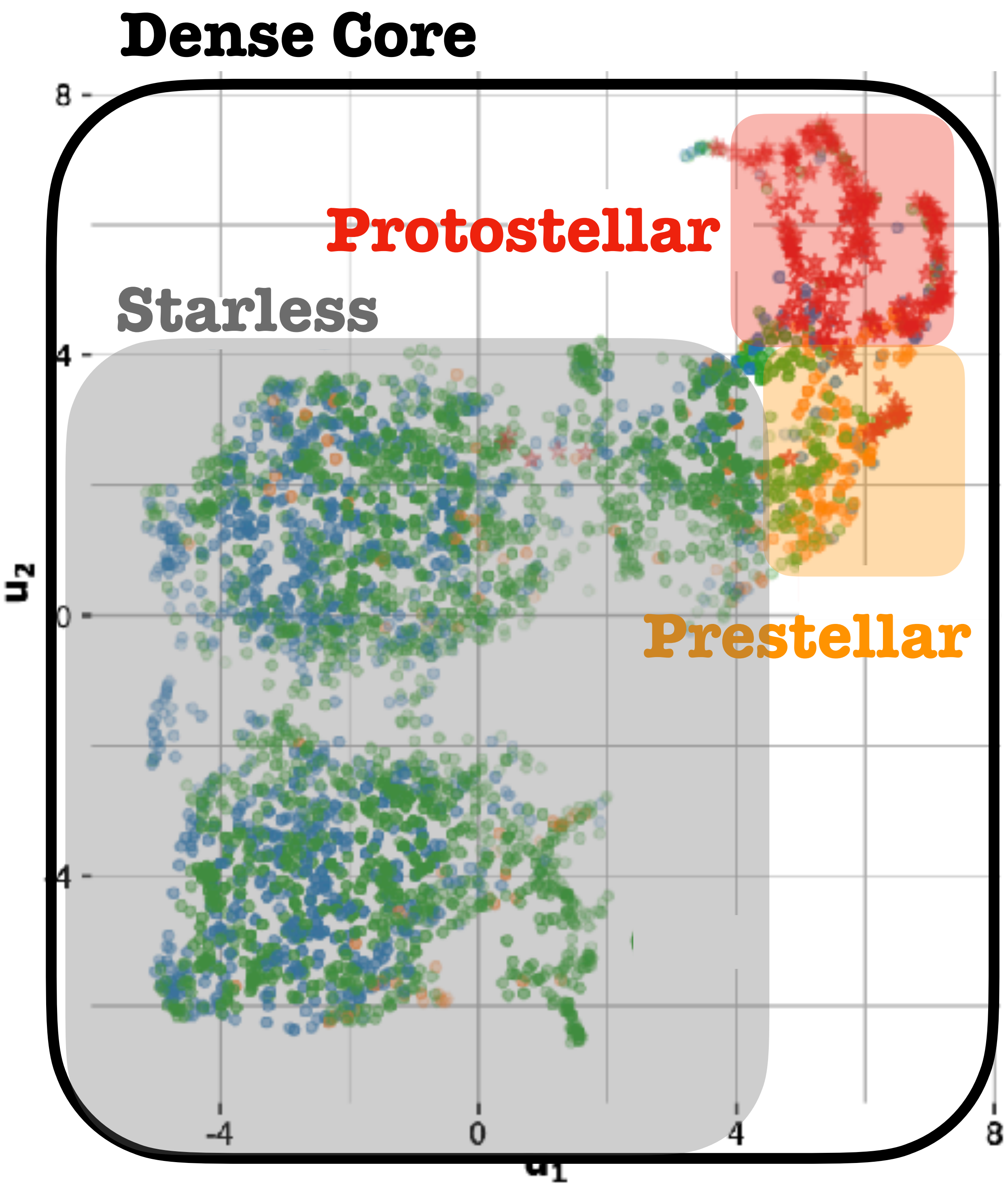
Starless

Coherent Core, Droplet

Gravitationally Bound

Pressure Confined

Core Dictionary



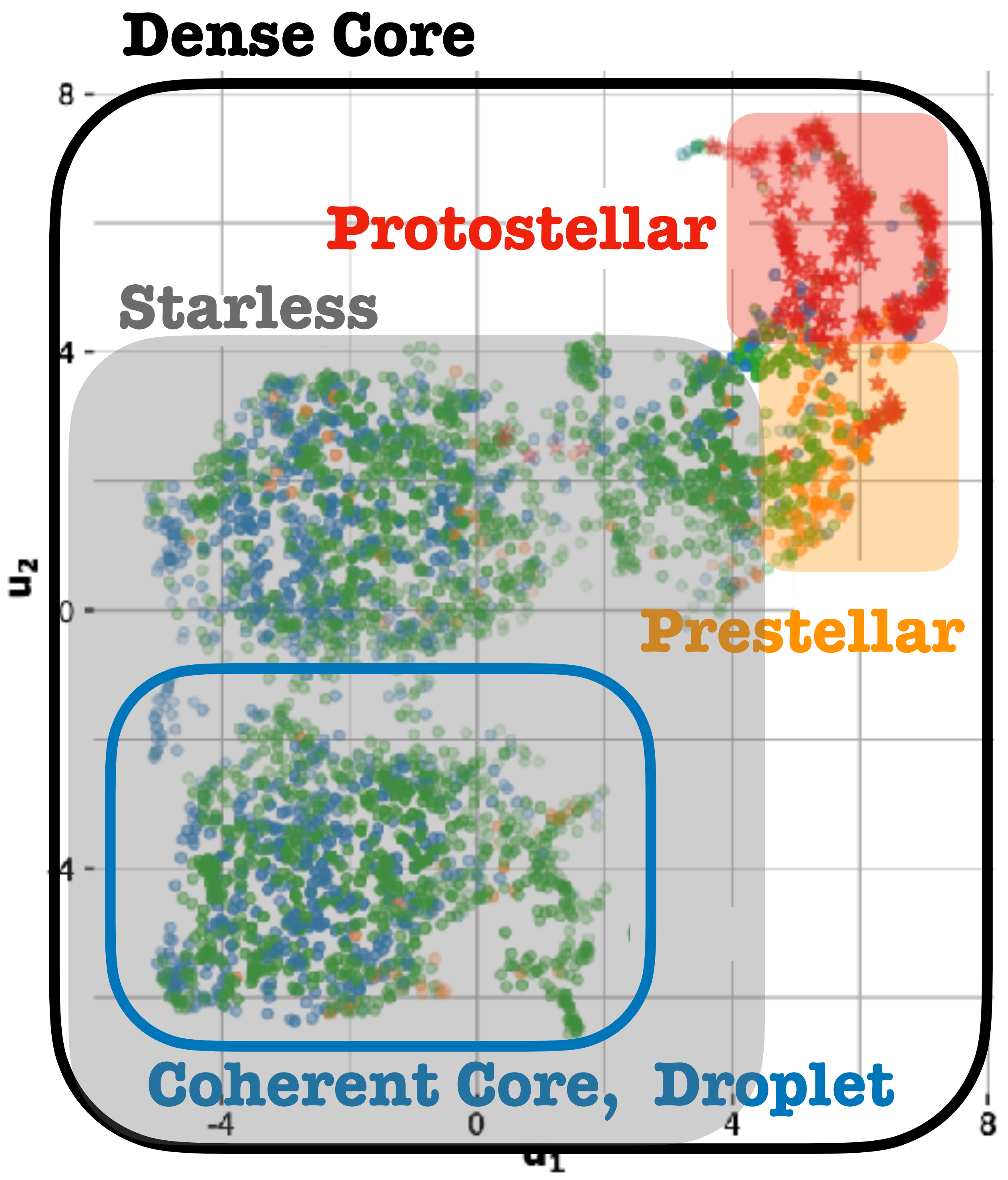
Terminology:

Coherent Core, Droplet

Gravitationally Bound

Pressure Confined

Core Dictionary

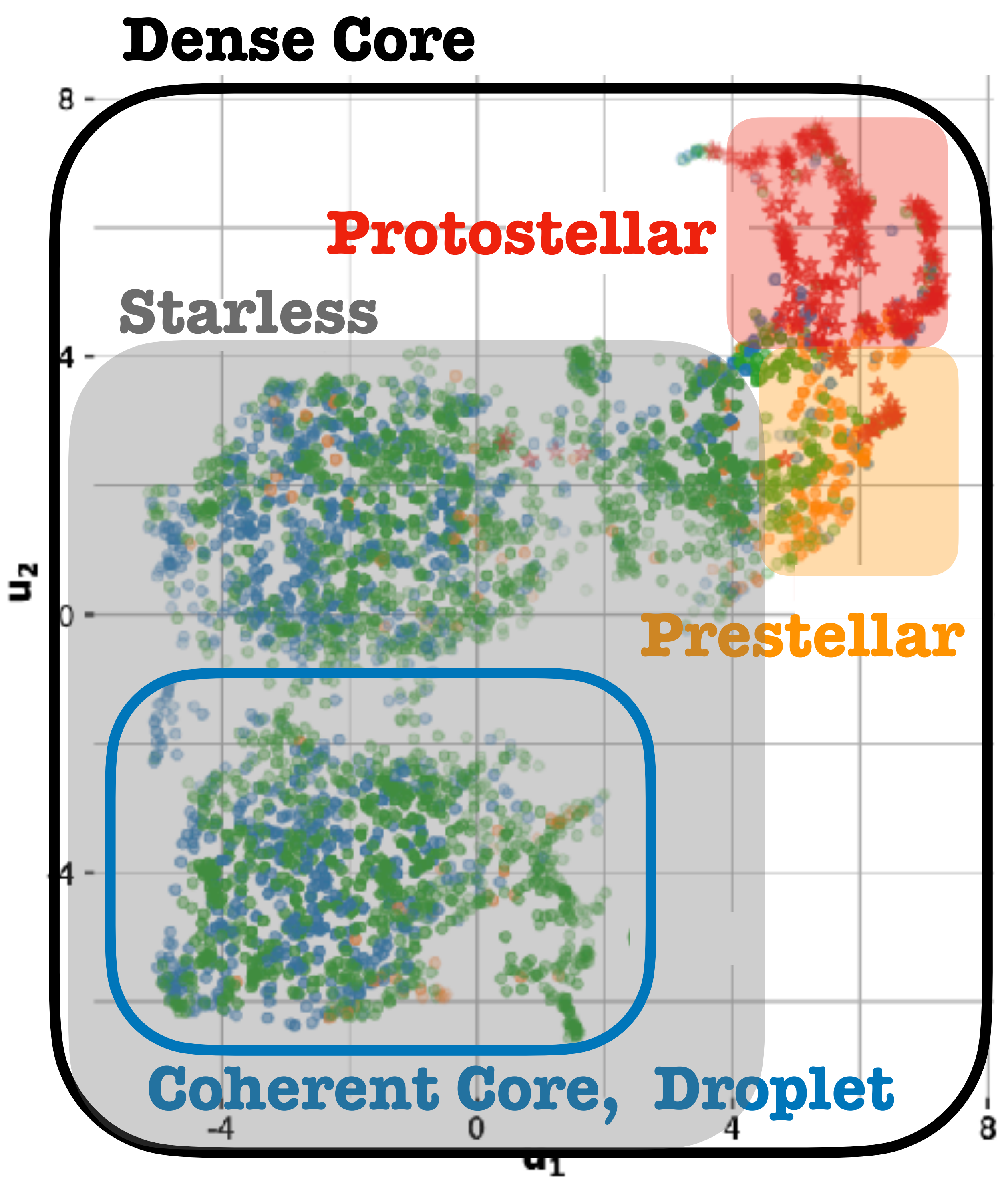


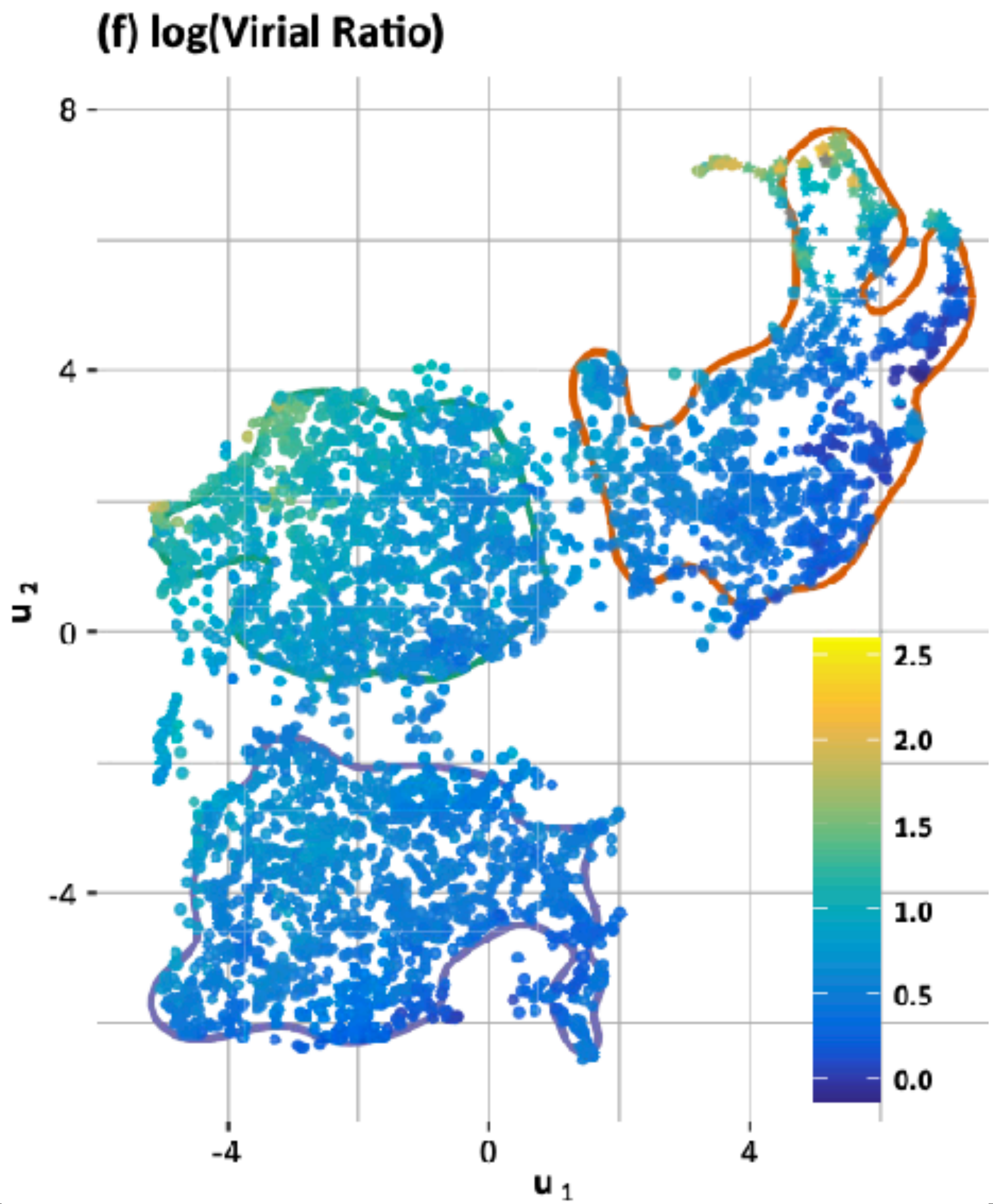
Terminology:

Gravitationally Bound

Pressure Confined

Core Dictionary

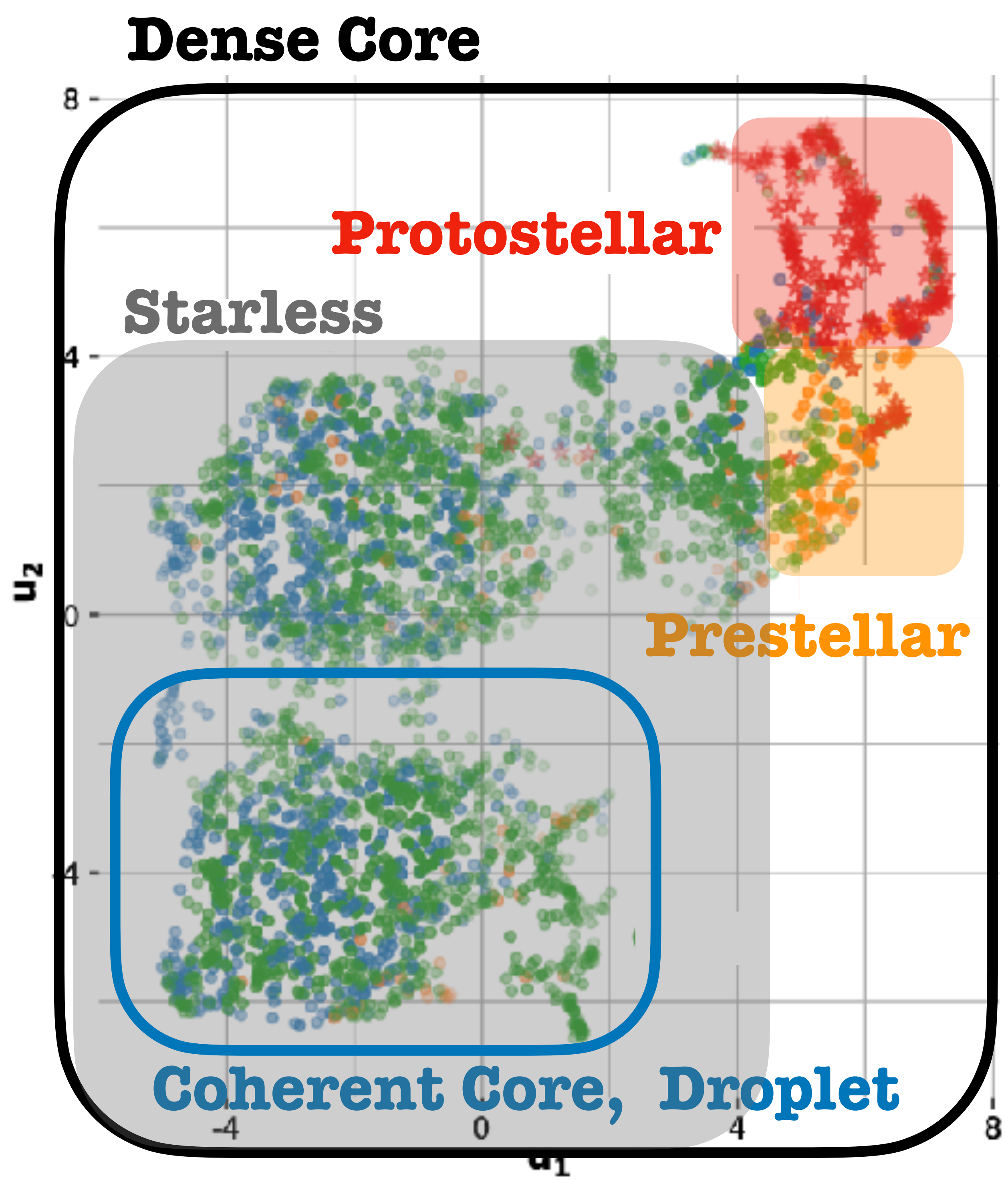


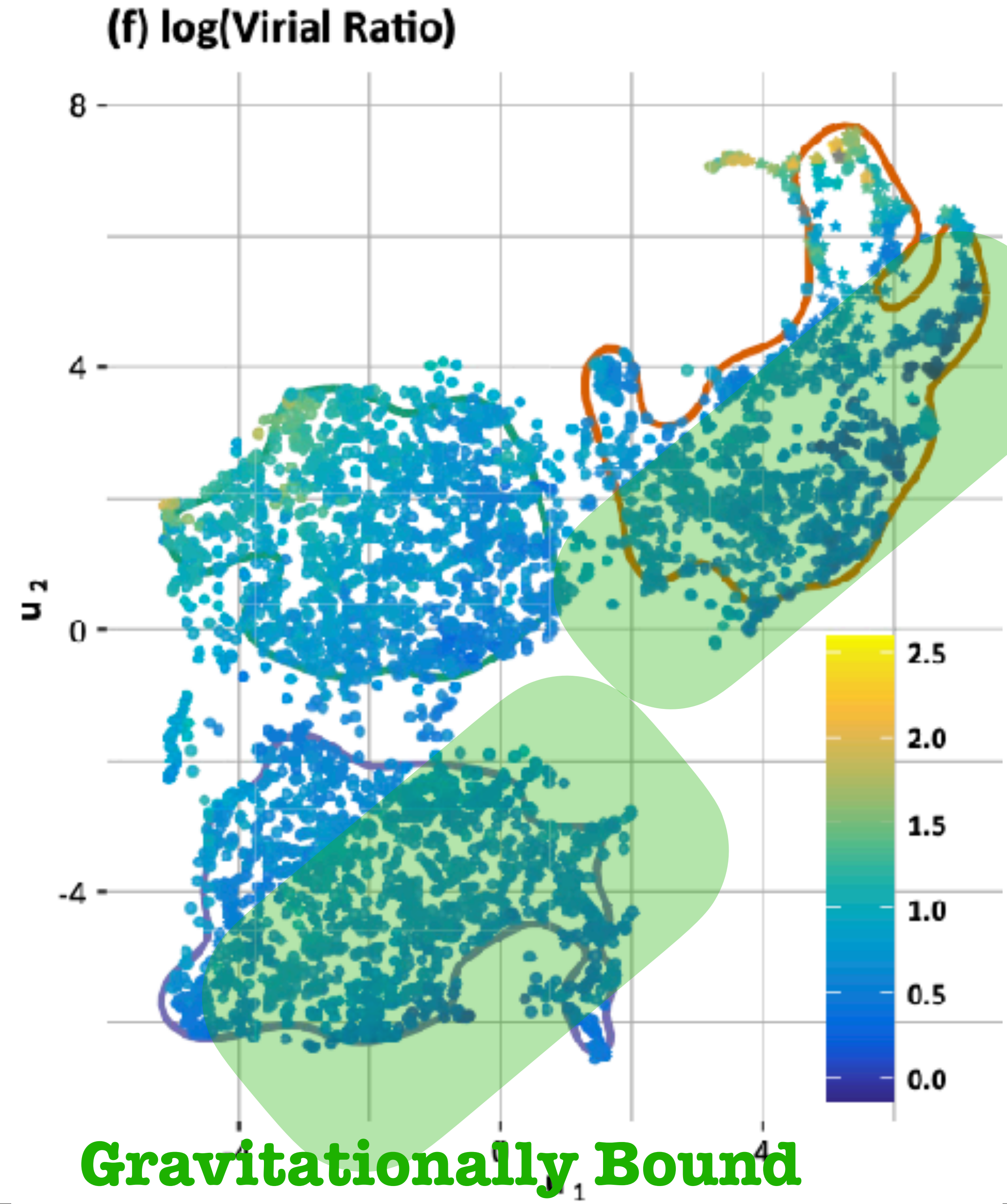


Gravitationally Bound

Pressure Confined

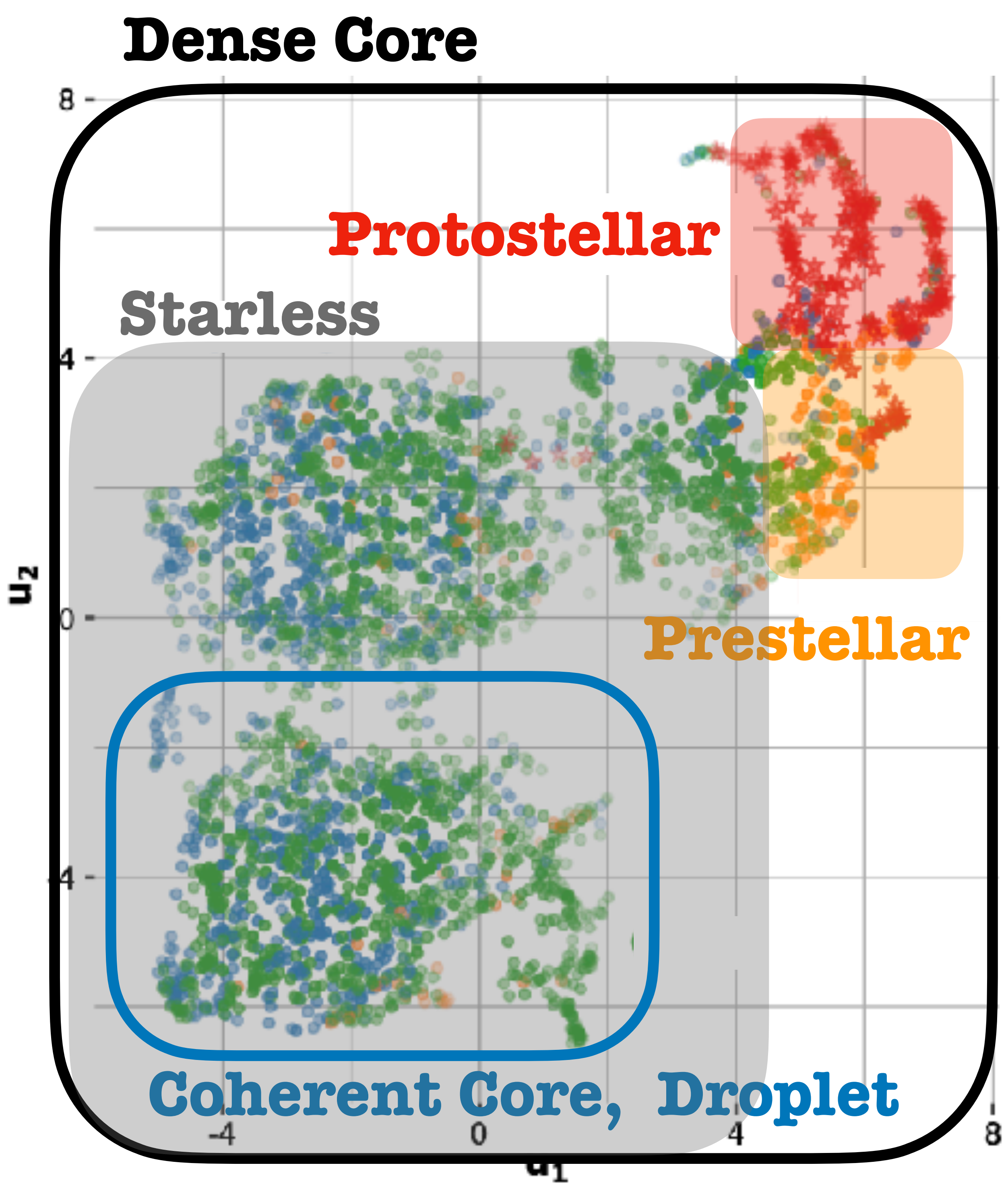
Core Dictionary

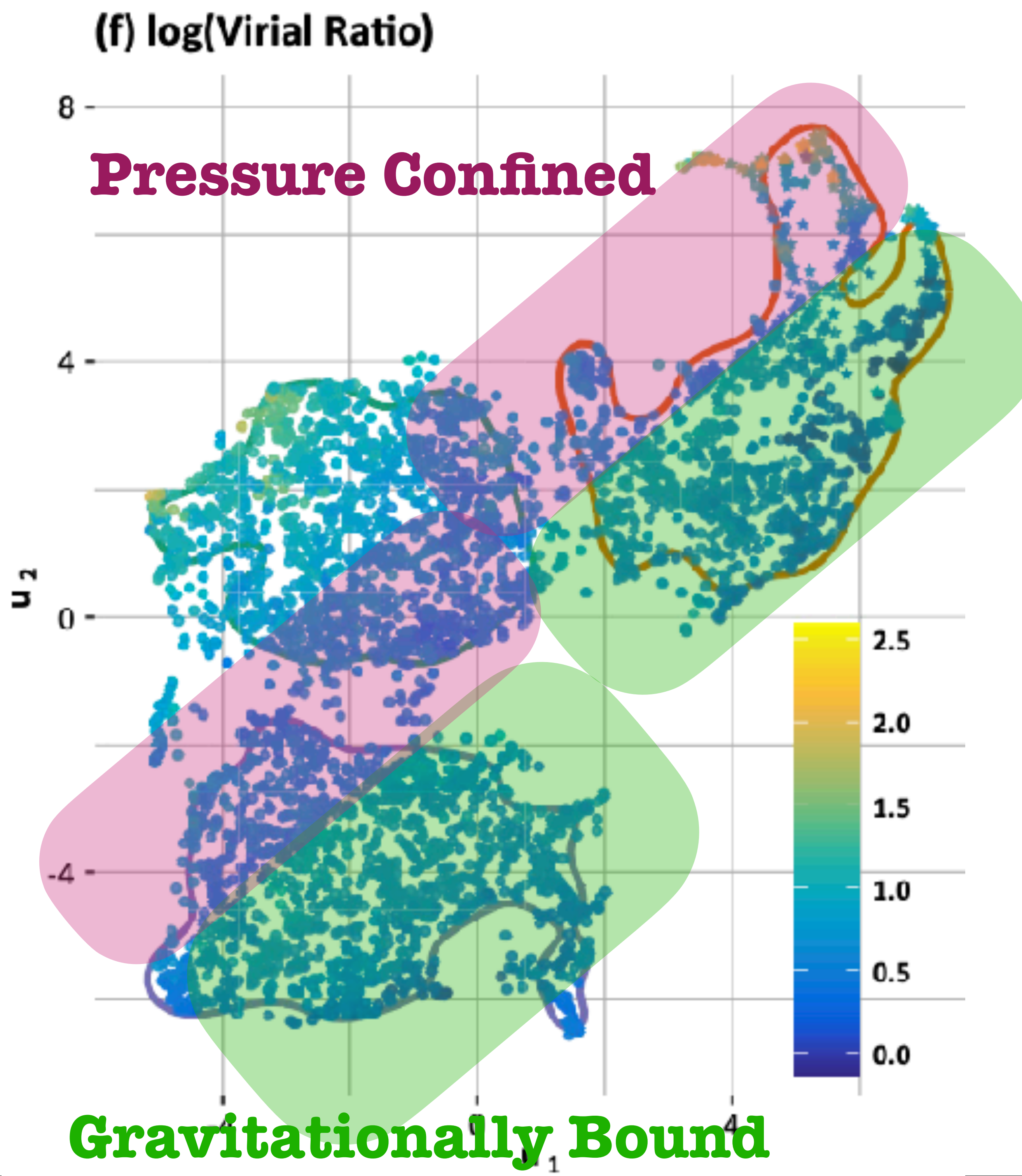




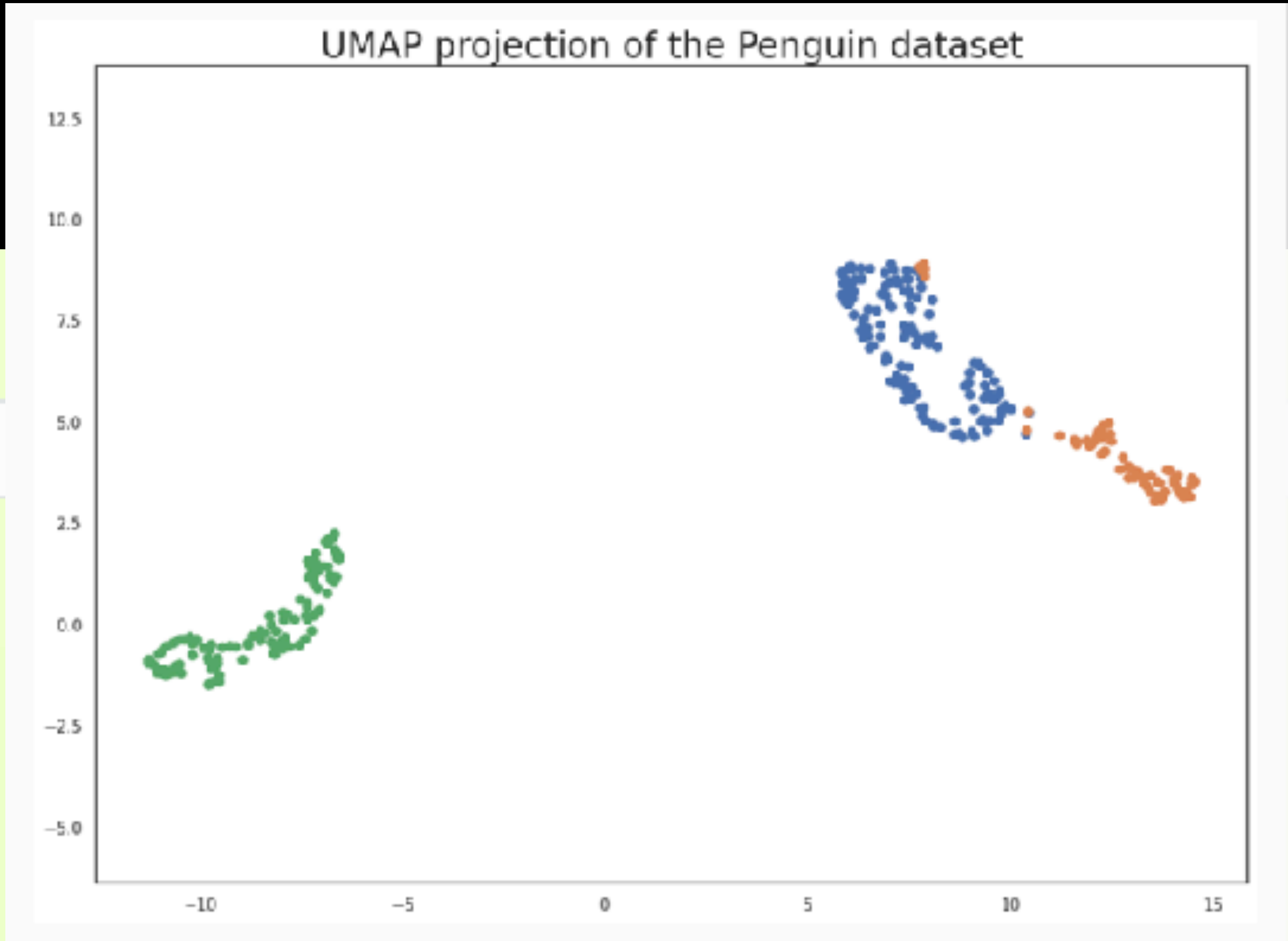
Pressure Confined

Core Dictionary





Python Example




```python
import umap

reducer = umap.UMAP()

penguin_data = penguins[
    [
        "bill_length_mm",
        "bill_depth_mm",
        "flipper_length_mm",
        "body_mass_g",
    ]
].values
scaled_penguin_data = StandardScaler().fit_transform(penguin_data)
embedding = reducer.fit_transform(scaled_penguin_data)

plt.scatter(
    embedding[:, 0],
    embedding[:, 1],
    c=[sns.color_palette()[x] for x in penguins.species.map({"Adelie":0, "Chinstrap":1, "Gentoo
```

https://umap-learn.readthedocs.io/en/latest/basic_usage.html

Summary Problem 3: High-D Data

- Unsupervised machine learning is able to **identify and visualize complex, hidden relationships**
- Cores evolve through 3 phases of evolution (turbulent, coherent, pre-protostellar)
- Can be **easily implemented** using umap-learn python package.

Problem 3: The data is messy, noisy, and complex

Taurus Molecular Cloud
Herschel

What is the distribution and impact of stellar feedback in molecular clouds?

Identification

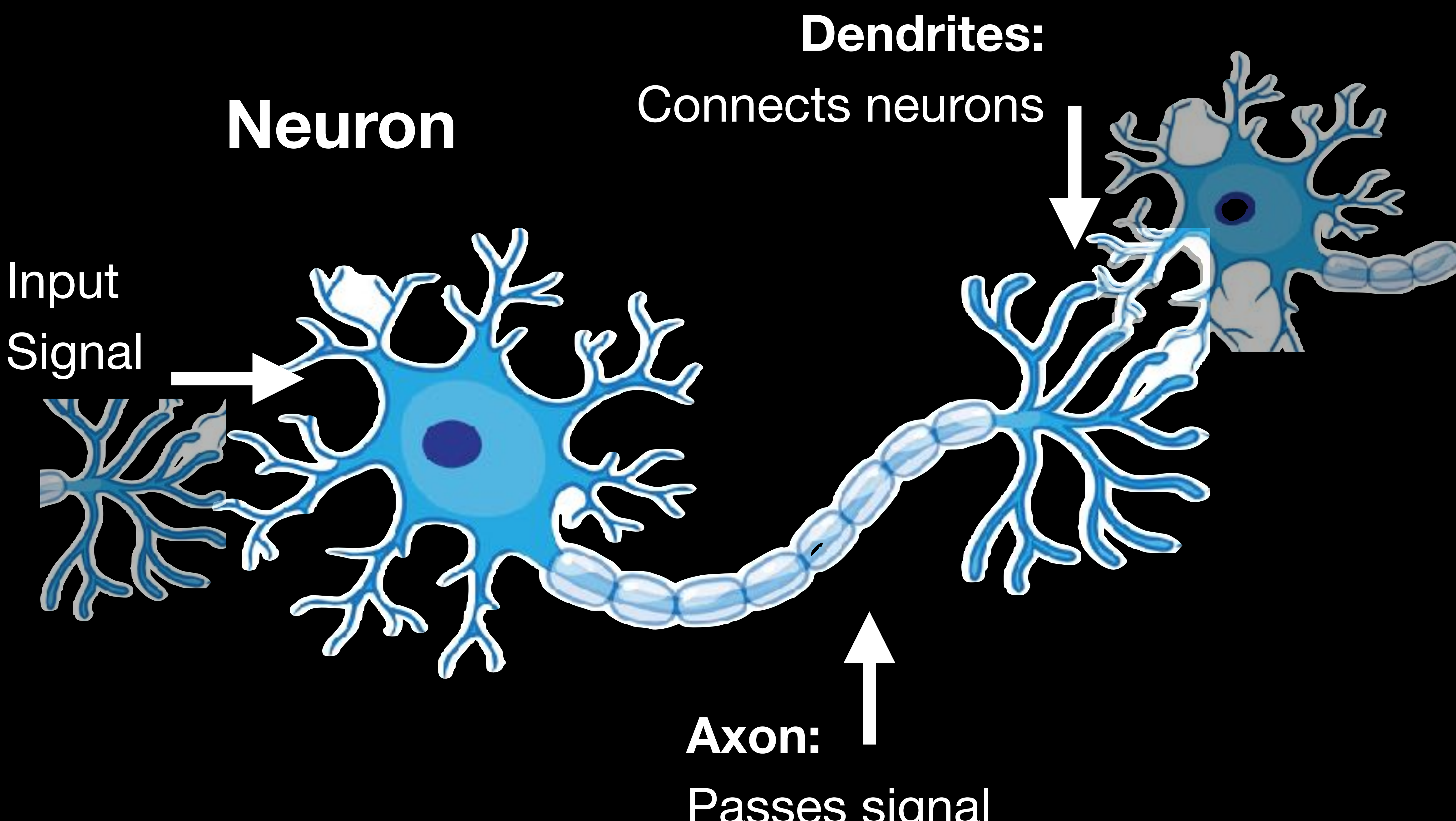
Finding Signals

Deep Learning

Neural Networks

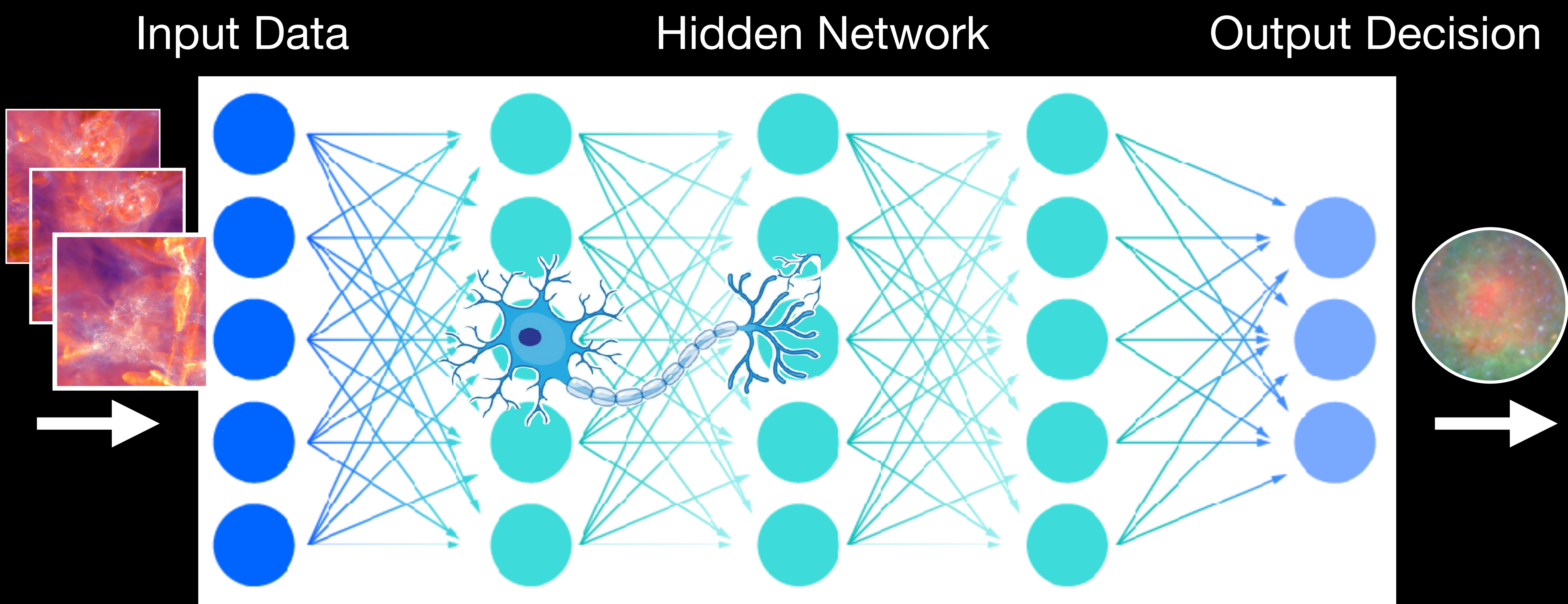
Deep Learning

Neural Networks



Deep Learning

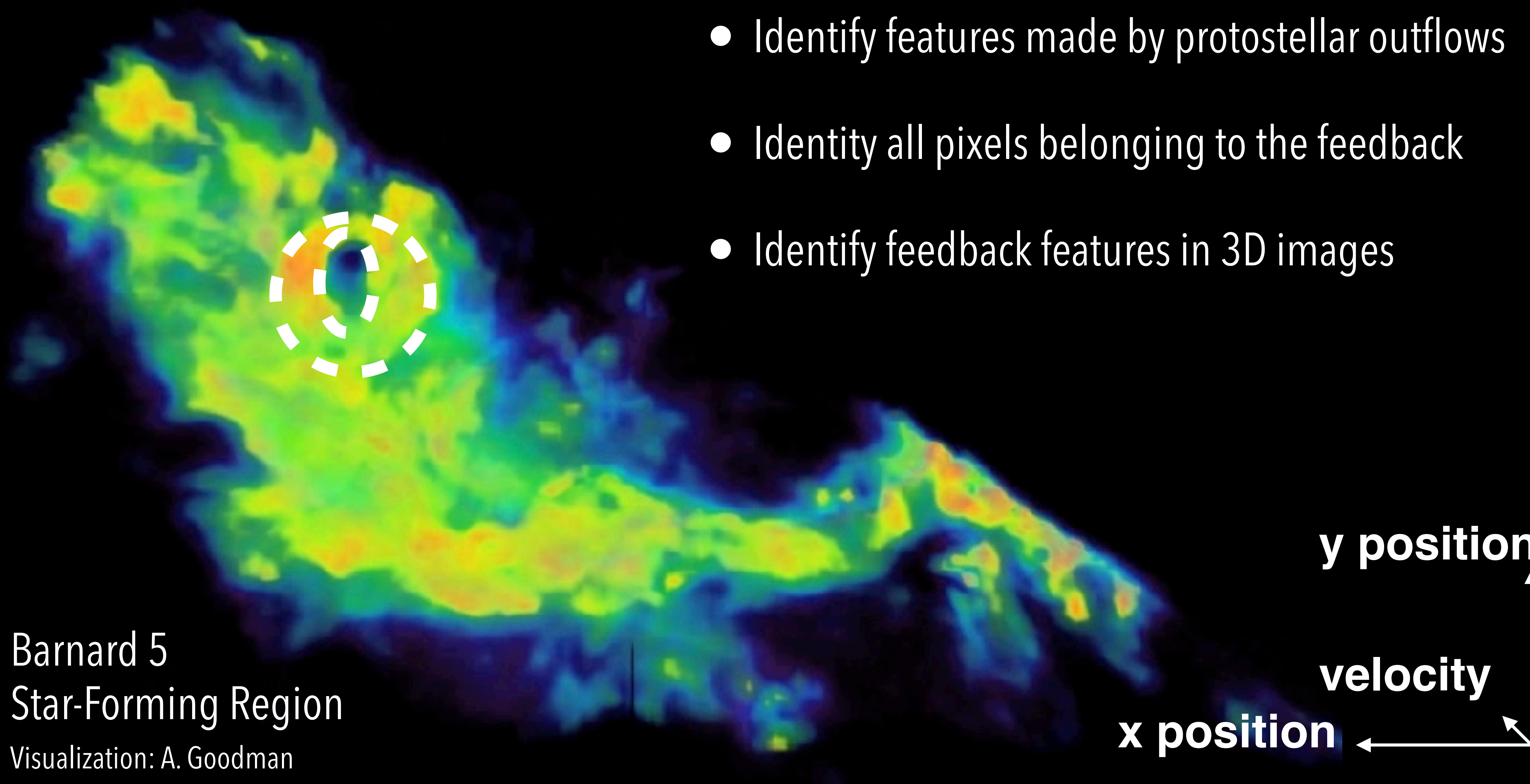
(Artificial) Neural Networks



Finding Stellar Feedback

Goals for our Neural Network:

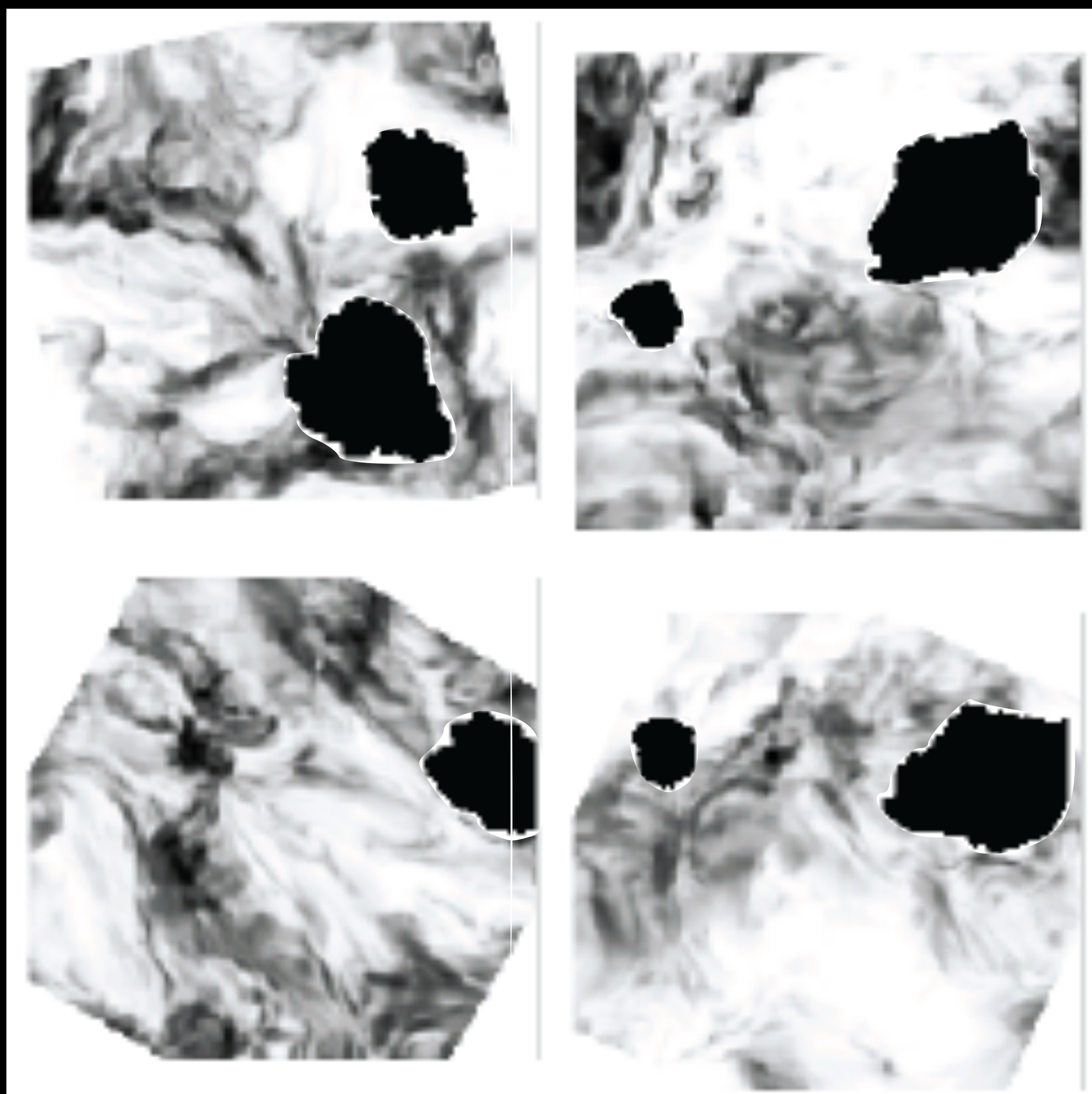
- Identify bubbles made by stellar winds
- Identify features made by protostellar outflows
- Identity all pixels belonging to the feedback
- Identify feedback features in 3D images



Barnard 5
Star-Forming Region
Visualization: A. Goodman

Finding Stellar Feedback

Convolutional Approach to Structure Identification (CASI-3D)



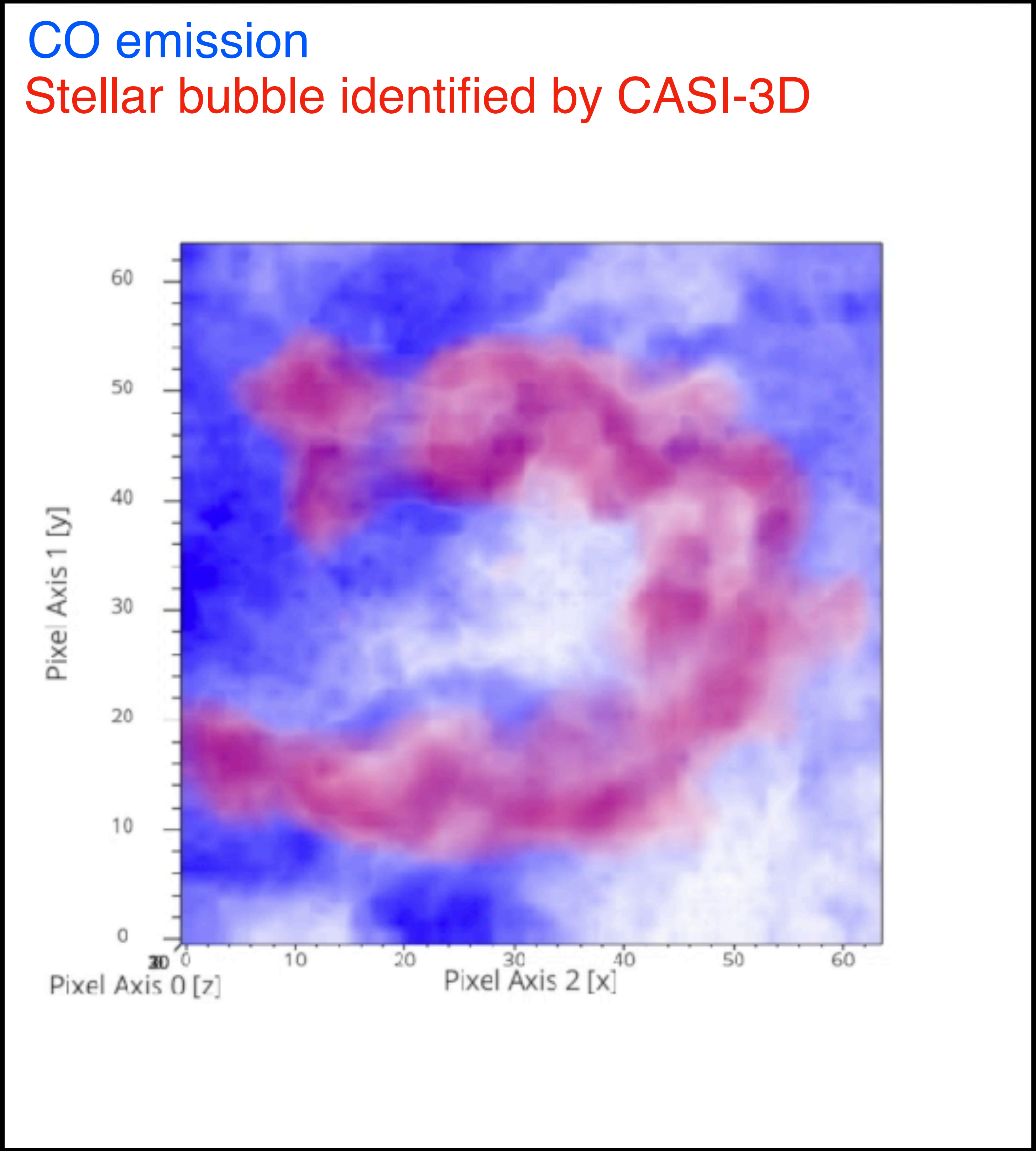
Training data

- Create neural network: CASI:3D
- Train with simulations of molecular clouds forming stars
- Create mock observations

Prediction:
Wind Bubbles

Van Oort et al. 2019

Apply to observations of molecular clouds



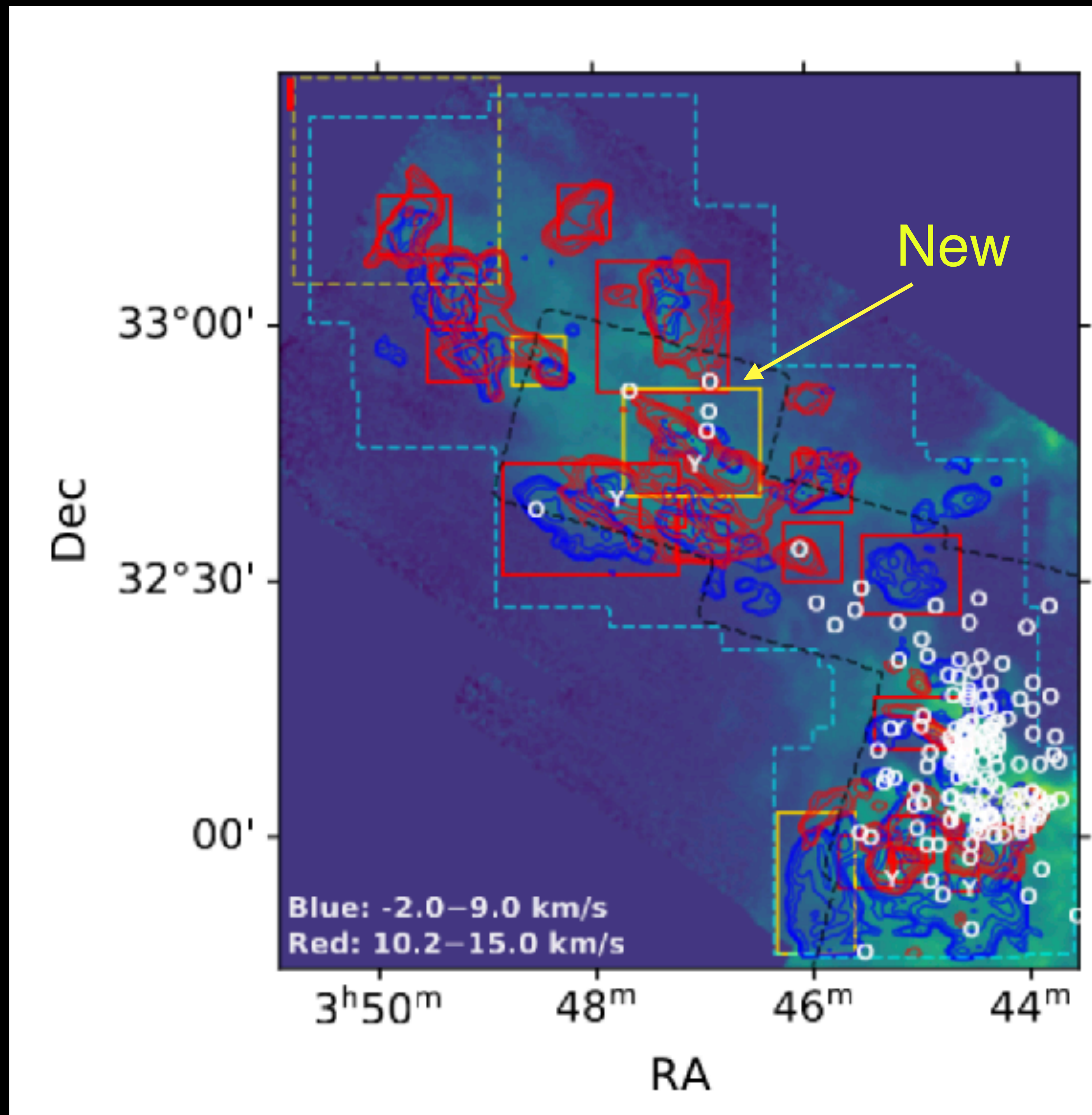
Stellar wind bubble in the Taurus Molecular Cloud

Xu et al. 2020

Predict Outflows in Perseus

- Identifies all 60 known visually identified outflows
- Identifies 20 new outflows!
- Identifies outflows in confused regions!

Machine-Identified Outflows

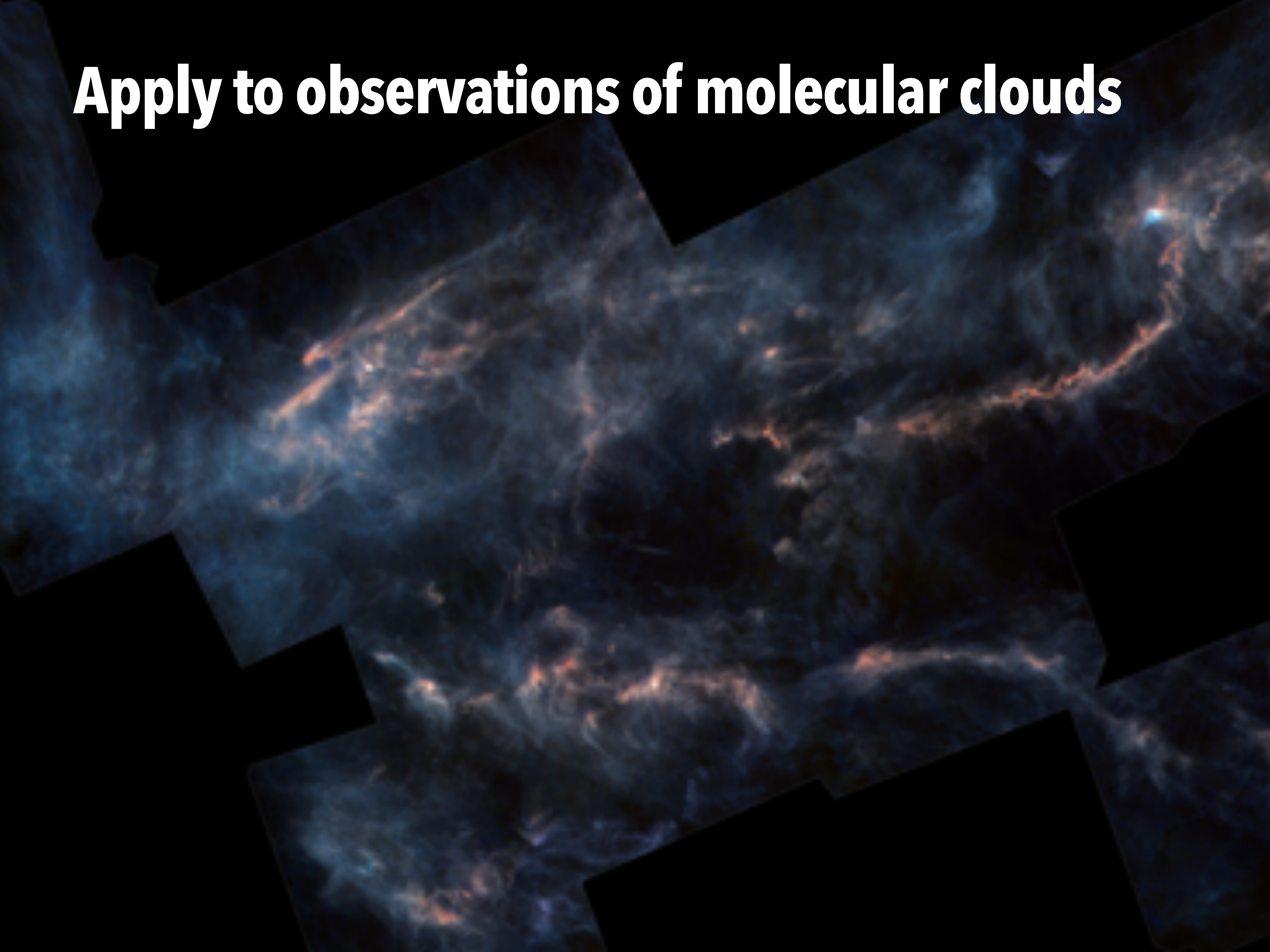


Y= young star
O = older young star

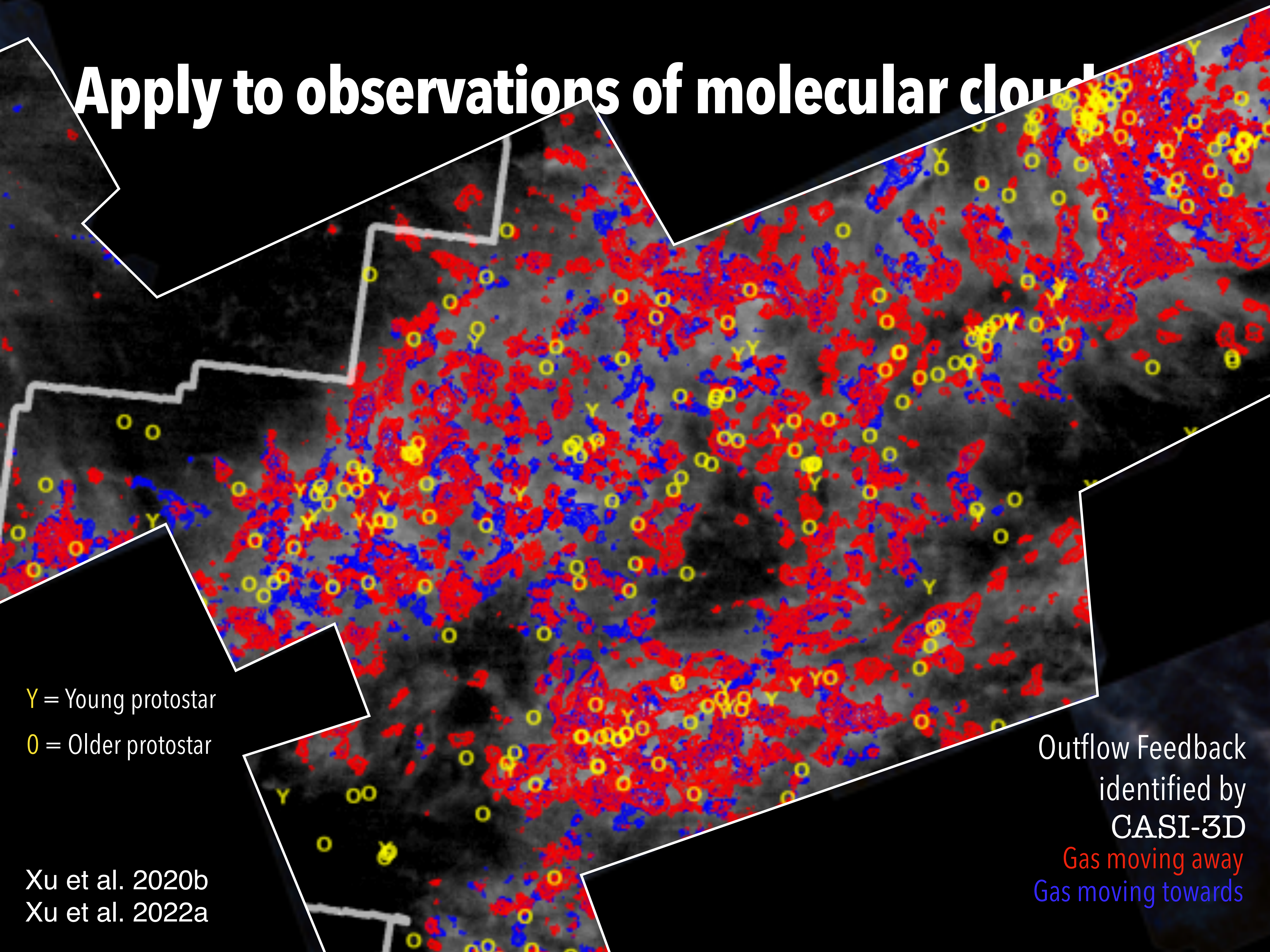
Cluster With ~100 young stars

Xu et al 2020b

Apply to observations of molecular clouds



Apply to observations of molecular cloud



Xu et al. 2020b
Xu et al. 2022a

Summary Problem 3: Messy, Noisy Data

- CNNs provide a **fast, flexible, automated** way to identify complex 2D and 3D structures..
- CASI-3D produces a feedback **map** not a catalog.
- Some new outflows and bubbles found!
- Impact of bubbles (stellar winds) is over-estimated by a factor of 10 due to observational bias; impact of protostellar outflows is comparable to previous visual estimates.
- **Intermediate difficulty** to implement, many public packages/examples.

https://gitlab.com/casi-project/casi-3d/-/tree/master

Problem 4: We observe photons…

- Basic quantities can't be directly measured: density, temperature, magnetic field
- We need to infer these quantities from the light we observe
- Only have a subset of wavelengths emitted + lots of complications

Mon R2
Star Forming Region

Credit: R. Pokhrel, Herschel

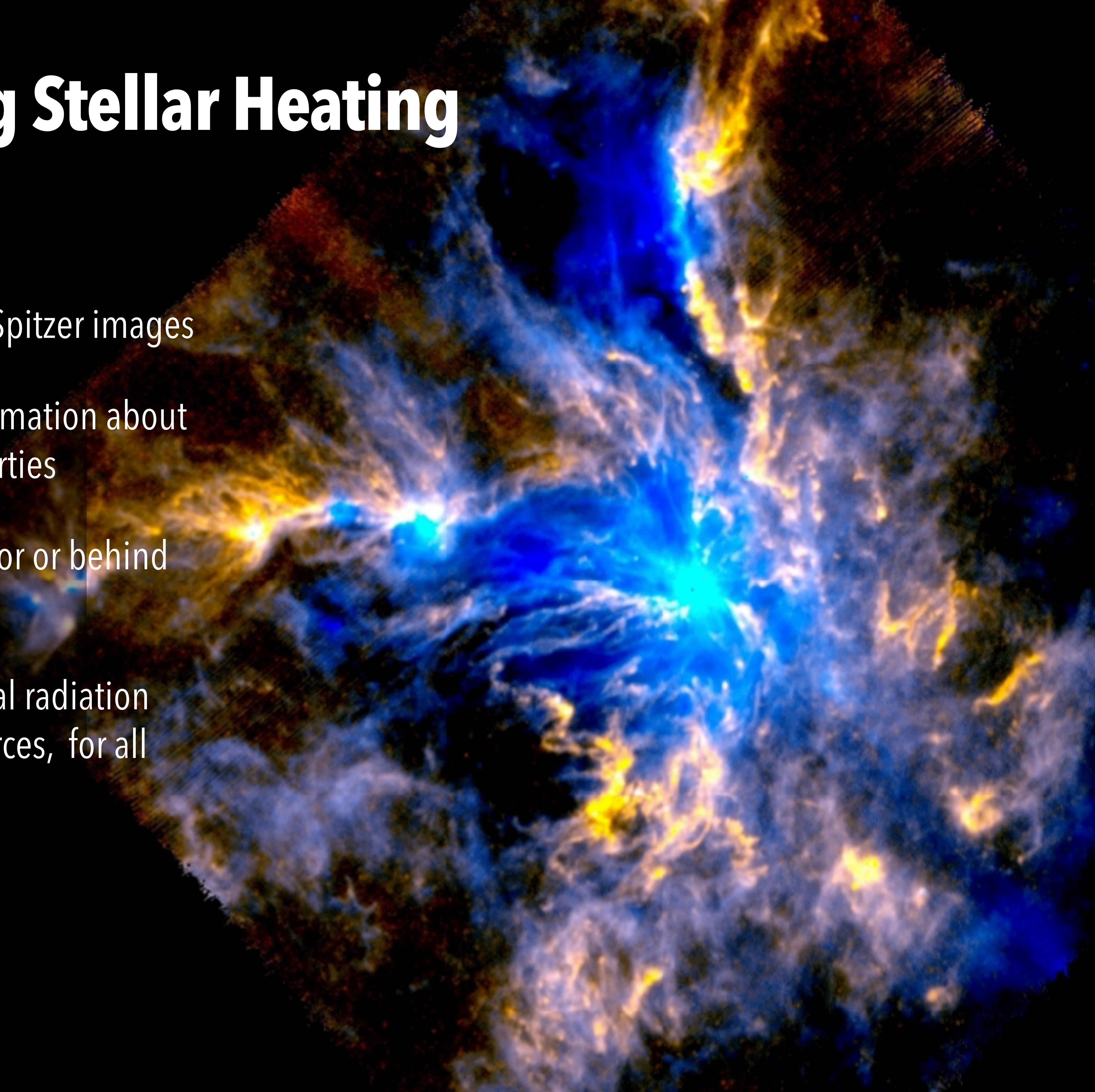
Predicting

Generative AI

Predicting Stellar Heating

Goals:

- **Input:** multi-band Spitzer images
- Predict *without* information about star locations, properties
- Ignore stars in front or or behind the cloud
- **Output:** predict total radiation energy, from all sources, for all pixels



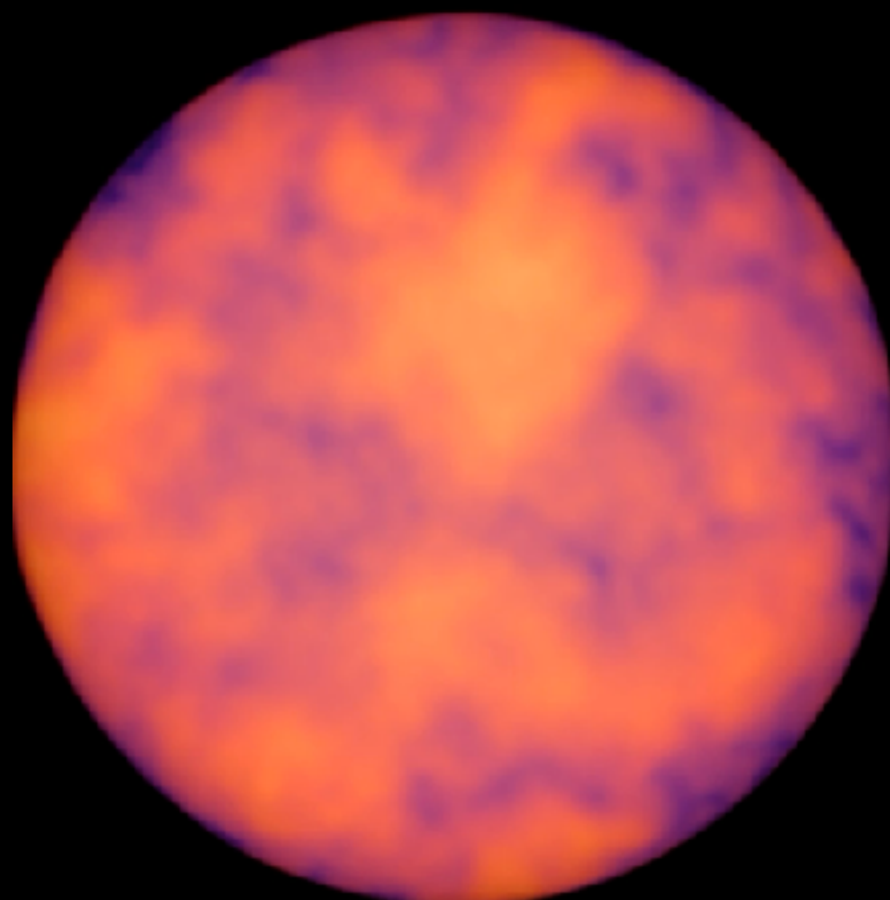
Mon R2
Star Forming Region

Credit: R. Pokhrel, Herschel

Fly-through + Time Animation

Low density (purple) ↔ orange ↔ **High** density (white)

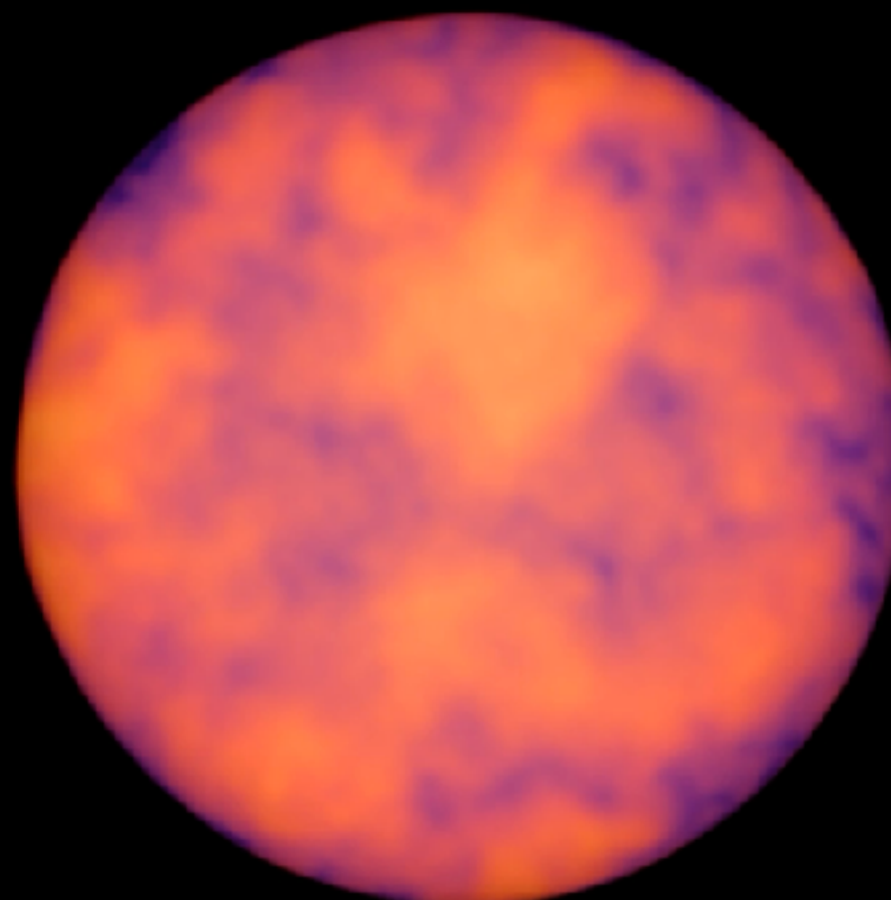
Yellow/Green ↔ hotter gas



STAR FORmation in **G**aseous **E**nvironments (**STARFORGE**)

STARFORGErs: Mike Grudic (Carnegie), Stella Offner, Phil Hopkins (Caltech), Anna Rosen (UCSD), Claude-Andre Facher-Giguere (Northwestern)

Grudic et al. 2021, 2022



Fly-through + Time Animation

Low density (purple) ↔ orange ↔ **High** density (white)

Yellow/Green ↔ hotter gas

20,000 Solar Masses

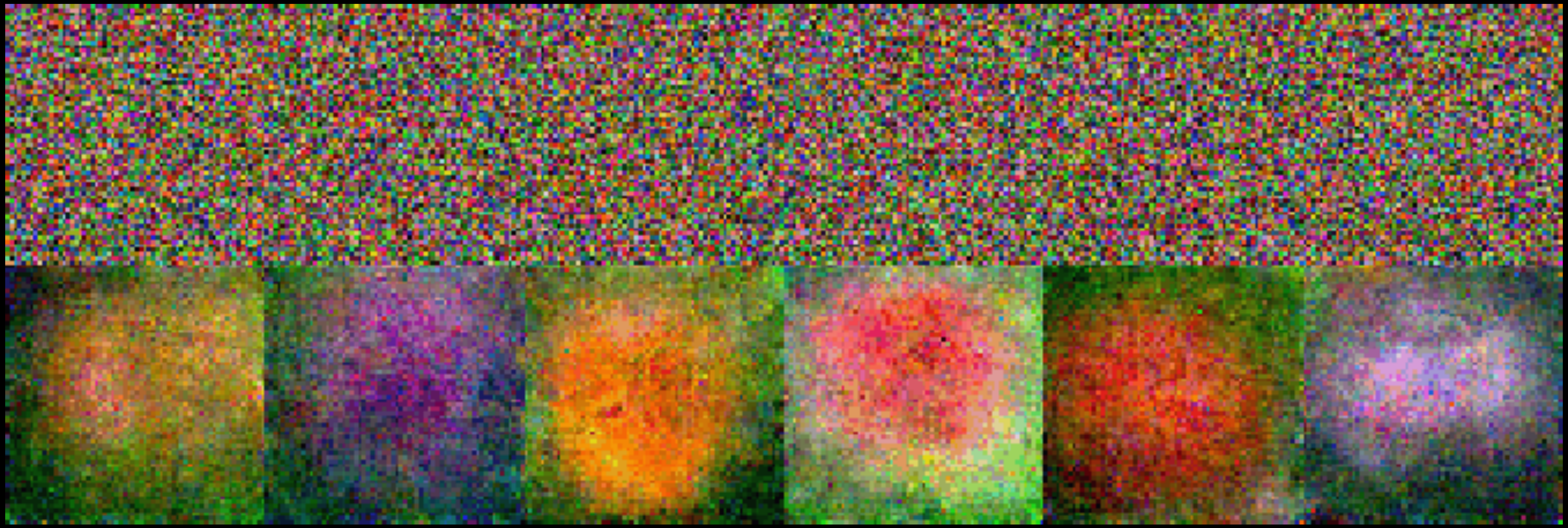
20,000,000 cells

Magnetic fields, radiation, stars, outflows, winds, supernovae

Prediction through Generative AI

Denoising Diffusion Probabilistic Models

Credit: Ho et al. 2020

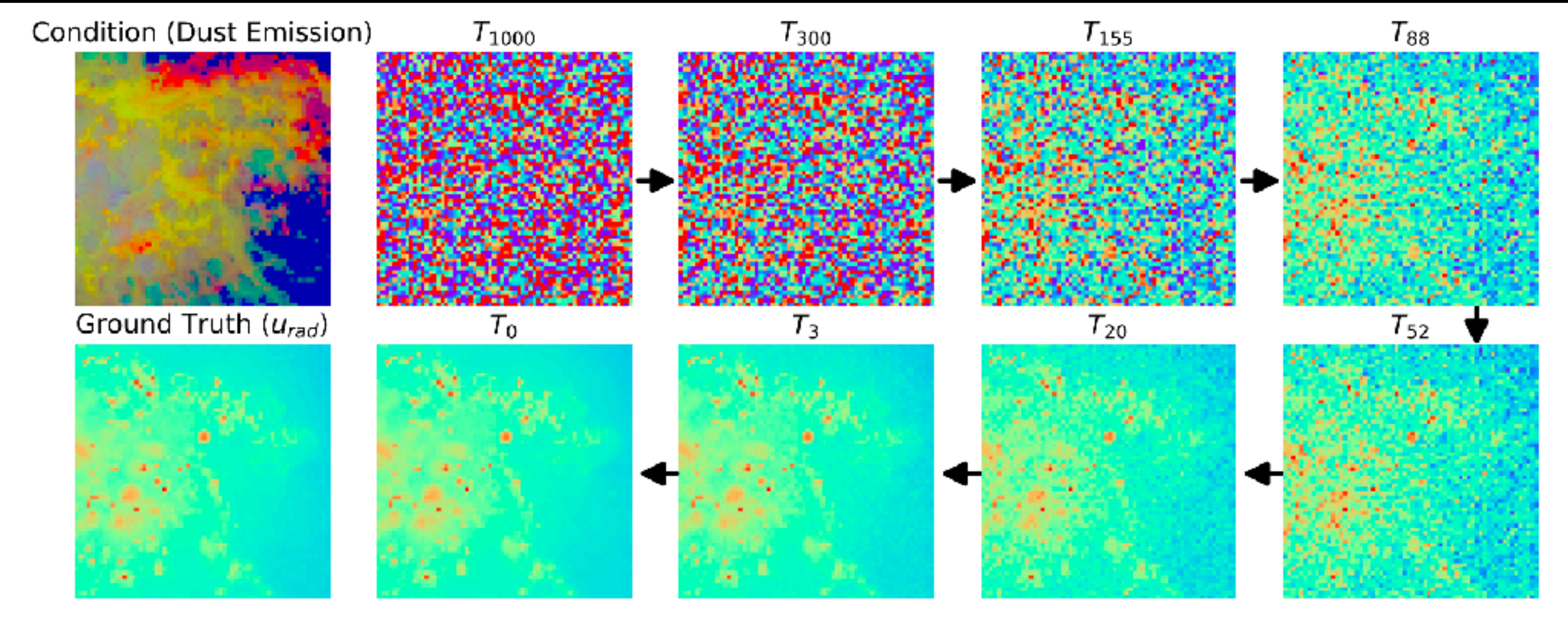


DALL-E 2

Credit: A. Beres

Training

Input: Simulated 3 band infrared emission

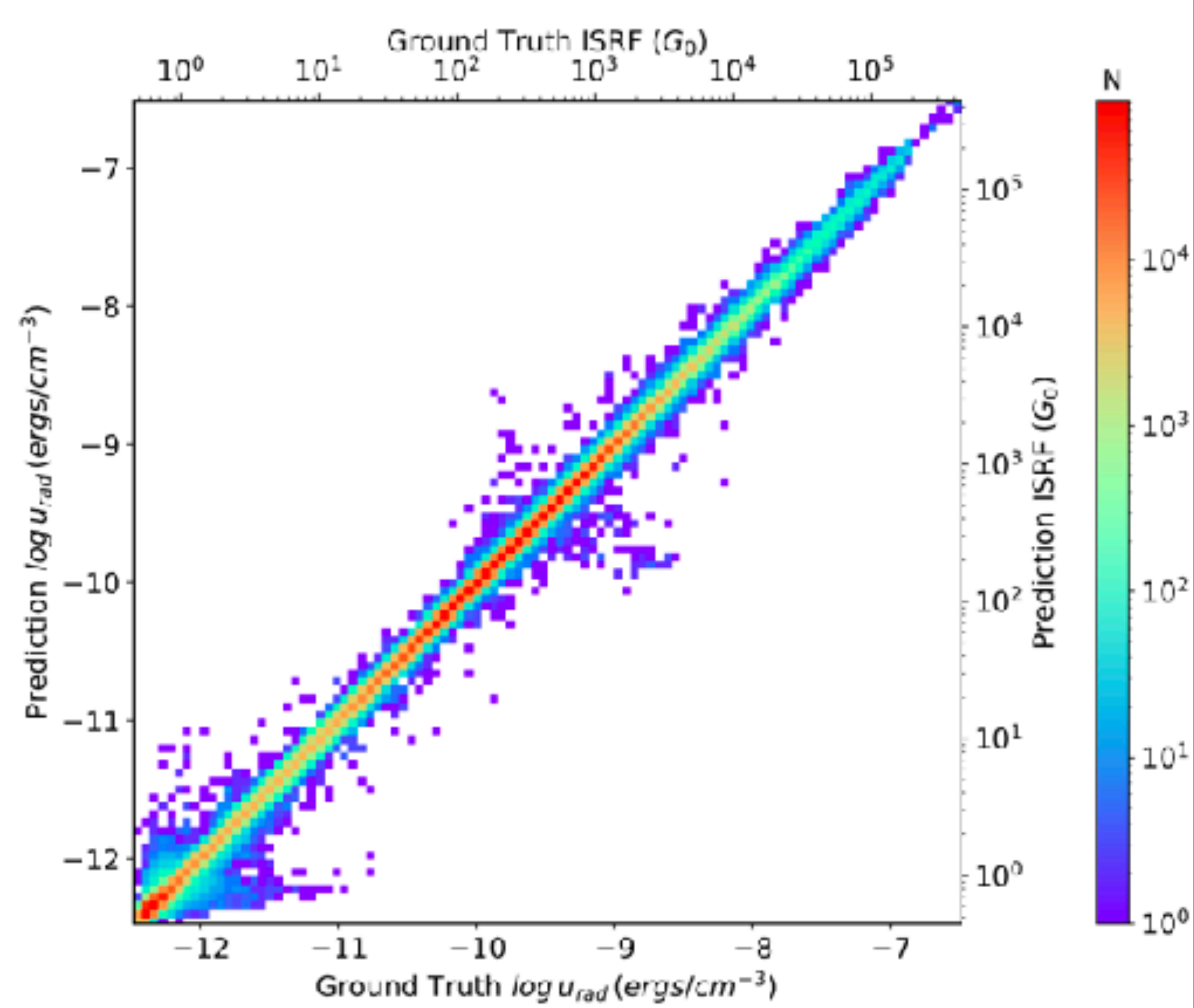


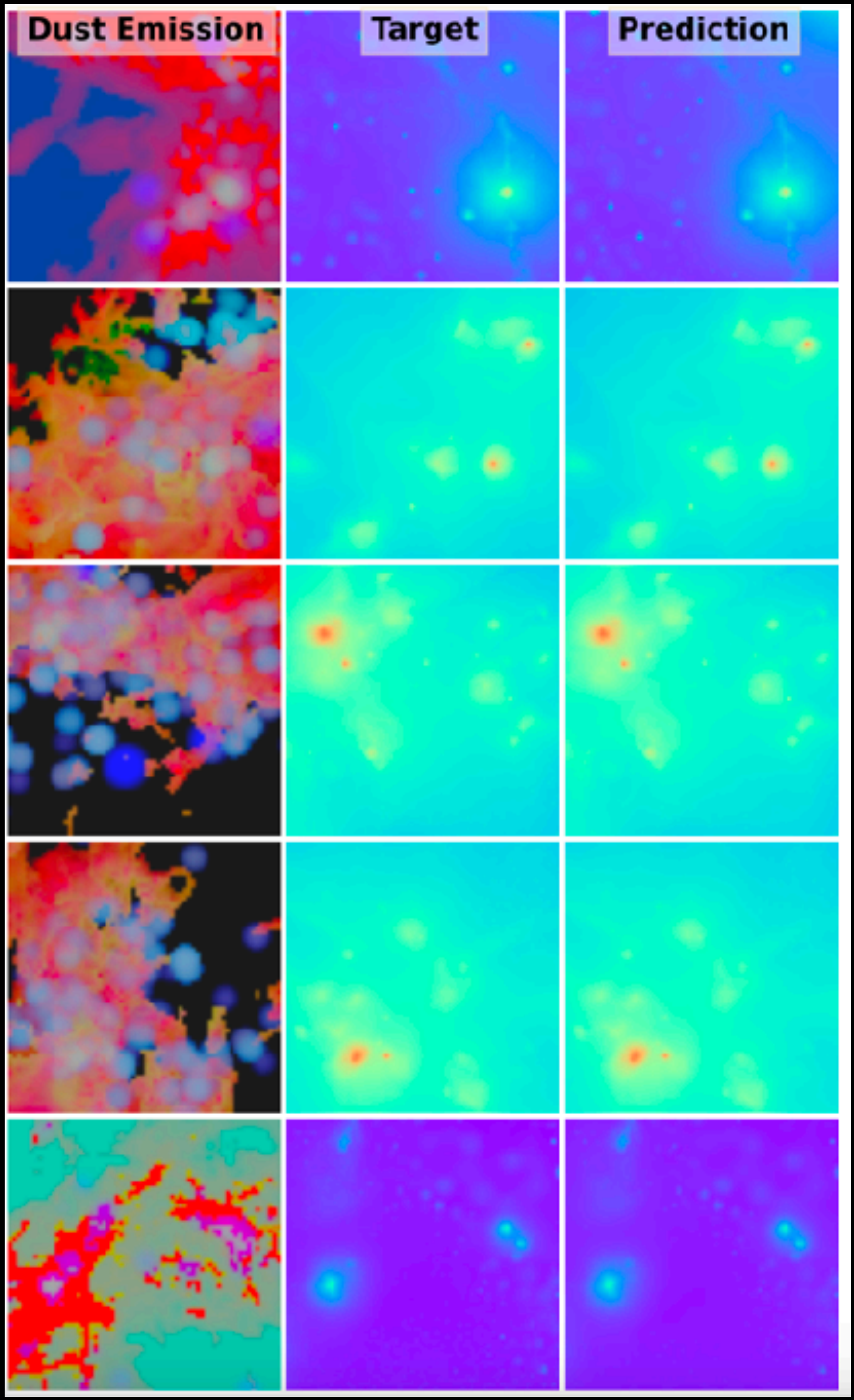
Output: Total radiation field

D. Xu, Offner et al. 2023

Training

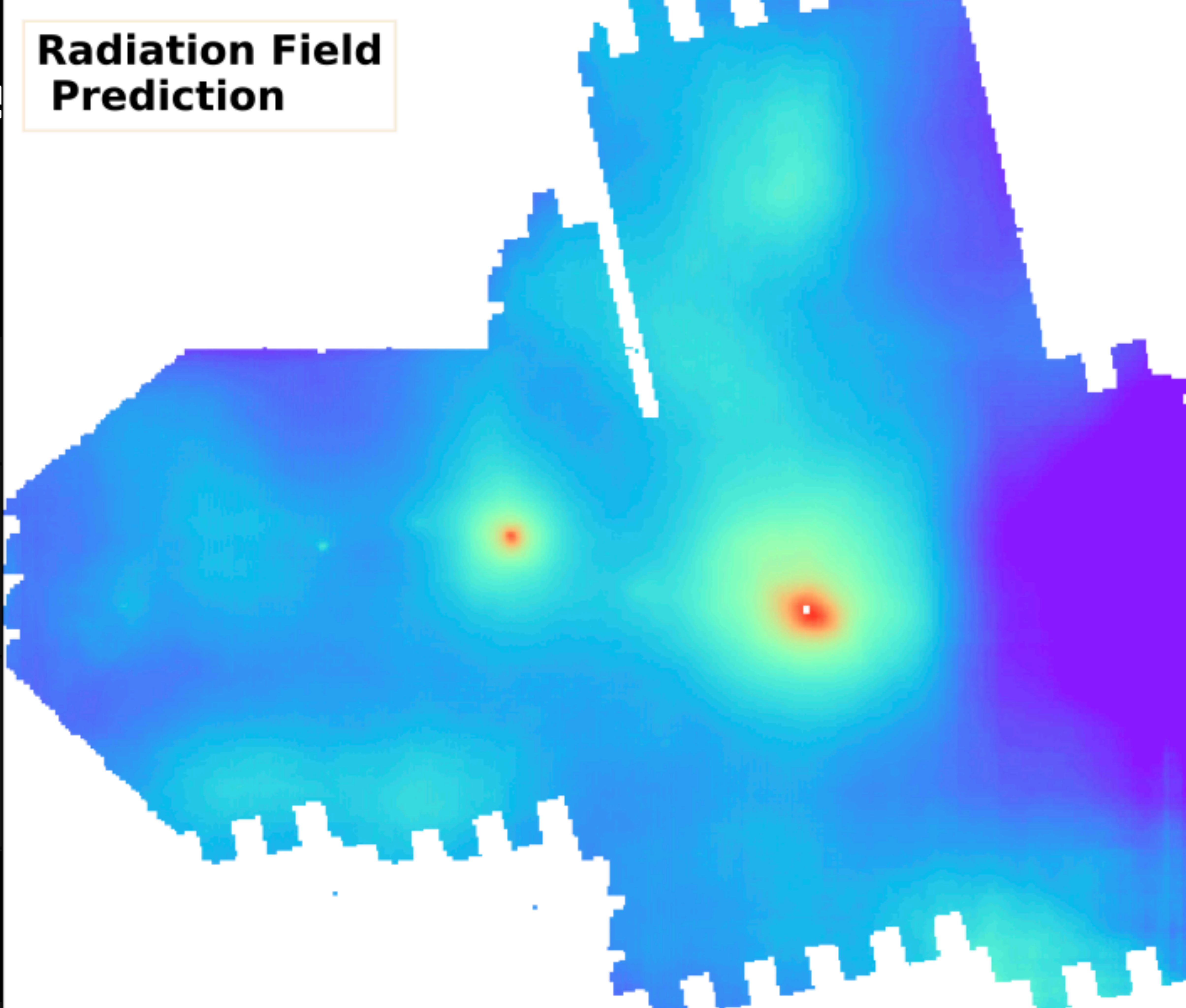
Performance on the test set:





D. Xu, Offner et al. 2023

Pre



Mon R2
Star Formi

Spitzer Infrared

Summary Problem 4: Observe Photons

- Generative AI methods are undergoing a **rapid revolution**
- **Huge unexplored potential** for scientific data analysis
- Diffusion methods can effectively and accurately predict complex physical properties, such as radiation
- Hard to implement but some public codes exist …

Problem 5: Long Evolution Timescales

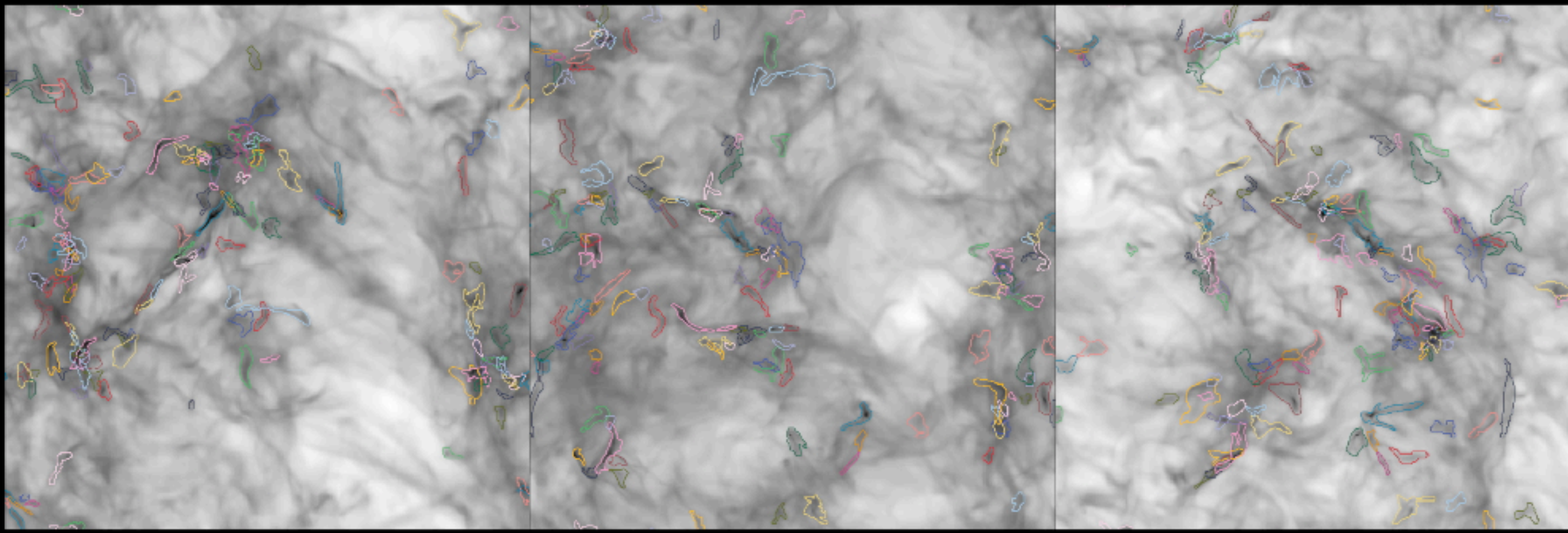
Big Bang Fountain, Olafur Eliasson (2014)

Modeling

Emulators/Accelerating Equation Solutions

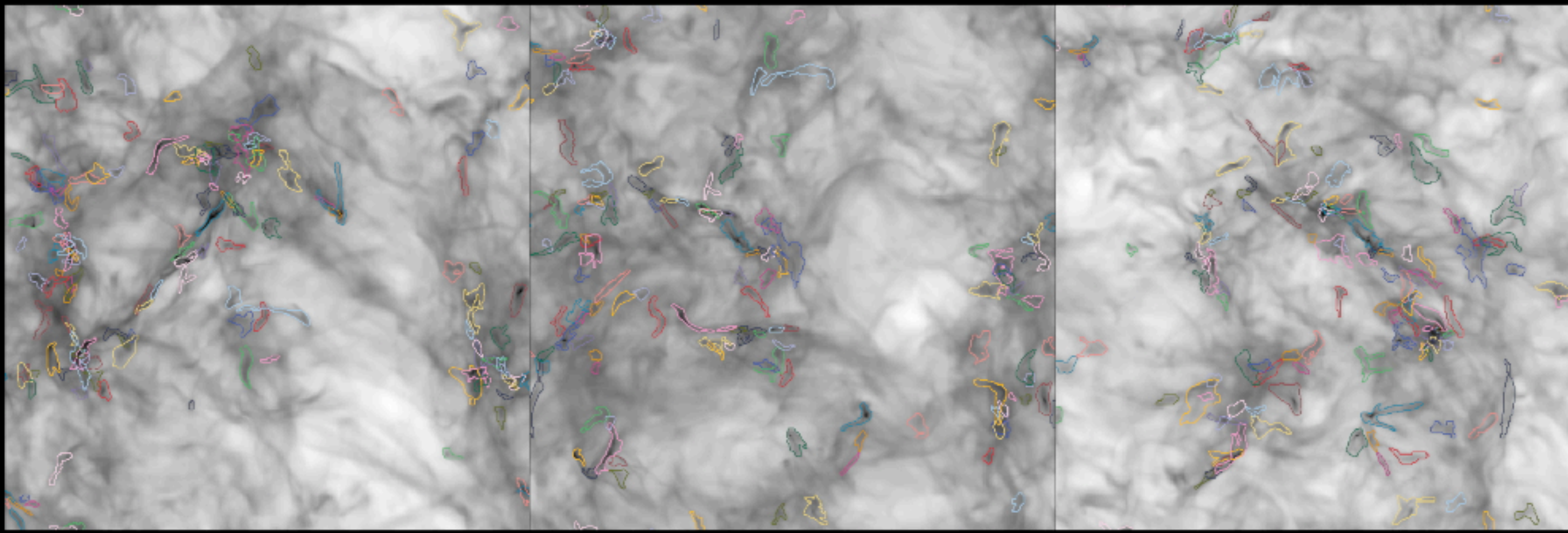
Neural Operators

"Conventional" Partial Differential Equation (PDE) Solver

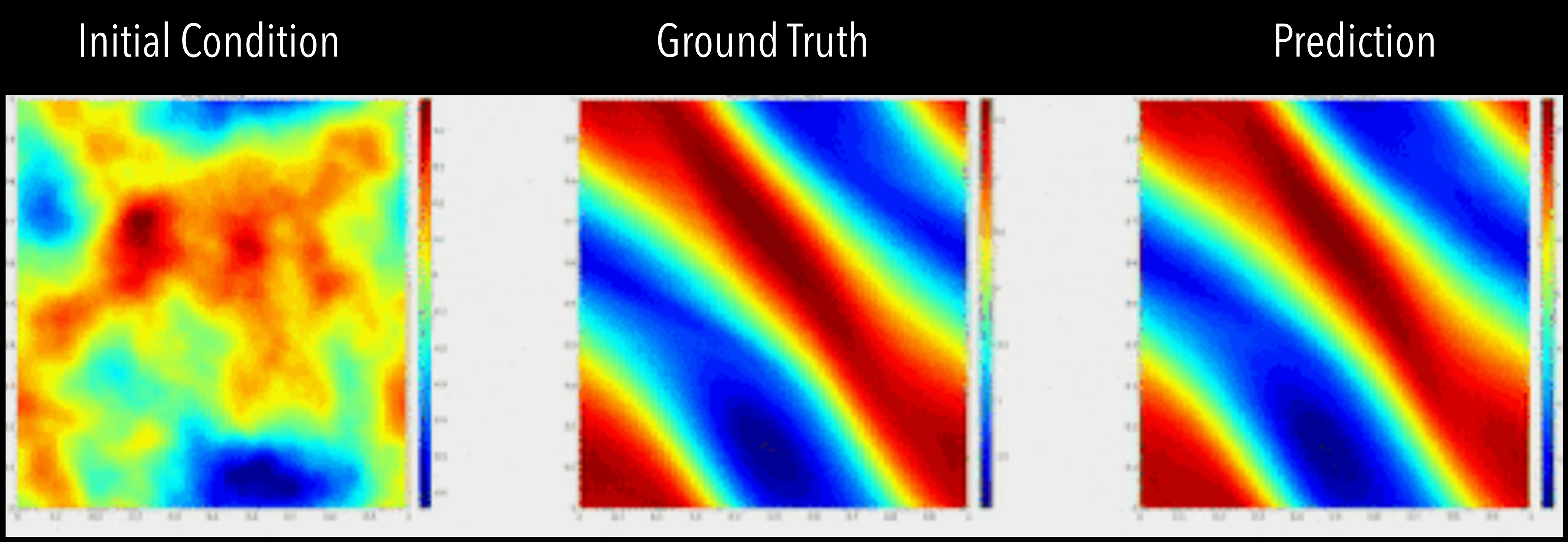


Neural Operators

"Conventional" Partial Differential Equation (PDE) Solver



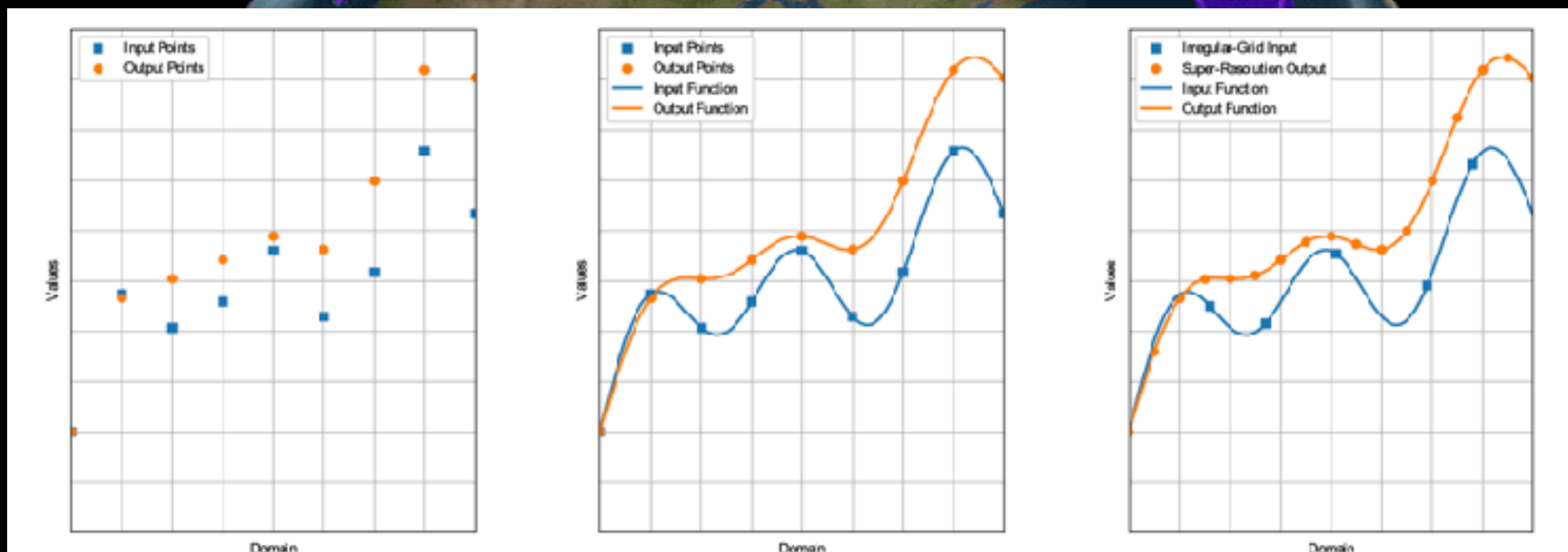
Fourier Neural Operator



Zongyi Li

"Conventional" PDE Solver vs Neural Operators

"Conventional" PDE Solver	Neural Operators
• Solve one instance.	Learn a family of PDEs
• Require an explicit form.	Data-driven, but black box
• Speed/accuracy trade-off in resolution.	Resolution & mesh invariant
• Slow on fine grids, fast on coarse.	Slow to train, fast to evaluate
• Need simple, well-posed initial conditions.	Does not.



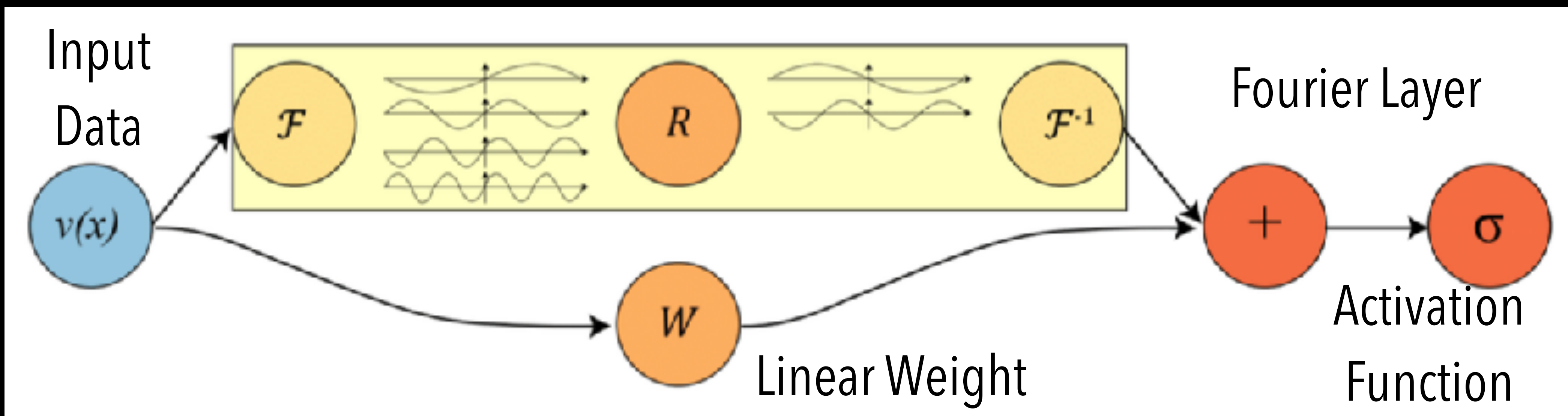
(a) NN learns a mapping between input and output points on a fixed, discrete grid.

(b) NO maps between functions on continuous domains, even when training data is on a fixed grid.

(c) NO maps between functions, so it accepts inputs outside the training grid, and can do super-resolution.

Azzizadenesheli et al. 2023

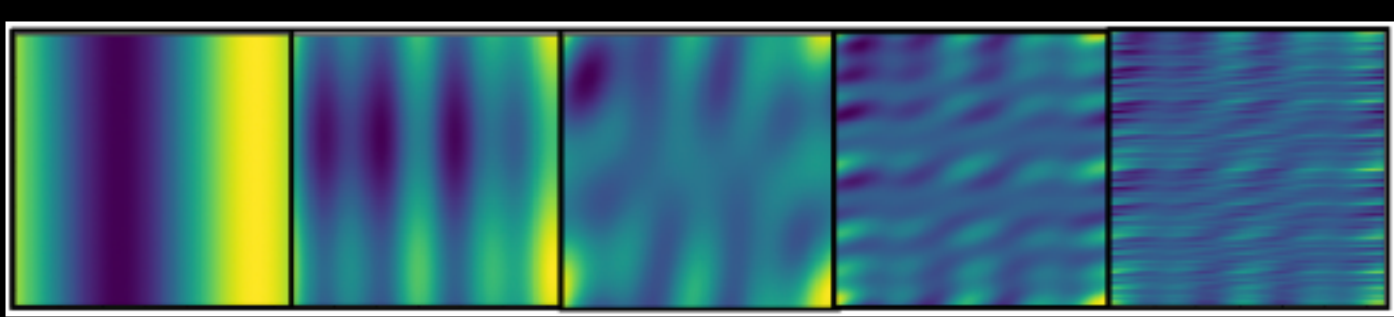
Fourier Neural Operators



Replace Convolution with FFT + LT + IFT

CNN Filters

Equivalent Fourier Filters

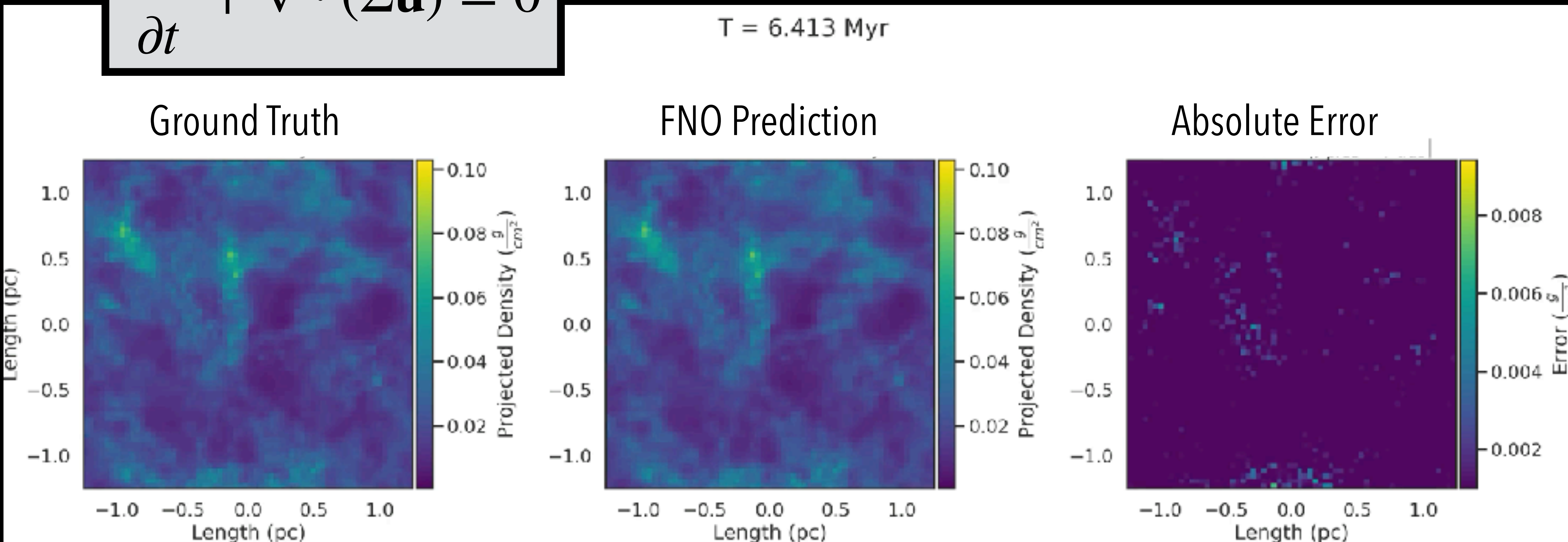


- FFTs are fast
- More efficient to represent continuous functions in Fourier space

Li et al. 2020

FNO Prediction of Supersonic Turbulence

$$\frac{\partial \Sigma}{\partial t} + \nabla \cdot (\Sigma \mathbf{u}) = 0$$



- Training: 11,900 initial conditions (sets), 60 turbulent seeds
- Trained on column density, on 5 time steps with dt= 8kyr
- <=10% for 5 consecutive steps

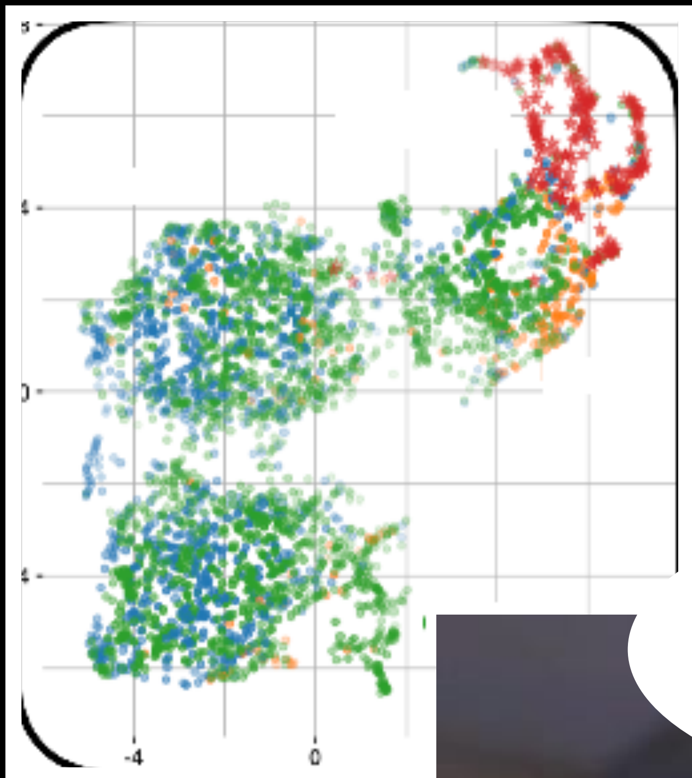
Keith Poletti

Poletti et al. Sub.

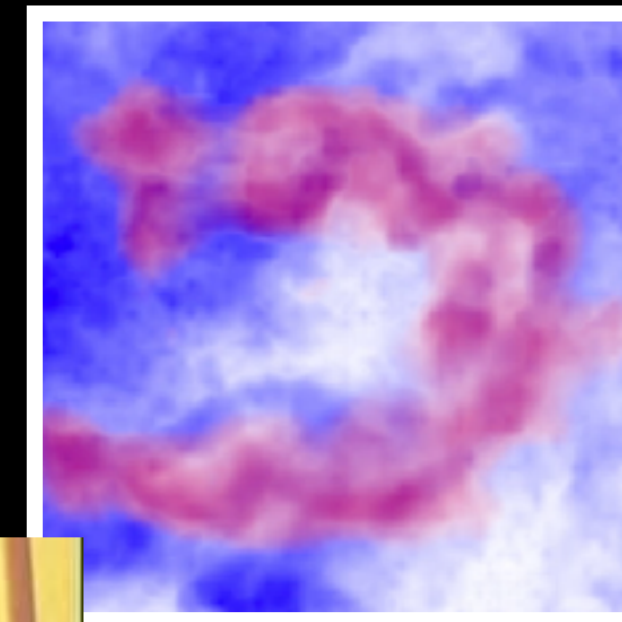
Summary Problem 5: Long Timescales

- Neural operators have **significant potential** to replace classic PDE solvers, including gravity, hydrodynamics, radiative transfer …
- Very fast but **requires extensive training data**
- Whether these techniques can reach the needed accuracy for modeling high-dimensional data is still TBD
- Hard to implement but some public codes exist …

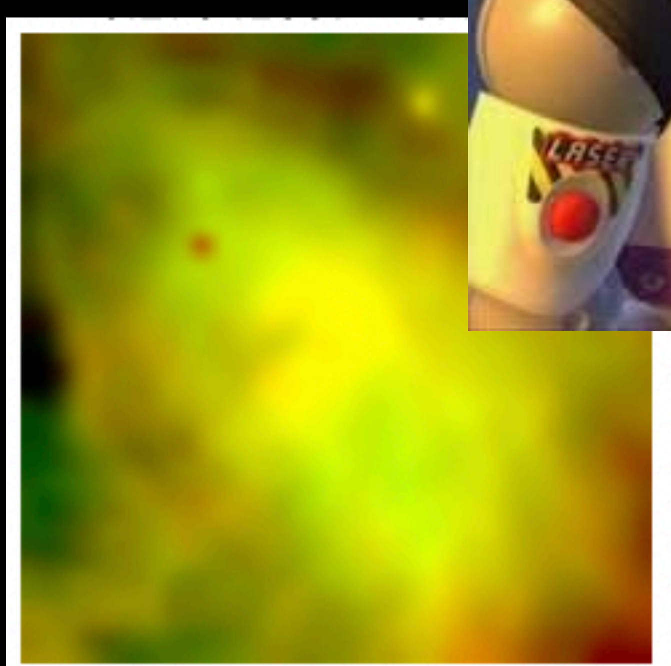
Conclusions



UMAP

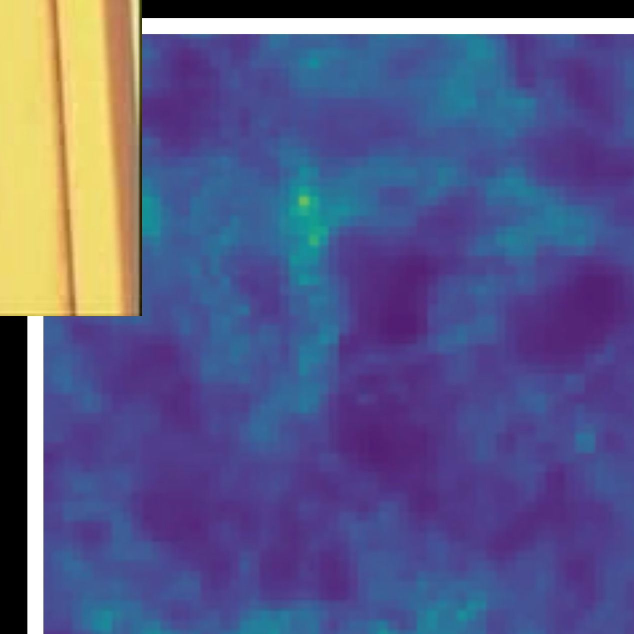


Neural Networks



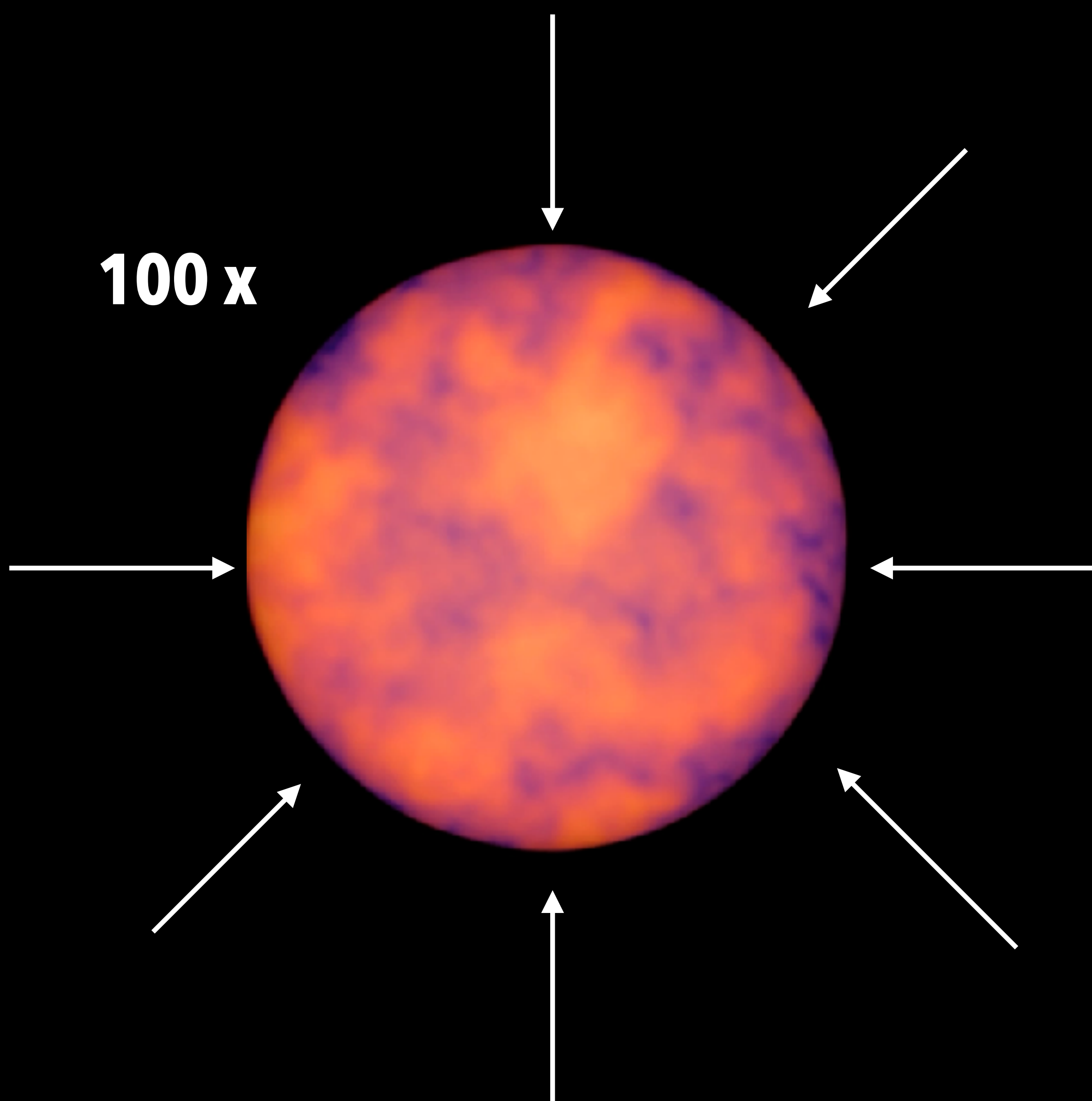
Random Forests

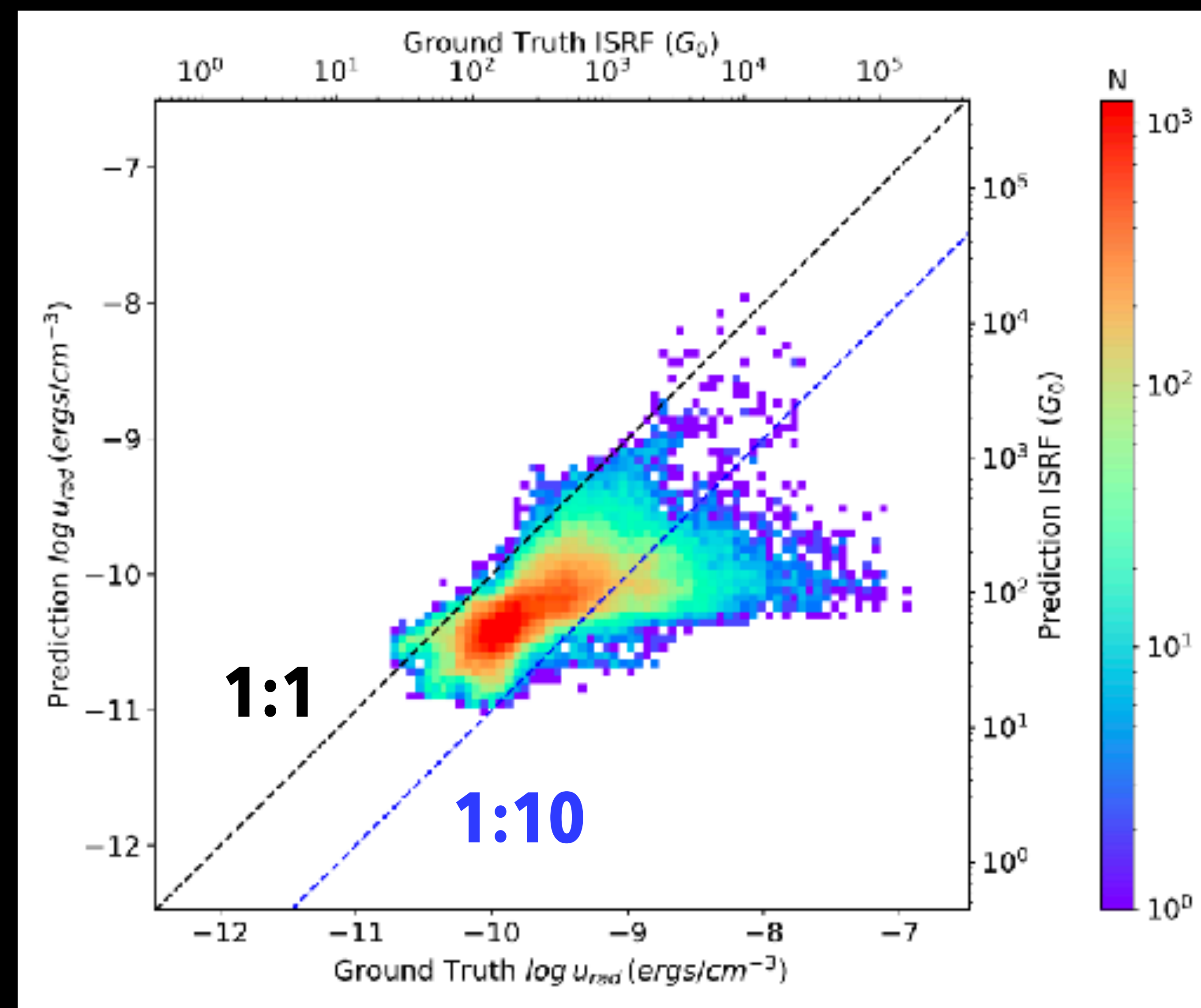
Neural Operators



More Testing: Out of Distribution Data

Performance on simulated data with 100 times higher background radiation





Predicted radiation is ~ 3x too low

D. Xu, Offner et al. 2023