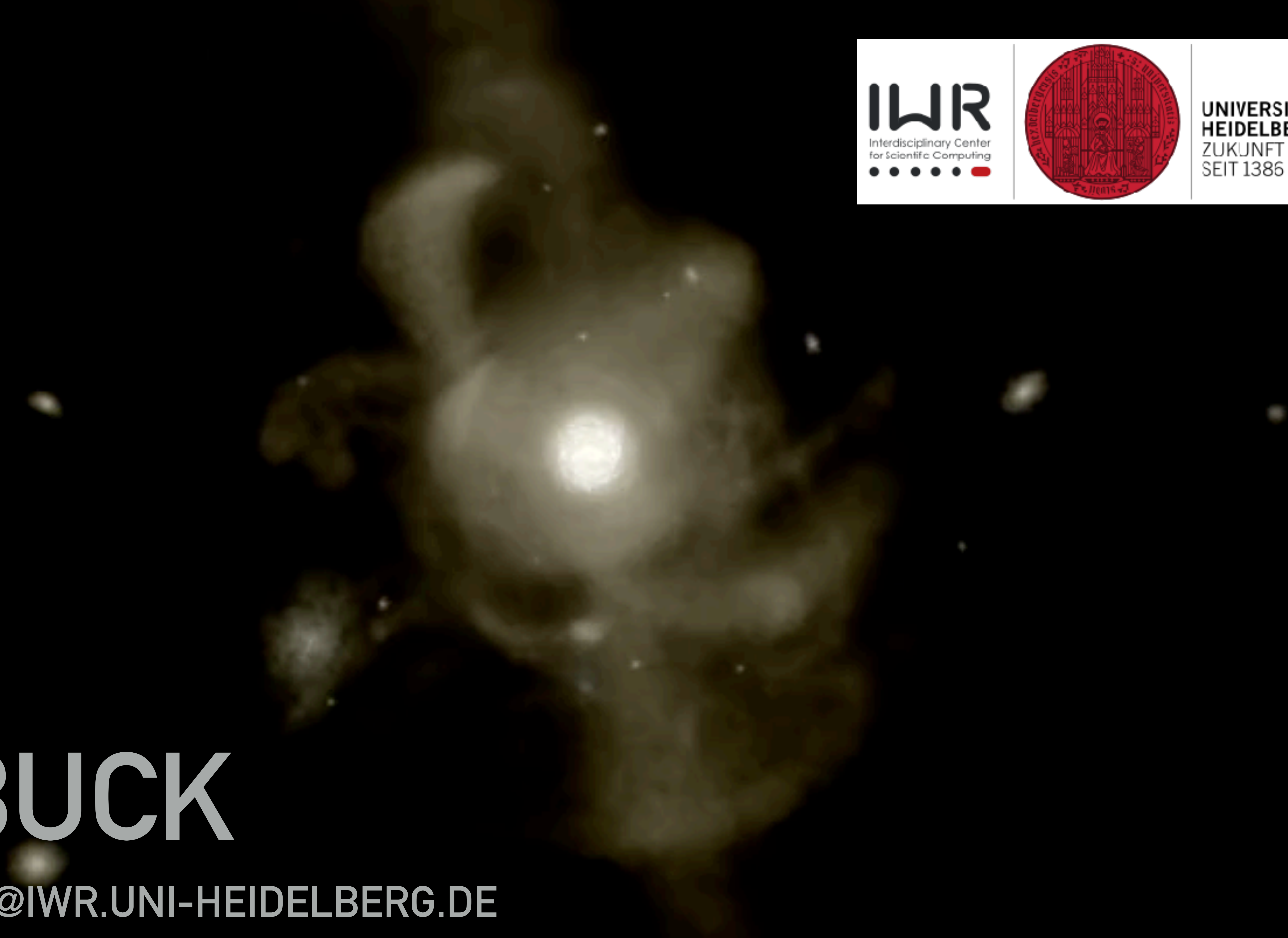


TOBIAS BUCK

25.9.2024 TOBIAS.BUCK@IWR.UNI-HEIDELBERG.DE

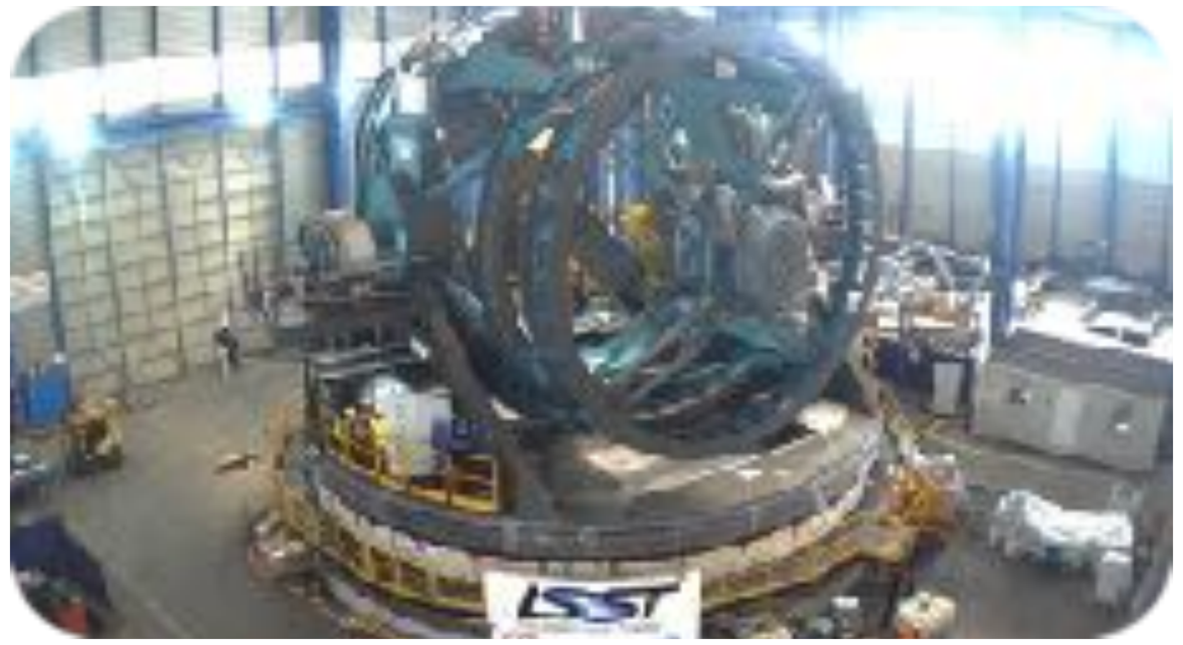
MACHINE LEARNING IN ASTROPHYSICS

MODERN COSMOLOGY AND ASTROPHYSICS

SKA 2.3 billion USD

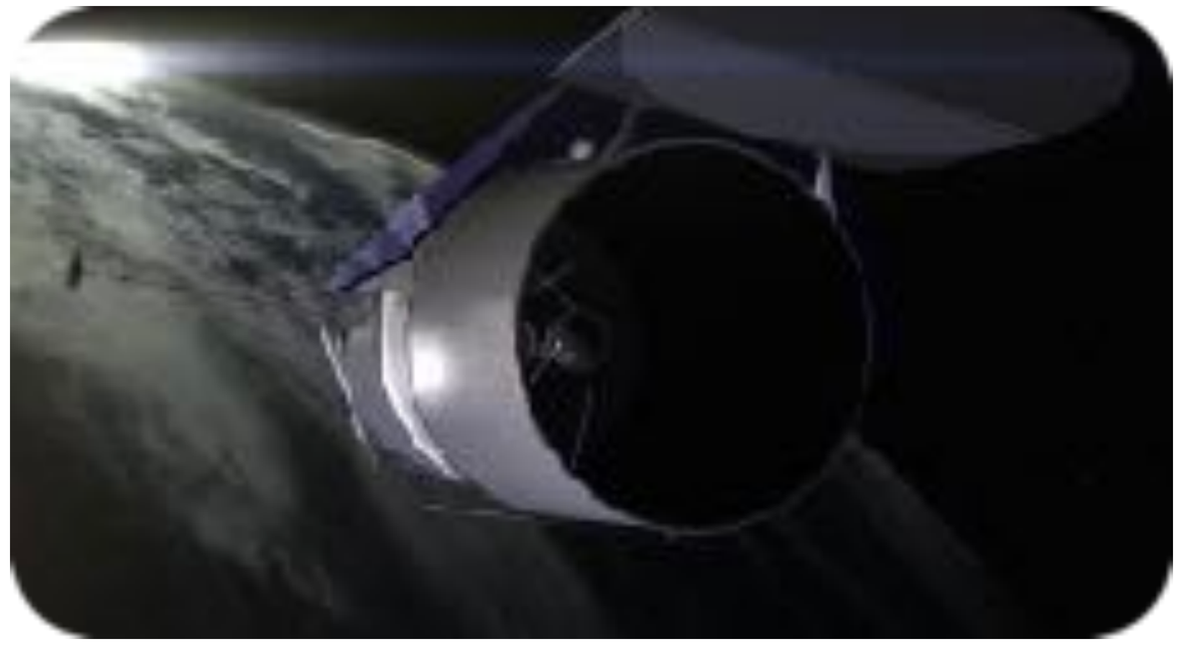
TMT 2.4 billion USD

JWST 10 billion USD



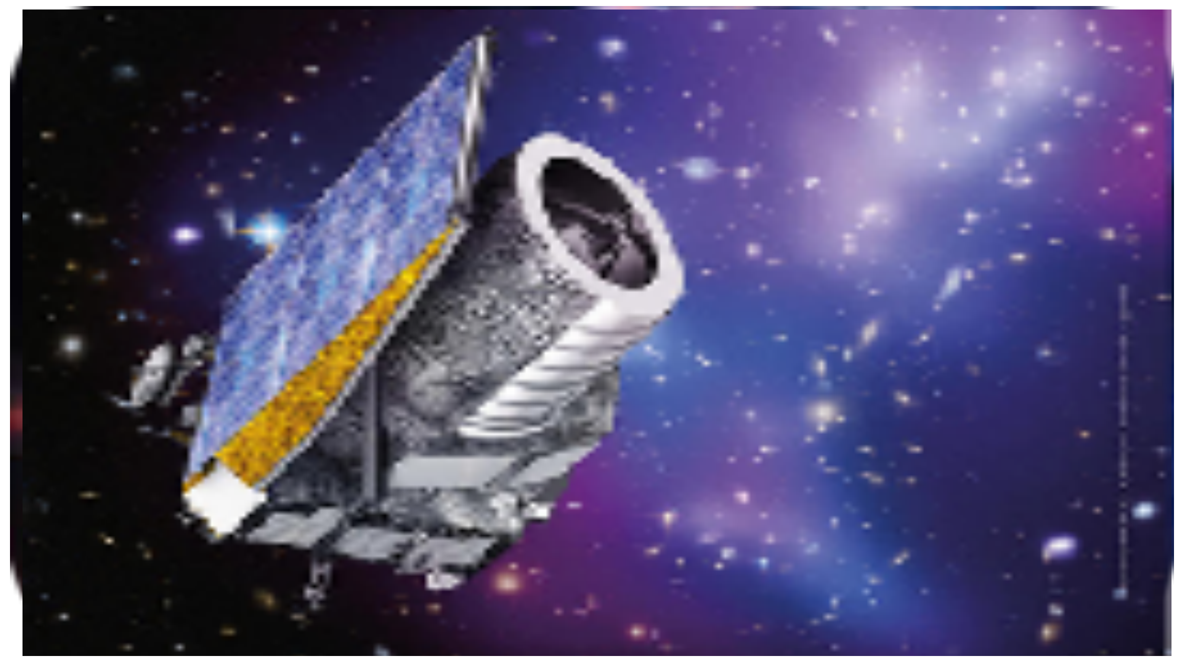
LSST 2 billion USD

ALMA 1.3 billion USD



WFIRST 3.2 billion USD

ATHENA 1 billion USD



Euclid 0.7 billion USD

DATA IS GETTING MORE AND MORE COMPLEX!

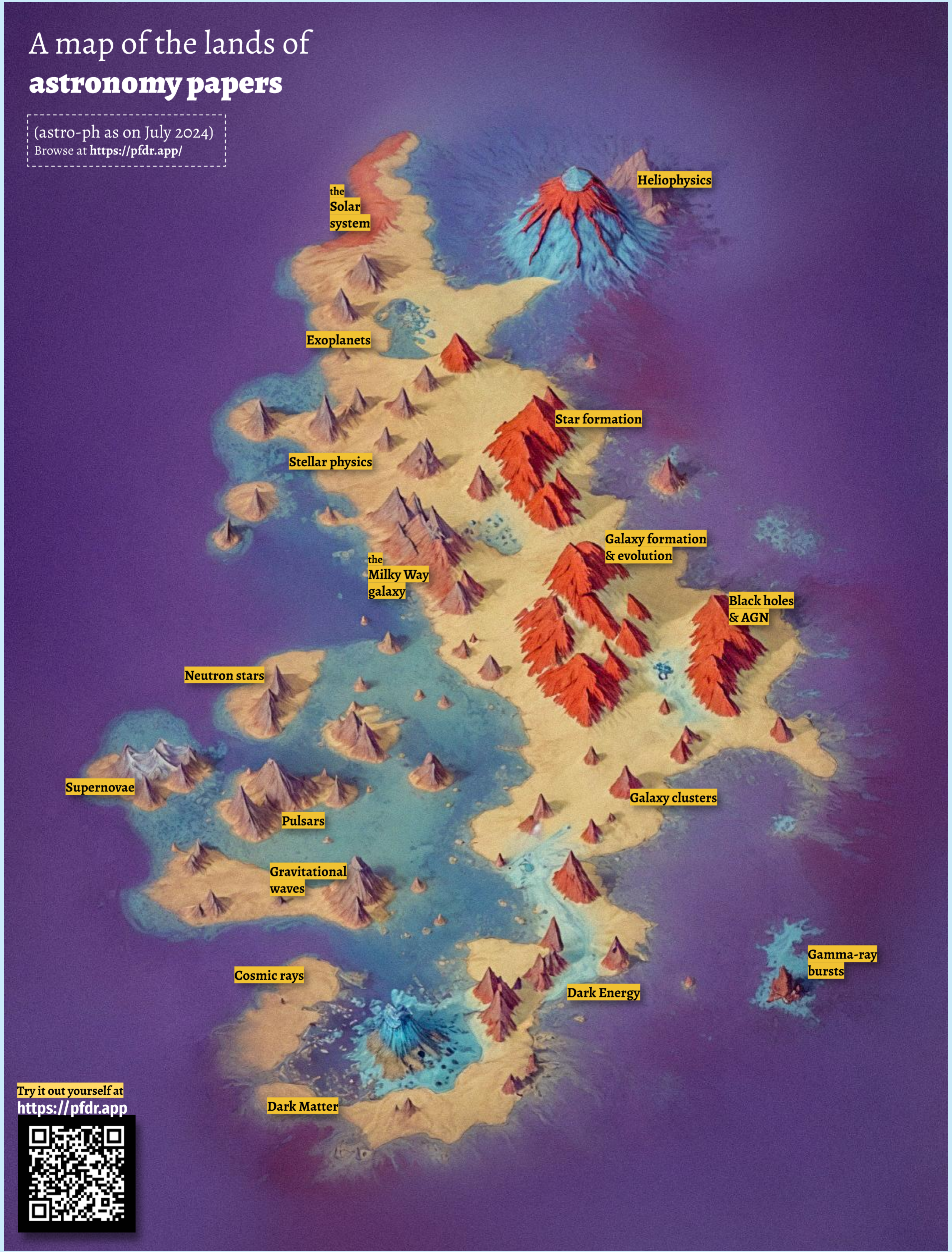
THIS MEANS: NEW OPPORTUNITIES BUT ALSO NEW CHALLENGES

AI ASSISTED PROGRAMMING

GitHub
Copilot

LARGE LANGUAGE MODELS IN ASTROPHYSICS

A public-friendly visualization of the 2d manifold of galaxy evolution papers created with UMAP+stable diffusion that shows the different areas of the astro-ph literature corpus. Following similar patterns as the heatmap, mountains indicate well-studied areas, plains indicate fields of active study, coastal regions are ‘hot topics’, and water denotes regions with no papers. Similar to a world map, the axes here do not hold a particular meaning. Regions close to each other have semantic similarity, while distant regions do not.



Iyer+2024, Pathfinder (UniverseTBD)

OUTLINE

1. Unsupervised Learning
   A. Clustering
   B. Representation Learning
2. Inference / Bayesian Modelling
   C. Modeling Complex Prob. Distributions
   D. Simulation-Based Inference
   E. Model Comparison & Model Misspecification
3. Forward Models and Emulators
   F. Neural ODEs and Operator Learning and PINNs

Unsupervised Clustering

CLUSTERING AND DIMENSIONALITY REDUCTION

most of the time we think in categories: stars vs. galaxies, elliptical vs. spirals, halos vs. filaments vs. voids

how to find/exploit structure in the data?

unsupervised clustering: Gaussian Mixture Models, k-means, HDBSCAN, … in general any halo finder

ASTROLINK

Oliver+2024

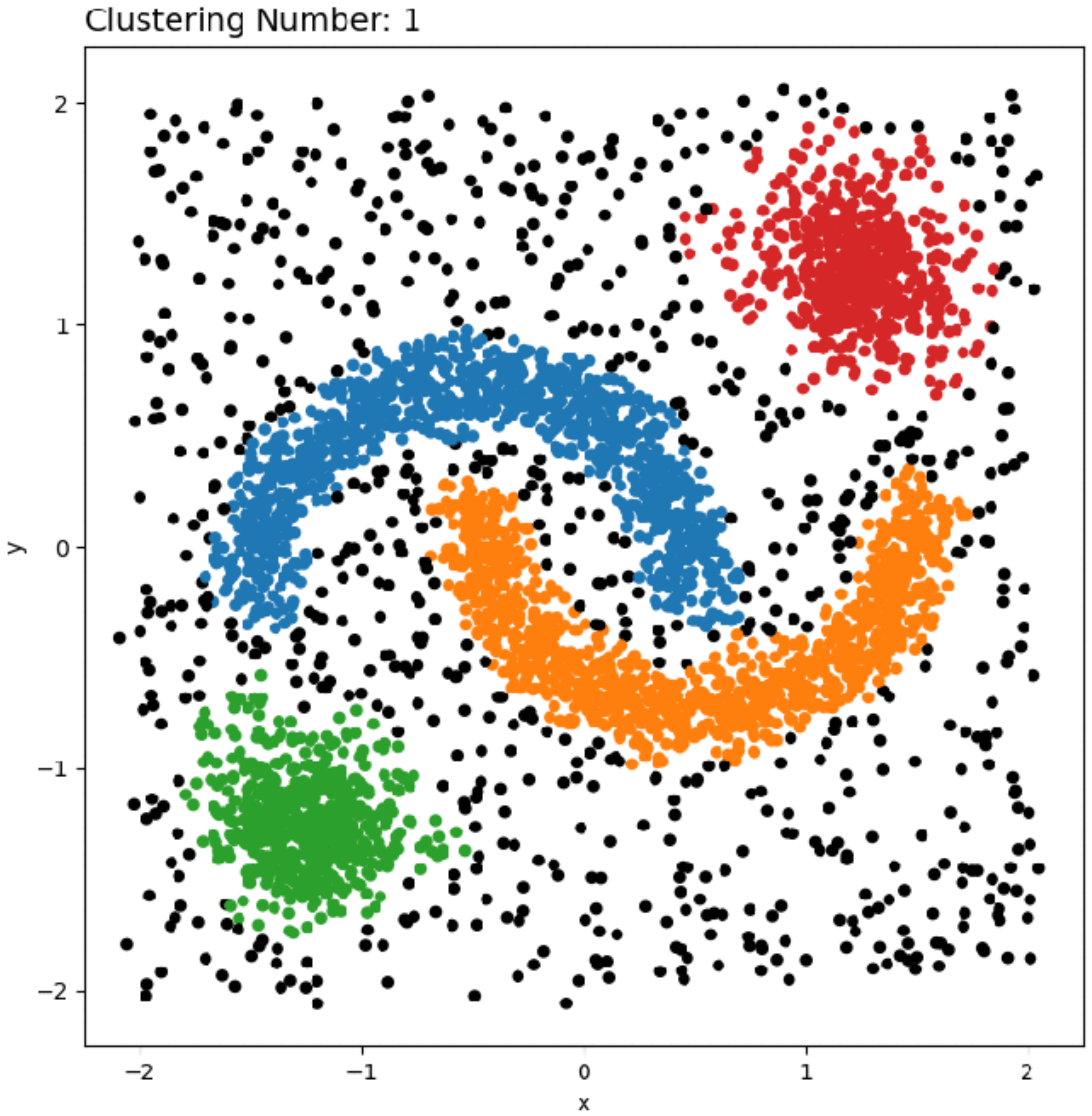
https://astrolink.readthedocs.io

CLUSTERING AND DIMENSIONALITY REDUCTION

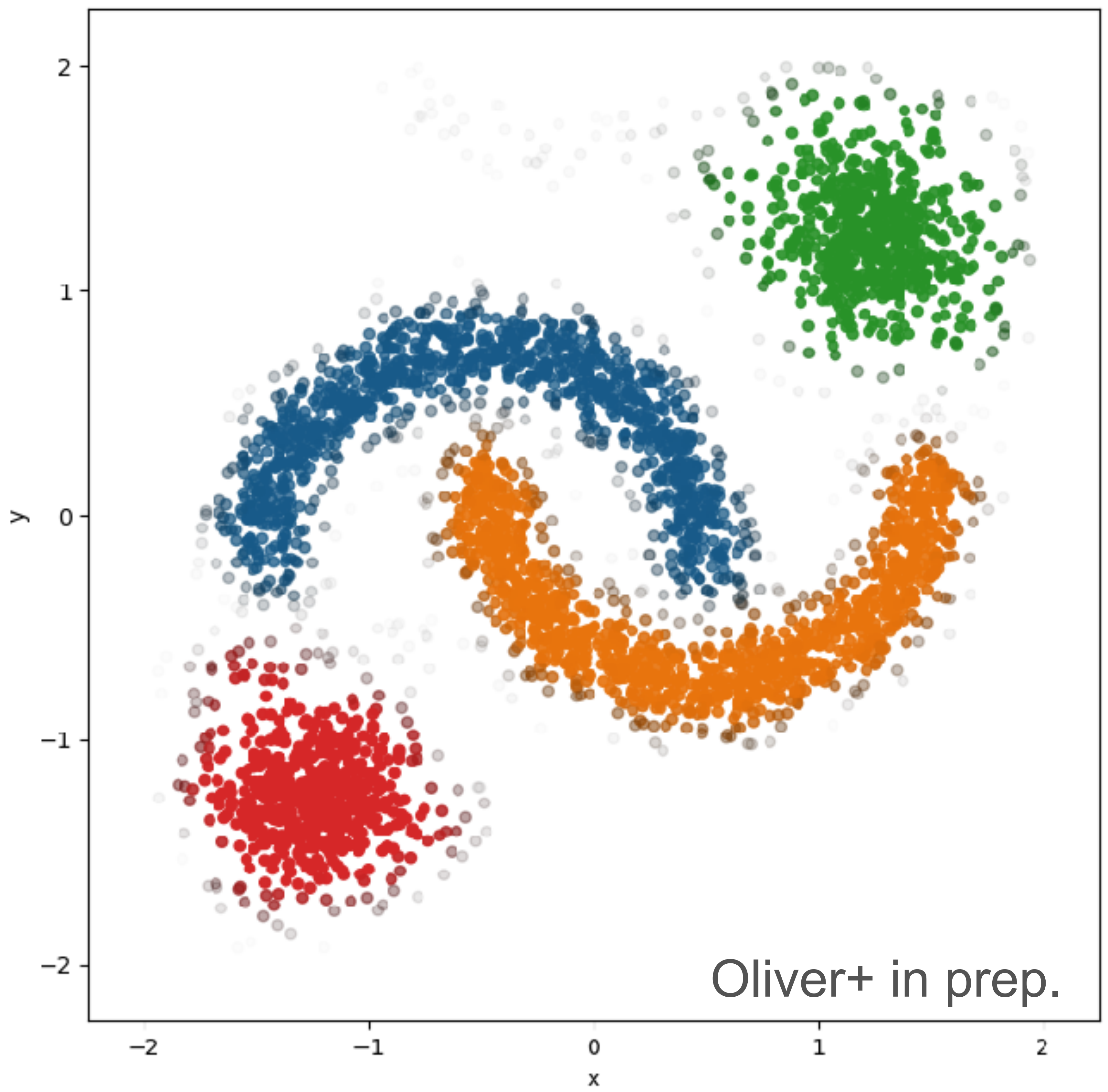
most of the time we think in categories: stars vs. galaxies, elliptical vs. spirals, halos vs. filaments vs. voids

how to find/exploit structure in the data?

What about measurement uncertainties or time evolving data?



https://fuzzycat.readthedocs.io

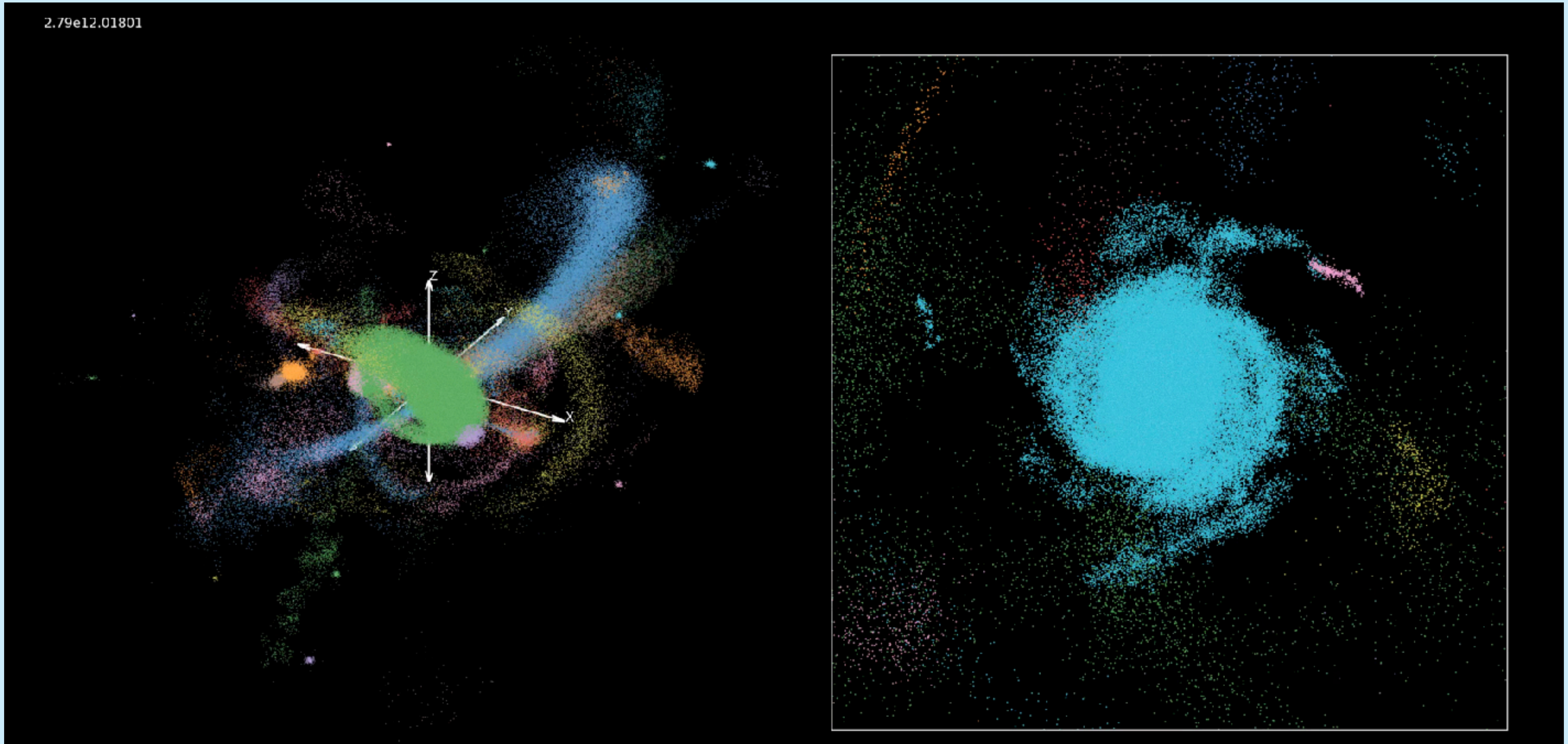


CLUSTERING AND DIMENSIONALITY REDUCTION

most of the time we think in categories: stars vs. galaxies, elliptical vs. spirals, halos vs. filaments vs. voids

how to find/exploit structure in the data?

What about measurement uncertainties or time evolving data?



https://fuzzycat.readthedocs.io

Oliver+ in prep.

REPRESENTATION LEARNING

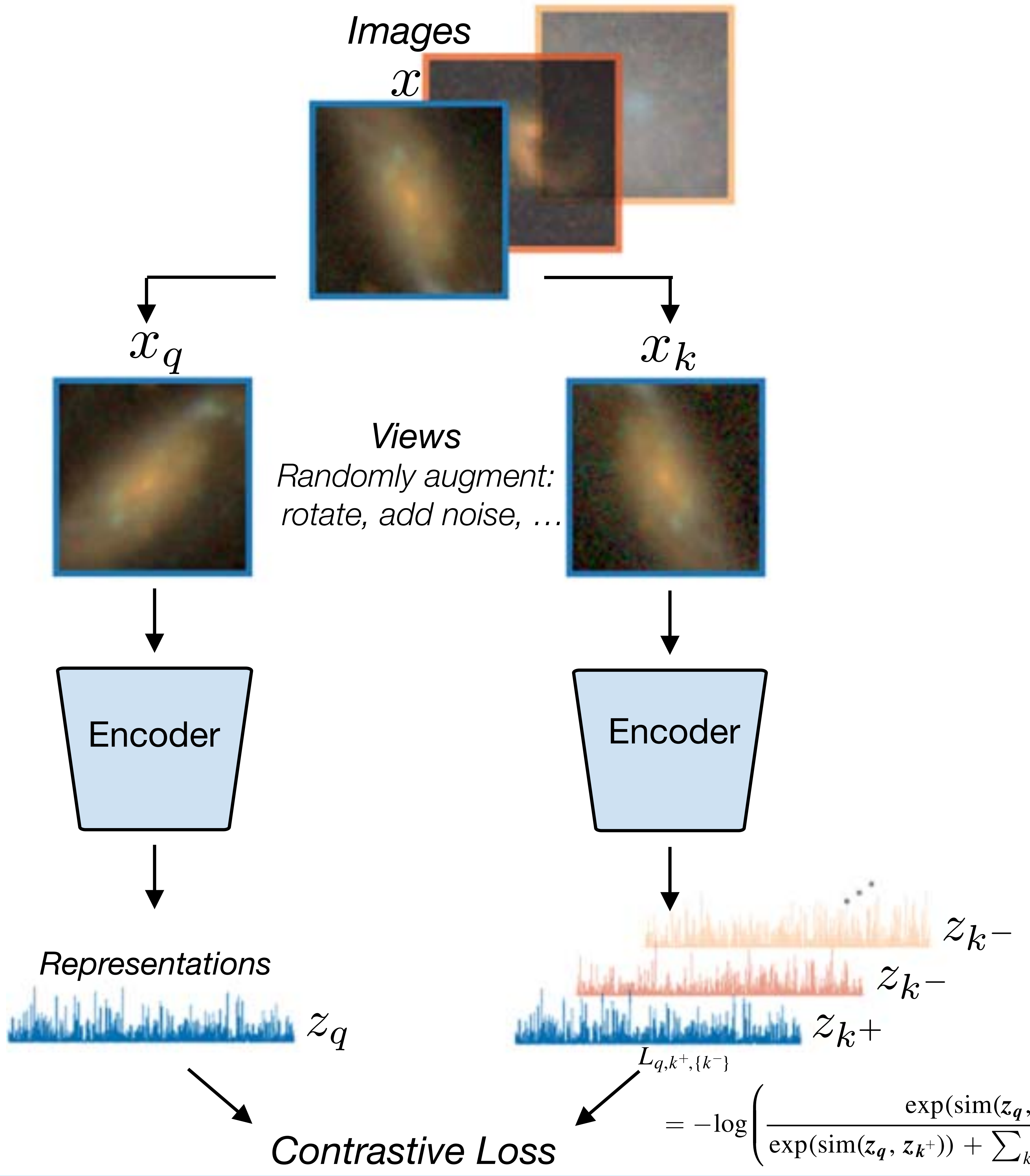
representation learning seeks to automatically discover the representations needed for feature detection or classification from raw data.

image credit: https://blog.fastforwardlabs.com/2020/11/15/representation-learning-101-for-software-engineers.html

REPRESENTATION LEARNING

1. Self-supervised contrastive representation learning

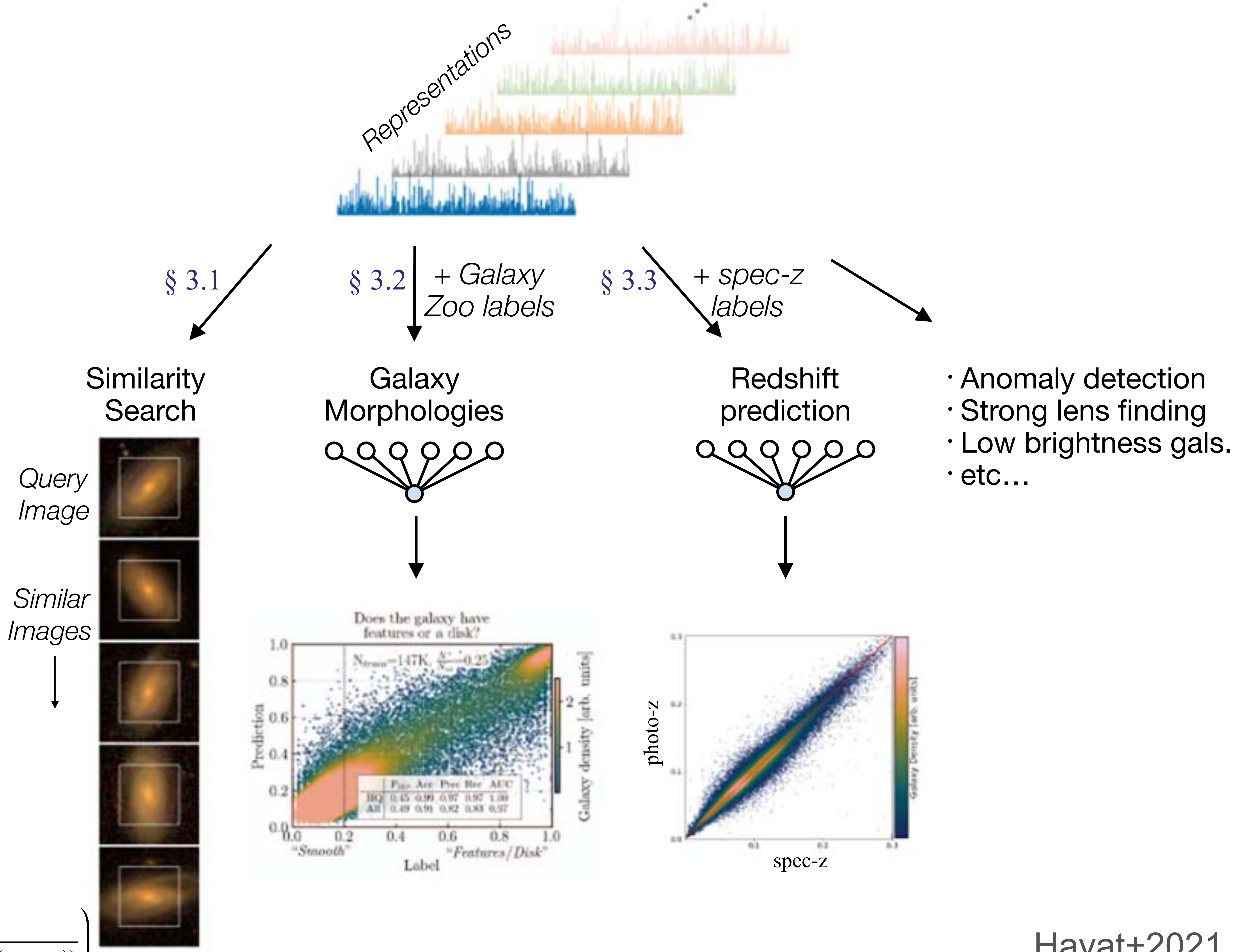
Learn representations in an unsupervised manner



$$= -\log\left(\frac{\exp(\mathrm{sim}(z_q, z_{k^+}))}{\exp(\mathrm{sim}(z_q, z_{k^+})) + \sum_{k^-}\exp(\mathrm{sim}(z_q, z_{k^-}))}\right)$$

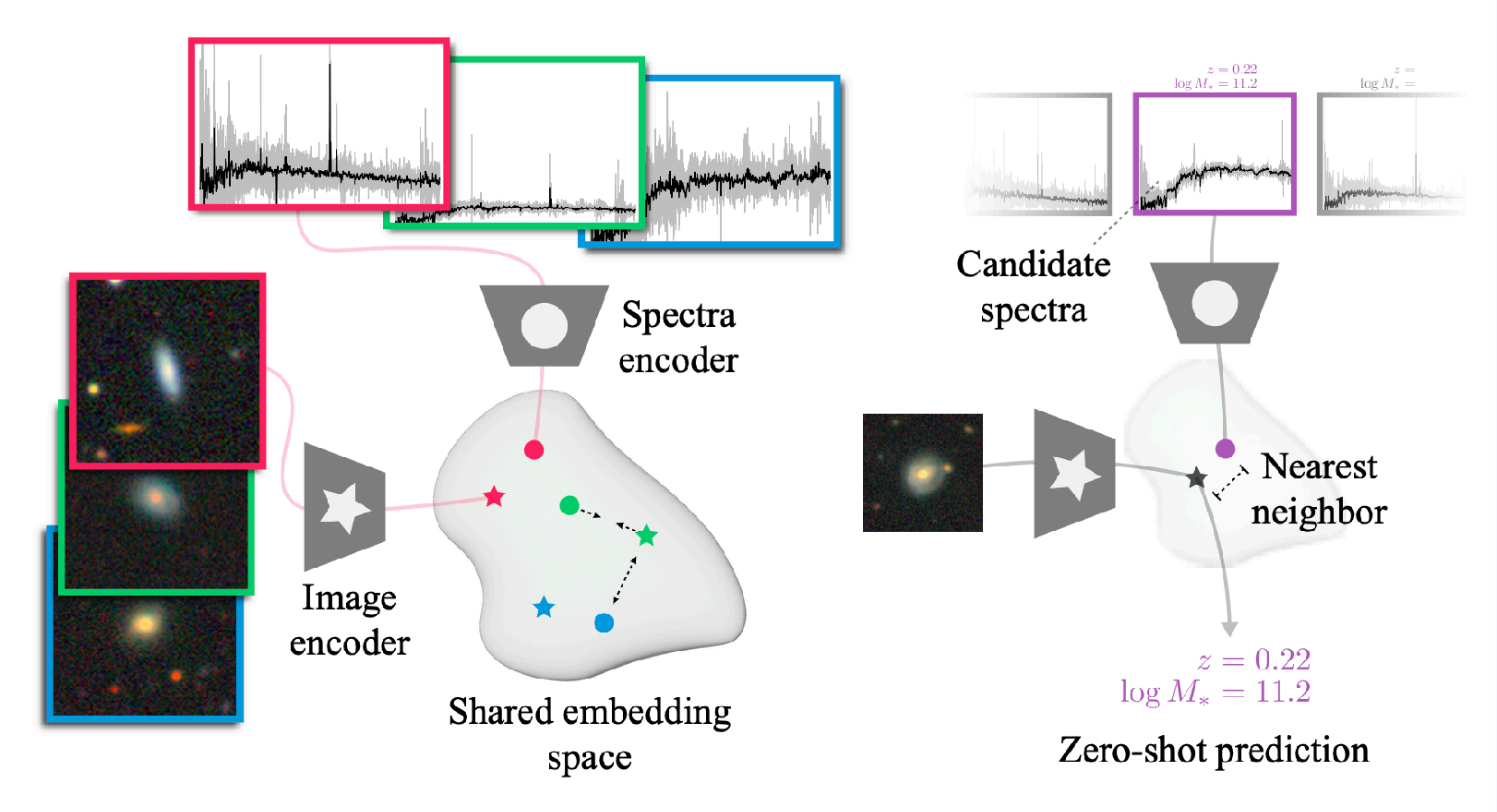
2. Downstream tasks

Use representations for a variety of applications



Hayat+2021

REPRESENTATION LEARNING



Parker+2024

REPRESENTATION LEARNING FOR STELLAR SPECTRA

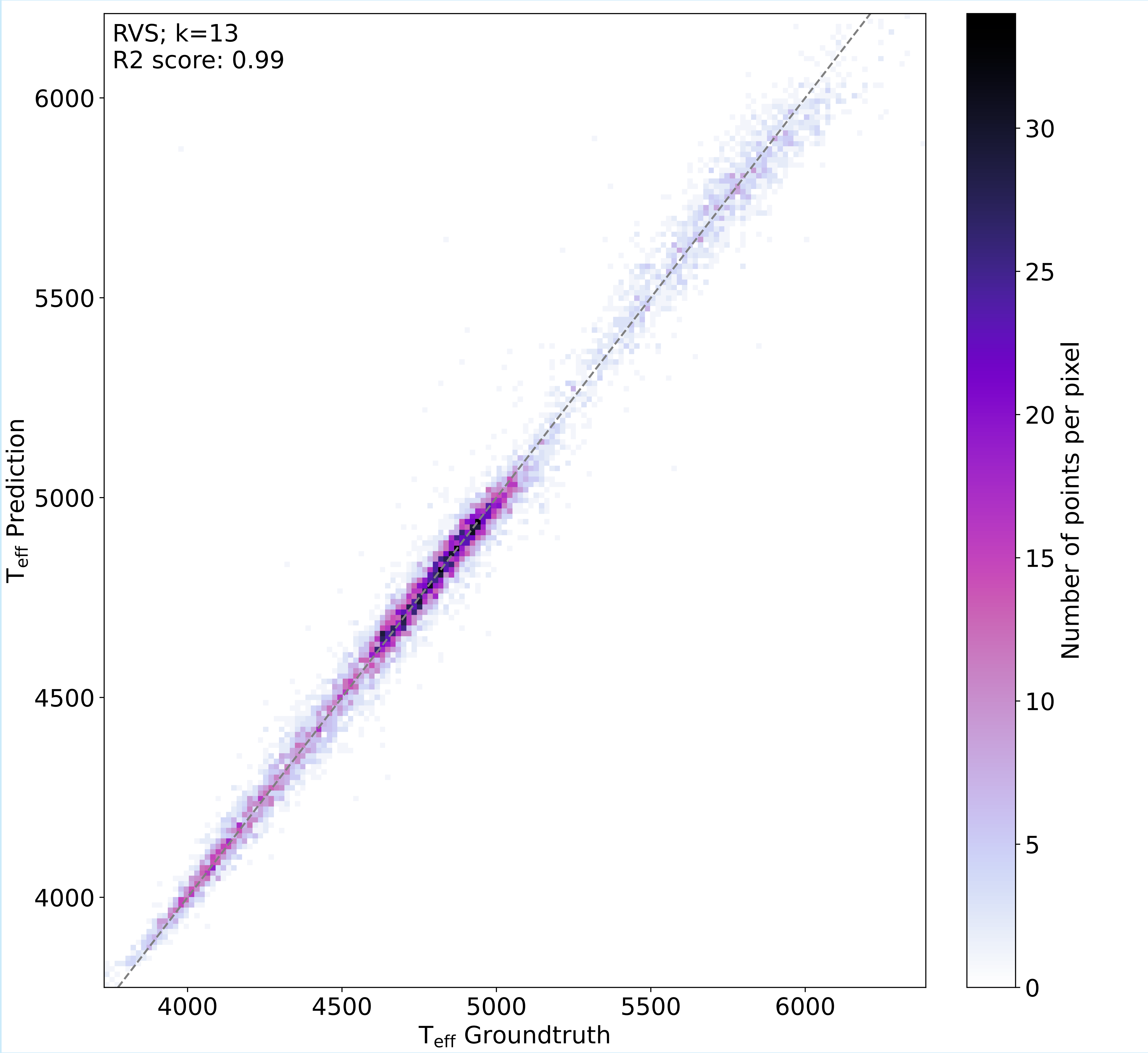
multi-modal data: Gaia RVS spectra and Gaia XP coefficients plus contrastive loss

$$\mathcal{L}_{CL} = -\frac{1}{N}\sum_{i}^{N} log\frac{exp(x_i^T y_i/\tau)}{\sum_{j=1}^{N} exp(x_i^T y_j/\tau)} - \frac{1}{N}\sum_{i}^{N} log\frac{exp(y_i^T x_i/\tau)}{\sum_{j=1}^{N} exp(y_i^T x_j/\tau)}$$

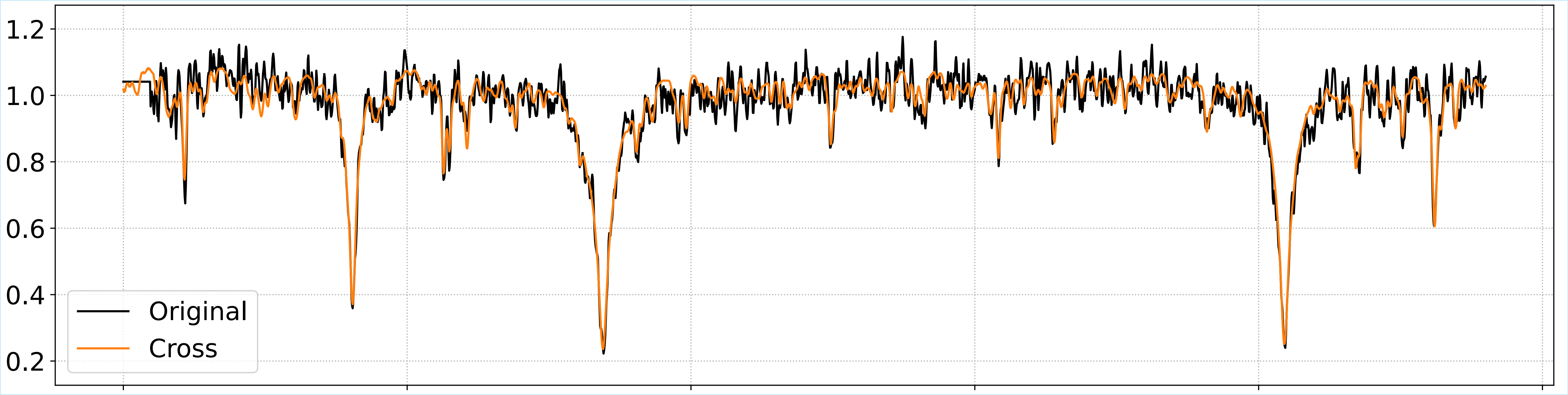
Buck+2024 in prep., see also Parker+2024 for AstroCLIP

REPRESENTATION LEARNING FOR STELLAR SPECTRA

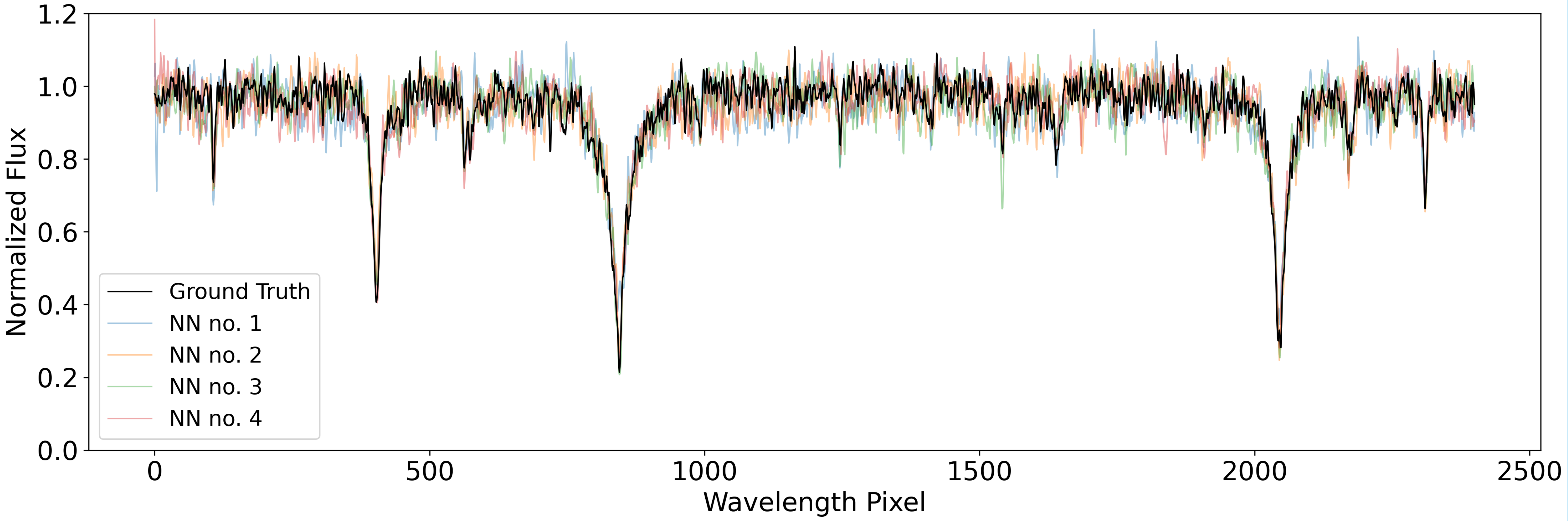
regression



cross-modal generation



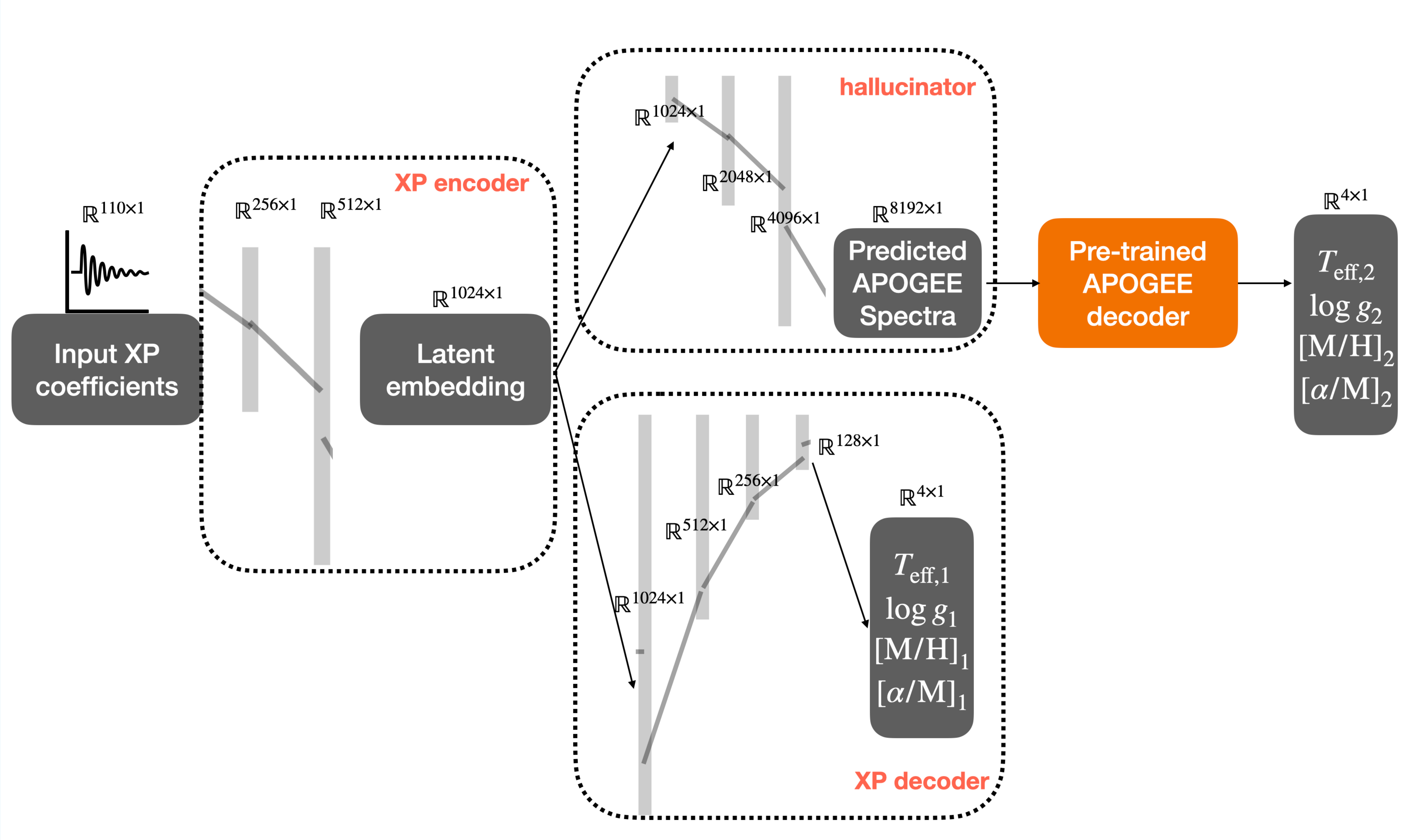
cross-modal look-up



Buck+2024 in prep.

REPRESENTATION LEARNING FOR STELLAR SPECTRA

cross-modal generation: AspGap



Li,Wong,Hogg+2024

REPRESENTATION LEARNING

Literature: Huertas-Company & Sarmiento 2023

A brief review of contrastive learning applied to astrophysics

Marc Huertas-Company[1,2,3,4,5]★, Regina Sarmiento[1,2] and Johan H. Knapen[1,2]
[1] *Instituto de Astrofísica de Canarias (IAC), La Laguna, E-38205, Spain*
[2] *Departamento de Astrofísica, Universidad de La Laguna (ULL), E-38200, La Laguna, Spain*
[3] *Observatoire de Paris, LERMA, PSL University, 61 avenue de l'Observatoire, F-75014 Paris, France*
[4] *Université Paris-Cité, 5 Rue Thomas Mann, 75014 Paris, France*
[5] *Center for Computation Astrophysics, Flatiron Institute, 162 5th avenue, 10015 New York, USA*

Accepted XXX. Received YYY; in original form ZZZ

ABSTRACT
Reliable tools to extract patterns from high-dimensionality spaces are becoming more necessary as astronomical datasets increase both in volume and complexity. Contrastive Learning is a self-supervised machine learning algorithm that extracts informative measurements from multi-dimensional datasets, which has become increasingly popular in the computer vision and Machine Learning communities in recent years. To do so, it maximizes the agreement between the information extracted from augmented versions of the same input data, making the final representation invariant to the applied transformations. Contrastive Learning is particularly useful in astronomy for removing known instrumental effects and for performing supervised classifications and regressions with a limited amount of available labels, showing a promising avenue towards *Foundation Models*. This short review paper briefly summarizes the main concepts behind contrastive learning and reviews the first promising applications to astronomy. We include some practical recommendations on which applications are particularly attractive for contrastive learning.

Key words: methods: data analysis – methods: statistical – methods: miscellaneous – techniques: miscellaneous

Inference and Bayesian Modelling

MODELING COMPLEX PROBABILITY DISTRIBUTIONS

Bayesian inference aims at determining $p(\theta \,|\, \mathbf{x_o})$

$$p(\theta \,|\, \mathbf{x_0}) = \frac{p(\mathbf{x_0} \,|\, \theta) p(\theta)}{p(\mathbf{x_0})} \propto p(\mathbf{x_0} \,|\, \theta) p(\theta)$$

In astrophysics, $\mathbf{x_o}$ typically results from a large number of mechanisms/effects that transform the data and involve a large number of latent variables z, hence the marginal likelihood $p(\mathbf{x_o} \,|\, \theta)$ is intractable.

$$p(\mathbf{x_0} \,|\, \theta) = \int p(\mathbf{x_0} \,|\, \theta, z) p(z) dz$$

MODELING COMPLEX PROBABILITY DISTRIBUTIONS

How can we approximate high-dimensional, complex probability distributions $p(\theta \mid \mathbf{x_o})$?

Goal:

- effectively: learn a model from the data!
- model $p(\theta \mid \mathbf{x_o})$ explicitly or implicitly
- sample and evaluate $p(\theta \mid \mathbf{x_o})$

Options:

- normalizing flows
- VAEs
- GANs
- score matching / flow matching
- and possibly more

implicit model: architectural constraints

explicit model: prone to mode collapse

GENERATIVE AI FLAVOURS

"Creating noise from data is easy; creating data from noise is generative modeling.“
(Song+2020)

- GANs: Sample noise z from a known $p(z)$ and use a generator $G(z)$ to get data.
- VAEs: Sample noise z from a prior $p(z)$ and use a decoder $p(x|z)$ to sample data.
- Normalizing Flows: Sample noise z from a base distribution $p(z)$ and use an invertible transformation f to get data, $x = f^{-1}(z)$
- score matching: Sample noise z from a Gaussian distribution $p(z)$ and use Langevin dynamics to denoise
- …

- don’t forget good old Gaussian Processes

Score Matching

SCORE MATCHING AND DIFFUSION MODELS

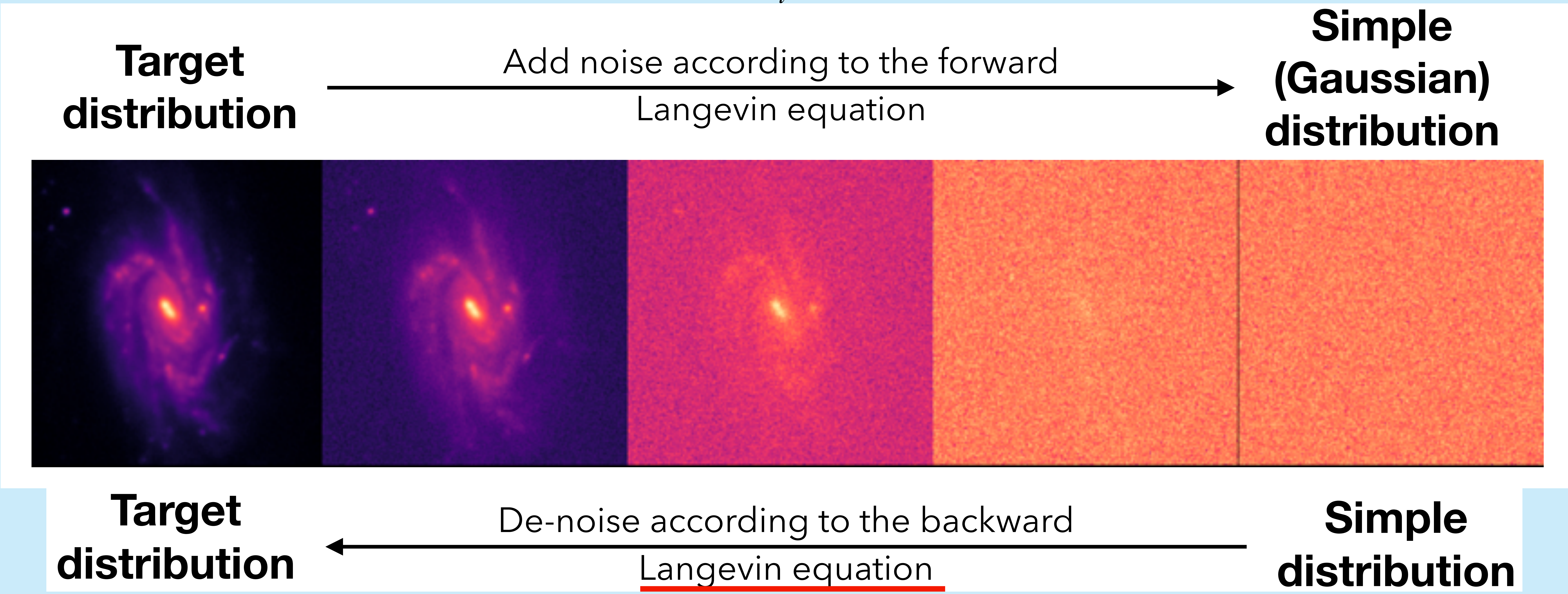
What is the score $s(\mathbf{x})$ of a pdf $p(\mathbf{x})$?

$$s(\mathbf{x}) = \nabla_{\mathbf{x}} \log p(\mathbf{x})$$

Idea: Learn $p(\mathbf{x})$ solely from data samples, then sample new instances.

But how does this work? → Langevin dynamics (Welling&Teh 2011)

$$x_{t+1} = x_t + \alpha \nabla_{x_t} \log p_{\text{real}}(x) + \eta \epsilon$$

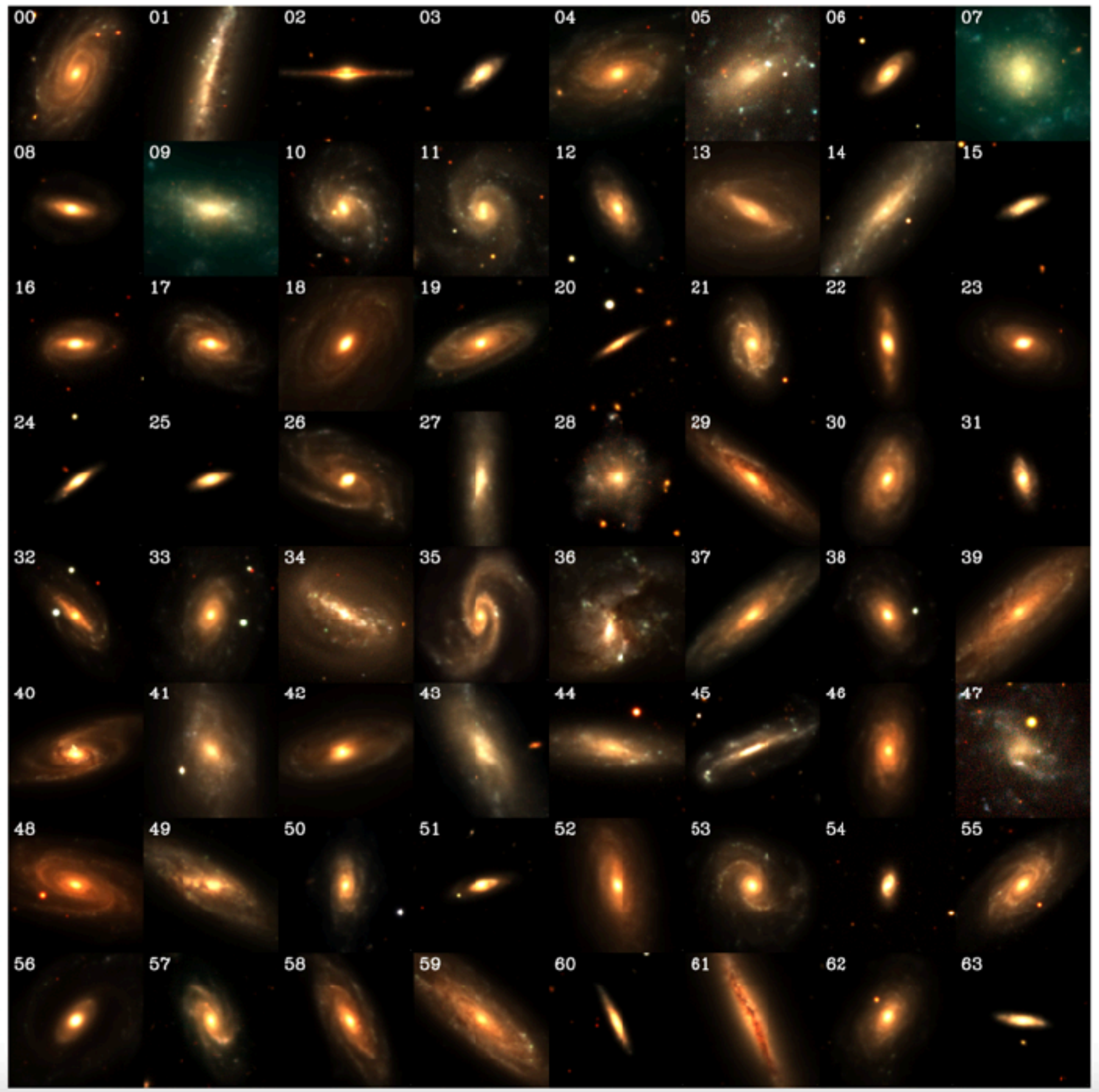


adapted from: slides by Laurence Perrault Levasseur and blog post by https://jmtomczak.github.io/blog/16/16_score_matching.html

SCORE MATCHING AND DIFFUSION MODELS

Learn generative model purely from data!

Smith+2021



http://www.mjjsmith.com/thisisnotagalaxy/

SCORE MATCHING AND DIFFUSION MODELS

Posterior samples with score-based priors

Posterior $p(x|y)$ with observation y is given by Bayes' theorem:

$$\log p(x|y) = \log p(y|x) + \log p(x) - \log p(y)$$

with $p(y|x)$ being the likelihood
and $p(x)$ the prior.

hence the score is given by:

$$\nabla_x \log p(x|y) = \nabla_x \log p(y|x) + \nabla_x \log p(x) - \nabla_x \log p(y)$$

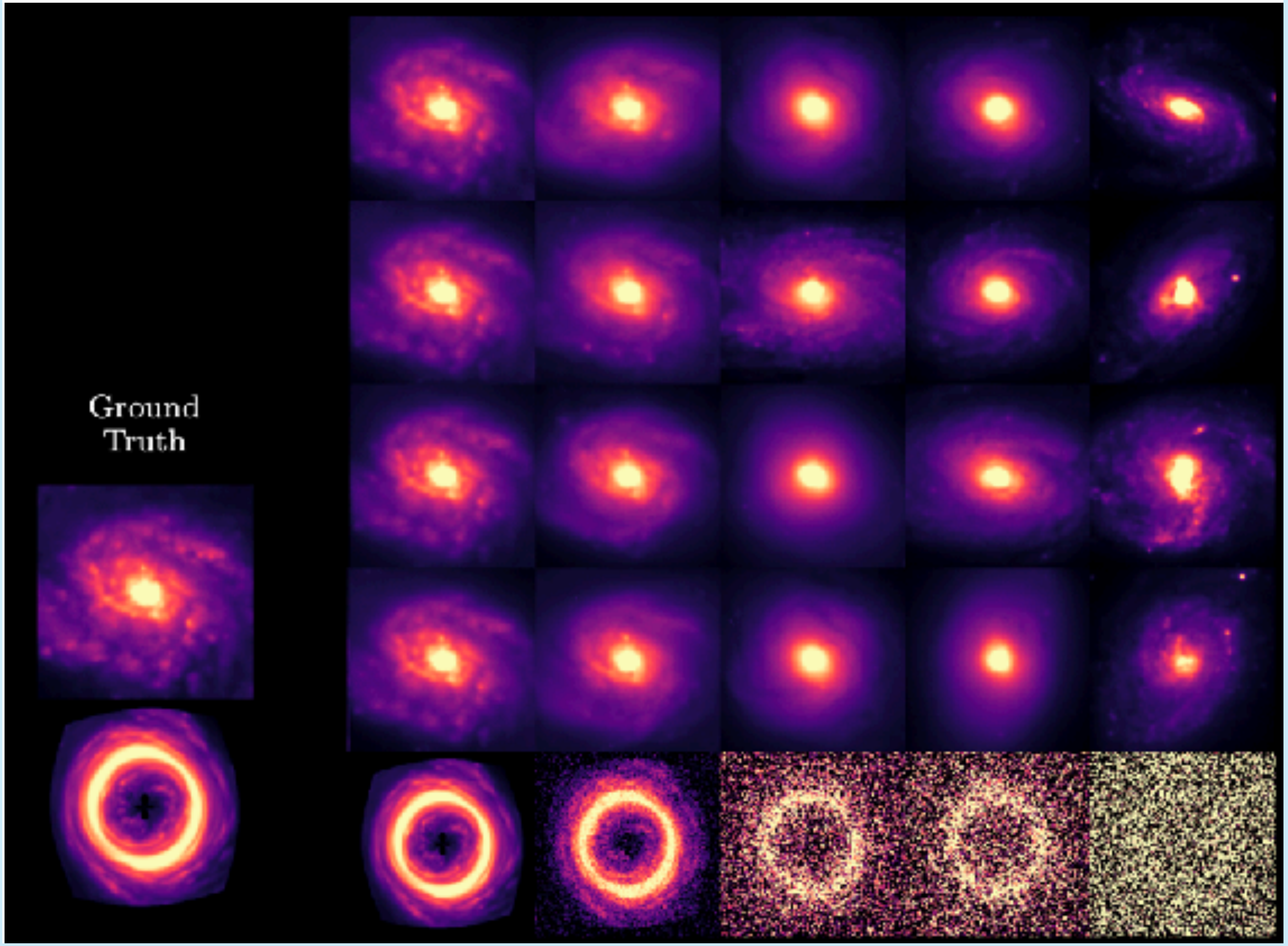
To a good approximation, we can calculate the likelihood score analytically if we assume it's Gaussian.

This is the score we learnt with Score Matching from data!

=0 because it does not depend on x.

SCORE MATCHING AND DIFFUSION MODELS

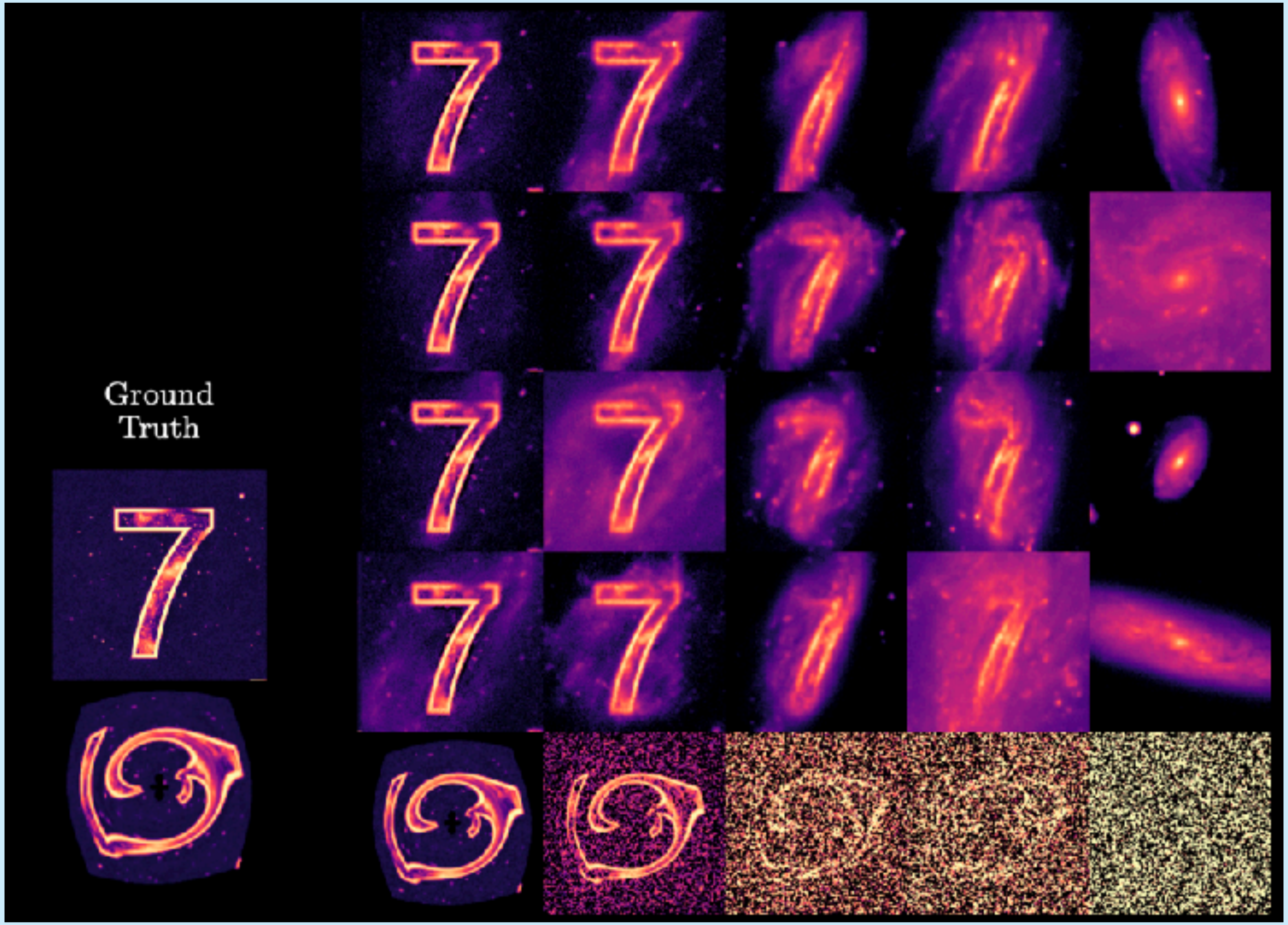
Posterior samples of source galaxies in strong gravitational lenses with score-based priors



Adam+2022

SCORE MATCHING AND DIFFUSION MODELS

Posterior samples for out-of-distribution galaxies



Adam+2022


Normalizing Flows

NORMALIZING FLOWS

How can we approximate high-dimensional, complex probability distributions $p(\theta \mid \mathbf{x_o})$?

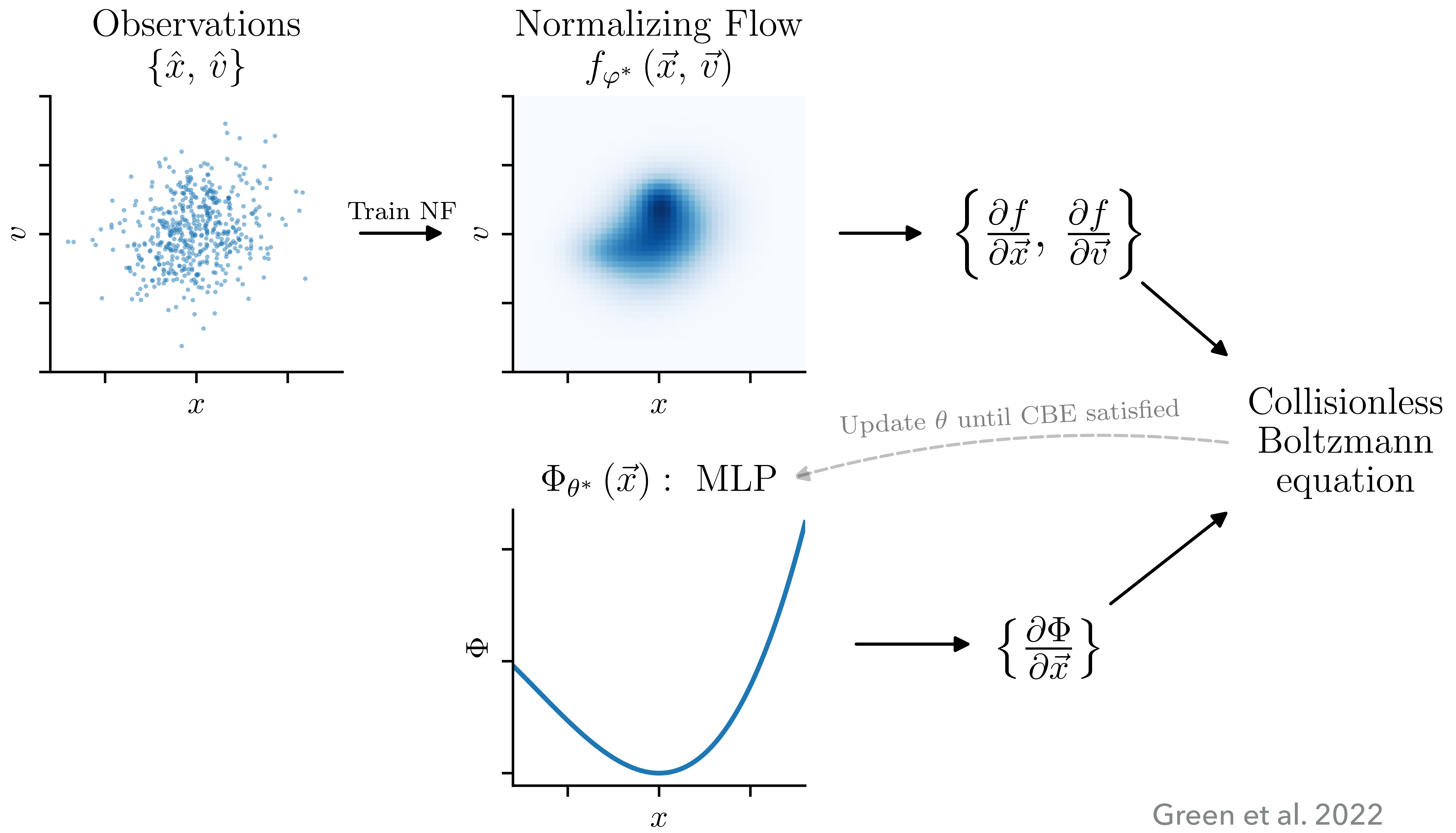
Goal:

- model $p(\theta \mid \mathbf{x_o})$ explicitly
- sample and evaluate $p(\theta \mid \mathbf{x_o})$

Idea:

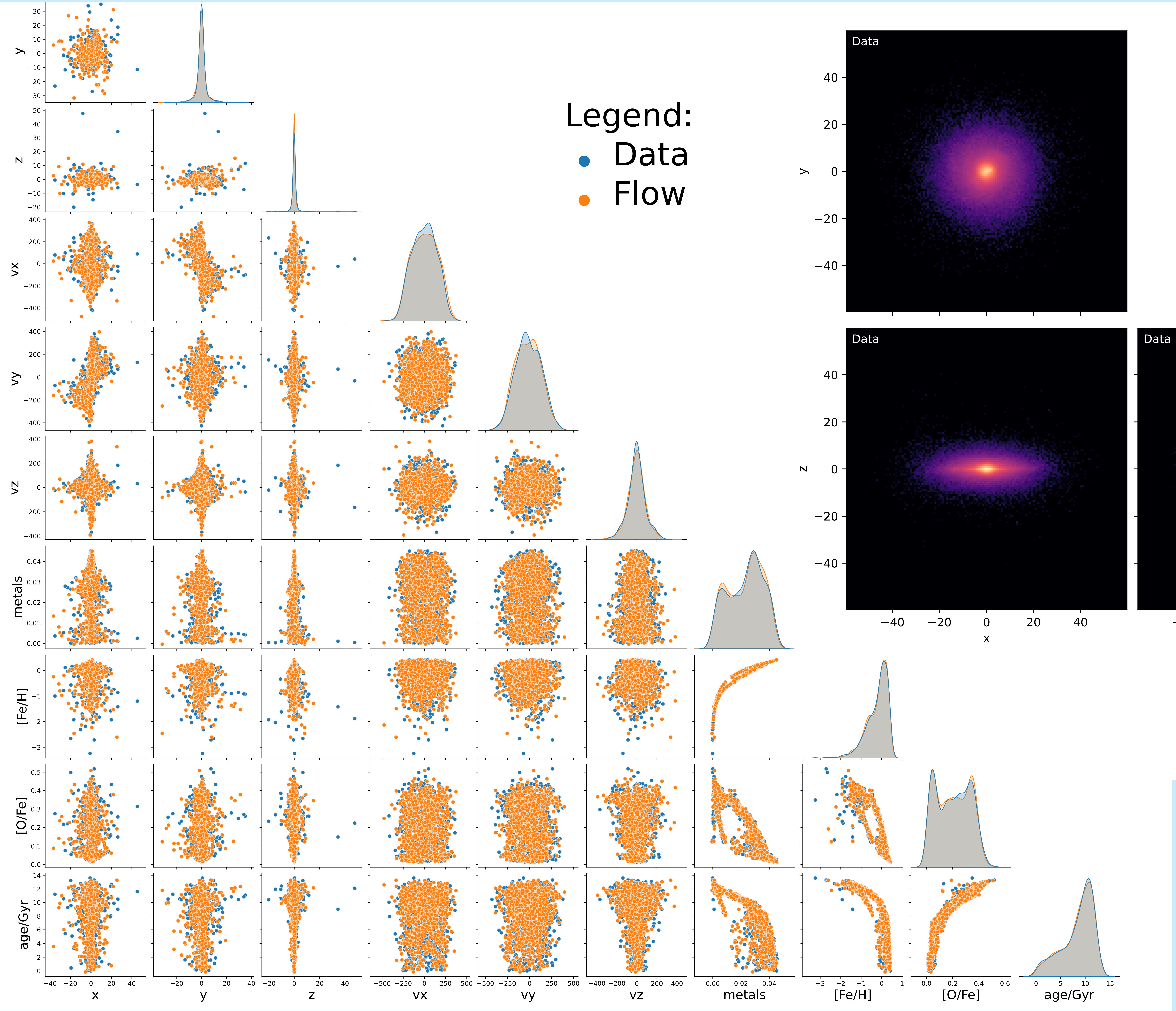
Transform a simple base distribution through a series of invertible transformations.

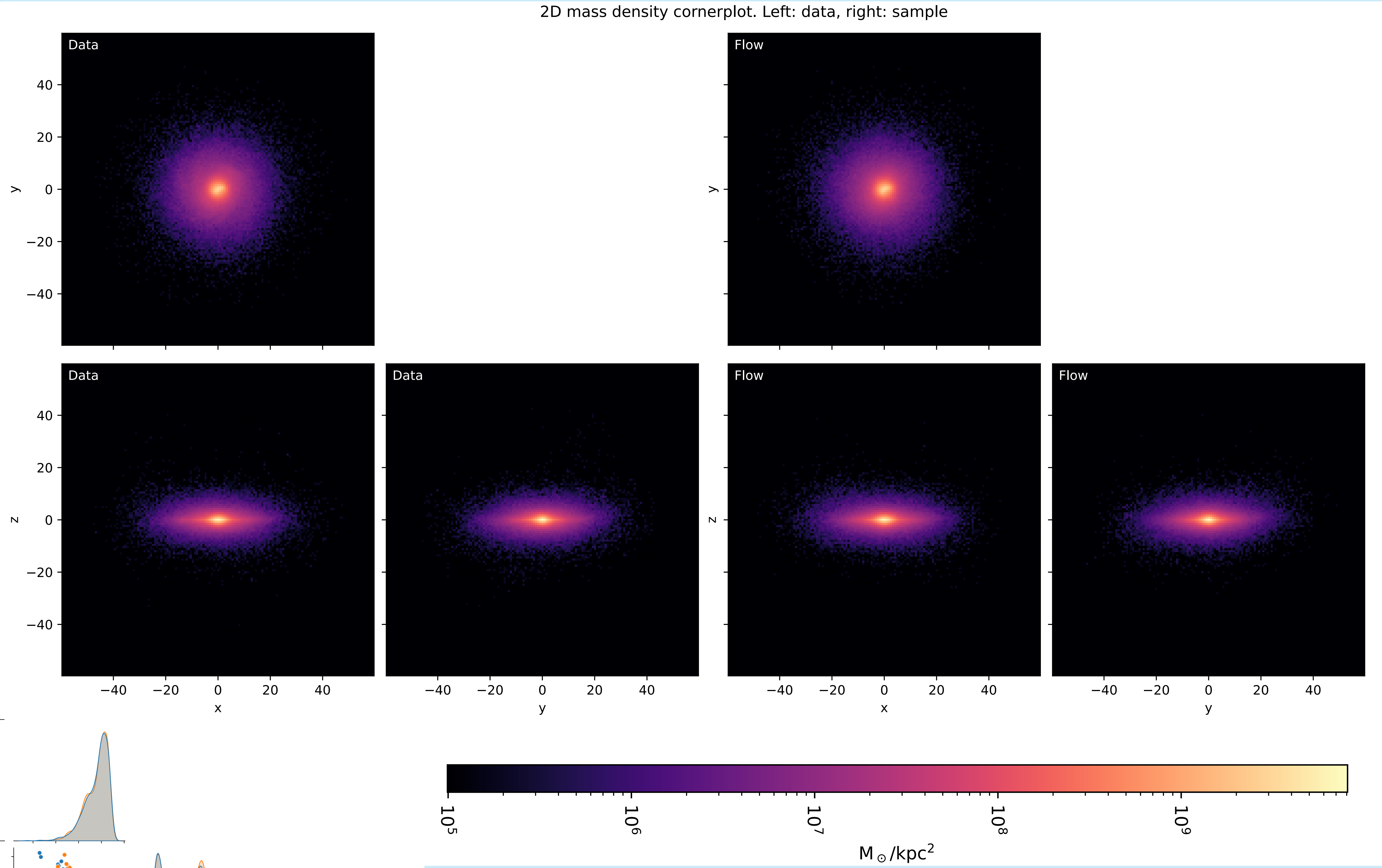
NORMALISING FLOW: APPLICATION I



Green et al. 2022

NORMALISING FLOW: APPLICATION II





Wolf+Buck 2023

NORMALISING FLOW: APPLICATION III

- Normalizing flows for random fields in cosmology (Rouhiainen+2021)
- Bayesian Stokes inversion with normalizing flows (Baso+2022)
- A Hierarchy of Normalizing Flows for Modelling the Galaxy-Halo Relationship (Lovell+2023)
- HIFlow: Generating Diverse Hi Maps and Inferring Cosmology while Marginalizing over Astrophysics Using Normalizing Flows (Hassan+2022)
- Normalizing Flows as an Avenue to Studying Overlapping Gravitational Wave Signals (Langendorff+2023)
- Charting Galactic Accelerations: When and How to Extract a Unique Potential from the Distribution Function (An+2021)
- Charting galactic accelerations II: how to 'learn' accelerations in the solar neighbourhood (Naik+2021)

- many many more…


Simulation-based Inference

SIMULATION-BASED INFERENCE - SBI

$$p(\theta|\mathbf{x}) = \frac{p(\mathbf{x}|\theta)p(\theta)}{p(\mathbf{x})} \propto p(\mathbf{x}|\theta)p(\theta)$$

- Insight: running a **stochastic simulator** with input θ gives an output $\mathbf{x}$ that is drawn from an implicit likelihood $p(\mathbf{x}|\theta)$
- „simulation-based inference“ or „likelihood-free inference“ or „implicit likelihood inference“ or … (review: Cranmer+2020)
- recent progress thanks to deep learning algorithms, e.g. conditional normalizing flows (Papamarkios+2019, Greenberg+2019, Hermans+2020, …)

SBI: NEURAL X ESTIMATION

- Use neural networks to approximate some quantities in Bayes' formula

$$p(\theta \,|\, \mathbf{x}) = \frac{p(\mathbf{x} \,|\, \theta)}{p(\mathbf{x})} p(\theta) = \frac{p(\mathbf{x} \,|\, \theta)}{p(\mathbf{x})} p(\theta)$$

- Neural Posterior Estimation (NPE)
- Neural Likelihood Estimation (NLE)
- Neural Ratio Estimation (NRE)

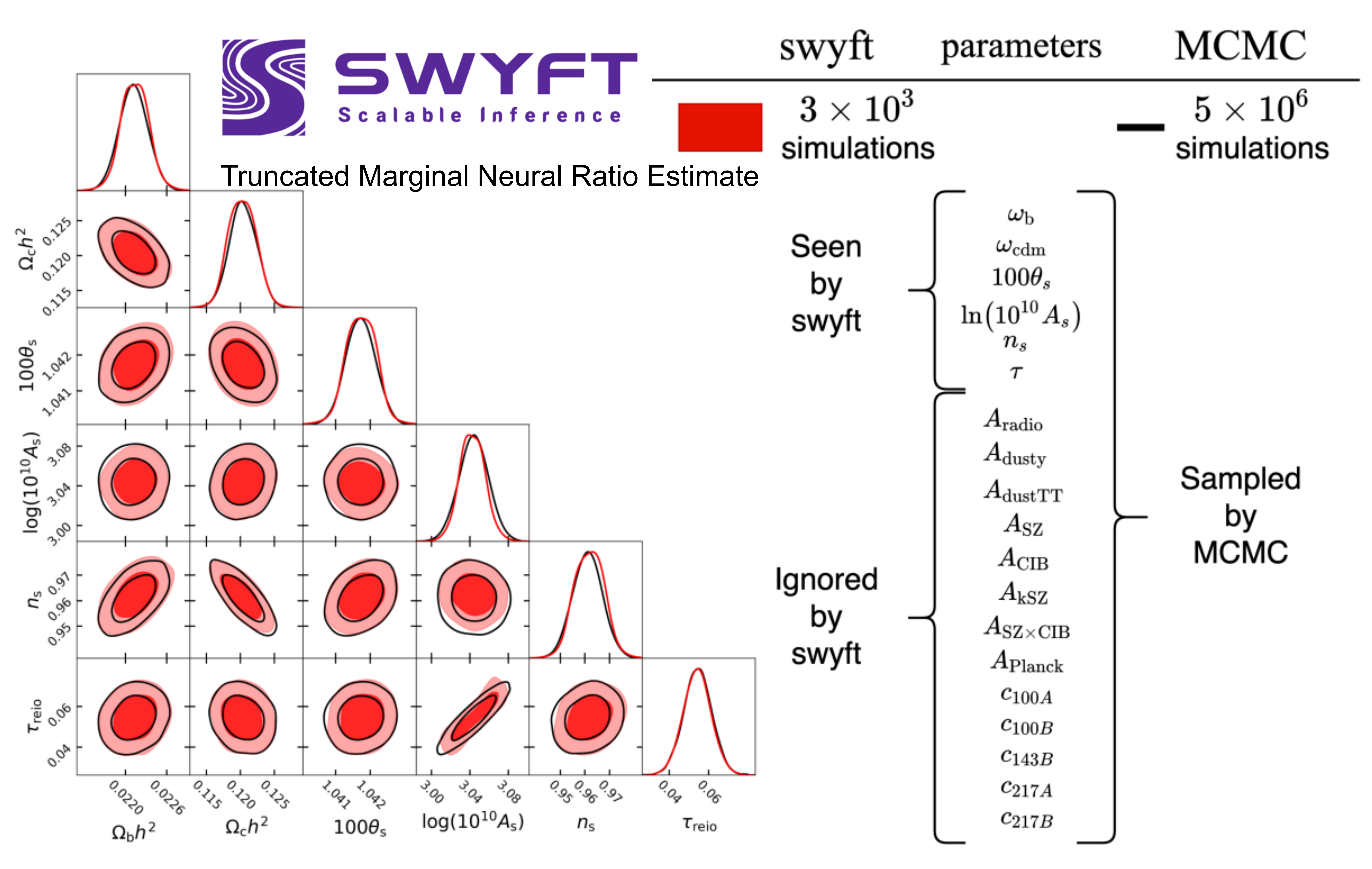
SBI: (CONDITIONAL) DENSITY ESTIMATION

- NLE and NPE both estimate normalised probability densities, hence:
  - restricted network architectures, e.g. normalizing flows or mixture density models. potentially difficult to train (Papamarkios+2021)
  - for high-dimensional data, compression/embedding network needed.
- but: restriction can be a good inductive bias, especially if posterior or likelihood is “perturbation around Gaussian distribution”
- automatic marginalization possible

c.f. pydelfi Alsing+2018,2019; moment networks Jeffrey+Wandelt 2020, SBI Jakob Macke, ltU-ili Ho+2024, Bayesflow Radev+2020,2023, swyft Miller+2021,2022

https://simulation-based-inference.org/
https://github.com/smsharma/awesome-neural-sbi for references to software and applications

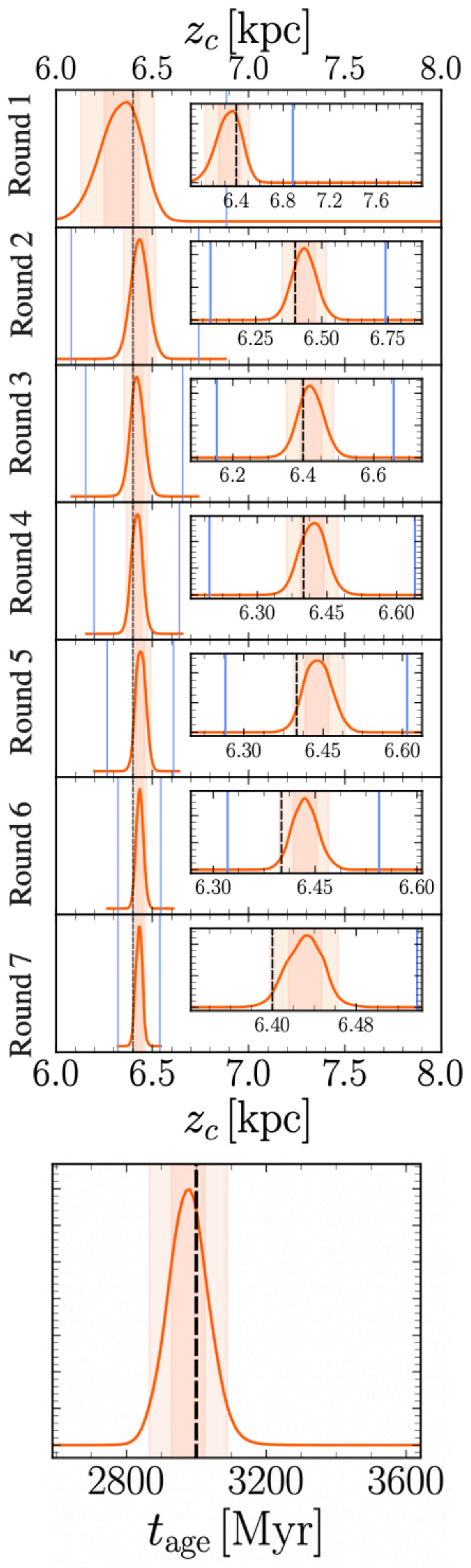
SBI: APPLICATION IN COSMOLOGY



Cole+2022

SBI: APPLICATION FOR STELLAR STREAMS

TRUNCATED MARGINAL NEURAL RATIO ESTIMATION (TMNRE) FOR STELLAR STREAMS



STEP 1: (RE-) SIMULATE

- Sample parameters θ from (truncated) prior $p(\theta)$
- Simulate data $x \sim p(x|\theta)$

$$\theta \equiv (t_{age}, \sigma_v, \ldots) \rightarrow x = \text{stream} + \text{bkg}\,.$$

STEP 2: RATIO ESTIMATION

- Train ratio estimators $r(x;\theta_i)$ on simulated data to approximate the posterior-to-prior ratio $r(x;\theta_i) \sim p(\theta_i|x)/p(\theta_i)$ for each parameter of interest θ_i

sstrax
FAST STELLAR STREAM MODELLING IN JAX

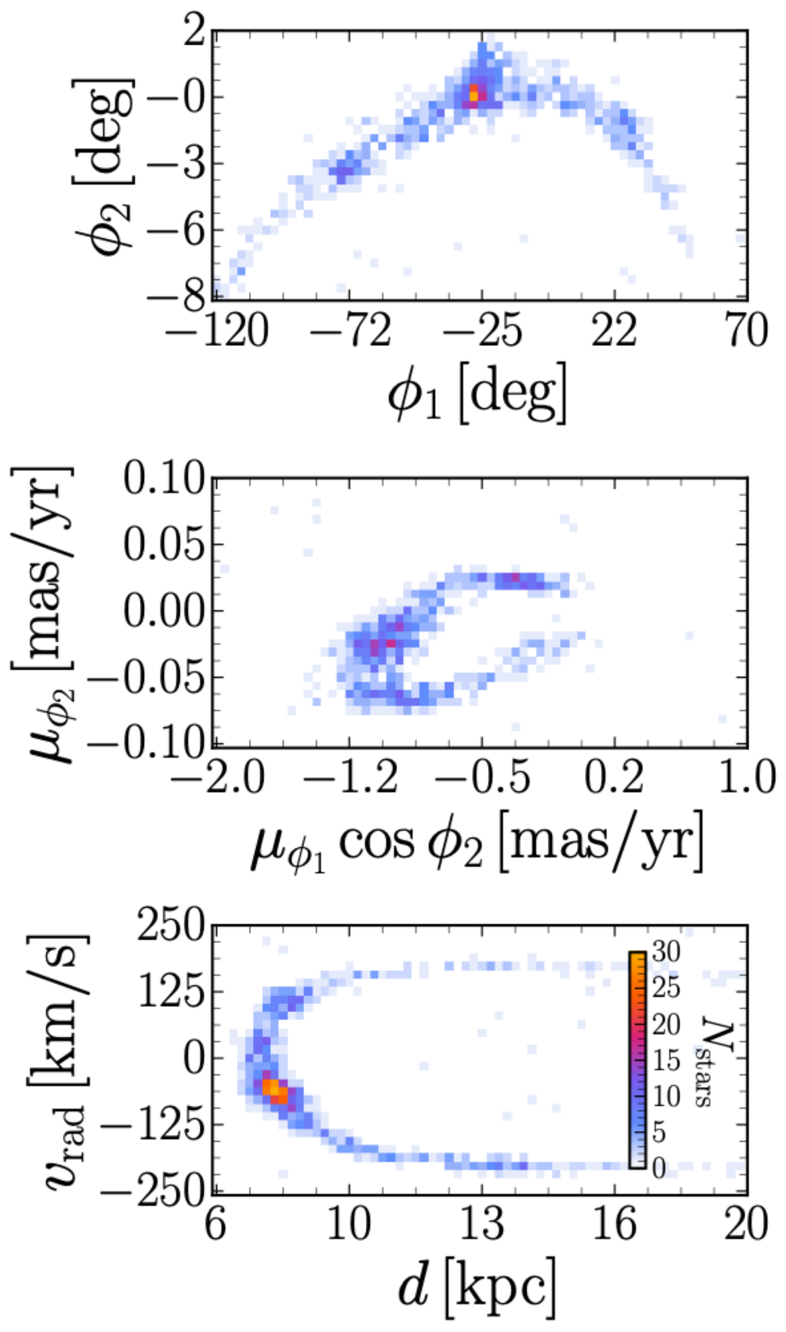
ALBATROSS
STELLAR STREAM PARAMETER INFERENCE WITH NEURAL RATIO ESTIMATION

STEP 4: TRUNCATION

- Use ratios $r(x_0;\theta_i)$ to remove regions with extremely low posterior density
- If this leads to a reduction in prior volume, re-simulate from **Step 1** with truncated prior
- Repeat until converged across all parameters, then obtain resulting posterior

STEP 3: INFERENCE

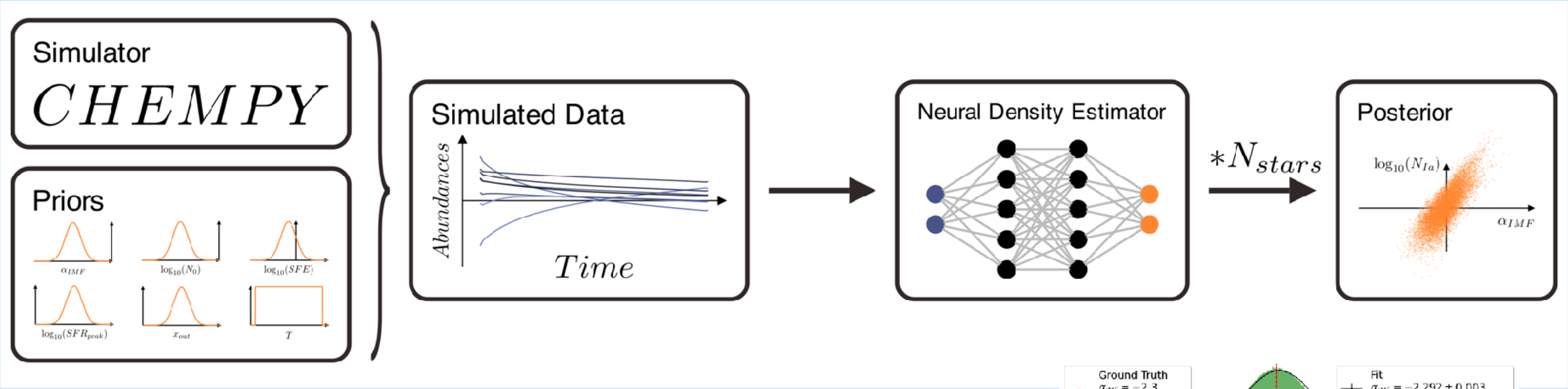
- Obtain a prior sample from $p(\theta_i)$
- Target a specific observation x_0 and compute the ratios $r(x_0;\theta_i)$ across the prior sample
- Weight the samples according to this ratio

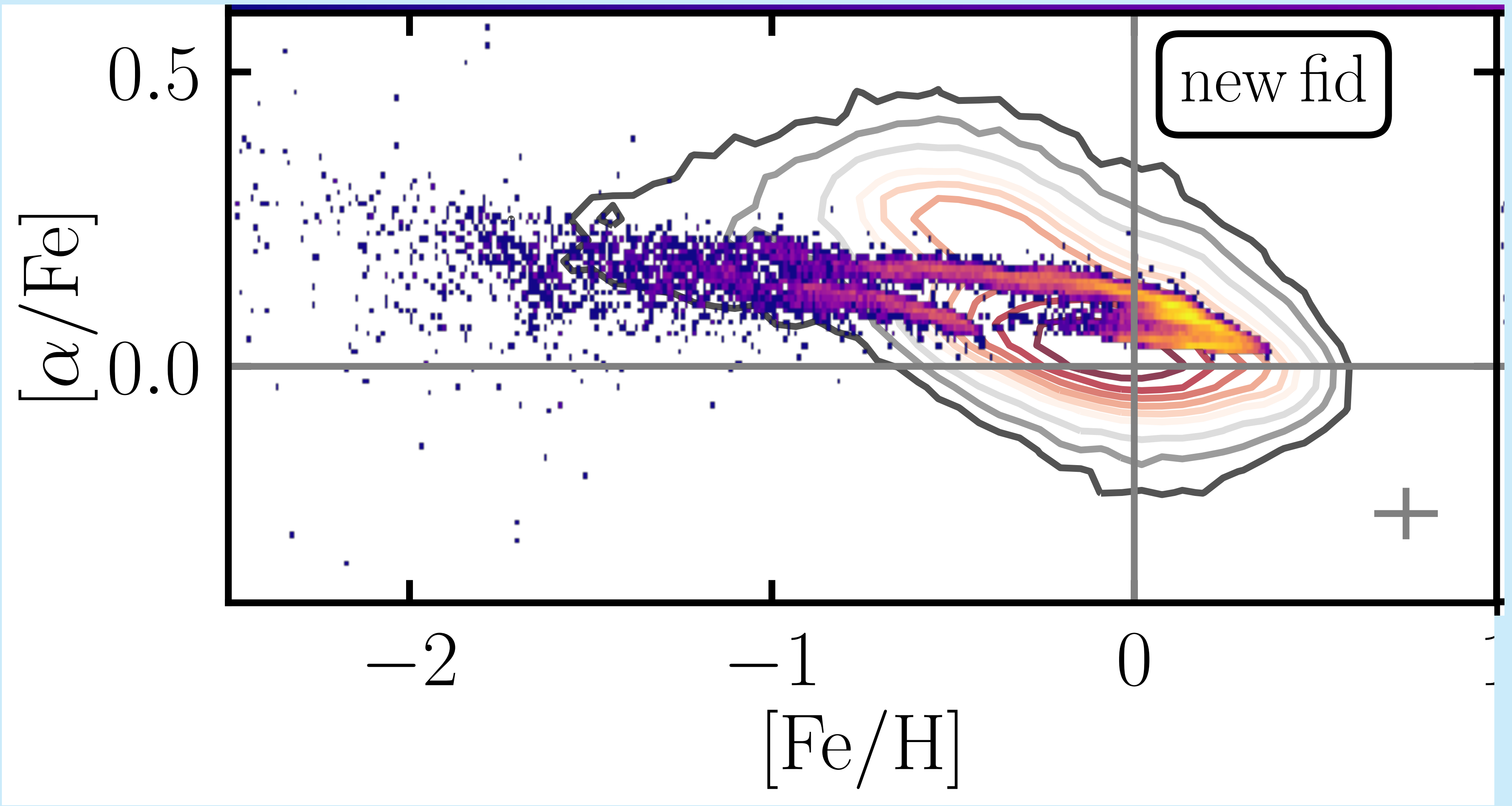


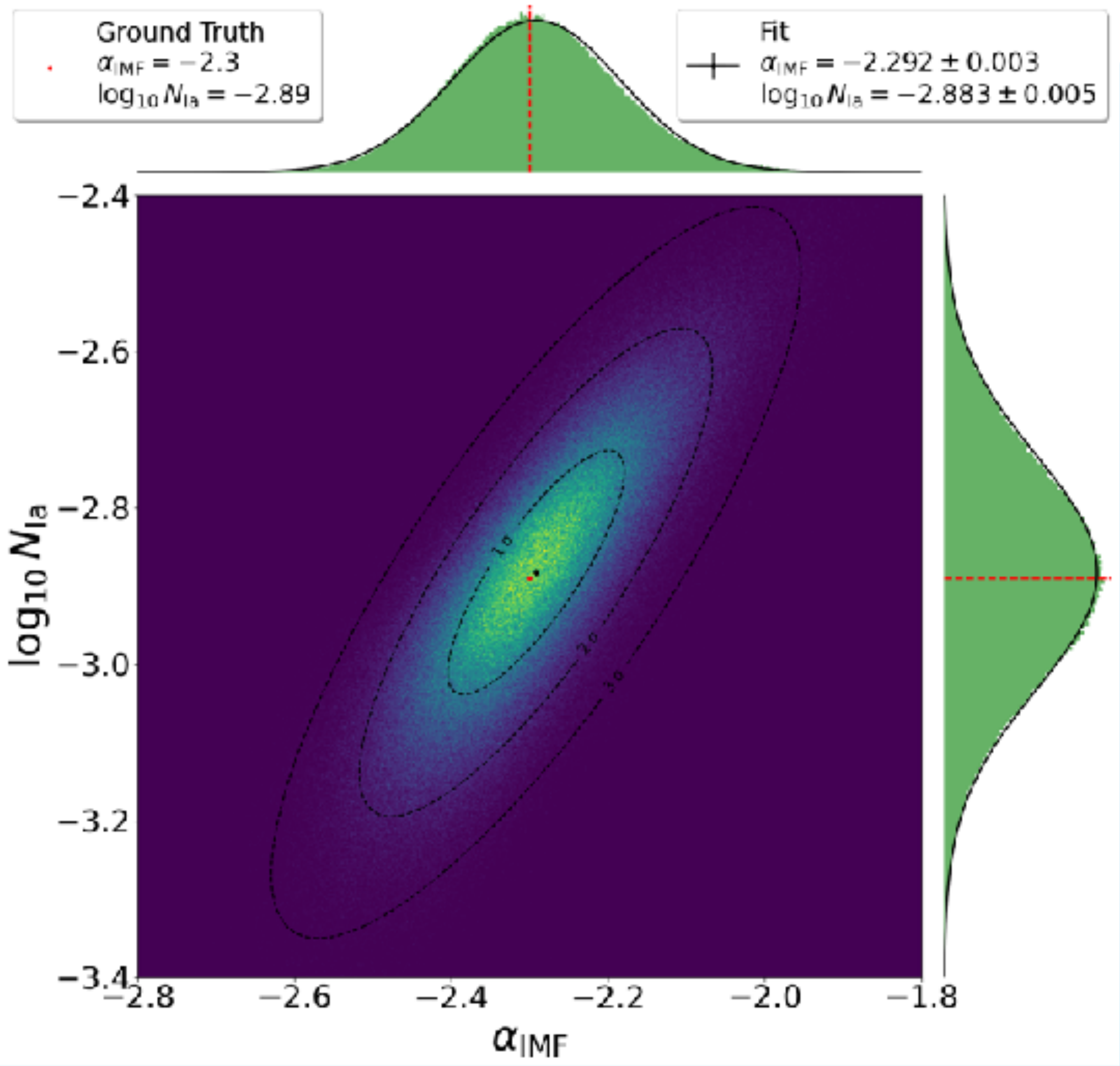
Observation, x_0

Alvey+2024

SBI: APPLICATION FOR GALACTIC CHEMICAL ENRICHMENT



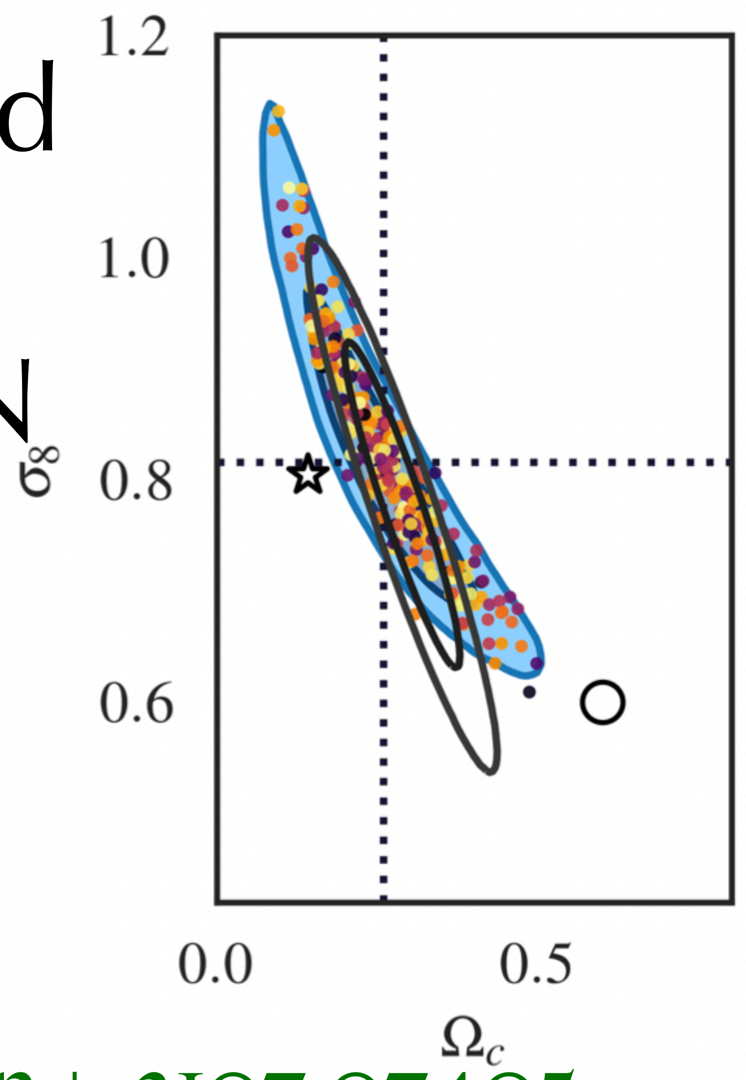




Günes+Buck subm.

SBI: APPLICATION IN COSMOLOGY

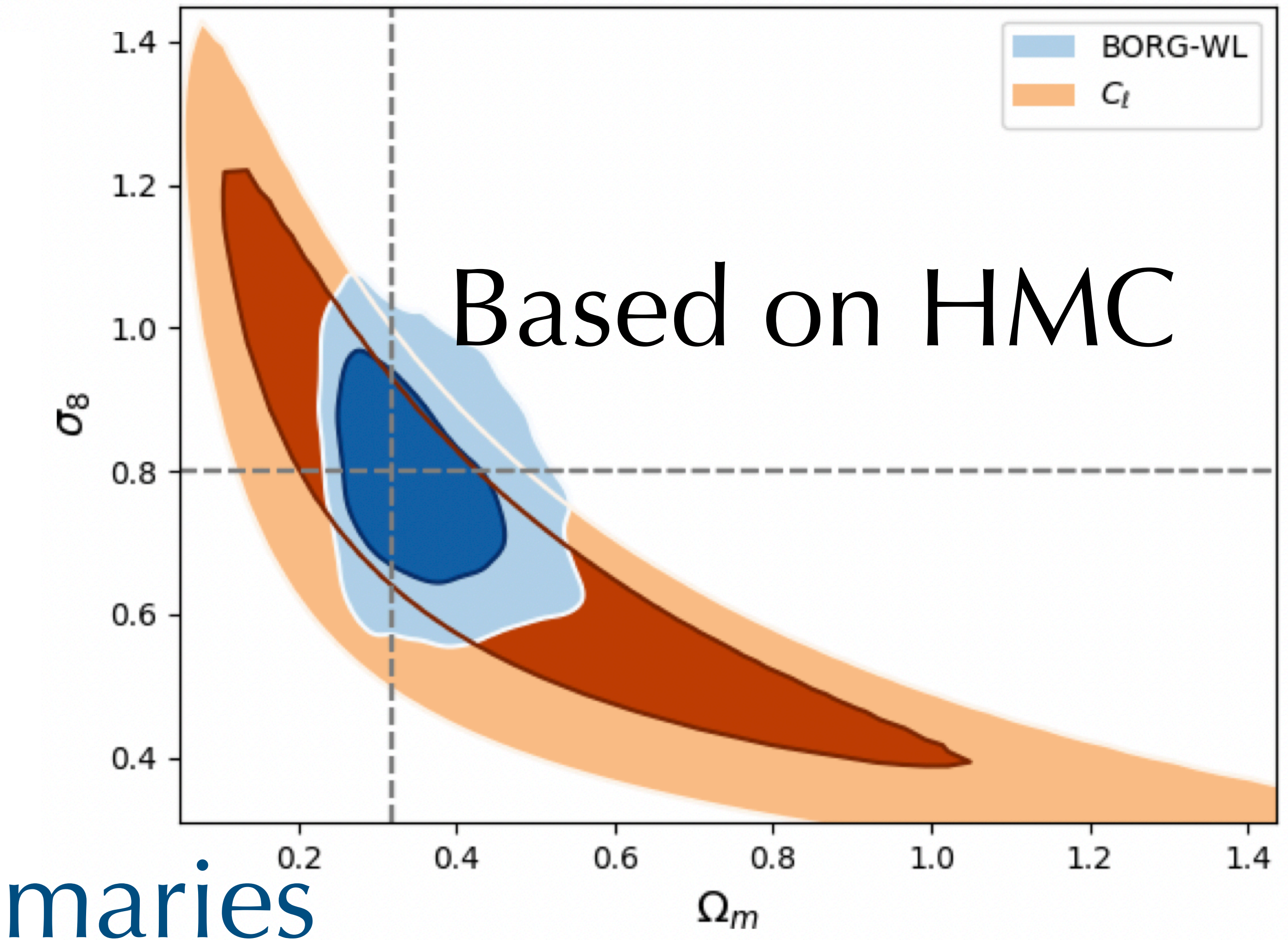
Based on CNN



Makinen+ 2107.07405

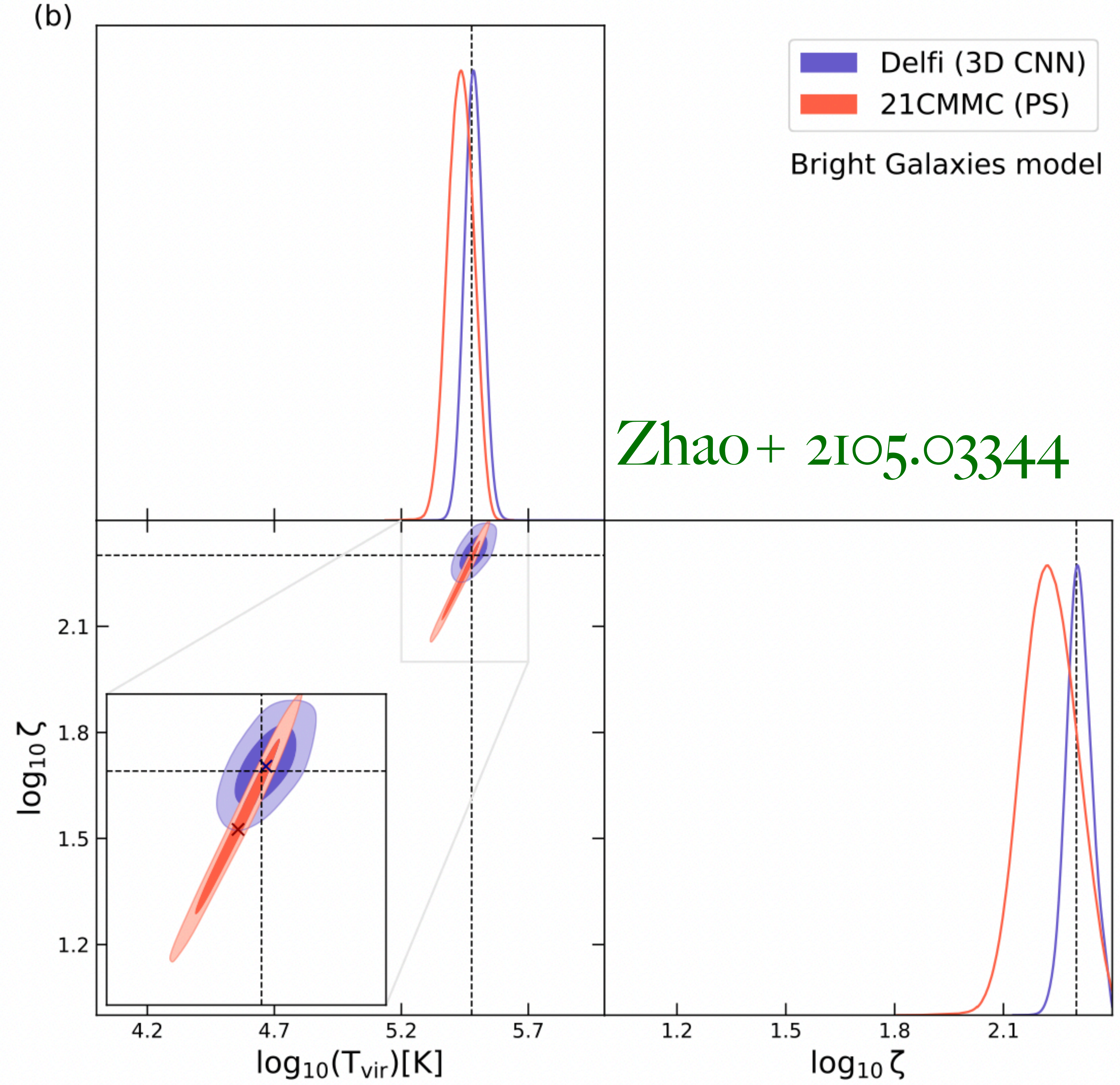
Breaking degeneracy between DM density and power-spectrum amplitude

Porqueres+ 2108.04825



Based on HMC

Alternative to:
Hand-crafted summaries



Zhao+ 2105.03344

Breaking degeneracy between ionisation parameters T_{vir} and ζ

slide from Cole

SBI: APPLICATION IN STRONG LENSING

Searching light DM halos

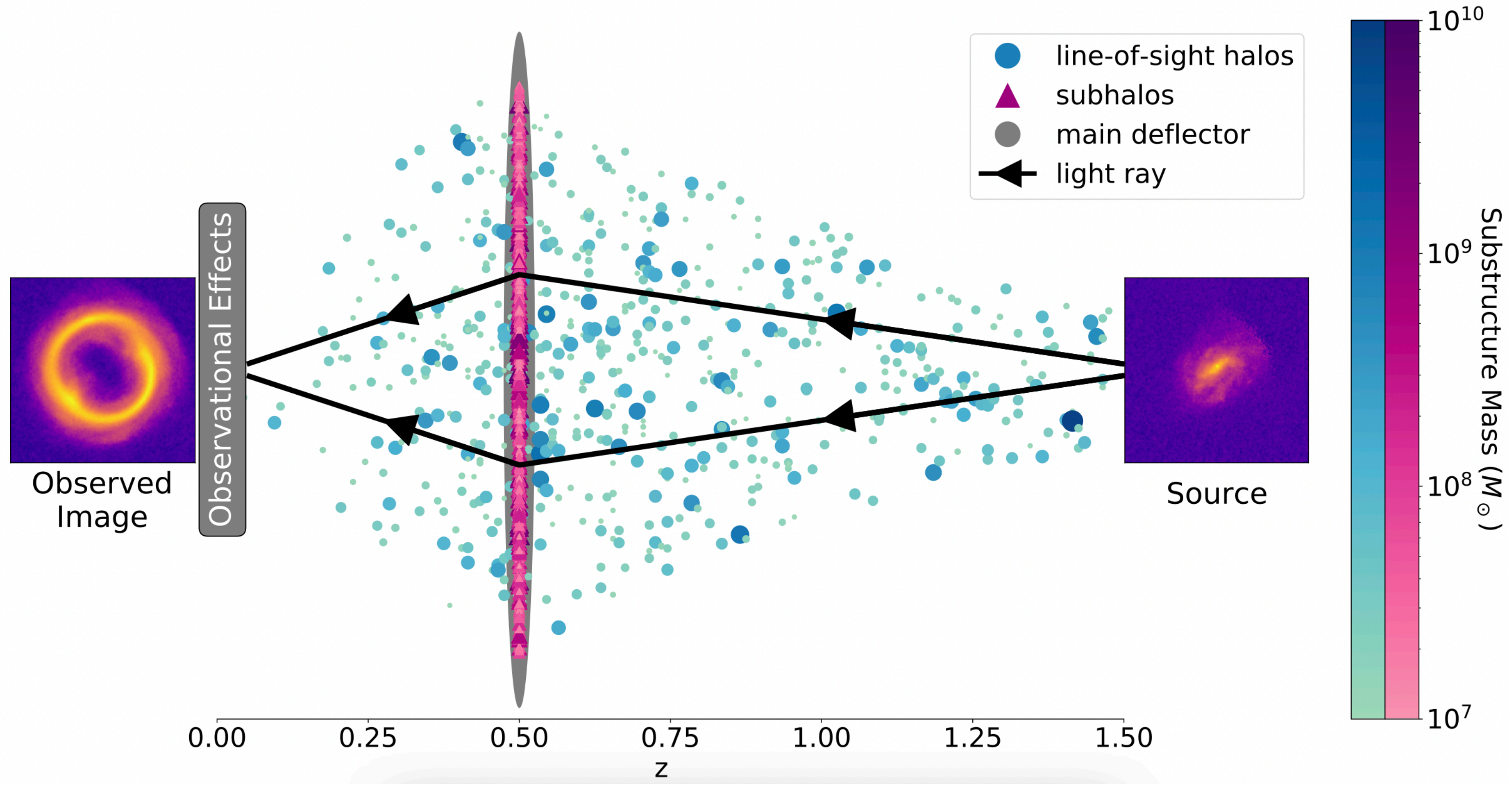
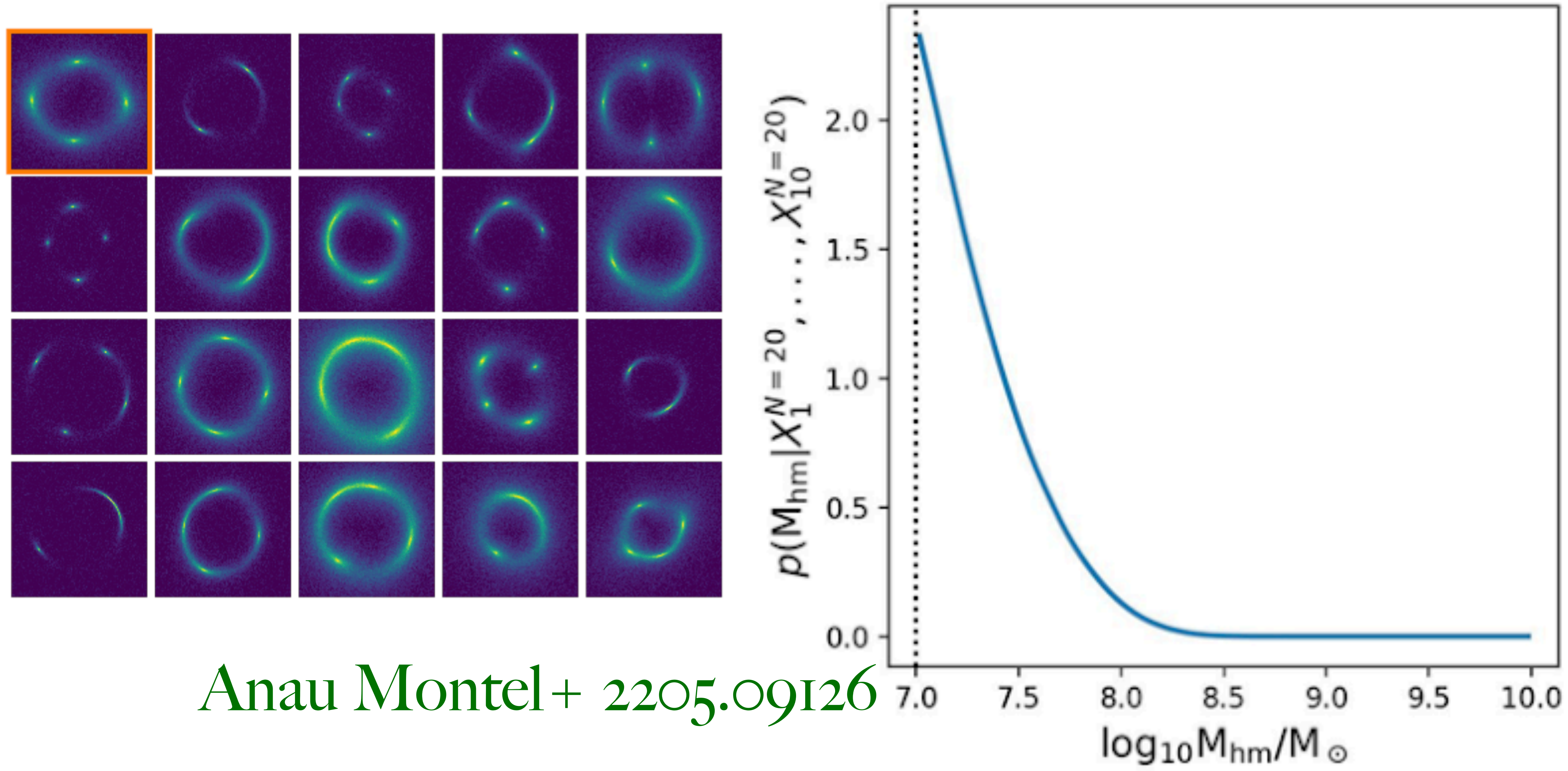


Image credit: Wagner-Carena+ 2203.00690

Halo mass function cutoff

Probing **population effects of light dark matter halos** rather than individual detections



Anau Montel+ 2205.09126

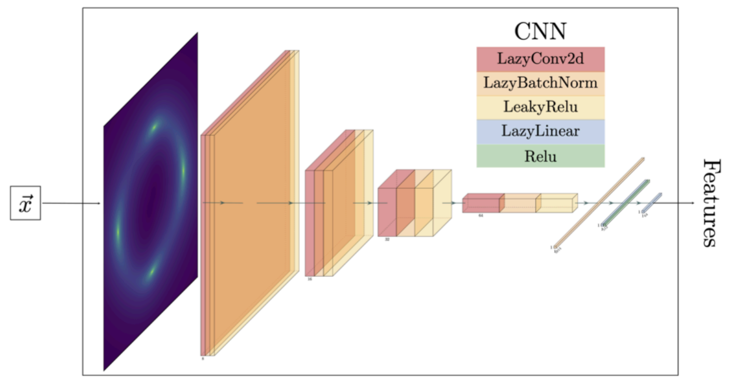
Alternative to:
HMC, parameter reduction, ABC, …

Related work: He+ 2010.13221 (similar in spirit, using ABC)

Wagner-Carena+ 2203.00690 (constraining subhalo mass function normalization)

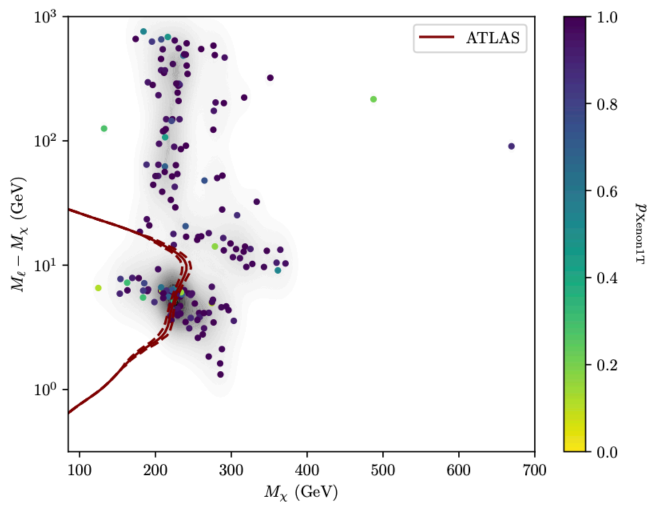
slide from Cole

SBI: APPLICATION IN STRONG LENSING



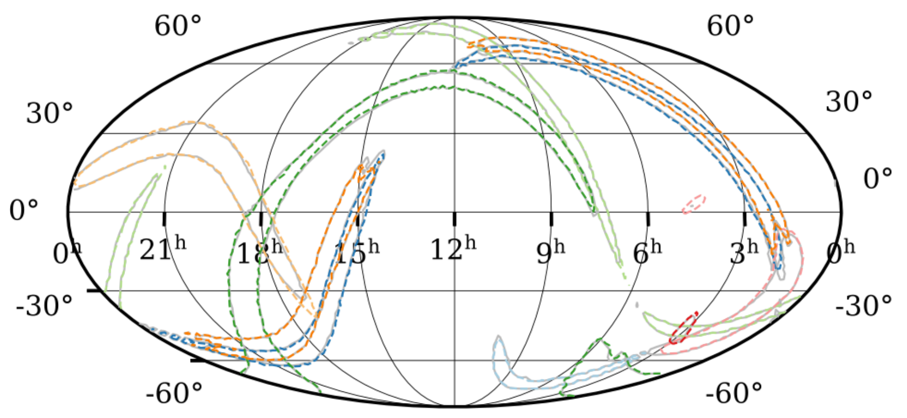
Strong lensing

Brehmer+ 1909.02005, Coogan+ 2010.07032, Legin+ 2112.05278, Wagner-Carena+ 2203.00690, Anau Montel+ 2205.09126, Coogan+ 2207.xxxxx



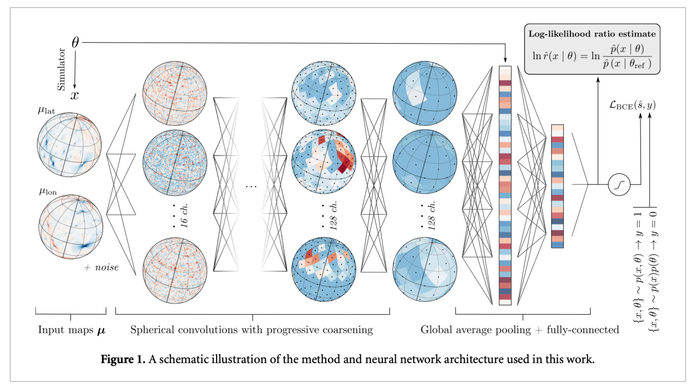
Effective field theory

Morrison+ 2203.13403



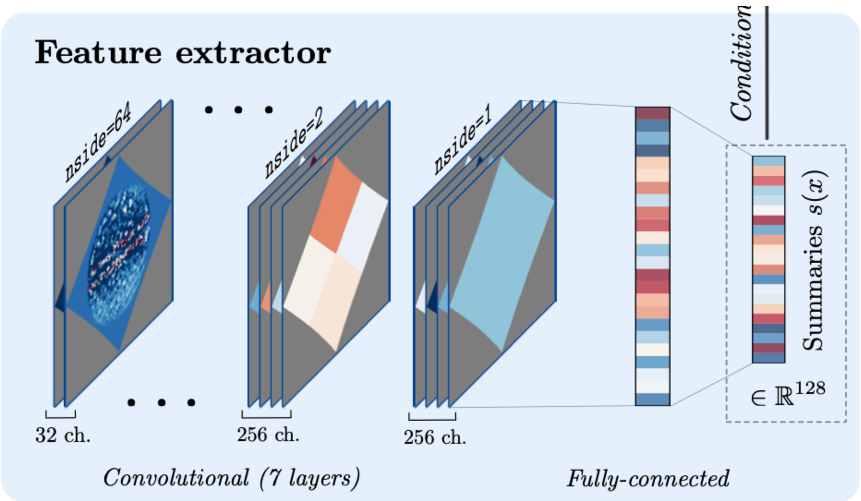
GW parameters

Delaunoy+ 2010.12931, Dax+ 2106.12594, ...



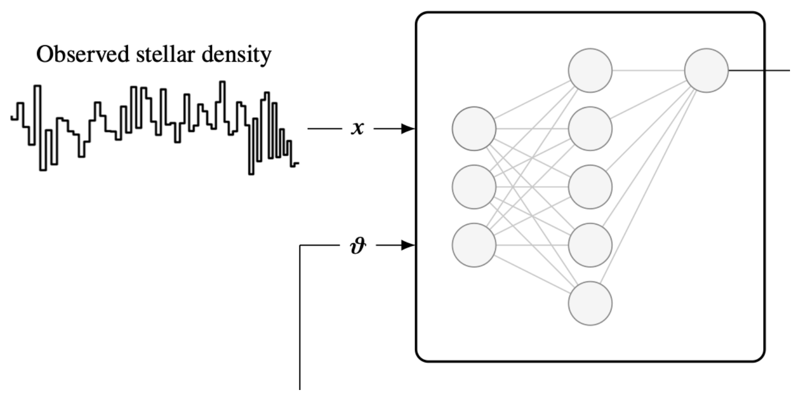
Astrometry

Mishra-Sharma+ 2110.01620



Fermi GeV excess

Mishra-Sharma+ 2110.06931



Stellar streams

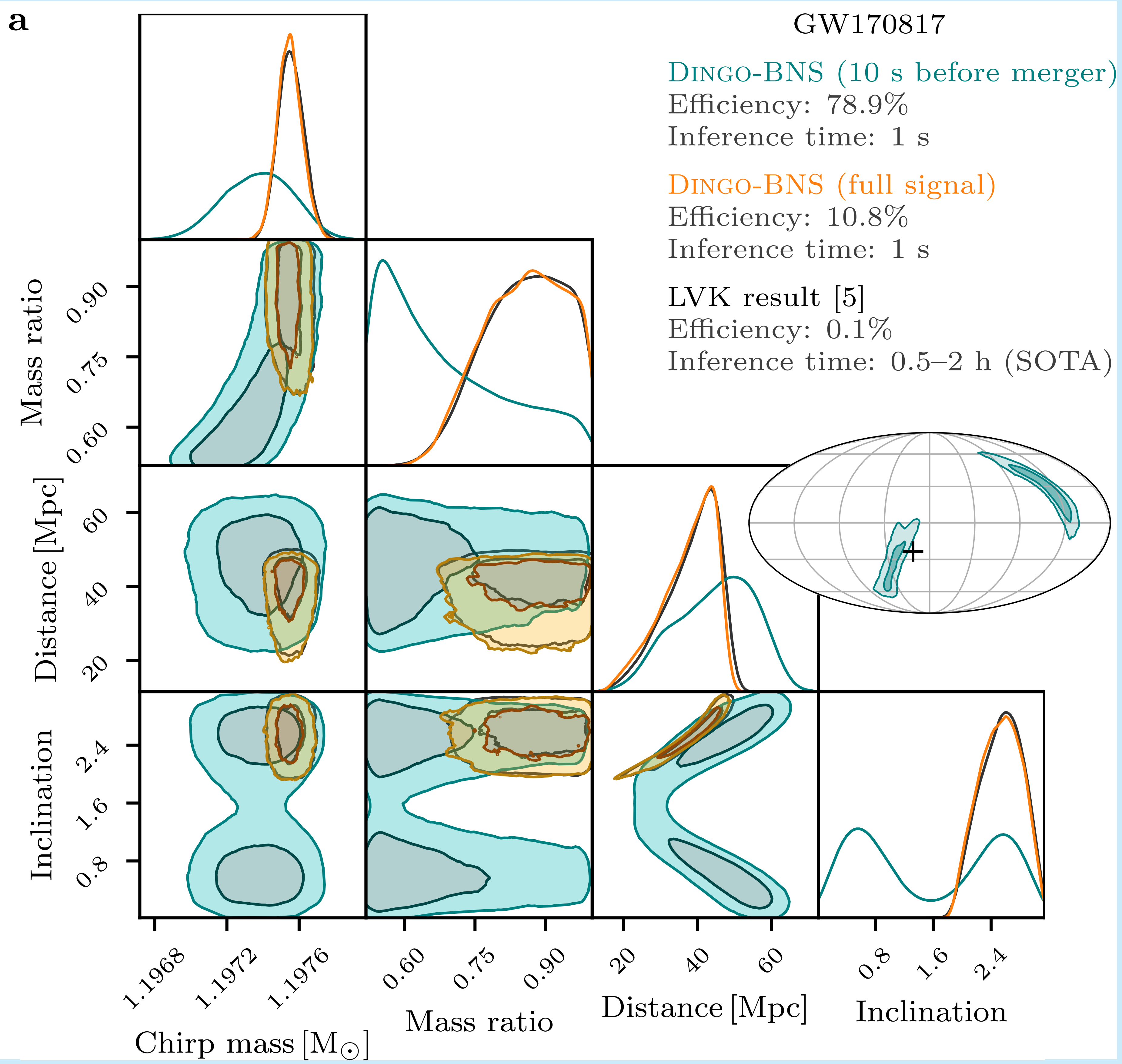
Hermans+ 2011.14923

Single frequency CMB B-mode inference with realistic foregrounds from a single training image

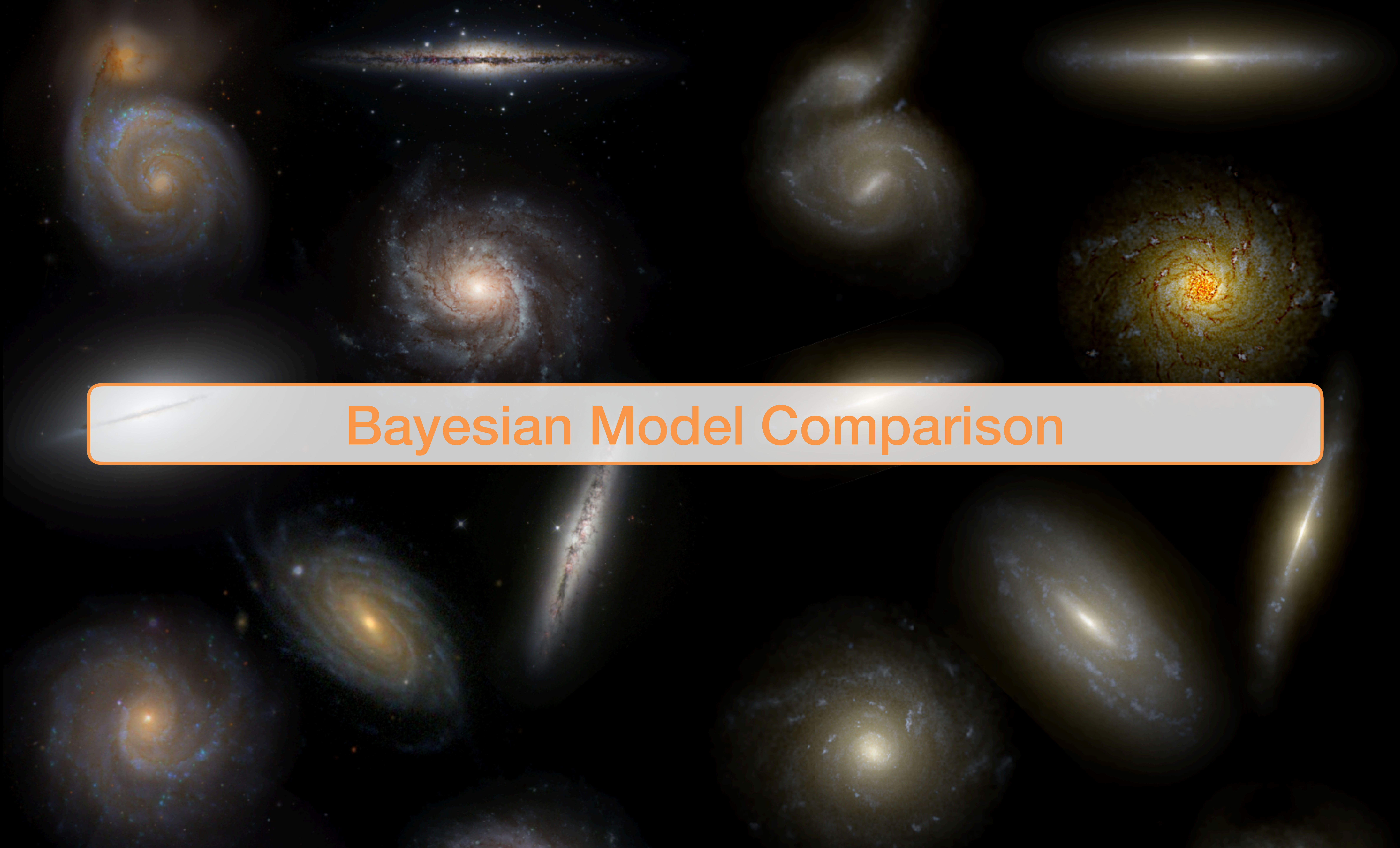
Niall Jeffrey,[1,2★] François Boulanger,[1] Benjamin D. Wandelt,[3,4] Bruno Regaldo-Saint Blancard,[1,5] Erwan Allys,[1] François Levrier[1]

slide from Cole

Gravitational wave parameter estimation



Dax+2021,2023,2024


Bayesian Model Comparison

MODEL COMPARISON & MODEL MISSPECIFICATION

Bayesian model selection assigns posterior probabilities $\mathrm{p}(\mathcal{M}_k \mid \mathbf{d})$ to models $\mathcal{M}_k \in \{\mathcal{M}_1, \ldots, \mathcal{M}_N\}$ (instead of to values of their parameters $\boldsymbol{\theta}_k$), conditional on observed data $\mathbf{d}$. The conventional approach is to compute the marginal likelihood (or *evidence*) $\mathrm{p}(\mathbf{d} \mid \mathcal{M}_k)$, which is the average likelihood $\mathrm{p}(\mathbf{d} \mid \boldsymbol{\theta}_k)$ of parameters distributed according to the prior $\mathrm{p}(\boldsymbol{\theta}_k)$:

$$\mathrm{p}(\mathbf{d} \mid \mathcal{M}_k) = \int \mathrm{p}(\mathbf{d} \mid \boldsymbol{\theta}_k)\, \mathrm{p}(\boldsymbol{\theta}_k)\, \mathrm{d}\boldsymbol{\theta}_k \qquad (1)$$

(where the presence of $\mathcal{M}_k$'s parameters $\boldsymbol{\theta}_k$ implies conditioning on $\mathcal{M}_k$ in the right-hand side). The prior belief in the model, $\mathrm{p}(\mathcal{M}_k)$, is then updated to its posterior probability in accordance with Bayes' theorem: $\mathrm{p}(\mathcal{M}_k \mid \mathbf{d}) \propto \mathrm{p}(\mathbf{d} \mid \mathcal{M}_k)\, \mathrm{p}(\mathcal{M}_k)$, normalised over all models considered.

MODEL COMPARISON & MODEL MISSPECIFICATION

Methods

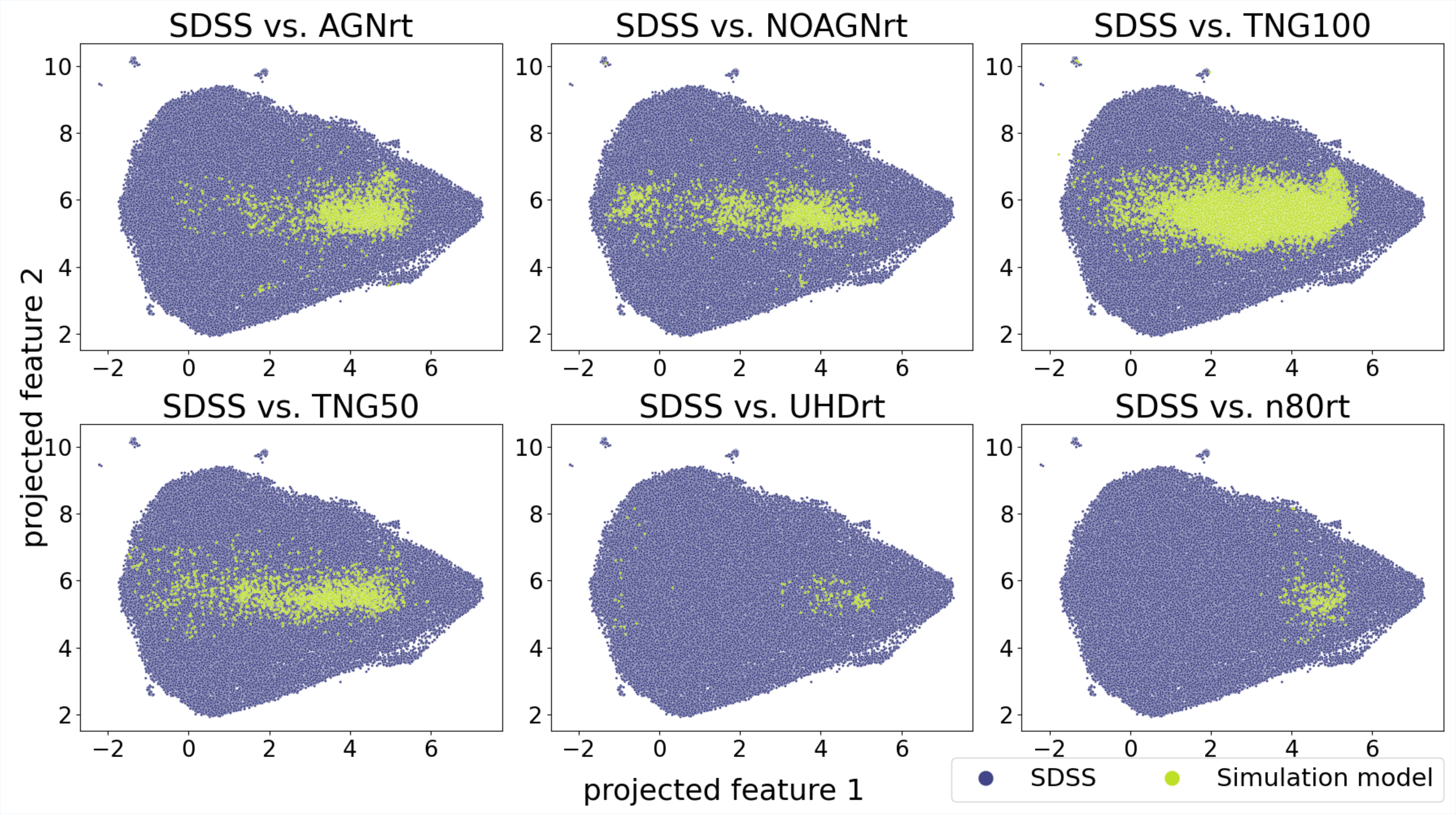
- Jeffrey+Wandelt 2023: loss functions for two-way model comparison with an emphasis on recovering accurate extreme Bayes factors
- Radev+2021: estimate a Dirichlet distribution over an arbitrary number of models using a NN and variational optimisation
- Elsemüller+2023 and Karchev+2024: use a cross-entropy loss for multi-class posterior probabilities
- Macciò+2022: Model selection for star formation prescriptions in cosmo sims
- Zhou+2024: Model misspecification plus model comparison for low simulation budget applications
- Jin+2024: Model comparison of cosmo sims via GANomaly scores

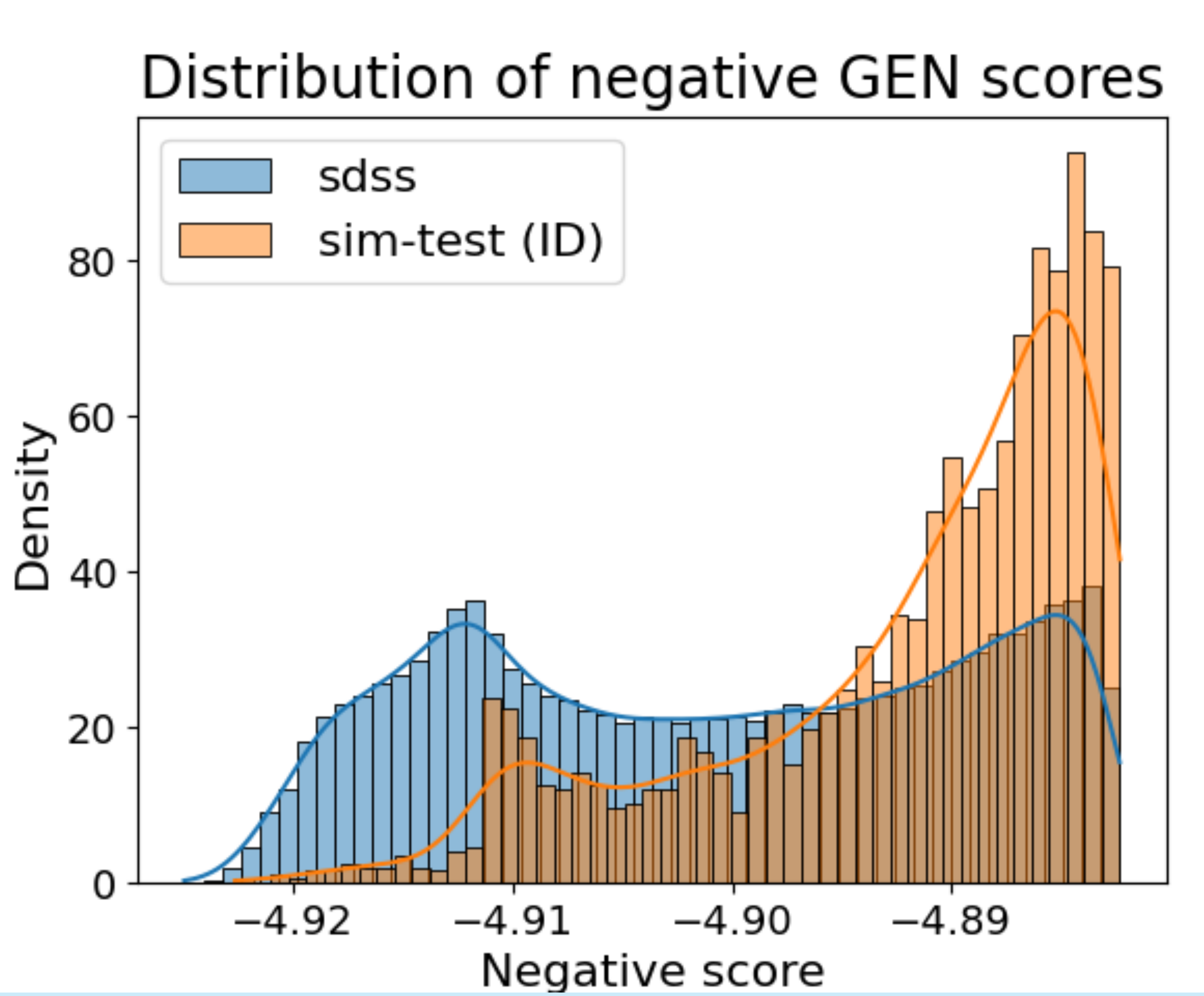
MODEL COMPARISON & MODEL MISSPECIFICATION

Idea:

- Train embedding network on ~600.000 SDSS images, then encode simulated SDSS images
- Train simulation classifiers on embeddings, apply to real SDSS images

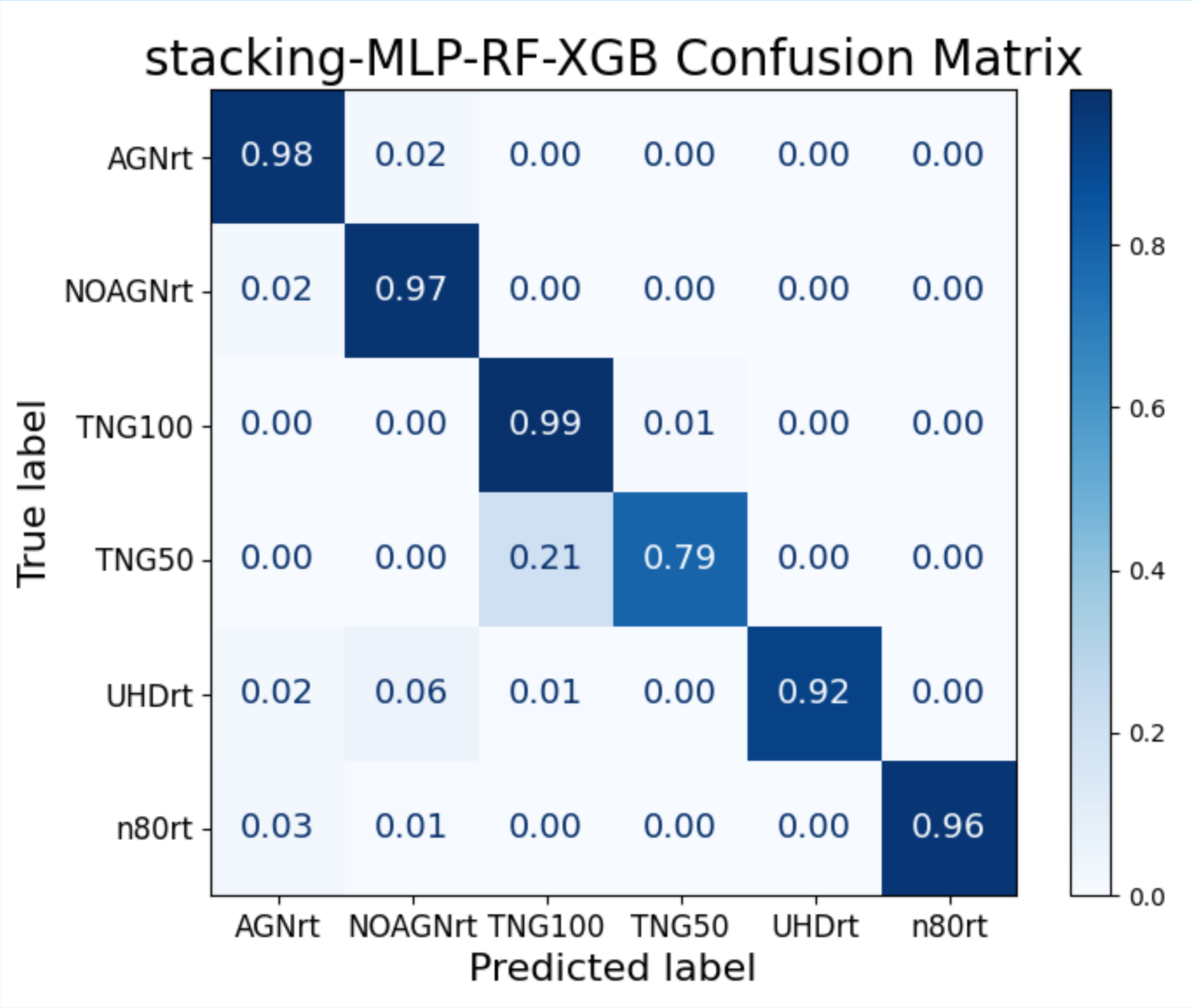
UMAP projection of k-sparse encoding of sims and Jobs

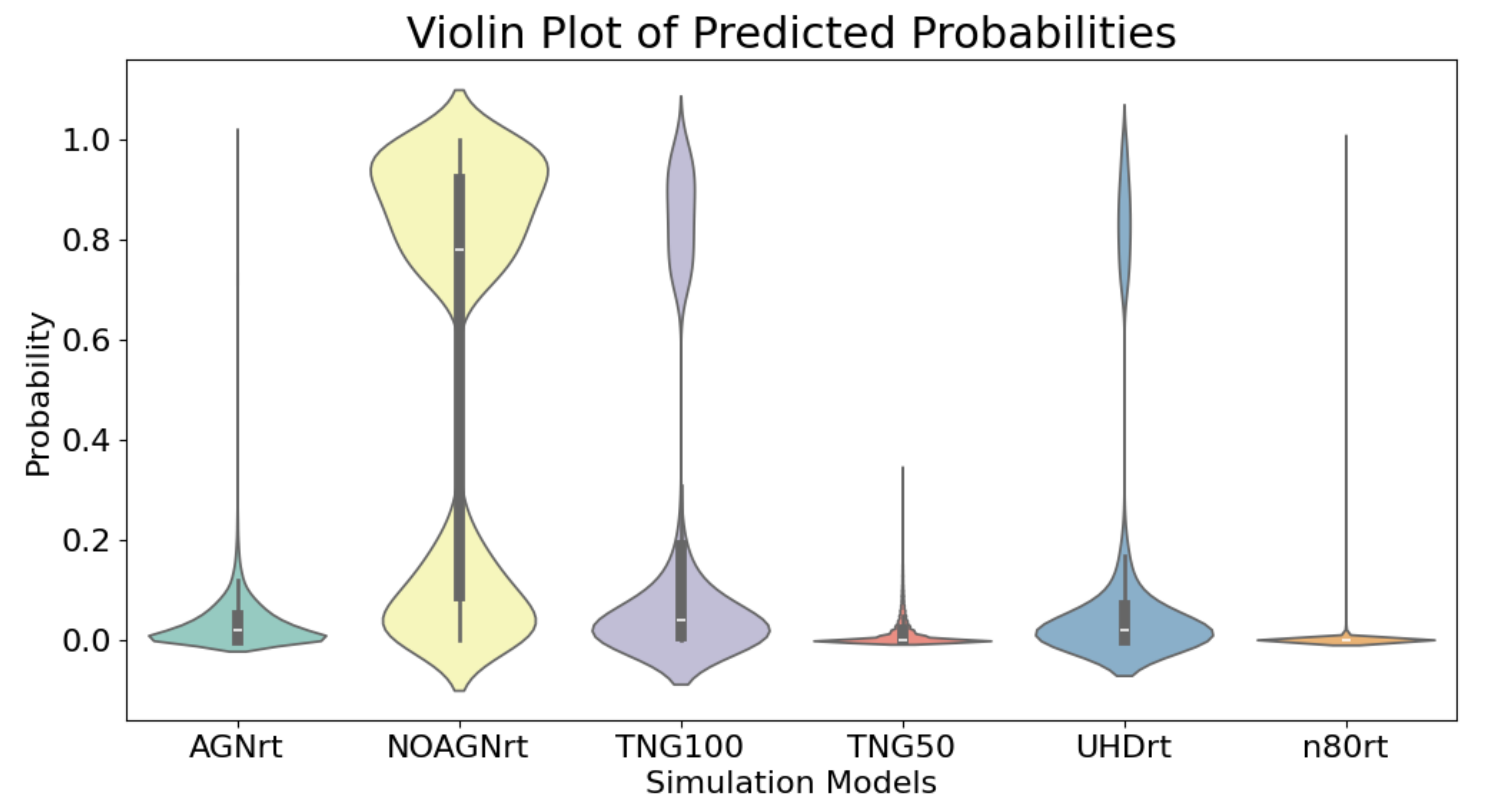




Zhou,Buck+2024 subm.

MODEL COMPARISON & MODEL MISSPECIFICATION



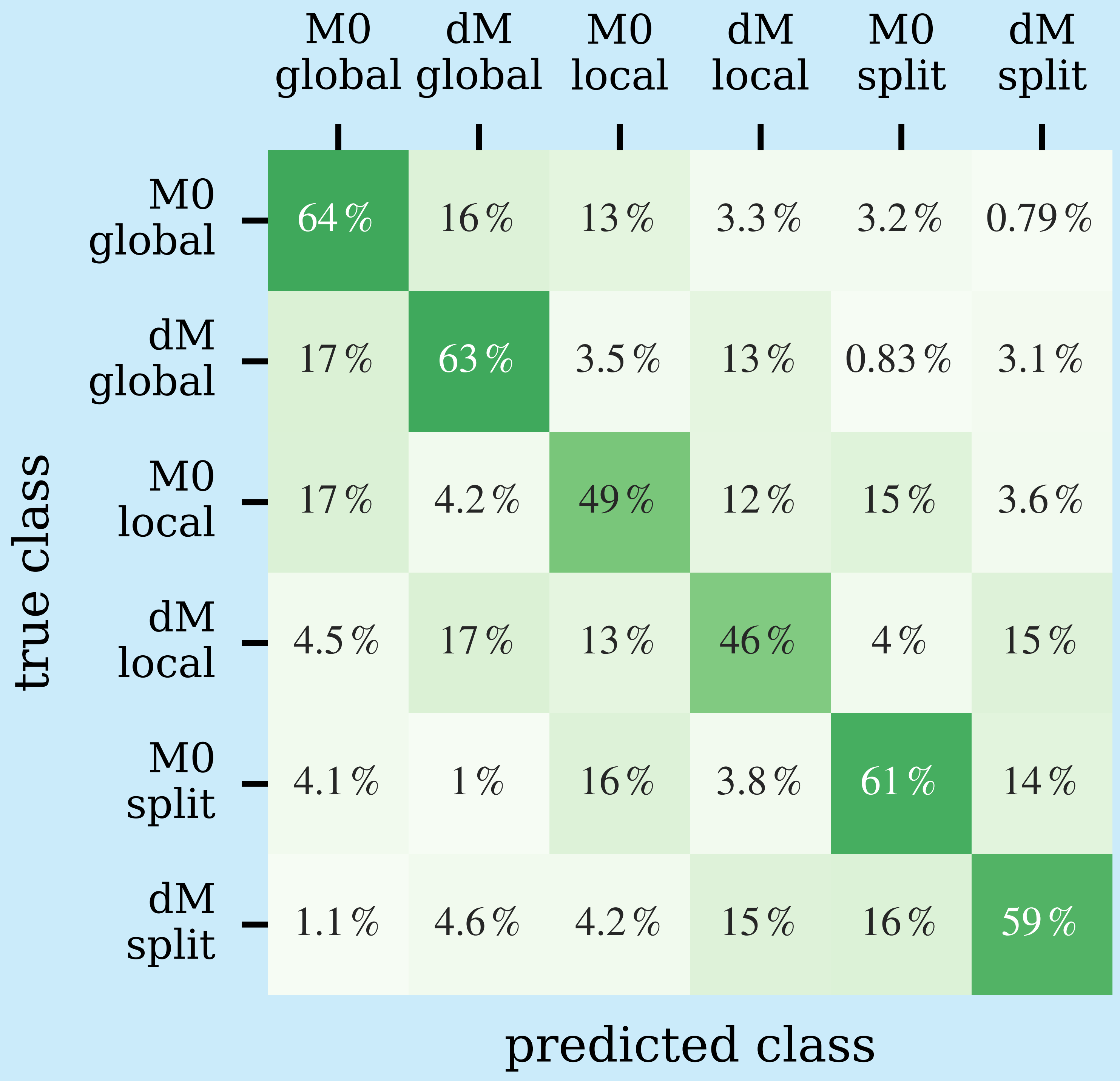


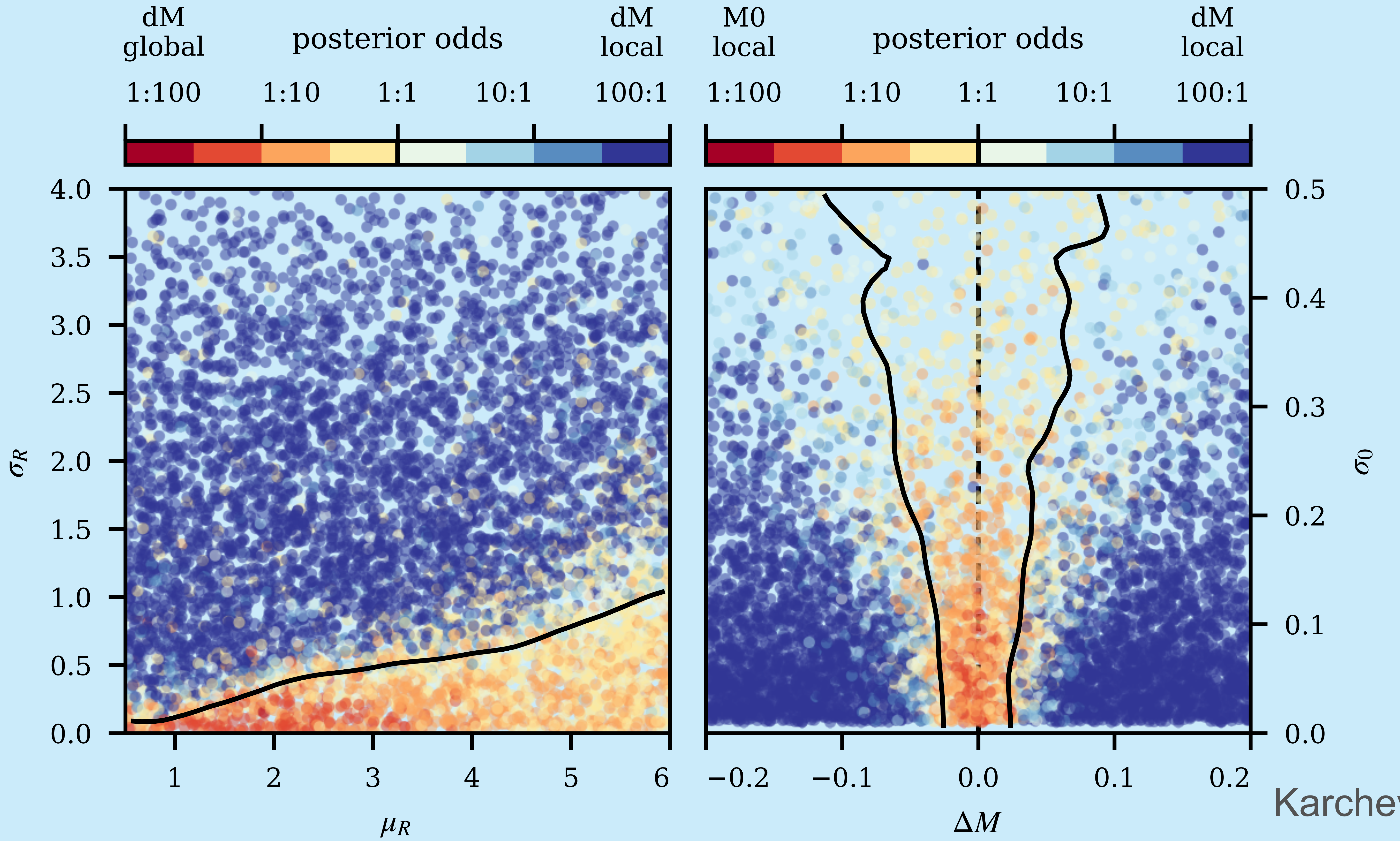
Zhou,Buck+2024 subm.

MODEL COMPARISON/SELECTION

SimSIMS: Simulation-based Supernova Ia Model Selection with thousands of latent variables

Konstantin Karchev[1] **Roberto Trotta[1,2,3,4]** **Christoph Weniger[5]**





Karchev+2024

Forward Models & Emulators

LEARNING THE SOLUTION OF ODES AND PDES — NEURAL ODES, OPERATOR LEARNING AND PINNS

- ODEs are good for:
  - population models
  - motion of the planets
  - structural integrity of a bridge
  - fluid dynamics
  - …

ODEs are kind of easy — only derivatives with respect to one variable

PDEs are more complicated — derivatives with respect to many variables and differential equations are local while solutions exhibit non-local properties

- Traditional solution: discretisation (in time and space) and iterative solution

NEURAL ODES, OPERATOR LEARNING AND PINNS

Neural ODE: $\frac{df}{dt} = h_\theta(x_0, t, p)$ (neural net = right hand side of diff eq.

solution: integrate entire neural net.)

Neural Operator: $G_\theta : X \to Y \quad u \mapsto G_\theta(u)$ with X, Y function spaces (infinite dimensional)

(neural net approximates the operator i.e. the map between function space)

PINN: $f(x, t) = h_\theta(x, t, p)$ with $\frac{df}{dt} = \frac{dh_\theta}{dt}, \quad \frac{df}{dx} = \frac{dh_\theta}{dx}$ need to fulfil the diff eq.

(solution is given by neural net, autodiff and diff eq. are used in loss)

NEURAL ODES

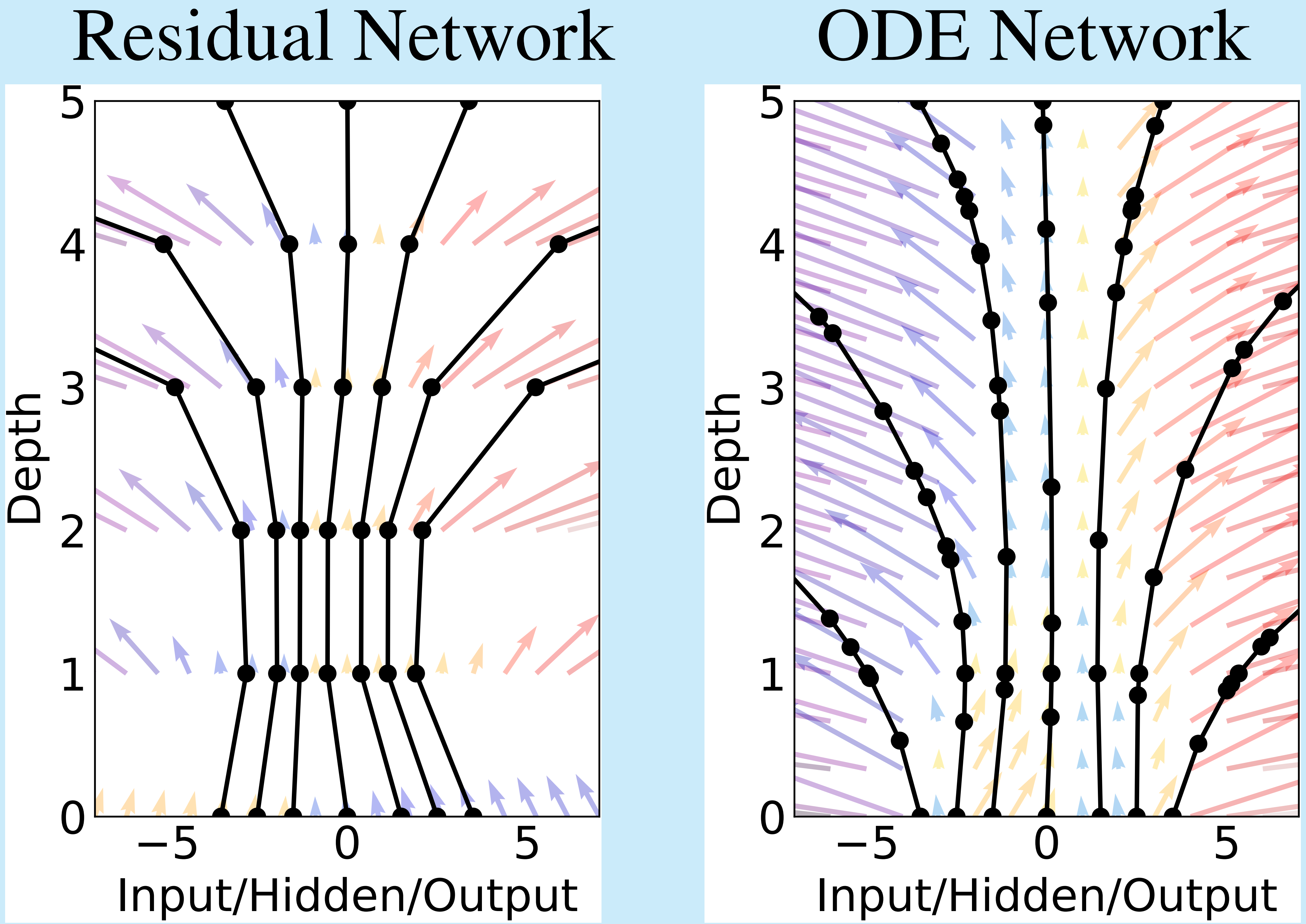
- Traditional solution: discretisation (in time and space) and iterative solution

Euler discretization

$$h_{t+1} = h_t + f_t(h_t, \theta_t)$$

$$h(t + \Delta t) = h(t) + \Delta t \cdot f(t, h(t), \theta)$$

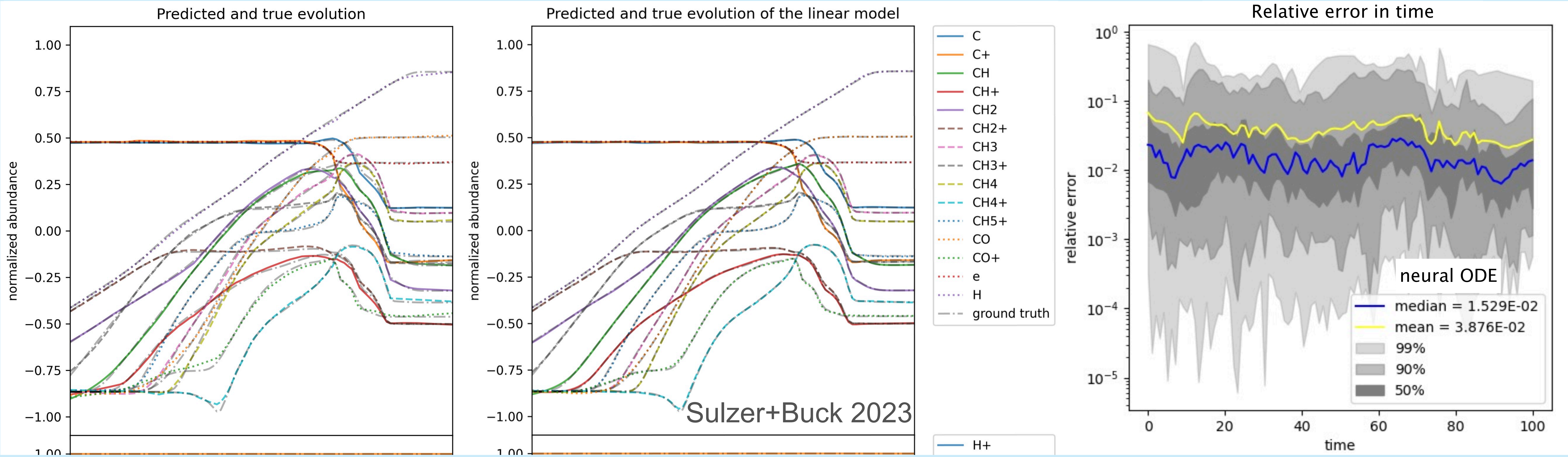
$$\frac{h(t + \Delta t) - h(t)}{\Delta t} = f(t, h(t), \theta)$$



Chen+2019

NEURAL ODES IN ASTROPHYSICS

- Neural Astrophysical Wind Models (Nguyen 2023)
- Neural ODEs as a discovery tool to characterize the structure of the hot galactic wind of M82 (Nguyen+2023)
- Speeding up astrochemical reaction networks with autoencoders and neural ODEs (Sulzer+Buck 2023)



PHYSICS INFORMED NEURAL NETS

Loss function for PINNs

Differential Equation: $\mathcal{F}[u(x,y)] = f(x,y)$

Dataset: $(x_i, y_i, u_i);\ i = 1, \ldots, N_{data}$

Collocation points: $(x_j, y_j); j = 1, \ldots, N_C$

Initial Condition: (x_o, y_0, u_0)

$$L_{DiffEq} = \frac{1}{N_c}\sum_{j=1}^{N_c}\left(\mathcal{F}[u(x_j,y_j)] - f(x_j,y_j)\right)^2$$

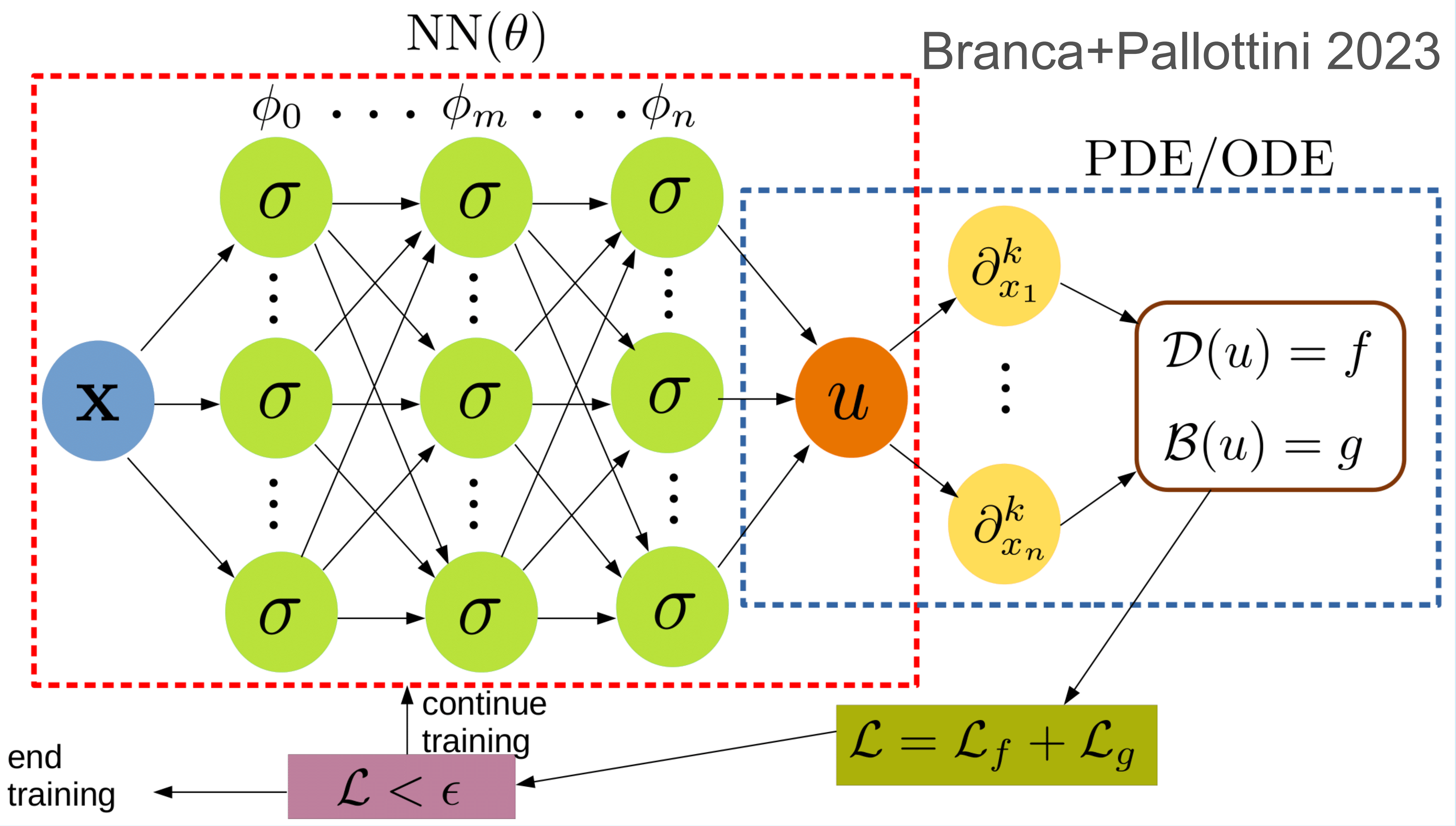
$$L_{total} = \omega_{data} \cdot L_{data} + \omega_{DiffEq} \cdot L_{DiffEq} + \omega_{IC} \cdot L_{IC}$$

$$L_{data} = \frac{1}{N_{data}}\sum_{i=1}^{N_{data}}(u(x_i,y_i) - u_i)^2$$

$$L_{IC} = (u(x_0,y_0) - u_0)^2$$

PINNS IN ASTROPHYSICS

- Physics-informed neural networks for modeling astrophysical shocks (Moschou+2023)
- Probing the solar coronal magnetic field with physics-informed neural networks (Jarolim+2022)
- Physics-informed neural networks in the recreation of hydrodynamic simulations from dark matter (Dai+2023)
- Physics informed neural networks for simulating radiative transfer (Mishra+Molinaro 2021)
- Neural networks: solving the chemistry of the interstellar medium (Branca+Pallottini 2023)

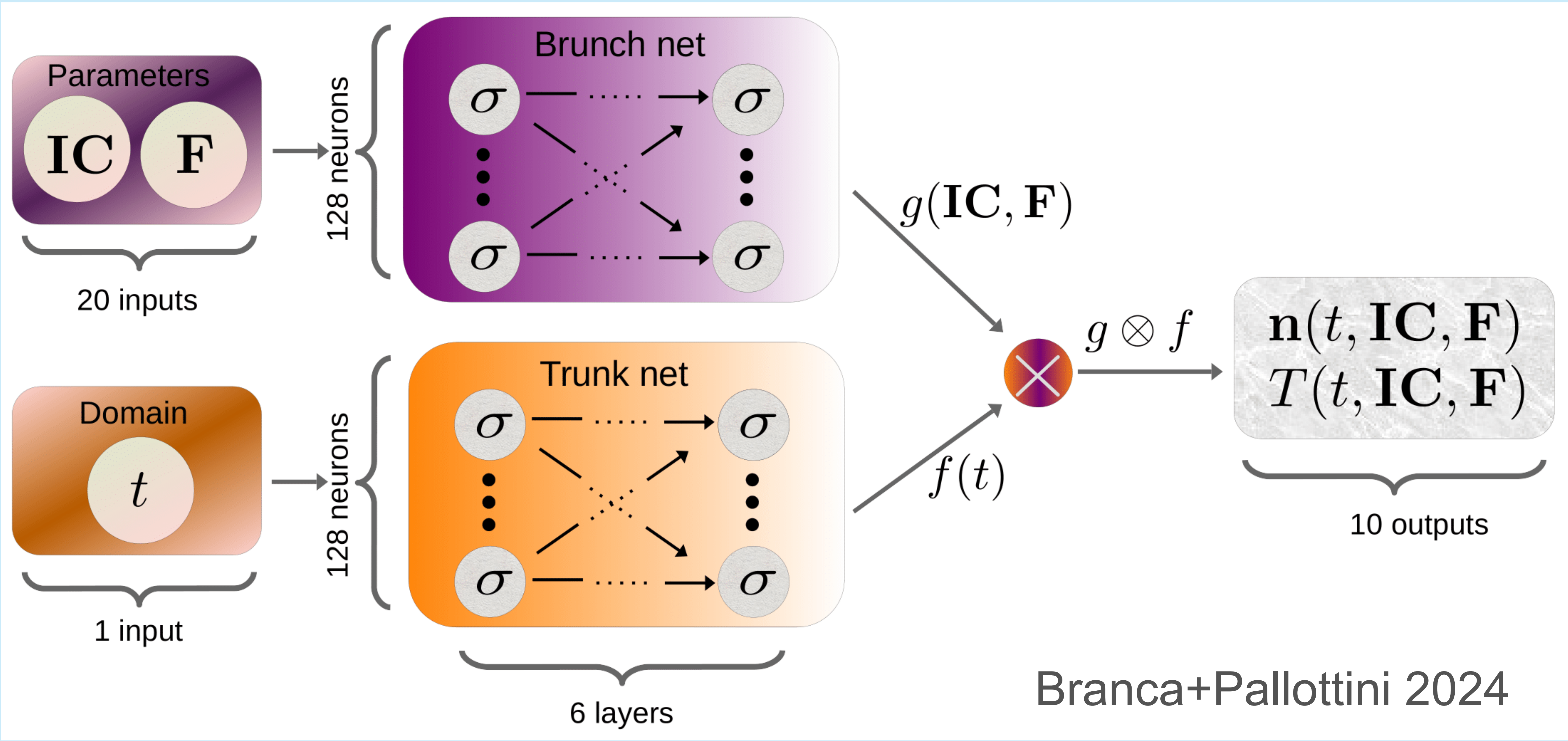


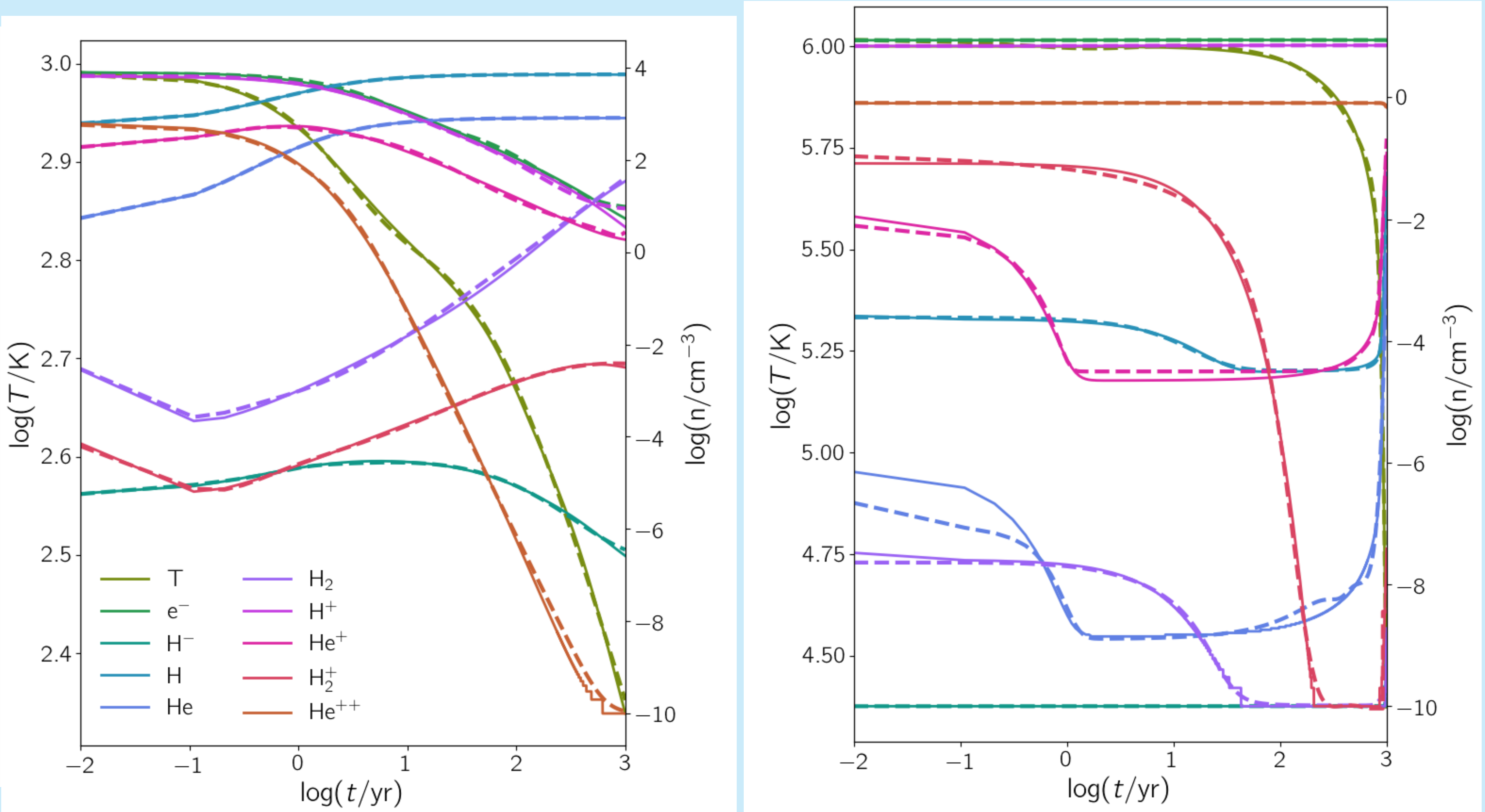
DEEP OPERATORS IN ASTROPHYSICS

- PPDONet: Deep Operator Networks for Fast Prediction of Steady-State Solutions in Disk-Planet Systems (Mao+2023)

- Emulating the interstellar medium chemistry with neural operators (Branca+Pallottini 2024)

- CODES Benchmark for neuralODEs and Operator Learning for astrochemistry (Janssen,Sulzer+Buck 2024)





SUMMARY & CONCLUSION

Main take away:

scientific motivated inductive bias helps to be more robust, more data efficient and better interpretable

My personal message:

Write better code!
Share more data!
Build more open-source software!

This will accelerate research cycles and lets you engage with peers early on!


THIS IS YOUR MACHINE LEARNING SYSTEM?
YUP! YOU POUR THE DATA INTO THIS BIG PILE OF LINEAR ALGEBRA, THEN COLLECT THE ANSWERS ON THE OTHER SIDE.
WHAT IF THE ANSWERS ARE WRONG?
JUST STIR THE PILE UNTIL THEY START LOOKING RIGHT.
DATA
ANSWERS


Differentiable Simulators

DIFFERENTIABLE SIMULATORS