Mario Pasquato
Marie Curie Fellow
Padua Observatory
(Italy)

Michela Mapelli
Alessandro Ballone
Piero Trevisan

Image in science out?

A proof of concept with deep learning on molecular cloud simulations

Mario Pasquato
Marie Curie Fellow
Padua Observatory
(Italy)

Michela Mapelli
Alessandro Ballone
Piero Trevisan

Image in science out?

not so fast

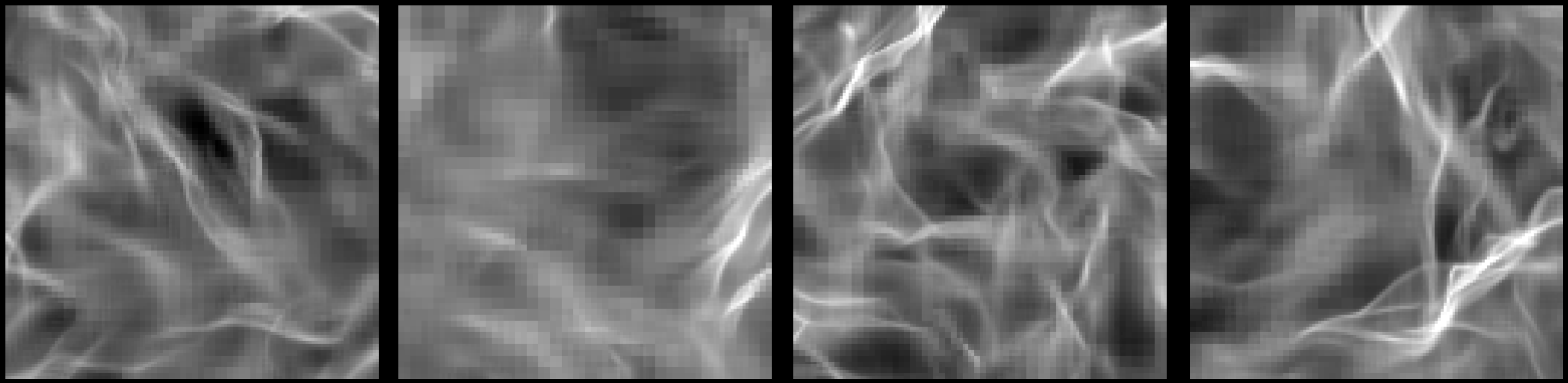
A proof of concept with deep learning on molecular cloud simulations

The astrophysical problem

- Turbulence in molecular clouds modulates star formation, physics still not fully understood [Elmegreen & Scalo 2004, Hennebelle & Falgarone 2012]
- Velocity power spectrum of turbulence can be measured directly through e.g. line-of-sight velocity [Koch 2019]

Question

- Can we measure the turbulence index of simulated turbulent gas from density maps?
- In particular discriminate between Kolmogorov $P_v(k) = k^{-11/3}$ and Burgers $P_v(k) = k^{-4}$ spectra



Simulations

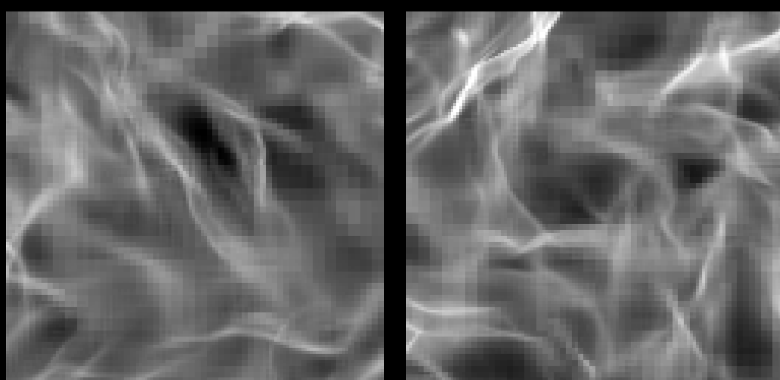
- 1000 simulations of turbulent gas with RAMSES2 [Teyssier 2002] AMR code
- 10x10x10 pc box, initially uniform density gas ($6.77\times10^{-22} g/cm^3$), total mass of $10^4 M_{sun}$.
- Gas kept isothermal at temperature T=10K
- Injected a divergence free, turbulent, supersonic (Mach 1.41) velocity field with spectrum index n=11/3 or 4
- Evolved for 0.5 Myr, solving Euler's equation with a Lax-Friedrichs Riemann Solver, periodic boundaries without self-gravity and magnetic fields

Train/test/holdout split

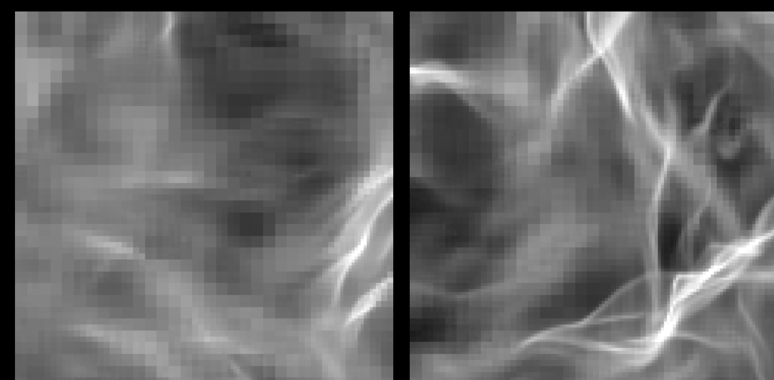
- 500 sims w. Kolmogorov index, 500 w. Burgers
- 400+400 build the train set -> 3 projections (x,y,z) X 4 flip/flop X 4-way cut = 38400 training images
- 50+50 in the test set = 4800 test images
- 50+50 never looked at (holdout set) = 4800 images

Images

- 250x250 pixels, grayscale; each image corresponds to ¼ of the box, seen in projection along an axis (x,y,z)
- Luminosity encodes log column density



Kolmogorov

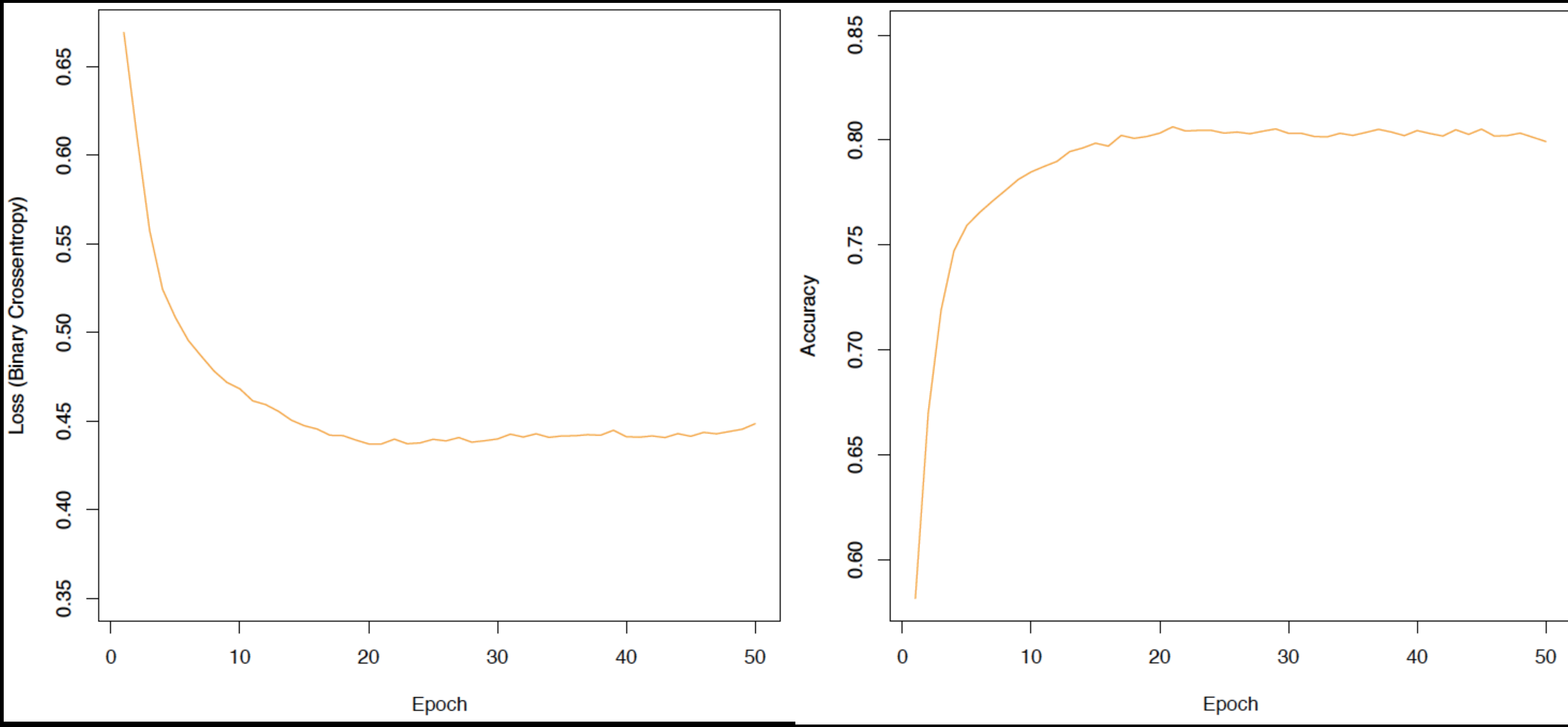


Burgers

DL setup

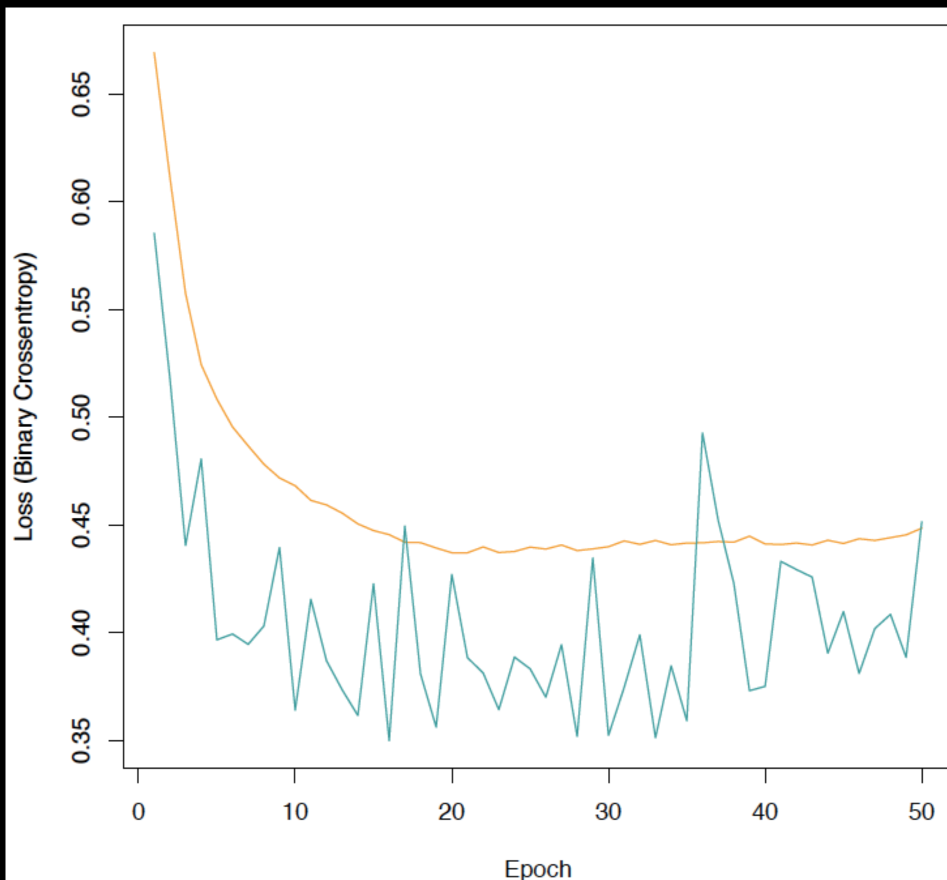
- Keras on top of Tensorflow on workstation with a Titan V GPU
- Four convolutional layers (with max pooling) + three dense layers
- RELU activations
- Dropout regularization
- RMSprop optimizer

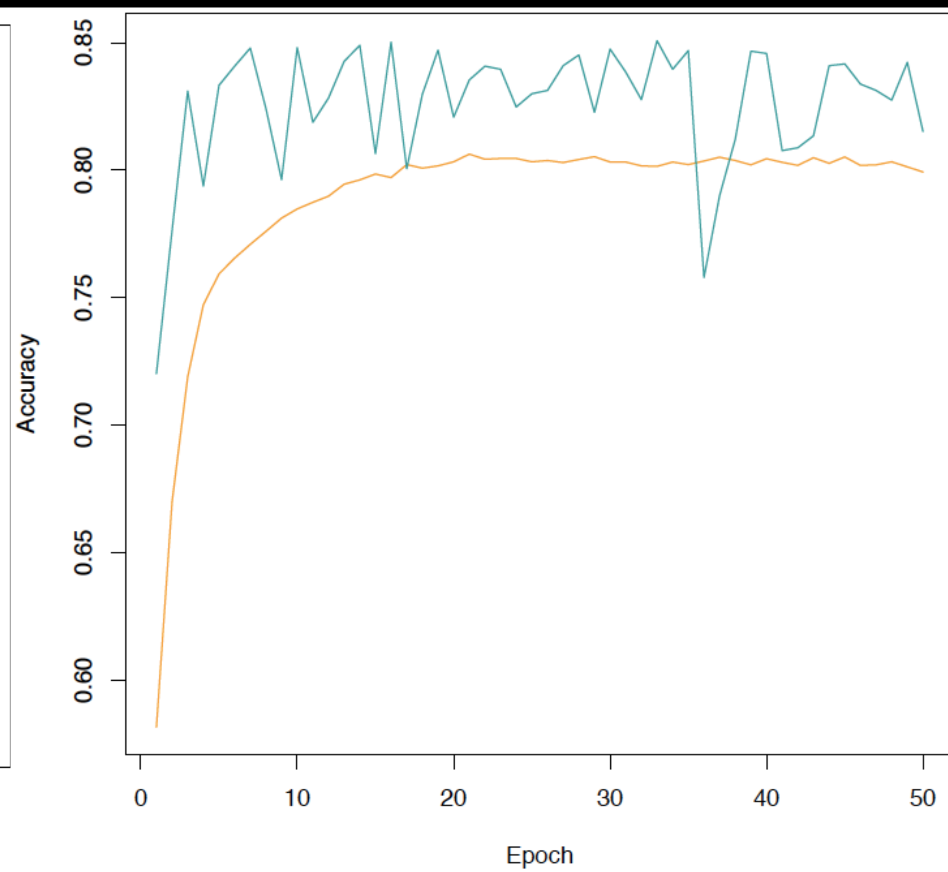
Learning curves



Training loss as a function of epoch

Learning curves

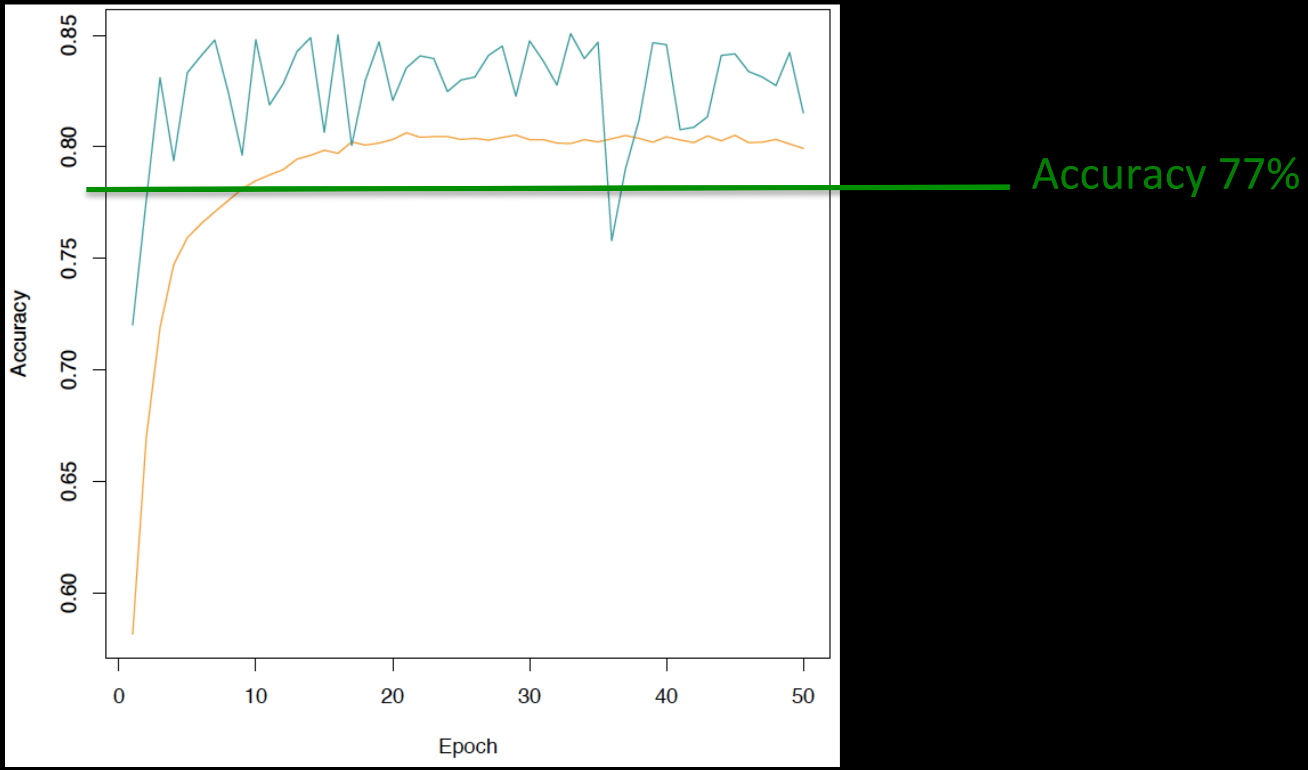




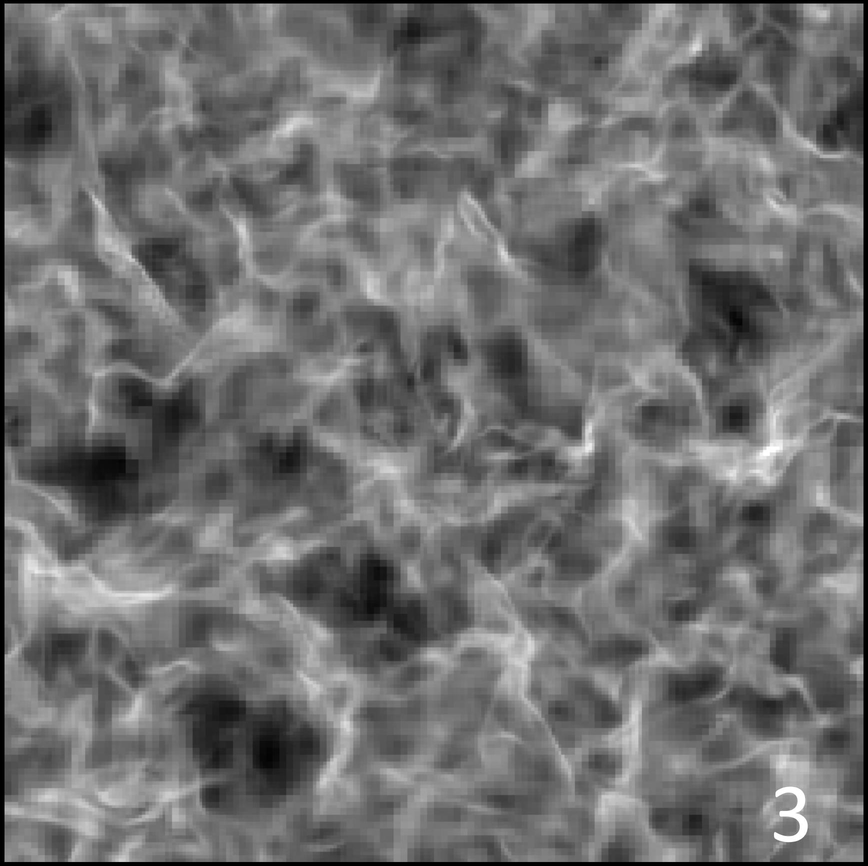
Training loss and validation loss as a function of epoch... something fishy? Dropout...

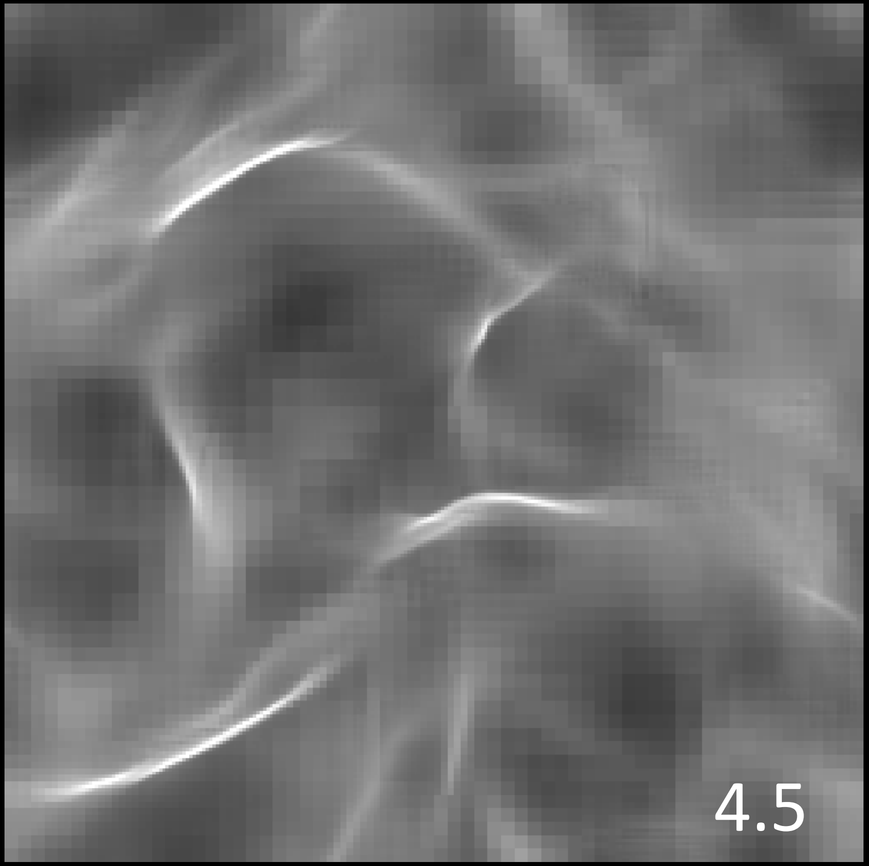
Performance on holdout set

	Predicted Kolmogorov	Predicted Burgers
Kolmogorov	2113	287
Burgers	812	1588



Testing on different indices



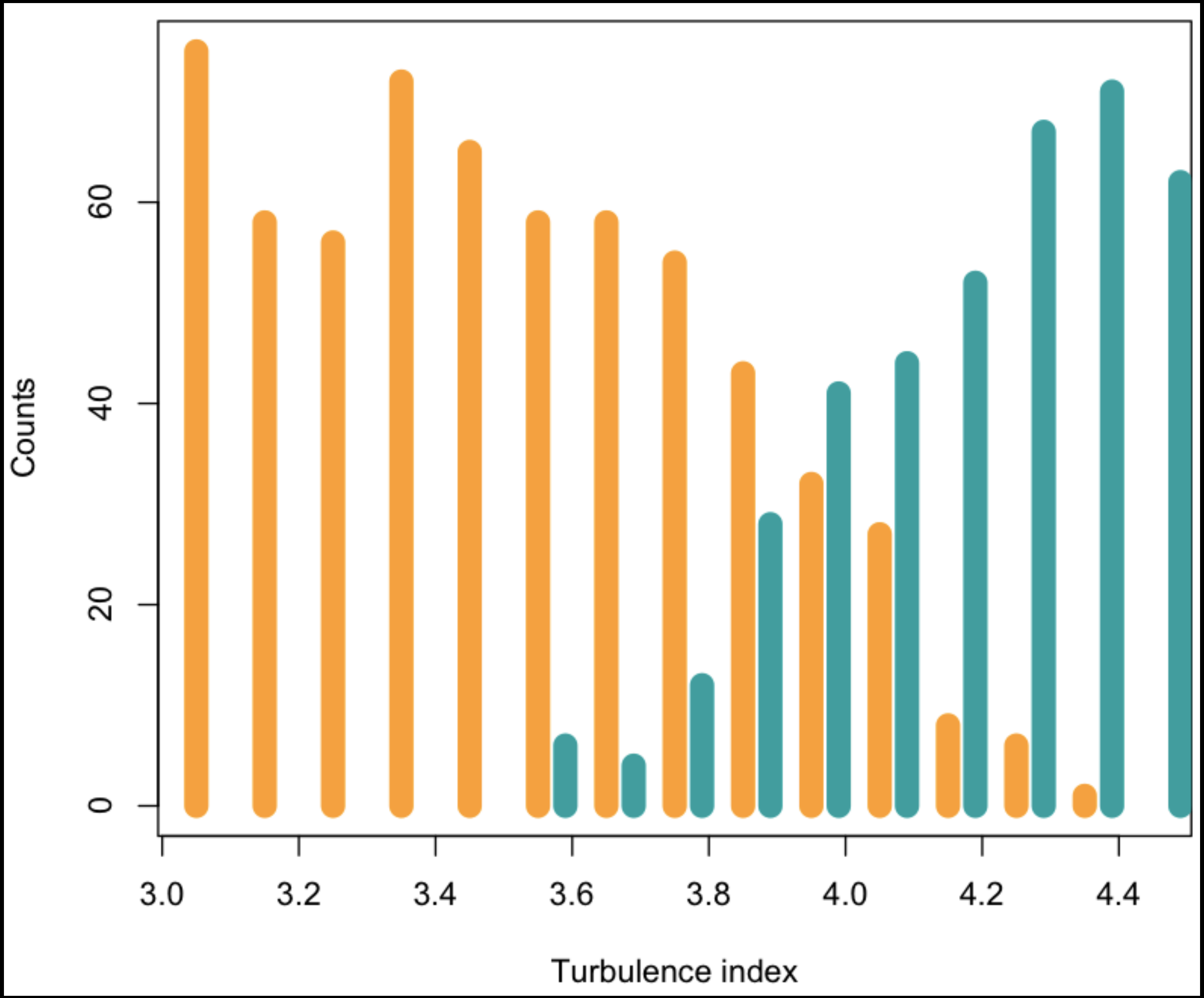


We ran 1000 more simulations with turbulence index that ranges continously from 3 (left) to 4.5 (right). What will the net predict?

Predictions

Predicted
Kolmogorov

Predicted
Burgers



Before we can use this for science

- What are the features used by the CNN? Genuine physical features or simulation artefacts?
  - Example: adaptive mesh refinement increases resolution in high density areas
- Zoom invariance?
  - we can't move closer to / away from a molecular cloud

Before we can use this for science

- What are the features used by the CNN? Genuine physical features or simulation artefacts?
  - Example: adaptive mesh refinement increases resolution in high density areas -> learned features could be useless/misleading on real data!
- Zoom invariance?
  - we can't move closer to / away from a molecular cloud -> zooming in/out should not affect classification performance

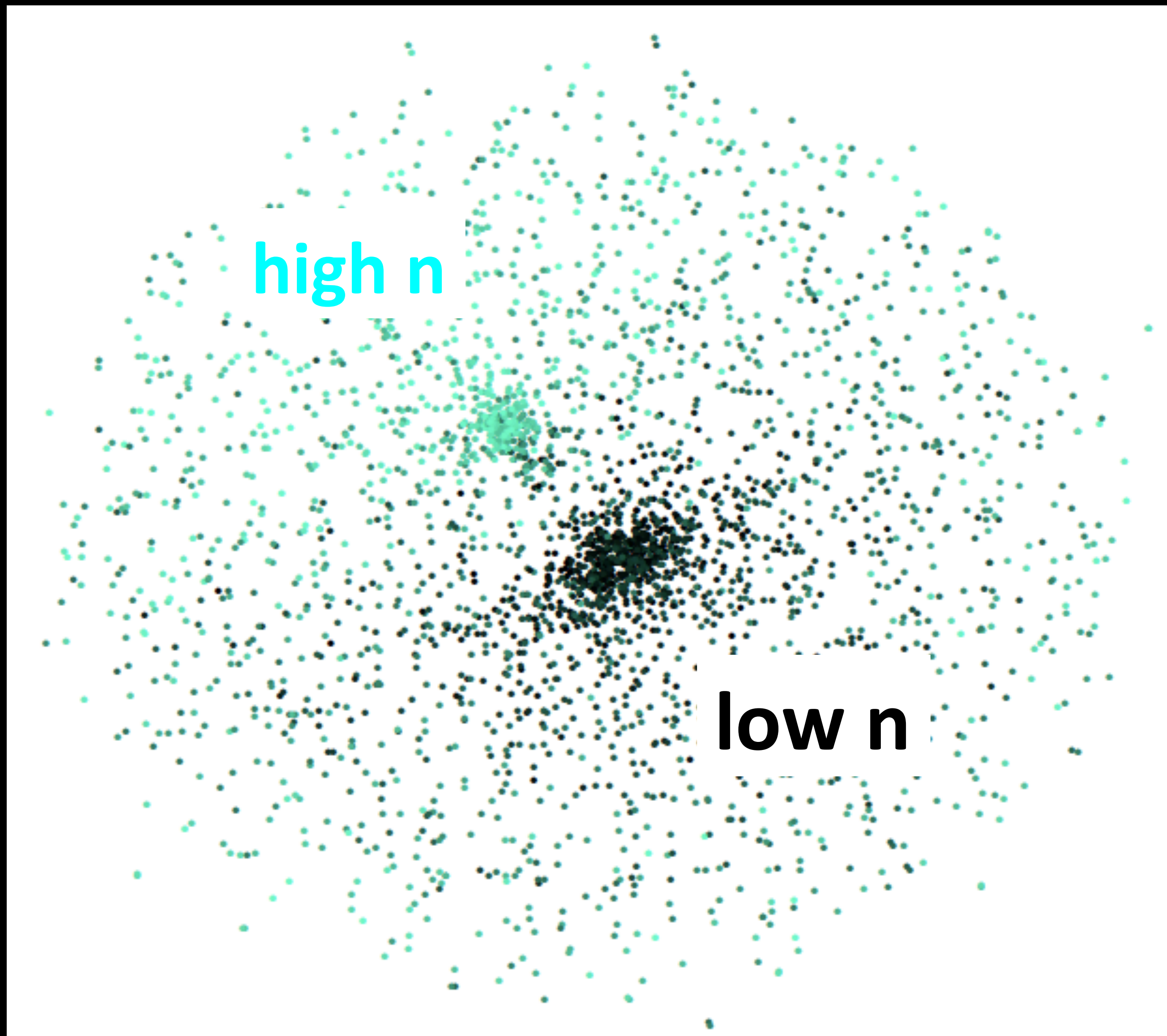
Possible solutions

- Instead of learning features, use e.g. Histogram of Oriented Gradient descriptor (Freeman & Roth 1994)
- Train a conventional machine learning algorithm on HOG features, e.g. SVM
- No point in using a CNN if it does not do better than this

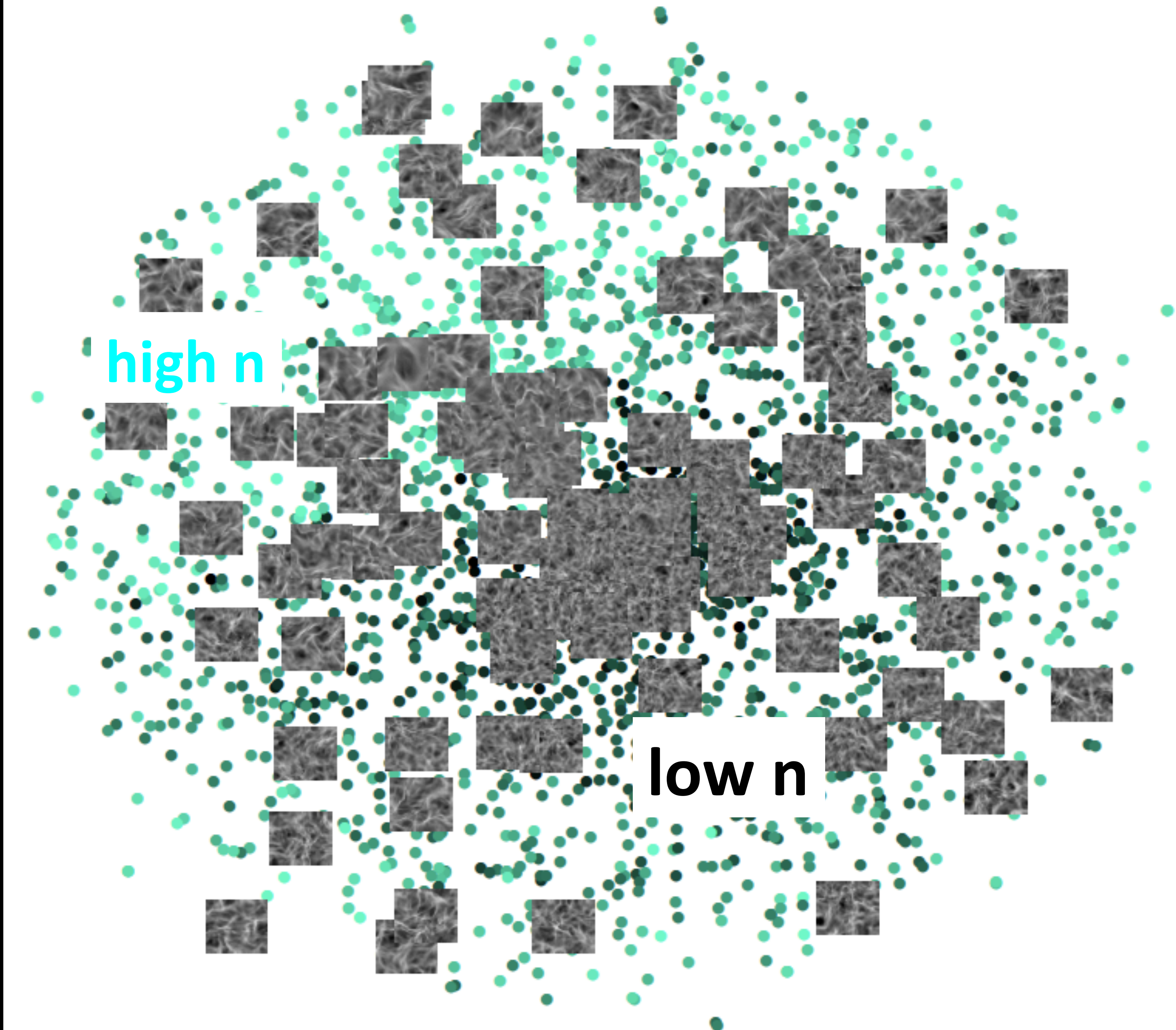
OR

- Learn features on data, use transfer-learning on the simulations

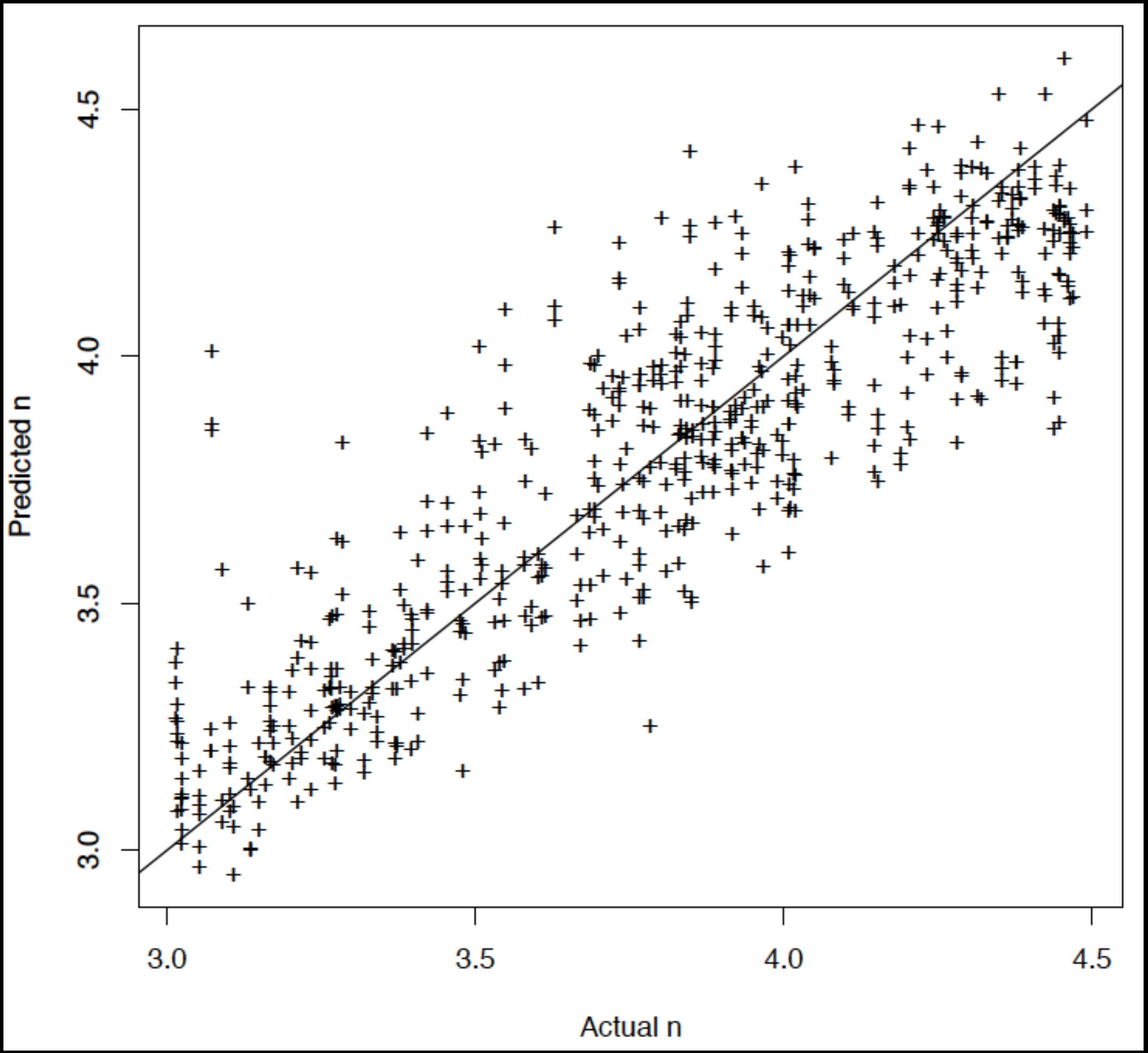
A t-SNE look at HOG features



12 cells
8 orientations


high n
low n

Predict the turbulence index with support-vector regression



Support vector regression on HOG features, tested on a holdout set

Questions?

This project has received funding from the
European Union's Horizon 2020
research and innovation programme
under the Marie Sklodowska-Curie grant agreement No.
664931

Can we do better with deep learning?

qua ci devi mettere
- lo schema della rete
- la learning curve
- un po' di hyperparameter optimization (forse)
- la roc curve su holdout
- confronto con HOG features + svm per classificazione (anche con ROC curve)
- t-SNE delle HOG features e magari dei pesi/output dell'ultimo layer convoluzionale
- Accenna a
  - saliency maps
  - transfer learning dal problema di regressione a questo

Is the CNN learning physically relevant information?

- Check with saliency maps
- Make sure the learned features are not affected by artifacts of the simulation procedure; solutions:
  - don't learn features (use e.g. HOG)
  - learn features somewhere else (transfer learning)
  - debias against features you do not want to learn (e.g. adversarial debiasing)
- Make simulations realistic (i.e. remove artifacts). Check realism:
  - a person cannot tell simulation and reality apart
  - a CNN classifier cannot tell (on the same features used for the science problem)
  - anomaly detection with autoencoders trained on real data, applied to simulations