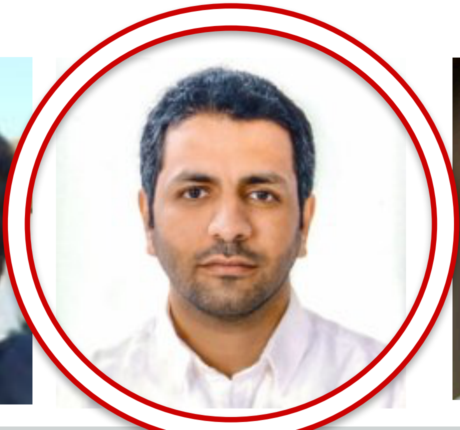
LSST: Codes and GPz

Matt Jarvis (Oxford)

On behalf of LSST-DESC-photo-z

(but mainly Ibrahim Almosallam)



GPz: Almosallam et al. 2016a,b

Using Gaussian Processes (not neural nets) for machine learning

- In a **Gaussian process**, every point in some continuous input space is associated with a normally distributed random variable.

Advantages

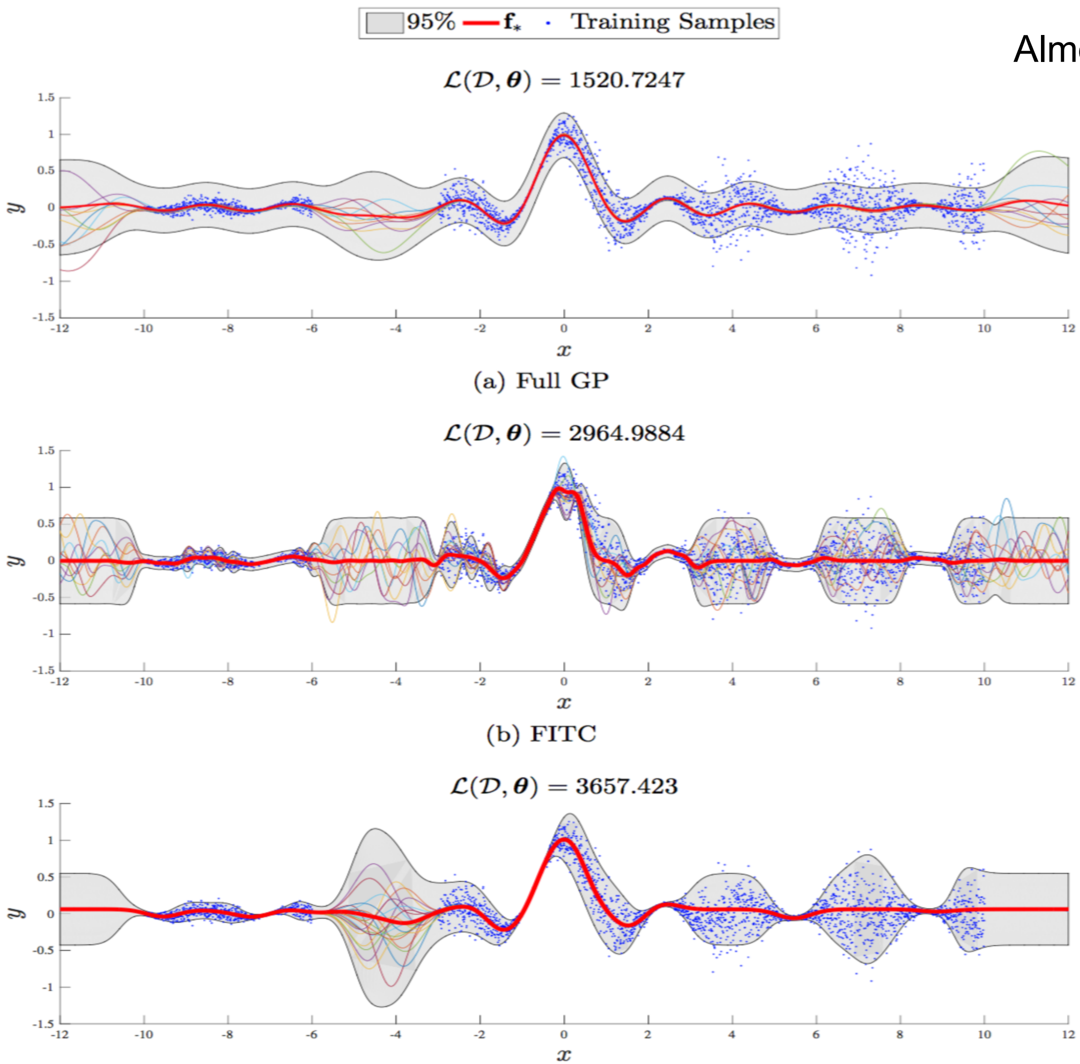
- Naturally Bayesian
- Do not have to define NN structure, N layers/hidden layers etc etc

GPz - Advances from standard GPs and other ML methods

- Cost sensitive learning (weight the training or test data in any way you want)
- Sparse GP with variable covariance/length (allows flexible modelling of the parameter space of interest without introducing more basis functions)
- Heteroscedastic noise (GP knows where there is not enough data and models the lack of data as part of the training)
- See Almosallam et al. 2016a,b, MNRAS for full details

GPz: Variance estimates

Almosallam, PhD Oxford, 2017



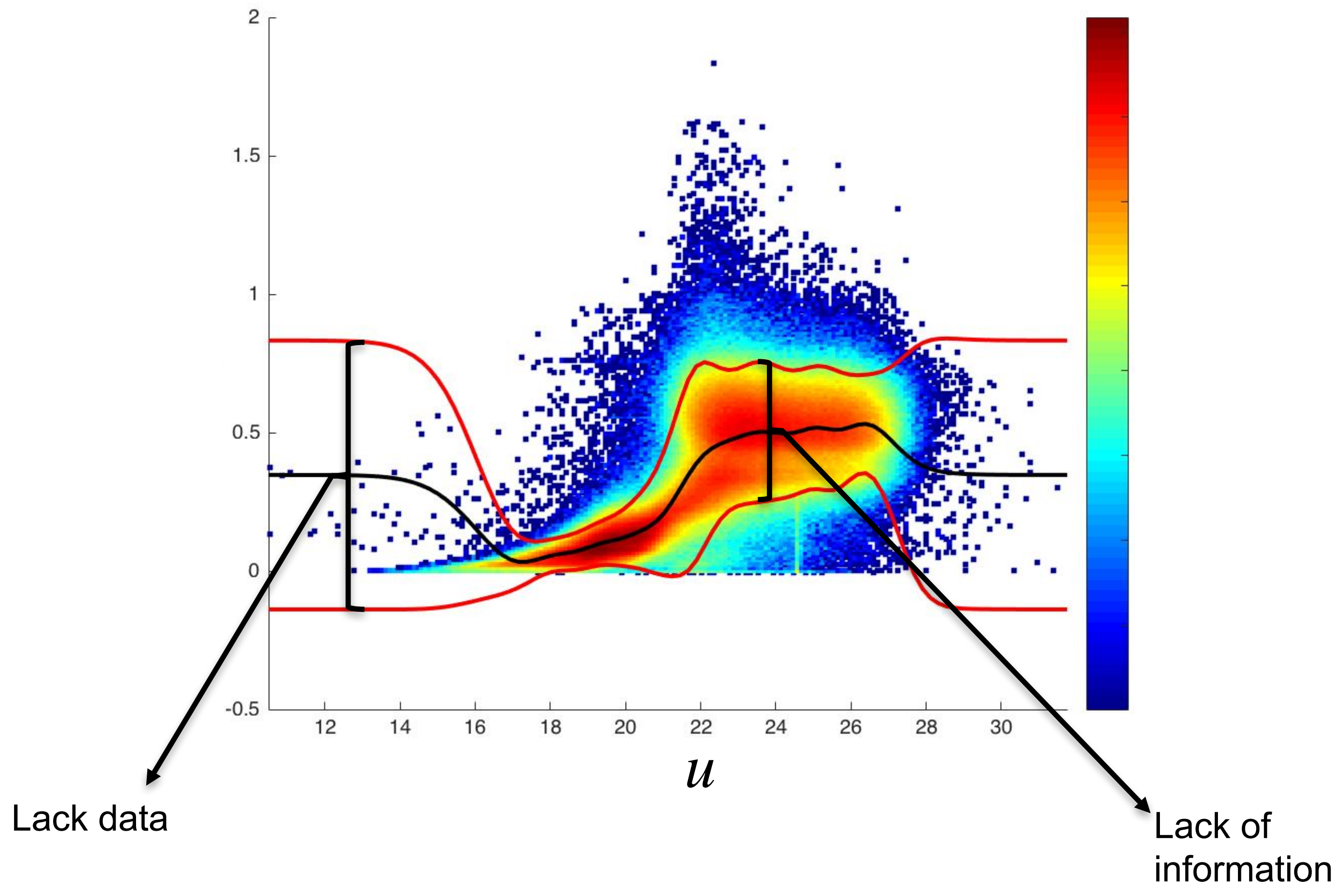
(a) Full GP

(b) FITC

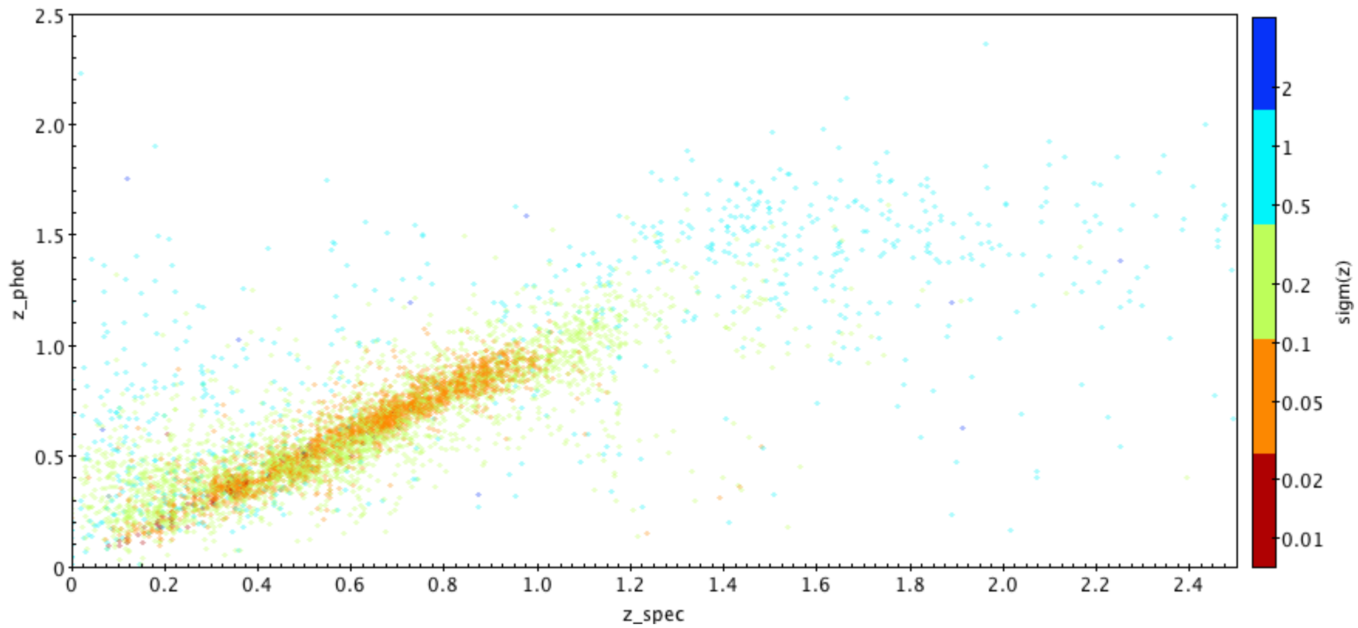
(c) BFM

Probability of z|u-band magnitude

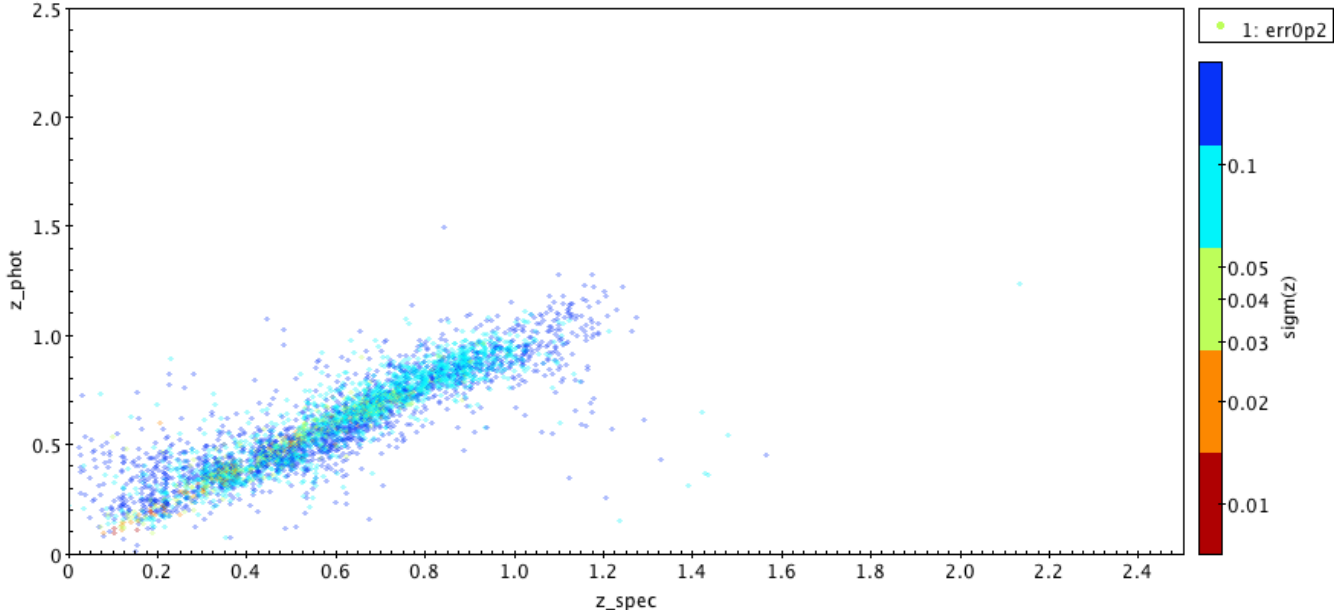
Almosallam, PhD Oxford, 2017



GPz Applied to HSC COSMOS

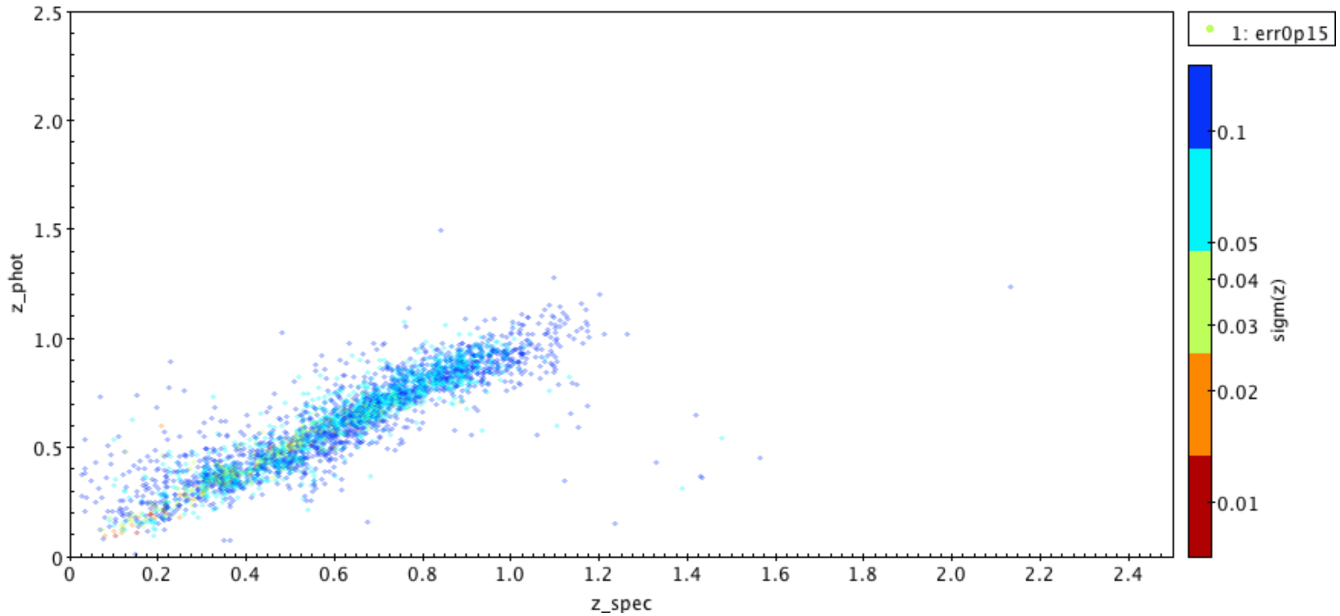


GPz Applied to HSC COSMOS



Using best 70% of data (i.e. sigma(z) < 0.2)

GPz Applied to HSC COSMOS

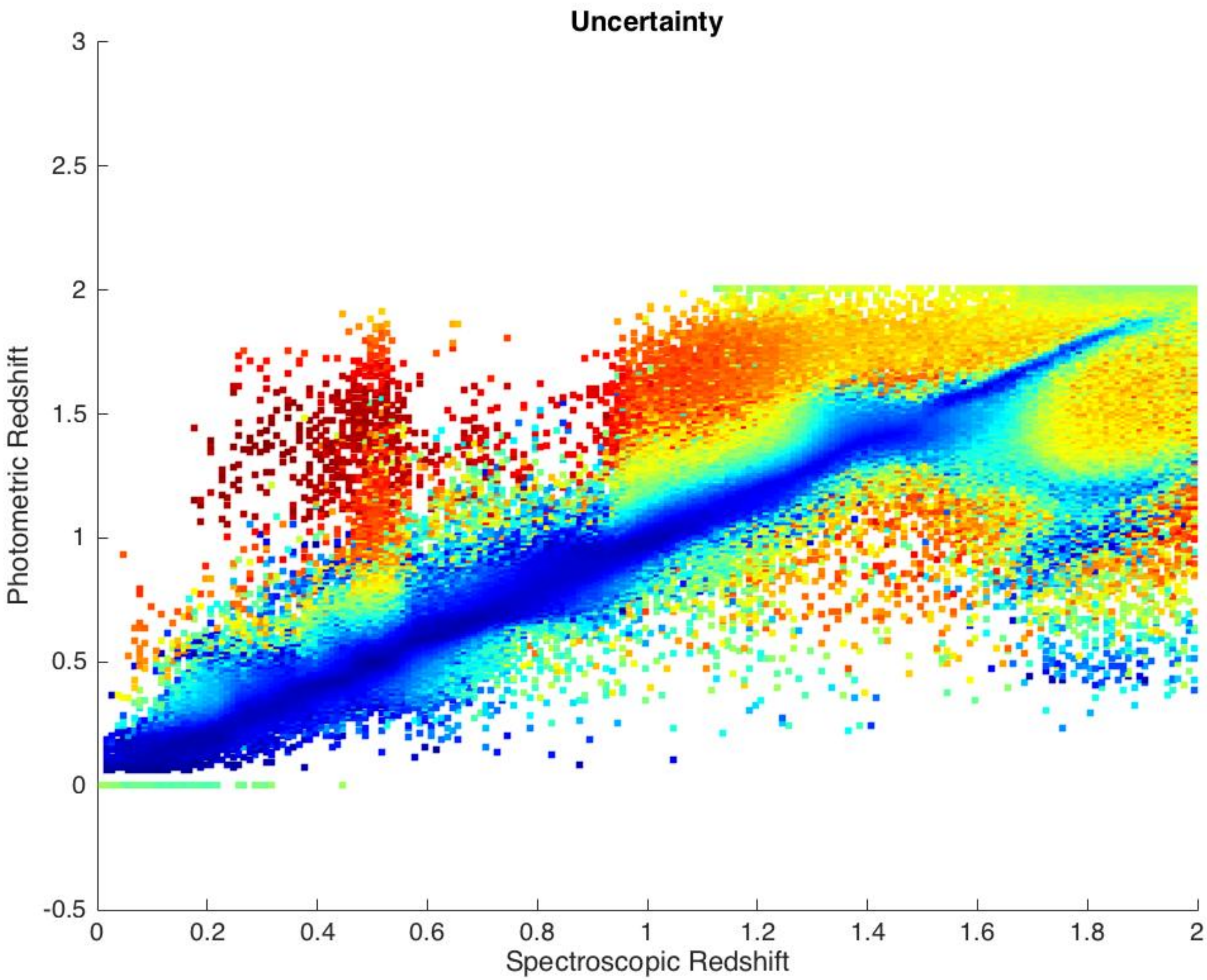


Using best 50% of data (i.e. sigma(z) < 0.135)

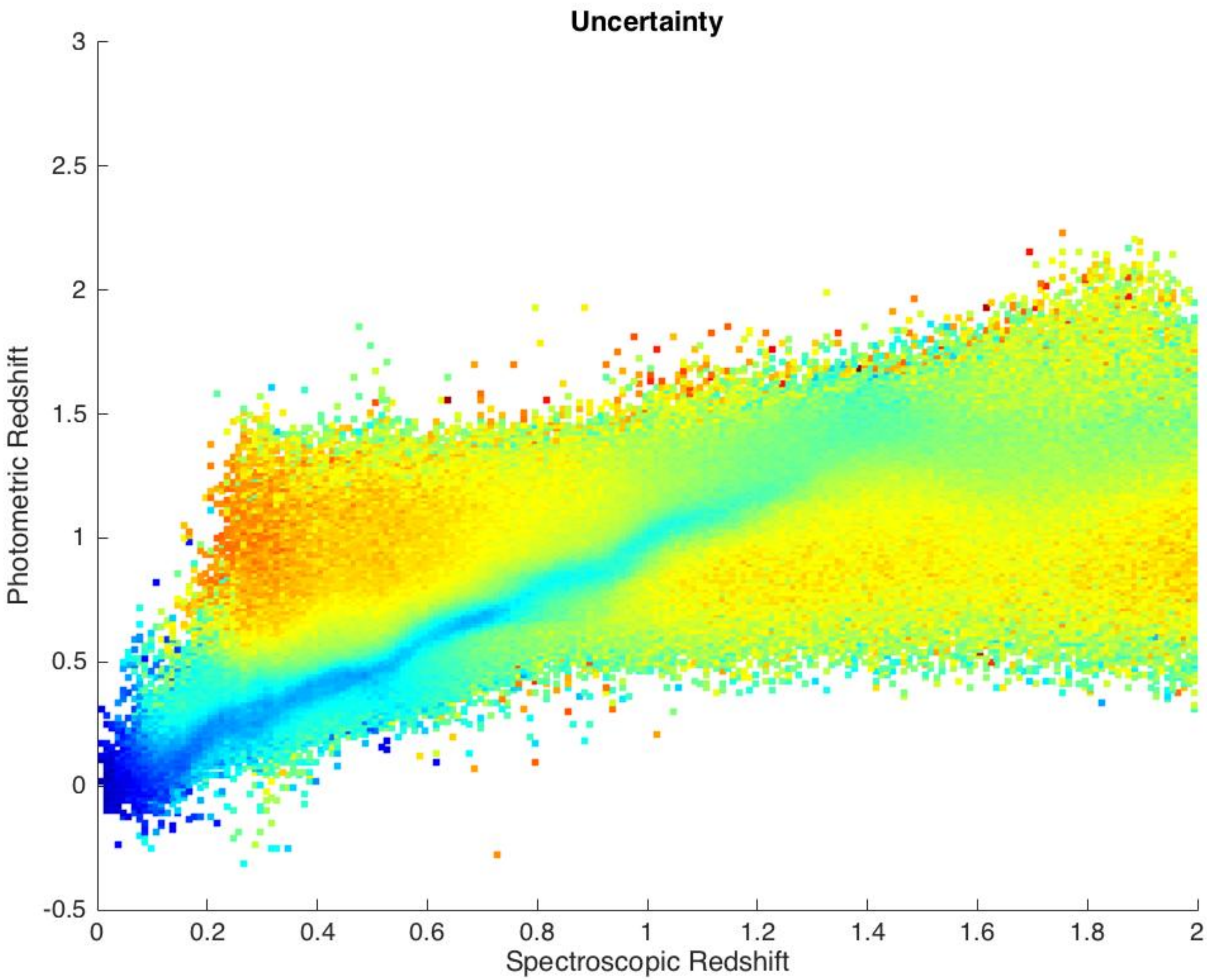
TPZ: 100 trees

Also applied to the Buzzard simulations as part of LSST data challenge along with other ML codes

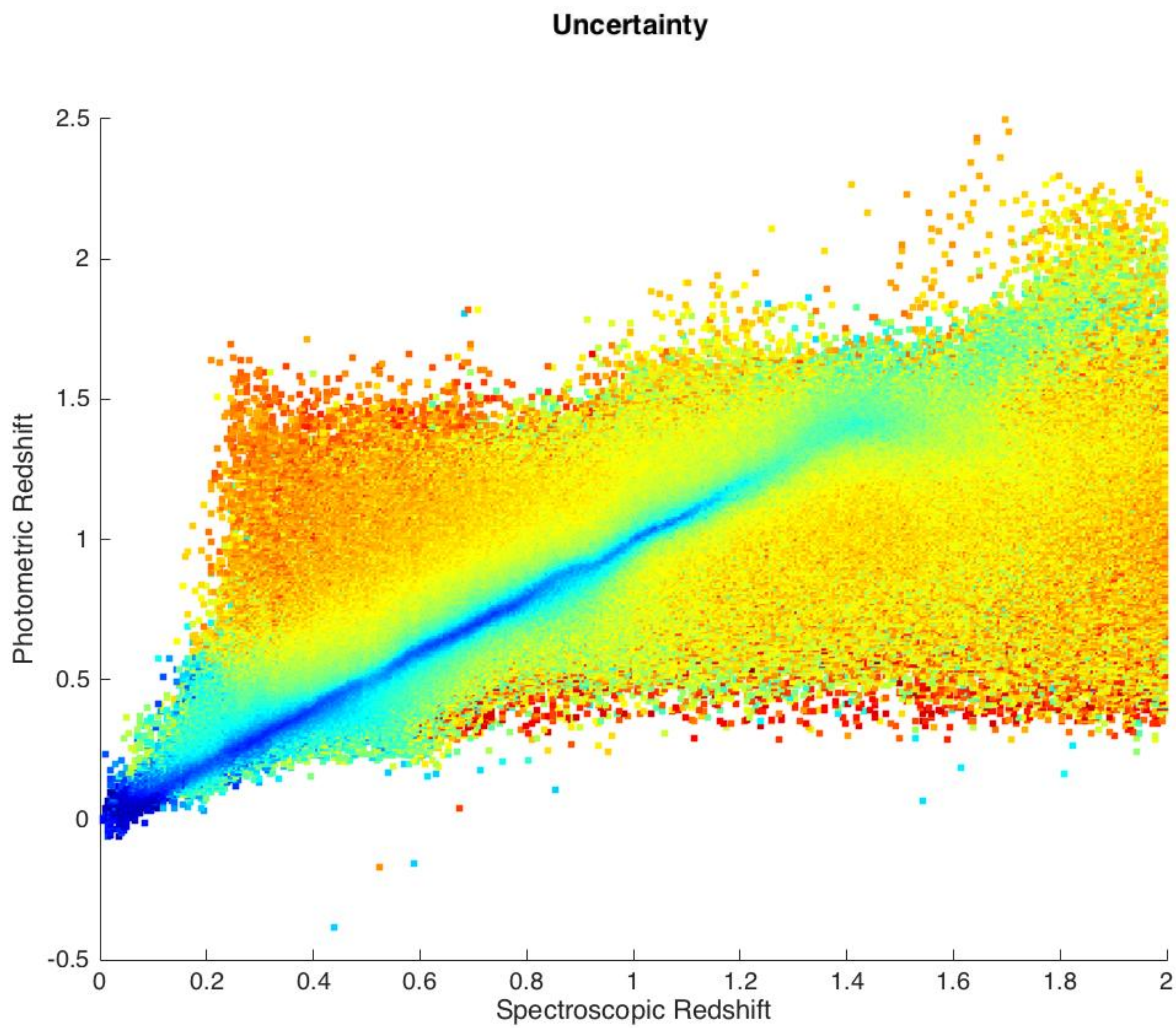
TPZ: 100 trees



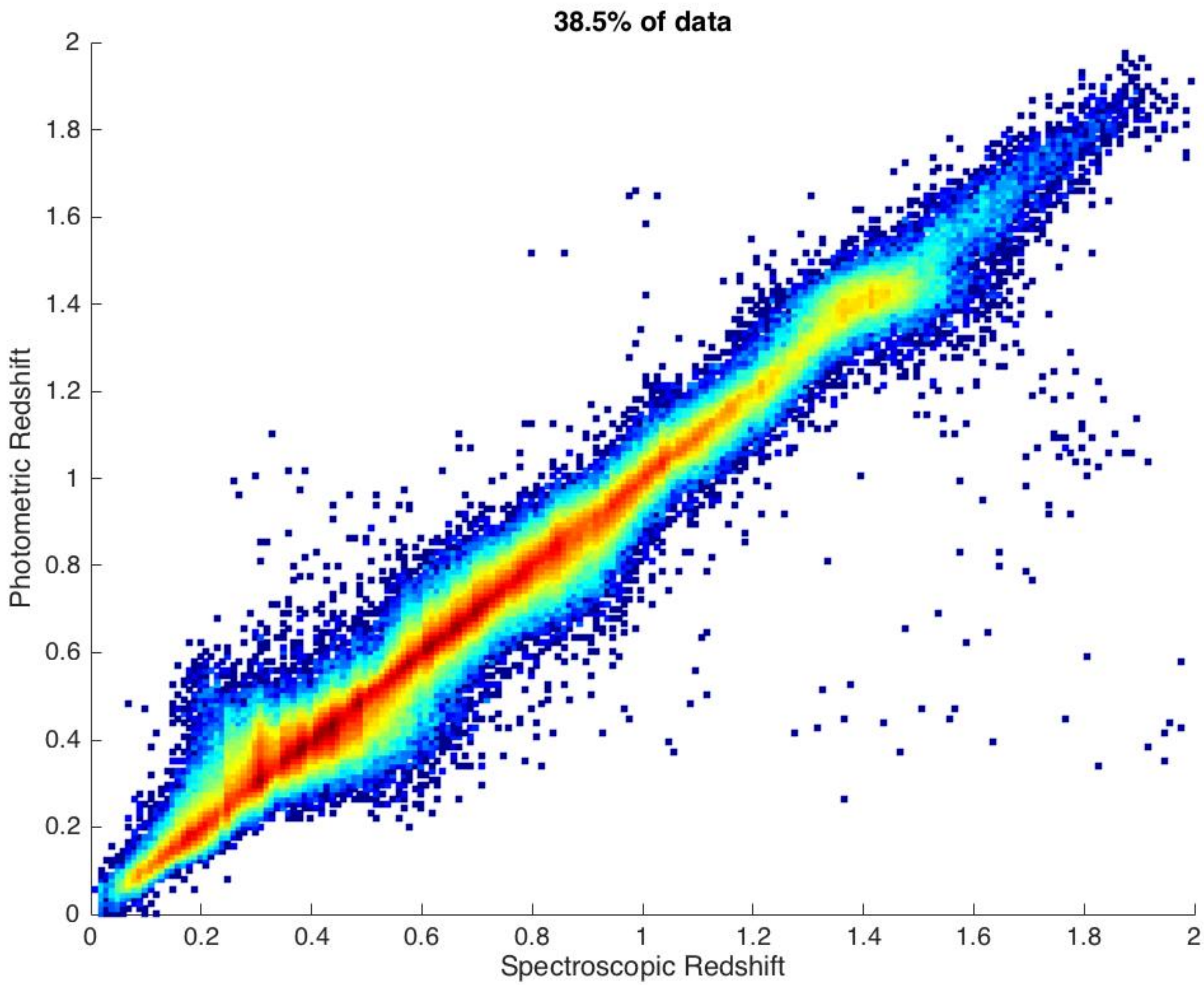
ANNz: 100 neurons



GPz: 100 basis functions

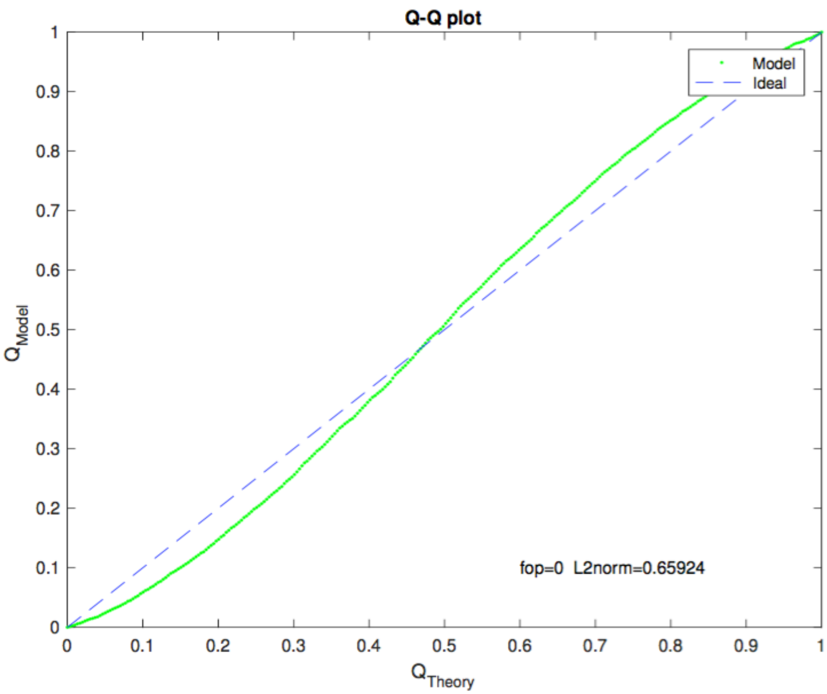


GPz: 100 basis functions

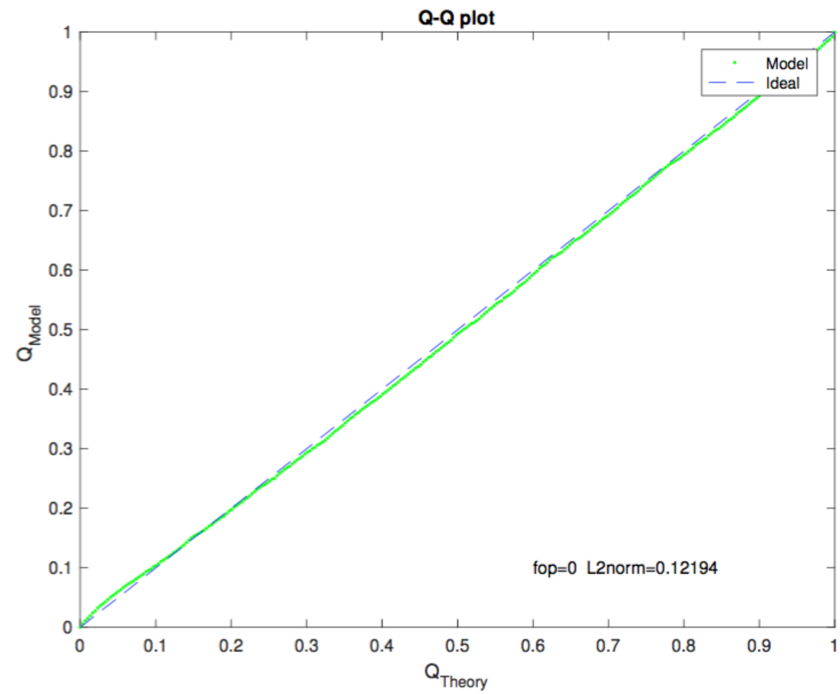


Are the posteriors reasonable?

Using GAMA + SDSS + UKIDSS (See Zahra's poster)



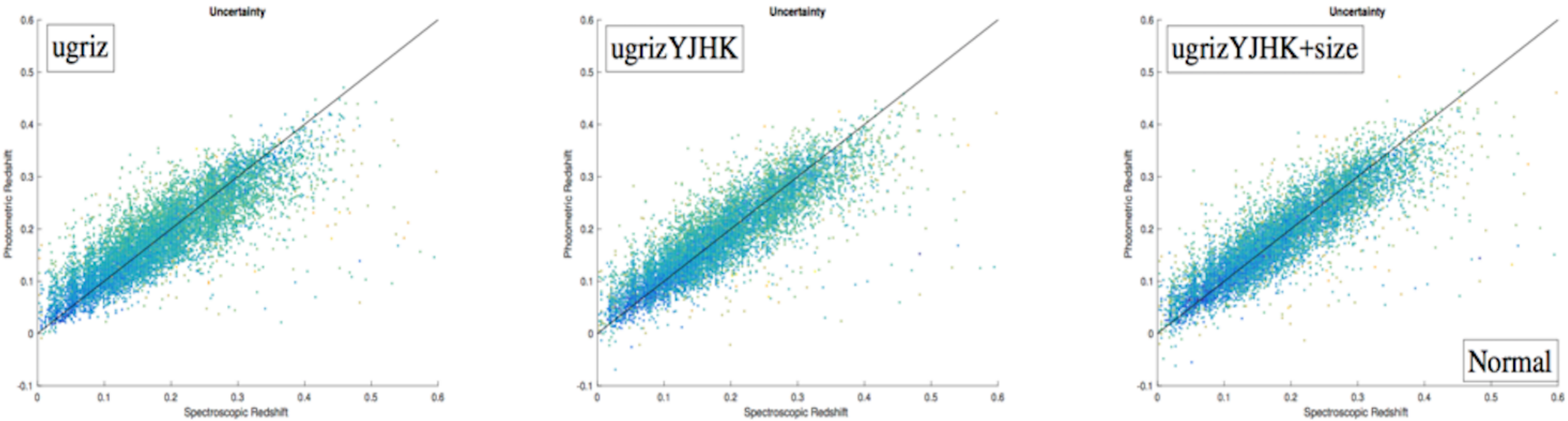
No Cost Sensitive Learning



Cost Sensitive Learning Weighted as 1/(1+z)

Gomes et al. in prep

Why stop at visible colours?

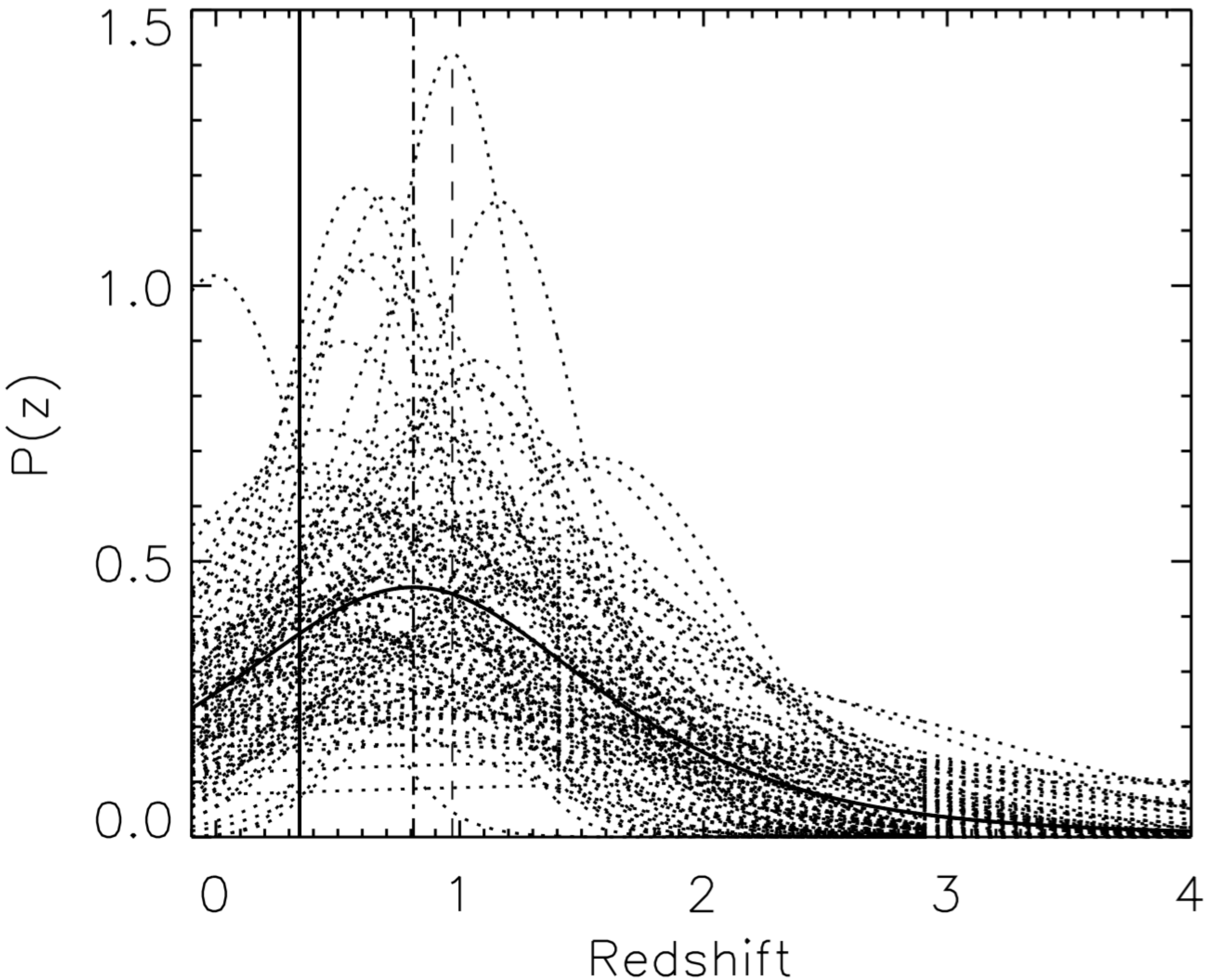


CSL Method	Filters	RMSE	NBIAS	MLL	$FR_{0.15}$	$FR_{0.05}$	Variance	Model Variance	Noise Variance
Normal	ugriz	0.0478	-0.0027	1.7714	99.1941	84.7809	0.0023	6.90E-06	0.0023
	ugrizYJHK	0.0438	-0.0024	1.8075	99.4523	87.7778	0.0018	6.80E-06	0.00184
	ugrizYJHK+size	0.042	-0.0024	**1.8792**	99.4366	89.1471	0.0018	7.20E-06	0.00180
Normalized	ugriz	0.0476	0.0001	1.7115	99.2958	84.8279	0.0015	6.68E-06	0.0015
	ugrizYJHK	0.0443	0.0001	1.7893	99.4366	87.8091	0.0013	6.99E-06	0.0013
	ugrizYJHK+size	**0.0418**	-0.0001	1.8188	**99.507**	**89.6401**	**0.0011**	**6.06E-06**	**0.0011**

Gomes, Jarvis, Almosallam & Roberts in prep

See Zahra's poster at the back of the room

GPz: non-Gaussian PDFs



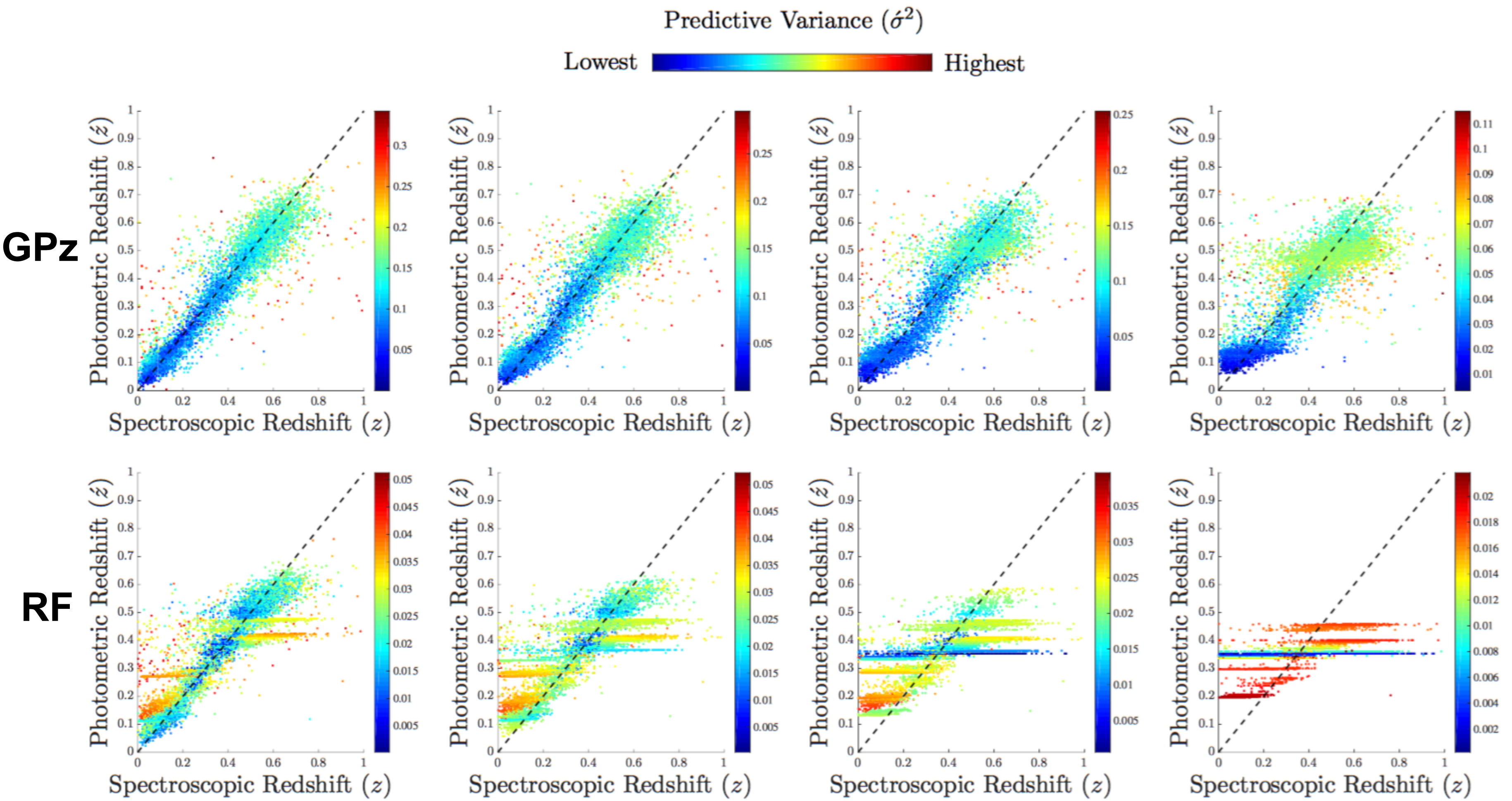
GPz: developments

- Add in more information – e.g. X-ray, radio, sizes etc (Zahra's poster)
- Use photometric noise without training with it
- Cost-sensitive learning based on colour distribution of "test" data
- Deal with missing data
  (not non-detections which we deal with anyway by using fluxes)

GPz: developments

- Add in more information – e.g. X-ray, radio, sizes etc (Zahra's poster)
- Use photometric noise without training with it
- Cost-sensitive learning based on colour distribution of "test" data
- Deal with missing data
  (not non-detections which we deal with anyway by using fluxes)
- Incorporate clustering within the training in fully consistent Bayesian framework
- Multi-modal Gaussian Process
  (would remove need for repeat training) and provide full PDF
- Regress to template fitting PDFs when GPz assesses it as the better option

GPz vs Random Forest



(b) 1 missing (c) 2 missing (d) 3 missing (e) 4 missing

Number of missing filters

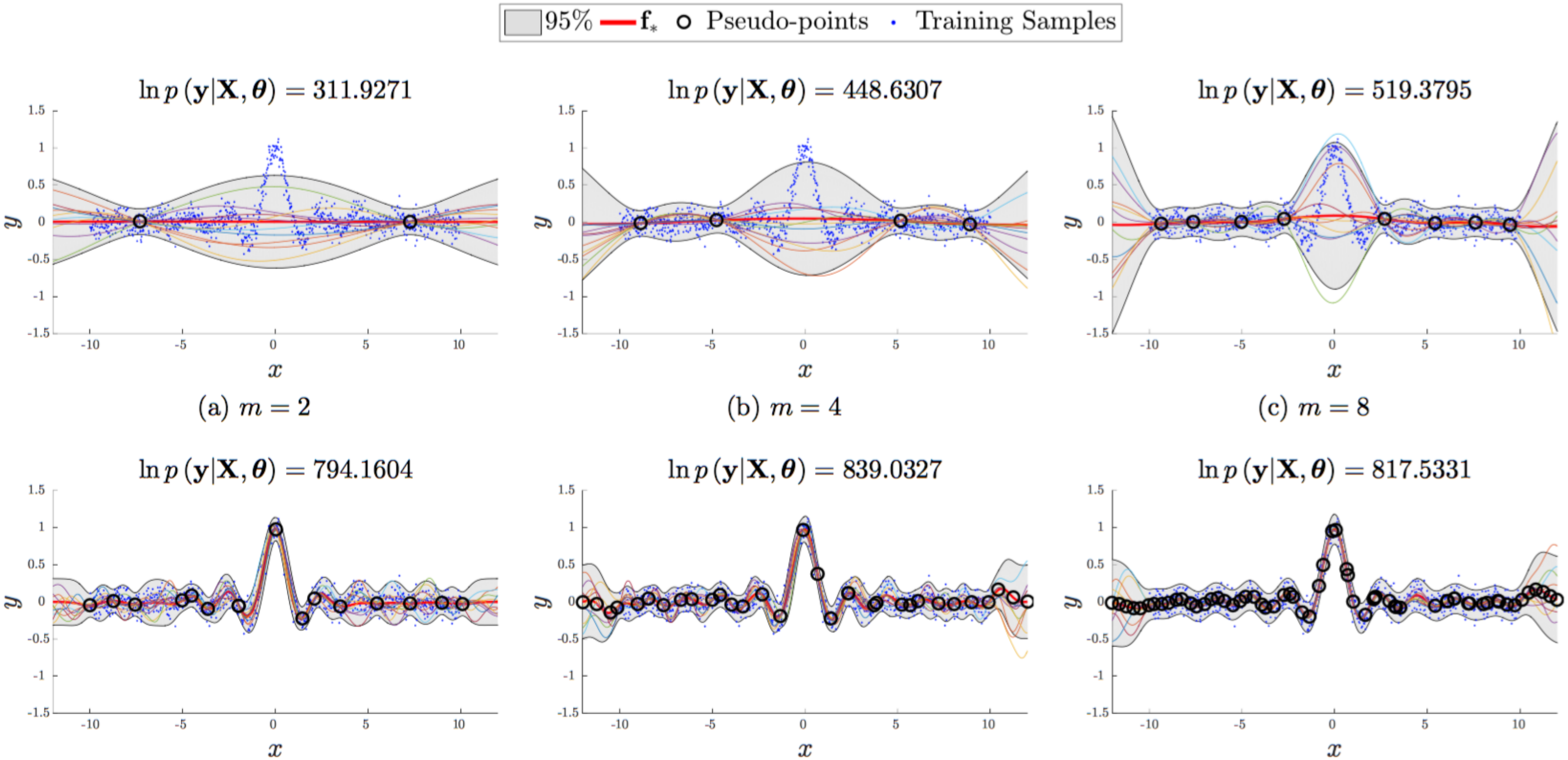
GPz: Summary

- Fully Bayesian ML framework
- Better point estimates and variance determination than ANNz2 (and possibly other NN-based codes – but need to check)
- Factor of ~100 faster than ANNz2 with PDFs
- Python and MATLAB versions available at

https://github.com/OxfordML/GPz

- We're happy to run with HSC, KIDs & DES data (if you can provide the training and test data sets)

Pseudo-points

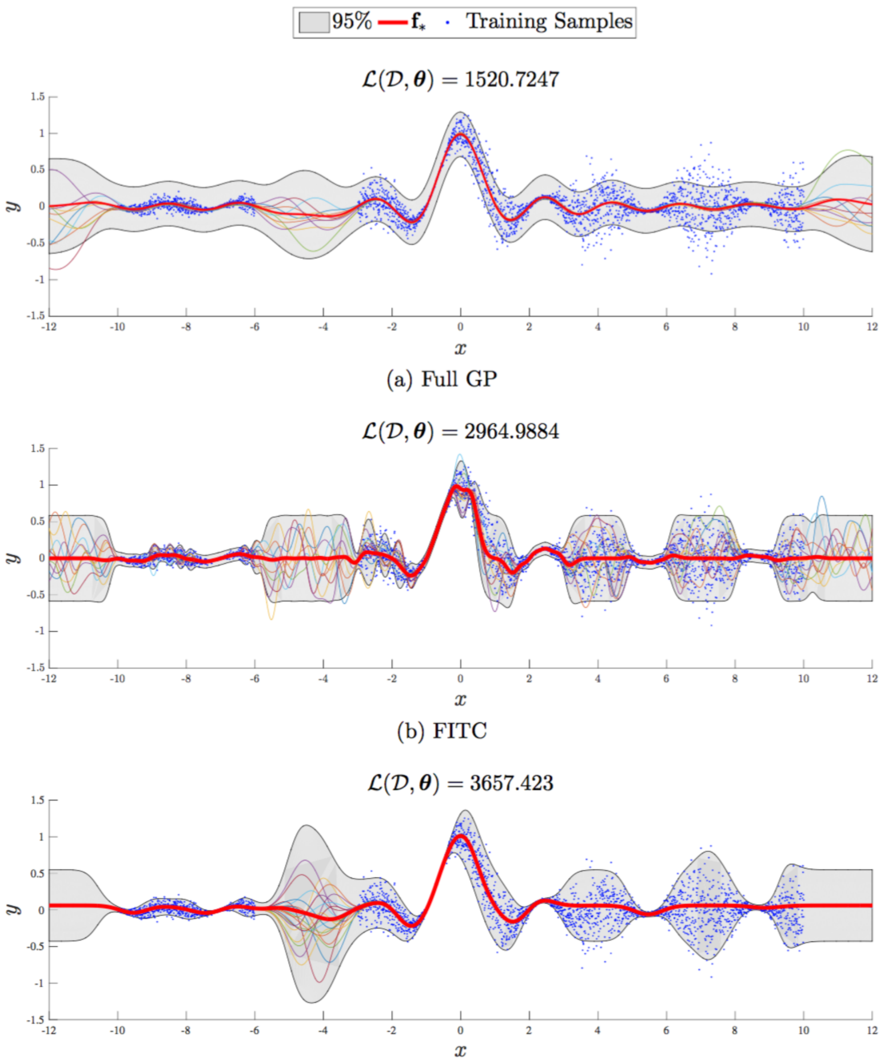


(a) $m = 2$ (b) $m = 4$ (c) $m = 8$

(d) $m = 16$ (e) $m = 32$ (f) $m = 64$

Figure 2.2: The effect of changing the number of pseudo-points (m), from (a) to (f) in multiples of 2, on FITC using an RBF kernel. The plots show the mean function (red), draws from the function distribution (Equation (2.41)), the 95% confidence range (grey), i.e. plus or minus two standard deviations from the mean, and the locations of the pseudo-points (black). The log marginal likelihood values are shown above each plot.

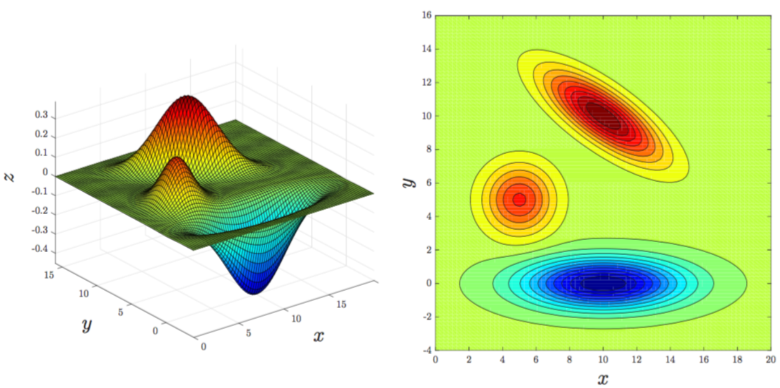
How to deal with noise



(a) Full GP

(b) FITC

(c) BFM



(a) Target function (b) Contour plot

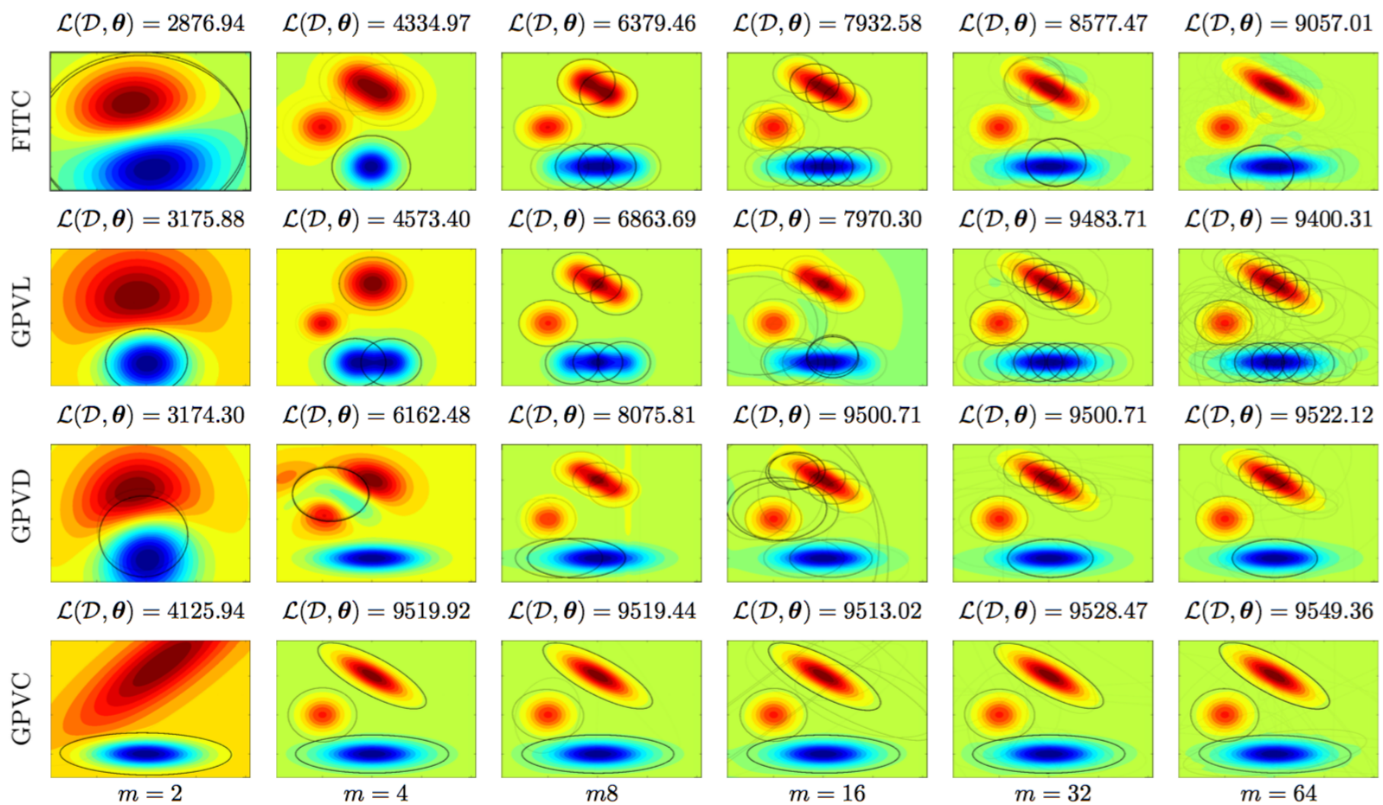


Figure 4.3: The results of training FITC, GPVL, GPVD and GPVC with different numbers of basis functions (m) on the same data collected using Equation (4.30). The ellipses represent the learned covariances of the RBFs, where the degree of transparency is proportional to the relevance (Equation (4.32)). The log marginal likelihoods are shown above each plot.