Stellar LAbel Machine (SLAM)

A forward model of stellar labels based on Support Vector Regression

章博，刘超，邓李才

Bo Zhang, Chao Liu, Li-Cai Deng

bozhang@nao.cas.cn

Code repository: https://github.com/hypergravity/slam (coming soon)

Current stellar label pipelines

- SSPP (Lee et al. 2008, →SDSS, a number of methods)
- LASP (Ulyss, Wu et al. 2011, ELODIE→LAMOST)

$$\mathrm{Obs}(\lambda) = P_n(\lambda) \times [\mathrm{TGM}(T_{\mathrm{eff}}, \log g, [\mathrm{Fe/H}], \lambda) \otimes G(v_{\mathrm{sys}}, \sigma)]$$

- LSP3 (interpolation, Xiang et al. 2015, →LAMOST)
- KPCA (Xiang et al. 2017, →LAMOST)

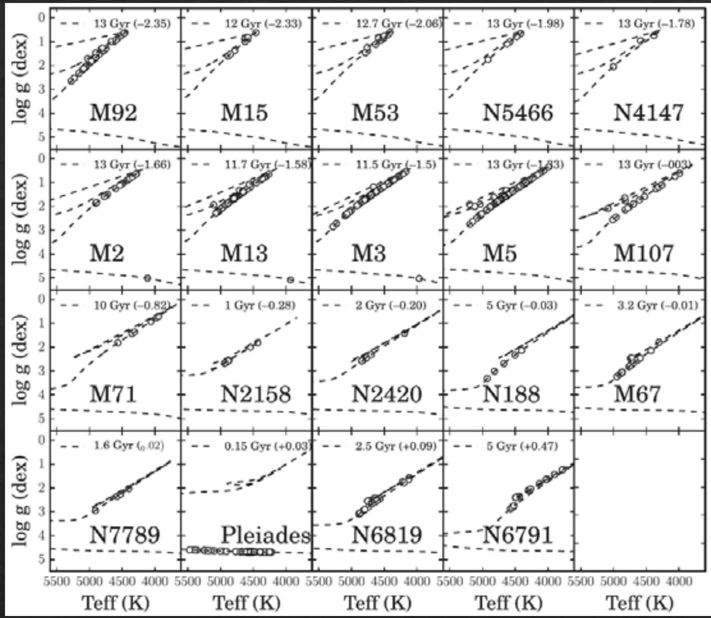
The concept of *Data-driven methods*

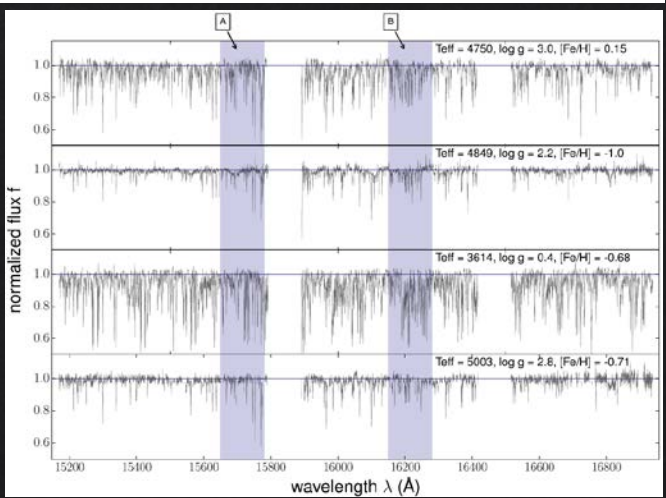
- TheCannon (Ness et al. 2015, APOGEE→APOGEE)
- TheCannon2 (Carsey et al. 2016, APOGEE→APOGEE)
- …

TheCannon: a Data-driven method to derive stellar labels

- A set of spectra with known stellar labels
- Pre-processing:
  - RV correction
  - Continuum normalization
  - Standardization for stellar labels
- Training:
  - Flux in each pixel is a function of stellar labels
- Prediction:
  - Predict stellar labels by maximizing likelihood

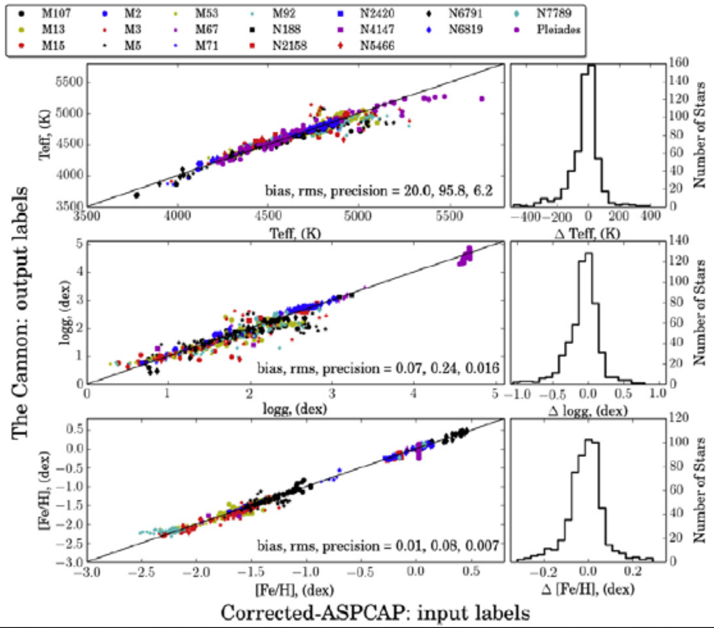
$$\ln p\left(f_{n\lambda} \,\middle|\, \boldsymbol{\theta}_\lambda^T, \boldsymbol{\ell}_n, s_\lambda^2\right) = -\frac{1}{2}\,\frac{\left[f_{n\lambda} - \boldsymbol{\theta}_\lambda^T \cdot \boldsymbol{\ell}_n\right]^2}{s_\lambda^2 + \sigma_{n\lambda}^2} - \frac{1}{2}\,\ln\left(s_\lambda^2 + \sigma_{n\lambda}^2\right).$$

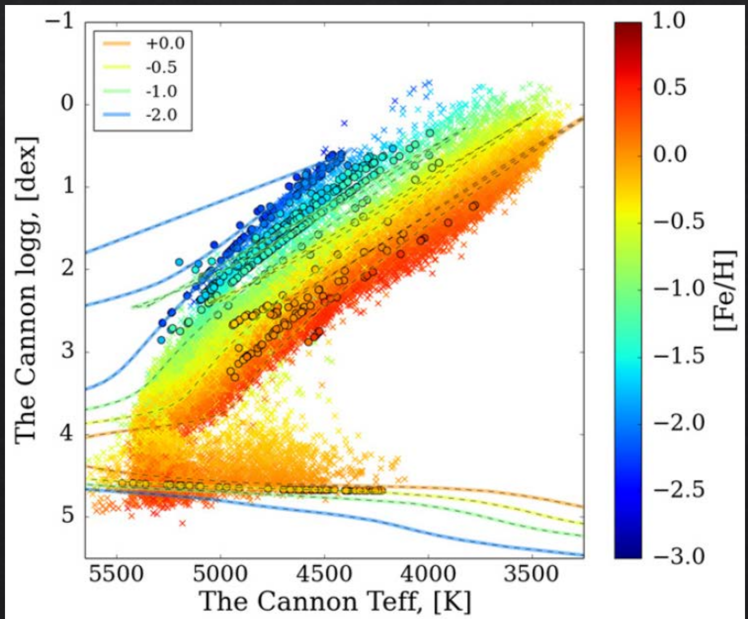


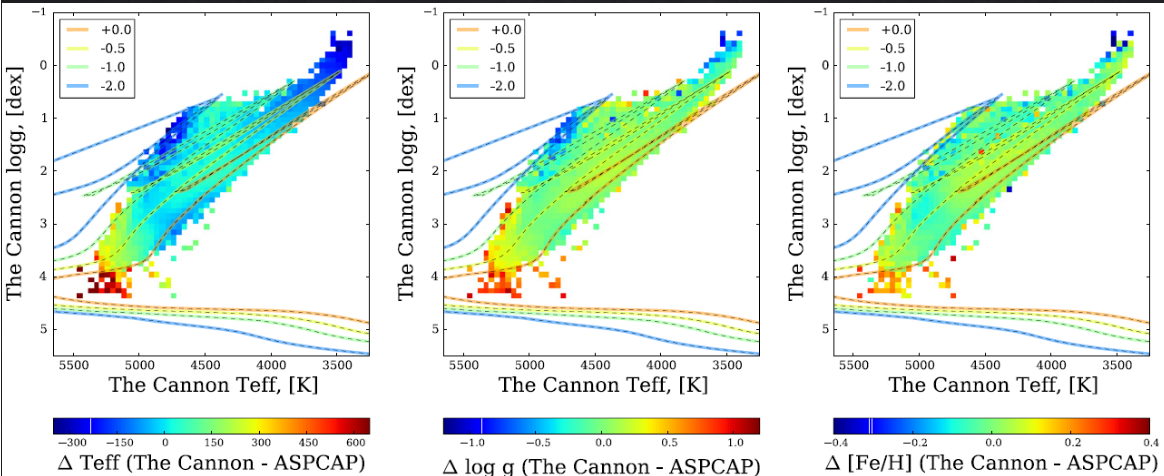


Performance: TheCannon

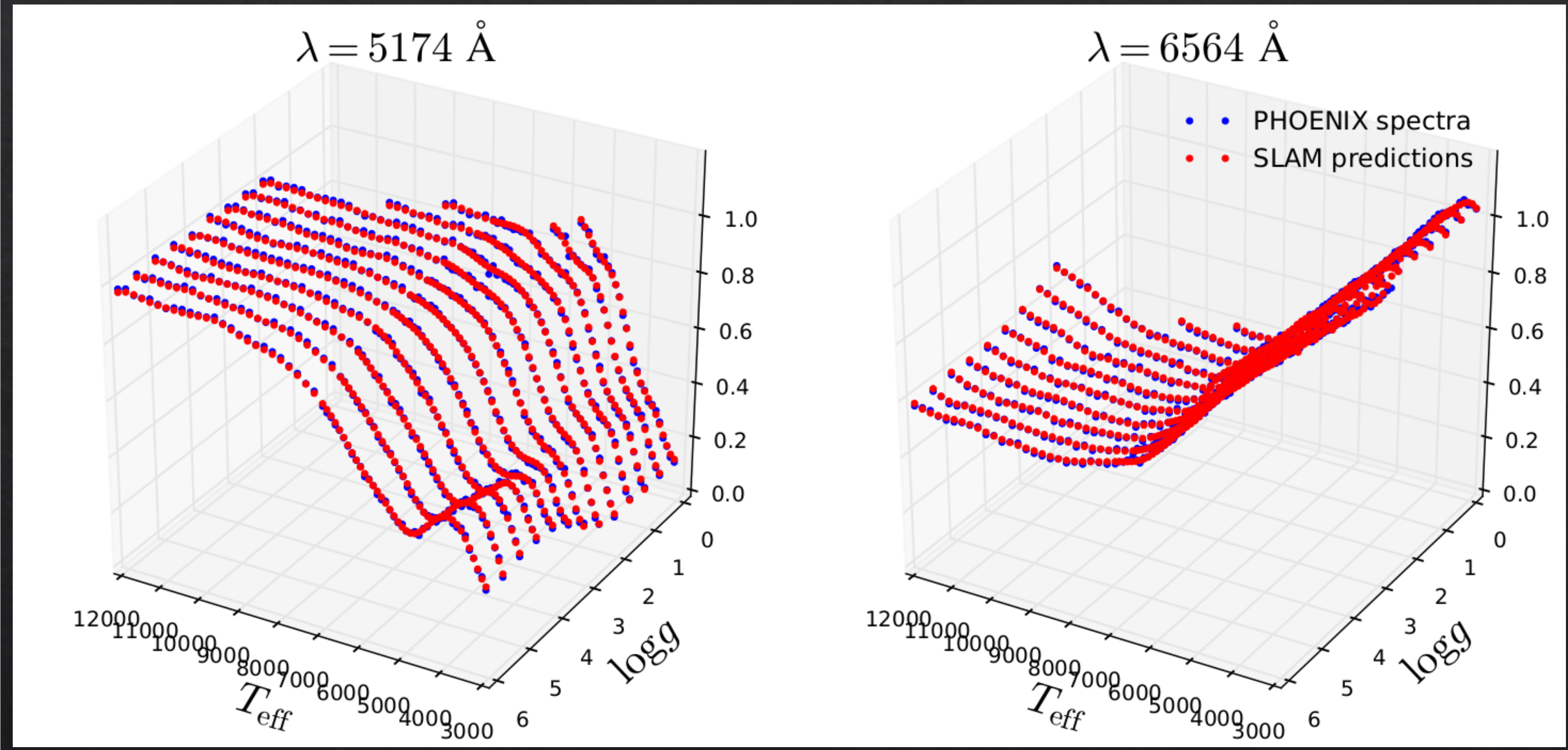
- δ[Teff, log g , [Fe/H]]
  - [73 K, 0.18 dex, 0.11 dex] for highest S/N
  - [100 K, 0.2 dex, 0.10 dex] for S/N~30–50
- with an S/N of ⩾25
  - Teff < 100 K
  - log g <0.2 dex
  - [Fe/H] <0.1 dex







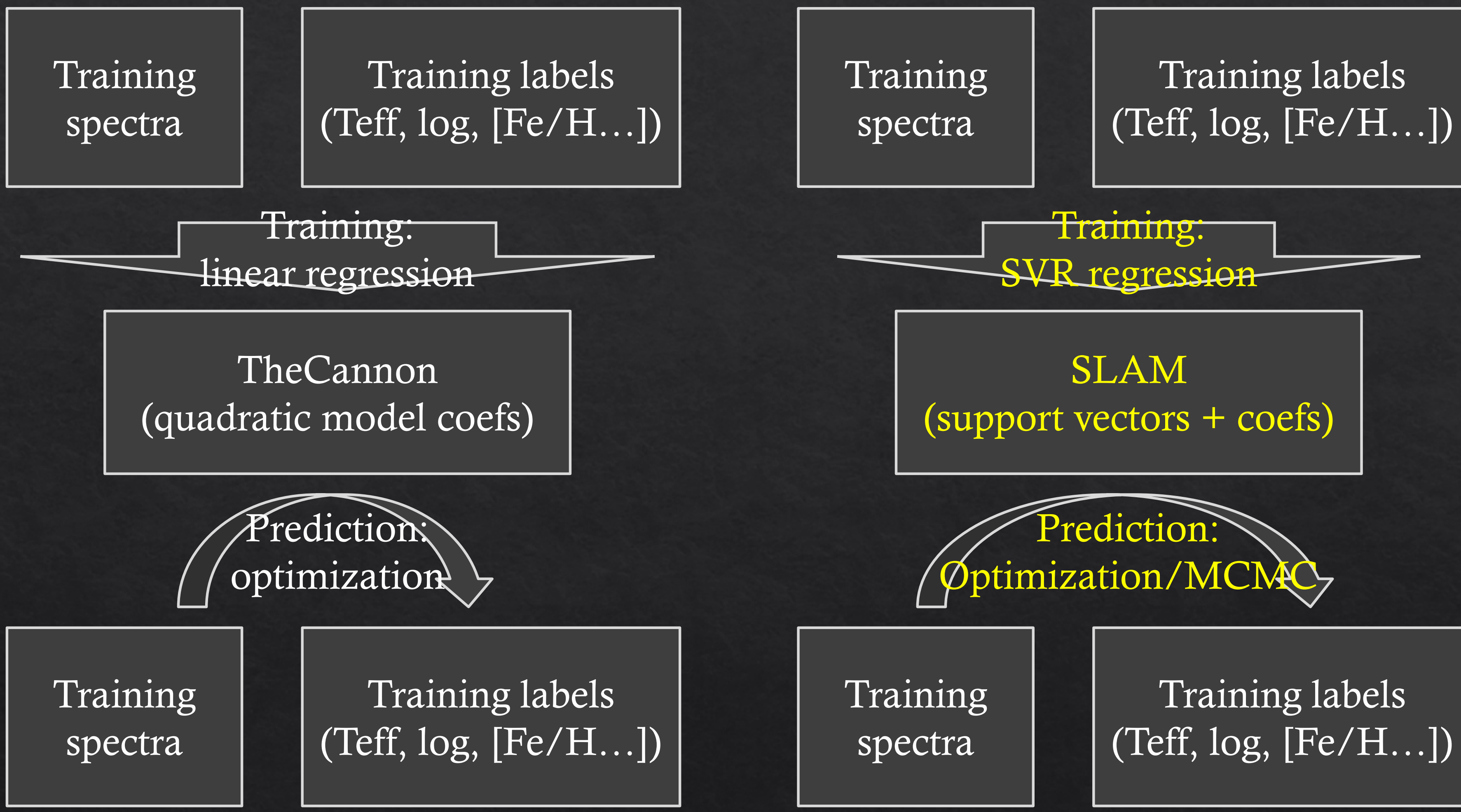
Quadratic model vs Non-parametric model



- Mgb & Ha center pixel trends ([Fe/H]=0.0)

TheCannon

SLAM



$$\ln p\left(\boldsymbol{\theta}|f_{\mathrm{obs}}\right) = -\frac{1}{2}\sum_{\lambda}\frac{\left|f_{\mathrm{obs}}\left(\lambda\right) - f_{\mathrm{SLAM}}\left(\lambda|\boldsymbol{\theta}\right)\right|^2}{\sigma_{\mathrm{obs}}\left(\lambda\right)^2} - \frac{1}{2}\sum_{\lambda}\ln\sigma_{\mathrm{obs}}\left(\lambda\right)^2$$

Support Vector Regression (SVR)

$$f(\boldsymbol{x}) = \sum_{i}^{\ell} \Big\langle \big(\alpha_i - \alpha_i^*\big)\Phi(\boldsymbol{x}_i),\, \Phi(\boldsymbol{x}) \Big\rangle + b, \qquad (1)$$

$$\text{minimize} \quad \frac{1}{2}\|\boldsymbol{w}\|^2 + C\sum_{i=1}^{\ell}\big(\xi_i + \xi_i^*\big)$$

$$\text{subject to} \quad \begin{cases} y_i - \langle \boldsymbol{w},\, \Phi(\boldsymbol{x}_i)\rangle - b \leqslant \epsilon + \xi_i \\ \langle \boldsymbol{w},\, \Phi(\boldsymbol{x}_i)\rangle + b - y_i \leqslant \epsilon + \xi_i^* \\ \xi_i,\, \xi_i^* \geqslant 0 \end{cases} \qquad (2)$$

- An ϵ-robust non-parametric regression method
- Kernel
  - RBF, linear, …
- Tuning parameters
  - C, ϵ, (gamma)
- To overcome overfitting
  - Cross-Validation + Grid/Random search

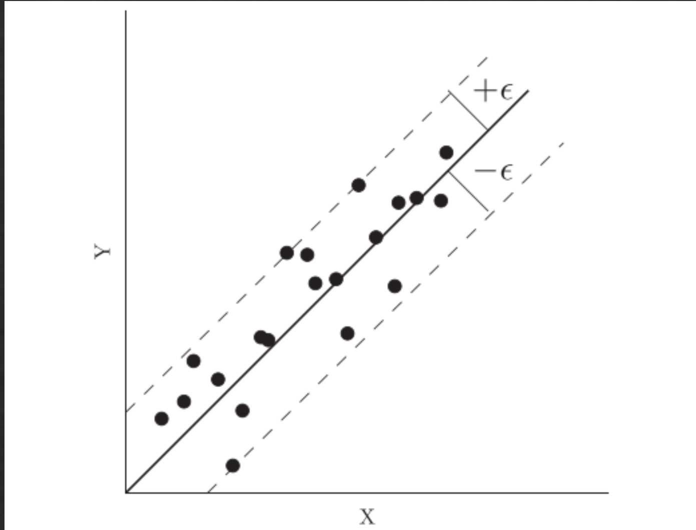
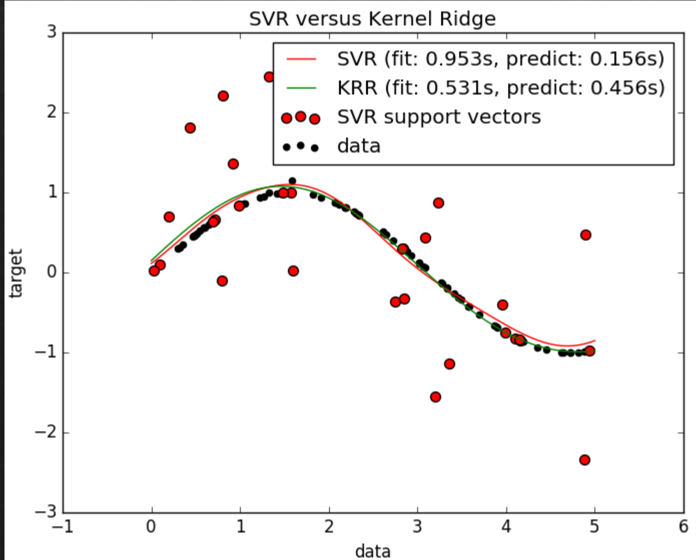
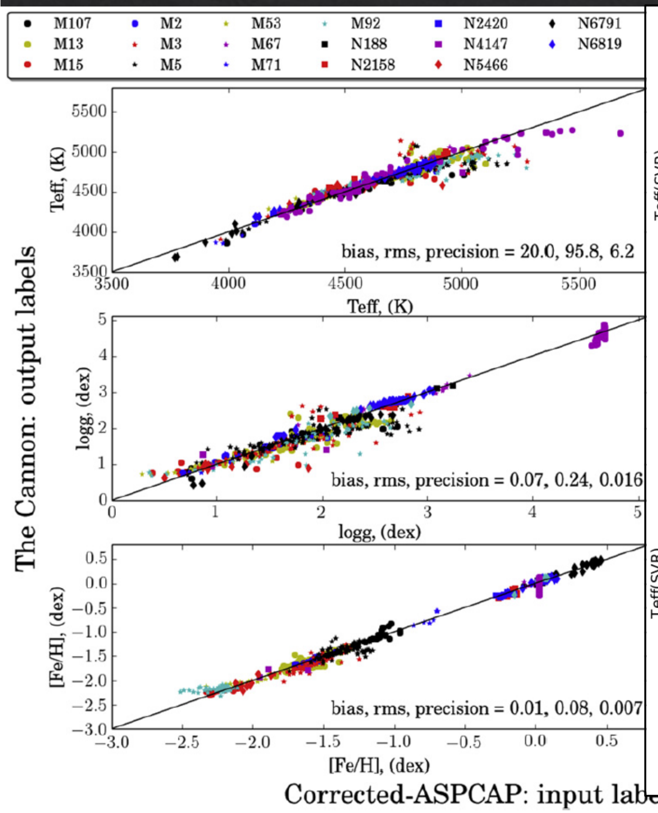


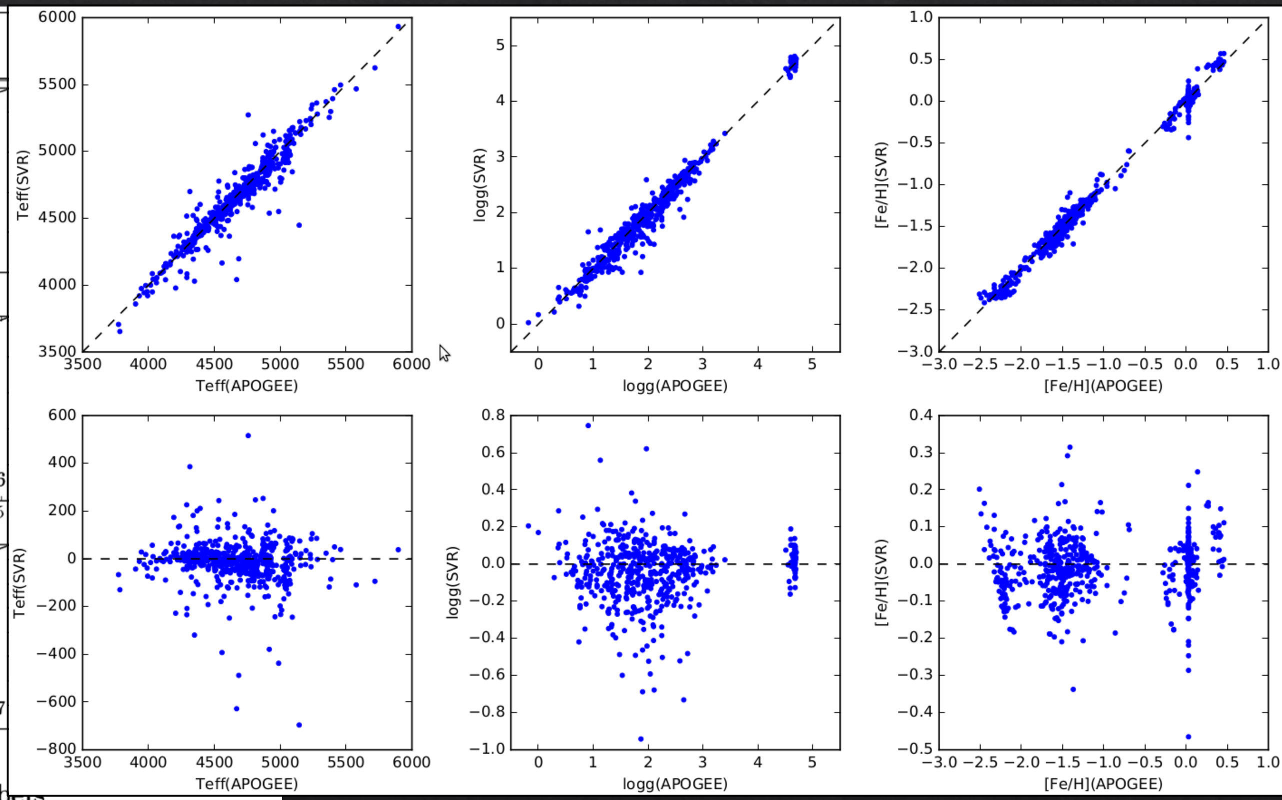
Figure 1. Demonstration of how the SVR model works in the linear case. The black dots are the mock data with arbitrary scales in *x*- and *y*-axes. The solid line stands for the SVR model, and the dashed lines show the tolerance of ϵ on each side of the linear model.



Performance of SLAM: APOGEE

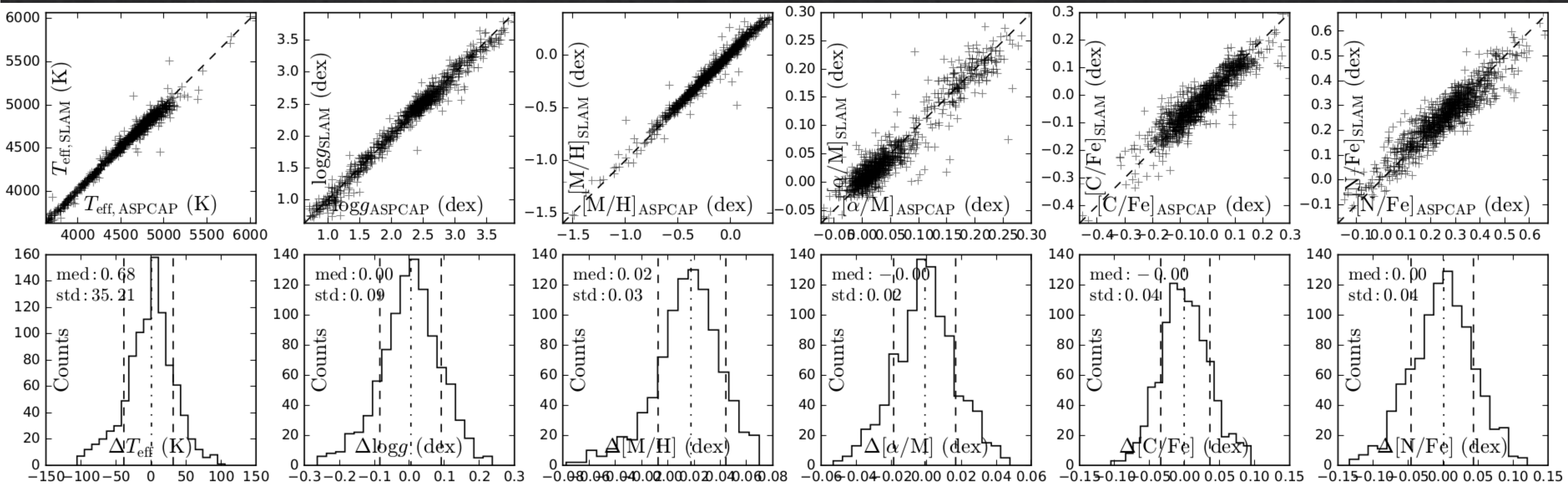
- GridSearch: search for the optimal set of hyper-parameters for each pixel (least MSE)
  - C = [2, 8], epsilon = [0.05, 0.10]
- Training error comparable with TheCannon (and maybe a little superior)





Performance of SLAM: APOGEE

- Relations between SLAM & ASPCAP –derived stellar labels



ΔTeff=35K ΔlogG=0.09 Δ[M/H]=0.03 Δ[a/M]=0.02 Δ[C/Fe]=0.04 Δ[N/Fe]=0.04

Performance of SLAM: APOGEE

- Relations between SLAM & ASPCAP – derived stellar labels

δTeff

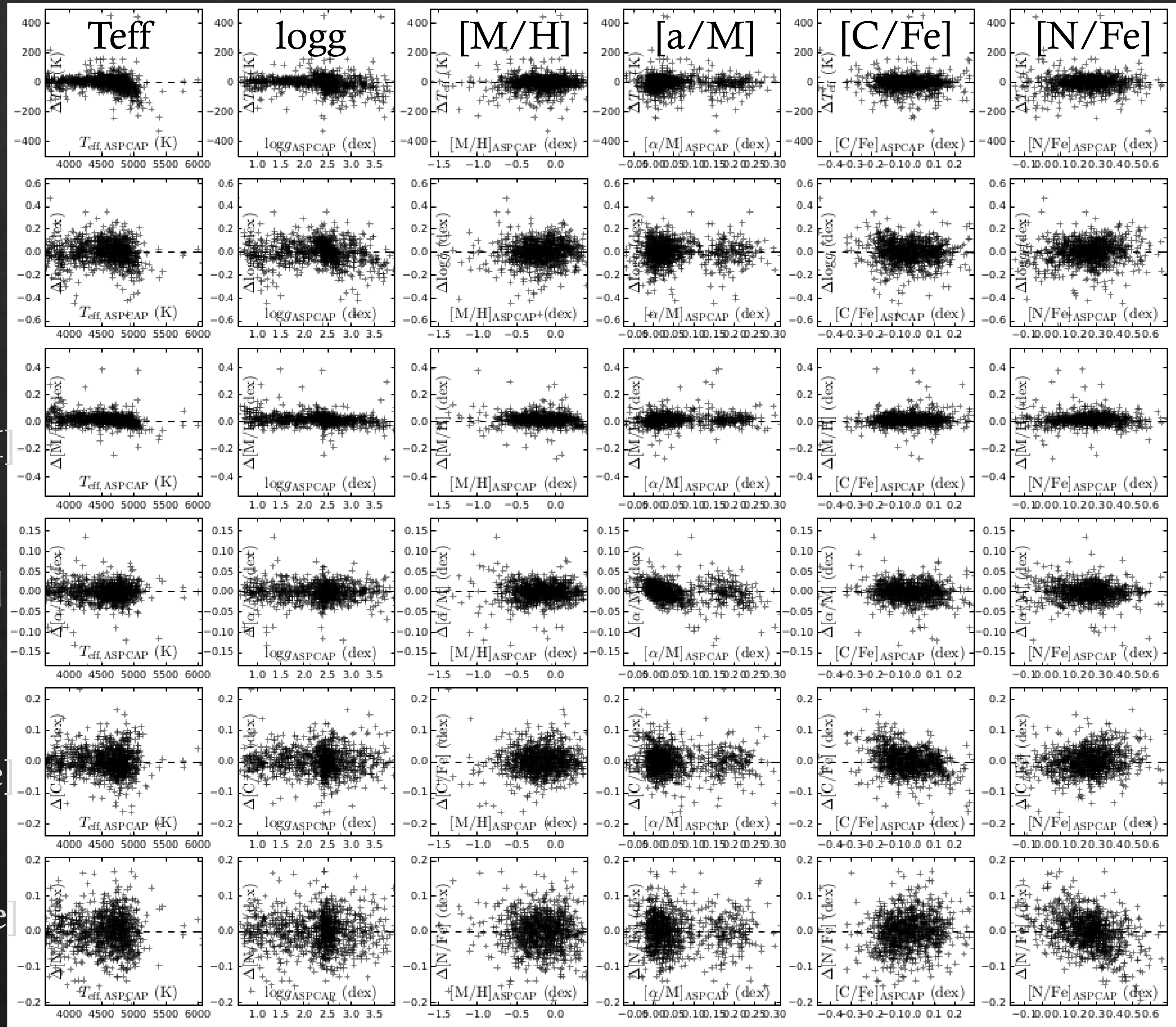
Δlogg

δ[M/H]

δ[a/M]

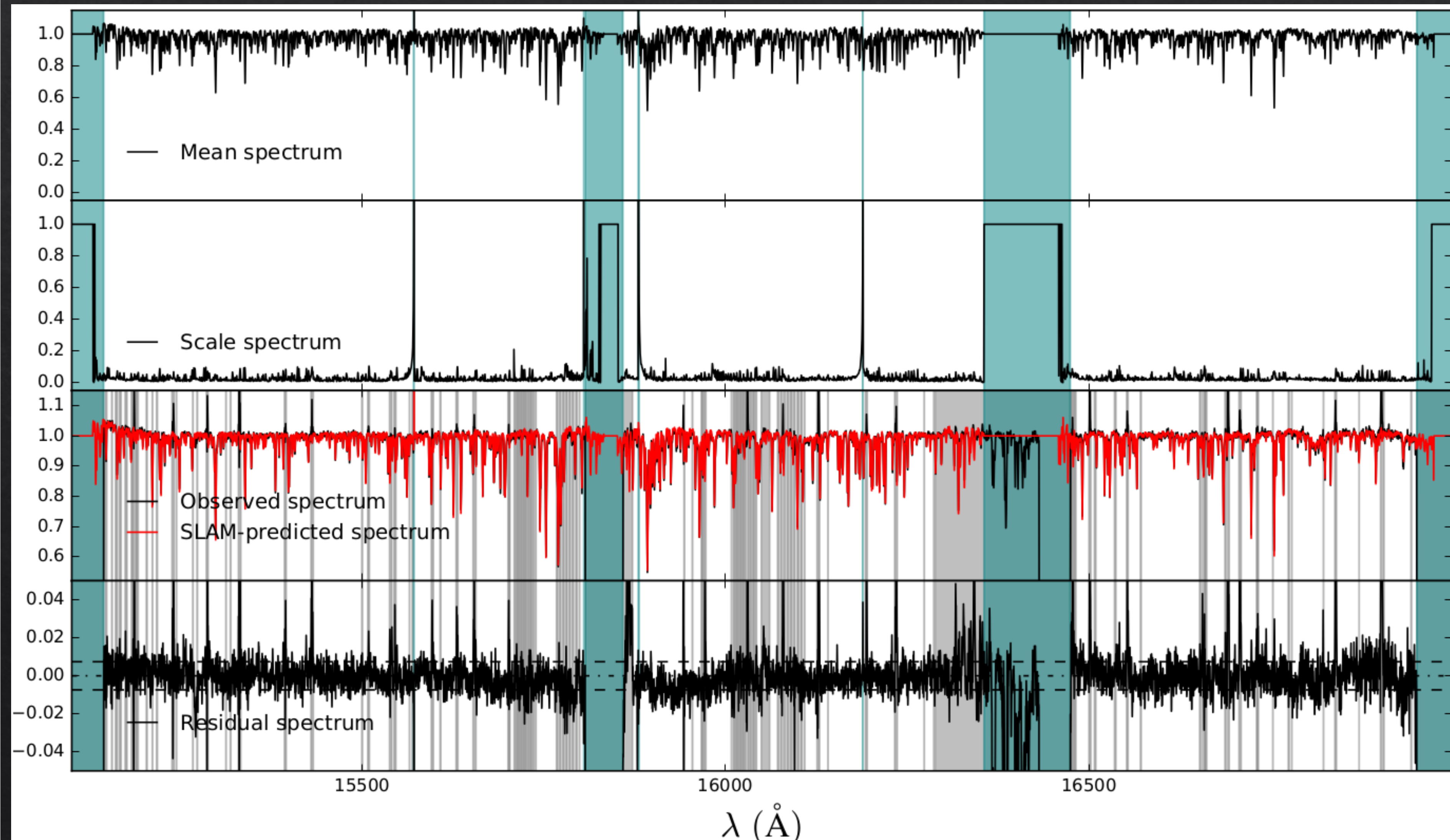
δ[C/Fe]

δ[N/Fe]



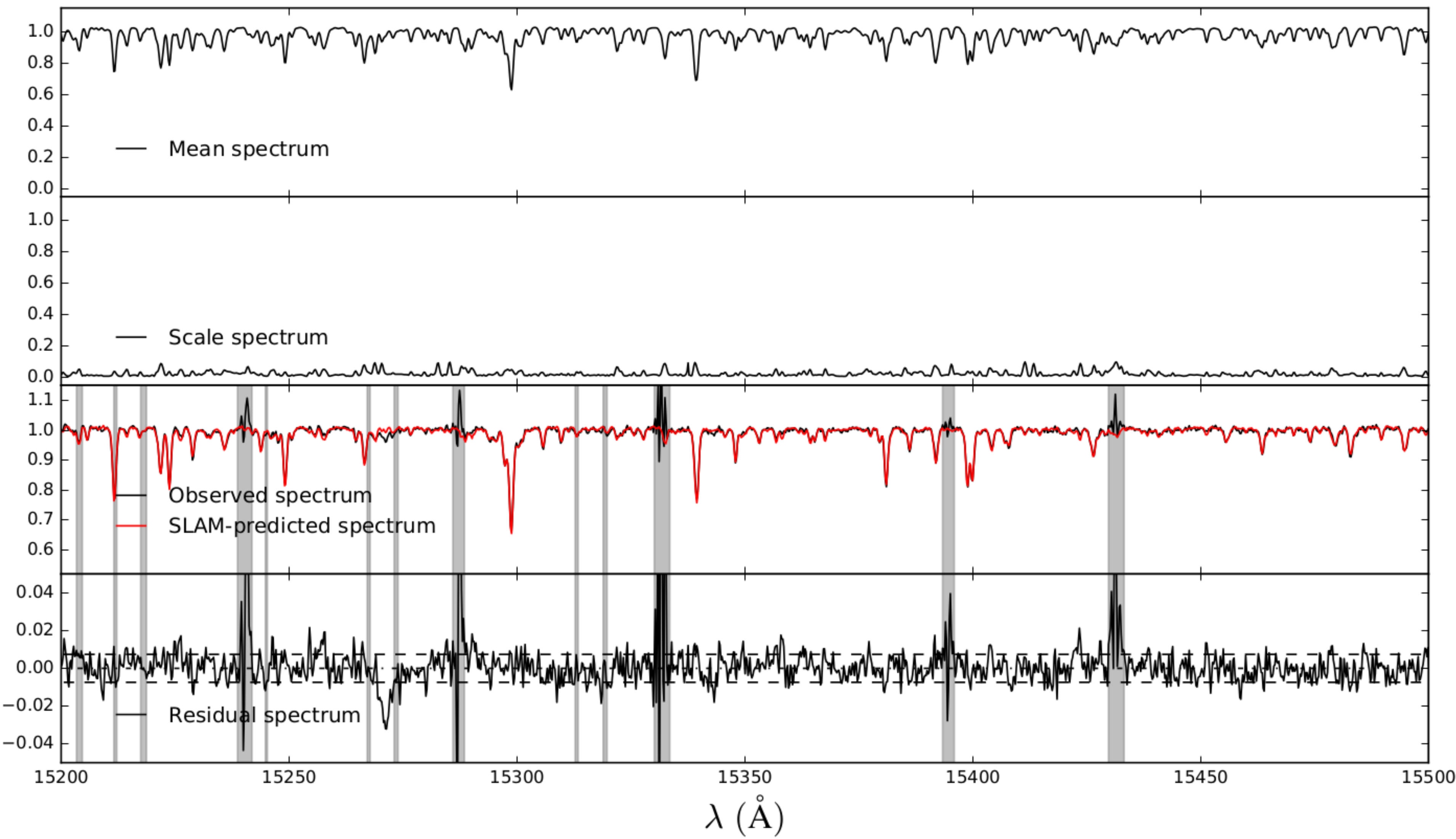
Performance of SLAM: APOGEE

- Example of Predicted spectra (APOGEE)



Performance of SLAM: APOGEE

- Example of Predicted spectra (APOGEE)



Performance of SLAM: APOGEE

- Relations between SLAM & ASPCAP –derived stellar labels

Table 1: Statistics of *SLAM*-derived stellar labels of test sample with different SNR.

SNR	parameter	$T_{\rm eff}$ (K)	$\log g$ (dex)	[M/H] (dex)	[α/M] (dex)	[C/Fe] (dex)	[N/Fe] (dex)
200 < SNR < 300	median	0.685	0.004	0.018	-0.001	-0.001	0.001
	STD	54.296	0.101	0.045	0.023	0.041	0.050
	MAD	21.871	0.056	0.023	0.011	0.023	0.030
100 < SNR < 200	median	-14.141	0.017	0.013	-0.000	-0.007	0.005
	STD	64.307	0.130	0.056	0.026	0.046	0.066
	MAD	31.335	0.071	0.024	0.014	0.025	0.032
60 < SNR < 100	median	-22.499	-0.009	0.008	-0.002	-0.017	0.007
	STD	88.372	0.187	0.067	0.031	0.064	0.076
	MAD	42.975	0.098	0.030	0.019	0.035	0.041
20 < SNR < 60	median	-31.845	-0.063	-0.004	-0.002	-0.017	0.001
	STD	91.737	0.210	0.068	0.033	0.067	0.087
	MAD	52.962	0.115	0.034	0.020	0.036	0.047

The performance of TheCannon: Ness et al. (2015):
δ[Teff, log g , [Fe/H]]
[73 K, 0.18 dex, 0.11 dex] for highest S/N
[100 K, 0.2 dex, 0.10 dex] for S/N~30–50

TODOs

- Training on LAMOST spectra + LASP/improved stellar labels to provide self-consistent/better stellar labels for LAMOST spectra
- Transfer APOGEE stellar labels to LAMOST (RG stars)

- Provide stellar label estimations for LAMOST II (more element abundances)

Thank you