Probabilistic Dalek - Emulator framework with probabilistic prediction for supernova tomography

Wolfgang Kerzendorf [*1 2] Nutan Chen [*3]
Patrick van der Smagt [3 4]

Abstract

Supernova spectral time series can be used to re-construct a spatially resolved model of the explosion known as supernova tomography. In addition to an observed spectral time series, a supernova tomography requires a radiative transfer model to perform the inverse problem with uncertainty quantification for a reconstruction. The smallest parametrizations of supernova tomography models are roughly a dozen parameters with a realistic one requiring more than 100. Realistic radiative transfer models require tens of CPU minutes for a single evaluation making the problem computationally intractable with traditional means requiring millions of MCMC samples for such a problem. A new method for accelerating simulations known as surrogate models or emulators using machine learning techniques offers to provides a solution for such problems and a way to understand progenitors/explosions from spectral time series. There exist emulators for the TARDIS supernova radiative transfer code but they only perform well on simplistic low-dimensional models (roughly a dozen parameters) with a small number of applications for knowledge gain in the supernova field. In this work, we present a new emulator for the radiative transfer code TARDIS that not only outperforms existing emulators but also provides uncertainties in its prediction. It presents the foundation for a future active learning based machinery that will be able to emulate very high dimensional spaces of hundreds of parameters crucial for unraveling urgent questions in supernova and related fields.

1. Introduction

Regular stars only allow for direct measurements of the properties of their surface and views into the interior are not directly possible. The dynamical nature of supernovae, however, encodes spatially resolved information about their interior in spectral time series (known as supernova tomography). Supernova tomography uses the recession of a surface of last scattering (photosphere) deeper into the envelope to perform parameter inference on shells of ejecta with theoretical radiative transfer simulations. Such data-driven physically motivated tomography models are directly comparable to explosion simulations and have the power to unlock the progenitor and explosion mechanisms for many different classes of exploding objects.

However, the parameter space for such models has at a minimum ≈ 12 dimensions for very simple parameterizations of the problem and coupled with evaluations times of tens of minutes per model with modern radiative transfer code results in a computationally intractable model (see e.g. Kerzendorf et al., 2021). Credible comparisons of supernova tomographies with explosion scenarios do not only require solving the inverse problem but also demand uncertainty quantification exacerbating the computational requirements.

Simulation based inference has addressed the uncertainty quantification by using emulators (also known as surrogate models) which are machine learning constructs provide approximations that are orders of magnitude faster to complex simulations which then allows their use in standard MCMC samplers (Cranmer et al., 2020). Vogl, C. et al. (2020); Kerzendorf et al. (2021); O'Brien et al. (2021) have developed emulators to accelerate the TARDIS (Kerzendorf & Sim, 2014; Vogl et al., 2019; Kerzendorf et al., 2022) radiative transfer code for problems up to 14 parameters and applied them to supernova tomography.

A full tomography and with it the understanding of explosion and progenitor mechanisms requires roughly an order of magnitude more parameters. Such large dimensionality is not feasible for the current generation of emulators. High-dimensional emulators will require an active learning component to ensure training sam-

[*]Equal contribution [1]Department of Physics and Astronomy, Michigan State University, East Lansing, MI 48824, USA [2]Department of Computational Mathematics, Science, and Engineering, Michigan State University, East Lansing, MI 48824, USA [3]Machine Learning Research Lab, Volkswagen AG, Munich, Germany [4]Faculty of Informatics, Eötvös Loránd University, Budapest, Hungary. Correspondence to: Wolfgang Kerzendorf <wkerzend@msu.edu>.

ICML 2022 Workshop on Machine Learning for Astrophysics, Baltimore, Maryland, USA, 2022. Copyright 2022 by the author(s).

ples are only produced in parts of the parameter space where they decrease the uncertainty of the emulator.

Numerous publications extend neural networks to quantify the prediction by uncertainties. Recently popular approaches of Bayesian methods include Monte Carlo (MC) dropout (Gal & Ghahramani, 2016) and weight uncertainty (Blundell et al., 2015). In addition, there are non-Bayesian approaches, for instance, post-hoc calibration by temperature scaling of a validation dataset (Guo et al., 2017), and deep ensembles (Lakshminarayanan et al., 2017). Amongst the classical methods, deep ensembles (Lakshminarayanan et al., 2017) generally perform best in uncertainty estimation (Ovadia et al., 2019). Methods such as weight uncertainty, MC dropout, and deep ensembles require multiple passes to obtain the uncertainty. For computational efficiency, deterministic methods in a single forward pass (Van Amersfoort et al., 2020; Liu et al., 2020) are presented.

Since our model is a relatively small network which does not take a large amount of computation, we choose the best uncertainty prediction, viz. deep ensembles. In our case, the prediction uncertainties are not only driven by the sampling sparseness of the parameter space but also by the Monte Carlo nature of the TARDIS radiative transfer code.

In Section 2, we describe the setup of our neural network model. We show various statistics about our model and comparison with TARDIS and its uncertainty in Section 3. We conclude and give an outlook over future work in Section 4.

2. Neural network model

The mapping from the parameters to the spectra is approbated by a feedforward neural network. For uncertainty estimation, we select proper scoring rules, and then train the ensemble as proposed in (Lakshminarayanan et al., 2017). An adversarial training (AT) is an option that probably improves the uncertainty measurement.

2.1. Single model for regression

We use the training set from (Kerzendorf et al., 2021) and thus have the ‘parameters’ as the inputs $\mathbf{x} \in \mathbb{R}^{12}$ and the ‘spectra’ as the outputs $\mathbf{y} \in \mathbb{R}^{500}$ of the neural networks. The dataset is split into 90 000 training data, 18 000 validation data and 18 000 test data. The data is preprocessed by a logarithmic scale and then normalised by removing the mean and scaling to unit variance (Pedregosa et al., 2011).

The data sets are comfortably large. That may be the reason that the neural networks are not very sensitive to hyperparameter variations, as we discovered in a very broad search on a compute cluster. The use of dropout did not change much, nor was layer normalisation essential. The best architectures were those with between 3 and 5 hidden layers of 200 to 400 softplus hidden units, training with batch sizes between 100 and 500 samples. All networks are trained with Adam.

van der Smagt & Hirzinger (1998); He et al. (2016) propose a residual architecture for deep layers to avoid degradation problems. This residual architecture aggregates the output from the previous layer and the current layer as the input to the next layer. In our network, we use concatenation for faster convergence as opposed to addition (as suggested in He et al., 2016).

We obtain the uncertainty for each model by outputting the mean μ, and the standard deviation (STD) σ, from the final hidden layer.

Scoring rules The quality of predictive uncertainty can be measured by scoring rules. The maximised likelihood $\log p_\theta(\mathbf{y}|\mathbf{x})$ is a proper scoring rule (Lakshminarayanan et al., 2017; Gneiting & Raftery, 2007). We use a maximum-a-posteriori (MAP) that is a negative log likelihood (NLL) with the weight regularisation $\log p(\theta)$. A standard NLL uses no prior knowledge about the expected distribution of the model weights θ and in our case leads to overfit. If the weights are standard distribution, $\log p(\theta)$ is approximate to L_2 norm of the weights. In our implementation, we set a hyper-parameter β as the coefficient of the regularisation, and the loss is minimised as:

$$\begin{aligned}\mathcal{L}(\mathbf{x}) &= -\log p_\theta(\mathbf{y}|\mathbf{x}) - \beta \log p(\theta) \\ &\propto \frac{\log \sigma^2_\theta(\mathbf{x})}{2} + \frac{\left(\mathbf{y} - \mu_\theta(\mathbf{x})\right)^2}{2\sigma^2_\theta(\mathbf{x})} + \beta \|\theta\|_2 + \text{constant}.\end{aligned} \tag{1}$$

Adversarial training An adversarial training (AT Szegedy et al., 2014; Goodfellow et al., 2015) is able to smooth predictive distributions and is possible to improve the uncertainty prediction (Lakshminarayanan et al., 2017). Adversarial examples are similar to the training samples, but are misclassified by NNs. The examples are generated by $\mathbf{x}' = \mathbf{x} + \epsilon\ \mathrm{sign}(\nabla_\mathbf{x} \mathcal{L}(\theta, \mathbf{x}, \mathbf{y}))$, where the perturbation is bounded in ϵ.

We then rewrite the loss:

$$\mathcal{L}_{AT}(\mathbf{x}) = \alpha \mathcal{L}(\mathbf{x}) + (1-\alpha)\mathcal{L}(\mathbf{x}'). \tag{2}$$

Table 1. Model architecture.

name	hyper-parameters
input dimension	12
output dimension	500
number of hidden layers	5
activation of hidden layers	Softplus
connection of hidden layers	residual with concatenation
hidden units	400
Activation of μ final layer	Linear
Activation of σ final layer	Softplus
β	5e-4
α	0.9
ϵ	5e-4
batch size	500
learning rate	2e-4
parameters initialisation	xavier normal

We select a good model by searching for the hyper-parameters based on the lowest loss value of the validation dataset. Searching in small ranges around the best hyper-parameters of (Kerzendorf et al., 2021), we obtain Table 1 using Polyaxon 0.5.6 (https://polyaxon.com) on a cluster with multiple NVIDIA Tesla V100 GPUs. The code is implemented in Pytorch 1.7.1 (Paszke et al., 2019).

2.2. Ensembles

To obtain multiple models from the best architecture, we have different weight initialisation and the order of the batch data selection, for each model. After training M models independently and in parallel, the prediction is measured using a uniformly-weighted mixture of Gaussian distributions

$$p(\mathbf{y}|\mathbf{x}) = M^{-1} \sum_{m=1}^{M} p_{\theta_m}(\mathbf{y}|\mathbf{x}, \theta_m). \tag{3}$$

We approximate the ensemble prediction as a Gaussian with a mean and variance equal to, respectively, the mean and variance of the mixture

$$\begin{aligned} \mu_*(\mathbf{x}) =& M^{-1} \sum_m \mu_{\theta_m}(\mathbf{x}), \\ \sigma_*^2(\mathbf{x}) =& M^{-1} \sum_m \big(\sigma_{\theta_m}^2(\mathbf{x}) + \mu_{\theta_m}^2(\mathbf{x})\big) - \mu_*^2(\mathbf{x}). \end{aligned} \tag{4}$$

3. Results

We evaluate the approach on the spectra simulation. We take the empirical variance as the baseline, which is commonly used in practice. Ensembles of NNs approximate the uncertainty from the empirical variance of multiple predictions. Usually, it uses mean square error (MSE) loss for the training. To simplify the comparison, we use the same models as the deep ensembles. We also compare the deep ensembles with a vanilla uncertainty estimation – measuring the uncertainty by the STD of a single model. Since the vanilla approach is the deep ensembles with $M = 1$, we do not separately demonstrate the results. Additionally, we evaluate how the optional term, the AT, affects the ensembles.

We use the mean and max of fractional error metrics as in (Vogl, C. et al., 2020; Kerzendorf et al., 2021) to quantify the results:

$$\begin{aligned} \text{MeanFE} =& \frac{1}{N} \sum_{i=0}^{N} \frac{|\mathbf{y}_i^{\text{emu}} - \mathbf{y}_i^{\text{test}}|}{\mathbf{y}_i^{\text{test}}}, \\ \text{MaxFE} =& \max_{i=0}^{N} \frac{|\mathbf{y}_i^{\text{emu}} - \mathbf{y}_i^{\text{test}}|}{\mathbf{y}_i^{\text{test}}}, \end{aligned}$$

where N is the dimension of spectra, and $\mathbf{y}_i$ represents the flux at the i-th dimension.

3.1. Spectra prediction

Figure 1 shows the accuracy of the prediction. In general, the ensembles with ATs outperforms the approaches without ATs. With the number of models in the ensemble more than six, the accuracy is not significantly improved.

Based on the above evaluation, the AT hardly improves the uncertainty prediction, but it improves the accuracy, especially when the number of models is small. The deep ensembles outperform the empirical and vanilla approaches since the latter is easily overconfident. Taking into the accuracy, computation efficiency, and the uncertainty prediction into consideration, the best number of the models for the deep ensembles is suggested to be six. Figure 2 illustrates two examples of the prediction of the testing dataset using deep ensembles with six models and thus in the following we will use six models for the ensemble in our further tests.

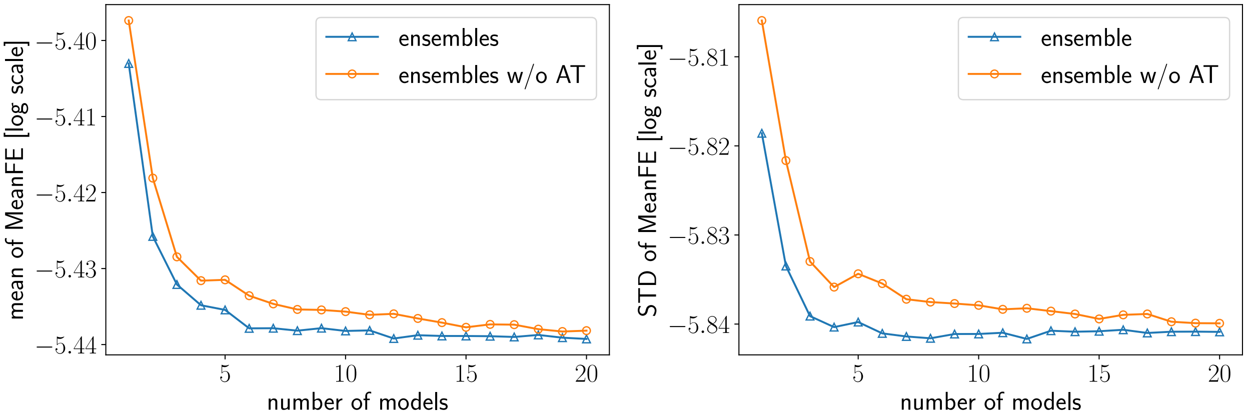

Figure 1. Accuracy. The horizontal-axis shows the number of NN in the ensembles. The mean and the STD of 18 000 testing samples are computed.

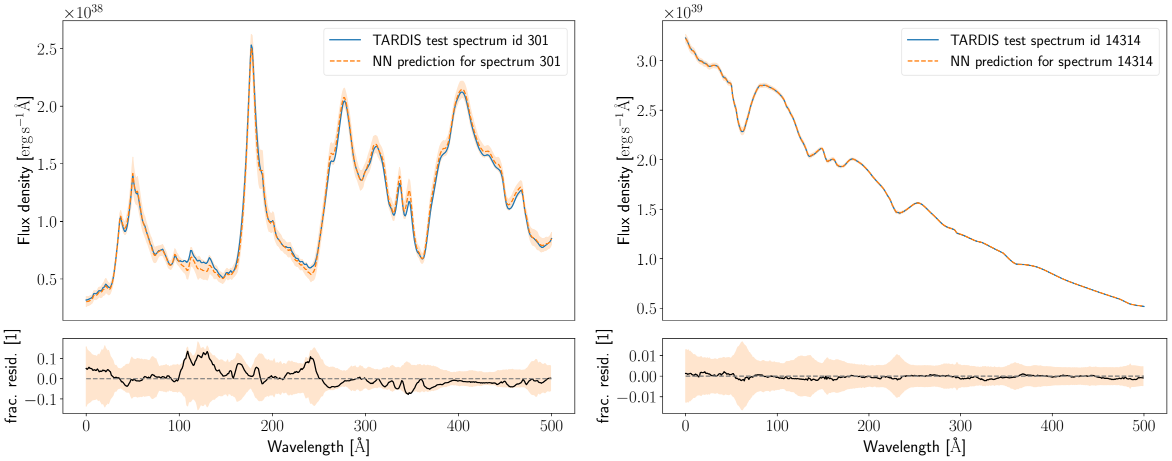

Figure 2. Examples of the uncertainty prediction. Highest and the lowest MaxFE from the test set predictions. The shaded areas denote 99.9 % confidence interval. In the lower figures, the black lines and dashed grey lines represent the residual and zero values, respectively.

3.2. Uncertainty prediction

The Dalek emulator has to contend with two types of uncertainties: 1) the prediction uncertainty stemming from a sparseness of sampling 2) the intrinsic uncertainty of the TARDIS simulator arising from the Monte Carlo radiative transfer. We have estimated the uncertainty given by TARDIS for the test-set spectrum with the highest max σ/μ-ratio (highest relative uncertainty; id=12625) and reran TARDIS (version hash ad91bef1a) with 100 different seeds to estimate the uncertainty arising from the Monte Carlo method. In Figure 3, we show that the network predicts a larger uncertainty that likely includes additional prediction uncertainty and mostly completely envelops the uncertainty given by TARDIS. Further tests are needed with a larger number of samples to explore the consistency of this result.

4. Conclusions

We present a probabilistic neural network model for the TARDIS supernova radiative transfer code. The emulator model is based on the deep ensemble approach given by (Lakshminarayanan et al., 2017) and even for a single model provides a MeanFE of $\approx 10^{-5}$ which is better than the model in the original Dalek emulator (MeanFe $\approx 10^{-3}$; Kerzendorf et al., 2021). In Figure 1, we show that for this architecture there is no substantial increase in accuracy with more than six models. We show that the test set TARDIS spectrum lies well within the predicted uncertainties (see Figure 3). We also show that (for now for a limited set) the uncertainty predicted by the neural network also

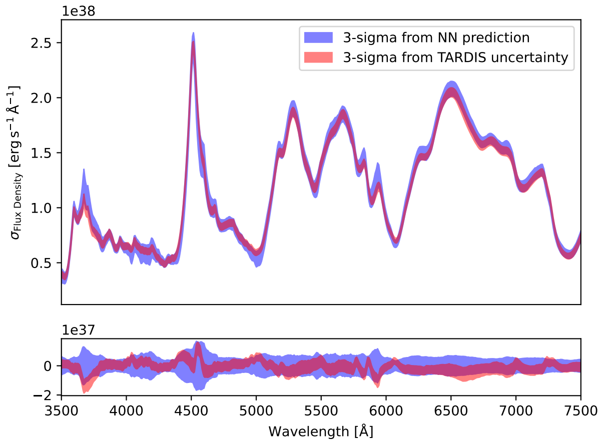

Figure 3. Comparison between the Monte Carlo uncertainty arising from TARDIS and the uncertainty estimated by the deep ensemble for test set sample 12625. We show $3-\sigma$ for visualization purposes.

captures the Monte Carlo uncertainty by TARDIS well. The preliminary work that is shown is very promising but still demands several more tests. As discussed in the introduction, the described work is part of a larger effort to build an active learning emulator for TARDIS with the express goal of doing supernova tomography and will be explored in future work.

References

Blundell, C., Cornebise, J., Kavukcuoglu, K., and Wierstra, D. Weight uncertainty in neural network. In International Conference on Machine Learning, pp. 1613–1622. PMLR, 2015.

Cranmer, K., Brehmer, J., and Louppe, G. The frontier of simulation-based inference. Proceedings of the National Academy of Sciences, 117(48):30055–30062, 2020. doi: 10.1073/pnas.1912789117.

Gal, Y. and Ghahramani, Z. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In international conference on machine learning, pp. 1050–1059. PMLR, 2016.

Gneiting, T. and Raftery, A. E. Strictly proper scoring rules, prediction, and estimation. Journal of the American statistical Association, 102(477):359–378, 2007.

Goodfellow, I. J., Shlens, J., and Szegedy, C. Explaining and harnessing adversarial examples. ICLR, 2015.

Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. On calibration of modern neural networks. In International Conference on Machine Learning, pp. 1321–1330. PMLR, 2017.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770–778, 2016.

Kerzendorf, W., Sim, S., Vogl, C., Williamson, M., Pássaro, E., Flörs, A., Camacho, Y., Jančauskas, V., Harpole, A., Nöbauer, U., Lietzau, S., Mishin, M., Tsamis, F., Boyle, A., Shingles, L., Gupta, V., Desai, K., Klauser, M., Beaujean, F., Suban-Loewen, A., Heringer, E., Barna, B., Gautam, G., Fullard, A., Cawley, K., Smith, I., Singhal, J., Arya, A., Sondhi, D., Barbosa, T., Yu, J., Patel, M., O'Brien, J., Varanasi, K., Gillanders, J., Savel, A., Reinecke, M., Eweis, Y., Bylund, T., Bentil, L., Eguren, J., Alam, A., Bartnik, M., Magee, M., Shields, J., Livneh, R., Rajagopalan, S., Chitchyan, S., Mishra, S., Reichenbach, J., Jain, R., Floers, A., Brar, A., Singh, S., Selsing, J., Sofiatti, C., Talegaonkar, C., Bot, T., Kowalski, N., Yap, K., Patel, P., Sharma, S., Prasad, S., Venkat, S., Dasgupta, D., Zaheer, M., Gupta, S., Volodin, D., Patra, N., Singh Rathore, P., Lemoine, T., Sarafina, N., Kolliboyina, C., Sandler, M., Nayak U, A., Aggarwal, Y., Kumar, A., Holas, A., Kharkar, A., kumar, a., and Wahi, U. tardis-sn/tardis: Tardis v2022.05.08, May 2022. URL https://doi.org/10.5281/zenodo.6527890.

Kerzendorf, W. E. and Sim, S. A. A spectral synthesis code for rapid modelling of supernovae. MNRAS, 440:387–404, May 2014. doi: 10.1093/mnras/stu055.

Kerzendorf, W. E., Vogl, C., Buchner, J., Contardo, G., Williamson, M., and van der Smagt, P. Dalek: A deep learning emulator for tardis. The Astrophysical Journal Letters, 910(2):L23, 2021.

Lakshminarayanan, B., Pritzel, A., and Blundell, C. Simple and scalable predictive uncertainty estimation using deep ensembles. In Advances in Neural Information Processing Systems, volume 30. Curran Associates, Inc., 2017.

Liu, J., Lin, Z., Padhy, S., Tran, D., Bedrax Weiss, T., and Lakshminarayanan, B. Simple and principled uncertainty estimation with deterministic deep learning via distance awareness. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. F., and Lin, H. (eds.), Advances in Neural Information Processing Systems, volume 33, pp. 7498–7512. Curran Associates, Inc., 2020.

O'Brien, J. T., Kerzendorf, W. E., Fullard, A., Williamson, M., Pakmor, R., Buchner, J., Hachinger, S., Vogl, C., Gillanders, J. H., Flörs, A., and van der Smagt, P. Probabilistic Reconstruction of Type Ia Supernova SN 2002bo. ApJ Letter, 916(2):L14, August 2021. doi: 10.3847/2041-8213/ac1173.

Ovadia, Y., Fertig, E., Ren, J., Nado, Z., Sculley, D., Nowozin, S., Dillon, J., Lakshminarayanan, B., and Snoek, J. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. In Advances in Neural Information Processing Systems, volume 32. Curran Associates, Inc., 2019.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. Pytorch: An imperative style, high-performance deep learning library. In Advances in Neural Information Processing Systems 32, pp. 8024–8035. Curran Associates, Inc., 2019.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. Scikit-learn: Machine learning in python. the Journal of machine Learning research, 12:2825–2830, 2011.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. Intriguing properties of neural networks. In International Conference on Learning Representations, 2014.

Van Amersfoort, J., Smith, L., Teh, Y. W., and Gal, Y. Uncertainty estimation using a single deep deterministic neural network. In International Conference on Machine Learning, pp. 9690–9700. PMLR, 2020.

van der Smagt, P. and Hirzinger, G. Solving the ill-conditioning in neural network learning. Lecture notes in computer science, pp. 193–206, 1998.

Vogl, C., Sim, S. A., Noebauer, U. M., Kerzendorf, W. E., and Hillebrandt, W. Spectral modeling of type II supernovae. I. Dilution factors. A&A, 621: A29, Jan 2019. doi: 10.1051/0004-6361/201833701.

Vogl, C., Kerzendorf, W. E., Sim, S. A., Noebauer, U. M., Lietzau, S., and Hillebrandt, W. Spectral modeling of type ii supernovae - ii. a machine-learning approach to quantitative spectroscopic analysis. A&A, 633:A88, 2020. doi: 10.1051/0004-6361/201936137.