

Using Empirical Models of Type Ia Supernovae to Prepare for Next-Generation
Cosmological Measurements

by

Samantha Mary Dixon

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Physics

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Saul Perlmutter, Chair

Professor Daniel Kasen

Professor Fernando Pérez

Fall 2020

Using Empirical Models of Type Ia Supernovae to Prepare for Next-Generation
Cosmological Measurements

Copyright 2020
by
Samantha Mary Dixon

Abstract

Using Empirical Models of Type Ia Supernovae to Prepare for Next-Generation
Cosmological Measurements

by

Samantha Mary Dixon

Doctor of Philosophy in Physics

University of California, Berkeley

Professor Saul Perlmutter, Chair

Type Ia supernovae (SNe Ia) remain one of the best astronomical tools available for measuring large distances. Observations of SNe Ia were instrumental in the discovery of the accelerating expansion of the universe, and will be key to understanding the nature of the “dark energy” driving this accelerating expansion. Surveys like LSST and the Roman Space Telescope supernova survey are currently being designed to discover and follow a large number of SNe Ia – a large enough number that the statistical error will be far subdominant to the systematic error. In order to avoid biases in the cosmological parameters constrained by these future surveys, it is essential that we understand the sources and effects of systematic uncertainties. This dissertation addresses some of these systematic biases.

In Chapter 2, we present two measurements of the extent of charge transfer inefficiency (CTI) in the detectors of the SNIFS instrument that was used to collect much of the data used throughout this dissertation. If the inefficiency is too high in spectroscopic instruments like SNIFS, the smearing that CTI causes can lead to misinterpretation of the resulting spectra. We find that the CTI is low in all detectors (about 1 photoelectron in every million is trapped), and that the low CTI remains stable over time.

Chapter 3 focuses on modeling a particular spectral region of Type Ia supernova spectra near maximum brightness. This region of the spectrum is used in some subclassification schemes of SNe Ia, and can also serve as a proxy for identifying changes in the populations of these subtypes with redshift. Our model can provide accurate measurements of ejecta velocities and the feature equivalent width using low-resolution and/or noisy spectra. Being able to use lower-quality spectra allows us to mitigate bias out to earlier eras of cosmic history by allowing us to monitor population drifts at higher redshifts.

In Chapter 4, we study two empirical models of Type Ia supernova spectral evolution (SALT2 and SNEMO) and measure how well they can capture a variety of near-maximum spectro-

scopic features. Our goal is to analyse how linear spectral models with differing number of parameters can capture non-linear features like ejecta velocities. In addition, we present a model for producing realistic mock spectra based on these models, allowing future studies to have access to spectral templates that capture the full range of supernova spectral behavior.

Chapter 5 centers on an assumption of linear regression that is often overlooked in supernova cosmology analyses. These analyses perform an initial linear regression to correct the observed SN absolute magnitudes for other properties of the SN. They then perform a second regression to correct the residuals of this first regression for an additional covariate. This practice is statistically sound only if the covariates in the initial regression are not correlated with the covariates used in the second regression. However, these correlations do exist. We present a toy model of this problem to calculate closed-form expressions and scaling relations of the size of the biases in the effect sizes and estimated scatter that come from this overlooked assumption. We also use simulations based on literature data to calculate the size of these biases and provide potential corrections.

Chapter 6 presents two new models of Type Ia supernova spectroscopy that were constructed using deep learning. These models extend the “twins embedding” model of Boone et al. (submitted) into a wide range of phases. The `spec2embed` model takes as input a spectrum observed at any phase from -10 to +40 days after maximum brightness, and predicts the spectrum’s phase and its supernova’s location in the twins embedding space. Using these predictions, we can standardize supernovae from single spectra with comparable precision to the original twins embedding work. The `embed2spec` model works in reverse, taking a phase (or range of phases) and location in the twins embedding space to predict a spectrum. With this, we can use forward-modeling fitting techniques to constrain a supernova’s location in the twins embedding space from multiple spectra, spectra with lower spectral resolution, or even broadband photometry.

Contents

Contents	i
List of Figures	iii
List of Tables	ix
1 Introduction	1
1.1 The Expansion History of the Universe	1
1.2 Using Standard Candles to Constrain Cosmological Parameters	4
1.3 Type Ia Supernovae as Standardizable Candles	5
1.4 SN Ia Standardization Beyond SALT2	7
1.5 Open Questions in Supernova Cosmology	9
1.6 Dissertation Outline	12
2 Measurements of Charge Transfer Efficiency in the SNIFS Detectors	15
2.1 Introduction	15
2.2 Initial Characterization with Fe ⁵⁵ X-rays	16
2.3 Cosmic Ray Measurement	21
2.4 Conclusion	25
3 A Study of the Morphology of the SiII λ6355 Feature in SNe Ia	28
3.1 Overview	28
3.2 Data	30
3.3 Spectral Feature Measurement	32
3.4 EMFA of the SiII λ 6355 Feature	36
3.5 Validation	41
3.6 Testing on External Data	42
3.7 Simulated Roman Prism Spectra	45
3.8 Conclusions	49
4 Kernel Density Estimates for Generating Mock Type Ia Supernova Observations with SALT2 and SNEMO	53
4.1 Overview	53

4.2	SNEMO	55
4.3	Data	55
4.4	Kernel Density Estimation	56
4.5	Modeling the Data and Making Mock Observations	57
4.6	Evaluating Spectral Diversity	62
4.7	Conclusions	69
5	Biases from Non-Simultaneous Regression with Correlated Covariates in Supernova Cosmology	71
5.1	Overview	71
5.2	Toy Model: Two-dimensional Linear Regression with Correlated Covariates	73
5.3	Step Function Corrections	74
5.4	Comparison to Data	75
5.5	Conclusions	79
6	High-Precision Distance Moduli from Single-Epoch Spectrophotometry Using Deep Learning	80
6.1	Introduction	80
6.2	The Twins Embedding Analysis	81
6.3	Data	83
6.4	Models	86
6.5	<code>spec2embed</code> Results	89
6.6	Recovering Spectra with <code>embed2spec</code>	97
6.7	Conclusions	101
7	Conclusions	104
Bibliography		106
A	Optimal Savitsky-Golay Smoothing Windows	113
B	Table of Si II λ6355 Velocities and Pseudo-equivalent Widths	115
C	Kernel Density Estimation	120
C.1	KDE in 1 Dimension	120
C.2	KDE in d Dimensions	121
D	Measuring Spectral Features	126
E	Regression Bias Derivations	128
E.1	Derivation of Regression Parameters in the Simultaneous Case	128
E.2	Derivation of Biases on Regression Parameters in the Non-Simultaneous Case	129
E.3	Step Function Correction Derivations	130

List of Figures

1.1	Hubble-Lemaître diagrams for different cosmological models. In blue, we have a fiducial, flat Λ CDM cosmology with density parameters matching those determined by Komatsu et al. (2009). In orange, we show another flat Λ CDM cosmology, but dominated entirely by matter (i.e. $\Omega_\Lambda = 0$). In green, we show an example of the distance-redshift relation for an alternate parametrization of dark energy with a varying equation of state parameter.	5
2.1	Example X-ray image for a single amplifier.	17
2.2	Example spectrum of the Fe ⁵⁵ X-rays events extracted from a single amplifier frame along with the best-fit model. The results of this fit are used to determine the amplifier gain from each frame.	18
2.3	The number of electrons in extracted Fe ⁵⁵ X-ray events as a function of y-location on the CCD for each camera amplifier (equivalent to number of transfers in the parallel direction). The slope of the best-fit line divided by the expected number of electrons gives us an estimate of the charge transfer inefficiency in the parallel registers.	19
2.4	Same as Figure 2.3, but for the x-location of events. The best-fit lines are used to estimate the charge transfer inefficiency in the serial registers.	20
2.5	Example identified cosmic ray hits passing our ellipticity cut. The two neighboring hits seen in the bottom right example would be removed from the final sample because they are too close to each other. This culling removes some of the noise from our final signal.	22
2.6	Example measurement of CTI in the parallel register from cosmic ray trails from a single night. All blue dots represent a single cosmic ray hit. The red points show the median fraction of counts in the peak of the hits that end up in the trails in each y-position bin. The best-fit line is also shown. The slope of this line gives us the CTI.	23
2.7	Same as Figure 2.6 but in the serial direction.	23
2.8	Same as Figure 2.6 but zoomed to show the median values and their associated uncertainties.	24
2.9	Same as Figure 2.8 but in the serial direction.	24

2.10 A search for potential time variation in the charge transfer efficiency of each of the spectroscopic cameras. CTI measurements in the parallel direction are shown in blue and those in the serial direction are shown in orange.	26
2.11 Same as Fig 2.10 but aggregated by year.	26
3.1 An example of the preprocessing steps taken to normalize the spectra using a 13-point spline. The upper figure shows the full near-max spectrum along with the best fit spline. The lower figure shows that same spectrum with the spline pseudo-continuum removed.	31
3.2 Zoom-in on the feature of interest for this work, with and without the spline normalization. The features plotted in the right figure are colored by the objects' SALT2 color to emphasize that the effect of this spline normalization is to remove the effects of color on the shape of the spectral feature.	32
3.3 An example Si II $\lambda 6355$ feature. The original data is shown in blue along with the uncertainties. The optimal smoothing is shown in the thick black line. The blue and red spans show the location of the wavelength windows used to search for the maxima defining the pseudo-continuum (the maxima are the large black points). The pseudo-continuum is plotted as the dashed black line. The location of the maximum absorption wavelength is also shown as a large black point. . .	34
3.4 Distributions of the spectral indicators of the Si II $\lambda 6355$ feature from the training set.	35
3.5 Comparison of the Si II $\lambda 6355$ feature fit to a Gaussian and the Savitsky-Golay smoothing for SN2007nq. We can see that the blue edge of the feature is not properly captured. The minimum of the feature is also located at different wavelengths, which will produce an inaccurate velocity measurement.	36
3.6 Histograms of residuals between the velocity and equivalent width of the Si II $\lambda 6355$ line measured using a Gaussian line profile and the true velocity. There is a clear bias, with the Gaussian measurement giving velocity values that are on average 200 km/s higher than the true values, and equivalent width values that are 5 Å narrower.	37
3.7 Visualization of the components (\mathbf{x}) of the EMFA model. Each subfigure shows the effect of adding various values of multiples of the components to the mean vector.	38
3.8 Corner plot showing the marginal distributions of the loading coefficients for the training set, colored by the measured velocity of the Si II $\lambda 6355$ feature. Each point in the scatter plot represents one supernova.	39
3.9 Same as Figure 3.8, but colored by the pseudo-equivalent width.	40
3.10 Scatter plots of the loading coefficients of the training data with their measured spectral indicators. We can see that each of the components is correlated with the velocity. Only the first two components are correlated with the pseudo-equivalent width.	41

3.11 A random selection of recovered spectral features at the native resolution of the SNfactory spectra. The gray lines show the observed data, the blue line shows the data smoothed by the optimal Savitzky-Golay filter, and the orange line is the reconstructed flux.	42
3.12 Histograms of the residuals between the velocities and equivalent widths measured from the flux reconstructed using the EMFA model and the original observed flux in the training data.	43
3.13 Histograms of the residuals between the velocities and equivalent widths measured from the flux reconstructed using the EMFA model and the original observed flux in the validation set. The outlier SN2005M has been removed from these plots (see Section 3.6)	43
3.14 Validation spectra where the EMFA fit and/or pseudo-continuum determination failed. The left frame is SN2007 gi, an extremely rapidly expanding object. The middle frame is SN2005M, which may be 1991T-like. The right frame shows an example of an object for which the pseudo-continuum determination failed due to underestimated variance in the spectrum. In each subfigure, the gray line represents the SG smoothed flux, and the blue line shows the best-fit EMFA flux. In the right figure, we also show the pseudo-continuum as a dashed black line.	44
3.15 Average signal-to-noise ratio for assumed for prism spectra used in the simulations. We report the average from 5600-6600 Å in the rest frame since this is the wavelength region of interest.	46
3.16 Example realizations of the training set near-max spectra observed with the Roman prism spectrograph with a one hour exposure time.	47
3.17 Example of the various methods tested to recover the spectral indicators from the low resolution, noisy spectra from the Roman prism. The original training set spectrum that was resampled and noised is shown in gray. The realization of the prism observation (with uncertainties) is shown as the black data points. The best-fit EMFA spectral feature is shown in blue, the smoothed version of the observed flux is shown in orange, and the best-fit Gaussian line profile is shown in green.	48
3.18 Per-redshift-bin average of the absolute value of differences between measured velocities and true velocities as a function of redshift for the simulated Roman prism spectra. At low redshifts, the SG filtering method captures the true value best, but as noise increases, the EMFA method outperforms all other methods.	49
3.19 Same as Figure 3.18, but for pseudo-equivalent width measurements. At low redshifts, all methods give comparable errors, but as the noise increases, the EMFA method is the preferred technique.	50
3.20 Per-redshift-bin average of the per-object standard deviations of the velocity measured with three techniques plotted as a function of redshift for the simulated Roman prism spectra. The EMFA recovery technique outperforms both the Gaussian and SG filter smoothing techniques at all redshifts.	51

3.21 Same as Figure 3.20, but for pseudo-equivalent width measurements. In contrast to the velocity measurement uncertainty, the SG filter smoothing methods seems to perform better than the Gaussian measurement in this metric. However, the EMFA still out-performs both methods.	52
4.1 Corner plot showing the one- and two-dimensional marginal parameter distributions of the SALT2 parameters for the SNfactory data set (black points and histogram lines), as well as the 1-, 2-, and 3- σ confidence intervals of the marginalized distribution of samples drawn from the KDE trained on these data.	58
4.2 Same as Fig. 4.1, but for SNEMO2	59
4.3 Same as Fig. 4.1, but for SNEMO7	60
4.4 Same as Fig. 4.1, but for SNEMO15	61
4.5 A series of synthesized spectral observations at a range of phases for a single object at redshift $z = 0.775$ generated from a random draw from the SNEMO15 KDE. The resolution matches the proposed design of the Roman Space Telescope prism spectrograph, and the signal-to-noise ratio representing the level that could be obtained with an hour of exposure time. The black lines represent the underlying spectral energy distribution generated by our model, and the blue points represent the flux values in each prism wavelength bin, along with their associated errors.	63
4.6 Synthetic photometry of the same object shown in Fig. 4.5, but observed photometrically in the Roman Wide Field Instrument bandpasses. Again, the lines represent the underlying flux of the generated supernova. The points represent an example of the noisy observed flux, and its associated errors.	64
4.7 Histograms of the residuals between the velocity of the Si II $\lambda 6355$ line as measured from the data and as measured from the spectrum generated with the best-fit spectral model parameters. Lower dimensional models (SALT2 and SNEMO2)	66
4.8 Normalized median absolute deviation of residuals between spectral features measured from data spectra and from spectra generated from the best-fit spectral model. In general, spectral models with more parameters more closely capture the spectral feature measurements.	67
4.9 Empirical cumulative distribution functions of each spectral indicator for the data set and samples from the kernel density estimate of the spectral model parameter spaces.	68
4.10 Same as Fig. 4.9, but for samples from the multivariate Gaussian estimation of each of the spectral model parameter spaces.	68
4.11 Cramér distances between spectral indicator distributions for the SNfactory data and samples from the Gaussian estimates of the SALT2 and SNEMO parameter distributions, the modeled at-max spectra for the training data, and the KDE estimates of the spectral model parameter distributions.	70

6.1	Histogram of SALT2 phases for the complete data set. The phases are estimated from SALT2 fits to light curves synthesized from the full spectral time-series data of each supernova.	85
6.2	Comparison of phase and extinction parameters determined by the <code>spec2embed</code> model and ground truth values presented in B20a. The solid lines in each of the upper panels represents equivalence. We also note the root-mean-squared-error (RMSE) and the normalized median absolute deviation (NMAD) of the residuals (predicted value – true value) for each property. The lower panels show these residuals as a function of the true values. Shaded regions in the lower panels indicate the 1-, 2-, and 3- σ fluctuations of these residuals about the median. The secondary axis in the bottom right panel indicates the linear scaling between the error on the extinction parameter and the final standardized magnitude.	91
6.3	Same as Figure 6.2, but comparing the predicted and ground-truth twins embedding coordinates ξ_1 , ξ_2 , and ξ_3	92
6.4	Standard deviation and NMAD of the difference between the magnitudes predicted from the neural network predicted embeddings and those predicted from the true embeddings as a function of the spectral phase.	94
6.5	Same as Figure 6.2, but using the leave-one-out cross-validation analysis. As in Figure 6.2, each point represents a single spectrum. However, the predictions are made using networks trained on every supernova except the supernova the spectrum comes from.	95
6.6	Same as Figure 6.5, but for the embedding coordinates.	95
6.7	Standard deviation and NMAD of the standardization residuals ($GP(pred) - \Delta m_{true}$) as a function of phase for the full leave-one-out cross-validation analysis. We also show the value of this metric for the twins embedding analysis, as well as for the SALT2 model.	97
6.8	Comparison of the observed spectral time-series data (in black dashed lines) to the predicted data (blue solid lines) for PTF09dnl, a supernova in our held-out test set.	98
6.9	Similar to Figure 6.8, but using synthesized photometry to compare the observer-frame light curves (dots) to the model predictions (solid lines). The band passes used are defined in Table 6.6.	99
6.10	RMS of <code>embed2spec</code> flux residuals in each wavelength bin, normalized by the average flux in the bin across all phases. The thick black line shows the overall normalized RMS flux residual, and the thinner colored lines show the normalized RMS flux residual for subgroups selected by phase.	100
6.11	RMS of the flux residuals normalized by the average flux over all wavelengths at each phase. Most of the outlying points belong to either SN2009hs, a 91bg-like peculiar supernova, or LSQ12lhs, an 02cx-like peculiar supernova.	101
6.12	Phase-averaged RMS flux residual for all observed spectra for each test set supernova as a function of twins embedding space coordinate.	102

C.1 Example of a kernel density estimate of a 1-dimensional, non-Gaussian distribution. The ticks represent the $n = 250$ observations drawn from the true distribution shown with the black dashed line. The blue, orange, and green lines represent KDEs with increasing bandwidth sizes.	121
C.2 Results from 5-fold cross-validation for the example distribution and data shown in Fig. C.1. The blue line and shading show the mean and standard deviation of the normalized scores from all of the cross-validation subsets. The vertical blue line shows the location of the maximum score. The dashed black line shows the location of the Silverman's rule-of-thumb estimate for the optimal bandwidth.	122
C.3 Example kernel density estimates in 2 dimensions for a joint probability distribution that is correlated and non-Gaussian. The $n = 250$ data points are shown as black dots. The black contour lines show the true 1, 2, and 3 σ contours, while the fading shades of blue show the 1, 2, and 3 σ estimates of the KDE. In each of these examples, we assume a Gaussian kernel with no covariance terms.	123
C.4 The process of transforming the data to use a kernel that captures the data covariance. The first panel shows the data in the original coordinates. The second panel shows the data after being transformed by the whitening transformation. The third panel shows the KDE fit with a normal Gaussian kernel, and the final panel shows the data and the KDE reprojected back into the original coordinates. In each panel, the same data points are shown as black dots. The true 1, 2, and 3 σ contours are shown in differing shades of blue. The black lines indicate the 1, 2, and 3 σ confidence intervals of the best-fit KDE.	125
C.5 Comparison of the final KDE for our toy example using a) the best-fit Gaussian kernel with no covariance transformation and b) the best-fit Gaussian kernel with whitening applied. In both panels, the blue shading indicates the 1, 2, and 3 σ contours of the true toy model distribution, and the black lines indicate similar contours for the kernel density estimates of the distribution.	125

List of Tables

2.1	Charge transfer inefficiency results from Fe ⁵⁵ X-ray characterization. All values are in units of 10^{-6}	18
2.2	CTI measurements over time from all dark frames collected from 2006 to 2012. All values are in units of 10^{-6}	25
3.1	Important wavelengths for measuring indicators of the Si II $\lambda 6355$ feature. λ_b and λ_r are the blue and red windows used when defining the pseudo-continuum.	33
3.2	Measurements of the spread of the residuals between spectral features measured from the EMFA modeled spectra and spectral features measured from the true spectra for both the SNfactory (training) set and the BSNIP (test) set.	44
5.1	Pearson correlation coefficients between SALT2 parameters and host galaxy masses (measured as $\log(\mathcal{M}_{\text{host}}/\mathcal{M}_\odot)$). Each data set shows a relatively strong correlation between x_1 and mass, indicating that biases can be introduced from non-simultaneous regression.	76
5.2	Ranges for the standardization hyperparameters used in the simulation analysis.	77
5.3	Linear transformation coefficients (see Equation 5.12) between the standardization hyperparameter α , representing the light curve shape-luminosity correction, obtained with a non-simultaneous fit and the true values.	78
5.4	Same as Table 5.3, but for the standardization hyperparameter β , representing the color-luminosity correction.	78
5.5	Same as Table 5.3, but for the standardization hyperparameter γ , representing the host mass-luminosity correction.	78
6.1	MLP hyperparameters values used in Bayesian optimization along with the preferred values for each model.	88
6.2	Summary of standardization results using the B20b embedding coordinates and extinction parameters using our training and test sets. Weighted measures of the spread are weighted by the number of spectral observations for an individual supernova. All values are consistent with the values found in B20b.	90

6.3	Standardization errors for the training and test data sets using the Gaussian process regression model conditioned on the B20a embedding coordinates to predict the absolute magnitudes based on the B20a embedding coordinates ($\text{GP}(\text{true})$) and the <code>spec2embed</code> neural-network predicted embedding coordinates ($\text{GP}(\text{pred})$), as well as the average difference between these predictions. Errors on the statistics are determined via bootstrap resampling.	93
6.4	Comparison of parameter accuracy between the test set evaluation in Section 6.5 and the LOOCV evaluations.	96
6.5	Standardization results using a full leave-one-out cross-validation analysis. Each measure of the spread was obtained by training a separate <code>spec2embed</code> model and GP regression model for each supernova, where both models are conditioned on the data from all SNe except that held-out SN. Uncertainties on the values of these metrics were determined via bootstrap resampling.	96
6.6	Top-hat band pass definitions.	99
B.1	The baseline spectral feature data used throughout the Chapter 3 analysis for both the training Nearby Supernova Factory data and the validation BSNIP sample.	115
D.1	Extrema limits and rest-frame minimum wavelengths for the spectral indicators studied.	127

Acknowledgments

Writing this dissertation would not have been possible without the advice, encouragement, and kindness of my advisors, colleagues, family, and friends.

I first want to thank my advisor, Saul Perlmutter, for being such an enthusiastic and supportive mentor. Working with Saul has shaped me as a scientist, and I'm deeply grateful to have had his guidance while also having the freedom to pursue the ideas that interested me the most. Greg Aldering has also been a valued mentor; I have him to thank for most of what I know about observing and for pointing me in the right directions early on in my research career. I also want to thank my undergraduate thesis advisor, Paolo Privitera, for encouraging me to go to graduate school in the first place, and for pushing me to aim high and apply to Berkeley. I'm only writing this now because of that push. Thank you to my qualification exam and dissertation committee members, Bill Holzapfel, Dan Kasen, and Fernando Pérez, who gave me plenty of advice to guide my work and to better this document.

Many thanks are due to all of the officemates I've had over the years, including Kyle Boone, Xiaosheng Huang, Kara Ponder, Ravi Gupta, Clare Saunders, Aleksander Cikota, and Brian Hayden. Each of them has taught me an immeasurable amount about physics, astronomy, statistics, computing, writing, and life that made all of the work here possible. I already miss all of our lunch-time conversations in the LBL cafeteria. Ben Rose, David Rubin, and Rebekah Hounsell have been wonderful colleagues over the years. I'm grateful for all of the Slack chats, paper editing sessions, code reviews, and answers to my silly questions.

Jeffrey Epstein and Jennet Dickinson have been my greatest friends throughout grad school, from staying up late first year to cram for prelims to getting up early 6 years later to make dissertation edits. I'm thankful to have had the world's best homework helpers, brunch partners, breakup text proofreaders, and close confidants on my side all these years. I also want to thank Jessica Avva for helping me to prove that being obsessed with the Bachelor doesn't have to be antithetical to becoming a great scientist, and for being such a strong and supportive friend. I could not have finished writing all this up without the tireless support (and homecooked meals) of Michael Mangus; thank you for always reminding me how capable I am.

Finally, I want to thank my parents for their unconditional love and support, and for trying their best to understand even when they don't have any clue what I'm talking about. Thank you for instilling in me the love of learning that brought me to Chicago and to Berkeley, and for raising me to have the confidence to see it all through.

Chapter 1

Introduction

1.1 The Expansion History of the Universe

Understanding how space has expanded over the history of the universe is key to understanding what the universe is made of, how it started, and how it may end. We start by defining the scale factor, a , which describes the ratio of the physical distances between objects in the past to the physical distances between the same objects today. Our quest to measure the expansion history of the universe, then, is equivalent to determining $a(t)$ – how the scale factor evolves with cosmic time.

The cosmological principle descends from the Copernican principle: that there is nothing special about our particular viewpoint of the universe. More precisely, it states that the expansion of the universe is homogenous (the same at any point in space) and isotropic (the same in all directions). Encoding this mathematically, we can obtain the following metric, known as the Friedmann-Lemaître-Robertson-Walker (FLRW) metric:

$$ds^2 = a(t)^2 ds_3^2 - c^2 dt^2 \quad (1.1)$$

where ds_3 is the three-dimensional spatial metric and c is the speed of light.

Using this metric in the Einstein field equations, we then derive the Friedmann equations, a general description of how a evolves with time:

$$\left(\frac{\dot{a}}{a}\right)^2 = \frac{8\pi G\rho}{3} - \frac{kc^2}{a^2} \quad (1.2)$$

$$\frac{\ddot{a}}{a} = -\frac{4\pi G}{3} \left(\rho + \frac{3p}{c^2}\right) \quad (1.3)$$

Dots here represent a time derivative, and G is Newton's gravitational constant. The energy density, represented by ρ , and the energy pressure, p are functions of time, like the scale factor a . The spatial curvature, k , is constant, but can take on a number of values.

To simplify our notation, we define the Hubble rate:

$$H = \frac{\dot{a}}{a}.$$

With this, the left hand side of Equation 1.2 can be replaced with H^2 and the left hand side of Equation 1.3 can be replaced with $\dot{H} + H^2$. The value of the Hubble rate today is known as the Hubble constant, H_0 .

Assuming $k = 0$ (an assumption that is experimentally justified from observations of the cosmic microwave background, see Planck Collaboration et al. (2016), for example), we can use Equation 1.2 to rewrite Equation 1.3 as

$$\dot{\rho} = -3H\left(\rho + \frac{p}{c^2}\right) \quad (1.4)$$

For linear combinations of perfect fluids, each with equation of state

$$p_f = w\rho_f c^2,$$

where w is known as the equation of state parameter, this equation has exact solutions of the form

$$\rho_f \propto a^{-3(1+w)}.$$

For ordinary matter, $w = 0$. For radiation, $w = 1/3$ and for a cosmological constant dark energy, $w = -1$. Thus, Equation 1.4 has solutions of the form:

$$\rho = \rho_m a^{-3} + \rho_r a^{-4} + \rho_\Lambda$$

Additionally, we define the critical density

$$\rho_c = \frac{3H_0^2}{8\pi G},$$

which describes the average energy density of a universe that would halt expansion after infinite time. Dividing the densities by this critical density gives us a unitless density parameter Ω . Putting everything, together, we can obtain the relationship between the Hubble parameter and scale factor, parametrized by the time-evolving densities of the constituent components of the universe:

$$H(a) = H_0 \sqrt{\Omega_m a^{-3} + \Omega_r a^{-4} + \Omega_\Lambda} \quad (1.5)$$

Oftentimes, it is convenient to use the redshift, z , in lieu of the scale factor, since the redshift is directly observable. We define the redshift as the Doppler-like shift observed in distant objects:

$$z = \frac{\lambda_o}{\lambda_e} - 1 \quad (1.6)$$

where λ_o is the observed wavelength and λ_e is the emitted wavelength of some observable spectral line. The redshift we observe has a number of sources. For our purposes, the main contributors are the cosmological redshift, where the expansion of space stretches of the wavelength of emitted light causing it to appear redder, and the peculiar velocity redshift, where the motion of objects relative to us creates a Doppler shift in the observed wavelengths. Generally, when we refer to the redshift of an object, we are referring to its cosmological redshift, which is related to the scale factor by $z = 1/a - 1$. Using this definition in Equation 1.5, we get

$$H(z) = H_0 \sqrt{\Omega_m(1+z)^3 + \Omega_r(1+z)^4 + \Omega_\Lambda} \quad (1.7)$$

We're now in a position to begin to define some useful measures of cosmological distance, using the equations that govern the evolution of distances described above. One important distance measure for the work described here is the comoving distance, χ , between an emitting object at scale factor a and us ($a = 1$). This distance remains constant with the universe's expansion and can be calculated by multiplying the speed of light with the aggregate of infinitesimal distances over time:

$$\chi = c \int_{t(a)}^{t_0} \frac{dt'}{a(t')} = -c \int_1^a \frac{da'}{a'^2 H(a)} = \frac{c}{H_0} \int_0^z \frac{dz'}{\sqrt{\Omega_m(1+z')^3 + \Omega_r(1+z')^4 + \Omega_\Lambda}} \quad (1.8)$$

The angular diameter distance describes the ratio of an object's physical size to its angular size on the sky, and is related to the comoving distance by

$$d_A(z) = \frac{\chi(z)}{1+z}$$

Finally, we have the most important distance measure for our work: the luminosity distance. This distance defined by

$$d_L(z) = \sqrt{\frac{L}{4\pi F}} \quad (1.9)$$

where L is the luminosity of an object and F is the flux (or observed brightness) of the object. Interestingly, it can be shown that this quantity is equivalent to ¹

$$d_L(z) = (1+z)^2 d_A(z) = (1+z)\chi(z). \quad (1.10)$$

In astrophysical contexts, it is common to work in logarithmic-scale magnitudes. This leads to the definition of the distance modulus, μ , defined as the difference between the apparent magnitude of an object, m , and its absolute magnitude, M , which is itself defined as the apparent magnitude of an object if it were seen from a distance of 10 parsecs. The distance modulus is therefore related to the luminosity distance by

$$\mu = m - M = -2.5 \log_{10} \left[\frac{F(d_L)}{F(d_L = 10 \text{ pc})} \right] = 5 \log_{10} \left(\frac{d_L}{10 \text{ pc}} \right) - 5 \quad (1.11)$$

¹The full proof of this equivalence (known as the Etherington reciprocity theorem) is quite complicated. However, we can obtain this relation by noting that the surface brightness of an object is reduced by a factor of $(1+z)^{-4}$ as it recedes, while the angular area decreases as d_A^{-2} Hogg (1999).

1.2 Using Standard Candles to Constrain Cosmological Parameters

A classic method for obtaining estimates of the luminosity distance is to observe standard candles. A standard candle is any class of astronomical objects that has a known intrinsic luminosity. By observing the apparent brightness of these objects, we directly obtain the luminosity distance as defined in Equation 1.9 (or equivalently the distance modulus as defined in Equation 1.11). We can also use spectral observations of these objects or their host galaxies to compare the observed frequencies of various emission and absorption lines in the spectrum to the known frequencies of these lines in order to calculate each object's redshift, as in Equation 1.6. Combining these measurements gives us the Hubble-Lemaître diagram, a plot of luminosity distance (or equivalently the distance modulus) as a function of redshift. Different cosmological models predict different relationships between distance and redshift, so our standard candle observations can be used to differentiate between and constrain the parameters of different theoretical models. In Figure 1.1, we show a few examples of distance-redshift relations for different cosmological models.

The standard cosmological model is the flat, Λ CDM model, which posits that the spatial curvature of the universe is flat ($\Omega_k = 0$), and that the energy density is composed primarily of dark energy that behaves as a cosmological constant, followed by cold dark matter, and finally ordinary baryonic matter. This model is supported by the findings of a number of different probes (Planck Collaboration et al., 2016; Wittman et al., 2000; Eisenstein et al., 2005). Its Hubble-Lemaître diagram is shown in blue in Figure 1.1, where we have take the density parameter values from Komatsu et al. (2009): $\Omega_\Lambda = 0.7135$ and $\Omega_m = 0.2865$. We can see that the distance-redshift relation for this cosmology differs significantly from a similarly parametrized cosmology but with $\Omega_\Lambda = 0$ and $\Omega_m = 1$ (seen in orange in Figure 1.1).

Λ CDM is not the only cosmological model. One popular alternate model envisions dark energy as a dynamical scalar field, rather than a cosmological constant. In this case, the equation of state parameter w of dark energy is allowed to vary with redshift (or equivalently, scale factor). This variation can be parametrized by

$$w(a) = w_0 + w_a(1 - a) = w_0 + w_a z / (1 + z) \quad (1.12)$$

(see Chevallier & Polarski (2001) or Linder (2003)). A cosmological constant can be encapsulated in this parametrization by setting $w_0 = -1$ and $w_a = 0$, so deviations from these values would serve as useful clues as to the precise nature of dark energy. An example distance-redshift relation with $w_0 = -0.8$ and $w_a = -1$ is shown in green in Figure 1.1. We can see that the difference in the current fiducial cosmology distance-luminosity relationship and that of this alternate cosmology is quite small, but increasing with redshift. In order to constrain the parameters of these alternate models, we need to have extremely precise measurements of the luminosity distance, and also be able to extend these measurements further back in time (i.e. to higher redshift).

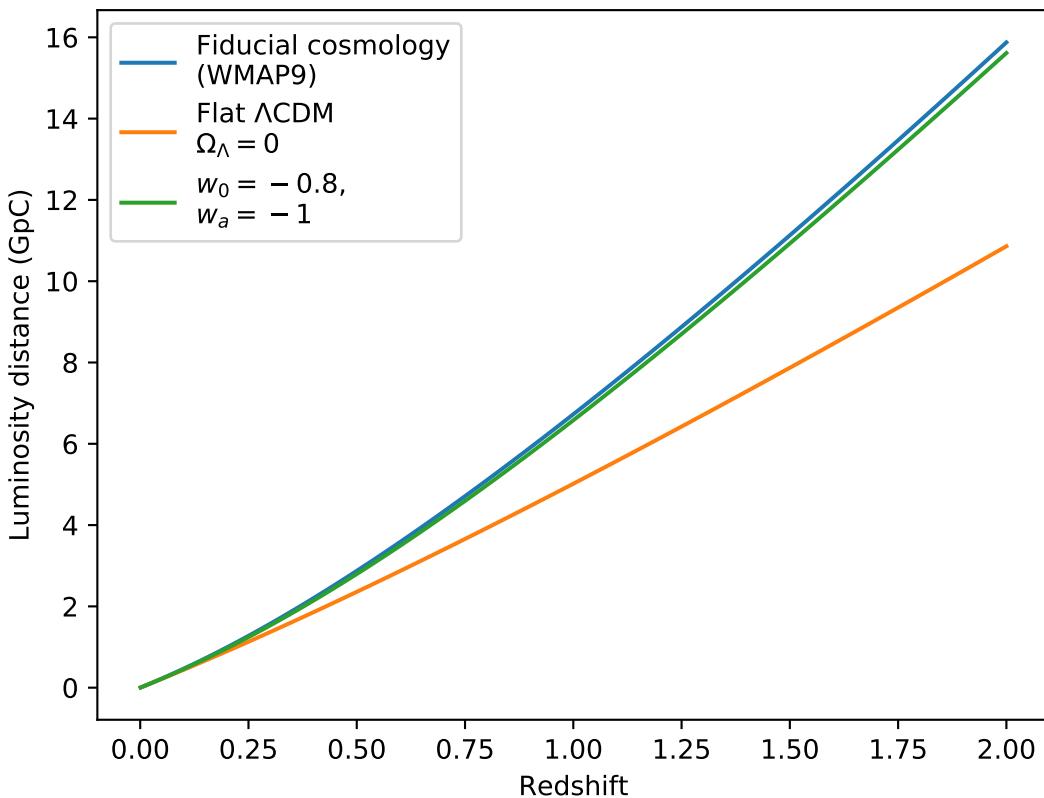


Figure 1.1: Hubble-Lemaître diagrams for different cosmological models. In blue, we have a fiducial, flat Λ CDM cosmology with density parameters matching those determined by Komatsu et al. (2009). In orange, we show another flat Λ CDM cosmology, but dominated entirely by matter (i.e. $\Omega_\Lambda = 0$). In green, we show an example of the distance-redshift relation for an alternate parametrization of dark energy with a varying equation of state parameter.

1.3 Type Ia Supernovae as Standardizable Candles

Type Ia supernovae (SNe Ia), exploding carbon-oxygen white dwarfs in binary systems, are excellent standard candle candidates. They are observed to have very similar intrinsic brightnesses. Moreover, they are extremely bright — about 5 billion times brighter than the Sun — so they can be seen out to large distances, and therefore probe earlier epochs of cosmic history. SNe Ia were instrumental in discovering that the expansion rate of the universe is accelerating, and they remain one of the best tools we have for constraining the properties of the dark energy that drives this accelerating expansion (Perlmutter et al., 1999; Riess et al., 1998).

While Type Ia supernovae do have similar intrinsic brightnesses, they are not perfect standard candles. The scatter in their peak absolute magnitudes is $\sim 40\%$. They are,

however, standardizable candles; that is, their peak brightnesses are strongly correlated with other observable properties of the explosion.

One of these observables is the light curve width (sometimes referred to as the decline rate), a measure of how quickly the event brightens and fades. SNe Ia with broader light curves tend to be intrinsically brighter. The relationship was first shown empirically in the 1970s (Rust, 1974; Pskovskii, 1977) and later refined with additional observations and extending out to higher redshifts (Phillips, 1993; Hamuy et al., 1996; Perlmutter et al., 1997). The color of the supernova is also known to correlate with the brightnesses of SNe Ia – bluer objects tend to be brighter. The width-luminosity relationship is theoretically explained as being due to differences in the opacity due to temperature differences (Kasen & Woosley, 2007). The color-luminosity relationship is not yet well-explained theoretically, but has been reproduced in a number of theoretical models (e.g. Kasen et al., 2009). Tripp (1998) introduced the combined correction for both width and color, enabling Type Ia supernova magnitudes to be standardized to within approximately 0.15 magnitudes, and similar relationships have been used in models like the multicolor light curve shapes model (MLCS, Riess et al., 1996).

Currently, the conventional method for standardization involves fitting observed broad-band light curves with the SALT2 model of Guy et al. (2007) in order to obtain measurements of the light curve width and the color. SALT2 does not directly measure the light curve width or color like previous methods, which use parameters like Δm_{15} , the decrease in Bessell B-band magnitudes from maximum light to 15 days after maximum. Instead, the model parametrizes the full evolution of the spectral fluxes of Type Ia supernovae as follows:

$$f(\lambda, p) = x_0 (M_0(\lambda, p) + x_1 M_1(\lambda, p)) \times \exp[c \times CL(\lambda)] \quad (1.13)$$

where λ represents wavelength and p is the number of rest-frame days after maximum brightness (also known as the phase). M_0 and M_1 are fixed spectral sequences that describe the mean spectral evolution of SNe Ia and their major sources of spectral variation, respectively. $CL(\lambda)$ is a fixed description of the color variation that remains fixed across phases, and includes contributions from both intrinsic color differences and color differences due to host galaxy dust extinction. This model can be thought of as akin to a principal component decomposition of SN Ia spectral evolution, with x_0 , x_1 , and c indicating where each supernova is located in the model basis and representing each supernova's overall brightness, light curve width, and redness, respectively. Using such a parametrization avoids systematic errors from K-corrections that stem from large variations in spectral features.

After using the model to fit each supernova in a given data set, we typically standardize their observed brightnesses using a modified version of the Tripp relation:

$$\mu = m_B^* - M + \alpha \times x_1 - \beta \times c \quad (1.14)$$

where m_B^* is the apparent B-band magnitude calculated from the best-fit model parameters x_0 , x_1 , and c for each supernova and μ is the distance modulus to each supernova. M , α , and β are global parameters that describe the overall standardized absolute magnitude in the

Bessell B-band, the light curve shape-luminosity relation, and the color-luminosity relation, respectively. Typically, we start by assuming some fiducial cosmology that defines a distance μ_{cosmo} , and we fit the global standardization parameters M , α , and β by minimizing the χ^2 defined by

$$\chi^2 = (\mu - \mu_{\text{cosmo}})^T C^{-1} (\mu - \mu_{\text{cosmo}}), \quad (1.15)$$

where C defines the covariance matrix between SN Ia observations. Then by fixing these values of M , α , and β , we can perform a similar χ^2 minimization that finds the best-fit cosmological parameters. The process repeats until convergence.

Alternatively, the UNITY framework (Rubin et al., 2015) performs this standardization and cosmology fit from light curve parameters with a fully Bayesian approach. In this case, the nuisance parameters and cosmological parameters, as well as their full posterior distributions, are obtained using Hamiltonian Monte Carlo techniques.

Regardless of the method of fitting, the residuals on the Hubble diagram (i.e. the differences between the distance moduli obtained from the best standardization of the light curve parameters and the distance moduli predicted by the best-fit cosmological parameters) still have a root-mean-square (RMS) dispersion of approximately 0.15 magnitudes. Some of this dispersion can be explained by the measurement, calibration, and model uncertainties of the photometry and light curve model fitting process. However, some amount of the dispersion (typically about 0.10 magnitudes) remains unexplained.

1.4 SN Ia Standardization Beyond SALT2

The remaining unexplained dispersion in standardized supernova magnitudes from using SALT2 and the Tripp relation indicates that there may be more progress to be made in SN Ia standardization. One method to make such an improvement is to look for additional parameters that correlate with the remaining variation and using these to correct the residuals further. Kelly et al. (2010) and Sullivan et al. (2010) found evidence of such a correlation between the host galaxy mass of supernovae and their Hubble residuals. Because of this, it is now standard to see supernova cosmology analyses include a correction term in their distance moduli calculations to account for this correlation. It has also inspired the search for other corrections to supernova magnitudes stemming from the supernova environment, like examinations of the global (entire galaxy) or local specific star formation rate (Rigault et al., 2013, 2015).

Some of these additional parameters are measured from the supernovae themselves, rather than their environments. A number of studies have searched for correlations between supernova spectral features and their absolute luminosities. Nugent et al. (1995) first outlined the existence of a spectroscopic sequence that relates the ratios of the depths of the Si II $\lambda 6355$ and Si II $\lambda 5972$ lines and the ratios of the Ca II H&K lines to the absolute magnitude. Bailey et al. (2009) used the ratio of the absolute fluxes at two specific wavelengths to improve the standardization beyond what was possible with the Tripp standardization. Other studies have attempted to subclassify Type Ia supernovae using spectral indicators like the width of

the Si II $\lambda 6355$ and Si II $\lambda 5972$ lines (Branch et al., 2006), or the ejecta velocity as measured by the blue shift of the minima of these lines (Wang et al., 2009; Foley & Kasen, 2011; Foley, 2012). By using different corrections for these subclasses, these studies have shown some improvements to the standardization precision.

An alternate approach is to create more flexible light curve or spectral models. This is the tack taken by the SNEMO models (Saunders et al., 2018). SNEMO extends the logic of SALT2 by modeling the full spectral time series evolution of SNe Ia fluxes with additional components beyond the light curve shape captured by SALT2’s x_1 parameter. The SNEMO flux model is

$$f(\lambda, p) = c_0 \left[e_0(\lambda, p) + \sum_{i=1}^k c_i \times e_i(\lambda, p) \right] \times \exp [A_s \times \text{FM07}(\lambda)] \quad (1.16)$$

where FM07 is the Fitzpatrick & Massa (2007) dust extinction law. The functional form of this equation is quite similar to that of Equation 1.13, however the number of linear components, k , is not fixed to one. SNEMO encompasses three separate models, each varying in the number of linear components that are used, and each intended for different uses. SNEMO2 ($k = 1$) is useful as a comparison to the typical light curve shape and color models like SALT2, as the training methodology is slightly different from that of SALT2. SNEMO15 ($k = 14$) was introduced as a model that captures as much of the spectral variation that exists in Type Ia supernovae without overfitting. The SNEMO7 model is intended as an intermediate model that can best standardize SN Ia magnitudes, using an extension of the Tripp relation:

$$\mu = m_B - \left(M + \beta A_s + \sum_{i=1}^k \alpha_k c_k \right) \quad (1.17)$$

The resulting dispersion with this parametrization is approximately 0.113 mag, of which 0.097 magnitudes are not explained by the measurement error. Thus, by extending the model used to describe supernova spectral variation, SNEMO is able to improve the standardization precision.

The SUGAR model presented in Léget et al. (2020), effectively combines both approaches – using spectral feature measurements as an alternate parametrization of a full spectral time-series model. Rather than performing a principal component decomposition of the full, interpolated, spectral time-series data (like SNEMO), SUGAR first measures a series of spectral indicators in the near-maximum spectra. Then, a PCA decomposition of these indicators is used to obtain a 4-dimensional projection vector corresponding to each supernova in the data set. Finally, the full spectro-temporal model is created by fitting for basis functions like the $e_i(\lambda, p)$ of SNEMO that, when linearly combined with the spectral indicator projections, accurately predict the spectral evolution of the supernovae. The SUGAR model does not yet have a corresponding standardization model, so we do not yet have an estimate of how well we can correct the brightnesses of supernova for the effects captured by the SUGAR model. However, the model is able to capture much more of the spectral variation in SNe Ia than SALT2, without a large increase in the number of parameters in the model.

So far, all of the methods discussed have been parametric and model both the spectral evolution (besides color) and the relation between SN parameters and SN absolute luminosities linearly. The work of Fakhouri et al. (2015) takes a non-parametric approach, directly comparing the spectra of supernovae to identify pairs of “twins,” i.e. supernovae with nearly identical spectral time-series data, up to a difference in host-galaxy dust extinction and apparent magnitude. They found that the best pairs of twins differed in absolute magnitude on average by only 0.083 ± 0.012 mag (and by only 0.072 ± 0.010 mag after accounting for peculiar velocities). This implies that a direct comparison of the spectra, without any intermediate parametrization and modeling of necessary corrections, could produce better standardization.

Boone et al. (2020a) sought to understand the mathematical structure of the “twinness space” amongst near-maximum-brightness spectra. First, they estimated the aspects of the spectral variation that were due to differences in dust extinction and the overall brightness difference. Then, using the at-max spectra corrected for the differences in brightness and extinction, they applied the Isomap algorithm (Tenenbaum, 2000) to find a low-dimensional representation of the remaining spectral variation that approximately preserves the twins distances from Fakhouri et al. (2015). The low-dimensional embedding faithfully reconstructs many of the previously discussed spectral indicators of supernova diversity. Furthermore, Boone et al. (2020b) presents an additional model that predicts the variation in a supernova’s brightness based on its location in the “twins embedding.” Using this technique, they were able to standardize the supernovae in the sample to within 0.084 ± 0.007 mag, a level comparable to the best twins comparison of Fakhouri et al. (2015), but with a parametric approach.

1.5 Open Questions in Supernova Cosmology

Optimal Parametrization of SNe Ia

We could see in the previous section that there has been much work put into the development of empirical models of Type Ia supernova spectral variation and evolution, including linearly parametrized models of the full spectral evolution (SALT2, SNEMO, SUGAR), non-parametric comparisons between at-max spectra (twins), and nonlinear parametrizations of at-max spectra (twins embedding). Each of these models attempts to find a low dimensional representation of the extremely complex spectral and temporal evolution of stellar explosions. However, it is not yet clear precisely how this dimensionality reduction should be accomplished.

Rubin (2020a) addresses one aspect of this question: estimating the optimal dimensionality of the latent parameteric space. Using counting statistics of the twin pairings found in Fakhouri et al. (2015), along with the insight from geometry that volumes concentrate on surfaces, Rubin argues that the ideal parameteric SN Ia model has about 3-4 parameters, excluding color. This number squares with the findings of Boone et al. (2020a), which found

that a non-linear 4-dimensional parameter space describes the near-maximum brightness spectral variation of SNe Ia, as well as with the SUGAR model, which found that 4 parameters were sufficient to model most of the variation in near-maximum spectral indicators.

This is however smaller than the 7 parameters of SNEMO7, which was found to best standardize supernovae when assuming a linear relationship between these parameters and the absolute magnitudes of supernovae, and the 15 parameters of SNEMO15 that were found to best capture the full diversity of the supernova spectral behavior. Having such high-dimensional models can also prove problematic when we don't have high quality spectrophotometry – Rose et al. (2020) found that it is difficult to constrain even 7 parameters with currently available photometric measurements. SUGAR's ability to capture most spectral variation with four parameters suggests that changing the way we determine the linear basis we use to describe supernovae may be of use in improving standardization using data that is already in hand. However, there is not yet a standardization model for SUGAR, i.e. a mapping from a supernova's location in parameter space to its absolute brightness, so it is not yet clear if this alternate basis is effective in the final standardization step.

The fact that the twinning techniques of Fakhouri et al. (2015) and Boone et al. (2020b) can standardize brightnesses so precisely may point to the idea that a non-linear standardization model may be part of the solution to the puzzle of the ideal description of Type Ia supernovae for cosmology. Some evidence for this non-linearity has already been seen; Rubin et al. (2015) found a preference for a piecewise linear color-luminosity relationship, and Kim et al. (2013) found that using a Gaussian process regression of light curve parameters can in some circumstances give better standardization results.

Additionally, the fact that the twins studies achieve such good standardization using only single spectra near maximum brightness may indicate that there is no need to observe the full spectral evolution (or even broadband evolution) of Type Ia supernovae to standardize them; all of the necessary information could be encoded in a single spectrum. However, obtaining these spectra may be prohibitively expensive, and it may be difficult to properly schedule observations to ensure that the spectra obtained are within the window of phases that enable these techniques. More work is needed to understand the trade-offs involved in using these alternate methods.

Each of the empirical models that have been developed have their merits and their pitfalls. There remains much work to be done in fully comparing the models to one another, in understanding how each model performs when using different types of data with varying quality, and in using the lessons learned in the construction and study of these models to inform new models. Some of this work will be presented here.

Population Drifts with Redshift

Many of the standardization methods we have discussed make the implicit assumption that the corrected magnitudes of SNe Ia are identical regardless of redshift, and that any unmodeled variation is due to processes that remain stable across redshift. These assumptions may not necessarily hold, if, for example, the standardized magnitude is related to the metallicity

of the supernova progenitor star, or if dust properties change throughout cosmic history. If these trends are not recognized and modeled, we will have a biased Hubble-Lemaître diagram, and thus have biased estimates of cosmological parameters.

It is imperative that when we are designing future surveys (like LSST or the Roman Space Telescope), we are prepared to catch and correct for this type of drift. Using the twins method of Fakhouri et al. (2015) may allow us to avoid the issue entirely, since that technique uses direct comparisons rather than modeling corrections to the observed magnitudes. However, as we have emphasized, using this technique requires high quality spectrophotometric observations of SNe Ia. There may still exist more efficient means of determining the twin-like subclassification of SNe Ia so that we can use them in this type of like-to-like direct measurement of relative distances. Alternatively, we can use existing parametric models to simulate the types of data that we could obtain from these future surveys, given the relevant instrument and observing parameters. Using more flexible models can allow us to get a better sense of how sensitive these future measurements will be to drifting population parameters, and, in so doing, better inform our survey design decisions.

Hubble Constant Discrepancy

In addition to understanding the evolution of the rate of expansion of the universe, we would also like to quantify its current value, the Hubble constant, H_0 . Type Ia supernovae cannot be used directly for this measurement, as they are relative distance indicators, not absolute distance indicators. However, more fundamental distance calibration measurements are only possible using very nearby objects, where the distance-redshift relation is dominated by peculiar velocities. To overcome this issue, we can use the nearby fundamental measurements to calibrate a series of increasingly distant relative distance indicators until we reach the Hubble flow, where peculiar velocities are no longer dominant. This technique is usually referred to as the distance ladder.

The value of the Hubble constant can also be determined by using measurements of the cosmic microwave background (CMB) to calibrate the length scale of fluctuations in the matter density in the universe. This length scale is embedded in both the CMB and the visible matter density due to a preferred length scale in acoustic waves in the primordial plasma of the universe, (baryon acoustic oscillations, or BAO).

Riess et al. (2016) and Riess et al. (2019) used this first method to measure H_0 , taking a geometric distance to the Large Magellanic Cloud (LMC) determined by observations of detached eclipsing binaries to calibrate the period-luminosity relationship of Cepheid variable stars, and using the distances determined from Cepheids to calibrate Type Ia supernovae. They found $H_0 = 74.03 \pm 1.42 \text{ km s}^{-1} \text{ Mpc}^{-1}$. However, using CMB measurements and assuming a Λ CDM cosmology, Planck Collaboration et al. (2016) found a value of $H_0 = 67.27 \pm 0.60 \text{ km s}^{-1} \text{ Mpc}^{-1}$. There is thus a 4.4σ discrepancy between the value of the Hubble constant measured using probes of the late universe (geometric distances, Cepheids, supernovae) and the value measured using probes of the early universe (CMB, BAO).

Some additional methods for determining the value of H_0 have been introduced, including using stars at the tip of the red giant branch in the Hertzsprung-Russell diagram in a galaxy to determine a distance to that galaxy (Freedman et al., 2020), using gravitationally-lensed quasars (Bonvin et al., 2017), or combining simultaneous observations of electromagnetic and gravitational wave to obtain distances from so-called “standard sirens” (Holz & Hughes, 2005). None of these techniques, however, have yet been able to conclusively resolve this dispute.

The tension in the values of H_0 may suggest the existence of new physics beyond the Λ CDM paradigm. However, it may also simply suggest the existence of a systematic bias in the techniques used to make these measurements. Understanding the systematic errors inherent in the Type Ia supernova standardization process and analysis is therefore necessary to fully understand and resolve this tension.

1.6 Dissertation Outline

In this dissertation, we present work that addresses some of these outstanding questions approached from a number of different angles. Each of these questions and studies may seem disparate, but they are all on some level connected by a larger theme: the need to identify, quantify, and mitigate systematic errors and potential bias in upcoming supernova surveys. In the age of large supernova surveys like the Rubin Observatory and Roman Space Telescope, the number of supernovae observed will be large enough that the statistical error on our distance estimates will be subdominant to the error stemming from systematic biases. Therefore, identifying the sources of bias, whether from detector-level effects (Chapter 2) or final analysis techniques (Chapter 5), is essential to ensuring the continued success of supernovae as probes of cosmic expansion. There is also a need to understand the impact that these errors can have on our final measurements after including observational effects like spectral resolution, temporal cadence, or signal-to-noise ratio. This motivates both an exploration of how much information can be extracted from different types of observations (Chapters 3, 4, and 6), as well as the creation of simulations that are representative of the wide range of behaviors that have been studied in SNe Ia (Chapter 4 and the later portions of Chapter 6).

Chapter 2 focuses on a measurement of a particular detector-level effect (charge transfer efficiency, or CTE). This effect is caused by “traps” in the silicon lattice of the charge-coupled device (CCD) detectors that are ubiquitous in optical astronomy. These traps delay the readout of the photoelectron signal in the detector array leading to visible trails in the resulting images. The smearing from charge transfer inefficiency is particularly problematic in spectroscopic contexts, as the trails often align with the dispersion axis of the images that are processed into spectra, confusing this detector smearing with, for example, physical spectral feature broadening. In this chapter, we use two different techniques to quantify the CTE in each of the detectors on the SuperNova Integral Field Spectrograph (SNIFS), the instrument that collected much of the data used in the remaining chapters.

Chapter 3 investigates a particular spectral region of Type Ia supernovae (encompassing the Si II $\lambda 5972$ and Si II $\lambda 6355$ features) near maximum brightness and presents a model that is able to extract accurate measurements of the location (velocity) and size (equivalent width) of these spectral features from spectra with lower signal-to-noise ratios and lower resolution. This region of the spectrum is frequently used in some of the subclassification schemes of SNe Ia mentioned in Section 1.3, and can also serve as a proxy for identifying changes in populations of certain subtypes of SNe Ia with redshift. Enabling the use of lower quality spectra is key to monitoring potential population drifts (and therefore mitigating systematic bias) at higher redshifts, allowing us to probe even earlier eras of cosmic history.

We investigate a more general approach to a similar problem in Chapter 4. There, we examine how well two existing linear empirical models of Type Ia supernova evolution (SALT2 and SNEMO) can capture a variety of near-maximum spectroscopic features (including the region studied in Chapter 3). We also provide a model for producing realistic fake spectra. The former study aims to begin exploring how well the inclusion of additional linear components to spectral models can capture non-linear features like ejecta velocities. The latter portion enables future studies that use spectral templates that capture the full range of supernova spectral behavior.

Chapter 5 addresses a general statistical problem that appears in a number of supernova standardization analyses. Supernova analyses will perform a fairly standard linear regression (see Equation 1.14), using a few covariates (e.g. light curve parameters) to predict some target values (e.g. absolute magnitude). Then, they will perform a second regression, using an additional covariate (e.g. host galaxy stellar mass) to predict the residual from the initial regression. We show that this practice is statistically sound if the covariates in the initial regression are not correlated with the covariates used in the second regression. However, these correlations do frequently exist in the studies that use this analysis workflow, and thus their results may be biased. We calculate closed-form solutions for the magnitude of these biases for a toy model of the problem, and also measure the size of this bias in the context of supernova standardization using publicly available data sets.

Finally, in Chapter 6, we present two new models of Type Ia supernova spectroscopy that make use of deep learning techniques, which we name `spec2embed` and `embed2spec`. Both of these models can be viewed as temporal extensions of the “twins embedding” models presented in Boone et al. (2020a) and Boone et al. (2020b). The `spec2embed` model uses single spectra, observed at any phase from -10 to +40 days after maximum brightness, to accurately predict both the phase of the spectrum and the location of the supernova that the spectrum came from in the twins embedding space. This allows us to perform comparably accurate standardization, and hence distance determination, using a wider range of spectra. Aside from extending the usefulness of the spectral embedding found by Boone et al. (2020a), the success of the `spec2embed` model in predicting the embedding coordinate across phases may also suggest that the spectral information content that determines the absolute magnitude of a Type Ia supernova is not restricted to observations near maximum brightness. The `embed2spec` model reverses this process, predicting the spectrum of a supernova given its location in the twins embedding space and its phase. This model enables a forward-modeling

approach to fitting, allowing us to constrain the location of a newly observed object using multiple spectra, spectra with lower spectral resolution, or even broadband photometry. We conclude with some suggestions for future studies that apply these new models.

Chapter 2

Measurements of Charge Transfer Efficiency in the SNIFS Detectors

2.1 Introduction

Charge-coupled devices (CCDs) are ubiquitous in optical and infrared astronomy. Incident photons generate electrons in the bulk silicon of the CCD, and these photoelectrons are then collected by a grid of pixels arranged on a series of parallel readout registers. By manipulating the voltages of the gates that separate the pixels of these registers, we can shuffle the collected charge row-by-row into a additional register, known as the serial register. Similar voltage manipulations of the serial register gates produce an output signal that can be amplified and digitized. The charge shuffling procedure alternates between the parallel and serial registers until all pixels have been read out, and an image can then be reconstructed from the resulting signal.

Ideally, 100% of the charge in each pixel would be transferred to the next pixel at each readout step. However, traps in the silicon lattice of the CCD can capture charge only to release it at a later time. The trapped and released charge leads to smearing of point sources along the direction of charge transfer and impedes our ability to get accurate photon counts. The magnitude of this effect is quantified by the charge transfer efficiency (CTE) of the CCD, defined as the fraction of charge that survives each pixel transfer. Because most of the CCDs used in astronomical applications make many transfers in each image (as they have thousands of pixels in each register), the CTE must be very close to unity, with typical values being around $1 - 10^{-6}$. Additionally, because a larger number of transfers results in a higher likelihood of encountering more traps, pixels that are further from the readout register are more affected by charge trapping. This effect is especially problematic for spectroscopic instruments since the increased smearing with distance to the amplifier can lead to effects like uneven broadening of spectral features.

In this work, we quantify the charge transfer efficiency of the CCDs in all channels (photometric, and blue and red spectroscopic) of the SuperNova Integral Field Spectrograph

(SNIFS). SNIFS is the main instrument used by the Nearby Supernova Factory (Aldering et al., 2002, SNfactory,) project, which was designed to spectrophotometrically observe Type Ia supernovae at redshifts around $z = 0.07$. The instrument is a two-channel integral field spectrograph, mounted on the University of Hawai'i 2.2m telescope on Maunakea. A dichroic mirror splits the light into the red and blue spectroscopic channels, covering the wavelength ranges of 3200–5200 Å and 5100–10,000 Å, respectively. Two microlens arrays split the 6" x 6" field of view into 225 samples of 0.4" x 0.4". The light eventually reaches the detectors. For the blue channel, the detector is a 2048 x 4096 pixel thin CCD, while for the red channel the detector is a 2048 x 4096 pixel deep-depletion CCD. In addition to the spectroscopic channels, there are also the guiding and photometric channel of the instrument. This channel is located at the focus of the telescope, where we have two deep-depleted CCD detectors – one used for guiding the telescope, and the other used in conjunction with various filters for photometric observations. Each of the CCD detectors in the SNIFS instrument have 2 amplifiers, which we label A and B in the spectroscopic channels. The two detectors of the guiding and photometric channels are treated as a single detector in the read out software, so we label the amplifiers of the first detector A and B, and call the amplifiers of the second detector C and D.

We use two techniques to measure the CTE in all of these detectors. The initial characterization of the SNIFS CCDs is done by first exposing them to a Fe⁵⁵ x-ray source in order to generate single-pixel events with a known energy spectrum and then measuring the photo-electron loss directly. Later, we perform a similar analysis using cosmic ray events extracted from the dark frames taken during each SNIFS observing run. These in-situ measurements of the CTE allow us to track changes in the efficiency over time.

2.2 Initial Characterization with Fe⁵⁵ X-rays

A standard method for measuring CTE involves exposing the CCD to photons with a known energy spectrum (Janesick, 2001). X-ray sources are high enough energy to generate many electron-hole pairs in a small volume in the CCD, typically resulting in a detectable event in the image that takes up a single pixel. Because we know the energy spectrum of these events, we can easily calculate the expected distribution of the number of electrons in these single-pixel events. By plotting the amount of charge measured in each event as a function of the row number, we can quantify the extent of charge trapping by measuring the decrease in charge with distance from the amplifier.

A common source for these characterizations is Fe⁵⁵, which produces strong lines from K_{α} and K_{β} emission at 5.9 keV and 6.2 keV. These emission lines corresponding to photoelectron signals in the CCD of 1620 e⁻ and 1778 e⁻, respectively, as the amount of energy necessary to produce an electron-hole pair in silicon is approximately 3.65 eV in the regime where the photon energy is greater than approximately 10 eV.

Each of the SNIFS CCDs were exposed to a Fe⁵⁵ source a number of times, producing images like the one shown in Figure 2.1. Using `sep` (Barbary et al., 2015a), a Python

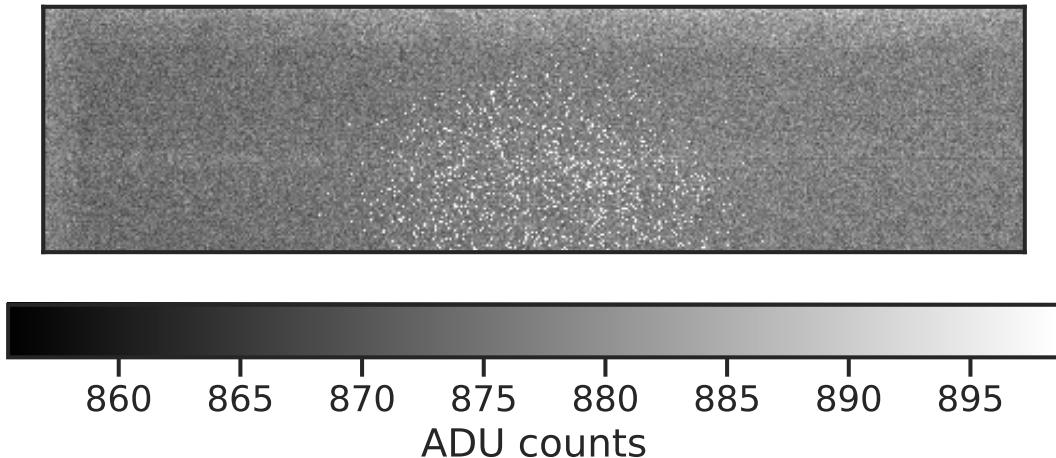


Figure 2.1: Example X-ray image for a single amplifier.

implementation of the commonly-used **SExtractor** package (Bertin & Arnouts, 1996), we selected all events in the resulting images that had signal levels $> 5\sigma$ above the background signal level determined with a filter size of 32 x 32 pixels. We then fit the histogram of digitized counts in these extracted events, binned into 10 count bins, to a simple double-Gaussian model of the Fe⁵⁵ spectrum:

$$s = A_\alpha \exp\left(-\frac{(f - 1620/g)^2}{2\sigma_\alpha^2}\right) + A_\beta \exp\left(-\frac{(f - 1778/g)^2}{2\sigma_\beta^2}\right). \quad (2.1)$$

This fit is done in order to measure the gain, g , of the CCD amplifier, allowing us to convert from ADU counts to electrons. The relative fraction of events from the K_α and K_β emission (A_α/A_β), as well as the widths of the emission lines (σ_α and σ_β) are allowed to float, and we treat these parameters as nuisance parameters in the fit to obtain the gain.

Using these gain measurements to convert the ADU counts to the number of electrons N_{e^-} , we are able to combine the events from several images per channel (22 for the blue channel, 26 for the red channel, and 8 for the photometric channel). We select the events that correspond to K_α emission with a simple cut, choosing events with $1550 < N_{e^-} < 1700$. We then plot the flux for each event as a function of distance from the amplifier, shown in Figure 2.3 for the parallel registers and Figure 2.4 for the serial register. The slope of the best-fit line (in units of e^- per transfer), divided by the expected number of electrons (1620), gives us our measurement of the charge transfer *inefficiency*, the number of electrons lost per transfer ($CTI = 1 - CTE$). The resulting measurements of the charge transfer inefficiency are shown as labels in Figs. 2.3 and 2.4, as well as in Table 2.1. All of the parallel values are on the order of 10^{-6} , i.e. about one electron in every million is lost in each transfer. The serial CTI values are slightly higher; this is acceptable because there are fewer transfers that need to be done in the serial direction.

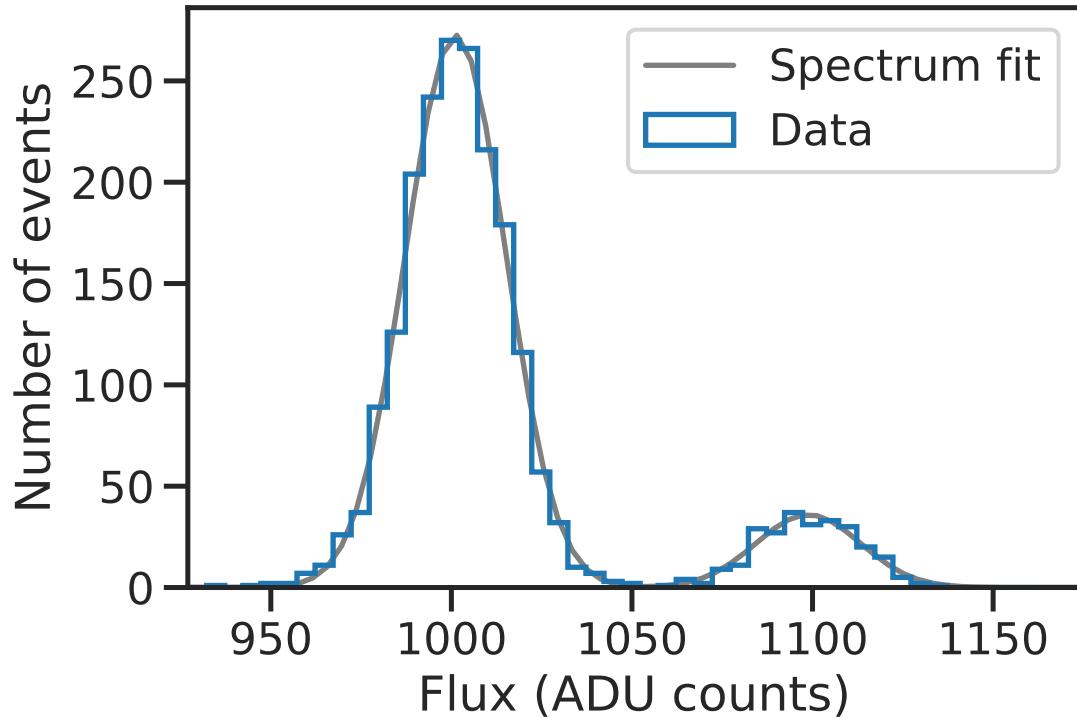


Figure 2.2: Example spectrum of the Fe^{55} X-rays events extracted from a single amplifier frame along with the best-fit model. The results of this fit are used to determine the amplifier gain from each frame.

Camera	Amp.	Parallel CTI	Serial CTI	Number of images	Number of events
B	A	0.783 ± 0.015	6.80 ± 0.03	22	15,925
	B	1.455 ± 0.017	2.99 ± 0.03		13,113
R	A	0.501 ± 0.013	9.94 ± 0.04	26	4,071
	B	0.609 ± 0.011	4.46 ± 0.03		4,415
P	A	2.696 ± 0.010	1.33 ± 0.02	8	15,308
	B	0.773 ± 0.008	0.93 ± 0.01		21,010
	C	1.111 ± 0.009	3.49 ± 0.02		14,498
	D	0.160 ± 0.008	3.66 ± 0.01		19,041

Table 2.1: Charge transfer inefficiency results from Fe^{55} X-ray characterization. All values are in units of 10^{-6} .

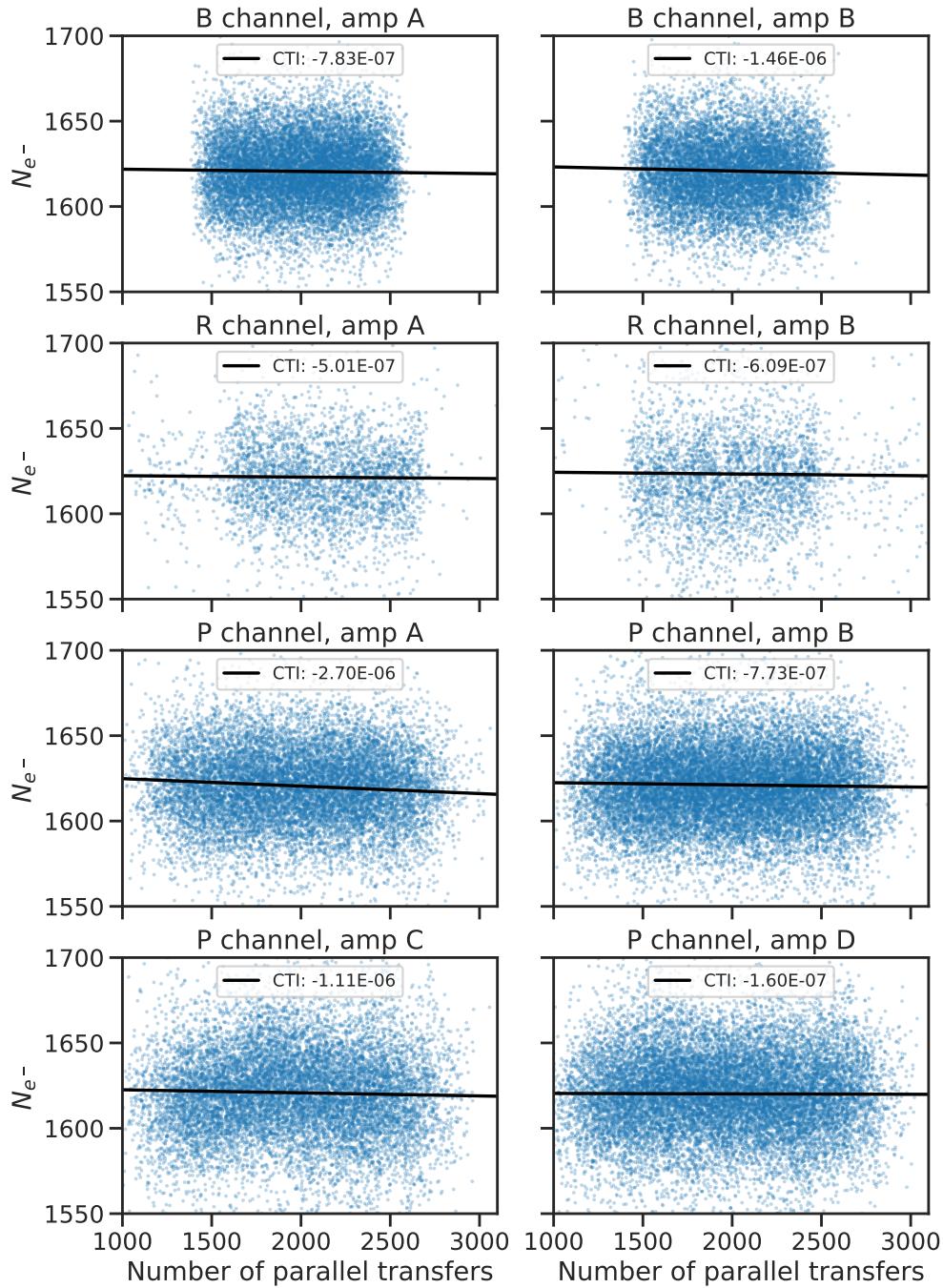


Figure 2.3: The number of electrons in extracted Fe^{55} X-ray events as a function of y-location on the CCD for each camera amplifier (equivalent to number of transfers in the parallel direction). The slope of the best-fit line divided by the expected number of electrons gives us an estimate of the charge transfer inefficiency in the parallel registers.

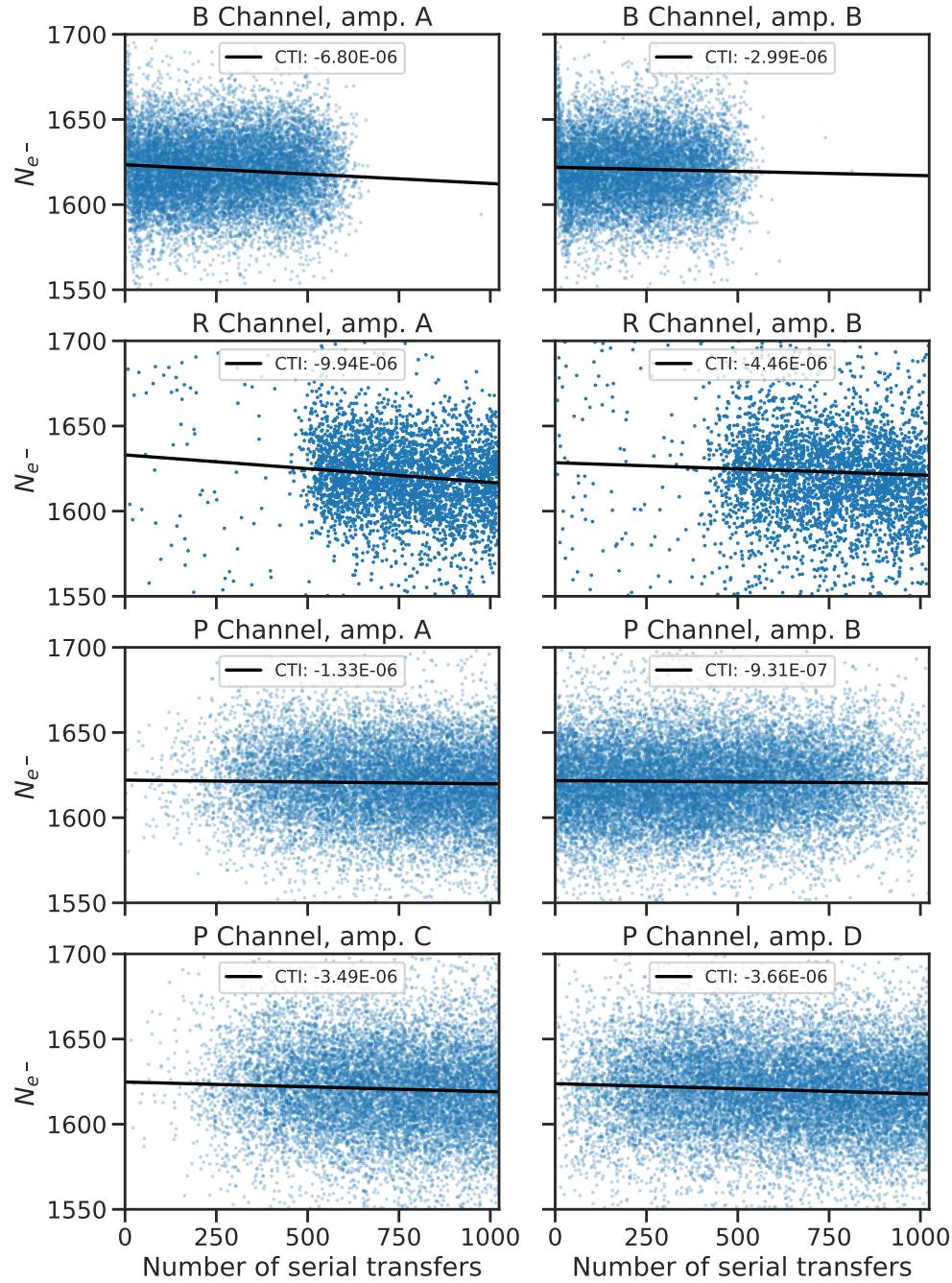


Figure 2.4: Same as Figure 2.3, but for the x-location of events. The best-fit lines are used to estimate the charge transfer inefficiency in the serial registers.

2.3 Cosmic Ray Measurement

The X-ray measurement allows for a precise determination of the CTE when we have physical access to the detector. However, we'd like to be able to track changes in the CTE with time in situ in order to quantify any degradation over the lifetime of the instrument. We can get such a measurement by making use of the cosmic ray hits found in the dark frames taken as part of normal observing procedure and measuring the average smearing of these hits due to charge transfer inefficiency, similar to the methodology presented in Riess et al. (1999).

Cosmic rays create small (approximate 6-7 pixels) events in CCD images. The shape of each individual event is driven by the incidence angle of the cosmic ray and charge diffusion (i.e. bleeding between pixels because of the structure of the CCD), but also by the CTI smearing effect. The first two effects are statistically symmetric about the highest pixel, so in principle we should be able to subtract the symmetric portions of the cosmic ray events, leaving only the asymmetric CTI trails.

We proceed very similarly to Riess et al. (1999). We restrict ourselves to dark frame images that had an exposure time of 1 hour in order to have a high enough count of cosmic ray events in the image. For each of these dark frames, we use `sep` to find cosmic ray hits, defined as the objects detected at $> 1.5\sigma$ over the background level, with a measured ellipticity < 0.2 and no flags raised by `sep`. The ellipticity cut serves to remove extremely oblique incidence events or coincident events from our sample, as both types of event add excess noise to the measurement. Example hits that pass these cuts are shown in Figure 2.5. Additionally, in order to avoid the noise potentially introduced by hits landing in the CTI trails of other nearby hits (see e.g. the bottom right example in Figure 2.5), we remove from our sample all pairs of events that are within 10 pixels of one another.

For each selected cosmic ray event, we subtract the value of the pixels further from the readout amplifier from the pixels closer to the readout amplifier in both the parallel and serial direction. We then calculate the fraction of charge that is left in the trail by summing the number of excess counts in the 5 trailing pixels and dividing that sum by the number of counts in the peak of the event. On average, this gives us an estimate of the fraction of charge lost to trapping.

To boost our statistics, we aggregate all of these measured charge loss fractions on a nightly basis. The individual values of the charge loss fraction are shown in blue in Figs. 2.6 and 2.7. These measurements are extremely noisy, so to reduce the noise, we group them in 16 bins along the direction of interest and take the median and normalized median absolute deviation (NMAD) of all of the measurements in each bin. These median values are shown in red in Figs. 2.6 and 2.7, as well as in Figure 2.8 and 2.9 in more detail. Finally, we fit a line to these median values weighted by the NMAD in each bin. As before, the slope of the line gives us a measure of the charge transfer inefficiency.

We made these measurements for every night's dark frames in order to check for time dependence. In Figure 2.10 we show the CTI in the parallel and serial registers as a function of time for each amplifier. Figure 2.11 shows the same data aggregated by year. We find that there is very little evidence of significant degradation over time. Indeed, linear fits to each of

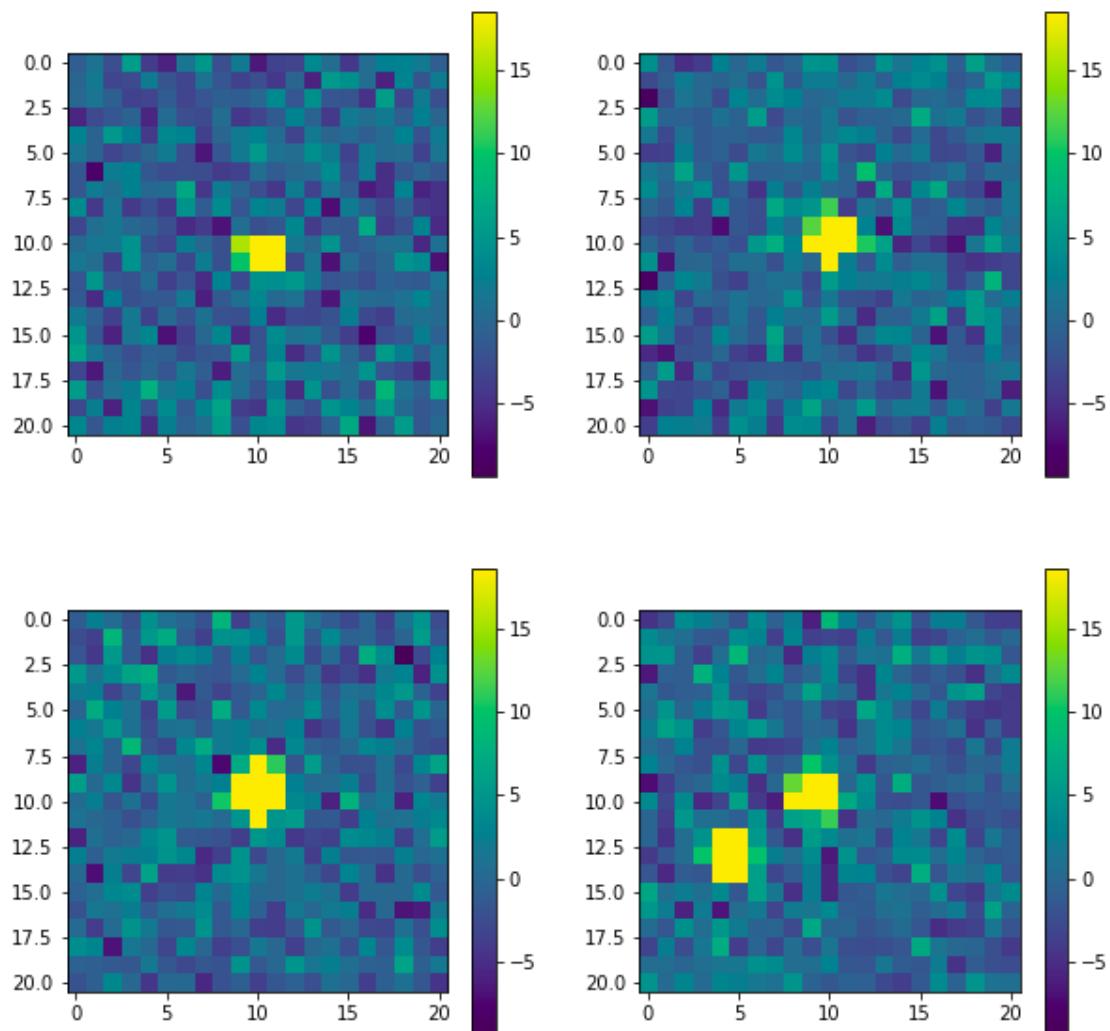


Figure 2.5: Example identified cosmic ray hits passing our ellipticity cut. The two neighbouring hits seen in the bottom right example would be removed from the final sample because they are too close to each other. This culling removes some of the noise from our final signal.

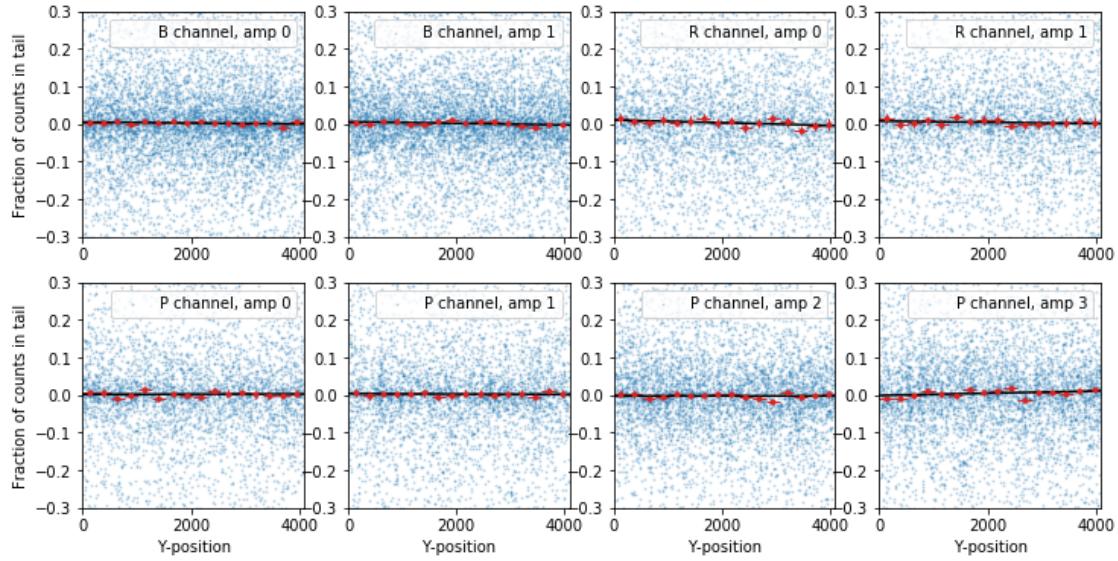


Figure 2.6: Example measurement of CTI in the parallel register from cosmic ray trails from a single night. All blue dots represent a single cosmic ray hit. The red points show the median fraction of counts in the peak of the hits that end up in the trails in each y-position bin. The best-fit line is also shown. The slope of this line gives us the CTI.

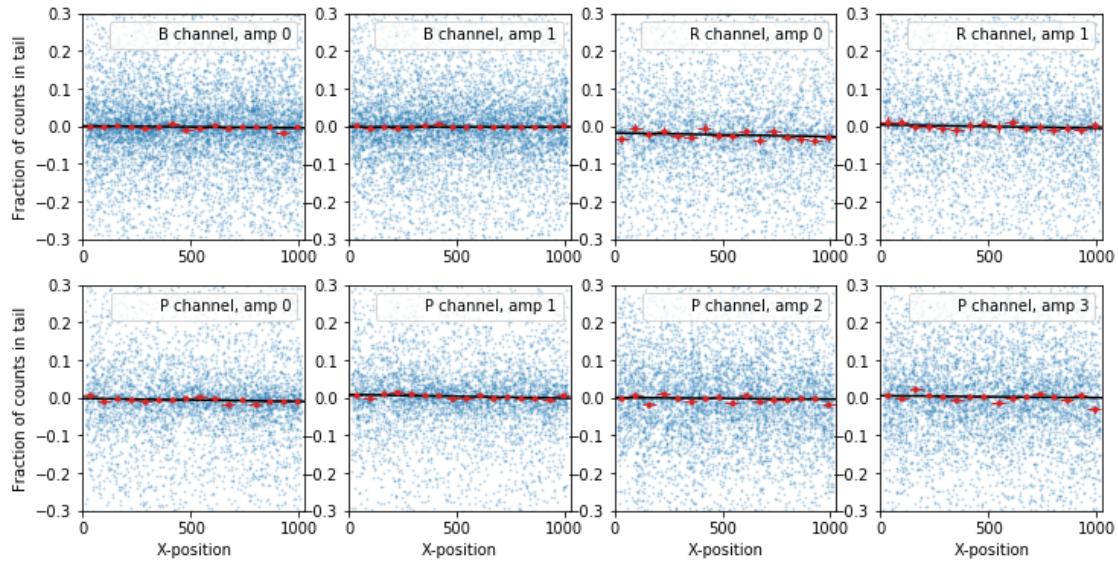


Figure 2.7: Same as Figure 2.6 but in the serial direction.

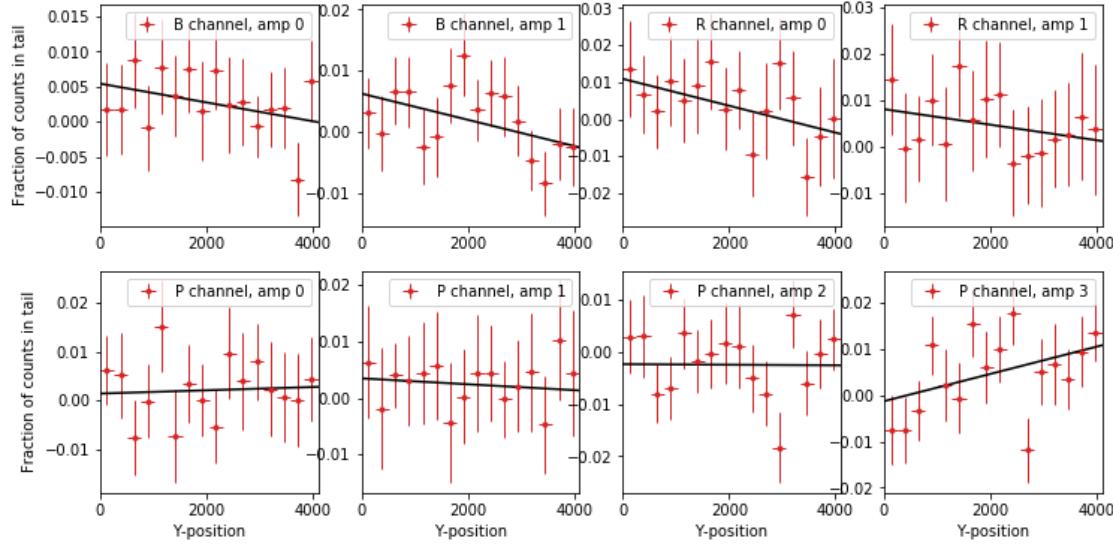


Figure 2.8: Same as Figure 2.6 but zoomed to show the median values and their associated uncertainties.

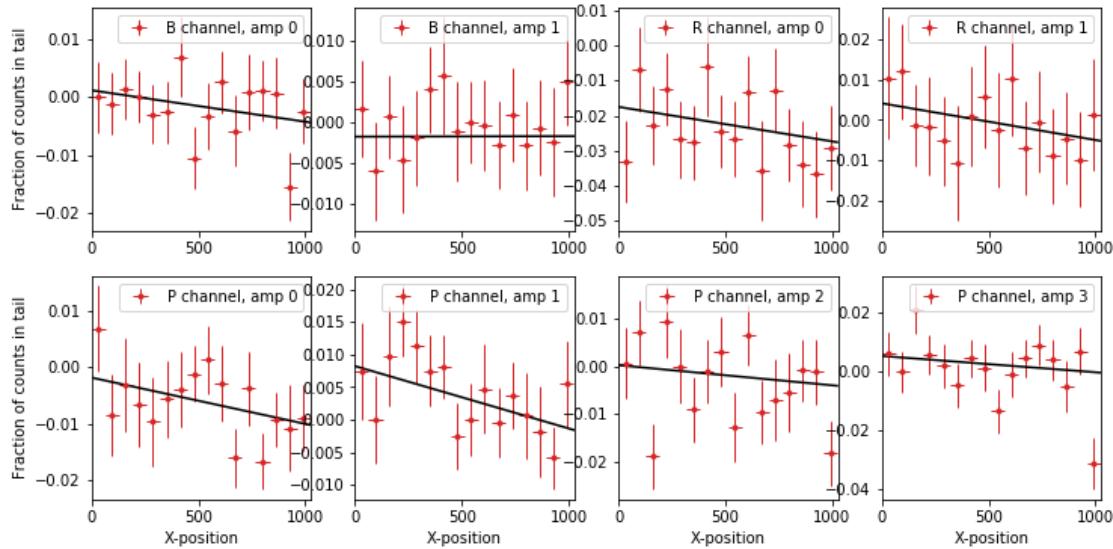


Figure 2.9: Same as Figure 2.8 but in the serial direction.

these yearly aggregated data sets have slopes consistent with zero. Finally, the median CTI measured in each amplifier using cosmic rays in dark frames are all summarized in Table 2.2.

The values are roughly consistent with what was found in the X-ray measurements, as all CTI values are still on the order of 10^6 , with the serial registers having higher CTI than the parallel registers. There are a number of potential reasons for the remaining differences between the measurements. One possible source of the difference in these two measurements could be that the X-ray measurement photons were focused on the center of each CCD, whereas the cosmic ray measurements are dispersed over the full chip. As a result, there were portions of the CCD that no X-ray photoelectrons passed through, so the effect of charge traps in these regions would be neglected by these measurements. There were also some challenges in identifying true single-pixel X-ray events, particularly in the deep-depleted CCDs (red spectroscopic and photometric channels), where the discrepancies are largest. The X-ray photoelectrons can be deposited deeper in the silicon of these CCDs, creating multi-pixel halos rather than the expected single-pixel events. Some of these fractional events may happen to be detected over our background threshold and thus be counted as a single-pixel event. This effect could also be exacerbated by event-crowding, where several fractional events are combined because they occur in neighboring pixels, and the combined events are misidentified as a single event. Overall, while the cosmic ray measurements are much noisier, as the measurement of the fraction of charge lost is much more subject to noise, these aggregated measurements are more reliable and more useful for tracking CCD performance over time.

Camera	Amplifier	Median Parallel CTI	Median Serial CTI
B	A	1.08 ± 0.04	0.96 ± 0.13
	B	0.97 ± 0.04	2.02 ± 0.13
R	A	1.49 ± 0.07	2.4 ± 0.2
	B	1.24 ± 0.07	2.3 ± 0.2
P	A	0.19 ± 0.18	5.1 ± 0.5
	B	0.18 ± 0.19	7.0 ± 0.4
	C	0.3 ± 0.2	5.4 ± 0.7
	D	0.5 ± 0.2	1.9 ± 0.5

Table 2.2: CTI measurements over time from all dark frames collected from 2006 to 2012. All values are in units of 10^{-6} .

2.4 Conclusion

We have presented two separate measurements of the charge transfer efficiency of the SNIFS CCD, first with an X-ray source in the laboratory, and later *in situ* using cosmic ray events

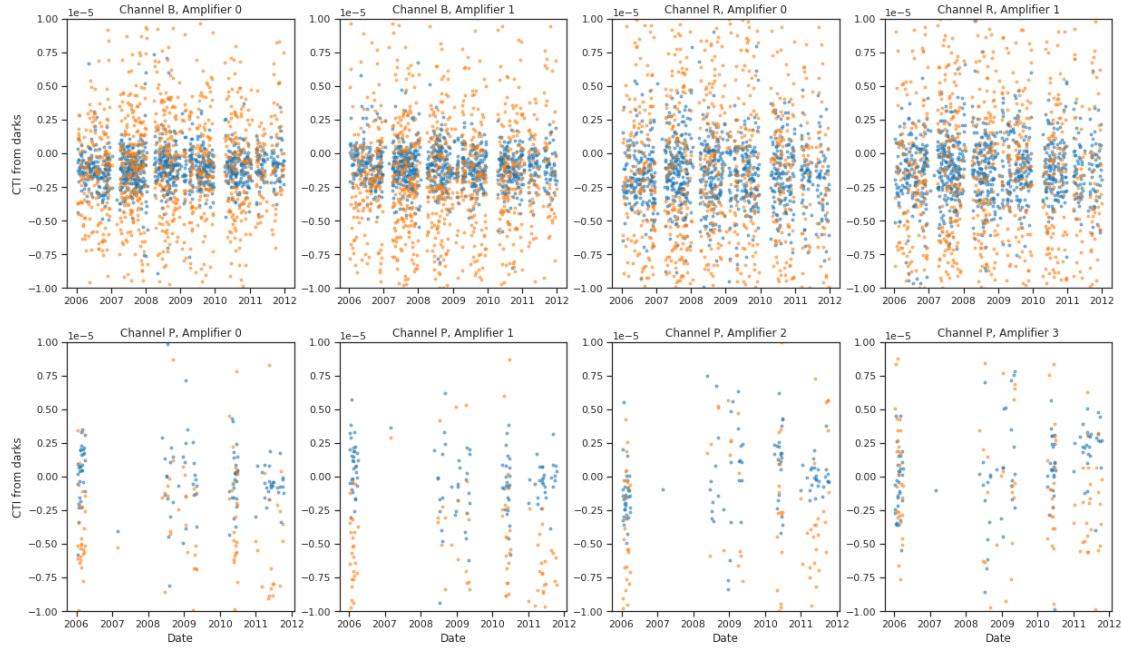


Figure 2.10: A search for potential time variation in the charge transfer efficiency of each of the spectroscopic cameras. CTI measurements in the parallel direction are shown in blue and those in the serial direction are shown in orange.

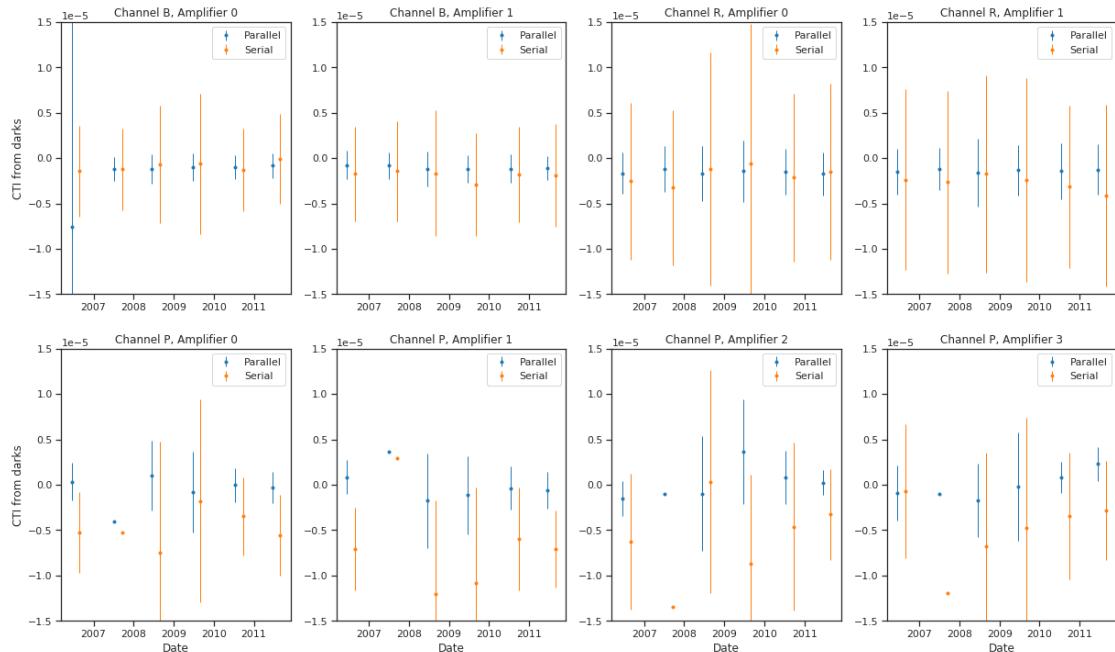


Figure 2.11: Same as Fig 2.10 but aggregated by year.

in dark frames to measure the average fraction of charge lost in readout. The latter measurement is noisier, but allowed us to track evolution of CTE over time. We see no evidence of such evolution. Our measurements are roughly consistent with one another, to within an order of magnitude, and both measurements show that the charge transfer inefficiency is sufficiently low for the detectors' use in spectroscopy. The code used in this work is publicly available at <https://www.github.com/sam-dixon/cte>.

Chapter 3

A Study of the Morphology of the SiII $\lambda 6355$ Feature in SNe Ia

3.1 Overview

A number of spectroscopic techniques have been shown to improve supernova standardization. The spectroscopic twinning technique presented in Fakhouri et al. (2015) and the twins embedding technique of Boone et al. (2020a) and Boone et al. (2020b) all show that the use of additional spectroscopic information can reduce the scatter in standardized magnitudes beyond the limit of photometric methods using only a single spectrum near maximum brightness. The extended spectrotemporal model SNEMO (Saunders et al., 2018) has also been shown to improve the spread in standardized magnitudes by capturing a wider array of supernova behavior through a more flexible model. Zheng et al. (2018) combines spectroscopic and photometric measurements to relate the rise time of the light curve and the photospheric velocity measured from a near-maximum spectrum to the peak magnitude of SNe Ia, and found a reduced dispersion in standardized magnitudes among normal velocity objects. Additionally, a number of subclassifications of SNe Ia based on their near-maximum brightness ejecta velocities, ejecta velocity time gradients, and equivalent widths of various spectral features have been introduced in the literature (Branch et al., 2006; Benetti et al., 2005; Wang et al., 2009, 2013). By splitting the supernovae into subgroups based on these spectral indicators, these studies have shown that the scatter in the intrinsic luminosities can be reduced. These studies all use one of the most prominent features of Type Ia supernovae spectra (indeed what separates Type Ia's from other Type I supernovae): the Si II $\lambda 6355$ feature, found at a rest-frame wavelength of about 6150 Å.

In addition to improving standardization, spectroscopy of the Si II $\lambda 6355$ region can be used to detect and probe systematic biases stemming from population drifts with redshift. As an example, we can take the empirical correlation between supernova ejecta velocity measured from the Si II $\lambda 6355$ line and intrinsic color that has been studied extensively (Wang et al., 2009; Foley & Kasen, 2011; Foley et al., 2011; Foley, 2012; Mandel et al.,

2014). Because of this correlation, uncertainties in velocity propagate into uncertainties in intrinsic color, which themselves propagate to uncertainties in distance modulus. If the distribution of velocities changes with redshift, this effect will lead to a bias when determining cosmological parameters.

While Foley (2012) found no significant difference in the distribution of silicon ejecta velocities between samples at low redshift and samples at somewhat higher redshifts ($0.1 < z < 0.4$), the author warns that “the high-redshift samples are still small, and even a small offset could affect cosmological measurements.” Future supernova surveys, like the Nancy Grace Roman Space Telescope supernova survey, will probe redshifts that are much higher than this previous work, and represent very different host galaxy age populations. Correlations between ejecta velocity and host-galaxy mass (Foley, 2012) and age and metallicity (Wang et al., 2013) suggest the possibility of a redshift evolution in ejecta velocity tied to these drifts in SN environments. In order to avoid these potential biases, it is imperative that we be able to measure the ejecta velocities of the supernovae in future surveys.

Typically, measuring any spectral indicator (i.e. velocities and equivalent widths) involves smoothing the spectrum to remove high-frequency noise and then removing some estimate of the continuum. The equivalent width of the line is obtained by integrating the measurements of the smoothed and continuum-divided spectrum in the region of interest, and the velocity of the line is determined from the blue shift of the wavelength of the minimum of the feature in the wavelength range of interest. The smoothing and minimum-finding method is very effective for spectral observations with high resolution and signal-to-noise, but exhibits some problems when the spectra have lower resolution or higher noise levels. One way of avoiding these types of errors is to assume some parametric form of the spectral feature shape (e.g. a Gaussian), but these parametric shapes often bias the results, as we will see.

In this chapter, we present a new method for reconstructing the velocity and equivalent width of the Si II λ 6355 feature of Type Ia supernova from low-resolution, noisy spectra that is more robust and less susceptible to bias. This method does not use the smoothing and interpolation techniques that can be problematic in the low-resolution regime, nor does it model the feature with mathematically convenient but physically unmotivated functional forms. Instead, the reconstruction is based on the available data, encapsulating much more of the morphological variation than a simple model would allow, without needing more expensive high-resolution observations. The recovered velocities and equivalent widths can be used with the previously mentioned improved standardization techniques, as well as to correct for possible redshift drift biases.

The chapter is structured as follows: the data used in this analysis is described in Section 3.2. In Section 3.3, we discuss the current measurement methods available for modeling the silicon absorption feature in medium- and high-resolution spectroscopy. Section 3.4 explains our new method, and Section 3.5 shows the results of this measurement method on an external validation set (BSNIP, Silverman et al., 2012). Section 3.7 evaluates the method on a simulated data set of supernovae at high-redshift observed with the proposed Roman Space Telescope prism. We conclude in Section 3.8.

3.2 Data

The spectra used in the training set were obtained with the Supernova Integral Field Spectrograph (SNIFS, Lantz et al., 2004) mounted on the south-bent Cassegrain port of the University of Hawai'i 2.2m telescope, operated remotely by members of the Nearby Supernova Factory collaboration (SNfactory, Aldering et al., 2002). The SNIFS integral field spectrograph uses lenslet arrays to divide its 6" x 6" field-of-view into a grid of 15 x 15 spatial elements (spaxels). Each spaxel is fed into a dual-channel spectrograph that covers 3200-5200 Å in the blue and 5100-10,000 Å in the red, with a spectral resolution of $R \sim 1000$. The data from this instrument was reduced using our dedicated pipeline, detailed in Ponder et al. (in preparation).

The SNfactory supernova sample¹ contains 275 objects with at least 5 spectral observations. These 275 objects have a total of 3731 spectro-photometric observations, with each object having an average of 13-14 spectral observations. The sample is subdivided into “good,” “bad” and “auxiliary” subsamples based on SALT2 light curve fits to synthesized photometry and spectral classification. All objects in the “good” category meet the following criteria:

- At least 5 total observations
- One epoch within -10 and +7 days of maximum brightness based on SALT2 fits to light curves generated through synthetic photometry
- 4 epochs between -10 and +35 days of maximum brightness

These quality cuts ensure that the supernova time series are detailed enough to properly constrain the light curve model parameters, particularly the time of maximum brightness. There are 223 objects in this “good” subsample, which we take as our full training set. Each of the spectra in this data set has been corrected for Milky Way dust reddening and shifted in wavelength to a redshift of $z = 0$. The overall flux has also been normalized so that each spectrum’s flux is as it would be if the supernova were located at a redshift of $z = 0.05$, assuming a fiducial flat Λ CDM cosmology with $H_0 = 70$ km/s/Mpc and $\Omega_m = 0.3$.

Our focus is on spectra near maximum brightness. Thus, we restrict the training set to those spectra observed within ± 2 rest-frame days of maximum light (as determined from SALT2.4 fits to the synthesized light curves). A total of 241 spectra from 163 objects meet this criterion, 127 spectra from 86 supernovae in the training subsample and 114 spectra from 77 supernovae in the validation subsample. When a single supernova has many observations within ± 2 days, we use the observation closest to maximum light. The ± 2 day window ensures that we can capture more supernovae into our training sample while also ensuring that the deviations from the maximal brightness spectrum are small.

Our analysis is also only focused on the relative sizes and shapes of the spectral features, and the overall flux and color calibration is irrelevant. To normalize the data, we divide

¹This analysis is based on the **ALLEG2a** internal production from the data reduction pipeline.

the spectrum by a spline fit to each spectrum with 13 evenly spaced knots from 2500 Å to 10,000 Å. This is the same normalization performed by the Supernova Identification code (SNID, Blondin & Tonry, 2007) and is performed to remove the effects of differential dust reddening in our line profile models. In Figure 3.1, we show an example of this spline fit and in Figure 3.2, we show what the Si II λ 6355 feature looks like with and without this normalization. This normalization has potential to affect the determination of the pseudo-continuum (defined later in Section 3.3), but this effect was found to have negligible impact on the measured velocity in our tests. The final spline-normalized spectra in the Si II λ 6355 region (5600-6600 Å) are made available in the code repository corresponding to this work.

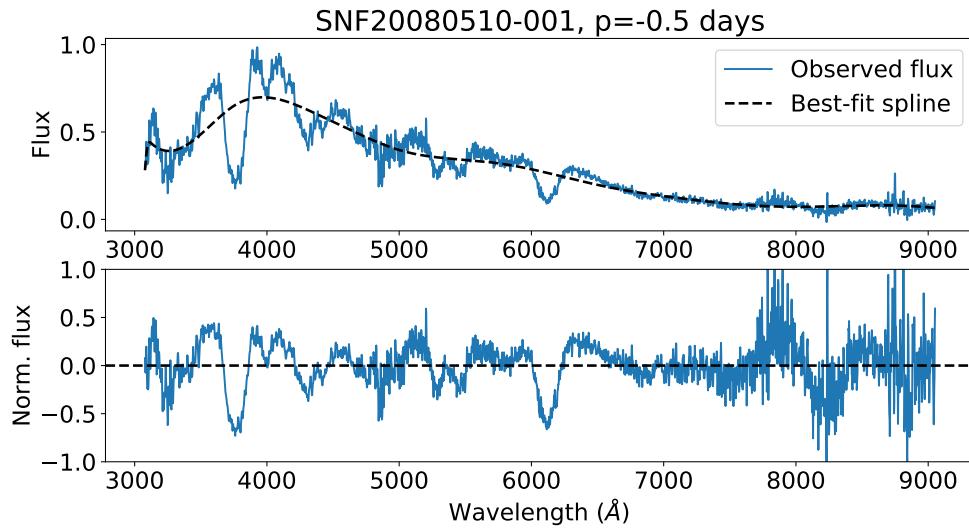


Figure 3.1: An example of the preprocessing steps taken to normalize the spectra using a 13-point spline. The upper figure shows the full near-max spectrum along with the best fit spline. The lower figure shows that same spectrum with the spline pseudo-continuum removed.

We also want to ensure that our model generalizes to unseen supernovae by evaluating its performance on an separate test set. This external validation set was taken from the Berkeley Supernova Ia Program (BSNIP, Silverman et al., 2012). From the 1298 spectra from 582 objects in this sample, we selected objects with a spectrum within ± 3.5 days of maximum brightness and exclude peculiar objects (i.e. those determined to be SN1991T- or SN1999bg-like by SNID). Once again, if one object has more than one spectrum within the phase range allowed, we select the spectrum closest to maximum brightness. This leaves us with a set of 88 spectra. We performed the same preprocessing to these spectra as we did the SNfactory sample spectra. The preprocessed line profiles are also available in the released code repository.

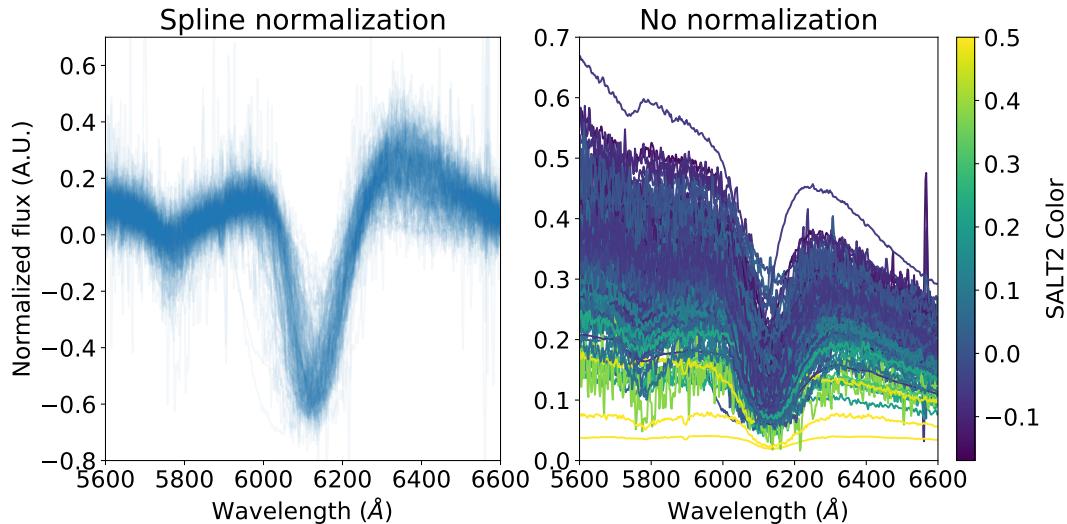


Figure 3.2: Zoom-in on the feature of interest for this work, with and without the spline normalization. The features plotted in the right figure are colored by the objects' SALT2 color to emphasize that the effect of this spline normalization is to remove the effects of color on the shape of the spectral feature.

3.3 Spectral Feature Measurement

As photons from radioactive activity in the inner layers of the supernova explosion make their way through the outer ejecta, they are absorbed by the outer material. This absorption results in the characteristic features in the supernova spectrum. The shape of these absorption lines is governed by myriad physical factors, including the velocity, temperature and density of the ejecta and the optical depth of the various layers of the explosion.

Supernova spectral features profiles are usually quantified using measures of their width and depth, along with their location in the spectrum. The width and depth of the line are summarized by the equivalent width: the width of a rectangle with a height determined by the flux of the continuum such that the area of the rectangle equals the area of the line flux under the continuum. Mathematically, this is

$$pEW = \int_{\lambda_b}^{\lambda_r} \left(1 - \frac{f(\lambda)}{f_c(\lambda)}\right) d\lambda, \quad (3.1)$$

where λ_b and λ_r are the wavelength limits of the feature on the blue and red sides, respectively, $f(\lambda)$ is the true flux of the object and $f_c(\lambda)$ is the flux of the continuum.

Calculating the continuum in supernova spectra is challenging, since the absorption features are actually blends of multiple wide lines. Instead, we define a pseudo-continuum by the line connecting the maximum flux values in predefined windows on either side of the absorption feature (see Figure 3.3). These windows are defined in Table 3.1. Throughout

the rest of this work, we will use the pseudo-equivalent width, i.e. where f_c in Eq. 3.1 is the pseudo-continuum.

Parameter name	Value
λ_0	6355
λ_{min}	5600
λ_{max}	6600
λ_b	5850-6015
λ_r	6250-6365

Table 3.1: Important wavelengths for measuring indicators of the Si II $\lambda 6355$ feature. λ_b and λ_r are the blue and red windows used when defining the pseudo-continuum.

We can also learn about the ejecta velocities from the location of the absorption feature. We measure the wavelength of maximum absorption λ and calculate the velocity using the relativistic Doppler formula:

$$v = c \left[\frac{(\lambda/\lambda_0)^2 - 1}{(\lambda/\lambda_0)^2 + 1} \right] \quad (3.2)$$

where c is the speed of light and λ_0 is the emission wavelength of the feature (Table 3.1). Because this line is always blue-shifted (the visible ejecta in the line of sight are moving toward the observer), we neglect the minus sign and refer to lines that are more blue-shifted as representing higher velocity ejecta.

Baseline: Savitsky-Golay Smoothing

We begin by establishing a ground truth measurement of the velocities and equivalent widths of all of the supernovae in our training and validation sets. We start by smoothing the spectrum with a Savitsky-Golay filter. The window for this filter is determined optimally as described in Appendix A. An example smoothed spectrum is shown in Figure 3.3.

Using this smoothed spectrum, we search for the wavelength of maximal absorption (minimum flux) within the window defined by the reddest edge of the blue pseudo-continuum window and the bluest edge of the red pseudo-continuum window and use this wavelength in Eq. 3.2. We also use the smoothed spectrum to calculate the pseudo-equivalent width with 3.1, where the integration is done with a Riemann sum of the smoothed flux. We estimate our uncertainty on both these measurements using Monte Carlo simulations, repeating the process for spectra with different realizations of the noise. Figure 3.4 shows the distribution of velocities measured with this technique for all of the supernovae in our training set, and Appendix B contains a table with all the velocity and pseudo-equivalent width measurements, along with their uncertainties. Our training set has a mean velocity of 11.0×10^3 km/s, with a standard deviation of 1.0×10^3 km/s. 16.5% of the objects in the sample are high-velocity

(defined as in Wang et al. (2013) as supernovae with $v_{Si} > 12000$ km/s). The distribution pseudo-equivalent widths has a mean of 101 Å and standard deviation of 26.2 Å.

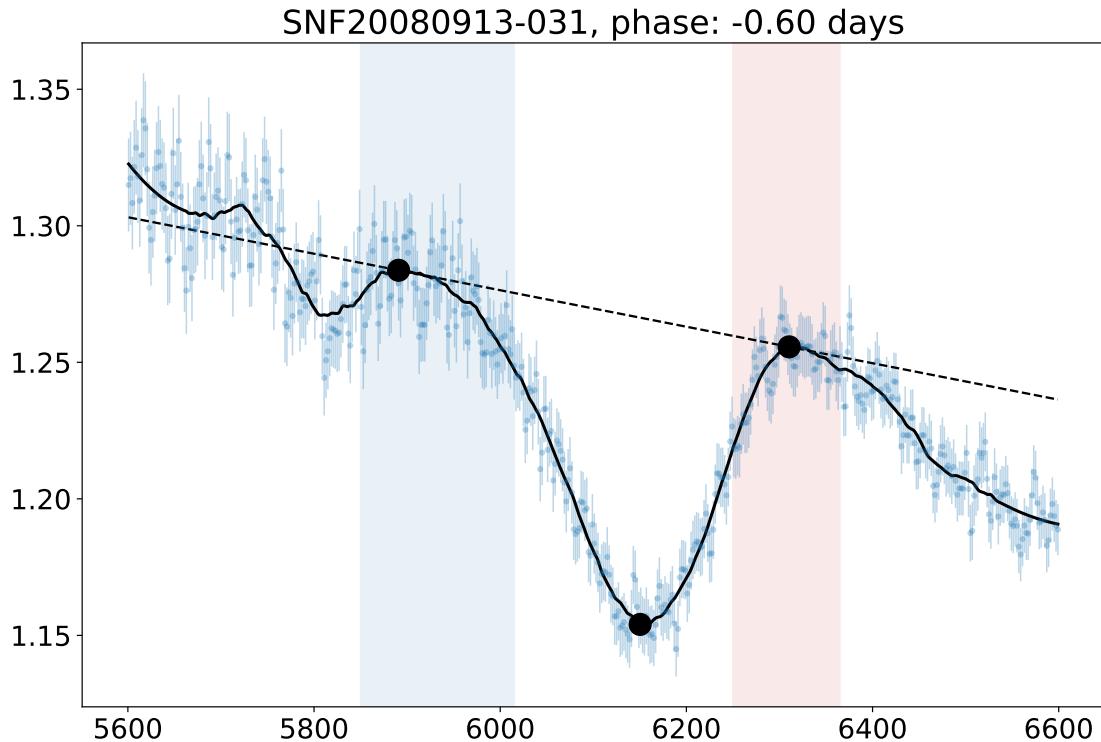


Figure 3.3: An example Si II λ 6355 feature. The original data is shown in blue along with the uncertainties. The optimal smoothing is shown in the thick black line. The blue and red spans show the location of the wavelength windows used to search for the maxima defining the pseudo-continuum (the maxima are the large black points). The pseudo-continuum is plotted as the dashed black line. The location of the maximum absorption wavelength is also shown as a large black point.

This technique for smoothing the spectrum works very well when the spectrum has reasonable resolution and a high signal-to-noise ratio. However, as the resolution and signal-to-noise level decreases, so too does our ability to recover both the limits of the pseudo-continuum and the true location of maximum absorption.

Gaussian Absorption Line Model

One way to work around the limitations of low resolution spectroscopy would be to assume some functional form for the absorption feature being studied. A common – though not very descriptive – choice is a Gaussian. Using such an inflexible function to capture the line information biases the results. To illustrate this, we fit the same data with a model assuming

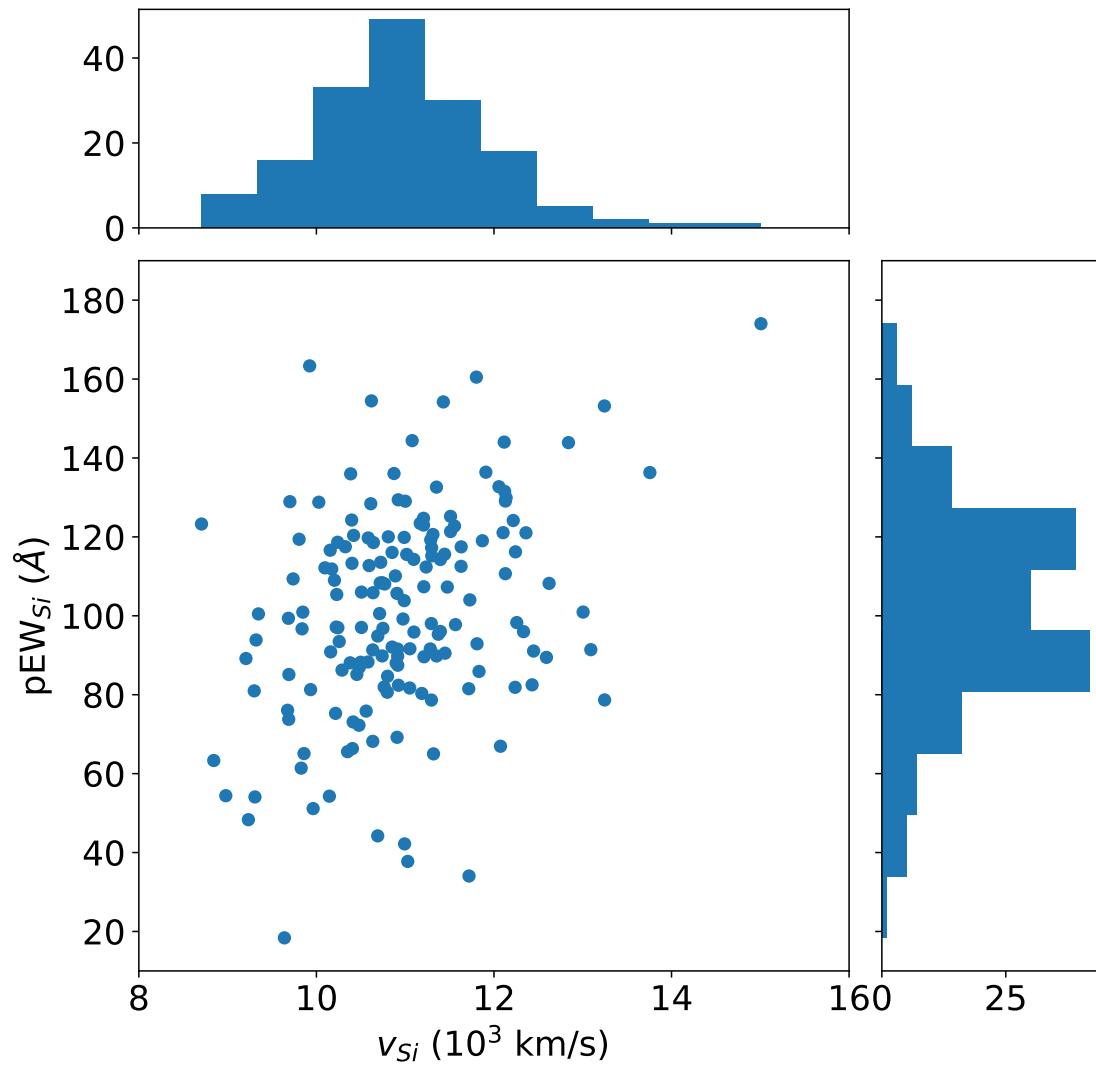


Figure 3.4: Distributions of the spectral indicators of the Si II λ 6355 feature from the training set.

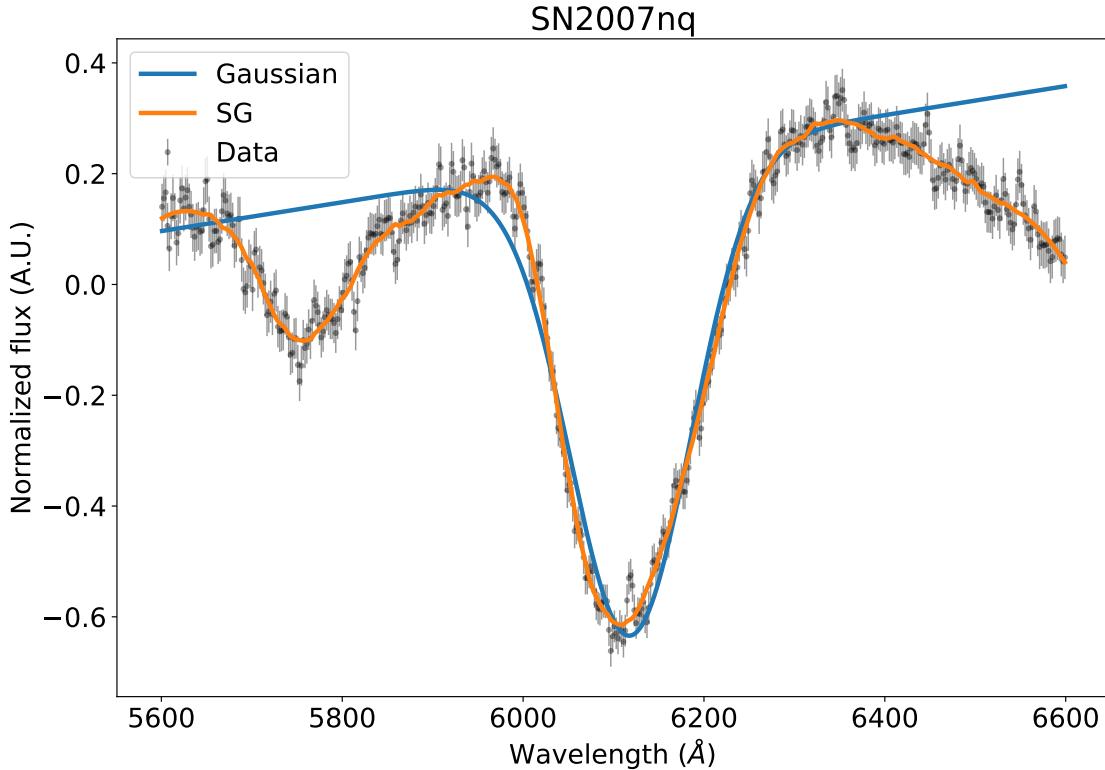


Figure 3.5: Comparison of the Si II λ 6355 feature fit to a Gaussian and the Savitsky-Golay smoothing for SN2007nq. We can see that the full morphology of the feature is not totally captured. More quantitatively, we show in Figure 3.6 a histogram of the difference between the velocity measured from using a Gaussian line profile with a linear continuum and the velocity measured from smoothing and finding the maximum absorption wavelength. There is a clear bias in the velocity; on average, the velocity measured with the Gaussian profile is 200 km/s higher than the true velocity. The pseudo-equivalent width measurements are also slightly biased; on average, the Gaussian measured pEWs are 5 \AA narrower than the true values.

a linear continuum in the wavelength range of interests along with a Gaussian absorption line. Figure 3.5 shows an example of such a fit. We can see that the full morphology of the feature is not totally captured. More quantitatively, we show in Figure 3.6 a histogram of the difference between the velocity measured from using a Gaussian line profile with a linear continuum and the velocity measured from smoothing and finding the maximum absorption wavelength. There is a clear bias in the velocity; on average, the velocity measured with the Gaussian profile is 200 km/s higher than the true velocity. The pseudo-equivalent width measurements are also slightly biased; on average, the Gaussian measured pEWs are 5 \AA narrower than the true values.

3.4 EMFA of the SIII λ 6355 Feature

Our goal is to introduce a method of inferring the shape of the Si II λ 6355 feature that is robust to noise and resolution degradation and also accounts for the true diversity of shapes

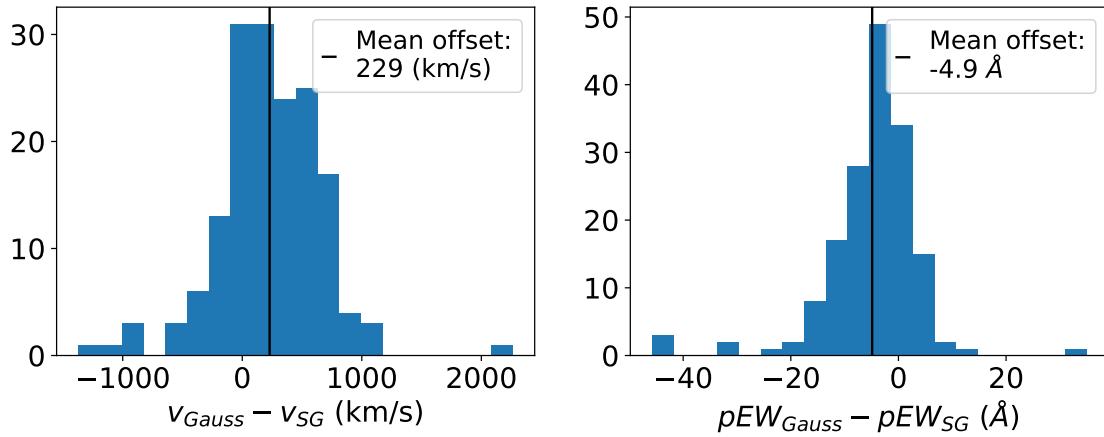


Figure 3.6: Histograms of residuals between the velocity and equivalent width of the Si II $\lambda 6355$ line measured using a Gaussian line profile and the true velocity. There is a clear bias, with the Gaussian measurement giving velocity values that are on average 200 km/s higher than the true values, and equivalent width values that are 5 \AA narrower.

the spectral feature can take on. We accomplish these goals by performing expectation-maximization factor analysis on the normalized spectral features.

Expectation Maximization Factor Analysis

A common unsupervised learning task is dimensionality reduction: using a data set with many features to find a smaller number of principal features that are sufficient to model the data set. The most commonly used technique for dimensionality reduction is principal component analysis, which linearly maps data to a lower-dimensional subspace of the original feature space in such a way as to maximize the data variance. In this analysis, we use a related dimensionality reduction technique: expectation-maximization factor analysis.

Consider a set of p n -dimensional data vectors $\{\mathbf{y}^1, \dots, \mathbf{y}^p\}$. Now assume that there exists some k unobserved latent variables $\{\mathbf{x}^1, \dots, \mathbf{x}^k\}$, themselves each n -dimensional vectors. Each data vector is then generated by

$$\mathbf{y} - \boldsymbol{\mu} = \boldsymbol{\Lambda}\mathbf{x} + \boldsymbol{\epsilon} \quad (3.3)$$

where $\boldsymbol{\mu}$ is an n -dimensional mean vector and $\boldsymbol{\epsilon}$ is a noise vector that is Gaussian distributed with mean 0 and covariance $\boldsymbol{\Psi}$. The matrix $\boldsymbol{\Lambda}$ is known as the loading matrix and describes the relative contributions of each factor to the observed variables.

This statistical formulation is quite similar to principle component analysis; indeed, the components found with factor analysis are often quite similar to those found using principle component analysis. However, the two techniques differ in their assumptions on the covariance matrix $\boldsymbol{\Psi}$. In the PCA framework, $\boldsymbol{\Psi} = \sigma^2 \mathbf{I}$, while in EMFA, $\boldsymbol{\Psi}$ can be any diagonal matrix. This means that EMFA gives more general description of the noise.

We train the factor loadings using an expectation-maximization (EM) approach. Expectation maximization is an iterative technique for maximizing a likelihood with latent variables. The first step (the E-step) finds the expectation value of the hidden variables given the model parameters. The next step (the M-step) fits the model parameters to maximize the likelihood given the expectation value of the hidden variables. These steps are repeated until convergence.

In our case, each of the observables is the flux in some wavelength bin, and the factors F_i are n -dimensional vectors, where n is the number of flux bins in the training spectra. For this analysis, we used the factor analysis implementation included in the `scikit-learn` Python package (Pedregosa et al., 2011).

Visualizing Model Components

The EMFA components are shown in Figure 3.7. Each figure shows the impact of adding a range of loading factors (elements of the loading matrix Λ) to the mean spectral feature. We qualitatively see how the velocity and equivalent width is affected by each components. Higher loadings of component 1 correspond to higher velocity and larger equivalent width lines. Higher loadings of component 2 correspond to high velocity, but shallower features. The third component modifies the shape of the bluer portion of the feature. Similar effects can be seen in the nearby Si II λ 5972 feature.

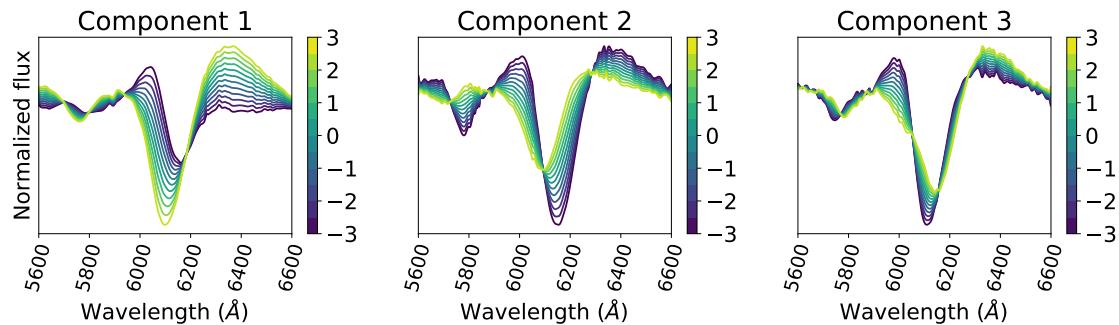


Figure 3.7: Visualization of the components (x) of the EMFA model. Each subfigure shows the effect of adding various values of multiples of the components to the mean vector.

The distributions of the loading coefficients are shown in Figure 3.8 and 3.9. The qualitative observations about the relationship between the loading coefficients and the velocity and equivalent widths of the feature are confirmed there, as well as in Figure 3.10 where the spectral indicator measurements are plotted directly against the loading coefficients of each object in the training sample.

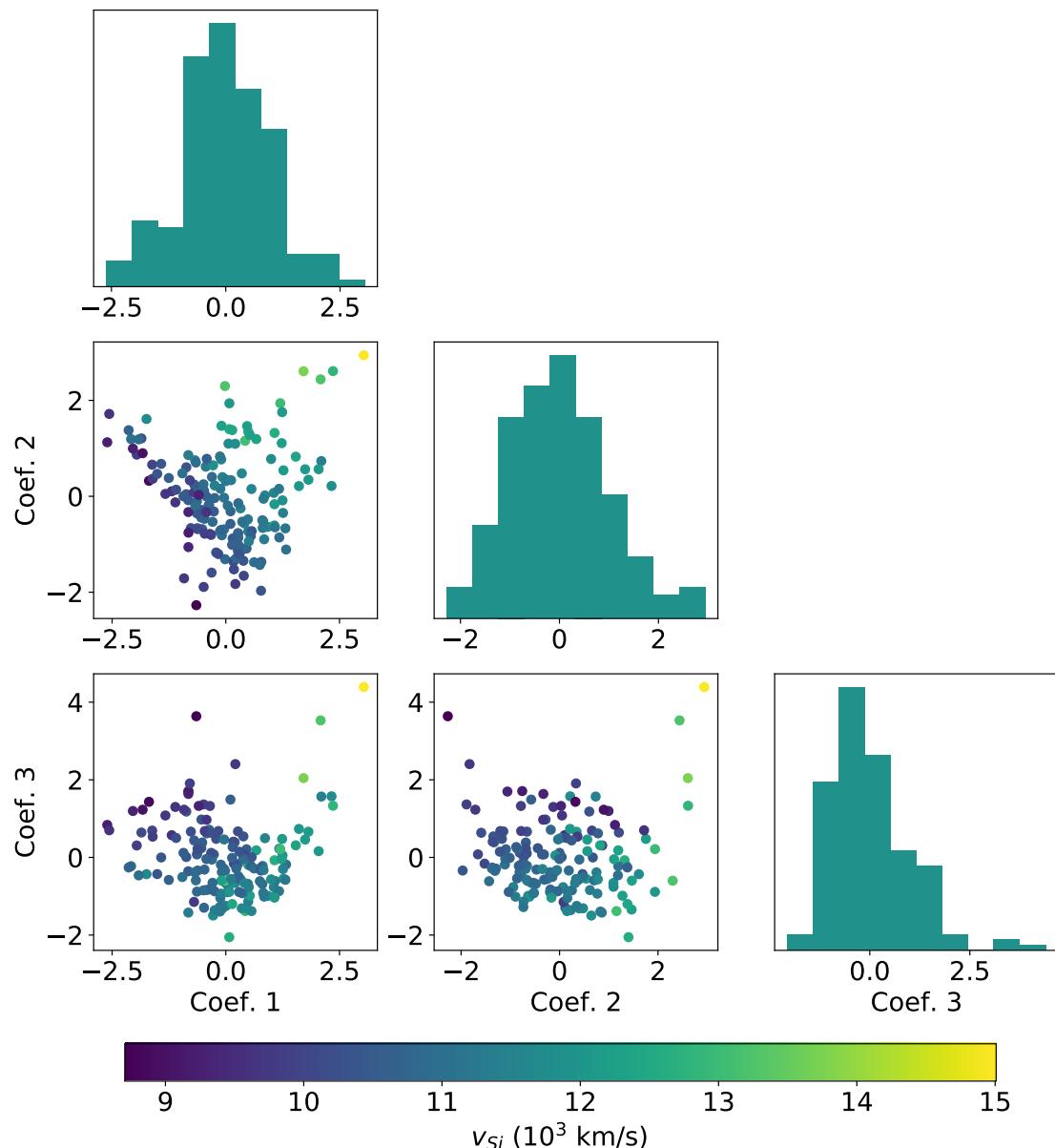


Figure 3.8: Corner plot showing the marginal distributions of the loading coefficients for the training set, colored by the measured velocity of the Si II $\lambda 6355$ feature. Each point in the scatter plot represents one supernova.

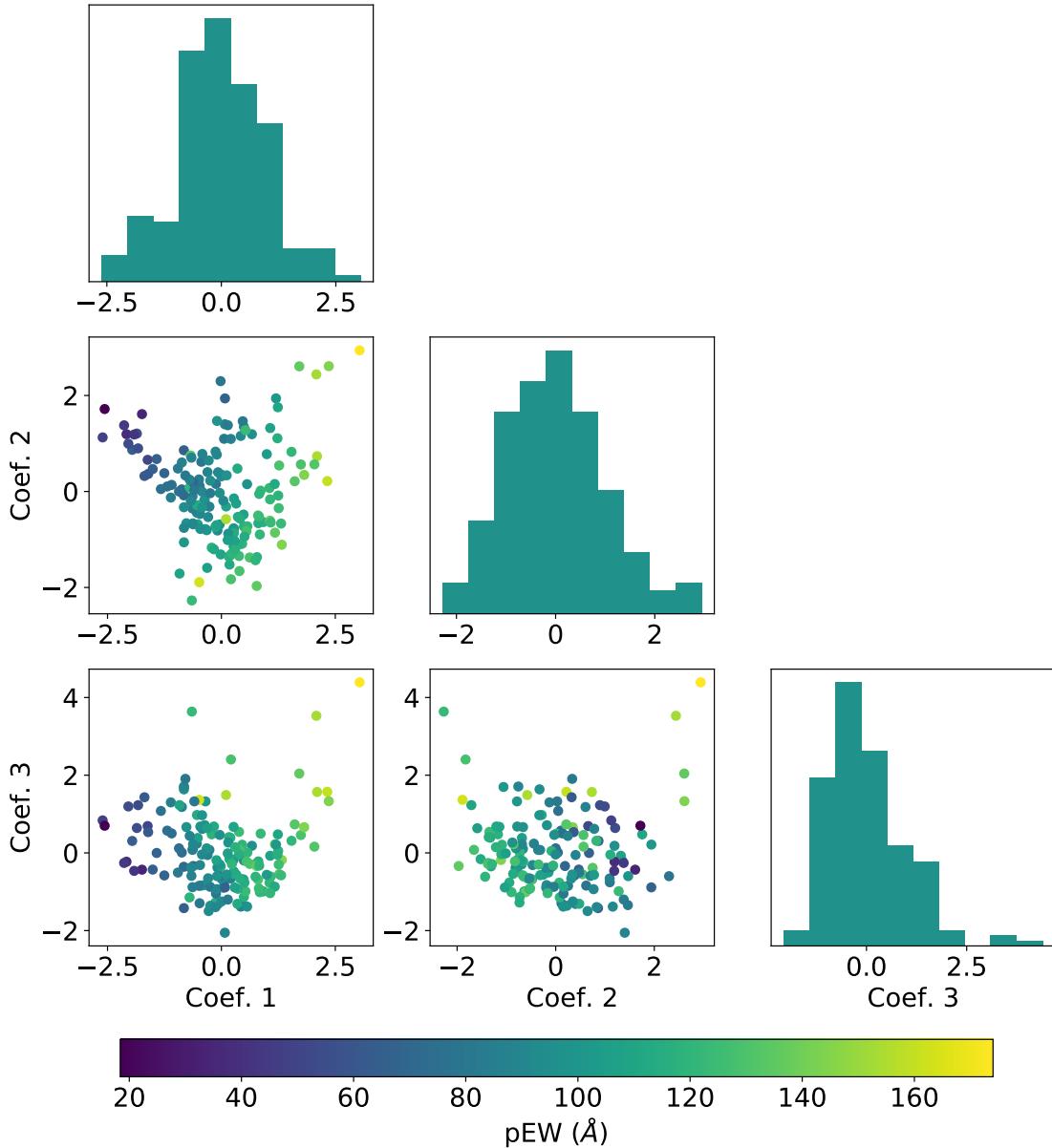


Figure 3.9: Same as Figure 3.8, but colored by the pseudo-equivalent width.

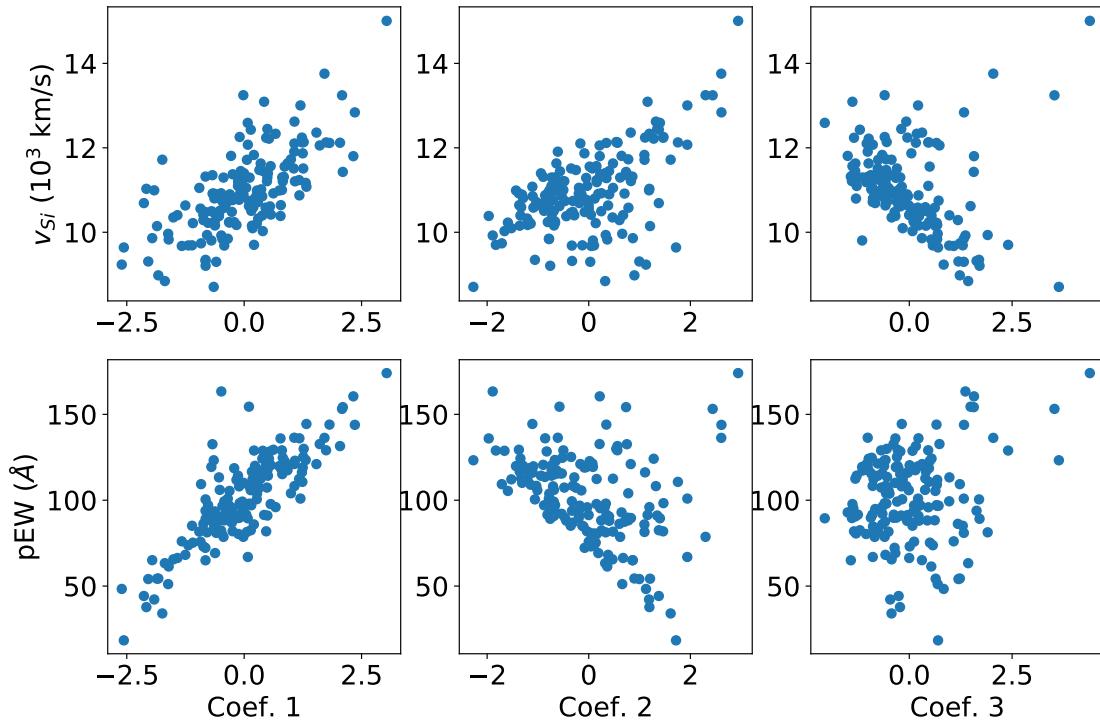


Figure 3.10: Scatter plots of the loading coefficients of the training data with their measured spectral indicators. We can see that each of the components is correlated with the velocity. Only the first two components are correlated with the pseudo-equivalent width.

3.5 Validation

Recovering Spectral Features at Native Resolution

We can recover the spectral indicator measurements by reconstructing the full spectral feature from the EMFA model and measuring the velocity and equivalent width of the resulting reconstruction. When reasonably high resolution spectra are available, this reconstruction is unnecessary, so the native resolution recovery presented here is meant to provide a baseline estimate of how well the EMFA model can capture the inherently non-linear Si II $\lambda 6355$ features. Its real power will come into play when we estimate the velocity and pseudo-equivalent width from lower resolution or noisier spectroscopy.

Some examples of the feature recovery at native resolution are shown in Figure 3.11. The histogram of residuals is shown in Figure 3.12. The width of these distributions tells us how well the EMFA is capturing the spectral features. We find that the standard deviations (equivalently the root-mean-square or RMS) of these residual distributions are 369 km/s and

5.8 \AA , respectively. The normalized median absolute deviations² of these distributions are 253 km/s and 3.6 \AA . These errors in the recovery values are comparable to the average error on the original measurements (385 km/s for velocity and 6.8 \AA for the pseudo-equivalent widths).

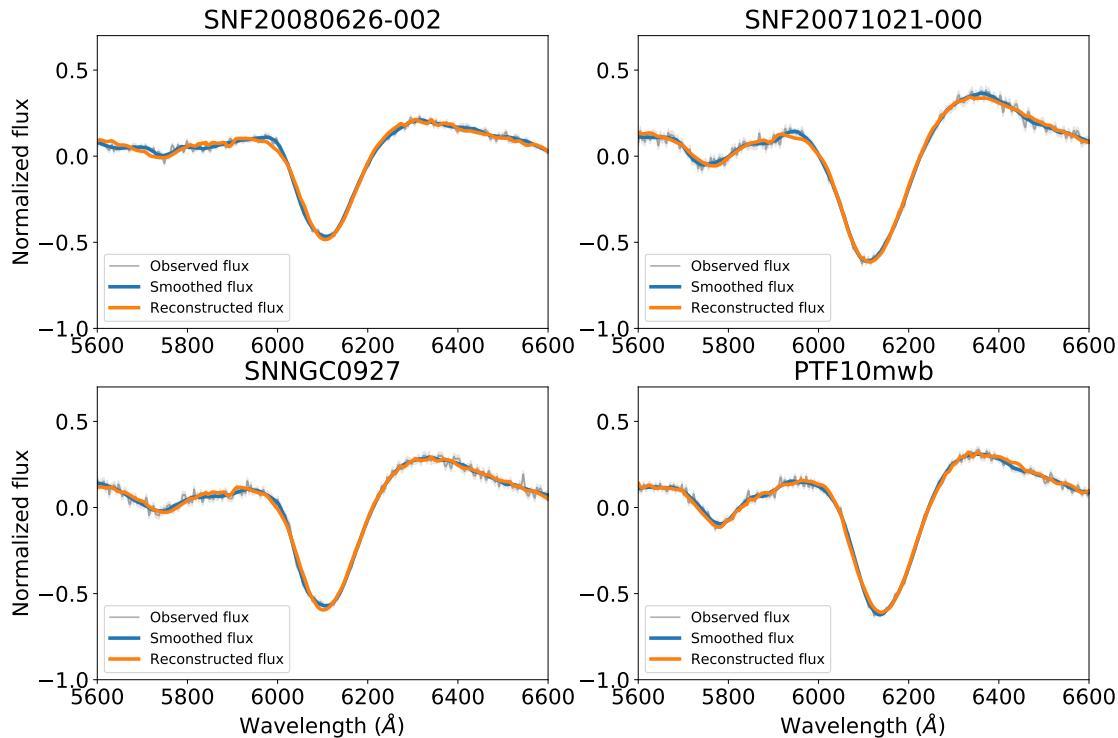


Figure 3.11: A random selection of recovered spectral features at the native resolution of the SNfactory spectra. The gray lines show the observed data, the blue line shows the data smoothed by the optimal Savitzky-Golay filter, and the orange line is the reconstructed flux.

3.6 Testing on External Data

In order to ensure that our model is not overfit to our training data, we evaluate the effectiveness of our EMFA of the Si II $\lambda 6355$ feature on our external data set from the Berkeley SuperNova Ia Project (BSNIP, Silverman et al., 2012) described in Section 3.2. Using the same techniques as in Section 3.5, we compare the velocities and pseudo-equivalent widths of the features inferred from the EMFA model and the true measured spectral indicators

²

$NMAD = 1.4826 \times \text{med}(|\mathbf{x} - \text{med}(\mathbf{x})|)$

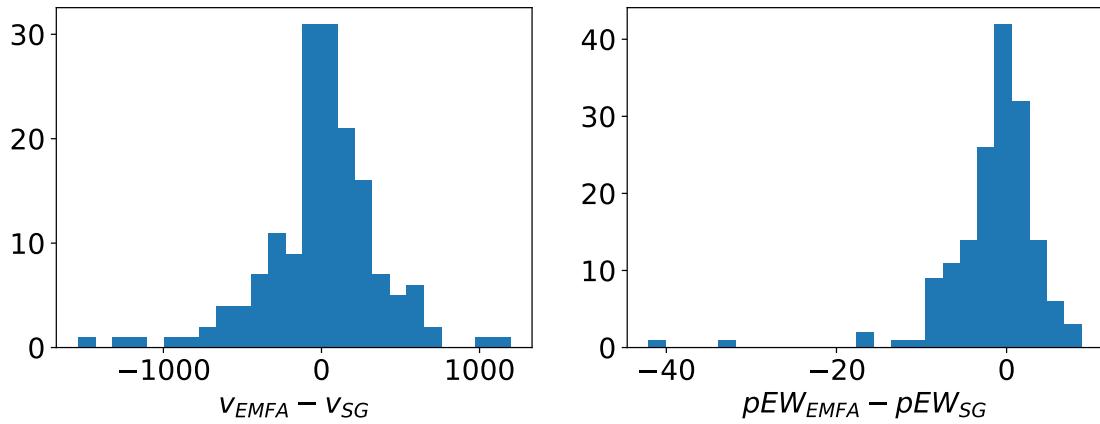


Figure 3.12: Histograms of the residuals between the velocities and equivalent widths measured from the flux reconstructed using the EMFA model and the original observed flux in the training data.

from the BSNIP spectra rebinned into a similar spectral resolution as the SNfactory training spectra. Figure 3.13 shows the residuals between the spectral indicators measured from the EMFA fits and the originally measured values. The results for both the SNfactory validation and the BSNIP test are all summarized in Table 3.2. The spreads of these distributions are slightly larger than those of the training sample, and we investigate some causes for this in Section 3.6.

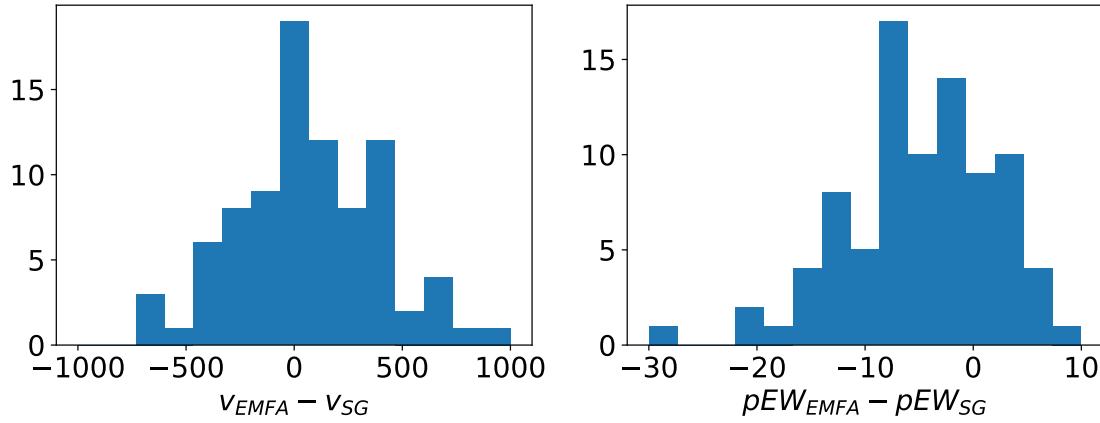


Figure 3.13: Histograms of the residuals between the velocities and equivalent widths measured from the flux reconstructed using the EMFA model and the original observed flux in the validation set. The outlier SN2005M has been removed from these plots (see Section 3.6)

Data set	Velocity resid.		Equiv. width resid.	
	(km/s)	(Å)	RMS	NMAD
SNfactory	369	253	5.8	3.6
BSNIP	392	355	7.7	7.1

Table 3.2: Measurements of the spread of the residuals between spectral features measured from the EMFA modeled spectra and spectral features measured from the true spectra for both the SNfactory (training) set and the BSNIP (test) set.

Investigating Large Residuals

There are a few failed reconstructions from our validation. We show them in Figure 3.14, and discuss them here.

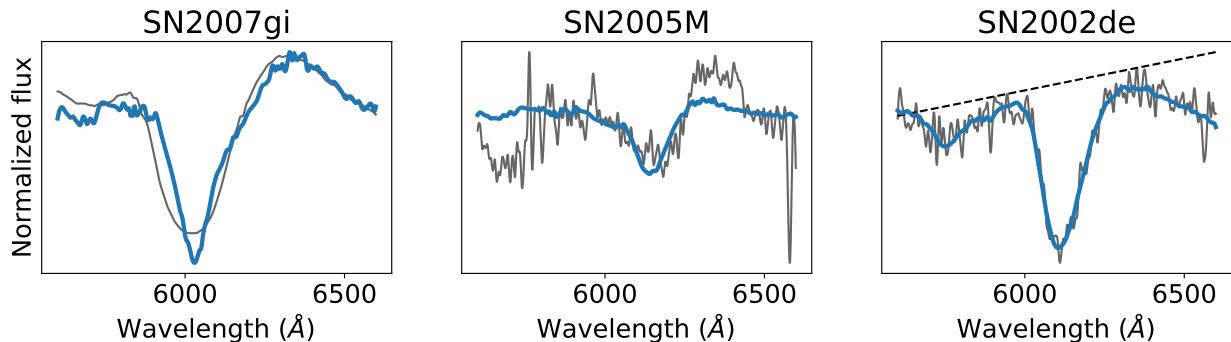


Figure 3.14: Validation spectra where the EMFA fit and/or pseudo-continuum determination failed. The left frame is SN2007 gi, an extremely rapidly expanding object. The middle frame is SN2005M, which may be 1991T-like. The right frame shows an example of an object for which the pseudo-continuum determination failed due to underestimated variance in the spectrum. In each subfigure, the gray line represents the SG smoothed flux, and the blue line shows the best-fit EMFA flux. In the right figure, we also show the pseudo-continuum as a dashed black line.

SN2007gi was a very well observed object with a high signal-to-noise spectrum, and thus precise measurements of the spectral indicators. Though the size of the residual was within the usual uncertainty of a measurement from a moderately well-observed spectra, the residual for this object was significantly larger than the measurement uncertainty. Looking at the recovered flux, we see that the EMFA model flux doesn't match the observed flux. SN2007gi is an extremely high velocity object ($v_{Si} = 15740 \pm 180$ km/s with a large equivalent width

($pEW = 176.9 \pm 1.6$ Å), indicating that we may be somewhat limited by the diversity of our training set.

SN2005M was also a significant outlier. SN2005M has the shallow silicon lines of a SN1991T-like object, but a low velocity (8300 ± 260 km/s) that is uncharacteristic of SN1991T-like objects, which have typical velocities of ~ 10000 km/s (Blondin et al., 2012). Since such shallow line objects were explicitly excluded from our training data set, the model is unable to capture this variation.

By eye checks of the remaining objects with large pseudo-equivalent width residuals reveal that it is not the EMFA that is failing to capture the feature, but failures in the pseudo-continuum determination in the SG filtering process. Usually, this is due to the variance spectra being underestimated, resulting in undersmoothing of the curve, and the limits of the pseudo-continuum being determined by noise spikes. An example of this is shown in the final panel of Figure 3.14 for SN2002de. We conclude that, aside from the specific outliers of SN2007gi and SN2005M, the slightly larger spread in our validation residuals is driven by this variance underestimation in the validation data set, and not overtraining of our model.

3.7 Simulated Roman Prism Spectra

Generating Roman Prism Spectra

The Nancy Grace Roman Space Telescope (hereafter Roman) is a future space telescope mission designed to constrain cosmological parameters with wide-field optical and near-infrared imaging. In addition to the imaging instrument (the Wide Field Channel, or WFC), a low-dispersion slitless prism has also been proposed as a tool to obtain spectroscopy for SNe Ia.

The full details of the prism simulation we use in this section can be found in Rubin et al. (in preparation), but we will present a summary here. The prism is still in design stages, so we assume a similar dispersion to the previously proposed Integral Field Channel (IFC), but with narrower wavelength coverage (0.7 to 1.8 μm). The survey simulation assumes an exposure time of one hour per pointing. This yields the at-max signal-to-noise ratios shown in Figure 3.15, where we report the average signal-to-noise ratio per pixel from 5600-6600 Å in the rest frame (the wavelength region of interest for this chapter). The average signal-to-noise ratio is calculated for a normal SN Ia in 38 evenly spaced redshift bins from 0.125 to 1.175. It is worth noting that the parameters of this survey (the wavelength coverage, dispersion, exposure times, etc) have not been optimized in any way; this survey serves as a benchmark for the spectral indicator measurement technique discussed here. Future work can use these analyses to find a more optimal survey strategy and instrument design.

The simulated data set is generated as follows. For each object in the training set and for each redshift bin, we artificially redshift the spectrum to the redshift of the bin, resample the spectrum into the resolution of the prism spectrograph, and generate 50 realizations of the noise. The separate realizations allow us to inspect how the uncertainty (our confidence in

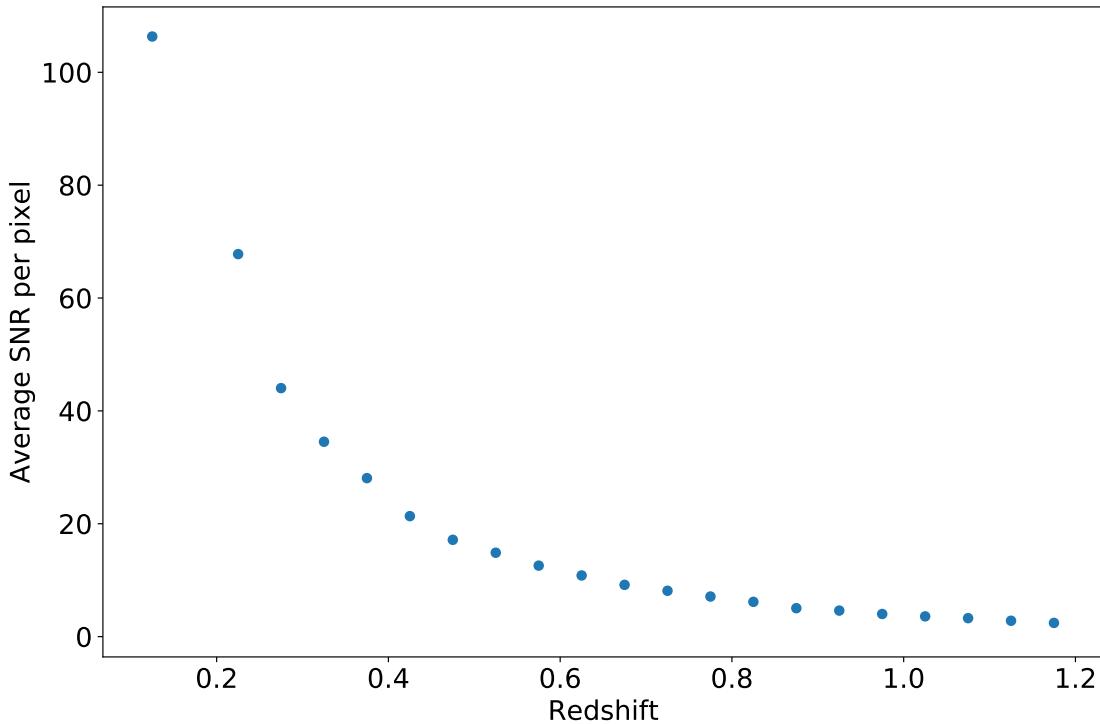


Figure 3.15: Average signal-to-noise ratio for assumed for prism spectra used in the simulations. We report the average from 5600-6600 Å in the rest frame since this is the wavelength region of interest.

the measurement value due to noise fluctuations) changes with redshift (or S/N) for spectra at these resolutions as well as how the errors (systematic offsets between the model and the true underlying data) change with redshift. A few example Roman prism spectra in a range of redshifts are shown in Figure 3.16.

Each of these spectra was preprocessed as described in Section 3.2 and is available as part of the data repository accompanying this work.

Spectral Indicator Recovery Results

For every spectrum generated, we measured the velocity and pseudo-equivalent width using the SG smoothing, the Gaussian fit, and the EMFA fit methods. We then compared the results of these fitting methods to the true values of these spectral indicators (i.e. those measured from the original, high resolution, low noise spectra from the training sample).

In the validation step, the spectra we were fitting were at the same resolution as the training data. Therefore the data vectors had the same length and correspond to the same wavelengths as the training data, so we do not need to interpolate either the data or the model. Now we no longer have the same resolution, so we instead interpolate the model

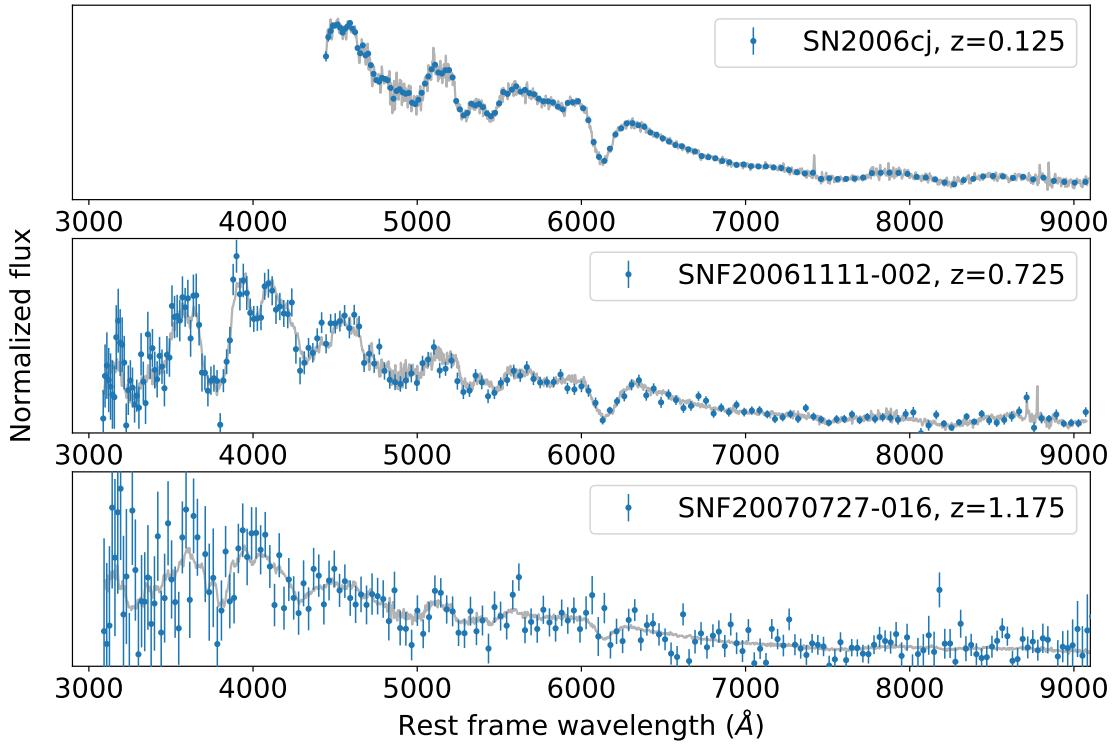


Figure 3.16: Example realizations of the training set near-max spectra observed with the Roman prism spectrograph with a one hour exposure time.

components using a spline and fit this interpolated model by minimizing

$$\chi^2 = \sum_{\lambda} \frac{(f_{mod}(\lambda) - f_{obs}(\lambda))^2}{\sigma_{obs}^2(\lambda)} \quad (3.4)$$

where λ indexes the wavelength bins of the observation, f_{mod} is the spline interpolated model spectrum evaluated at the wavelength bin λ , and f_{obs} is the observed flux. An example of the recovered flux from one realization of one object is shown in Figure 3.17.

First, we examine how the error in the measurements changes with redshift, where the error is defined by the difference between the measurements obtained from the noisy, degraded spectrum and the original data. Figure 3.18 and 3.19 show the average absolute difference between the velocities and pseudo-equivalent widths measured from the noisy data and those measured from the original data. At low redshifts (high signal-to-noise), each of the methods are roughly comparable. As the noise increases, though, the EMFA reconstruction does significantly better at recovering both the velocities and equivalent widths. Using the EMFA, we can obtain an estimate of the velocity to within 500 km/s out to a redshift of $z \sim 0.75$, compared to $z < 0.6$ with the SG filtering method.

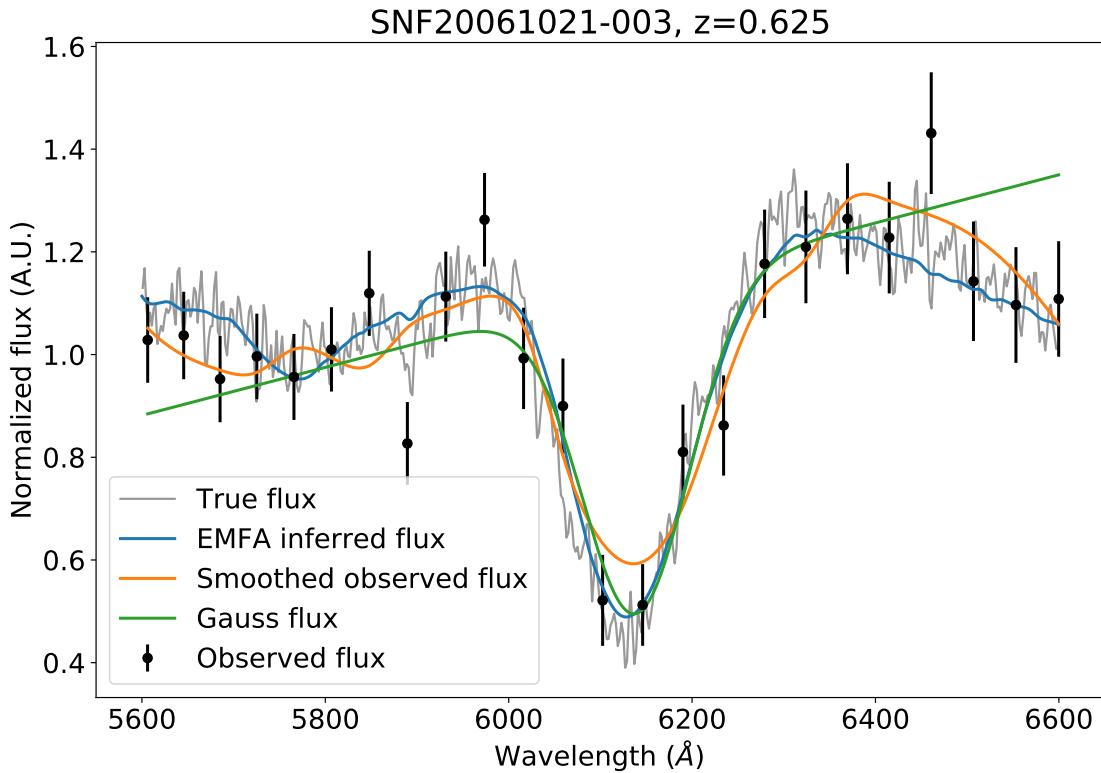


Figure 3.17: Example of the various methods tested to recover the spectral indicators from the low resolution, noisy spectra from the Roman prism. The original training set spectrum that was resampled and noised is shown in gray. The realization of the prism observation (with uncertainties) is shown as the black data points. The best-fit EMFA spectral feature is shown in blue, the smoothed version of the observed flux is shown in orange, and the best-fit Gaussian line profile is shown in green.

We also examine how the uncertainty in each of these measurements changes with the redshift in this prism survey. We look at the spread of the spectral indicator measurements among the 50 realizations of each object in each redshift bin. Figure 3.20 shows the average uncertainty of the velocity measurements as a function of redshift for each measurement technique. Figure 3.21 shows the same but for the pseudo-equivalent width measurements. Once again, we see that all measurement techniques are roughly comparable in both metrics at low redshift (high signal-to-noise). At higher redshifts, the EMFA again outperforms the other techniques.

From this benchmark prism survey simulation, we find that our new method for recovering the Si II $\lambda 6355$ spectral indicators is both more precise and more accurate than other commonly used measurement techniques. By using a model of the feature informed by the data, we are able to extract more useful information from noisier data, allowing us to obtain the same spectral information in shorter exposure times, or equivalently out to higher

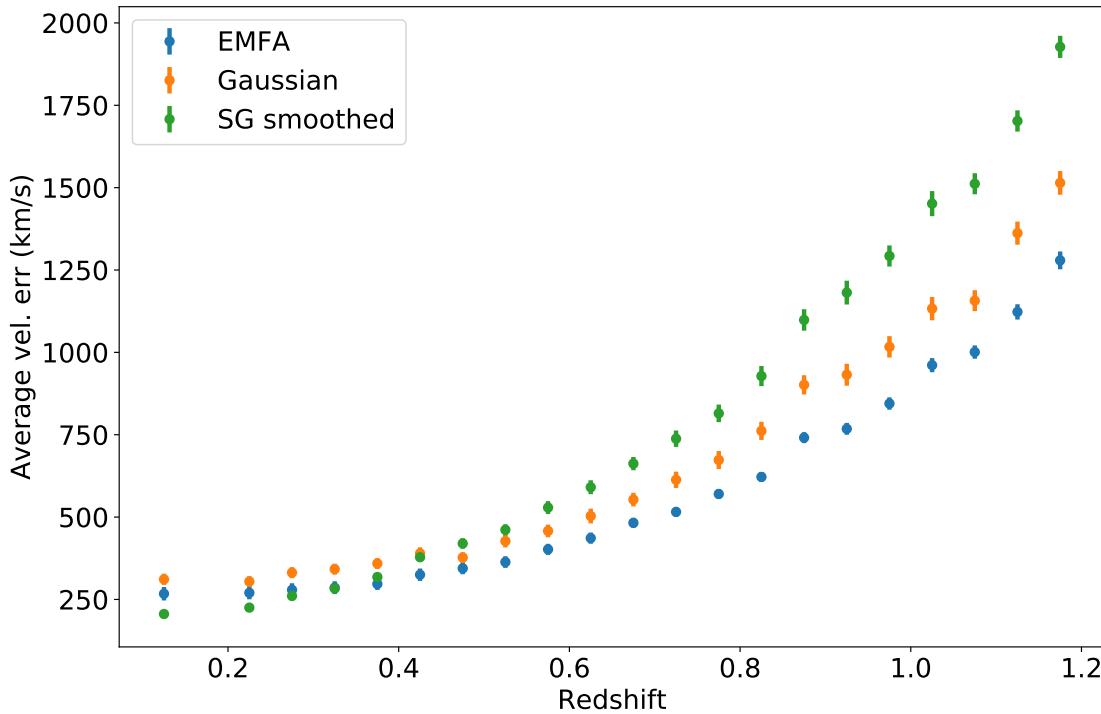


Figure 3.18: Per-redshift-bin average of the absolute value of differences between measured velocities and true velocities as a function of redshift for the simulated Roman prism spectra. At low redshifts, the SG filtering method captures the true value best, but as noise increases, the EMFA method outperforms all other methods.

redshifts with the same exposure time.

3.8 Conclusions

We have presented a new method for reconstructing the Si II $\lambda 6355$ spectral feature of Type Ia supernovae. By using available high-resolution spectroscopic data, we are able to recover the velocity and equivalent width of the feature in low-resolution, noisy spectra with more precision and accuracy than the other methods shown. We have validated our model on an outside data set to ensure that the model was not overtrained and could generalize to other data sets. We also tested the performance of the model on simulated lower-resolution data for a range of signal-to-noise ratios as a benchmark, finding significant improvements in the measurement uncertainty and systematic error when using this new model instead of other techniques.

The results of the simulations can be used in future work to optimize cadence, observation, and instrument designs for upcoming supernova surveys. The improved performance in this

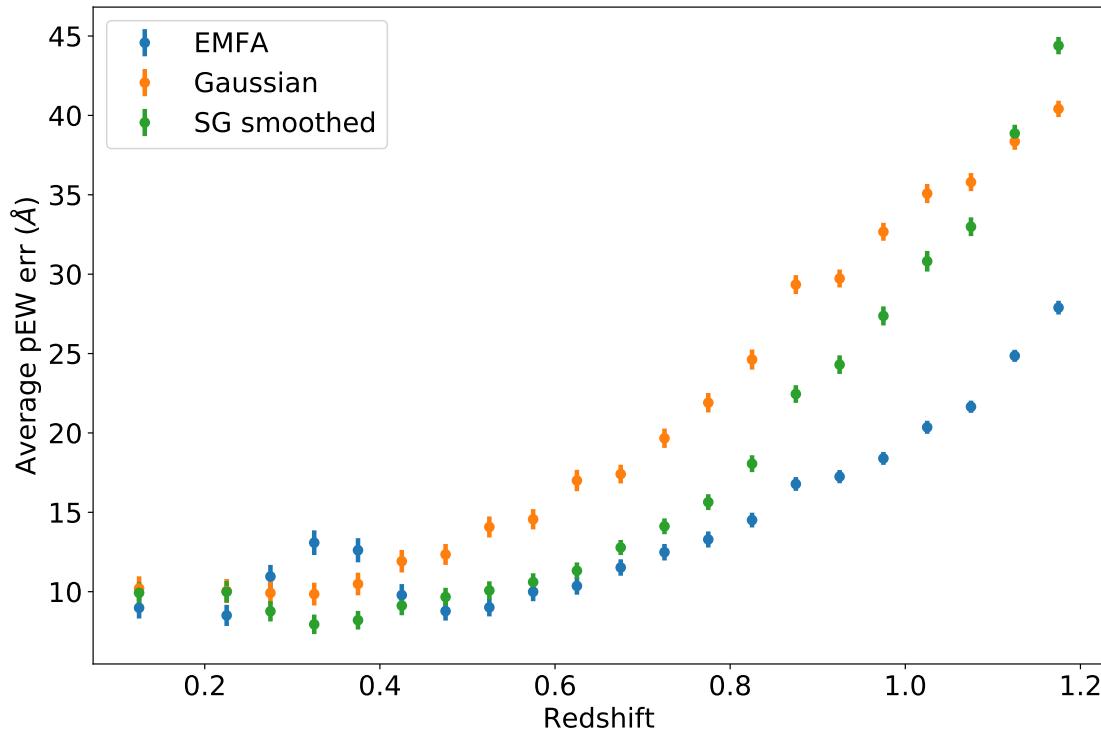


Figure 3.19: Same as Figure 3.18, but for pseudo-equivalent width measurements. At low redshifts, all methods give comparable errors, but as the noise increases, the EMFA method is the preferred technique.

metric could allow for more objects to be observed, or for even more accurate estimates of spectral indicators with the same exposure times.

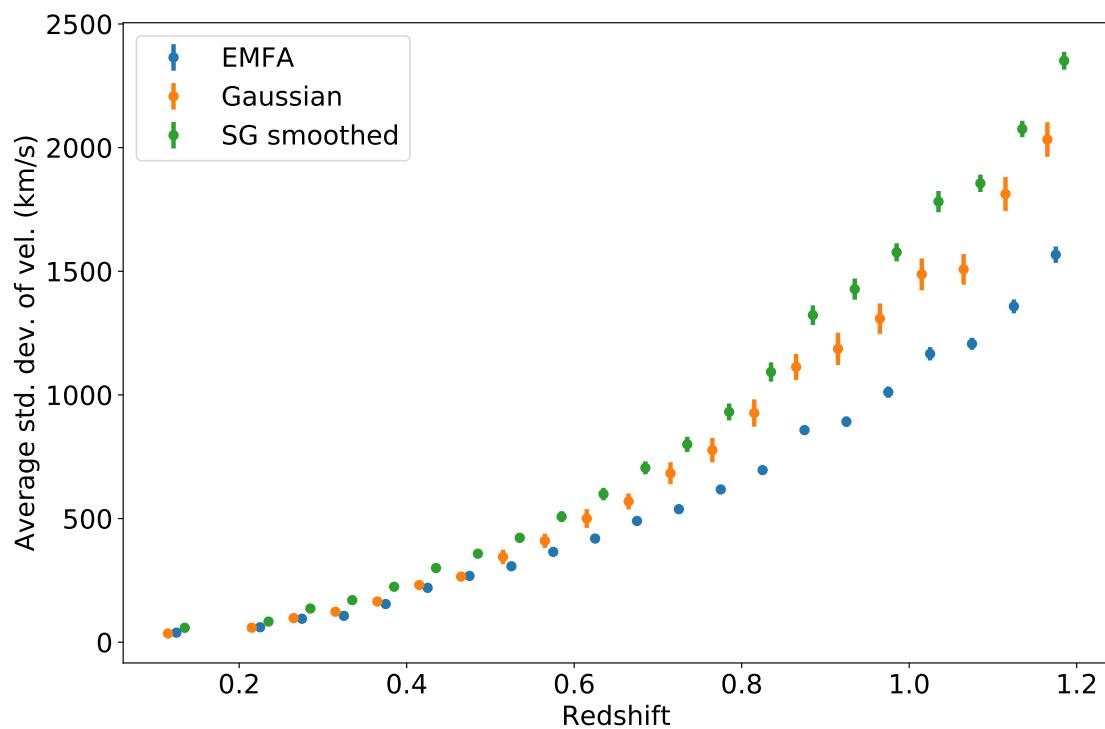


Figure 3.20: Per-redshift-bin average of the per-object standard deviations of the velocity measured with three techniques plotted as a function of redshift for the simulated Roman prism spectra. The EMFA recovery technique outperforms both the Gaussian and SG filter smoothing techniques at all redshifts.

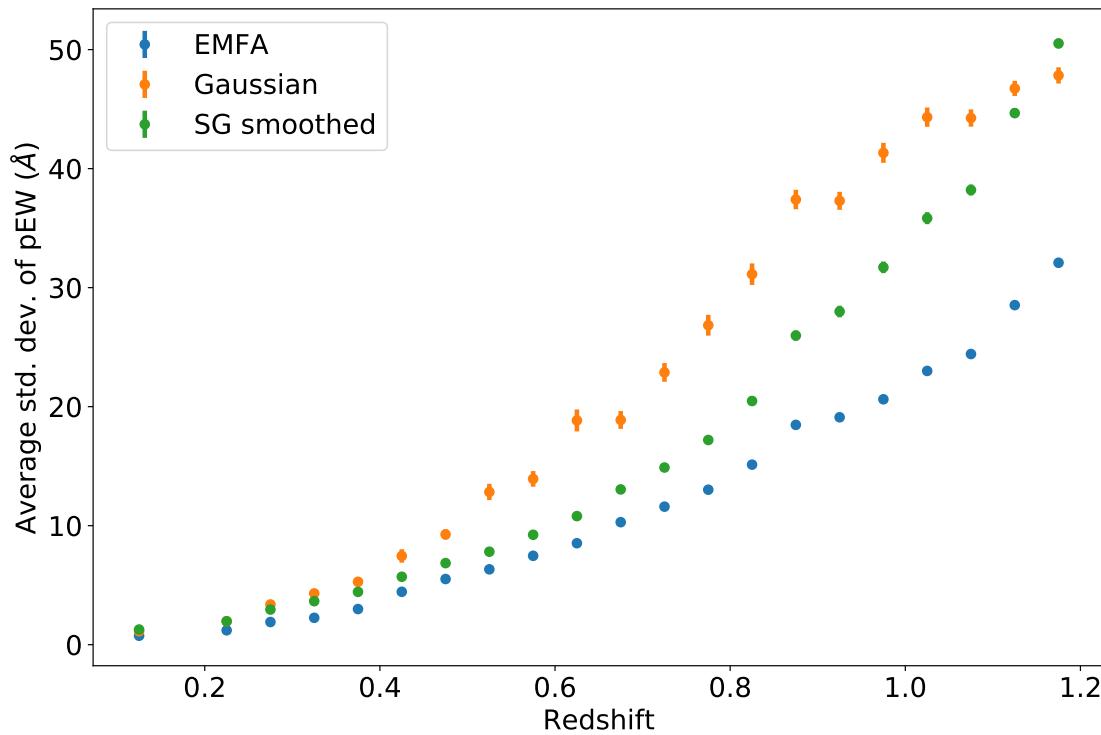


Figure 3.21: Same as Figure 3.20, but for pseudo-equivalent width measurements. In contrast to the velocity measurement uncertainty, the SG filter smoothing method seems to perform better than the Gaussian measurement in this metric. However, the EMFA still out-performs both methods.

Chapter 4

Kernel Density Estimates for Generating Mock Type Ia Supernova Observations with SALT2 and SNEMO

4.1 Overview

As discussed in Chapter 1, the most common method currently for calibrating supernova brightnesses for cosmology uses the SALT2 spectral model of Guy et al. (2007) along with an assumed linear relationship between the model parameters and absolute magnitude at maximum brightness. The SALT2 model assumes that the flux f_{mod} at wavelength λ and phase p is given by

$$f_{mod}(\lambda, p) = x_0 [M_0(\lambda, p) + x_1 M_1(\lambda, p)] \times \exp [c CL(\lambda)] \quad (4.1)$$

where x_0 , x_1 , and c are parameters approximately describing the overall scale, light curve decay rate, and color of each supernova respectively, $M_0(\lambda, p)$ and $M_1(\lambda, p)$ are functions of wavelength and phase describing the average spectral sequence of SNe Ia and typical variance SN spectral sequences, and CL is a function of wavelength, representing the effects of both intrinsic and extrinsic color variation on observed flux. In a typical analysis, each observed light curve is fit with this model, giving x_0 , x_1 , and c values for each supernova. The distance modulus to each supernova is then modeled linearly (Tripp, 1998; Tripp & Branch, 1999) by

$$\mu = m_B^* + \alpha x_1 - \beta c - M \quad (4.2)$$

where m_B^* is the apparent magnitude in the Bessell B-band at maximum brightness of a supernova with the observed x_0 , x_1 , and c (as predicted by the SALT2 model), and α , β , and M are global standardization parameters obtained from a simultaneous fit of μ and the distance modulus as a function of cosmological parameters.

After parametrizing the light curves with the SALT2 model and accounting for empirical relations between these parameters and luminosity, there still is scatter of approximately 0.14 mag remaining in the standardized magnitudes. Some portion of this scatter may be intrinsic, but several studies have shown that at least some of the scatter stems from flawed assumptions of the model, as we now discuss.

Firstly, the light curve parametrizations themselves do not capture all of the diverse ways that observable supernova properties correlate with luminosity. The supernova twins analysis of Fakhouri et al. (2015) points to this particular problem with the standard SALT2 analysis pipeline by showing that a direct comparison of maximum brightness spectra, without any parametrization of the broadband light curves, results in a smaller scatter in standardized brightnesses. The existence of spectral subclasses of Type Ia supernovae (e.g. Branch et al. (2006)) also suggest that there is more information contained within the spectra of SNe Ia than the SALT2 model is capturing. The SuperNova Empirical Models (SNEMO, Saunders et al. (2018); described in more detail in Section 4.2) were introduced in part to mitigate the effects of using such incomplete models of supernova variation by including additional time-series flux components.

Even with a perfectly descriptive supernova flux model, a straight-forward linear relation may be unable to capture all of the details of the relationship between model parameters and luminosity, resulting in additional unexplained dispersion in standardized magnitudes. For example, Rubin et al. (2015) found a preference for a broken-linear relationship between light curve color and luminosity. Rose et al. (2020) showed that a non-linear Gaussian process may better encapsulate the relationship between SN color and absolute luminosity. This issue has been significantly less well-studied than the inflexibility of the flux models, but further studies would require improved simulations to understand the relative fraction of uncertainty stemming from the inflexibility of the flux parametrization model as opposed to the inflexibility of the standardization model.

Both of these issues are further complicated by selection effects and observational error. Without accounting for these uncertainties and propagating them through the analysis, the resulting cosmological parameter measurements are potentially biased. In order to have accurate simulations to quantify and correct for these biases, we must have a data-informed model of the underlying parameter populations. Indeed, Scolnic & Kessler (2016) show that using incorrect estimates of the underlying stretch and color distributions of the SALT2 parameters results in a small bias in the dark energy equation-of-state parameter, w . Simulations are an important tool for quantitatively disentangling the intrinsic scatter from the flux parametrization error and supernova standardization error; creating tools for these more descriptive simulations is the main motivating factor of this work.

In this analysis, we present a tool for flexibly estimating the underlying population distributions of the model parameters for the SALT2 and SNEMO models in order to build accurate simulations that can address some of these issues. We present an overview of the SNEMO model in Section 4.2 and discuss the collection of spectro-photometric time-series data that we use throughout the analysis in Section 4.3. In Section 4.4, we present the general framework for estimating latent joint probability distributions and the methods for

quantitatively comparing these estimates to the simpler distribution estimation models like a multivariate Gaussian. We apply this methodology to our data and explain how to move from these parameter distribution estimates to mock observations in Section 4.5. Finally, we compare the distributions of maximum-brightness spectral feature measurements in Section 4.6, and conclude with some proposed further analyses in Section 4.7.

4.2 SNEMO

Again as we discussed in Chapter 1, the SuperNova Empirical MOdels (SNEMO) are a family of linear models that were built to capture more of the spectral variation than is captured by fewer-parameter models like SALT2. The mathematical form of the SNEMO flux model is very similar to that of SALT2, but with a larger number of parameters. The flux at wavelength λ and phase p is modeled by

$$f_{\text{mod}}(\lambda, p) = c_0 \left[e_0(\lambda, p) + \sum_{i=1}^k c_i e_i(\lambda, p) \right] \times 10^{-0.4A_s CL(\lambda)} \quad (4.3)$$

where $e_0(\lambda, p)$ is the average spectral sequence, and each of the $e_i(\lambda, p)$ represent orthogonal aspects of spectral variation. $CL(\lambda)$ is the Fitzpatrick (1999) extinction relation and is fixed across all time scales to represent extinction from dust. The model parameters are then the overall scaling c_0 , the variational parameters c_i and the dust-reddening parameter A_s . These $k + 1$ parameters represent the full spectral time-series data in the model space, similar to how the parameters $\{x_0, x_1, c\}$ represent the time-series of supernova spectra in the SALT2 model.

Saunders et al. (2018) presents three different variations of the model, each with different numbers of parameters and intended for different uses. SNEMO2 has a single spectral variation vector ($k = 1$) in addition to the overall scale and color parameters, and serves as a point of comparison to SALT2. SNEMO7 has six spectral variation vectors ($k = 6$) and is presented as a model for supernova standardization, as it minimizes the unexplained dispersion in standardized magnitudes of supernovae after using the coefficients of this model to linearly standardize the luminosity. Finally, SNEMO15 ($k = 14$) is presented as a model to explain as much of the spectral variability as possible, as measured by the total χ^2 difference between the model and the observed fluxes at all of the modeled wavelengths and phases. Throughout this work, we make use of and compare each of these three models along with the SALT2 model.

4.3 Data

In order to generate new spectral time-series data from this model, we first need measurements of the spectral model parameters from a representative data set. The representative data set we are using throughout this analysis is the spectral time-series data from the Nearby

Supernova Factory (SNfactory; Aldering et al., 2002). SNfactory has collected spectrophotometry from over 400 SNe Ia. Of these, 228 are considered high quality enough for light curve fitting, and have light curve parameters within the typical ranges used in cosmology analyses.¹ With each spectral model (SALT2 and each SNEMO model), we obtain a vector of model parameters for each supernova by minimizing

$$\chi^2 = \sum_{\lambda,p} f_{\text{obs}}(\lambda, p) - f_{\text{mod}}(\lambda, p | \Theta) \quad (4.4)$$

with respect to the model parameters Θ , where f_{obs} is the observed flux (after correcting for Milky Way dust extinction), and f_{mod} is the model flux from Eqn. 4.1 for SALT2 or Eqn. 4.3 for the SNEMO models.

The overall scaling parameter (x_0 for the SALT2 model and c_0 for the SNEMO models) measured for each supernova depends not only on the absolute magnitude of the object, but also on the object's redshift. We would like our generative model to be able capture the range of intrinsic absolute magnitude fluctuations irrespective of the redshift and to include correlations between these fluctuations and the spectral model parameters. To accomplish this, we convert these scaling parameter value to a value representing the absolute magnitude of each object (which we denote by M_B^*) by calculating the apparent magnitude in the Bessel B-band at maximum brightness of a supernova with the best-fit x_1 and c (or c_i and A_s) values and subtracting the distance modulus to an object at the same redshift assuming a fixed, fiducial Λ CDM cosmology with $H_0 = 70$ km/s/Mpc and $\Omega_m = 0.3$. To simplify our plots and our kernel density estimates, we subtract a typical value of this parameter ($\langle M_B^* \rangle \approx -19.1$).

4.4 Kernel Density Estimation

With these measurements in hand, we move on to estimating the underlying joint probability distribution of these observations. Kernel density estimation is one such method for making these approximations. Simply put, provides an estimate fo the joint probability distribution by weighting each of the observed data points by some kernel function $k(\mathbf{x}_1, \mathbf{x}_2)$ and summing these weights to produce a smooth curve representing the probability density across parameter space (Silverman, 1986; Wand & Jones, 1995; Scott, 2014). Although the resulting estimate of the latent probability distribution is non-parametric, the kernel function used is usually parametrized by a parameter known as the bandwidth. This hyperparameter controls the level of smoothing by controlling the relative weights of data points spread further apart; in the case of a one-dimensional Gaussian kernel, this is the standard deviation of the Gaussian, σ .

Typically, this parameter is determined through k -fold cross-validation. In k -fold cross validation, the sample is first split into k groups. One of these groups is held out while the model is trained on the data in the remaining groups, and then evaluated via some scoring

¹This analysis is based on the CASCADe production of the SNfactory pipeline.

metric on the held-out set. This process is repeated for a range of model hyperparameters, and the best values of the hyperparameters are those that maximize the average score on the held out set. In our case, we use the sum of log likelihoods of the test data under the model fit using the training data as our scoring metric.

In Appendix C.1, we provide a detailed example of using this process to fit a KDE to a one-dimensional probability distribution. Finding the proper bandwidth matrix to fit a KDE in multiple dimensions is slightly more complicated. However, we can reduce the problem of bandwidth selection in many dimensions to the problem of bandwidth selection in one dimension by finding a whitening matrix, which defines a linear transformation of the data so that the covariance matrix of the transformed data is proportional to the identity matrix. Applying this transformation, fitting a KDE with the ideal bandwidth for this transformed data, and inverting the transformation gives us the desired fit. Appendix C.2 contains a detailed two-dimensional example, including a proof of a form of the whitening matrix.

4.5 Modeling the Data and Making Mock Observations

Comparing Parameter Distributions

We apply the multidimensional KDE fitting process with 5-fold cross validation of the bandwidth to the spectral model parameter measurements found in Section 4.3. Corner plots of the data points and similarly sized samples drawn from the resulting transformed KDE are shown in Figs. 4.1-4.4.

By eye, the marginal probability distributions of the KDE samples appear to match the data quite well across all models. This is true even of the skewed parameter distributions, like those corresponding to the color terms in the model (SALT2 c and SNEMO A_s). The two-dimensional marginalized distributions are also visually close to those of the data, even when the distributions deviate from a Gaussian, as they do for c_1 and c_2 , for example, in SNEMO7. In Section 4.6, we quantify these qualitative descriptions by comparing the distributions of spectral feature indicators from the data to the distributions obtained from spectra simulated using the KDE model and a multivariate Gaussian model of the spectral model parameters.

Generating Mock Observations

With the modeled latent space in hand, we can easily obtain new SN Ia instances to use in further analyses by drawing from the underlying joint probability distribution, calculating the scaling coefficient x_0 or c_0 , and plugging the resulting parameters into Eqn. 4.1 or 4.3. This process gives us a grid of flux values across the model wavelength range (3305-8685 Å for the SNEMO models or 2000-9200 Å for SALT2) and phase range (-10 to +40 rest-frame

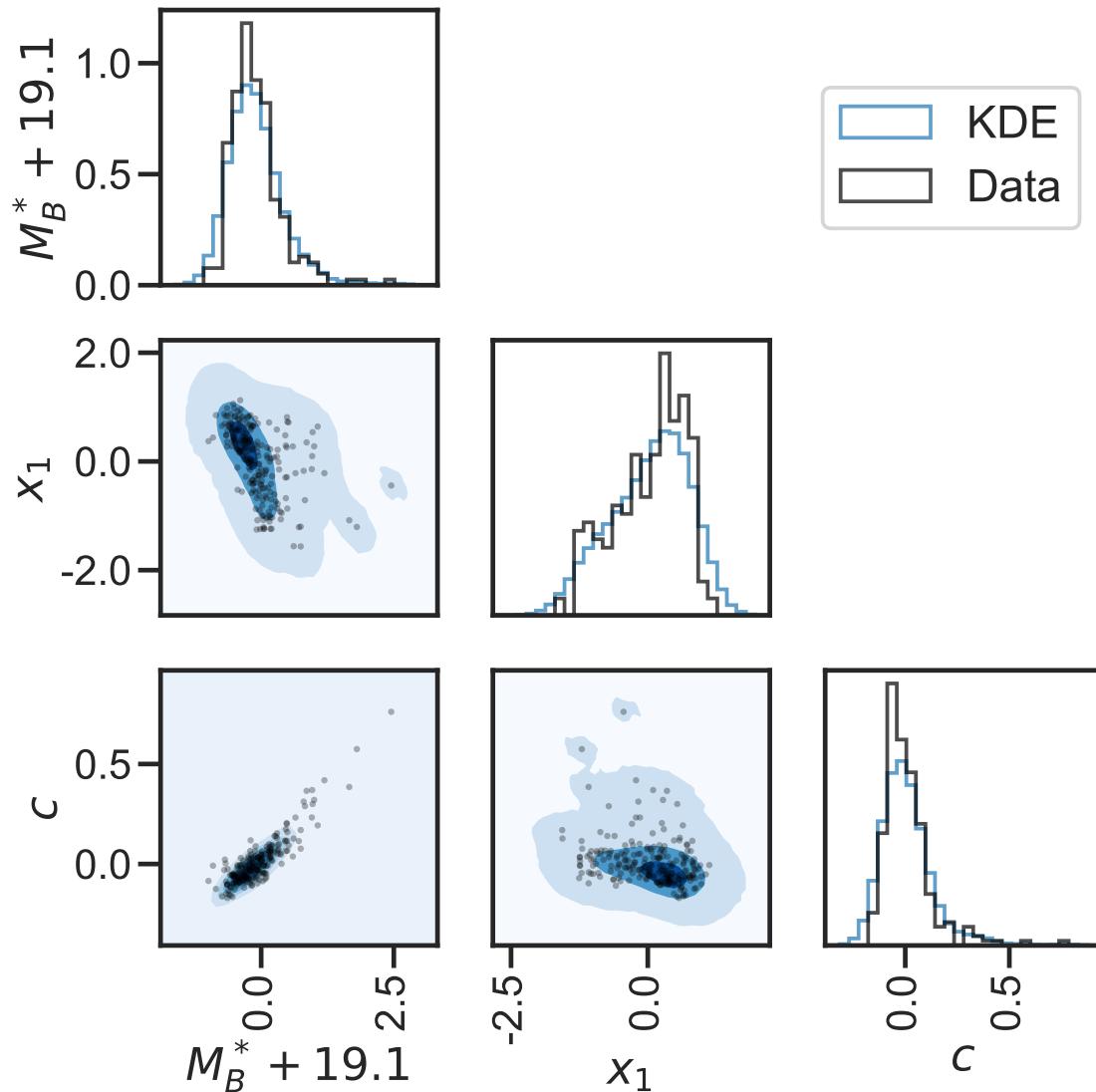


Figure 4.1: Corner plot showing the one- and two-dimensional marginal parameter distributions of the SALT2 parameters for the SNfactory data set (black points and histogram lines), as well as the 1-, 2-, and 3- σ confidence intervals of the marginalized distribution of samples drawn from the KDE trained on these data.

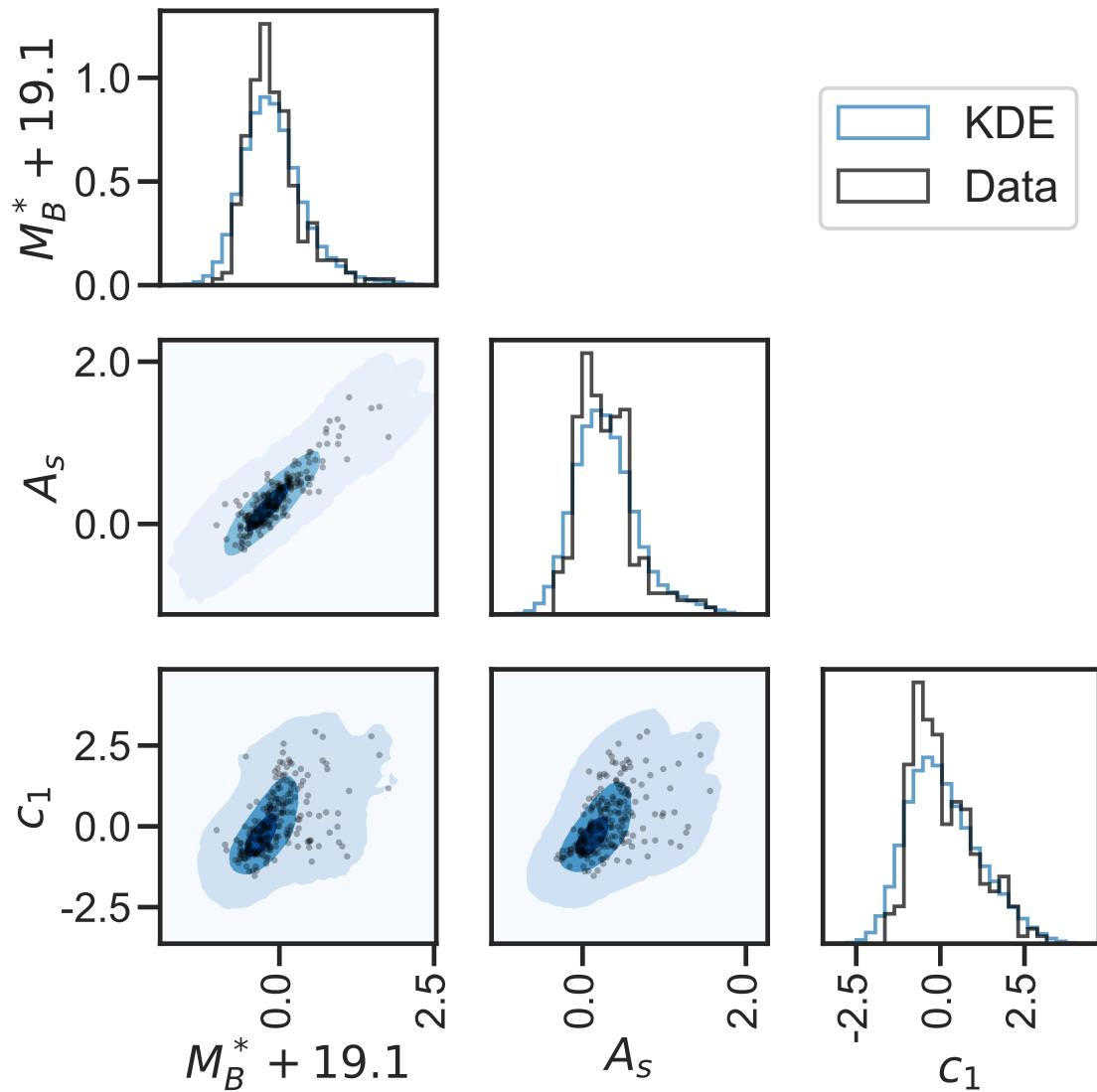


Figure 4.2: Same as Fig. 4.1, but for SNEMO2

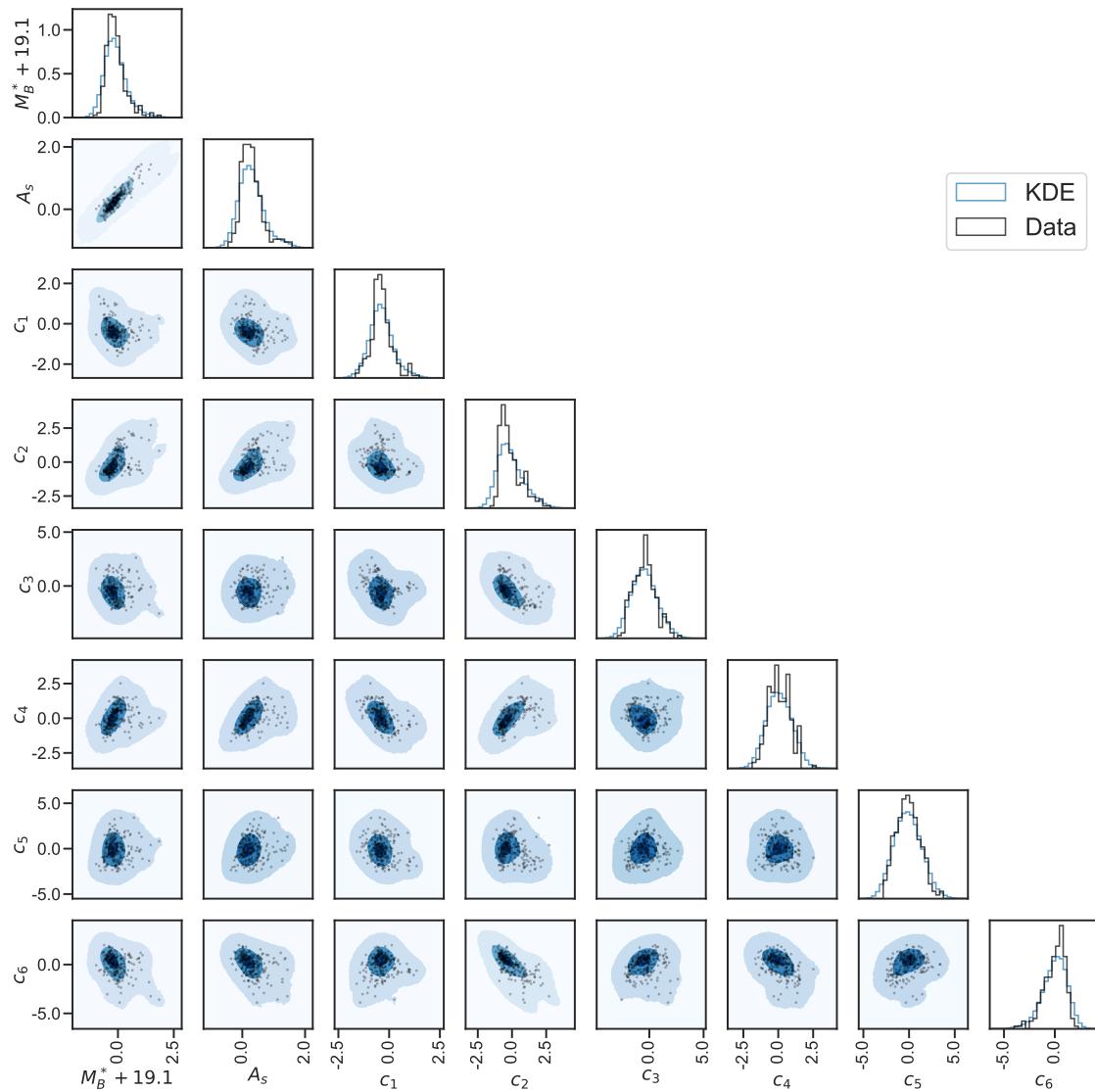


Figure 4.3: Same as Fig. 4.1, but for SNEMO7

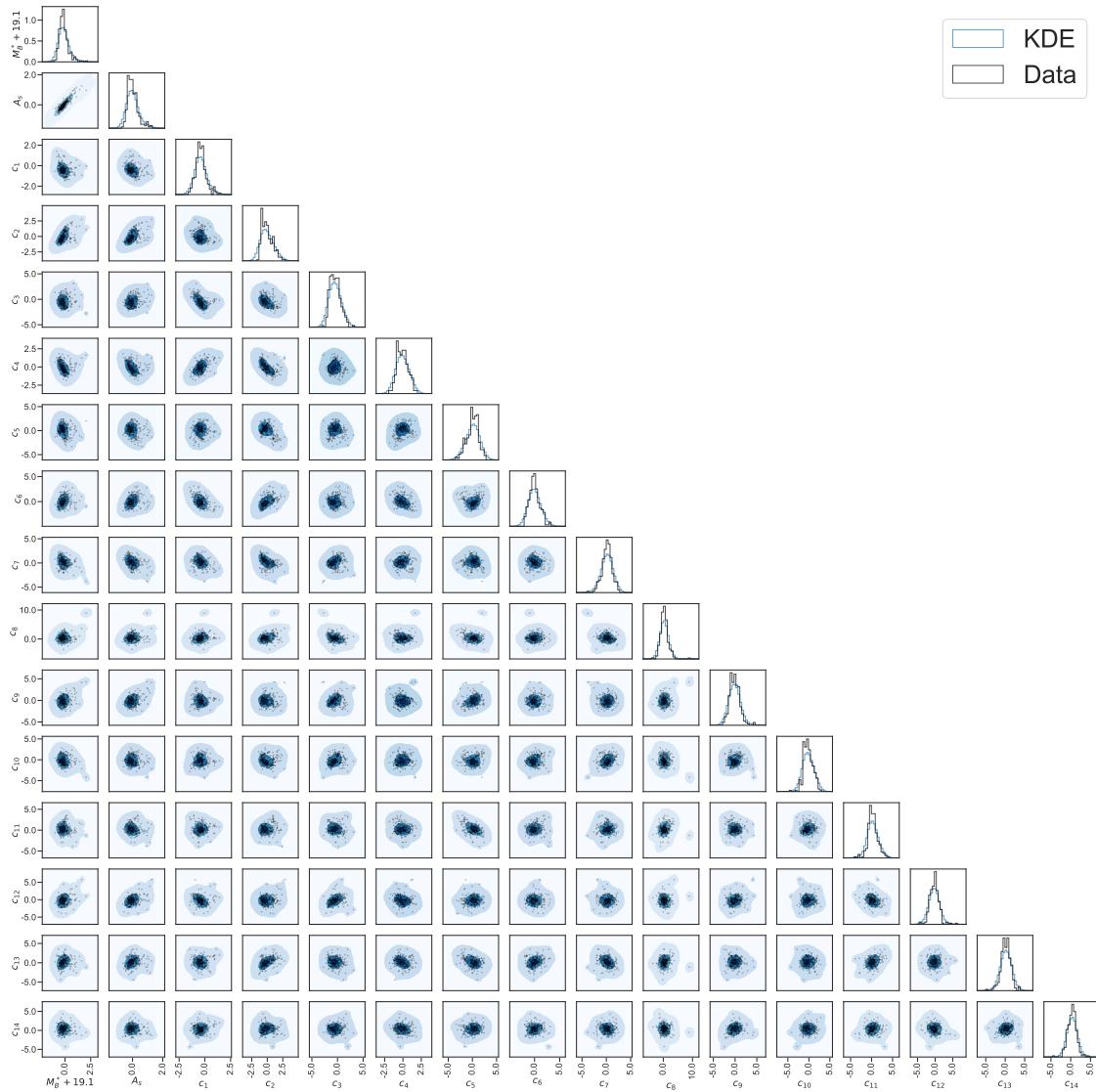


Figure 4.4: Same as Fig. 4.1, but for SNEMO15

days after maximum brightness for the SNEMO models or -20 to $+50$ rest-frame days for SALT2).

As explained in Section 4.3, we have modeled the absolute magnitude, rather than the redshift-dependent scaling parameters x_0 and c_0 . To convert the $M_B^* + 19.1$ value to x_0 or c_0 , we first choose a redshift z for the supernova instance based on the needs of our analysis and calculate m_B , the peak apparent magnitude of an object at that redshift with $c_0 = 1$ and all of the remaining parameters set to values determined by the draw from the modeled distribution. We also calculate the desired apparent magnitude, m_B^* , by adding the distance modulus $\mu(z)$ from our fiducial cosmology to the apparent magnitude M_B^* drawn from the modeled distribution. The final scale factor is then given by

$$c_0 = 10^{-0.4(m_B^* - m_B)}.$$

Once we have our grid of flux values $f_{mod}(\lambda, p)$, we can then easily synthesize spectroscopy or photometry of any resolution or signal-to-noise ratio using a tool like `sncosmo`.² In Fig. 4.5, we show spectra of an example object at redshift $z = 0.775$ with intrinsic flux determined by a draw from the KDE model of the SNEMO15 model parameters, using the spectral resolution of the proposed Roman Space Telescope prism spectrograph and signal-to-noise equivalent to an exposure time of roughly one hour. Fig. 4.6 shows the same object but observed through photometry in the band passes proposed for the Roman Wide Field Instrument for a similar exposure time (Rubin, 2020b; Roman Space Telescope Reference Information, 2019). Code for generating these spectra and light curves is available publicly³.

4.6 Evaluating Spectral Diversity

As another means of quantifying the usefulness of this tool, as well as a concrete example of the kind of analysis that is uniquely enabled by both a non-parametric model of spectral model parameters and the use of spectral models with more degrees of freedom (i.e. SNEMO7 and SNEMO15), we compare the distributions of several spectral features measured from the training spectra to those obtained with data simulated by the techniques introduced in this work. The development of the SNEMO models was largely motivated by the recognition that spectral models like SALT2 do not capture the full range of spectral variation that is seen in Type Ia supernovae (Saunders et al., 2018). This study aims to quantify how well higher-dimensional linear models and non-parametric models of the latent parameter space of these linear models can capture the non-linear features that may provide a better understanding of supernova standardization and supernova physics.

We chose to focus on the velocities and pseudo-equivalent widths of the Ca II H&K doublet, the Si II $\lambda 5972$ line, and the Si II $\lambda 6355$ line at maximum brightness. As discussed in Chapters 1 and 3, these spectral indicators are commonly used in studies aiming to

²<https://sncosmo.readthedocs.io/en/v2.1.x/>

³https://github.com/sam-dixon/snemo_generator

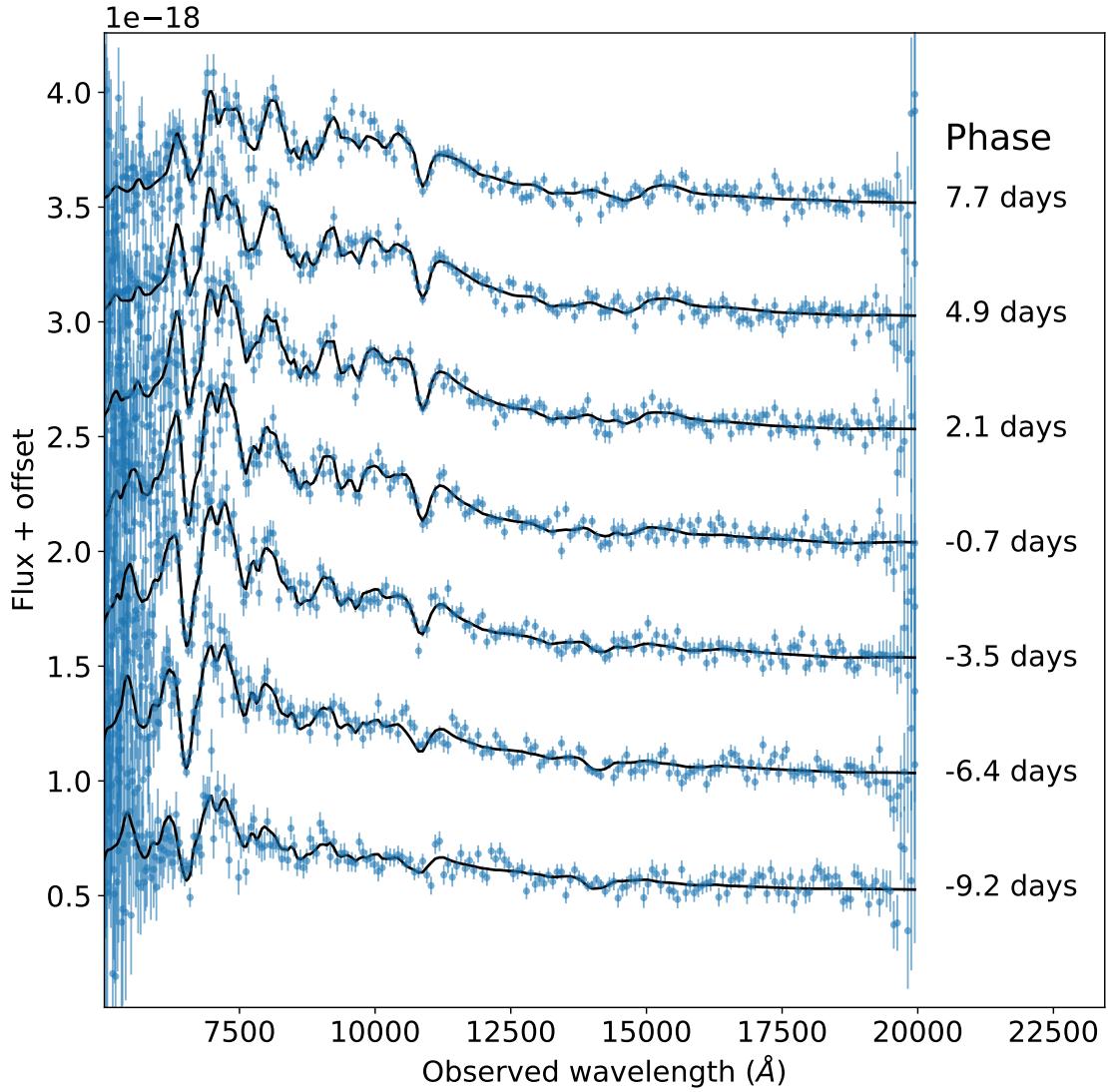


Figure 4.5: A series of synthesized spectral observations at a range of phases for a single object at redshift $z = 0.775$ generated from a random draw from the SNEMO15 KDE. The resolution matches the proposed design of the Roman Space Telescope prism spectrograph, and the signal-to-noise ratio representing the level that could be obtained with an hour of exposure time. The black lines represent the underlying spectral energy distribution generated by our model, and the blue points represent the flux values in each prism wavelength bin, along with their associated errors.

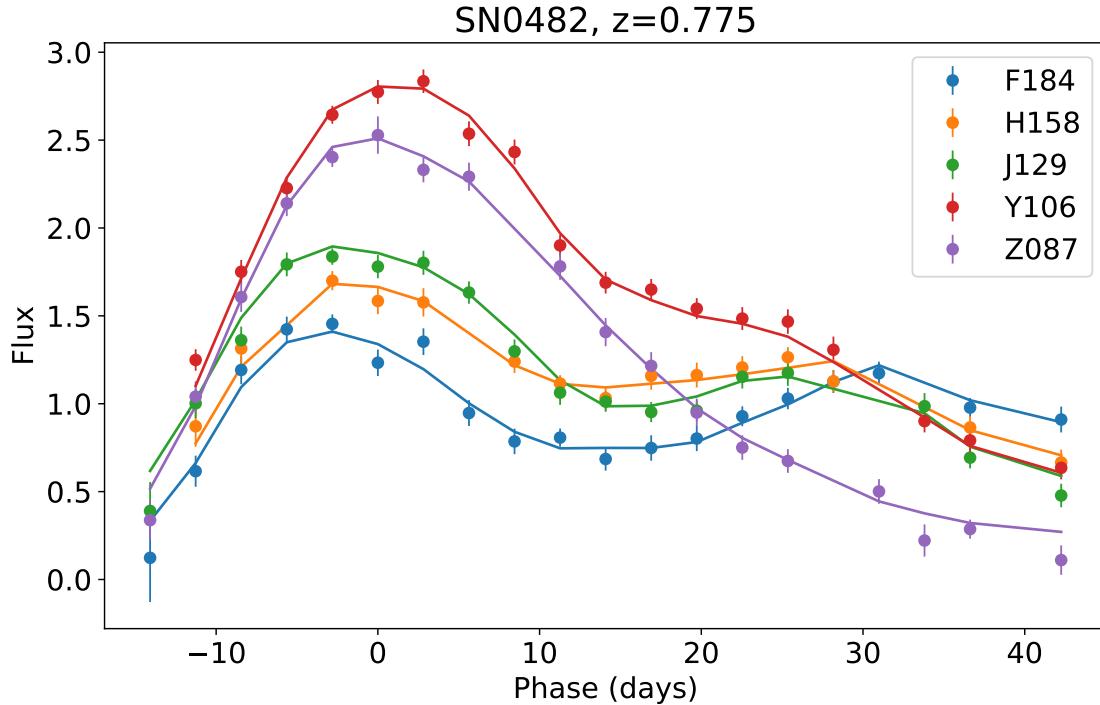


Figure 4.6: Synthetic photometry of the same object shown in Fig. 4.5, but observed photometrically in the Roman Wide Field Instrument bandpasses. Again, the lines represent the underlying flux of the generated supernova. The points represent an example of the noisy observed flux, and its associated errors.

improve the standardization of supernova brightnesses beyond light curve shape and color, or to quantitatively subclassify Type Ia supernovae in order to gain a better understanding of their physics. The ejecta velocities of SNe Ia, measured by the line velocities of Si II $\lambda 6355$ and Ca II H&K, have been shown to correlate with their intrinsic colors (Foley & Kasen, 2011; Foley et al., 2011; Foley, 2012; Mandel et al., 2014). The width of the Si II $\lambda 6355$ line shows a similar correlation (Foley et al., 2011). All of these relationships can lead to potential redshift-dependent distance bias if left uncorrected. An example subclassification scheme using these parameters is the Branch classification scheme (Branch et al., 2006), which arranges SNe Ia by the widths of their Si II $\lambda 5972$ and Si II $\lambda 6355$ lines, showing that there is a wide range in spectral feature behavior within the Ia class.

For the observed data, we measured these features using a method similar to Blondin et al. (2006), in which we first smooth the spectrum, then define a pseudo-continuum from the local maxima near the features in question. Using the smoothed, pseudo-continuum-removed flux, we use the relativistic Doppler formula to calculate the eject velocities, and integrate the flux to find the pseudo-equivalent width. When making measurements of the spectral features from the generated spectra, we proceed similarly, but skip the smoothing

step, as the spectrum is already assumed to be noiseless. Full details of this process are presented in Appendix D.

The spectral indicators were measured from the spectrum of each supernova in our data set that was closest to the SALT2-predicted time of B-band maximum brightness, t_0 . To minimize the impact of phase evolution of the features, we remove from the analysis all objects that do not have a spectrum within ± 5 rest-frame days of t_0 . This leaves 213 supernovae. Code to perform these measurements is publicly available through the `spectral_lines` package⁴.

Comparing Features Measured from Data and from Best-fit Spectral Models

We would like to separate our quantification of how well the spectral flux models themselves are able to capture the full distributions of spectral features from how closely samples generated from the KDE models of the spectral model parameters mimic the true distribution. To answer the first question, we make measurements of the spectral features from noiseless, at-max spectra synthesized directly from the best-fit spectral model parameters for each supernova and each spectral model. We will refer to these measurements as the model measurements.

Fig. 4.7 shows histograms of the residuals between these model-measured velocity of Si II $\lambda 6355$ (v_{6355}) and the data-measured velocity for each object in our data set. We can see that as the number of model components increases, the scatter on these residuals decreases, indicating that the increased flexibility of these higher-dimensional spectral models allows them to capture these spectral features. For SNEMO15, the average difference between the model and data measurements of the velocity of this line is comparable in size to the typical measurement error of this feature.

This general trend is seen across all the spectral indicators studied, with the exception of the width of the Si II $\lambda 5972$ line; we can see this in Fig. 4.8, where we compare the normalized median absolute deviation (NMAD)⁵ of the residuals between model and data spectra for each spectral indicator across spectral models. It is not immediately obvious why the width of the Si II $\lambda 5972$ line is captured nearly as well by SALT2 and SNEMO2 as it is by SNEMO15, but not captured by SNEMO7. It may be due to the fact that it is a relatively small feature, and therefore both more difficult to measure precisely on the data spectrum and poorly sampled in the SNEMO spectral eigenvectors. Regardless, we find that SNEMO15 is the best at capturing all of the spectral indicators that we studied.

⁴https://github.com/sam-dixon/spectral_lines

⁵ $NMAD(\mathbf{x}) = 1.4826 \text{ median}(|\mathbf{x} - \text{median}(\mathbf{x})|)$

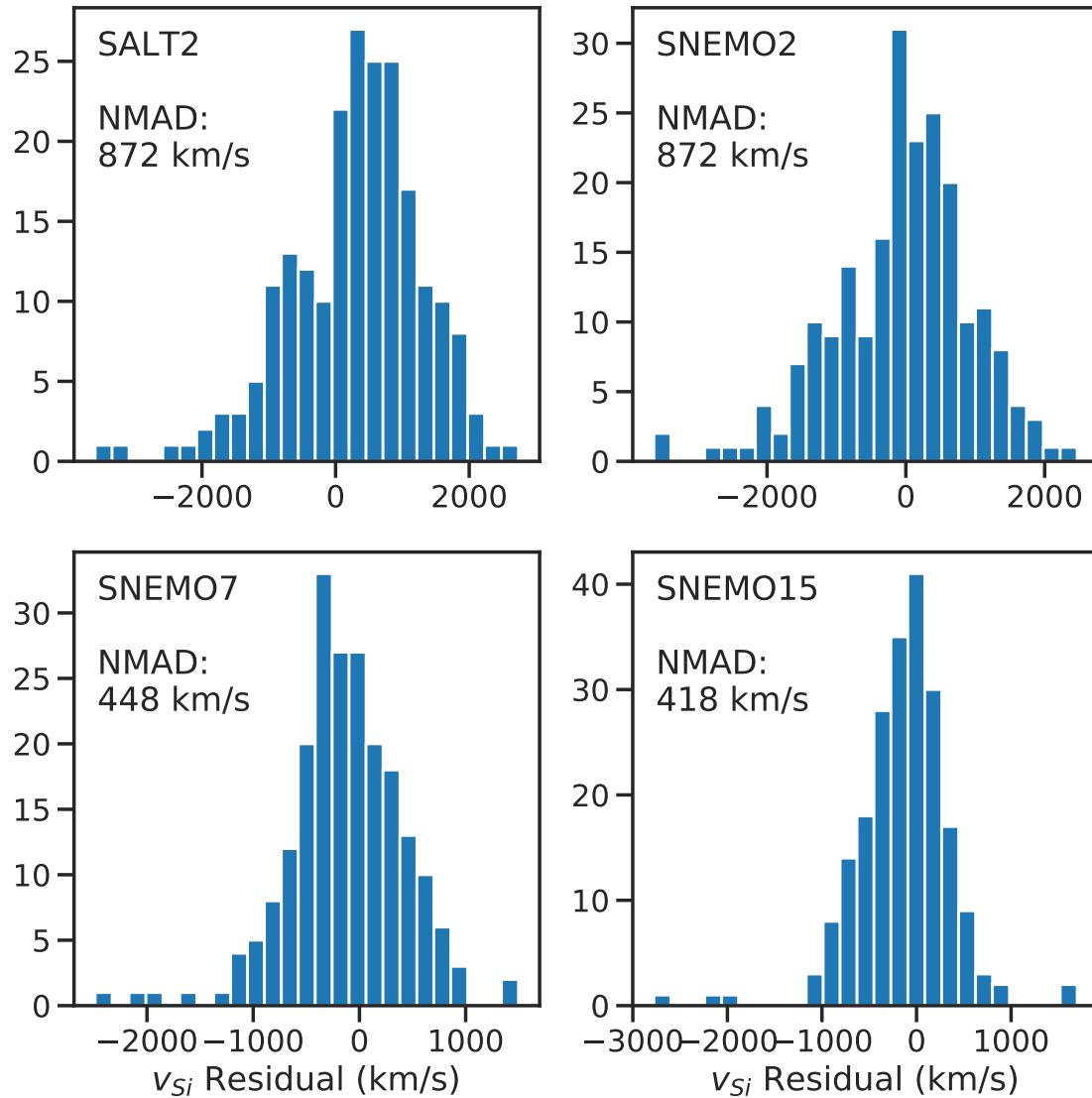


Figure 4.7: Histograms of the residuals between the velocity of the Si II $\lambda 6355$ line as measured from the data and as measured from the spectrum generated with the best-fit spectral model parameters. Lower dimensional models (SALT2 and SNEMO2)

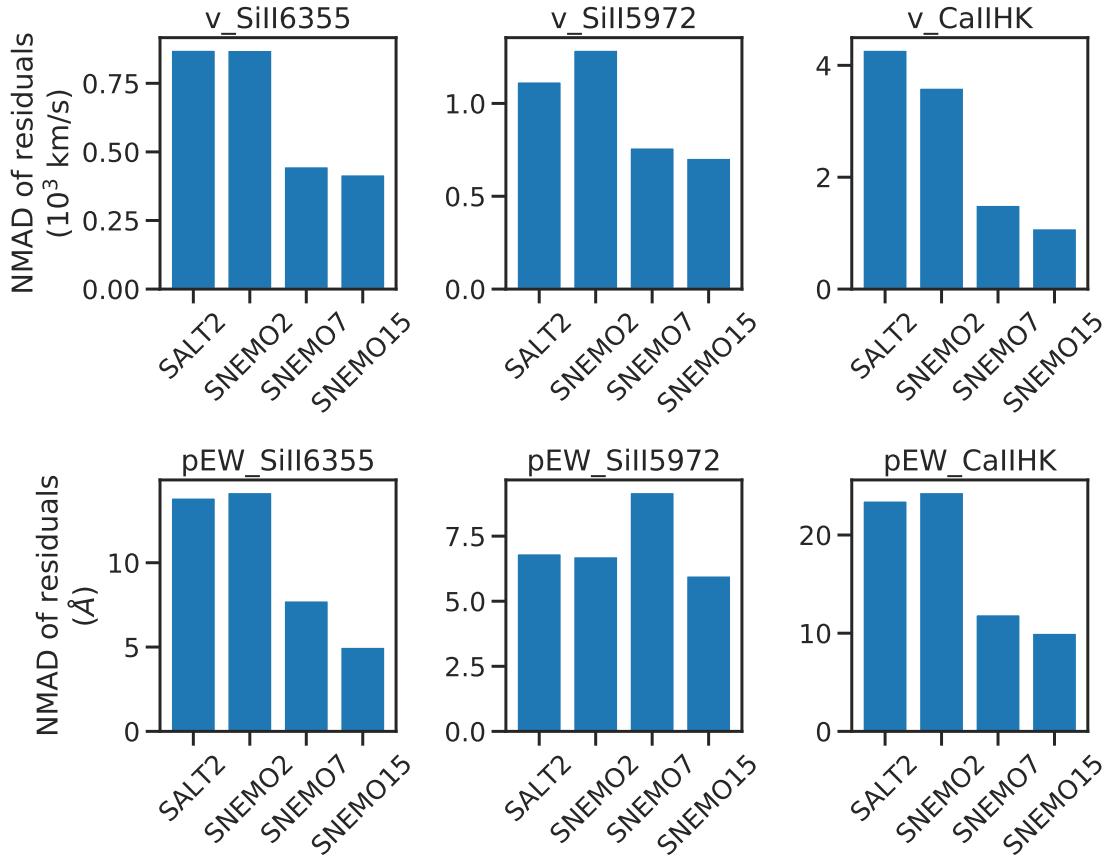


Figure 4.8: Normalized median absolute deviation of residuals between spectral features measured from data spectra and from spectra generated from the best-fit spectral model. In general, spectral models with more parameters more closely capture the spectral feature measurements.

Comparing Spectral Feature Distributions

We can also compare the distributions of the spectral indicators measured from spectra generated by the KDE to those measured from spectra in our data set. To do so, we generate 1000 noiseless, at-max spectra from the KDE model of the spectral feature parameters and from a multivariate Gaussian model fit to the same parameters using a standard maximum-likelihood analysis, and measure the same six spectral indicators (velocity and equivalent width of Si II $\lambda 6355$, Si II $\lambda 5972$, and Ca II H&K) for each of these spectra.

The empirical cumulative distributions of the data and the KDE distributions for each of the spectral models are shown in Fig. 4.9. A similar plot, but using the multivariate Gaussian model of the spectral model parameter space, is found in Fig. 4.10. These figures look quite similar, though we can pick out some differences (like the difference in the lower velocity portion of SNEMO15 distribution of $v_{\text{Si II } \lambda 5972}$, or the differing relative fractions in

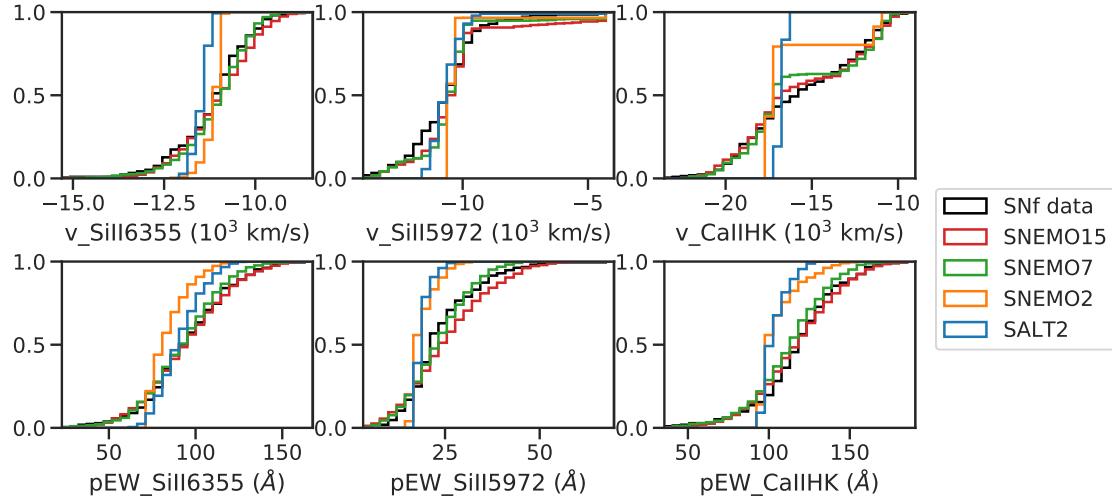


Figure 4.9: Empirical cumulative distribution functions of each spectral indicator for the data set and samples from the kernel density estimate of the spectral model parameter spaces.

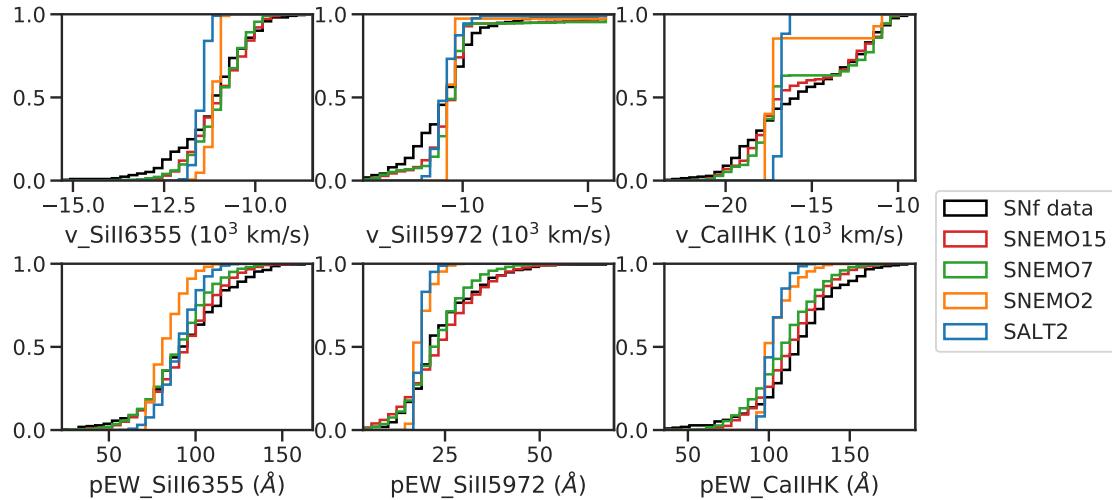


Figure 4.10: Same as Fig. 4.9, but for samples from the multivariate Gaussian estimation of each of the spectral model parameter spaces.

each mode of the bimodal $v_{\text{Ca II H\&K}}$ distributions for SNEMO15).

We can quantify these differences by calculating the two-sample Cramér distances between the distributions of the data and the distributions of the samples for each feature. The Cramér distance ω between a probability distribution with the empirical cumulative distribution function $F(x)$ and a second probability distribution with the empirical cumulative

distribution function $G(x)$ is defined by

$$\omega^2 = \int_{-\infty}^{\infty} [F(x) - G(x)]^2 dx \quad (4.5)$$

Smaller values indicate closer agreement between the two distributions. This distance metric has the advantage over other more well-known test statistics (like the Kolmogorov-Smirnov statistic) of being sensitive to differences in the shape of distributions beyond just shifts in the global mean or standard deviation, particularly in the tails of the distribution. This is ideal for our task because we want our parameter space estimates to match the true distribution across all portions of parameter space. We can also get obtain an estimate of the error on these distances via bootstrap resampling.

The calculated Cramér distances between the KDE and Gaussian estimates are shown, along with the similarly calculated model-to-data distances, in Fig. 4.11. We see a pattern similar to Fig. 4.8 in the spectral feature distribution similarity across models – in every case, spectral models with more parameters have distributions of the spectral indicators that more closely resemble the data. Additionally, for each of the spectral models, the kernel density estimate of the model parameter distributions creates spectral feature distributions that are as or more similar to the true data distribution than the Gaussian estimates and the best-fit parameter spectra. Overall, this shows that using more flexible spectral models along with more flexible parameter space models allows for a generative model that can accurately reproduce the full range of spectral behavior for simulations.

4.7 Conclusions

We have presented flexible estimates of the joint probability distributions of model parameters for the SALT2 and the SNEMO models of Saunders et al. (2018). These estimates can be used to generate synthetic spectra and photometry in simulations that exhibit more spectral diversity than current state-of-the-art simulation techniques. This increased variety makes possible a number of different analyses, from examining the robustness of the twinning technique presented in Fakhouri et al. (2015), to evaluating spectral feature measurement techniques under different observing conditions. There are a number of spectral properties of Type Ia supernovae beyond the two-parameter light curve shape and color parameters that have been shown to ultimately effect our cosmological parameter measurements. This work presents a simulation tool that properly incorporates these variations, allowing us to properly understand their impacts for future cosmological surveys.

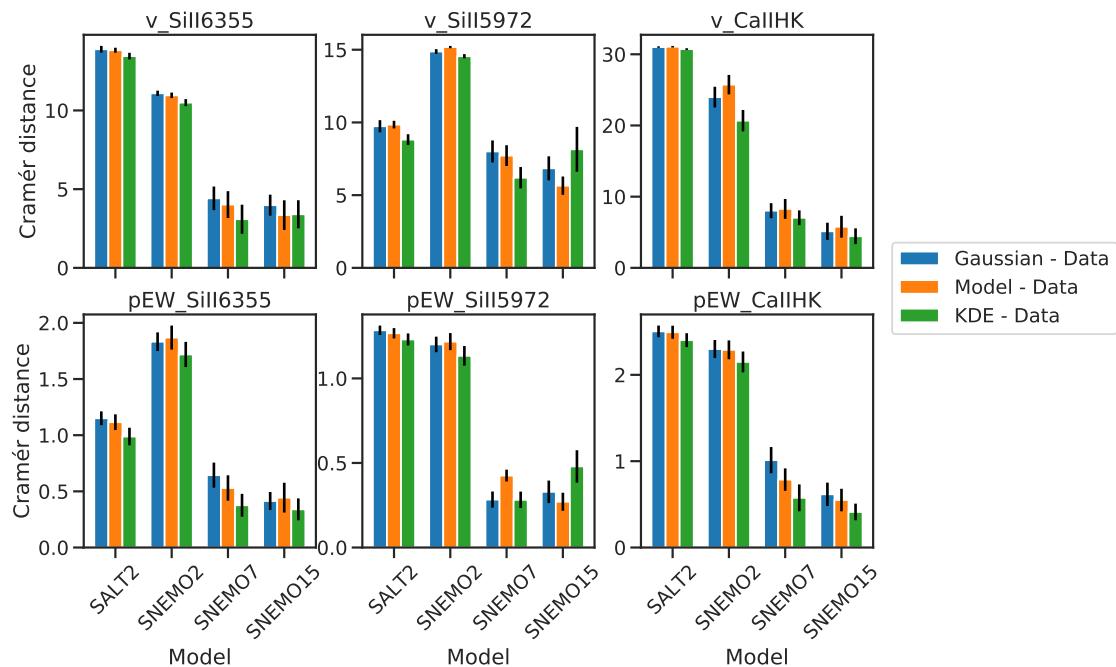


Figure 4.11: Cramér distances between spectral indicator distributions for the SNfactory data and samples from the Gaussian estimates of the SALT2 and SNEMO parameter distributions, the modeled at-max spectra for the training data, and the KDE estimates of the spectral model parameter distributions.

Chapter 5

Biases from Non-Simultaneous Regression with Correlated Covariates in Supernova Cosmology

5.1 Overview

As discussed in Chapter 1, the process of using various properties of Type Ia supernova observations to correct their absolute magnitudes is the key to using supernovae as cosmological distance indicators. This technique was instrumental in the discovery of the accelerating expansion of the Universe (Perlmutter et al., 1999; Riess et al., 1998), and continues to serve as a powerful probe of the nature of the dark energy driving this acceleration.

A common analysis method for standardizing supernova brightnesses uses the SALT2 spectral model (Guy et al., 2007; Betoule et al., 2014; Mosher et al., 2014) to parametrize SN Ia light curves. The model parameters represent an individual supernova’s peak apparent brightness in the Bessell B-band (m_B^*), temporal width (x_1), and observed color (c). The distance modulus μ to each object i at redshift z_i is then modeled as a linear combination of these parameters:

$$\mu_i(z_i) = m_{B,i}^*(z_i) - M + \alpha x_{1,i} - \beta c_i \quad (5.1)$$

Typically, we would find the values of M , α , and β by minimizing the following quantity with respect to these parameters as well as the cosmological parameters of interest:

$$\chi^2 = \sum_i \frac{\mu_i(z_i; m_{B,i}^*, x_{1,i}, c_i) - \mu_{\text{cosmo}}(z_i; \Theta)}{\sigma_{\text{obs},i}^2 + \sigma_{\text{int}}^2}, \quad (5.2)$$

where $\mu_{\text{cosmo}}(z_i; \Theta)$ is the distance modulus-redshift relation determined by the cosmological parameters Θ , and $\sigma_{\text{obs},i}$ is the observational uncertainty of the measurements. σ_{int} is the intrinsic dispersion of standardized magnitudes, usually found by iteratively calculating the

value of σ_{int} that ensures the minimum value of χ^2 is equal to 1.¹ This process is effectively a familiar linear regression.

The need to add an additional uncertainty term in the form of σ_{int} suggests that the linear relationship between SALT2 parameters and absolute magnitude may not capture all of the variation in supernova magnitudes, or that the parametrization provided with SALT2 may not capture all of the information that is needed to fully standardize supernova magnitudes (Saunders et al., 2018). This motivates the search for other observable properties of SNe Ia that might explain this remaining variation, as well as the use of these other properties for standardization. One way to search for such properties is to measure correlations between these properties and the Hubble residuals $\mu_i(z_i; m_B^*, x_1, c) - \mu_{\text{cosmo}}(z_i; \Theta)$. A number of studies (Kelly et al., 2010; Lampeitl et al., 2010; Sullivan et al., 2010; Childress et al., 2013) have observed such a correlation with the host galaxy stellar mass: supernovae in galaxies with $\log(M/M_\odot) > 10$ are ~ 0.1 magnitudes brighter after standardization than supernovae in galaxies with $\log(M/M_\odot) < 10$. Rigault et al. (2013), Childress et al. (2014), and Rigault et al. (2015) show that this effect could be due to similar correlations with host galaxy age. However, the significance of some of these correlations has been debated (e.g. Jones et al., 2015, 2018), indicating that care must be taken in making these measurements.

Reporting the size of correlations with the linear regression residuals is mathematically well-motivated if the covariate used to predict these residuals is not itself correlated with those used in the original regression (if, for example, host mass were not correlated with light curve parameters). However, if this key assumption is violated, we find ourselves in a situation referred to in the statistics and econometrics literature as multicollinearity (e.g. Farrar & Glauber, 1967). Multicollinearity results in unreliable and biased estimates of effect sizes. A related concern discussed frequently in these fields is omitted variable bias, in which a misspecification of the regression problem results in biased estimates of the true regression parameters (Clarke, 2005; Wooldridge, 2013). Indeed, Smith et al. (2020) shows some evidence of these effects in supernova cosmology, by examining the bias on the measurement of the host galaxy mass step (which in turn biases estimates on the dark energy equation-of-state parameter) due to the correlation between host galaxy mass and SALT2 x_1 . Rigault et al. (2018) has also presented two different values of the size of the luminosity difference between supernovae in environments with differing star-formation rates when measuring the step with a sequential regression versus a simultaneous regression, using the larger, simultaneously-determined effect size as their main result.

In this work, we explore and quantify the general impact of the non-simultaneous regression methodology used in some Type Ia supernova analyses on reported effect sizes for both linear and step-function residual trends when multicollinearity exists. In Section 5.2, we work through an example using a generalized two-dimensional linear regression problem with correlated covariates. In Section 5.3, we analyze a similar model that includes a step function and compare the results to those obtained in the linear case. We then calculate the

¹Equation 5.2 is equivalent to Equation 1.15, assuming the covariance matrix is diagonal and contains just the measurement and intrinsic error terms

effect that this general mathematical model has in the particular case of measuring the host galaxy mass step using literature data of SALT2 parameters and host galaxy masses in Section 5.4, and conclude in Section 5.5 by identifying previous results that have overlooked this effect and recommending that future analyses use fully simultaneous regression techniques.

5.2 Toy Model: Two-dimensional Linear Regression with Correlated Covariates

We consider the following toy model: A series of n observations² $\{(x_1^{(1)}, x_2^{(1)}), \dots, (x_1^{(n)}, x_2^{(n)})\}$ is drawn from a two-dimensional Gaussian distribution with $\mu = (0, 0)$ and a covariance matrix given by

$$\Sigma = \begin{pmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{pmatrix} \quad (5.3)$$

σ_1 and σ_2 are the standard deviations of the observations in the x_1 and x_2 dimensions, respectively, and ρ is the Pearson correlation coefficient between them. They are not measurement errors, but measures of the natural spread in the distributions. We then define

$$y_i = \beta_1 x_1^{(i)} + \beta_2 x_2^{(i)} + \epsilon^{(i)} \quad (5.4)$$

where β_1 and β_2 are the regression coefficients, and ϵ is a noise vector drawn from a univariate normal distribution $\mathcal{N}(0, \sigma_{\text{int}}^2 + \sigma_{\text{obs}}^2)$. This noise vector represents a combination of the intrinsic scatter in the model, as well as the observational measurement error. We can reformulate this as a matrix equation by denoting the data matrix as $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2)$ and the coefficient vector as $\boldsymbol{\beta} = (\beta_1, \beta_2)$, giving $\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$.

Standard simultaneous two-dimensional least-squares regression gives us the estimated coefficient vector $\hat{\boldsymbol{\beta}}$ which minimizes the square of the residuals between the values predicted by the model ($\hat{\mathbf{Y}} \equiv \hat{\beta}_1 \mathbf{x}_1 + \hat{\beta}_2 \mathbf{x}_2 \equiv \mathbf{X}\hat{\boldsymbol{\beta}}$) and the data. As we show in Appendix E.1, these estimated values are

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T (\mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}) \quad (5.5)$$

Since the expectation value of $\boldsymbol{\epsilon}$ is 0 by definition, the expectation value of the recovered coefficients from simultaneous regression is identical to the coefficients ($\langle \hat{\boldsymbol{\beta}} \rangle = \boldsymbol{\beta}$) regardless of the values of the regression coefficients, the covariance matrix components, or the size of the intrinsic scatter. We also show in Appendix E.1 that the standard deviation of the residuals ($\mathbf{r} \equiv \hat{\mathbf{Y}} - \mathbf{Y}$) is simply $\sqrt{\sigma_{\text{int}}^2 + \sigma_{\text{obs}}^2}$.

In summary, when treating this data set with a simultaneous linear regression, we are able to reliably recover both the true regression coefficients and intrinsic dispersion. Though there is some uncertainty on the values of the regression coefficients that does depend on the correlation between the covariates, this uncertainty is also inversely proportional to the

²Note that x_1 here is just the vector of observations in the first dimension and should not be confused with the x_1 parameter of the SALT2 model discussed in Equation 5.1.

number of samples fit in the regression and is therefore able to be controlled in the case where N is sufficiently large (see Equation E.2 in Appendix E.1).

However, as described above, oftentimes in Type Ia supernova studies, we do not perform a full simultaneous fit of all of our regression parameters. Instead, we fit the distance modulus as a linear function of SALT2 parameters and then add a correction to these distance moduli by fitting the distance modulus residuals as a function of some other parameter. This can be thought of as being analogous to performing this multivariate linear regression one covariate at a time.

We will show that in this case, no biases are introduced if there is no correlation between the parameters used in the first regression and second regressions (i.e. $\rho = 0$). However, if there is some correlation, we find that both the regression coefficients and the estimated scatter on the residuals are biased.

We introduce the notation we will use to treat this situation in our toy example. Without loss of generality, we can first fit \mathbf{Y} as a function of \mathbf{x}_1 . The estimate of the slope will be denoted $\hat{\beta}'_1$ (the prime serves to differentiate this value from the coefficient estimated from the full two-dimensional regression). The residuals of this regression will be denoted \mathbf{r}_1 . We then perform a second regression, predicting the residuals of the first regression \mathbf{r}_2 as a function of \mathbf{x}_2 . The slope in this case will similarly be denoted $\hat{\beta}'_2$, and the residuals will be denoted by \mathbf{r}_2 .

In Appendix E.2, we obtain the forms of the expectation values for the regression coefficients resulting from this process, finding that

$$\langle \hat{\beta}'_1 \rangle = \beta_1 + \frac{\beta_2 \rho \sigma_2}{\sigma_1} \quad \text{and} \quad \langle \hat{\beta}'_2 \rangle = \beta_2 - \beta_2 \rho^2.$$

As we can see, both slopes are biased if $\rho \neq 0$. The size of the bias on both parameters is proportional to the size of the effect and the correlation between the covariates. Additionally, we can recognize that the bias on the first slope is identical to the omitted variable bias. This is expected, as performing this first part of the non-simultaneous regression perfectly simulates the textbook situation presented to describe the omitted variable bias.

We also calculate the spread of the final residuals in Appendix E.2, finding

$$\sigma_{\mathbf{r}_2}^2 = \beta_2 \rho^2 \sigma_2^2 (1 - \rho^2) + \sigma_{\text{int}}^2 + \sigma_{\text{obs}}^2 \quad (5.6)$$

The standard deviation on the residuals from this analysis, often reported as the root-mean-squared (RMS) residuals, is in fact inflated by a value that scales quadratically with the correlation between the parameters and linearly with the size of the secondary effect. This bias is maximized for a given slope when $\rho = \sqrt{1/2} \approx 0.707$.

5.3 Step Function Corrections

Many common analyses used in supernova cosmology do not use a linear model to correct the Hubble diagram residuals for host mass; they use a step function, motivated by the evolution

of host galaxy stellar populations with redshift.³ We'll modify the toy model presented in Section 5.2, and consider instead

$$y_i = \alpha x_1^{(i)} + \frac{\gamma}{2} \text{sgn}(x_2^{(i)}) \quad (5.7)$$

In the simultaneous case, the expected values of the best-fit regression coefficients $\hat{\alpha}$ and $\hat{\gamma}$ are equivalent to the true values. The proof of this is very similar to the proof for the bilinear toy model presented in Appendix E.1, so we do not present any further details here.

In Appendix E.3, we have worked through the non-simultaneous case where we fit the linear relationship first, followed by the step function correction to the resulting residuals. The expectation value of the best-fit linear slope ($\hat{\alpha}'$) is

$$\langle \hat{\alpha}' \rangle = \alpha + \frac{\gamma \rho}{\sigma_1 \sqrt{2\pi}} \quad (5.8)$$

The expected step size obtained from the residuals after correcting for the linear relationship is

$$\langle \hat{\gamma}' \rangle = \gamma \left(1 - \frac{2\rho^2}{\pi} \right), \quad (5.9)$$

and the spread of the remaining residuals is

$$\sigma_{r_\beta}^2 = \frac{\gamma^2 \rho^2}{2\pi} \left(1 - \frac{2\rho^2}{\pi} \right) + \sigma_{\text{int}}^2 + \sigma_{\text{obs}}^2 \quad (5.10)$$

So, using a step-function secondary correction gives us similar biases to the linear secondary correction. The size of the step is underestimated by a factor that scales quadratically with the correlation coefficient between covariates and linearly with the true step size. Additionally, the size of the linear correction term is overestimated by a factor that scales linearly with the step size and the correlation coefficient. Finally, the variance of the residuals after correction is inflated by a factor that scales similarly. The bias on the variance of the residuals is maximal when $\rho = \sqrt{\pi/4} \approx 0.886$.

5.4 Comparison to Data

The remaining difference between our toy models and the actual data is that the true distributions of x_1 , c , and $\mathcal{M}_{\text{host}}$ are not purely Gaussian. While we cannot derive closed-form relations describing the impact of non-simultaneous fitting, we can simulate these effects. In this analysis, we take published values of x_1 , c , and $\log(\mathcal{M}_{\text{host}}/\mathcal{M}_\odot)$ from the low- and

³In order to maintain the differentiability of this function, some analyses approximate a step function with a logistic function with a large growth rate. To ease our calculations (particularly in calculating the expected covariance in Equation E.3), we use the sign function. The differences between the two are negligible for our purposes.

mid-redshift samples of supernovae from the first three years of the Dark Energy Survey (Abbott et al., 2019, hereafter referred to as the Low- z and DES subsamples), along with the Pantheon data set (Scolnic et al., 2018), which combines spectroscopically-classified supernovae from PanSTARRS supernovae (PS1; Rest et al., 2014; Scolnic et al., 2014) with supernovae from the SuperNova Legacy Survey (SNLS; Conley et al., 2011; Sullivan et al., 2011) and the Sloan Digital Sky Survey (SDSS; Frieman et al., 2008; Kessler et al., 2009; Sako et al., 2018).⁴ Each of these data sets shows a fairly strong correlation between x_1 and host mass, as seen in Table 5.1, so we can expect to find some non-simultaneous regression biases.

Data set	$\rho_{x_1,c}$	$\rho_{x_1,\text{mass}}$	$\rho_{c,\text{mass}}$
DES	-0.087	-0.371	0.1811
PS1	-0.041	-0.248	0.0610
SDSS	-0.035	-0.297	0.0002
SNLS	0.016	-0.304	0.0629
Low- z	0.130	-0.347	-0.1052

Table 5.1: Pearson correlation coefficients between SALT2 parameters and host galaxy masses (measured as $\log(\mathcal{M}_{\text{host}}/\mathcal{M}_\odot)$). Each data set shows a relatively strong correlation between x_1 and mass, indicating that biases can be introduced from non-simultaneous regression.

To simulate the magnitude of these effects with non-Gaussian distributions, we begin by modeling δ , a quantity akin to the Hubble residuals without any corrections for the light curve shape or color parameters and assuming a fixed cosmology:

$$\delta = M + \alpha x_1 + \beta c + \frac{\gamma}{2} \text{sgn} \left[\log \left(\frac{\mathcal{M}_{\text{host}}}{\mathcal{M}_\odot} \right) - 10 \right] + \epsilon \quad (5.11)$$

where ϵ is a Gaussian distributed noise vector with variance σ_{noise}^2 . For each data set, we calculate 50 instances of δ with different noise vectors for nearly 12,000 different combinations of α , β , γ , and σ_{int} in the ranges described in Table 5.2. We are motivated to simulate various combinations of the regression coefficients and noise values by the toy model, which showed that each of these values is intrinsically linked to the others. The overall magnitude value M was fixed to -19.1 , as the value of this offset in our model does not affect our results. For each of these simulated data sets, we perform both the full simultaneous linear and step function fit, as well as the non-simultaneous linear fit followed by a fit of the step function to the residuals of the linear fit. Note that in both cases the linear portion of the fit is done simultaneously, as is done in typical cosmology analyses.

⁴The DES and Low- z sample data can be downloaded at <https://des.ncsa.illinois.edu/releases/sn>, and the Pantheon data may be found at <https://archive.stsci.edu/prepds/ps1cosmo/index.html>.

Parameter	Range
α	(0.05, 0.25)
β	(2.5, 3.5)
γ	(−0.1, 0.1)
σ_{noise}	(0, 0.2)

Table 5.2: Ranges for the standardization hyperparameters used in the simulation analysis.

The results of these simulations are tables of true values of α , β , and γ , simultaneous best-fit values $\hat{\alpha}$, $\hat{\beta}$, and $\hat{\gamma}$, as well as non-simultaneous best-fit values $\hat{\alpha}'$, $\hat{\beta}'$, and $\hat{\gamma}'$, for each data set. Regardless of true parameter value, the simultaneous fit parameters all match the true parameters. However, the magnitude of the error on the non-simultaneous best-fit parameters depends on the data subset in question as well as on the true values of the parameters. The relationships are all linear, i.e.

$$\gamma = c_{\gamma,0} + \sum_{i \in \{\hat{\alpha}, \hat{\beta}, \hat{\gamma}\}} c_{\gamma,i} i, \quad (5.12)$$

where the c values are the linear coefficients relating the best-fit standardization parameters to their true values.⁵ Similar relationships exist for α and β as well. This is not unexpected; we see this linear relationship in our toy models as well (see Equation 5.8, for example). Non-zero values of coefficients other than $c_{x,0}$ indicate that there is “leakage” from one standardization parameter to the other; for example, if $c_{\gamma,\alpha} \neq 0$, then the size of the α correction impacts the reported size of the γ corrections. Moreover, these coefficients define a linear transformation between the true regression parameters and those coming from a non-simultaneous fit, so the inverse of these transformations can be used to correct previous non-simultaneous regressions. The transformations we obtained from our simulations are presented in Tables 5.3, 5.4, and 5.5.

We can see that there is significant leakage between the size of the host mass step and the stretch and color standardization parameters α and β . Multiplying the coefficients relating the non-simultaneously obtained step-size by the typical size of the measured step (0.07 mag.), we can see that this leakage results in a 5–10% error on the typical size (0.14) of the stretch parameter α and a ∼1% error on the typical size (3.0) of the color parameter β .

More importantly, the coefficients relating the non-simultaneous step size to the true step size are greater than one for each data set. This means that by fitting the step function separately from other corrections, the true size of the step is underestimated by 10–30%, and by a factor of two for the Low- z subsample.

⁵These coefficients are not to be confused with the SALT2 color parameter c (with no subscript).

Data set	α			
	$c_{\alpha,0}$	$c_{\alpha,\hat{\alpha}}$	$c_{\alpha,\hat{\beta}}$	$c_{\alpha,\hat{\gamma}}$
DES	0.000	1.000	0.000	0.335
PS1	0.000	1.000	0.000	0.135
SDSS	0.000	1.000	0.000	0.125
SNLS	0.000	1.000	0.000	0.203
Low- z	0.000	1.000	0.000	0.194

Table 5.3: Linear transformation coefficients (see Equation 5.12) between the standardization hyperparameter α , representing the light curve shape-luminosity correction, obtained with a non-simultaneous fit and the true values.

Data set	β			
	$c_{\beta,0}$	$c_{\beta,\hat{\alpha}}$	$c_{\beta,\hat{\beta}}$	$c_{\beta,\hat{\gamma}}$
DES	0.001	0.000	0.999	-0.702
PS1	0.002	0.000	0.999	-0.607
SDSS	0.002	-0.002	1.000	-0.134
SNLS	0.002	-0.001	0.999	-0.565
Low- z	0.003	0.001	0.999	1.258

Table 5.4: Same as Table 5.3, but for the standardization hyperparameter β , representing the color-luminosity correction.

Data set	γ			
	$c_{\gamma,0}$	$c_{\gamma,\hat{\alpha}}$	$c_{\gamma,\hat{\beta}}$	$c_{\gamma,\hat{\gamma}}$
DES	0.000	0.000	0.000	1.302
PS1	0.000	0.000	0.000	1.111
SDSS	0.000	0.000	0.000	1.237
SNLS	0.000	0.000	0.000	1.140
Low- z	0.000	0.000	0.000	2.072

Table 5.5: Same as Table 5.3, but for the standardization hyperparameter γ , representing the host mass-luminosity correction.

5.5 Conclusions

We have worked through a pedagogical example to show that performing linear regression one covariate at a time produces biased estimates of both the regression coefficients and spread of residuals when the covariates are correlated. The sizes of these biases depend directly on the magnitude of the correlation, and there are linear relationships between the error on the estimated slopes and the size of the factor that inflates the estimate of the spread of the remaining scatter. We have proven that similar relationships also hold when fitting step functions to the residuals of a linear regression (as is sometimes done in supernova cosmology) if there are correlations between the parameters being fit in each step.

We have also presented numerical simulations based on observed data to find corrections to the biases that are introduced from non-simultaneous regression methods. Each data set studied shows the possibility of a large underestimate of the size of the host mass step regardless of values of other nuisance parameters. There are also minor biases in the model parameters governing the relationship between luminosity and light curve width (SALT2 α) and luminosity and color (SALT2 β).

Biases are introduced when the assumptions underlying an analysis method are overlooked. In this particular case, there is an implicit assumption that all covariates must be uncorrelated in order to prevent biases from performing a two-step regression. A number of studies (e.g. Kelly et al., 2010; Sullivan et al., 2010; Childress et al., 2013; Jones et al., 2015, 2018; Rose et al., 2019; Kelsey et al., 2020) have neglected this effect, leading to underestimated sizes and significances of the effect sizes they report. For the most part, cosmology analyses, (e.g. Betoule et al., 2014; Scolnic et al., 2018; Smith et al., 2020), do properly account for this affect by fitting for the host mass step size simultaneously with the other standardization parameters. However, it is not yet clear if the host mass correlations are properly accounted for in the bias corrections, as discussed in Smith et al. (2020). Care must be taken in presenting the size and significance of these relationships, and propagating these correlations throughout the analysis. The biases presented here can be easily avoided by fitting all nuisance parameters simultaneously when presenting measurements of the mass step.

An interactive notebook with data simulations showing the derived relationships between effect sizes and correlation coefficients for our toy model, as well as the data and code used for the simulations with literature data are all publicly available at https://github.com/sam-dixon/sn_multicollinearity, and reproducible via Binder (Project Jupyter et al., 2018).

Chapter 6

High-Precision Distance Moduli from Single-Epoch Spectrophotometry Using Deep Learning

6.1 Introduction

Boone et al. (2020a) (hereafter B20a) aimed to elucidate the mathematical structure underlying the twinning method of Fakhouri et al. (2015) by applying manifold learning techniques to identify a low-dimensional representation of the twinnness space. Using projections into this lower dimensional space, Boone et al. (2020b) (hereafter B20b) was then able to model the necessary corrections to the supernova magnitudes based on these projections, standardizing supernovae with an RMS error of 0.101 ± 0.007 mag (0.084 ± 0.009 mag after accounting for contributions from peculiar velocities).

However, the B20b standardization analysis requires at least one spectrum in a narrow phase range around maximum brightness. The goal of the work we present here is two-fold: first to extend the use of the twins embedding by determining if the same standardization information can be extracted from a wider range of phases, and second to create a generative model that can be used in simulations or fitting lower-resolution spectroscopy or broadband photometry. The first goal is accomplished with a model that we refer to throughout the work as `spec2embed`, which takes a spectrum and predicts its phase along with the extinction parameter and twins embedding coordinate of the supernova that spectrum was taken from. To accomplish the second goal, we build a model that we call `embed2spec` to predict the spectral energy distribution (SED) of a supernova given its location in the twins embedding space and dust extinction parameter, up to a gray magnitude offset.

Both of these models are built using deep neural networks. This work is not the first application of deep learning to problems relevant to Type Ia supernovae or supernova cosmology. Sasdelli et al. (2016) used an unsupervised deep autoencoder network to model the intrinsic spectral diversity of SNe Ia. Muthukrishna et al. (2019) used supervised deep

learning to automate the classification of transient types, redshifts, ages, and host morphologies based on spectra. Both of these models are used only in spectroscopic classification – neither makes an attempt to improve the standardization of the supernovae under study. Stahl et al. (2020) is the first work to use deep learning to predict light curve properties that can be used in standardization from spectra, using deep convolutional neural networks to predict the phase and light curve shape parameter Δm_{15} . This network could predict Δm_{15} to within 0.056 magnitudes, which propagates to a standardized magnitude error (using the Phillips relation of Phillips (1993)) of approximately 0.151 mag. Our work is motivated by a desire to combine the success of deep learning methods in identifying valuable non-linear relationships between supernova spectra and low-dimensional representations of their behavior, with the success of the B20a and B20b analyses in standardizing supernovae using these low-dimensional representations.

We begin in Section 6.2 by summarizing the steps of the twins embedding analysis in more detail. In Section 6.3, we explain the data set used throughout our analysis. In Section 6.4, we present the neural network architectures used for both the models, as well as our techniques for finding optimal model hyperparameters. We examine how well our neural network is able to recover the twins embedding coordinates, how this recovery fidelity changes with phase, and how the embedding coordinate errors propagate to standardization errors in Section 6.5. In Section 6.6, we evaluate the performance of the `embed2spec` model. We conclude in Section 6.7 with some suggestions for further analyses using these tools.

6.2 The Twins Embedding Analysis

The following is a summary of the main methods and results of the twins embedding analyses presented in B20a and B20b that are needed to understand the work presented here. Full details can be found in the cited papers. The twins embedding analysis consists of four main parts: the differential time evolution model, the “read between the lines” analysis, the Isomap embedding analysis, and the Gaussian process model of the relationship between the color and embedding coordinates of a supernova and its deviation in brightness on the Hubble diagram. The first three subanalyses are presented in B20a, while the final portion is presented in B20b.

The differential time evolution model interpolates all observed spectra from -5 to $+5$ days after maximum brightness to the time of maximum brightness. The model takes advantage of the fact that while the spectra of each supernova can look quite different at maximum light, their differential evolution near maximum is remarkably similar. Additionally, because the differential evolution is being modeled, multiplicative effects on the spectrum (e.g. magnitude differences from distance uncertainties or color differences from dust extinction) have no effect on the model. The time evolution of the spectrum flux of each supernova is modeled as a quadratic function of the phase; this is an excellent approximation near maximum brightness, but breaks down beyond ± 5 rest-frame days, motivating an alternate modeling methodology for phases outside of this quadratic regime.

With the interpolated at-maximum spectra in hand, the “read between the lines” analysis (hereafter RBTL) is then used to describe the differences in brightness (Δm) and differential dust extinction parameter (ΔA_V) between each spectrum and a mean spectrum. The mean spectrum is the weighted average of all of the at-max spectra, with the weights being determined by the simultaneously-fit intrinsic variation at each wavelength. This weighting scheme means that the model in effect uses only the regions of the spectrum that are most intrinsically standard to determine the magnitude heterogeneity and extinction of the supernovae. Using just the RBTL magnitude offsets without any additional corrections, the SNe Ia are able to be standardized to within 0.131 mag. At this stage, though, some intrinsic variation in the spectra can be confused for extrinsic variation if this intrinsic variation looks like a difference in extinction or brightness. However, if we make the assumption that this variation also makes changes that can be measured from changes to spectral features, we can later correct for these differences. This motivates the final two subanalyses: the Isomap embedding and estimation of magnitudes using Gaussian process regression.

Isomap is a non-linear machine learning technique for finding approximately isometric low-dimensional embeddings of high-dimensional spaces. The B20a analysis uses this algorithm on spectra that have been corrected for the brightness and color differences identified by the RBTL analysis, and so learns a low-dimensional representation of the intrinsic variation of at-max spectra. This embedding (the twins embedding) is three-dimensional, with coordinates labeled by (ξ_1, ξ_2, ξ_3) . The twins embedding qualitatively reflects the diversity of SN Iaspectra, clustering objects with similar subclassifications together and pushing objects known to have peculiar subtypes (91T-like, 91bg-like, and 02cx-like) out to the periphery of the embedding space.

The twins embedding coordinates can also be used to correct for the intrinsic contributions to the magnitude residuals from RBTL. B20b presents a Gaussian process (GP) regression model to estimate the RBTL magnitude residual from the Isomap embedding coordinate. The GP allows us to capture non-linear relationships between the twins embedding coordinates and the magnitudes and also provides an estimate of the error. Mathematically, they model:

$$\vec{m} \sim \mathcal{GP} \left(m_{\text{ref}} + \omega \Delta A_V, \mathbb{I} \cdot (\vec{\sigma}_{\text{pv}}^2 + \sigma_u^2) + K_{3/2}(\vec{\xi}, \vec{\xi}; A, l) \right) \quad (6.1)$$

The $\omega \Delta A_V$ term and arbitrary m_{ref} in the mean function corrects for the possibility of introducing a correlation between extinction and magnitude from an incorrect fiducial R_V . The correlation between embedding coordinate and magnitude is captured by a 3-dimension Matérn 3/2 kernel with length scale l and amplitude A . The per-supernova uncorrelated errors from peculiar velocities enter into the model through σ_{pv} . Any unexplained error (including systematic error from the analysis) is captured by σ_u , which is constant for all objects in the sample.

Using leave-one-out cross-validation, the full analysis is found to be able to standardize the studied sampled of SNe Ia with an RMS of 0.101 ± 0.007 mag. After accounting for the contributions from peculiar velocities, this RMS is reduced to 0.084 ± 0.009 mag.

Our first model, `spec2embed`, seeks to mimic the first three subanalyses, taking a spectrum and determining its phase (in lieu of interpolating to maximum brightness), relative dust extinction parameter ΔA_V , and twins embedding coordinate (ξ_1, ξ_2, ξ_3) using a deep neural network. The final GP regression analysis step remains unchanged, but we examine the increase in residual spread by propagating the errors due to replacing the RBTL and Isomap process with a neural network. Our second model, `embed2spec`, can be thought of as an inversion of the Isomap embedding analysis, taking the twins embedding coordinate and predicting the spectrum at any phase p , corrected for Δm and ΔA_V . The differences in magnitude and dust extinction can then be added using their known behavior. The result is a model that predicts the spectrum at a wide range of phases that takes $(\Delta m, \Delta A_V, \xi_1, \xi_2, \xi_3)$ as input.

6.3 Data

Both the twins embedding analysis and our work were trained using spectrophotometry from of the Nearby Supernova Factory (SNfactory) data set. All of the spectra were obtained with the Super Nova Integral Field Spectrograph (SNIFS; Lantz et al. (2004)) mounted on the University of Hawai‘i 2.2-meter telescope on Maunakea. The spectra were reduced using the SNfactory data reduction pipeline (Ponder, et al. 2020, in preparation), calibrated via the process presented in Buton et al. (2013), and host galaxy subtracted as described in Bongard et al. (2011). Reddening by Milky Way dust in the line of sight is removed assuming a Cardelli (Cardelli et al., 1989) extinction law and using the dust maps from Schlegel et al. (1998).

After host galaxy subtraction and correction for Milky Way dust, we shifted the wavelengths of each spectrum in our analysis into the rest frame and rebinned the flux and flux variance onto a common wavelength grid from $3300 - 8600$ Å with 1000 km/s-wide bins using `sncosmo`. This binning is identical to that used by Fakhouri et al. (2015), Saunders et al. (2018), and Boone et al. (2020a). We also normalized the rebinned spectral flux and flux variance to a reference redshift of $z_{\text{ref}} = 0.05$ using

$$F_{\text{rest}} = F_{\text{obs}} \left(\frac{1+z}{1+z_{\text{ref}}} \right) \left(\frac{d_L(z_{\text{helio}}, z_{\text{CMB}})}{d_L(z_{\text{ref}}, z_{\text{ref}})} \right)^2 \times 10^{15} \quad (6.2)$$

where $d_L(z_{\text{helio}}, z_{\text{CMB}})$ is the luminosity distance at redshift z , assuming a flat, Λ CDM cosmology with $H_0 = 70$ km s $^{-1}$ Mpc $^{-1}$ and $\Omega_M = 0.3$ for the heliocentric and CMB-frame redshifts:

$$d_L(z_{\text{helio}}, z_{\text{CMB}}) = \frac{c}{H_0} (1 + z_{\text{helio}}) \int_0^{z_{\text{CMB}}} \frac{dz}{\sqrt{\Omega_M(1+z)^3 + (1-\Omega_M)}}. \quad (6.3)$$

This normalization serves to adjust the observed magnitudes of every object in our sample to match the magnitudes they would have if they were all at the same reference redshift. The factor of 10^{15} in Eqn. 6.2 serves only to make the flux and variance values $\mathcal{O}(1)$ (in units of

$\text{ergs cm}^{-2} \text{ s}^{-1} \text{ \AA}^{-1}$). Note that in the redshift range of our sample, the difference between the fiducial cosmological model used and a pure Hubble law (i.e. a linear relationship between redshift and distance) is ~ 0.01 magnitudes. Thus, the analysis is insensitive to changes in the exact values of the cosmological parameters.

In addition to the spectral observations, the SNfactory pipeline synthesizes photometry for each of the observed SNe and uses these synthesized magnitudes to find the best-fit SALT2 model parameters x_1 , c , and m_B^* for each object, as well as an estimate of the time of maximum brightness t_0 . The synthesized photometry SALT2 estimate of t_0 and the measured host galaxy redshift z is what determines the true phase value p of each spectrum used throughout this analysis:

$$p = (t - t_0)/(1 + z) \quad (6.4)$$

Our data selection cuts (explained below) ensure that each supernova used in the analysis has an uncertainty on the time of maximum brightness of less than 1 day.

The list of supernovae that are used to train and test both of our neural networks is identical to that used in the RBTL analysis of B20a. These objects each have at least 5 total observations, an uncertainty on the time of maximum brightness of less than 1 day, at least one spectrum within ± 5 days of maximum light, and at least one spectrum with sufficiently high signal-to-noise (SNR from $3300 - 3800 \text{ \AA} > 100$). This data set contains 203 supernovae. Only 173 of these objects were used in training the Isomap embedding, because the uncertainty of the at-max interpolation was significantly less than the intrinsic spectral variability. This cut was made to avoid the Isomap algorithm from confusing intrinsic and extrinsic variability of the spectra. However, once the embedding space is found, all 203 objects can be placed in the embedding space, and we use all of these objects in our neural network training and evaluation.

B20a only considers the spectra of these supernovae that are within 5 days of maximum brightness and interpolates these spectra to a single phase. We instead consider all of the spectra with SALT2 phases between -10 and $+40$ rest-frame days beyond maximum of the 203 supernovae analyzed by Boone et al. (2020a), resulting in a data set of 2623 spectra. The phase distribution histogram is shown in Figure 6.1.

These 2623 spectra serve as the inputs to the `spec2embed` model. The labels are the phases determined from SALT2 fits to synthesized photometry, along with the per-supernova twins embedding coordinates (ξ_1, ξ_2, ξ_3) , magnitude offsets Δm , and relative dust extinction values ΔA_V from B20a. For the `embed2spec` model, we use the normalized versions of these spectra (i.e. the spectra corrected for Δm and ΔA_V) as the labels for the network and the inputs are the corresponding embedding coordinates.

For the standardization analysis, B20b uses a smaller subset of 173 SNe Ia used to train the twins embedding in order to avoid errors from inaccurate host galaxy redshifts, peculiar velocities, and differences in dust properties. Thus, the standardization analyses in B20b and our work use only objects that have a high-quality host galaxy redshift, $z < 0.02$, and $\Delta A_V < 0.5$. This subset consists of 134 objects.

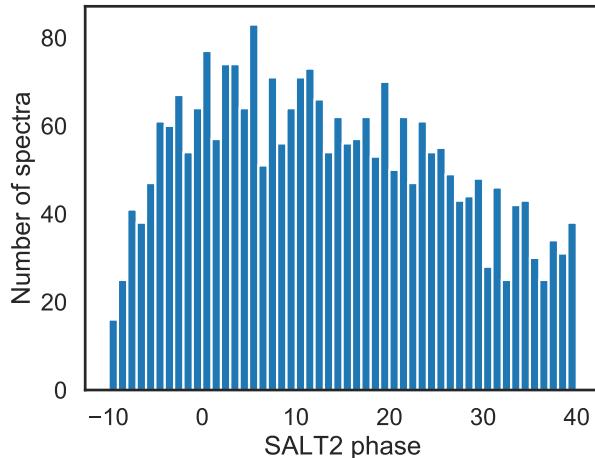


Figure 6.1: Histogram of SALT2 phases for the complete data set. The phases are estimated from SALT2 fits to light curves synthesized from the full spectral time-series data of each supernova.

Training, Validation, and Test Sets and Cross-Validation

To avoid overtraining and ensure the generalizability of our models, we separated our data into training, validation, and test subsets. The sizes of these subsets are 142, 20, and 41 (70%, 10%, and 20%) respectively. The supernovae were placed into each group randomly, and we ensured consistent placement in each group by fixing the pseudo-random generator seed in our analyses. We separated the data on a supernova-by-supernova basis rather than a spectrum-by-spectrum basis in order to avoid data leakage from e.g. spectra with multiple observations within a few nights.

During the hyperparameter tuning stage (Section 6.4), we trained each model using the training set and evaluated its performance using the validation set. To ensure robustness to changes in the validation set, we performed 5-fold cross-validation on the combined training and validation sets to select the best hyperparameter combinations from the initial sweep. k -fold cross-validation is a commonly used model evaluation technique used for smaller data sets, wherein we split our data set into k segments, train our model on $k-1$ of those segments and evaluate the trained model on the held-out segment. Using the average performance across these segments, we have a robust estimate of how well the predictions will generalize to unseen data sets. Once the hyperparameters were selected, we combined the training and validation sets (162 total SNe Ia) to train the final models. The test set SNe Ia were only used once all analysis steps were finalized.

This standard workflow of splitting data sets into training, validation, and test subsets is common in deep learning contexts because deep learning studies often operate in the regime where the size of the data set is extremely large and training an individual instance of the model is computationally expensive. However, in this work, we have a relatively small model

trained on a fairly small data set. Cross-validation, like k -fold cross-validation, can help us to bolster the strength of our claims about out-of-sample performance by using a larger effective data set.

The logical extension of k -fold cross-validation is to set k equal to the number of observations in the data set, a technique known as leave-one-out cross-validation (LOOCV). For every object in our data set, we train a separate model using data from every other object, then evaluate that model on the left-out object. This technique was used to evaluate the out-of-sample performance of the Gaussian process regression model used in B20b, and we perform a similar final analysis to complete the comparison for our standardization study. This final study used the full data set (training, validation, and test) to perform LOOCV for both the `spec2embed` embedding neural network training and the conditioning of the Gaussian process regression model.

6.4 Models

The models used in this analysis are multi-layer perceptrons (MLPs), simple artificial neural networks capable of capturing non-linear mappings from m -dimensional data to n -dimensional labels. Each network consists of several layers of nodes using either rectified linear unit (ReLU, Nair & Hinton (2010)) or tanh activation functions, with each layer fully connected to its neighboring layers.

The first model we explore, `spec2embed`, takes a single deredshifted and rebinned spectrum of a supernova (see Section 6.3) and predicts the phase of that spectrum, along with the relative extinction parameter ΔA_V and twins embedding coordinate (ξ_1, ξ_2, ξ_3) of the supernova that the spectrum came from. This model allows us to use the B20b standardization method on spectra at a wider range of phases. The second model, `embed2spec`, reverses this operation, taking an embedding coordinate and phase and producing a spectrum. This model can then be used for generating mock spectra for simulations, or for forward-model fitting to photometry. Both neural network models are implemented in PyTorch (Paszke et al., 2019), and the full generative model of `embed2spec` is implemented as an `sncosmo` Source object (Barbary et al., 2015b).

Model Architecture and Hyperparameter Optimization

Selecting the optimal architecture of an MLP (i.e. the number of layers and the number of nodes per layer) remains an open field of research. The universal approximation theorem states that an MLP with a sufficiently large number of nodes in a single hidden layer is able to approximate any function with any desired error. In practice, however, meeting the “sufficiently large hidden layer” criterion can be unfeasible, and though the network may be able to represent the function, the learning algorithms used to find the ideal weights may not be well-suited to ensure the learned weights produce this representation during training.

Empirically, deep neural nets (i.e. MLPs with more than one hidden layer) have been shown to be more tractable solutions that generalize well after training (Goodfellow et al., 2016).

The choice of activation function is also important. The ReLU activation function (linear for values greater than 0 and 0 otherwise) is empirically quite successful, due to its computational simplicity and properties that allow it to skirt the “vanishing gradient” problem (when the partial derivatives used to update the model weights through backpropagation become vanishingly small, halting training prematurely). ReLU also encourages the learned model weights to be sparse. However, this function is discontinuous, and can therefore cause the resulting outputs to be discontinuous as well. Discontinuity is acceptable in the `spec2embed` model, since we make no a priori assumptions about the smoothness of the twins embedding space. For the `embed2spec` model, on the other hand, we know that our outputs should be smooth because we know that the evolution of spectra is smoothly varying. For this reason, we use the tanh activation function in the `embed2spec` model and the ReLU activation function in `spec2embed`.

There are additional hyperparameters beyond the architectural design that we can include in the networks. Dropout layers (Srivastava et al., 2014) are commonly used regularization tools that randomly drop nodes from the network with some tunable probability. They can help to avoid overfitting and improve generalization, but the probability used to determine the number of connections to drop must be selected. The batch size (the number of training samples used in each training iteration) can also impact our results regardless of final architecture.

There are few strict rules governing the choice of these hyperparameters, so they are typically selected by experimentation. For our work, we used the Bayesian hyperparameter optimization framework implemented in the Weights and Biases package (`wandb`, Biewald, 2020) to search over a range of potential model architectures and other hyperparameters (see Table 6.1). For each combination of hyperparameters, we trained a model on our training data set, and evaluated the performance of the model by calculating the average mean squared error of the spectra in the validation set. These validation loss evaluations are then used to train a Gaussian process model in the hyperparameter space, and this model informs the choice of the next combination of hyperparameters to use in the sweep (see Snoek et al. (2012)). The initial sweep used 239 trials for the `spec2embedding` model and 252 trials for the `embedding2spec` model (approximately 6 CPU-hours for each model, run serially). Each model was trained with 150 epochs – enough to ensure convergence.

As explained in Section 6.3, we perform 5-fold cross-validation on the top 10 best-performing models from this initial sweep to obtain a final model that is robust to variations in which supernovae end up in the validation set. The final model that we selected was the model with the lowest average cross-validation loss. The preferred hyperparameters from this second round of validation are also listed in Table 6.1.

We can see that a deeper architecture is preferred for both the `spec2embed` and `embed2spec` models. This is consistent with the qualitative assessment that sequential network layers learn a hierarchical series of features. Both models also prefer smaller values for the dropout probability.

Hyperparameter	Value options	spec2embed preferred	embed2spec preferred
Network depth	[1, 2, 3, 4]	3	4
Nodes per layer	[72, 100, 128, 144, 172, 200, 228, 256, 288]	256	144
Dropout probability	[0, 0.001, 0.005, 0.01, 0.05, 0.1]	0.001	0
Batch size	[2, 4, 8, 16, 32]	16	32

Table 6.1: MLP hyperparameters values used in Bayesian optimization along with the preferred values for each model.

Using the hyperparameters in Table 6.1, we performed our final training of the `spec2embed` and `embed2spec` networks using the full combined training and validation sets. These are the final model weights used to evaluate the model accuracy on the held-out test set in Sections 6.5 and 6.6.

Magnitude Standardization Model

The methodology for standardization given a twins embedding coordinate remains unchanged from the steps laid out in B20b; we use the same Gaussian process (GP) regression model hyperparameters and minimally modify the training code. To condition the GP, we use the embedding coordinates and extinction parameters used in B20b as input and the magnitude offsets found in the same work as output. We do not at this stage use the coordinates predicted by the `spec2embed` model. As mentioned in Section 6.3, we made the same selection cuts for standardization that are made in the B20b analysis, removing objects that are very nearby ($z < 0.02$), highly extincted ($A_V > 0.5$), with large uncertainties in redshift, or with poor data quality. This smaller subsample consists of 134 supernovae, 111 of which are in our combined training and validation set, and the remaining 23 are in our held-out test set.

Repeating the leave-one-out cross-validation analysis on the full data set yielded the same results as those presented in B20b. To gather a baseline for a later standardization comparison, we used the same training process to condition the GP on the standardization training subset (111 SNe Ia) and evaluated the model on the standardization test set (23 objects). We quantify our performance by calculating the residuals between the predicted values of Δm and the values of Δm found in B20a, then measuring the RMS and normalized median absolute deviation (NMAD) of these residuals. The results from both the leave-one-out reproduction and the train-test split are summarized in Table 6.2. Note that all of the embedding coordinate values used here are identical to those in B20b; our intention is to examine how well the model performs on our smaller test set before we add in the additional uncertainty that comes from our neural network model. As expected, the results are all consistent with the leave-one-out cross-validation results, though the RMS values are slightly larger due to the smaller sample size.

Our analysis also differs from the B20b analysis in that we are interested in building a model that can predict the supernova properties (like extinction and embedding coordinate) from multiple spectra. Thus, we also present the same measures of the spread in the per-supernova metrics weighted by the number of spectra that appear for those supernovae in our data set. The resulting values are all consistent with one another, but the error bars for the weighted values are smaller, reflecting the increase in the number of observations.

Generating Spectra with `embed2spec`

When training the `embed2spec` model, we used the deredshifted, rebinned, dereddened, and magnitude-normalized spectra as our model labels (see Section 6.3). By doing so, the network only needs to learn the variation in the spectra due to changes in its phase and its parent supernova’s location in the embedding space. The effects of differences in extinction and overall brightness can easily be added afterward. Mathematically, the deredshifted spectrum $f(\lambda)$ of a supernova with embedding coordinate (ξ_1, ξ_2, ξ_3) , extinction A_V , and magnitude offset Δm observed as phase p is modeled as

$$f(\lambda; p, \xi_1, \xi_2, \xi_3, A_V, \Delta m) = \text{embed2spec}(p, \xi_1, \xi_2, \xi_3) \times 10^{0.4[\text{F99}(\lambda; A_V, R_V=2.8) + \Delta m]} \quad (6.5)$$

where $\text{F99}(\lambda; A_V, R_V = 2.8)$ is the extinction-color relation of Fitzpatrick (1999) using a total-to-selective extinction ratio of $R_V = 2.8$. Using Equation 6.5, we can then easily synthesize lower-resolution or broadband photometric observations, enabling forward modeling fits to these alternate data types.

6.5 `spec2embed` Results

Recovering Phase, Extinction, and Twins Embedding Coordinates

Using the final trained `spec2embed` model, we now evaluate its performance on the test set that has thus far been completely unseen by the model. Figure 6.2 shows the predicted phases and extinction parameters as a function of the true phases and extinction parameters, and Figure 6.3 shows the same for each of the twins embedding coordinates (ξ_1, ξ_2, ξ_3) . Using the NMAD of the residuals to measure our accuracy, we find that the model is able to constrain the phase of the spectrum to within ± 2.4 days, the extinction to within ± 0.1 mag, and the embedding coordinates to within 0.69, 0.52, and 0.59 for ξ_1 , ξ_2 , and ξ_3 respectively.

The phase recovery is slightly more accurate than the commonly used SNID program (Blondin & Tonry, 2007), which can constrain the phase to within ± 2.9 days. It is not quite as accurate as other neural network based techniques like Stahl et al. (2020), which constrains the phase to with $\sim \pm 1$ day. Even in the restricted phase baseline of -4 to +12 days beyond maximum brightness that is used in Stahl et al. (2020), we have an NMAD of the residuals of 2.4 days. However, as we will see in Section 6.5, the phase determination is not as important for standardization as the accuracy in determining the extinction and

Metric	Full sample	LOO	Training/val	Test
N_{SNe}	134		111	23
N_{spec}	1683		1395	288
RMS residuals (mag)	0.102 ± 0.008	0.082 ± 0.007	0.120 ± 0.023	
RMS residuals, pec. vel. removed (mag)	0.086 ± 0.009	0.061 ± 0.009	0.099 ± 0.027	
NMAD residuals (mag)	0.085 ± 0.011	0.068 ± 0.009	0.071 ± 0.025	
Weighted RMS residuals (mag)	0.105 ± 0.002	0.083 ± 0.002	0.127 ± 0.006	
Weighted RMS residuals, pec. vel. removed (mag)	0.089 ± 0.003	0.064 ± 0.003	0.110 ± 0.007	
Weighted NMAD residuals (mag)	0.085 ± 0.003	0.065 ± 0.002	0.075 ± 0.007	

Table 6.2: Summary of standardization results using the B20b embedding coordinates and extinction parameters using our training and test sets. Weighted measures of the spread are weighted by the number of spectral observations for an individual supernova. All values are consistent with the values found in B20b.

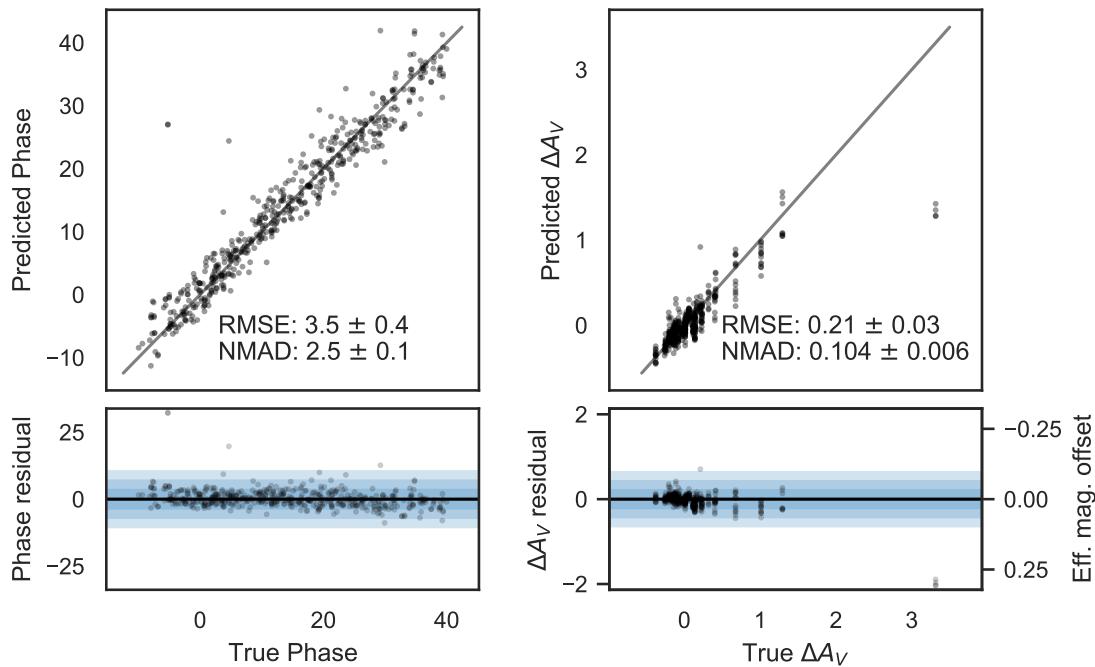


Figure 6.2: Comparison of phase and extinction parameters determined by the `spec2embed` model and ground truth values presented in B20a. The solid lines in each of the upper panels represents equivalence. We also note the root-mean-squared-error (RMSE) and the normalized median absolute deviation (NMAD) of the residuals (predicted value – true value) for each property. The lower panels show these residuals as a function of the true values. Shaded regions in the lower panels indicate the 1-, 2-, and 3- σ fluctuations of these residuals about the median. The secondary axis in the bottom right panel indicates the linear scaling between the error on the extinction parameter and the final standardized magnitude.

location in the twins embedding space of each object, and ultimately this algorithm can better estimate distance moduli.

The relationship between the extinction parameter and the standardized magnitude is linear (see Equation 6.1). The slope of this linear correction, ω , therefore tells us how the errors in the extinction parameter propagate to errors on the standardization. This linear transformation is included in the secondary axis of Figure 6.2 as the “effective magnitude offset.” The value of ω we find in Section 6.5 is -0.158, telling us that the accuracy in determining the extinction parameter is sufficient for standardization, as the 0.1 error in ΔA_V propagates to a 0.015 mag error in the final standardized brightness. There is one large outlier in color in our test set: SN2006X. This object is highly extincted ($\Delta A_V = 3.31$), more so than the most highly extincted object in the training set (SN2012cu, $\Delta A_V = 2.8$). Both of these highly extincted objects are excluded from the standardization analysis to avoid biases from uncertainties in interstellar dust properties, so the errors here do not propagate

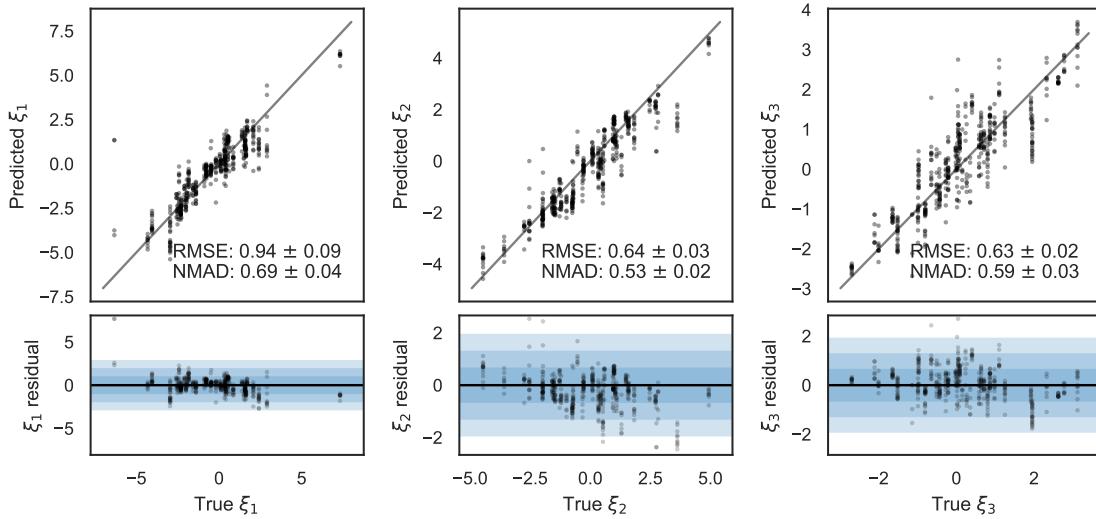


Figure 6.3: Same as Figure 6.2, but comparing the predicted and ground-truth twins embedding coordinates ξ_1 , ξ_2 , and ξ_3 .

to the standardization analysis.

Standardization Results

We now use the Gaussian process regression model conditioned on the true embedding coordinates and extinction parameters of the training and validation sets to make predictions of the magnitudes based on the embedding coordinates that were predicted by the trained `spec2embed` network. The results are summarized in Table 6.3. The first results column is identical to the weighted results of Table 6.2, since the results are weighted on a spectrum-by-spectrum basis rather than a supernova-by-supernova basis.

Because the embedding predictions for the training set are so similar to the ground truth predictions, the resulting predictions for the standardized magnitudes are also very similar (differing on average by less than 0.01 mag). The test set gives us a better sense of how well this method of standardization will perform on outside data, as all of the observations of these objects were unseen in both the neural network training and the standardization model conditioning. Here, we see a slight inflation in the errors; on average the error in the embedding coordinate prediction propagates to an 0.024 magnitude error in the standardized brightness. This is however subdominant to the standardization model error, indicating that these neural-network-predicted embedding coordinates at all phases can be used about as successfully as the RBTL and Isomap fits from maximum brightness spectra.

In Figure 6.4, we show the standard deviation and NMAD of the differences between the standardized magnitudes predicted from the `spec2embed`-predicted embedding coordinates and the magnitudes predicted from the true embedding coordinates (i.e. $\text{GP}(\text{pred}) -$

Metric		Training	Test
		Single epoch	Single epoch
N		1395	288
$\text{GP}(\text{true}) - \Delta m_{\text{true}}$	RMS	0.083 ± 0.002	0.127 ± 0.006
	Pec. vel. removed	0.064 ± 0.003	0.110 ± 0.007
	NMAD	0.064 ± 0.003	0.083 ± 0.010
$\text{GP}(\text{pred}) - \Delta m_{\text{true}}$	RMS	0.085 ± 0.002	0.130 ± 0.006
	Pec. vel. removed	0.066 ± 0.003	0.114 ± 0.007
	NMAD	0.068 ± 0.003	0.087 ± 0.008
$\text{GP}(\text{pred}) - \text{GP}(\text{true})$	RMS	0.009 ± 0.0003	0.024 ± 0.001
	NMAD	0.008 ± 0.0002	0.021 ± 0.002

Table 6.3: Standardization errors for the training and test data sets using the Gaussian process regression model conditioned on the B20a embedding coordinates to predict the absolute magnitudes based on the B20a embedding coordinates ($\text{GP}(\text{true})$) and the `spec2embed` neural-network predicted embedding coordinates ($\text{GP}(\text{pred})$), as well as the average difference between these predictions. Errors on the statistics are determined via bootstrap resampling.

$\text{GP}(\text{true})$ in Table 6.3) as a function of the spectrum phase for the test set. The spread of the errors is roughly consistent across phases, providing further credence to the claim that the neural network gives us the ability to standardize across a wide range of phases. The large error in the first phase bin (-10 to -7.5 days) is driven mainly by the paucity of test data in this region.

Leave-One-Out Analysis

Our claims about how well this technique is able to standardize SNe Ia are perhaps limited by the small size of our test set. In order to make our claims more precise, as well as to have a fair comparison with the standardization results of B20b, we perform a full leave-one-out cross-validation analysis of both the neural network training and the GP regression conditioning. We used the complete data set of 203 supernovae to train 203 instances of the `spec2embed` model, each using the data from all but one supernova. We performed a similar analysis, training 134 versions of the GP regression model using the embedding coordinates from B20b. When evaluating the standardization by combining the two models, we ensure that the same supernova was held out in both models.

In Figures 6.5 and 6.6, we show the comparison of the true phases, extinctions, and embedding coordinates from this leave-one-out test for the entire data set. The spreads of the residuals are all roughly similar to their values for the smaller test set, and we present an explicit comparison in Table 6.4.

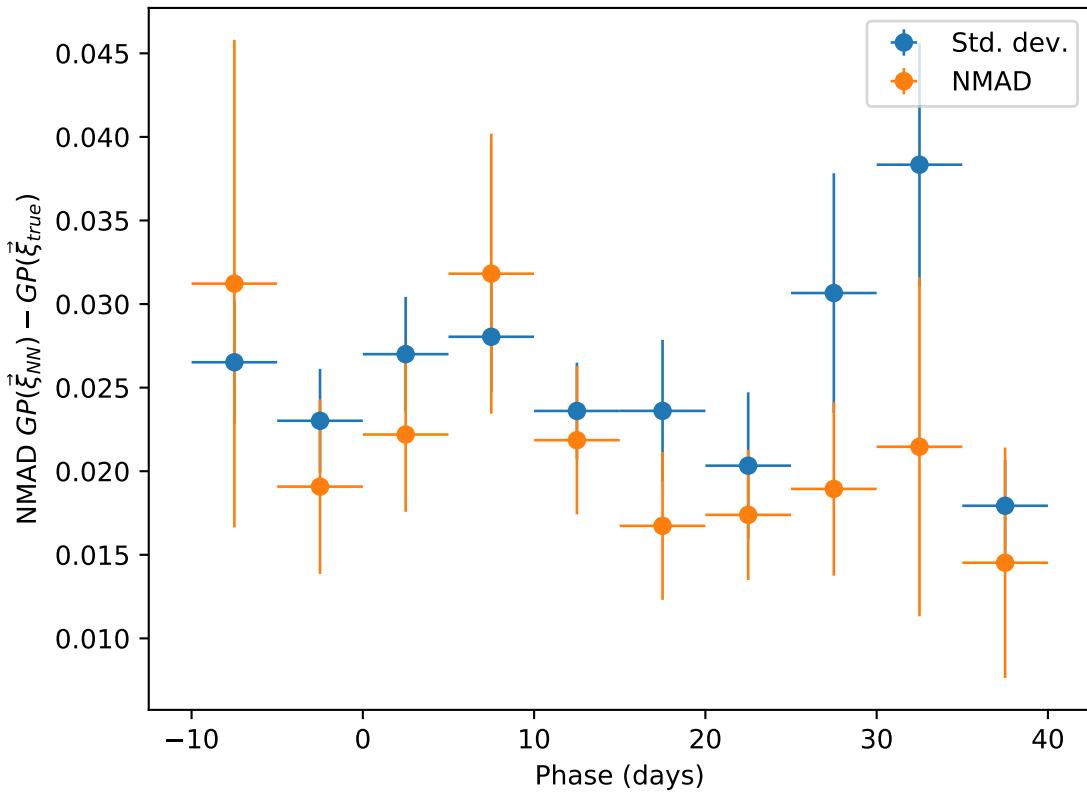


Figure 6.4: Standard deviation and NMAD of the difference between the magnitudes predicted from the neural network predicted embeddings and those predicted from the true embeddings as a function of the spectral phase.

In Table 6.5, we summarize the size of the standardization errors as determined by this full leave-one-out analysis. We find that with our neural network model, we can standardize supernova magnitudes with a precision comparable to the near-maximum twins embedding analysis, even when using spectra far from maximum brightness. We can see this more clearly in Figure 6.7, where we show the standard deviation NMAD of the LOOCV (leave-one-out cross validation) magnitude residuals as a function of phase. Again, the large error bar in the -10 day to -7.5 day bin is due to how few spectra there are available in that window. We can also see in this figure that our single spectrum standardization method surpasses the standardization power of the commonly used SALT2 model, which requires the observation of a full broadband light curve.

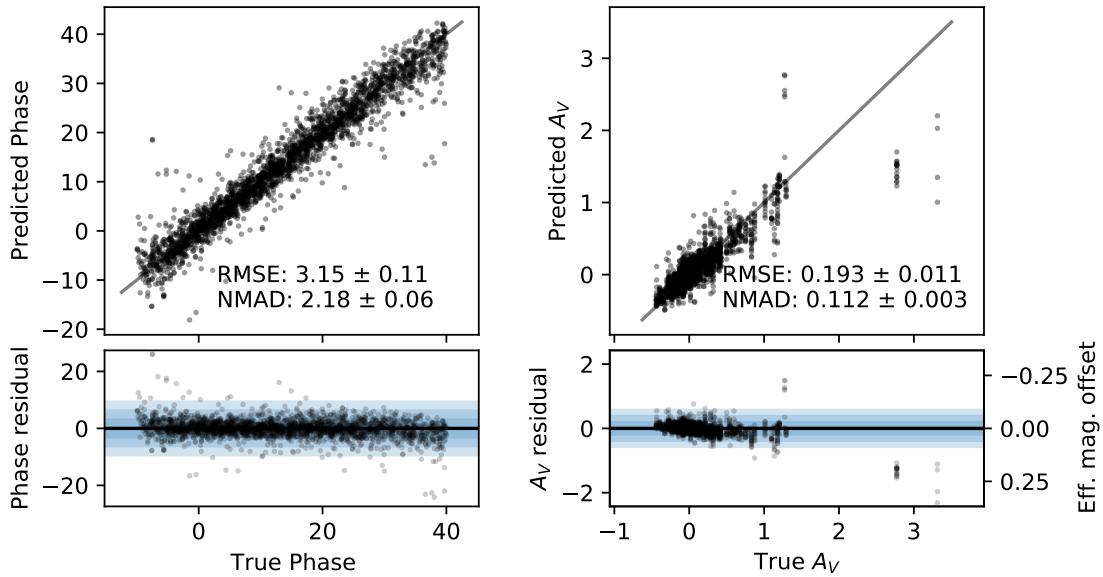


Figure 6.5: Same as Figure 6.2, but using the leave-one-out cross-validation analysis. As in Figure 6.2, each point represents a single spectrum. However, the predictions are made using networks trained on every supernova except the supernova the spectrum comes from.

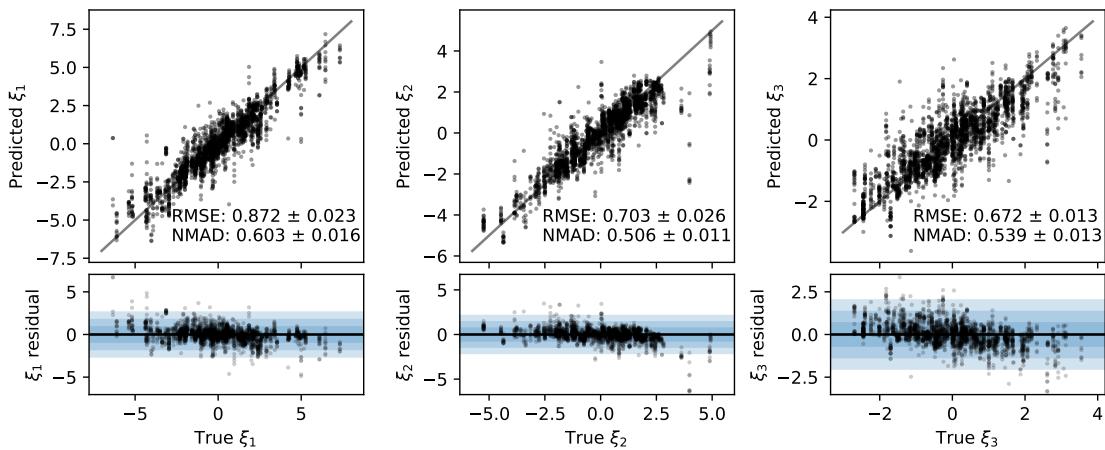


Figure 6.6: Same as Figure 6.5, but for the embedding coordinates.

Parameter	Metric	Test set	LOOCV
Phase	RMS	3.5 ± 0.4	3.0 ± 0.10
	NMAD	2.48 ± 0.12	2.18 ± 0.06
ΔA_V	RMS	0.21 ± 0.04	0.192 ± 0.010
	NMAD	0.106 ± 0.007	0.112 ± 0.003
ξ_1	RMS	0.95 ± 0.09	0.87 ± 0.02
	NMAD	0.70 ± 0.04	0.60 ± 0.02
ξ_2	RMS	0.65 ± 0.03	0.70 ± 0.03
	NMAD	0.52 ± 0.02	0.506 ± 0.011
ξ_3	RMS	0.64 ± 0.02	0.672 ± 0.013
	NMAD	0.60 ± 0.03	0.539 ± 0.013

Table 6.4: Comparison of parameter accuracy between the test set evaluation in Section 6.5 and the LOOCV evaluations.

Metric	LOOCV value	
$GP(\text{true}) - \Delta m_{\text{true}}$	RMS	0.105 ± 0.002
	Pec. vel. removed	0.089 ± 0.003
	NMAD	0.085 ± 0.003
$GP(\text{pred}) - \Delta m_{\text{true}}$	RMS	0.107 ± 0.002
	Pec. vel. removed	0.092 ± 0.003
	NMAD	0.092 ± 0.003
$GP(\text{pred}) - GP(\text{true})$	RMS	0.0327 ± 0.0012
	NMAD	0.0220 ± 0.0008

Table 6.5: Standardization results using a full leave-one-out cross-validation analysis. Each measure of the spread was obtained by training a separate `spec2embed` model and GP regression model for each supernova, where both models are conditioned on the data from all SNe except that held-out SN. Uncertainties on the values of these metrics were determined via bootstrap resampling.

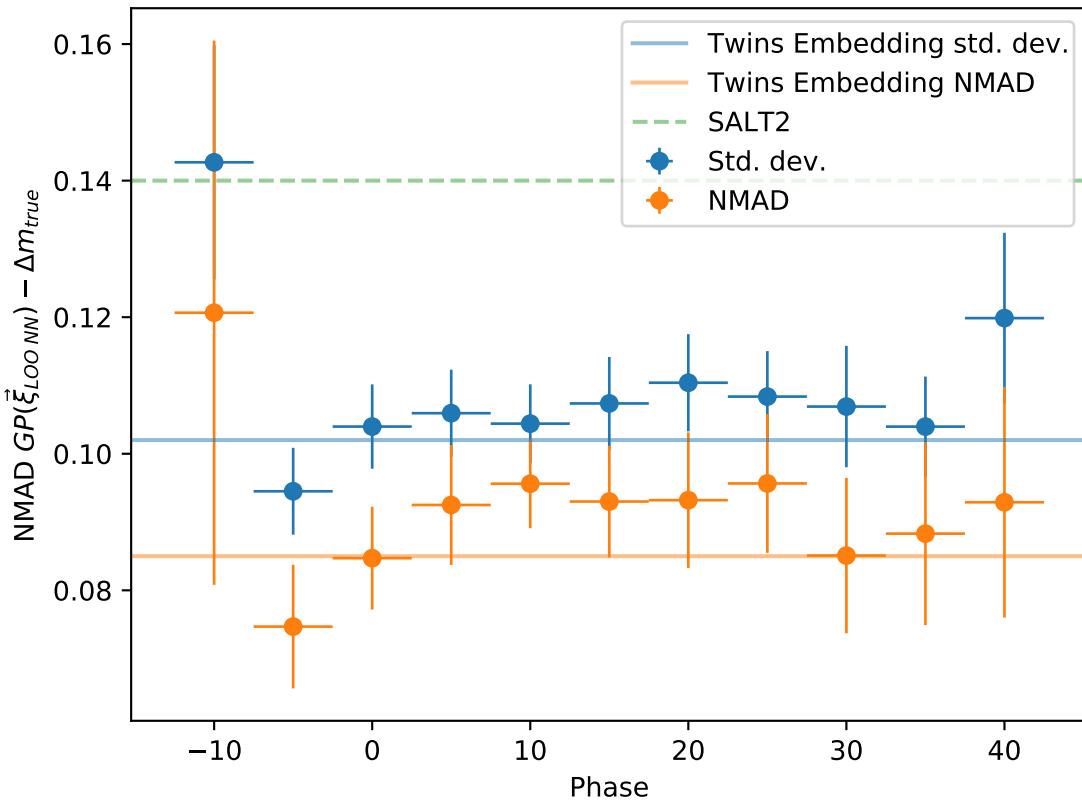


Figure 6.7: Standard deviation and NMAD of the standardization residuals ($\text{GP}(\text{pred}) - \Delta m_{\text{true}}$) as a function of phase for the full leave-one-out cross-validation analysis. We also show the value of this metric for the twins embedding analysis, as well as for the SALT2 model.

6.6 Recovering Spectra with `embed2spec`

The `embed2spec` model allows us to extend the twins embedding into a generative model that can then be used to fit other types of observations, like photometric light curves. We have implemented a version of the trained model as a `Source` object in `sncosmo`, a Python package for supernova cosmology. This allows us to easily add the effects of redshifting and dust extinction, as well as providing methods for synthesizing photometry.

To get a qualitative sense of the performance of this model, we show some example reconstructions. In Figure 6.8, we compare the observed spectral time-series data for PTF09dnl, a supernova in our held-out test set, to the spectra time-series predicted by `embed2spec`. Figure 6.9 shows a similar comparison using synthesized photometry in top-hat band passes corresponding roughly to Bessell BVRI bands (Table 6.6). The observed and predicted data align fairly well, though at earlier phases, the agreement is less strong, particularly near the CaII H&K feature from 3500 to 3990 Å.

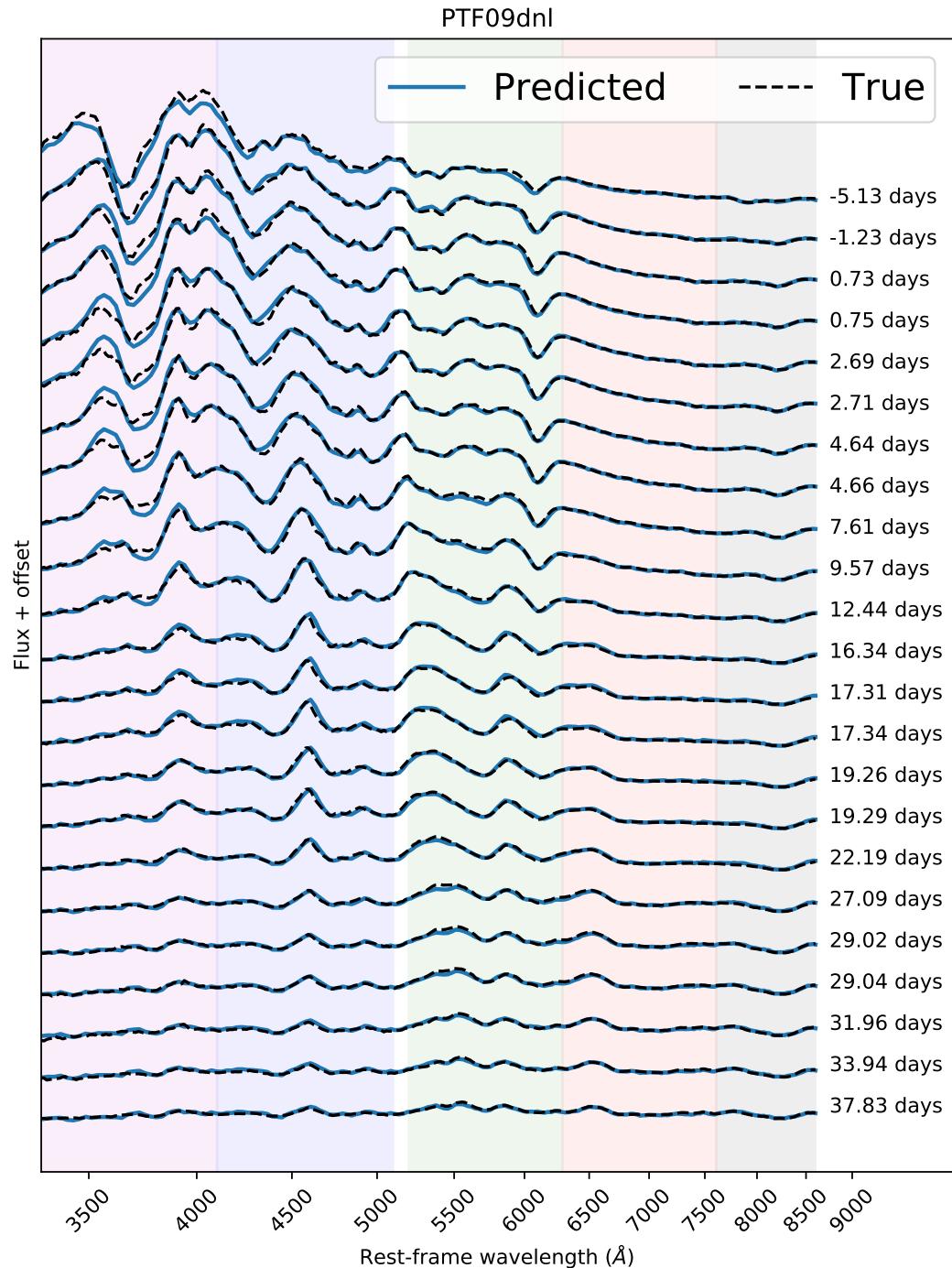


Figure 6.8: Comparison of the observed spectral time-series data (in black dashed lines) to the predicted data (blue solid lines) for PTF09dnl, a supernova in our held-out test set.

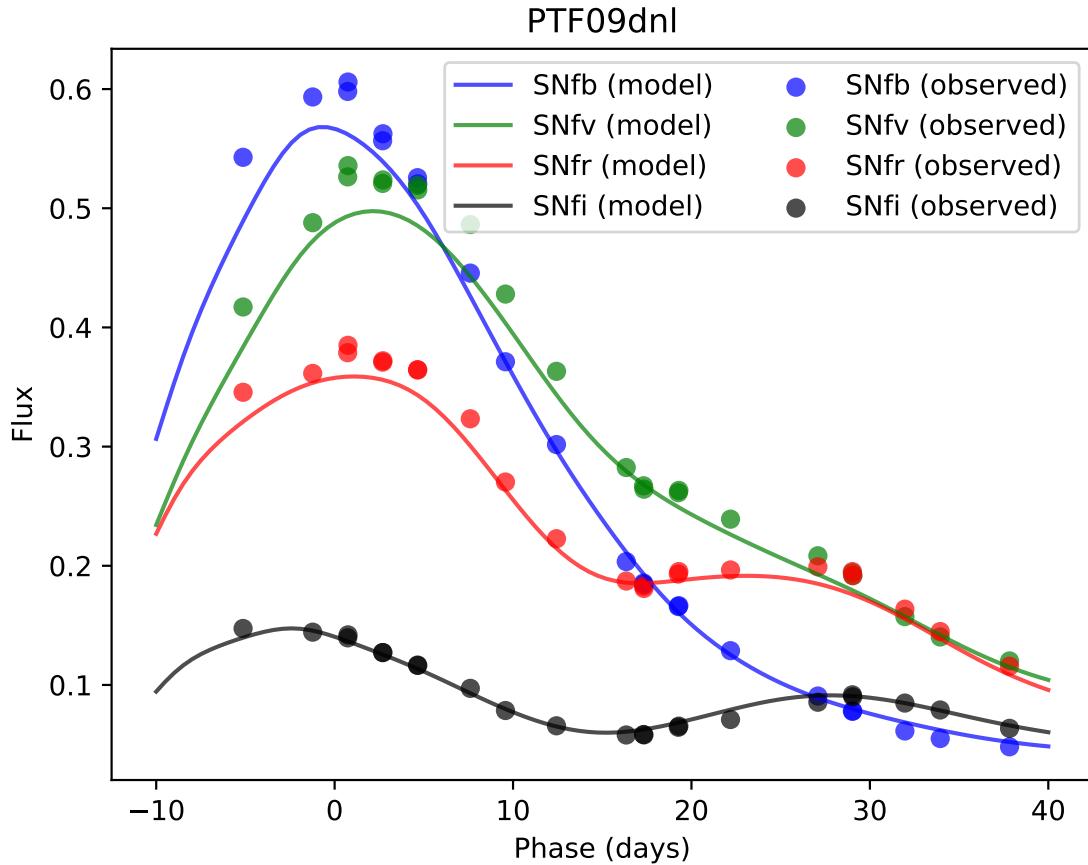


Figure 6.9: Similar to Figure 6.8, but using synthesized photometry to compare the observer-frame light curves (dots) to the model predictions (solid lines). The band passes used are defined in Table 6.6.

Name	Wavelength range (\AA)
SNfu	(3300, 4102)
SNfb	(4102, 5100)
SNfv	(5200, 6289)
SNfr	(6289, 7607)
SNfi	(7607, 8600)

Table 6.6: Top-hat band pass definitions.

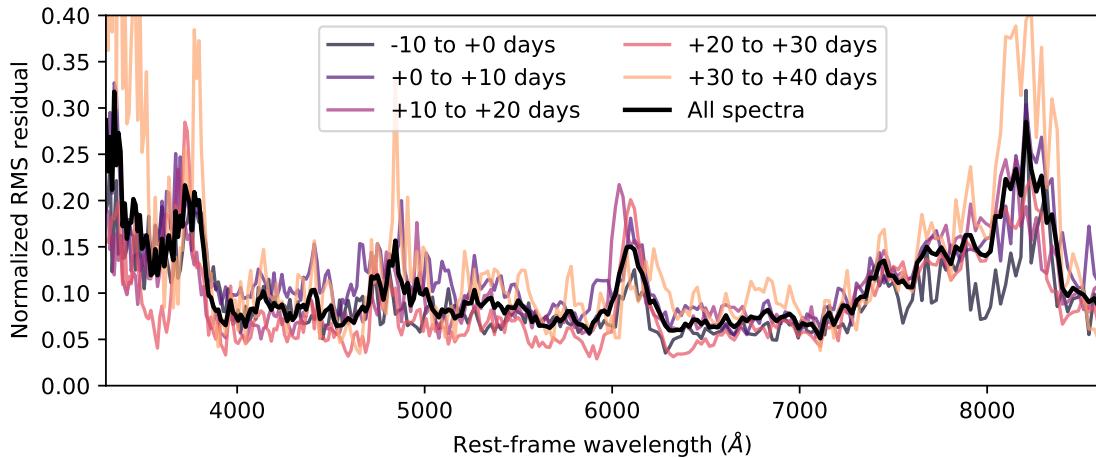


Figure 6.10: RMS of `embed2spec` flux residuals in each wavelength bin, normalized by the average flux in the bin across all phases. The thick black line shows the overall normalized RMS flux residual, and the thinner colored lines show the normalized RMS flux residual for subgroups selected by phase.

We also examined the performance of this model over the full test set more quantitatively. For each spectrum of the supernovae in the test set, we calculated the flux residuals: the difference between the observed and predicted values. In Figure 6.10 we show the RMS of these residuals as a function of wavelength bin, normalized by the average flux in each wavelength bin marginalized over all observed phases. Between 4000 and 8000 Å, the normalized RMS remains below 0.2 across all phases, indicating that the model is able to capture the flux to within 20%. There are a few regions of the spectrum that have larger relative dispersions in the flux residuals – these each correspond to well-known spectral features like the Ca II H&K doublet, the Si II $\lambda 6355$ Å line, the O triplet, and the Ca II IR triplet. These features are also identified in B20a as regions of higher variability near maximum brightness, which may contribute to this wider spread. Away from these regions, the spread of the residuals is at the 5-10% level.

In addition to quantifying the model accuracy as a function of wavelength, we measure the performance as a function of phase. In Figure 6.11, we plot the RMS of the flux residuals, normalized by the average flux for each spectrum as a function of the spectrum phase (after marginalizing over all wavelengths). As we may have anticipated, the errors are lowest near-maximum brightness, as there is less interpolation error and greater spectral coverage. Most of the larger outlying points correspond to spectra of SN2009hs and LSQ12fhs, two peculiar supernovae with subtypes that have little representation in the training set (SN1991bg-like and SN2002cx-like, respectively). Excluding these two objects, our model is able to capture the spectrum flux to within approximately 20% at all phases between -10 and $+30$ days after maximum brightness.

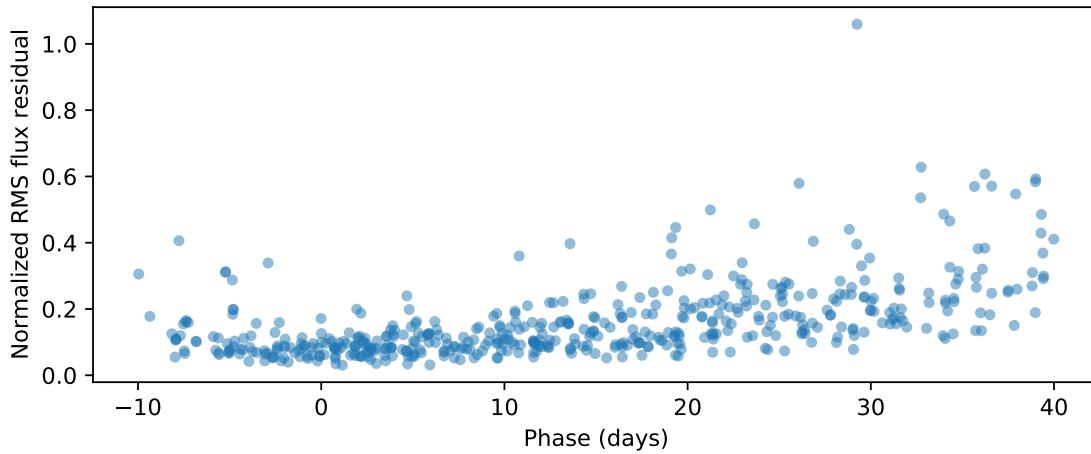


Figure 6.11: RMS of the flux residuals normalized by the average flux over all wavelengths at each phase. Most of the outlying points belong to either SN2009hs, a 91bg-like peculiar supernova, or LSQ12fhs, an 02cx-like peculiar supernova.

Finally, we examine the fidelity of the `embed2spec` network results as a function of location in the twins embedding space. Figure 6.12 shows the location of each test set supernova in the twins embedding space colored by its average normalized RMS flux residual. We can see that the supernovae that the model performs most poorly on are the objects on the periphery of the embedding space. The two largest outliers, SN2009hs and LSQ12fhs, are also made apparent in this plot. These peculiar subclasses of objects are not very well-represented in the training set (two 91bg-like objects and a single 02cx-like object), so this is unsurprising. Retraining this network with more observations of peculiar objects could further refine this work. Also, additional studies are necessary to assess how well this new model can be constrained by noisy, lower-resolution spectroscopy or photometry, and how these errors ultimately propagate to the standardization error. However, this model can already serve as a starting model to simulate spectra for survey design studies.

6.7 Conclusions

We have presented an extension of the twins embedding analyses of B20a and B20b allowing the use of a wider range of phases to standardize Type Ia supernovae for cosmology. Our `spec2embed` model can predict the phase, relative extinction, and location in the twins embedding space of a SN Ia using any photometrically calibrated spectrum from -10 to $+40$ days after maximum brightness with high fidelity. Using these neural network predicted extinction and embedding coordinates, we can use the standardization model from B20b to predict the absolute brightness of a supernova from any of its spectra with an RMS error of 0.107 ± 0.002 mag. If we remove the effect of peculiar velocities to gain some insight

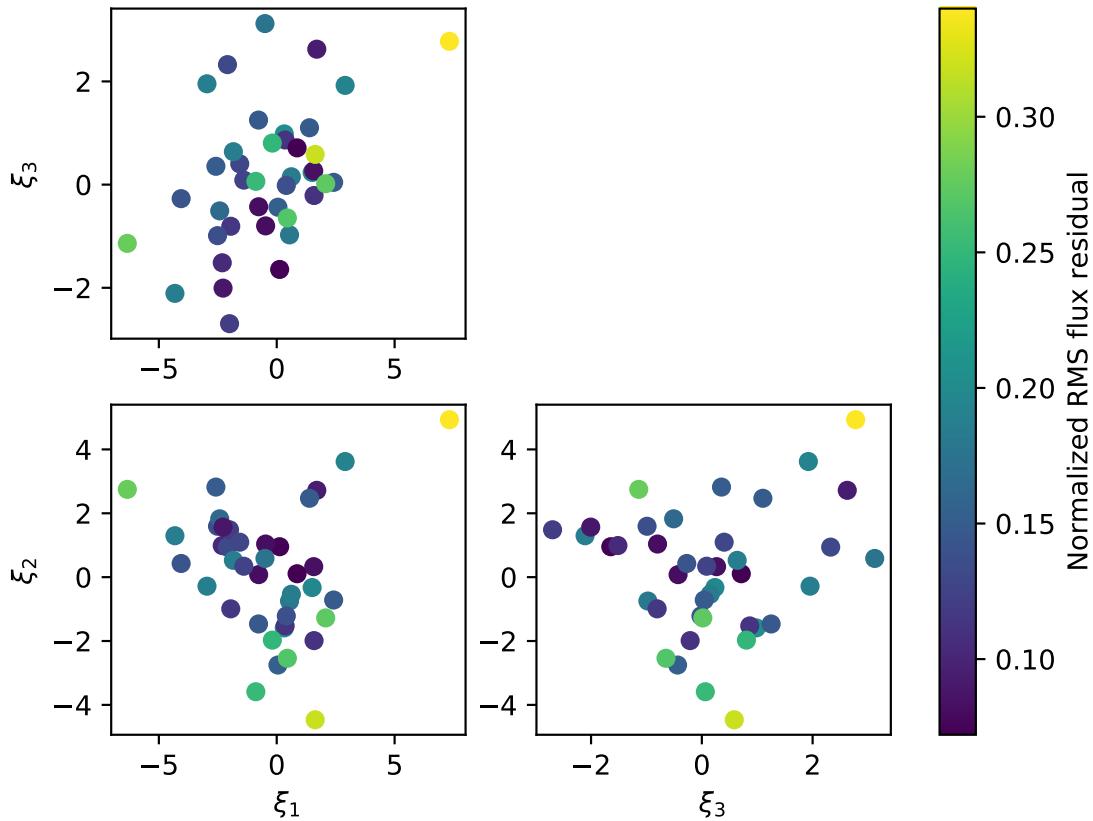


Figure 6.12: Phase-averaged RMS flux residual for all observed spectra for each test set supernova as a function of twins embedding space coordinate.

into how well this model would perform in a higher redshift study, this RMS error shrinks to 0.092 ± 0.003 mag. This standardization error is comparable to what was found using near-maximum spectra in B20b, and is far superior to the error we can obtain using light curve based techniques like SALT2.

Additionally, we have trained and evaluated a separate model, `embed2spec`, which capable of generating spectral time-series data based on an embedding coordinate. This model can be used in simulations and in forward modeling fitting approaches, potentially allowing us to obtain the twins embedding coordinate of a supernova from more readily available photometry or lower-resolution spectroscopy.

There are a number of studies that may be done to extend this work even further. We have not yet studied how robust this model is to noise or made an attempt to quantify the accuracy of the forward-modeling approach of `embed2spec` as a function of spectral signal-to-noise ratio or resolution. The `spec2embed` model requires a photometrically calibrated spectrum with sufficient resolution and wavelength coverage to create the binned, deredshifted spectra similar to those used in training. Many spectrographs do not have the level of calibration

precision available with SNIFS. Using data augmentation techniques may prove useful in improving the results of the model (Boone, 2019) and inuring the model from biases or errors stemming from calibration errors, lower signal-to-noise, or differences in wavelength coverage. Retraining the network with spectra preprocessed similarly to the spectra used in e.g. Stahl et al. (2020) may also allow us to use these same techniques while relaxing the spectrophotometricity constraint.

However, even without these additional studies, we have already produced a method for standardizing Type Ia supernova magnitudes with only a single spectrum observed anywhere within a wide range of phases. Moreover, the precision of this standardization is dramatically better than light curve-based techniques that require several observations. Spectra with the calibration necessary to use this technique are already obtainable. At lower redshifts, this technique may prove powerful in studies of the growth of structure via peculiar velocity measurements, or in measurements of the Hubble constant. At higher redshifts, this technique can provide a strong probe of systematic errors due to population drifts with redshift.

All software and data used in this analysis has been made publicly available on Github: https://github.com/sam-dixon/nntwins_embedding.

Chapter 7

Conclusions

We have presented a number of tools for finding and measuring potential systemic errors and biases throughout the process of making cosmological measurements using Type Ia supernovae, from detection through standardization.

In Chapter 2, we presented measurements of the charge transfer inefficiency of CCD detectors used to collect much of the data used throughout the rest of this dissertation. The laboratory measurements provide a baseline for future analyses, and the in-situ measurements allow for tracking the charge transfer efficiency over time. We found that the charge transfer inefficiency was stable over time, and small enough to have little impact on further measurements.

Chapter 3 presented a model of the Si II $\lambda 5972$ and Si II $\lambda 6355$ region of Type Ia supernova spectra. This model was able to accurately extract measurements of this feature from low-resolution and noisy spectra, enabling the use of spectra from prism spectroscopy supernova surveys like that planned for the Roman Space Telescope. Since this spectral region can act as a probe of potential population drifts, this model provides a tool for monitoring these drifts out to higher redshifts.

We investigated a similar problem in Chapter 4, measuring how well a number of the inherently non-linear near-maximum brightness spectral features could be measured using SALT2 and SNEMO, two linear empirical models of Type Ia supernova spectral evolution. We found that, in general, linear spectral models with more parameters can more accurately capture these non-linear features. In this Chapter we also constructed a model for producing realistic fake spectra using kernel density estimation of the latent parameter space of these models. This more flexible latent space model was better able to mimic the distributions of the spectral features than simpler (e.g. multivariate Gaussian) models of these latent spaces.

We addressed a generalized statistical problem in Chapter 5, as it appears in a number of Type Ia supernova. A number of analyses, particularly those studying host galaxy properties, perform a sequential regression which biases results when the regression covariates are correlated. Using a toy model of the problem, we calculated closed-form solutions for the size of these biases. We also used simulations based on publicly available Type Ia supernova data sets to measure the size of this bias in this context and provided correction terms.

Finally, in Chapter 6, we presented two new deep learning models of Type Ia supernova spectroscopy. These models extend the “twins embedding” models presented in Boone et al. (2020a) and Boone et al. (2020b) beyond the narrow phase range in their original presentation. The first model, `spec2embed`, accurately predicts a supernova’s location in the twins embedding space from a single spectrum from -10 to $+40$ days after maximum brightness, allowing us to standardize supernova brightnesses, and thus determine supernova distances, with comparable precision to the Boone et al. (2020b) result using a wide range of spectra. The second model, `embed2spec`, allows us to use a forward-modeling approach to fitting observations to the twins embedding space, as it predicts a model spectral energy distribution given its phase and location in the twins embedding space.

There are a number of paths this research can progress. The `spec2embed` model can be used with existing instruments to precisely measure distances to nearby Type Ia supernovae, making possible measurements of growth of structure through observations of peculiar velocities, or to help constrain measurements of the Hubble constant. Combining the work of Chapters 4 and 6 would also be of immediate interest; the kernel density estimate technique applied to the SALT2 and SNEMO latent parameter spaces in Chapter 4 can easily be extended to the embedding space studied in Chapter 6, creating a fully generative, non-linear empirical model of Type Ia supernova spectral time-series data. This new generative model could be compared to the linear models of differing dimensionality studied in Chapter 4 as a means of quantitatively addressing the open question of the proper parametrization or description of Type Ia supernovae for cosmology. This model can also be used to generate low-resolution, noisy spectra (as we did in Chapter 3) or even photometric light curves, to see how well this information can be extracted from different types of measurements. The success of the `spec2embed` model in predicting the embedding coordinate across phases already suggests that the information content determining the absolute magnitude of a Type Ia supernova may be able to be extracted from data collected away from maximum brightness.

Upcoming surveys, like those that will be a part of the Rubin Observatory and Roman Space Telescope, will provide unprecedented numbers of supernovae, so many so that systematic biases will far outpace statistical uncertainties. Developing tools to identify, quantify, and mitigate these biases, as we have begun to do here, will be crucial in ensuring the accuracy of these cosmological measurements.

Bibliography

- Abbott, T. M. C., Allam, S., Andersen, P., et al. 2019, ApJ, 872, L30, doi: [10.3847/2041-8213/ab04fa](https://doi.org/10.3847/2041-8213/ab04fa)
- Aldering, G., Adam, G., Antilogus, P., et al. 2002, in Survey and Other Telescope Technologies and Discoveries, ed. J. A. Tyson & S. Wolff, Vol. 4836 (Spie), 61 – 72, doi: [10.1117/12.458107](https://doi.org/10.1117/12.458107)
- Bailey, S., Aldering, G., Antilogus, P., et al. 2009, Astronomy and Astrophysics, 500, L17, doi: [10.1051/0004-6361/200911973](https://doi.org/10.1051/0004-6361/200911973)
- Barbary, K., Boone, K., & Deil, C. 2015a, sep: v0.3.0, doi: [10.5281/zenodo.15669](https://doi.org/10.5281/zenodo.15669)
- Barbary, K., Rbiswas4, Rodney, S., et al. 2015b, Sncosmo: V1.0.0, Zenodo, doi: [10.5281/zenodo.592747](https://doi.org/10.5281/zenodo.592747)
- Benetti, S., Cappellaro, E., Mazzali, P. A., et al. 2005, ApJ, 623, 1011, doi: [10.1086/428608](https://doi.org/10.1086/428608)
- Bertin, E., & Arnouts, S. 1996, A&AS, 117, 393, doi: [10.1051/aas:1996164](https://doi.org/10.1051/aas:1996164)
- Betoule, M., Kessler, R., Guy, J., et al. 2014, A&A, 568, A22, doi: [10.1051/0004-6361/201423413](https://doi.org/10.1051/0004-6361/201423413)
- Biewald, L. 2020, Experiment Tracking with Weights and Biases. <https://www.wandb.com/>
- Blondin, S., & Tonry, J. L. 2007, ApJ, 666, 1024, doi: [10.1086/520494](https://doi.org/10.1086/520494)
- Blondin, S., Dessart, L., Leibundgut, B., et al. 2006, AJ, 131, 1648, doi: [10.1086/498724](https://doi.org/10.1086/498724)
- Blondin, S., Matheson, T., Kirshner, R. P., et al. 2012, AJ, 143, 126, doi: [10.1088/0004-6256/143/5/126](https://doi.org/10.1088/0004-6256/143/5/126)
- Bongard, S., Soulez, F., Thiébaut, É., & Pecontal, É. 2011, MNRAS, 418, 258, doi: [10.1111/j.1365-2966.2011.19480.x](https://doi.org/10.1111/j.1365-2966.2011.19480.x)
- Bonvin, V., Courbin, F., Suyu, S. H., et al. 2017, MNRAS, 465, 4914, doi: [10.1093/mnras/stw3006](https://doi.org/10.1093/mnras/stw3006)

- Boone, K. 2019, AJ, 158, 257, doi: 10.3847/1538-3881/ab5182
- Boone, K., Aldering, G., Antilogus, P., et al. 2020a, ApJ
—. 2020b, ApJ
- Branch, D., Dang, L. C., Hall, N., et al. 2006, PASP, 118, 560, doi: 10.1086/502778
- Buton, C., Copin, Y., Aldering, G., et al. 2013, A&A, 549, A8, doi: 10.1051/0004-6361/201219834
- Cardelli, J. A., Clayton, G. C., & Mathis, J. S. 1989, ApJ, 345, 245, doi: 10.1086/167900
- Chevallier, M., & Polarski, D. 2001, International Journal of Modern Physics D, 10, 213, doi: 10.1142/s0218271801000822
- Childress, M., Aldering, G., Antilogus, P., et al. 2013, ApJ, 770, 108, doi: 10.1088/0004-637X/770/2/108
- Childress, M. J., Wolf, C., & Zahid, H. J. 2014, MNRAS, 445, 1898, doi: 10.1093/mnras/stu1892
- Clarke, K. A. 2005, Conflict Management and Peace Science, 22, 341, doi: 10.1080/07388940500339183
- Conley, A., Guy, J., Sullivan, M., et al. 2011, ApJS, 192, 1, doi: 10.1088/0067-0049/192/1/1
- Eisenstein, D. J., Zehavi, I., Hogg, D. W., et al. 2005, ApJ, 633, 560, doi: 10.1086/466512
- Fakhouri, H. K., Boone, K., Aldering, G., et al. 2015, ApJ, 815, 58, doi: 10.1088/0004-637X/815/1/58
- Farrar, D. E., & Glauber, R. R. 1967, The Review of Economics and Statistics, 49, 92. <http://www.jstor.org/stable/1937887>
- Fitzpatrick, E. L. 1999, PASP, 111, 63, doi: 10.1086/316293
- Fitzpatrick, E. L., & Massa, D. 2007, ApJ, 663, 320, doi: 10.1086/518158
- Foley, R. J. 2012, ApJ, 748, 127, doi: 10.1088/0004-637X/748/2/127
- Foley, R. J., & Kasen, D. 2011, ApJ, 729, 55, doi: 10.1088/0004-637X/729/1/55
- Foley, R. J., Sanders, N. E., & Kirshner, R. P. 2011, ApJ, 742, 89, doi: 10.1088/0004-637X/742/2/89
- Freedman, W. L., Madore, B. F., Hoyt, T., et al. 2020, ApJ, 891, 57, doi: 10.3847/1538-4357/ab7339

- Frieman, J. A., Bassett, B., Becker, A., et al. 2008, AJ, 135, 338, doi: 10.1088/0004-6256/135/1/338
- Goodfellow, I. J., Bengio, Y., & Courville, A. 2016, Deep Learning (Cambridge, MA, USA: MIT Press)
- Guy, J., Astier, P., Baumont, S., et al. 2007, A&A, 466, 11, doi: 10.1051/0004-6361:20066930
- Hamuy, M., Phillips, M. M., Suntzeff, N. B., et al. 1996, AJ, 112, 2438, doi: 10.1086/118193
- Hogg, D. W. 1999, arXiv e-prints, astro. <https://arxiv.org/abs/astro-ph/9905116>
- Holz, D. E., & Hughes, S. A. 2005, ApJ, 629, 15, doi: 10.1086/431341
- Janesick, J. R. 2001, Scientific Charge-Coupled Devices (SPIE), doi: 10.1117/3.374903
- Jones, D. O., Riess, A. G., & Scolnic, D. M. 2015, ApJ, 812, 31, doi: 10.1088/0004-637X/812/1/31
- Jones, D. O., Riess, A. G., Scolnic, D. M., et al. 2018, ApJ, 867, 108, doi: 10.3847/1538-4357/aae2b9
- Kasen, D., Röpke, F. K., & Woosley, S. E. 2009, Nature, 460, 869, doi: 10.1038/nature08256
- Kasen, D., & Woosley, S. E. 2007, ApJ, 656, 661, doi: 10.1086/510375
- Kelly, P. L., Hicken, M., Burke, D. L., Mandel, K. S., & Kirshner, R. P. 2010, ApJ, 715, 743, doi: 10.1088/0004-637X/715/2/743
- Kelsey, L., Sullivan, M., Smith, M., et al. 2020, arXiv e-prints, arXiv:2008.12101. <https://arxiv.org/abs/2008.12101>
- Kessler, R., Becker, A. C., Cinabro, D., et al. 2009, ApJS, 185, 32, doi: 10.1088/0067-0049/185/1/32
- Kim, A. G., Thomas, R. C., Aldering, G., et al. 2013, ApJ, 766, 84, doi: 10.1088/0004-637x/766/2/84
- Komatsu, E., Dunkley, J., Nolta, M. R., et al. 2009, ApJS, 180, 330, doi: 10.1088/0067-0049/180/2/330
- Lampeitl, H., Smith, M., Nichol, R. C., et al. 2010, ApJ, 722, 566, doi: 10.1088/0004-637X/722/1/566
- Lantz, B., Aldering, G., Antilogus, P., et al. 2004, in Proc. SPIE, Vol. 5249, Optical Design and Engineering, ed. L. Mazuray, P. J. Rogers, & R. Wartmann, 146–155, doi: 10.1117/12.512493

- Linder, E. V. 2003, Phys. Rev. Lett., 90, 091301, doi: 10.1103/PhysRevLett.90.091301
- Léget, P. F., Gangler, E., Mondon, F., et al. 2020, A&A, 636, A46, doi: 10.1051/0004-6361/201834954
- Mandel, K. S., Foley, R. J., & Kirshner, R. P. 2014, ApJ, 797, 75, doi: 10.1088/0004-637X/797/2/75
- Mosher, J., Guy, J., Kessler, R., et al. 2014, ApJ, 793, 16, doi: 10.1088/0004-637X/793/1/16
- Muthukrishna, D., Parkinson, D., & Tucker, B. E. 2019, ApJ, 885, 85, doi: 10.3847/1538-4357/ab48f4
- Nair, V., & Hinton, G. E. 2010, in Proceedings of the 27th International Conference on International Conference on Machine Learning, Icml'10 (Madison, WI, USA: Omnipress), 807–814
- Nugent, P., Phillips, M., Baron, E., Branch, D., & Hauschildt, P. 1995, ApJ, 455, L147, doi: 10.1086/309846
- Paszke, A., Gross, S., Massa, F., et al. 2019, in Advances in Neural Information Processing Systems 32, ed. H. Wallach, H. Larochelle, A. Beygelzimer, F. d' Alché-Buc, E. Fox, & R. Garnett (Curran Associates, Inc.), 8024–8035
- Pedregosa, F., Varoquaux, G., Gramfort, A., et al. 2011, Journal of Machine Learning Research, 12, 2825. <http://www.jmlr.org/papers/v12/pedregosa11a.html>
- Perlmutter, S., Gabi, S., Goldhaber, G., et al. 1997, ApJ, 483, 565, doi: 10.1086/304265
- Perlmutter, S., Aldering, G., Goldhaber, G., et al. 1999, ApJ, 517, 565, doi: 10.1086/307221
- Phillips, M. M. 1993, ApJ, 413, L105, doi: 10.1086/186970
- Planck Collaboration, Ade, P. A. R., Aghanim, N., et al. 2016, A&A, 594, A13, doi: 10.1051/0004-6361/201525830
- Project Jupyter, Matthias Bussonnier, Jessica Forde, et al. 2018, in Proceedings of the 17th Python in Science Conference, ed. Fatih Akici, David Lippa, Dillon Niederhut, & M. Pacer, 113 – 120, doi: 10.25080/Majora-4af1f417-011
- Pskovskii, I. P. 1977, Soviet Ast., 21, 675
- Rest, A., Scolnic, D., Foley, R. J., et al. 2014, ApJ, 795, 44, doi: 10.1088/0004-637X/795/1/44
- Riess, A., Biretta, J., & Casertano, S. 1999, WFPC2 Instrument Science Report, 99. http://www.stsci.edu/hst/wfpc2/documents/isr/wfpc2_isr9904.pdf

- Riess, A. G., Casertano, S., Yuan, W., Macri, L. M., & Scolnic, D. 2019, ApJ, 876, 85, doi: 10.3847/1538-4357/ab1422
- Riess, A. G., Press, W. H., & Kirshner, R. P. 1996, ApJ, 473, 88, doi: 10.1086/178129
- Riess, A. G., Filippenko, A. V., Challis, P., et al. 1998, AJ, 116, 1009, doi: 10.1086/300499
- Riess, A. G., Macri, L. M., Hoffmann, S. L., et al. 2016, ApJ, 826, 56. <http://iopscience.iop.org/article/10.3847/0004-637X/826/1/56/meta>
- Rigault, M., Copin, Y., Aldering, G., et al. 2013, A&A, 560, A66, doi: 10.1051/0004-6361/201322104
- Rigault, M., Aldering, G., Kowalski, M., et al. 2015, ApJ, 802, 20, doi: 10.1088/0004-637X/802/1/20
- Rigault, M., Brinnel, V., Aldering, G., et al. 2018, arXiv e-prints, arXiv:1806.03849. <https://arxiv.org/abs/1806.03849>
- Roman Space Telescope Reference Information. 2019, Roman Space Telescope (WFIRST) Reference Information, Phase B. <https://roman.gsfc.nasa.gov/science/Roman%5FReference%5FInformation.html>
- Rose, B. M., Garnavich, P. M., & Berg, M. A. 2019, ApJ, 874, 32, doi: 10.3847/1538-4357/ab0704
- Rose, B. M., Dixon, S., Rubin, D., et al. 2020, ApJ, 890, 60, doi: 10.3847/1538-4357/ab698d
- Rubin, D. 2020a, ApJ, 897, 40, doi: 10.3847/1538-4357/ab12de
- . 2020b
- Rubin, D., Aldering, G., Barbary, K., et al. 2015, ApJ, 813, 137, doi: 10.1088/0004-637X/813/2/137
- Rust, B. W. 1974, PhD Thesis, Oak Ridge National Lab., TN.
- Sako, M., Bassett, B., Becker, A. C., et al. 2018, PASP, 130, 064002, doi: 10.1088/1538-3873/aab4e0
- Sasdelli, M., Ishida, E. E. O., Vilalta, R., et al. 2016, Monthly Notices of the Royal Astronomical Society, 461, 2044, doi: 10.1093/mnras/stw1228
- Saunders, C., Aldering, G., Antilogus, P., et al. 2018, ApJ, 869, 167, doi: 10.3847/1538-4357/aaec7e
- Schlegel, D. J., Finkbeiner, D. P., & Davis, M. 1998, ApJ, 500, 525, doi: 10.1086/305772

- Scolnic, D., & Kessler, R. 2016, ApJ, 822, L35, doi: 10.3847/2041-8205/822/2/L35
- Scolnic, D. M., Riess, A. G., Foley, R. J., et al. 2014, ApJ, 780, 37, doi: 10.1088/0004-637x/780/1/37
- Scolnic, D. M., Jones, D. O., Rest, A., et al. 2018, ApJ, 859, 101, doi: 10.3847/1538-4357/aab9bb
- Scott, D. W. 2014, Multivariate density estimation: theory, practice, and visualization, second edition edn. (Hoboken, New Jersey: Wiley)
- Silverman, B. W. 1986, Density estimation for statistics and data analysis, Monographs on Statistics and Applied Probability (London: Chapman and Hall). <https://cds.cern.ch/record/1070306>
- Silverman, J. M., Kong, J. J., & Filippenko, A. V. 2012, Monthly Notices of the Royal Astronomical Society, 425, 1819, doi: 10.1111/j.1365-2966.2012.21269.x
- Smith, M., Sullivan, M., Wiseman, P., et al. 2020, MNRAS, 494, 4426, doi: 10.1093/mnras/staa946
- Snoek, J., Larochelle, H., & Adams, R. P. 2012, arXiv:1206.2944 [cs, stat]. <http://arxiv.org/abs/1206.2944>
- Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. 2014, Journal of Machine Learning Research, 15, 1929. <http://jmlr.org/papers/v15/srivastava14a.html>
- Stahl, B. E., Martínez-Palomera, J., Zheng, W., et al. 2020, MNRAS, 496, 3553, doi: 10.1093/mnras/staa1706
- Sullivan, M., Conley, A., Howell, D. A., et al. 2010, MNRAS, 406, 782, doi: 10.1111/j.1365-2966.2010.16731.x
- Sullivan, M., Guy, J., Conley, A., et al. 2011, ApJ, 737, 102, doi: 10.1088/0004-637X/737/2/102
- Tenenbaum, J. B. 2000, Science, 290, 2319, doi: 10.1126/science.290.5500.2319
- Tripp, R. 1998, A&A, 331, 815
- Tripp, R., & Branch, D. 1999, ApJ, 525, 209, doi: 10.1086/307883
- Wand, M. P., & Jones, M. C. 1995, Kernel smoothing, 1st edn., Monographs on statistics and applied probability No. 60 (London ; New York: Chapman & Hall)
- Wang, X., Wang, L., Filippenko, A. V., Zhang, T., & Zhao, X. 2013, Science, 340, 170, doi: 10.1126/science.1234414

Wang, X., Filippenko, A. V., Ganeshalingam, M., et al. 2009, ApJ, 699, L139, doi: 10.1088/0004-637x/699/2/1139

Wittman, D. M., Tyson, J. A., Kirkman, D., Dell'Antonio, I., & Bernstein, G. 2000, Nature, 405, 143, doi: 10.1038/35012001

Wooldridge, J. M. 2013, Introductory econometrics: a modern approach, 5th edn. (Mason, OH: South-Western Cengage Learning)

Zheng, W., Kelly, P. L., & Filippenko, A. V. 2018, ApJ, 858, 104, doi: 10.3847/1538-4357/aabaeb

Appendix A

Optimal Savitzky-Golay Smoothing Windows

We wish to find the optimal size of a Savitzky-Golay smoothing window to best estimate the true flux of a spectrum, which we call \hat{Y} . We call the observed, noisy spectrum Y , and the smoothed spectrum Y' . Our noisy spectrum can be written as

$$Y = \hat{Y} + N \quad (\text{A.1})$$

where N is a noise vector. An ideal smoothing algorithm will make Y' as close to \hat{Y} as possible, that is it should minimize

$$\|Y' - \hat{Y}\| \equiv (Y' - \hat{Y})^\dagger W (Y' - \hat{Y}) \quad (\text{A.2})$$

where W is a weight matrix given by the inverse of the covariance matrix of the observed noisy spectrum. We assume no covariance between the wavelength bins, so W is a diagonal matrix.

With the Savitzky-Golay algorithm, we have

$$Y' = B_w Y \quad (\text{A.3})$$

where B_w is a matrix that is a function of the size of the smoothing window w . Additionally, smoothing the true spectrum should give us the true spectrum:

$$B_w \hat{Y} = \hat{Y} \quad (\text{A.4})$$

In order to find the optimal value of w , we need an estimator of the unknown distance between the smoothed estimate and the true flux value. Using the definitions above, and $E[N^\dagger A N] = \text{Tr}(A)$ where N is a Gaussian random variable, we can rewrite

$$\begin{aligned}
||Y' - \hat{Y}|| &= (Y' - Y + N)^\dagger W (Y' - Y + N) \\
&= (Y' - Y)^\dagger W (Y' - Y) + N^\dagger W (Y' - Y) + (Y' - Y)^\dagger W N + N^\dagger W N \\
&= ||Y' - Y|| + 2N^\dagger W (Y' - Y) + N^\dagger W N \\
&= ||Y' - Y|| + 2N^\dagger W (B_w - I) Y + N^\dagger W N \\
&= ||Y' - Y|| + 2N^\dagger W (B_w - I) (\hat{Y} + N) + N^\dagger W N \\
&= ||Y' - Y|| + 2N^\dagger W (B_w \hat{Y} - \hat{Y}) + 2N^\dagger W B_w N - N^\dagger W N \\
&= ||Y' - Y|| + 2N^\dagger W B_w N - N^\dagger W N
\end{aligned}$$

Using properties of quadratic forms and the definition of the weight matrix W and the noise vector N , for any symmetric matrix A , we have

$$E[N^\dagger A N] = \text{Tr}(A W^{-1}) \quad (\text{A.5})$$

So, our estimator becomes

$$E[||Y' - \hat{Y}||] = E[||Y' - Y||] + 2\text{Tr}(B_w) - n \quad (\text{A.6})$$

where n is the rank of the noise vector N . Thus, the optimal window size s is given by

$$s = \arg \max_w (||Y' - Y|| + 2\text{Tr}(B_w) - n) \quad (\text{A.7})$$

Appendix B

Table of Si II $\lambda 6355$ Velocities and Pseudo-equivalent Widths

Table B.1: The baseline spectral feature data used throughout the Chapter 3 analysis for both the training Nearby Supernova Factory data and the validation BSNIP sample.

Name	v_{Si} (km/s)	Δv_{Si} (km/s)	pEW (Å)	Δ pEW (Å)
SNF20060911-014	11301	266	78.6	6.1
PTF11bnx	10225	591	125.2	5.5
PTF09dn1	12300	50	92.5	2.2
PTF09dnp	13176	27	92.5	2.8
SNF20080514-002	11038	250	95.9	2.0
SNBOSS38	10789	315	81.2	0.7
SN2006ob	11221	795	134.9	5.6
SNF20070817-003	10875	839	134.9	5.9
SNNGC4424	10424	57	73.2	1.2
PTF10qyz	11054	397	155.5	4.9
LSQ12fxd	11043	331	80.5	2.5
SNF20080821-000	11409	296	99.5	8.0
SN2010dt	10667	391	124.8	3.3
SNF20080623-001	11085	236	110.1	4.6
LSQ12fhe	11665	302	37.0	2.5
PTF11mty	9770	478	122.5	10.6
SNF20080512-010	11523	549	122.6	6.6
PTF10tce	12690	236	110.8	2.6

Table continues

Continued

Name	v_{Si} (km/s)	Δv_{Si} (km/s)	pEW (Å)	Δ pEW (Å)
SN2005ir	13804	197	150.6	7.0
PTF10nlg	11416	565	165.1	4.4
PTF10hmv	9604	282	75.5	3.4
SNF20071015-000	10786	73	101.1	3.5
SNhunt89	11966	792	144.6	4.9
SNF20080707-012	12467	520	166.0	9.3
PTF09dlc	10428	437	99.4	6.4
SN2012cu	10573	309	75.5	0.6
SNF20080919-000	10193	391	98.3	4.3
SNF20080919-001	9970	395	58.1	3.7
SN2010kg	13201	129	154.2	1.1
SNF20080714-008	11752	1260	168.4	8.0
SNF20061111-002	10410	159	133.6	5.2
SNNGC6343	10980	266	114.1	2.5
PTF10wof	12078	311	134.9	3.0
PTF10ufj	10792	280	120.5	6.4
CSS130502_01	11022	52	102.0	3.4
SNF20080626-002	11629	194	83.1	1.9
SNF20060621-015	10488	527	98.0	2.7
SNF20070701-005	9612	295	86.4	6.0
SNF20080909-030	8925	154	69.7	3.4
SNNGC2691	11140	127	37.8	1.8
PTF13asv	9848	354	55.0	2.7
CSS110918_01	12195	128	83.2	2.5
CSS110918_02	10659	308	107.4	3.3
SNF20080918-002	10719	309	117.3	6.8
SNIC3573	10833	570	110.7	5.1
SNF20080918-000	11845	440	103.7	5.9
SNF20060512-002	12749	259	107.5	3.7
PTF12iiq	14955	266	176.7	1.7
SNF20070725-001	10157	553	93.8	5.8
SNF20070902-018	11237	715	146.4	9.3
LSQ12gxj	11777	532	97.6	4.6
SN2006cj	10964	377	90.8	4.2
SNF20080815-017	11149	751	142.9	12.2
PTF10icb	10325	198	87.2	0.9
SNF20061022-005	9576	311	76.8	6.9

Table continues

Continued

Name	v_{Si} (km/s)	Δv_{Si} (km/s)	pEW (Å)	Δ pEW (Å)
PTF13azs	10647	650	128.1	7.5
SNF20080612-003	11484	167	83.4	3.6
SNF20070820-000	9859	509	127.3	6.5
SNF20071108-021	9101	582	74.3	8.4
PTF12evo	10068	535	106.4	8.4
SNF20050624-000	10393	679	104.7	8.8
SNF20080531-000	11386	334	119.9	3.4
PTF12ikt	10155	442	101.8	3.9
PTF12eer	10418	687	175.5	13.0
SN2005hj	10542	600	72.6	4.7
SNPGC51271	11264	173	113.5	4.1
SNPGC027923	10930	255	69.7	2.2
SNF20080522-011	10929	223	98.2	4.2
SNhunt46	10975	236	91.7	1.8
SNF20071021-000	11636	297	123.6	1.7
SNF20070831-015	10421	467	93.5	5.5
SNF20080806-002	10666	324	99.4	4.1
SNF20080822-005	10553	311	76.4	5.8
PTF10zdk	9766	316	106.9	2.7
SNF20070424-003	10957	425	122.0	6.6
SNF20070714-007	9446	533	129.3	2.9
PTF11cao	12017	518	144.5	2.5
SNF20061021-003	10932	445	115.0	4.6
PTF11pdk	10379	265	129.8	8.2
SNF20070506-006	9734	229	67.3	3.8
SN2005hc	11117	458	102.8	4.0
SN2007bd	12412	472	125.4	2.4
PTF10wnm	11452	227	102.1	5.6
SN2010ex	10598	452	85.0	3.8
SNF20080802-006	11827	933	139.0	9.7
PTF10xyt	12134	141	129.2	6.9
SNF20070727-016	10518	127	72.4	4.6
SNF20070818-001	12023	668	158.2	11.9
SNF20070427-001	10330	491	87.2	7.0
PTF10ops	9617	1520	207.9	20.9
SNF20080614-010	10635	418	145.7	6.4
PTF10ndc	11145	192	102.0	6.4

Table continues

Continued

Name	v_{Si} (km/s)	Δv_{Si} (km/s)	pEW (Å)	Δ pEW (Å)
SNF20080323-009	9902	246	107.7	8.4
SNF20080516-022	9136	278	106.6	4.3
SNF20070802-000	11799	674	145.1	8.9
SN2004gs	11180	45	137.0	1.5
SNF20050824-002	11286	1034	133.9	13.1
PTF10mwb	10816	313	117.5	1.9
PTF09foz	10183	220	115.7	2.4
PTF09fox	11345	603	109.8	3.8
SNF20061030-010	11365	172	114.3	2.6
SNF20061020-000	11406	422	119.0	3.3
PTF12ena	10758	334	86.8	2.1
PTF13anh	8674	485	133.2	4.9
SNF20080725-004	11216	109	110.1	6.9
SNF20070403-001	11055	596	139.3	10.5
LSQ12dbr	10486	116	72.3	3.1
SNF20070902-021	10452	685	123.1	5.8
PTF11mkx	9592	529	56.0	6.8
SNF20060907-000	10176	237	117.1	2.9
SNF20080717-000	12735	408	99.9	5.4
SNF20080825-010	11233	616	127.9	3.6
SNF20070717-003	9812	876	149.6	6.9
SN2004ef	11924	502	134.8	2.2
SNNGC7589	13249	138	81.1	2.6
SNF20080516-000	10809	627	101.7	8.5
SNF20080620-000	10876	546	127.2	3.6
SNF20050728-006	11324	407	128.5	6.7
SNF20050821-007	10567	332	96.3	5.0
SNF20080920-000	12995	432	109.0	4.7
SNF20080918-004	10161	430	124.7	4.5
SN2012fr	12085	36	67.2	0.7
SNF20080610-000	10739	341	129.9	6.5
SNF20060512-001	8890	295	58.2	4.7
SN2007kk	11833	407	117.9	5.1
PTF11bgv	9681	218	99.9	3.2
SNF20070806-026	11418	744	126.5	3.2
PTF11kly	10502	89	97.3	0.3
SNF20060912-000	9392	309	104.2	7.5

Table continues

Continued

Name	v_{Si} (km/s)	Δv_{Si} (km/s)	pEW (Å)	Δ pEW (Å)
SNF20071003-016	11002	824	133.7	6.4
SNF20070803-005	9719	252	25.4	4.0
PTF12fuu	10872	502	96.3	6.7
SN2007nq	11954	490	127.2	3.2
SNF20080810-001	10446	340	111.5	2.0
SN2008ec	10892	170	120.8	1.0
SNF20060618-014	11717	754	123.3	6.9
SNNGC2370	10878	121	120.5	2.4
SNF20070403-000	9413	663	122.7	7.6
SNNGC4076	9488	432	81.8	2.5
SNF20070630-006	11130	321	124.8	12.0
SNF20061108-004	11839	404	129.2	9.1
SNF20080914-001	9290	301	93.2	2.5
SNF20060609-002	10734	497	89.9	3.9
SN2006do	12363	153	120.4	2.4
SN2006dm	11280	363	121.1	1.1
SNF20060526-003	11200	385	100.7	5.5
PTF10qjq	10884	24	92.1	2.4
SNF20080507-000	10752	268	98.6	7.0
SNF20070330-024	12422	56	85.5	3.6
PTF11bju	10683	260	45.3	3.9
SNF20060521-001	10712	482	119.6	5.4
PTF11drz	11295	434	129.0	4.4
SNNGC0927	11726	288	106.8	2.5
PTF12dym	10654	823	146.1	6.7
SNF20080803-000	11231	475	112.0	7.1
SNF20060511-014	10345	289	132.0	5.1
PTF12ghy	10657	586	107.7	2.8
SNF20070531-011	11694	588	130.4	2.8
SNF20080522-000	11054	553	52.0	4.8
SNF20080510-005	11391	566	104.0	8.6
SNF20070712-003	9892	608	110.6	5.5
SNF20080913-031	10075	728	93.9	7.8
SNF20080510-001	11347	438	133.5	6.3

Appendix C

Kernel Density Estimation

C.1 KDE in 1 Dimension

Let's first consider the one-dimensional case. Fig. C.1 shows a histogram for a series of observations drawn from an exponentially-modified Gaussian distribution. We choose this distribution to illustrate the impact of non-Gaussianity (in particular skewness) on the resulting estimates of the underlying probability distribution. The true probability distribution function is shown as a black line in Fig. C.1. The remaining lines in Fig. C.1 show the resulting kernel density estimates using a Gaussian kernel with a range of bandwidth values. We can qualitatively see that a bandwidth of 0.53 results in an estimate that closely resembles the true probability distribution, while a value of 0.1 is too narrow (overfitting/high variance) and 3 is too wide (underfitting/high bias).

There are some general rules-of-thumb for selecting an optimal bandwidth; for example, Silverman's rule of thumb:

$$h = 0.9 \times \min \left(\hat{\sigma}, \frac{\text{IQR}}{1.34} \right) \times n^{-1/5}$$

where $\hat{\sigma}$ is the standard deviation of the sample, IQR is the interquartile range of the sample, and n is the number of data points in the sample (Silverman, 1986). This rule can perform quite well in most circumstances. However, these types of rules-of-thumb generalize quite poorly when the data is highly non-Gaussian (e.g. multi-modal distributions) or when the dimensionality of the data is high.

Another way to find the best bandwidth parameter is to try several values of the parameter and compare the results via some metric. A commonly used method for performing this type of evaluation is k -fold cross-validation. In k -fold cross-validation, we split the sample into k groups. We then hold one of these groups out and train the model on the examples in the remaining $k - 1$ groups. The model is then evaluated on the examples in the held-out set. This process is repeated until each example in the data set has been used in the training and validation. The overall model evaluation is then usually taken to be the mean of the

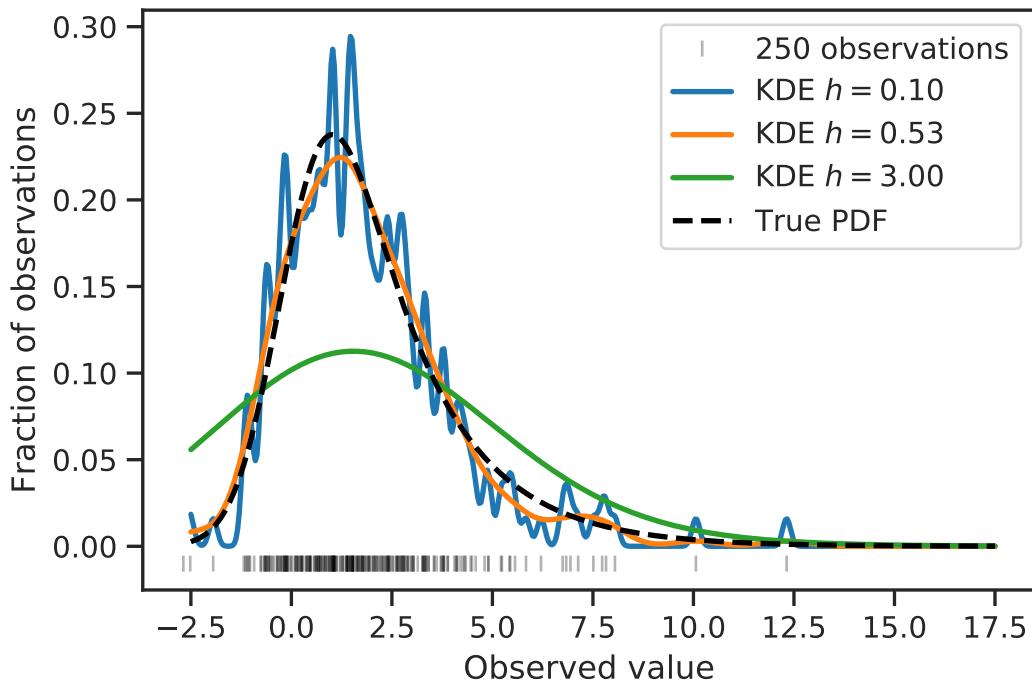


Figure C.1: Example of a kernel density estimate of a 1-dimensional, non-Gaussian distribution. The ticks represent the $n = 250$ observations drawn from the true distribution shown with the black dashed line. The blue, orange, and green lines represent KDEs with increasing bandwidth sizes.

evaluation metrics found in each of the cross-validation rounds, and the standard deviation of these metrics can be used to estimate the uncertainty on that metric.

Using this technique on our toy example from above, we find an optimal bandwidth of 0.53, as evidenced by the maximum at this point in Fig. C.2, plotting the total log probability of the k -folds as a function of bandwidth parameter. This differs from the location of the rule-of-thumb estimate; however, the difference in the average log-likelihoods of the data under a model with the rule-of-thumb bandwidth and the data under a model with the cross-validation selected bandwidth is small. Because the rule-of-thumb estimates have many downsides (and indeed are not well-defined in multiple dimensions) and because cross-validation gives us bandwidths that perform equally well or better, we proceed with the cross-validation bandwidth selection.

C.2 Appendix: KDE in d Dimensions

The case in $d > 1$ is quite similar to the 1-dimensional case. The kernel, though, is now in d dimensions, and as such, the bandwidth is no longer a single parameter but a $d \times d$

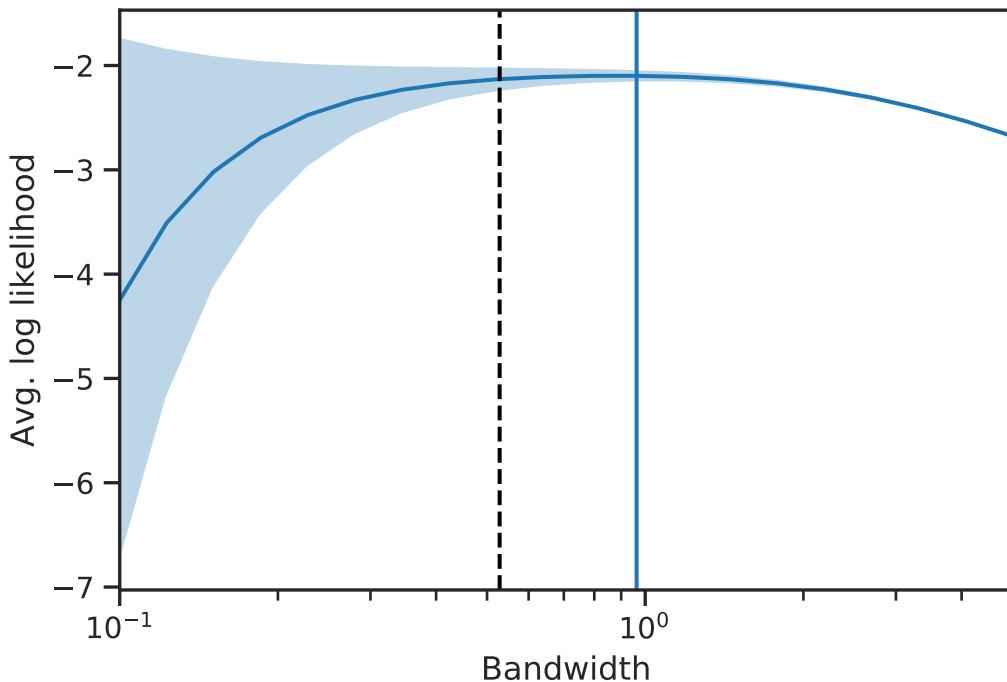


Figure C.2: Results from 5-fold cross-validation for the example distribution and data shown in Fig. C.1. The blue line and shading show the mean and standard deviation of the normalized scores from all of the cross-validation subsets. The vertical blue line shows the location of the maximum score. The dashed black line shows the location of the Silverman’s rule-of-thumb estimate for the optimal bandwidth.

symmetric matrix describing the bandwidth in each dimension along with the correlations between dimensions. The higher-dimensional case is further complicated by the so-called “curse of dimensionality”, where the sparseness of the data in the higher dimensional space causes the density estimate to converge more slowly. We will explore this first detail with a similar toy model to that used in C.1 but in 2 dimensions for ease of visualization.

Consider a 2-dimensional sample drawn from a bivariate Gaussian distribution with mean μ and covariance Σ (i.e. $(y_1, y_2) \sim \mathcal{N}(\mu, \Sigma)$) and then consider a non-linear transformation to that sample, transforming y_2 to $z_2 = \exp(y_2)$.

This example joint distribution is correlated and nonlinear, and as such works as a good test of the power and limitations of our methodology. We start by making estimates width a Gaussian kernel for a range of bandwidths as we did in the 1-dimensional case, giving us the results shown in Fig. C.3. However, the standard `sklearn` implementation of the kernel density estimator only allows for the a kernel with a covariance matrix proportional to the identity matrix, so in our k -fold cross-validation, we are tuning a single bandwidth parameter h .

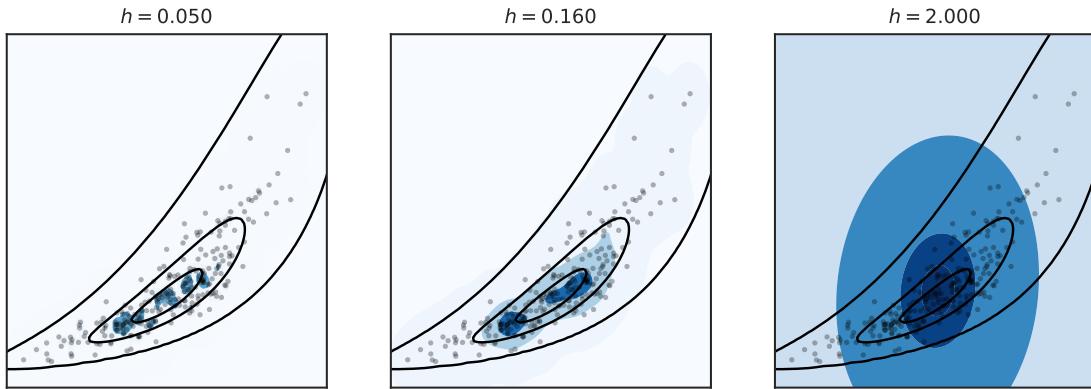


Figure C.3: Example kernel density estimates in 2 dimensions for a joint probability distribution that is correlated and non-Gaussian. The $n = 250$ data points are shown as black dots. The black contour lines show the true 1, 2, and 3σ contours, while the fading shades of blue show the 1, 2, and 3σ estimates of the KDE. In each of these examples, we assume a Gaussian kernel with no covariance terms.

We can see in the central panel of this figure that even the best-fit value of $h = 0.143$ obtained from k -fold cross-validation is not a very good estimate of the true joint probability density. In particular, the density estimate makes the distribution appear multimodal. This occurs because the diagonal kernel does not reflect the full range of scales in the different dimensions of the space, so it takes on an h value that reflects an average length scale.

We can improve our estimate by allowing the data to dictate the form of the kernel. We want to understand the relative fractions of the dispersion of the data in each dimension, as well as how the data covaries in each dimension. This will allow us to create an *effective* Gaussian kernel with a covariance matrix that is more reflective of the data, and therefore give us a better fit. We do this by finding a whitening transformation of the data, i.e. a transformation W that turns our data matrix X with covariance Σ_X into a data matrix $Y = WX$ with covariance $\Sigma_Y = \mathbb{I}$. With this whitening transformation, we can reproject our data into whitened space, fit the KDE as usual, and then apply the inverse transformation to the KDE sample to obtain a KDE with a tuned kernel that matches the data.

A commonly used whitening matrix is $W = \Lambda^{-1/2}U^\top$, where Λ is the diagonal matrix of eigenvalues and U is the matrix whose columns are the eigenvectors of the covariance matrix Σ_x . A matrix W that satisfies $W^\top W = \Sigma_X^{-1}$ is a whitening matrix, i.e. if X is a data matrix

with covariance Σ_X , then $Y = WX$ has covariance $\Sigma_Y = \mathbb{I}$.

$$\begin{aligned}\text{cov}(Y) &= \text{cov}(WX) \\ &= W\text{cov}(X)W^\top \\ &= W\Sigma_X W^\top \\ &= W(W^\top W)^{-1}W^\top \\ &= WW^{-1}(W^\top)^{-1}W^\top = \mathbb{I}\end{aligned}$$

Because the covariance matrix is positive-definite and symmetric, we can diagonalize it, finding a decomposition

$$\Sigma_X = U\Lambda U^T$$

where Λ is a diagonal matrix whose entries are the eigenvalues of the matrix and where the columns of U are the eigenvectors. Then,

$$\Sigma_X^{-1} = (U\Lambda U^T)^{-1} = U\Lambda^{-1}U^T = (\Lambda^{-1/2}U^T)^\top(\Lambda^{-1/2}U^T)$$

and thus $W = \Lambda^{-1/2}U^T$ is a whitening transformation.

Using this transformation, we can complete the process of transforming our data, fitting the KDE, and inverting the transformation to find a better fit to the data distribution. Fig. C.4 shows an explicit example, and Fig. C.5 directly compares the estimates of joint probability distribution that come from fits with and without this process. The predicted likelihood distribution is no longer multimodal, and the overall similarity between the distributions is improved.

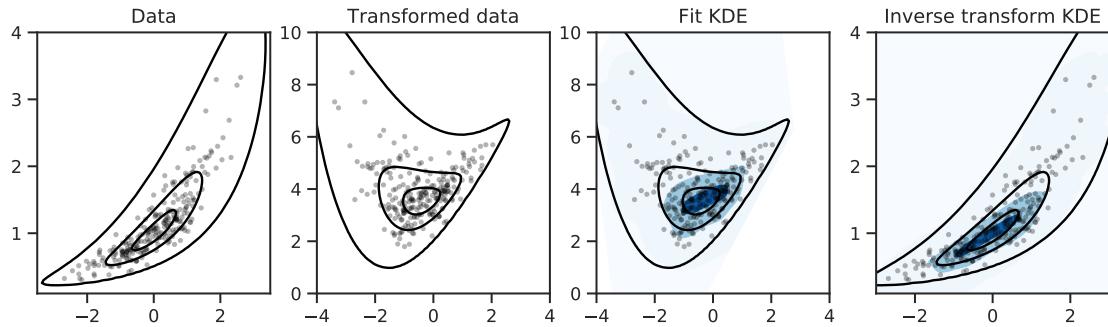


Figure C.4: The process of transforming the data to use a kernel that captures the data covariance. The first panel shows the data in the original coordinates. The second panel shows the data after being transformed by the whitening transformation. The third panel shows the KDE fit with a normal Gaussian kernel, and the final panel shows the data and the KDE reprojected back into the original coordinates. In each panel, the same data points are shown as black dots. The true 1 , 2 , and 3σ contours are shown in differing shades of blue. The black lines indicate the 1 , 2 , and 3σ confidence intervals of the best-fit KDE.

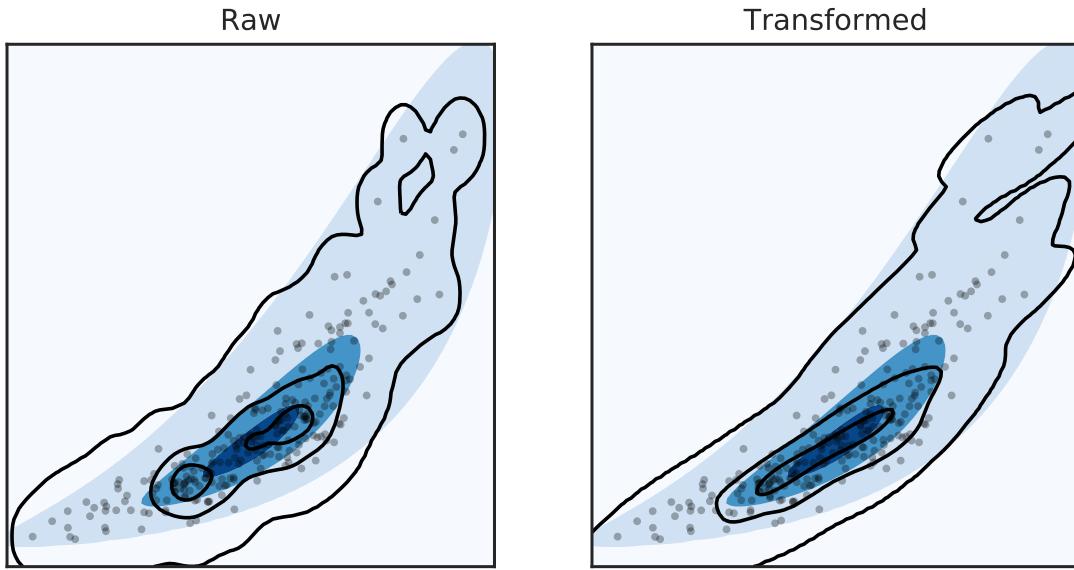


Figure C.5: Comparison of the final KDE for our toy example using a) the best-fit Gaussian kernel with no covariance transformation and b) the best-fit Gaussian kernel with whitening applied. In both panels, the blue shading indicates the 1 , 2 , and 3σ contours of the true toy model distribution, and the black lines indicate similar contours for the kernel density estimates of the distribution.

Appendix D

Measuring Spectral Features

To measure spectral features from real data (with associated flux noise), we first smooth the spectrum by convolving the observed flux $f(\lambda)$ with a Gaussian window weighted by the inverse of the flux variance, as is done in Blondin et al. (2006), to get a smoothed spectrum $f_s(\lambda)$. Simulated spectra have no noise, so we do not smooth them. We do however interpolate both the smoothed, observed data and the simulated spectra onto a wavelength grid with 0.1 Å resolution so that our identification of local extrema is not limited by the data resolution. For each feature region, we define a pseudo-continuum by identifying the local maxima λ_b and λ_r in the wavelength ranges blueward and redward of each feature described in Table D.1 and calculating the line that connects these two points. We divide the flux by that line to obtain the pseudo-continuum-removed local feature spectrum $f_c(\lambda)$. The velocity of each line is determined by finding the wavelength of the minimum of the pseudo-continuum-removed local feature spectrum (λ_{\min}) and using this value in the relativistic Doppler formula along with the rest-frame minimum wavelength λ_0 listed for each feature in Table D.1:

$$\frac{v}{c} = \frac{(\lambda_{\min}/\lambda_0)^2 - 1}{(\lambda_{\min}/\lambda_0)^2 + 1} \quad (\text{D.1})$$

The pseudo-equivalent width is calculated by integrating

$$\text{pEW} = \int_{\lambda_b}^{\lambda_r} \left[1 - \frac{f_s(\lambda)}{f_c(\lambda)} \right] d\lambda \quad (\text{D.2})$$

`spectral_lines`¹ is a package for measuring these spectral features using this inverse-variance weighted Gaussian smoothing, the Savitsky-Golay smoothing technique used for the spectral feature measurements in Chapter 3, and a simple Gaussian model of each absorption feature.

¹https://github.com/sam-dixon/spectral_lines

Feature name	λ_b range (Å)	λ_r range (Å)	λ_0 (Å)
Ca II H&K	3504 - 3687	3887 - 3990	3945
Si II λ 5972	5550 - 5681	5850 - 6015	5972
Si II λ 6355	5850 - 6015	6250 - 6365	6355

Table D.1: Extrema limits and rest-frame minimum wavelengths for the spectral indicators studied.

Appendix E

Regression Bias Derivations

E.1 Derivation of Regression Parameters in the Simultaneous Case

In ordinary least-squares regression, our goal is to minimize the loss function, L , defined by the square of the residuals between the y values predicted by our model ($\hat{\mathbf{Y}} \equiv \hat{\beta}_1 \mathbf{x}_1 + \hat{\beta}_2 \mathbf{x}_2 \equiv \mathbf{X}\hat{\boldsymbol{\beta}}$) and the data. This is equivalent to maximizing the likelihood, assuming the residuals are Gaussian.

$$\begin{aligned} L &= \|\hat{\mathbf{Y}} - \mathbf{Y}\|^2 \\ &= (\mathbf{X}\hat{\boldsymbol{\beta}} - \mathbf{Y})^T(\mathbf{X}\hat{\boldsymbol{\beta}} - \mathbf{Y}) \\ &= \hat{\boldsymbol{\beta}}^T \mathbf{X}^T \mathbf{X} \hat{\boldsymbol{\beta}} - \hat{\boldsymbol{\beta}}^T \mathbf{X}^T \mathbf{Y} - \mathbf{Y}^T \mathbf{X} \hat{\boldsymbol{\beta}} + \mathbf{Y}^T \mathbf{Y} \end{aligned}$$

We can minimize this by taking the gradient as a function of $\hat{\boldsymbol{\beta}}$ and setting it equal to zero.

$$\begin{aligned} \frac{\partial L}{\partial \hat{\boldsymbol{\beta}}} &= (\mathbf{X}^T \mathbf{X} \hat{\boldsymbol{\beta}})^T + \hat{\boldsymbol{\beta}}^T \mathbf{X}^T \mathbf{X} - (\mathbf{X}^T \mathbf{Y})^T - \mathbf{Y}^T \mathbf{X} \\ &= 2\hat{\boldsymbol{\beta}}^T \mathbf{X}^T \mathbf{X} - 2\mathbf{Y}^T \mathbf{X} \end{aligned}$$

$$\frac{\partial L}{\partial \hat{\boldsymbol{\beta}}} = 0 \Rightarrow \hat{\boldsymbol{\beta}} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{Y}$$

Plugging in our definition of \mathbf{Y} , we get

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T (\mathbf{X} \boldsymbol{\beta} + \boldsymbol{\epsilon}) \tag{E.1}$$

Since, by definition, $\langle \boldsymbol{\epsilon} \rangle = 0$, $\langle \hat{\boldsymbol{\beta}} \rangle = \boldsymbol{\beta}$, we can show that the spread of the residuals ($\mathbf{r} \equiv \hat{\mathbf{Y}} - \mathbf{Y}$) is simply $\sqrt{\sigma_{\text{int}}^2 + \sigma_{\text{obs}}^2}$:

$$\begin{aligned}\text{var}(\mathbf{r}) &= \langle \mathbf{r}^2 \rangle - \langle \mathbf{r} \rangle^2 \\ &= \langle (\mathbf{X}\hat{\boldsymbol{\beta}} - \mathbf{X}\boldsymbol{\beta} - \boldsymbol{\epsilon})(\mathbf{X}\hat{\boldsymbol{\beta}} - \mathbf{X}\boldsymbol{\beta} - \boldsymbol{\epsilon})^T \rangle - \langle (\mathbf{X}\hat{\boldsymbol{\beta}} - \mathbf{X}\boldsymbol{\beta} - \boldsymbol{\epsilon}) \rangle^2 \\ &= \langle \boldsymbol{\epsilon}\boldsymbol{\epsilon}^T \rangle - \langle \boldsymbol{\epsilon} \rangle^2 \\ &= \text{var}(\boldsymbol{\epsilon}) = \sigma_{\text{int}}^2 + \sigma_{\text{obs}}^2\end{aligned}$$

The variance on these regression coefficients can also be calculated. First, we calculate $\langle \hat{\boldsymbol{\beta}}^2 \rangle$:

$$\begin{aligned}\langle \hat{\boldsymbol{\beta}}^2 \rangle &= \langle \hat{\boldsymbol{\beta}}\hat{\boldsymbol{\beta}}^T \rangle \\ &= \langle (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}\mathbf{Y}^T\mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1} \rangle \\ &= \langle (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T(\mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon})(\boldsymbol{\beta}^T\mathbf{X}^T + \boldsymbol{\epsilon})\mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1} \rangle \\ &= \boldsymbol{\beta}\boldsymbol{\beta}^T + \sigma_{\text{int}}^2(\mathbf{X}^T\mathbf{X})^{-1}\end{aligned}$$

Then, by using the definition $\langle \hat{\boldsymbol{\beta}} \rangle^2 = \boldsymbol{\beta}\boldsymbol{\beta}^T$, we have

$$\text{var}(\hat{\boldsymbol{\beta}}) = \langle \hat{\boldsymbol{\beta}}^2 \rangle - \langle \hat{\boldsymbol{\beta}} \rangle^2 = (\sigma_{\text{int}}^2 + \sigma_{\text{obs}}^2)(\mathbf{X}^T\mathbf{X})^{-1}.$$

Calculating the individual components of this variance matrix in our two-dimensional case gives

$$\text{var}(\hat{\beta}_1) = \frac{\sigma_{\text{int}}^2 + \sigma_{\text{obs}}^2}{N\sigma_1^2(1-\rho^2)} \quad \text{and} \quad \text{var}(\hat{\beta}_2) = \frac{\sigma_{\text{int}}^2 + \sigma_{\text{obs}}^2}{N\sigma_2^2(1-\rho^2)}. \quad (\text{E.2})$$

E.2 Derivation of Biases on Regression Parameters in the Non-Simultaneous Case

Here we derive the biases on the regression parameters that occur when we perform bivariate linear regression one covariate at a time. We assume without loss of generality that we fit \mathbf{Y} as a function of \mathbf{x}_1 , giving an estimate of the slope that we denote $\hat{\beta}'_1$.

We can modify Equation E.1 to obtain the predicted value of this slope in the first fit:

$$\begin{aligned}\langle \hat{\beta}'_1 \rangle &= \langle (\mathbf{x}_1^T\mathbf{x}_1)^{-1}\mathbf{x}_1^T\mathbf{Y} \rangle \\ &= \langle (\mathbf{x}_1^T\mathbf{x}_1)^{-1}\mathbf{x}_1^T\mathbf{x}_1\beta_1 + (\mathbf{x}_1^T\mathbf{x}_1)^{-1}\mathbf{x}_1^T\mathbf{x}_2\beta_2 + (\mathbf{x}_1^T\mathbf{x}_1)^{-1}\mathbf{x}_1^T\boldsymbol{\epsilon} \rangle \\ &= \beta_1 + \beta_2\langle (\mathbf{x}_1^T\mathbf{x}_1)^{-1}\mathbf{x}_1^T\mathbf{x}_2 \rangle \\ &= \beta_1 + \frac{\beta_2\rho\sigma_2}{\sigma_1}\end{aligned}$$

The residuals from this first regression are

$$\begin{aligned}\mathbf{r}_1 &= \mathbf{Y} - \hat{\mathbf{Y}}_1 \\ &= \beta_1 \mathbf{x}_1 + \beta_2 \mathbf{x}_2 + \boldsymbol{\epsilon} - \hat{\beta}'_1 \mathbf{x}_1 \\ &= \beta_2 \mathbf{x}_2 - \frac{\beta_2 \rho \sigma_2}{\sigma_1} \mathbf{x}_1 + \boldsymbol{\epsilon}\end{aligned}$$

We can go through a similar analysis to find the predicted secondary effect from fitting the residuals of the first regression \mathbf{r}_1 as a function of \mathbf{x}_2 . This gives

$$\begin{aligned}\langle \hat{\beta}'_2 \rangle &= \langle (\mathbf{x}_2^T \mathbf{x}_2)^{-1} \mathbf{x}_2^T \mathbf{r}_1 \rangle \\ &= \langle (\mathbf{x}_2^T \mathbf{x}_2)^{-1} \mathbf{x}_2^T (\beta_2 \mathbf{x}_2 - \frac{\beta_2 \rho \sigma_2}{\sigma_1} \mathbf{x}_1 + \boldsymbol{\epsilon}) \rangle \\ &= \beta_2 - \beta_2 \rho^2\end{aligned}$$

Calculating the final residuals gives

$$\begin{aligned}\mathbf{r}_2 &= \mathbf{r}_1 - \hat{\mathbf{r}}_1 \\ &= \beta_2 \mathbf{x}_2 - \frac{\beta_2 \rho \sigma_2}{\sigma_1} \mathbf{x}_1 + \boldsymbol{\epsilon} - \hat{\beta}'_2 \mathbf{x}_2 \\ &= -\frac{\beta_2 \rho \sigma_2}{\sigma_1} \mathbf{x}_1 + \beta_2 \rho^2 \mathbf{x}_2 + \boldsymbol{\epsilon},\end{aligned}$$

and using the typical propagation of uncertainty formulae to find the variance of these residuals, we obtain

$$\begin{aligned}\sigma_{\mathbf{r}_2}^2 &= \frac{\beta_2^2 \rho^2 \sigma_2^2}{\sigma_1^2} \sigma_1^2 + \beta_2^2 \rho^4 \sigma_2^2 - 2 \frac{\beta_2^2 \rho^3 \sigma_2}{\sigma_1} \rho \sigma_1 \sigma_2 + \sigma_{\text{int}}^2 \\ &= \beta_2^2 \rho^2 \sigma_2^2 (1 - \rho^2) + \sigma_{\text{int}}^2.\end{aligned}$$

E.3 Step Function Correction Derivations

Again, we start with Equation E.1 to obtain the expected value of the best-fit slope from the linear portion of the fit.

$$\begin{aligned}\langle \hat{\alpha}' \rangle &= \langle \mathbf{x}_1^T \mathbf{x}_1 \rangle^{-1} \mathbf{x}_1^T \mathbf{Y} \\ &= \langle (\mathbf{x}_1^T \mathbf{x}_1)^{-1} \mathbf{x}_1^T \mathbf{x}_1 \alpha + (\mathbf{x}_1^T \mathbf{x}_1)^{-1} \mathbf{x}_1^T \text{sgn}(\mathbf{x}_2) \frac{\gamma}{2} + (\mathbf{x}_1^T \mathbf{x}_1)^{-1} \mathbf{x}_1^T \boldsymbol{\epsilon} \rangle \\ &= \alpha + \frac{\gamma}{2\sigma_1^2} \langle \mathbf{x}_1^T \text{sgn}(\mathbf{x}_2) \rangle \\ &= \alpha + \frac{\gamma \rho}{\sigma_1 \sqrt{2\pi}}\end{aligned}$$

The proof of the final step is as follows, where $p(x_1, x_2)$ is a bivariate Gaussian distribution with mean $(0, 0)$ and covariance matrix like that in Equation 5.3:

$$\begin{aligned}
 \langle \mathbf{x}_1 \text{sgn}(\mathbf{x}_2) \rangle &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x_1 \text{sgn}(x_2) p(x_1, x_2) dx_1 dx_2 \\
 &= \int_{-\infty}^0 \int_{-\infty}^{\infty} -x_1 p(x_1, x_2) dx_1 dx_2 + \int_0^{\infty} \int_{-\infty}^{\infty} x_1 p(x_1, x_2) dx_1 dx_2 \\
 &= 2 \int_0^{\infty} \int_{-\infty}^{\infty} x_1 p(x_1, x_2) dx_1 dx_2 \\
 &= \frac{1}{\pi \sigma_1 \sigma_2 \sqrt{1 - \rho^2}} \int_0^{\infty} \int_{-\infty}^{\infty} x_1 \exp \left[-\frac{1}{2(1 - \rho^2)} \left(\frac{x_1^2}{\sigma_1^2} + \frac{x_2^2}{\sigma_2^2} - \frac{2\rho x_1 x_2}{\sigma_1 \sigma_2} \right) \right] dx_1 dx_2 \\
 &= \sqrt{\frac{2}{\pi}} \rho \sigma_1
 \end{aligned} \tag{E.3}$$

The residuals that remain after correcting for the linear slope are

$$\begin{aligned}
 \mathbf{r}_{\alpha} &= \mathbf{Y} - \hat{\mathbf{Y}}_{\alpha} \\
 &= \alpha \mathbf{x}_1 + \frac{\gamma}{2} \text{sgn}(\mathbf{x}_2) + \boldsymbol{\epsilon} - \hat{\alpha}' \mathbf{x}_1 \\
 &= \frac{\gamma}{2} \text{sgn}(\mathbf{x}_2) - \frac{\gamma \rho}{\sigma_1 \sqrt{2\pi}} \mathbf{x}_1 + \boldsymbol{\epsilon}
 \end{aligned}$$

We can find what the step size γ would be when fit to these residuals by finding the value of $\hat{\gamma}'$ that minimizes $L = \left\| \mathbf{r}_{\alpha} - \frac{\hat{\gamma}'}{2} \text{sgn}(\mathbf{x}_2) \right\|^2$.

$$\begin{aligned}
 L &= \left\| \mathbf{r}_{\alpha} - \frac{\hat{\gamma}'}{2} \text{sgn}(\mathbf{x}_2) \right\|^2 \\
 &= \mathbf{r}_{\alpha}^2 - \hat{\gamma}' \mathbf{r}_{\alpha} \text{sgn}(\mathbf{x}_2) + \frac{\hat{\gamma}'^2}{4} \\
 \frac{\partial L}{\partial \hat{\gamma}'} &= -\mathbf{r}_{\alpha} \text{sgn}(\mathbf{x}_2) + \frac{\hat{\gamma}'}{2}
 \end{aligned}$$

Setting this derivative to zero, we find

$$\begin{aligned}
 \hat{\gamma}' &= 2 \mathbf{r}_{\alpha} \text{sgn}(\mathbf{x}_2) \\
 &= \gamma - \frac{2\gamma\rho}{\sigma_1 \sqrt{2\pi}} \mathbf{x}_1 \text{sgn}(\mathbf{x}_2) + 2 \boldsymbol{\epsilon} \text{sgn}(\mathbf{x}_2)
 \end{aligned}$$

The expectation value is

$$\begin{aligned}
 \langle \hat{\gamma}' \rangle &= \gamma - \frac{2\gamma\rho}{\sigma_1 \sqrt{2\pi}} \langle \mathbf{x}_1 \text{sgn}(\mathbf{x}_2) \rangle + 2 \langle \boldsymbol{\epsilon} \text{sgn}(\mathbf{x}_2) \rangle \\
 &= \gamma - \frac{2\gamma\rho^2}{\pi}
 \end{aligned}$$

where we used the result of Equation E.3 to evaluate $\langle \mathbf{x}_1 \text{sgn}(\mathbf{x}_2) \rangle$. Our final residuals after the two-step regression are then

$$\begin{aligned}\mathbf{r}_\beta &= \mathbf{r}_\alpha - \hat{\mathbf{r}}_\alpha \\ &= \frac{\gamma}{2} \text{sgn}(\mathbf{x}_2) - \frac{\gamma\rho}{\sigma_1\sqrt{2\pi}} \mathbf{x}_1 + \boldsymbol{\epsilon} - \frac{\gamma}{2} \text{sgn}(\mathbf{x}_2) + \frac{\gamma\rho^2}{\pi} \text{sgn}(\mathbf{x}_2) \\ &= -\frac{\gamma\rho}{\sigma_1\sqrt{2\pi}} \mathbf{x}_1 + \frac{\gamma\rho^2}{\pi} \text{sgn}(\mathbf{x}_2) + \boldsymbol{\epsilon}\end{aligned}$$

The variance of these residuals is

$$\begin{aligned}\sigma_{\mathbf{r}_\beta}^2 &= \frac{\gamma^2\rho^2}{2\pi\sigma_1^2} \sigma_1^2 + \frac{\gamma^2\rho^4}{\pi^2} - \frac{2\gamma^2\rho^3}{\sigma_1\sqrt{2\pi^3}} \langle \mathbf{x}_1 \text{sgn}(\mathbf{x}_2) \rangle + \sigma_{\text{int}}^2 \\ &= \frac{\gamma^2\rho^2}{2\pi} + \frac{\gamma^2\rho^4}{\pi^2} - \frac{2\gamma^2\rho^4}{\pi^2} + \sigma_{\text{int}}^2 \\ &= \frac{\gamma^2\rho^2}{2\pi} \left(1 - \frac{2\rho^2}{\pi}\right) + \sigma_{\text{int}}^2\end{aligned}$$