The Astrophysical Journal, 890:177 (11pp), 2020 February 20 https://doi.org/10.3847/1538-4357/ab6601

A New Method to Classify Type IIP/IIL Supernovae Based on Their Spectra

Shihao Kou, Xingzhuo Chen, and Xuewen Liu
College of Physics, Sichuan University, Chengdu, 610064, People's Republic Of China; liuxuew@scu.edu.cn, chenxingzhuo@tamu.edu

Received 2018 September 5; revised 2019 December 27; accepted 2019 December 28; published 2020 February 26

Abstract

Type IIP and Type IIL supernovae are defined on the basis of their light curves, but the spectral criteria for distinguishing these two types of supernovae (SNe) remain unclear. We propose a spectral classification method. First, we subtract the principal components of different wavelength bands in the spectra based on the functional principal components analysis method. Then, we use support vector machine and artificial neural network to classify these two types of SNe. The best F1_Score of our classifier is 0.871 for SNe IIL, and 0.974 for SNe IIP. We found that by only using the H_α line at 6150–6800 Å for classification, the F1_Score up to 0.961 for Type IIP, and 0.818 for Type IIL SNe can be obtained. These results indicate that the profile of the H_α spectral line is the key to distinguishing the two types of SNe.

Unified Astronomy Thesaurus concepts: Astronomy data analysis (1858); Supernovae (1668)

1. Introduction

Originating from massive stars ($8 \sim 20 M_\odot$), hydrogen-rich, core-collapse supernovae, also known as Type II Supernovae (SNe II), are one kind of supernova (SN) with a conspicuous hydrogen spectral line at 6335 Å (Gal-Yam 2017). Some SNe II are generally divided into four subtypes, IIP, IIL, IIb, and IIn, while some SNe with peculiar spectroscopic or light-curve features are divided into SNe II-pec, such as SN1987A (McCray 2017). Helium spectral lines at the wavelengths of 5876, 6678, and 7065 Å are characteristic lines observed in SNe IIb, while SNe IIn usually have a narrow H_α line at 6563 Å. Because of similar spectral features, SNe IIP/IIL are defined by their light curves.

The light curves of SNe are generally produced by four different mechanisms (Arcavi 2017): shock breakout; shock cooling and ejecta recombination; radioactive decay; and circumstellar material interaction. The main feature of SNe IIP are their plateau-like light curves, which quickly rise to the peak (∼15 days) after explosion and follow a plateau phase (∼90 days) that is powered from the cooling of ejecta. After the plateau phase, the luminosity declines linearly (>1.4 Mag/100 day) because of the nickel decay chain $^{56}\mathrm{Ni} \rightarrow\ ^{56}\mathrm{Co} \rightarrow\ ^{56}\mathrm{Fe}$. In contrast, the cooling phase of SNe IIL (typically less than 10 days) is much shorter than that of SNe IIP, and its luminosity linearly decreases as ∼0.3Mag/15 day after the peak (Gal-Yam 2017). Unlike the simple shape of SNe IIP/IIL, SNe IIb have versatile light curves, some of which even show a double-peak light curve. The first peak is originated from the fast cooling effect after the explosion, and the second one links to the high-energy photon escaping the relatively small-mass envelope (Richmond et al. 1994; Li et al. 2011; Arcavi 2017).

Many studies indicates that the progenitors of SNe IIP are red super giants (RSG) with a mass range of 8.5–16.5 $M_\odot$ (Smartt 2015), whereas SNe IIb are believed originated from yellow super giants (YSG; Anderson et al. 2012). Some studies suggest SNe IIn progenitors are luminous blue variables (Schlegel 1990). The progenitor of SNe IIL is still on debate. Only one observational evidence of SN2009kr suggests that an RSG or YSG could be its progenitor (Fraser et al. 2010; Elias-Rosa et al. 2011).

Previous studies of SNe IIP/IIL showed that the absorption–emission ratio of a H_α P-Cygni profile in SNe IIL is relatively smaller than that of SNe IIP after the luminosity peak, which is possibly because of their different envelope mass and density (Patat et al. 1994; Gutiérrez et al. 2014). Moreover, a study of H_α and O(I)7774 equivalent width (EW) indicates that SNe IIP have a smaller spectral ratio $\mathrm{EW}_{\mathrm{OI}7774}/\mathrm{EW}_{\mathrm{H}_\alpha}$ (Faran et al. 2014). In addition, the absolute peak luminosities of SNe IIL/IIP are -17.44 ± 0.22 and -15.66 ± 0.16, respectively (Li et al. 2011). However, unlike the classification of SNe I, no adequate spectroscopic criteria are presented to distinguish SNe IIP from SNe IIL (Sun & Gal-Yam 2017).

The current classification scheme of SNe, including numerous types and subtypes according to their spectra and photometric properties, is universally recognized, even though there are controversies regarding the definitions and specific progenitor stars (Gal-Yam 2017). In the coming era, existing and upcoming wide-field optical surveys and facilities, e.g., the Large Synoptic Survey Telescope (Ivezic et al. 2019), the Dark Energy Survey (Abbott et al. 2016), the All-Sky Automated Survey for Supernovae (Kochanek et al. 2017), the Panoramic Survey Telescope & Rapid Response System (Pan-STARRS; Chambers & Pan-STARRS Team 2017), and the Zwicky Transient Facilities (Wozniak et al. 2014) will quickly expand the SNe number from a few thousand to millions of events. It is becoming possible to find the direct observational evidence to clarify the controversies in classification by analyzing the large amount of forthcoming data, which needs new effective automatic classification methods. Since the launch of the Supernova Photometric Classification Challenge (SPCC) in 2010 (Kessler et al. 2010), several machine-learning algorithms were proposed for SN classification by using the features of light curves extracted from the SPCC data set, including either parametric fits to the light curves, template fitting, or model-independent wavelet decomposition of the data. Lochner et al. (2016) found that the boost decision trees performed most effectively. Besides photometric features, spectral information is also applied to SN classification. Sasdelli et al. (2016) combined the principal component analysis method with K-means algorithms to use spectral features to classify subtypes of SNe Ia. Muthukrishna et al. (2019) has constructed a 4-layered neural network to automatically estimate the

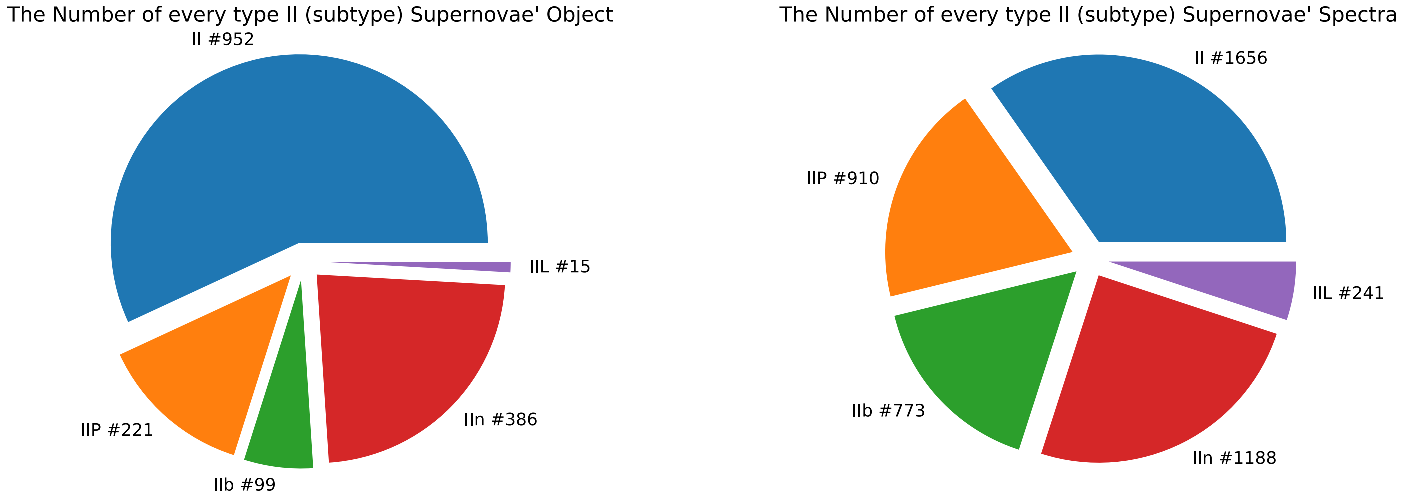

Figure 1. Number of different SNe and their spectra in our analysis.

Table 1
Extra Spectra of Six SNe IIL from Other Literature

Supernovae Name	Number of Spectra	Type in WISeREP	Reference
SN1980K	6	II	(Eldridge & Tout 2004)
SN1979C	4	II	(Ray et al. 2001)
SN2013hj	5	II	(Bose et al. 2016)
SN2013by	18	II	(Valenti et al. 2015)
SN2013ej	112	II	(Yuan et al. 2016)
SN2009kr	2	IIn	(Anderson et al. 2012)

redshift and classify the type of SNe based on spectra, which perform quite well on general-type classification (SNe Ia, Ib and Ic, II) but failed in the classification on the subtypes.

Based on spectra of SNe IIP/IIL for the first time, in this paper, we apply the Functional Principal Component Analysis (FPCA) to extract a series of principle components from SNe IIP/IIL spectra, then feed into the machine-learning algorithm such as the support vector machine (SVM) and artificial neural network (ANN) to classify these two type of SNe. In Section 2, we will introduce the data source and the algorithms (FPCA, SVM, and ANN) used in the paper. In Section 3, we will discuss the performance of our classifiers. The prospective and the summary will be presented in Section 4. All the codes used in the analysis are uploaded onto https://github.com/GeronimoChen/IIL-IIP-SNe.

2. Methods

2.1. Data Overview

The data set includes all spectra of SNe II downloaded from the Weizmann Interactive Supernova Data Repository (WISeREP), which served as an archive of SN spectra and photometry (Yaron & Gal-Yam 2012). To expand the data set, we use the spectra of another six SNe IIL from the literature as shown in Table 1. The number of different Type of SNe and their spectra are shown in Figure 1.

We discuss in detail the preprocessing of the spectra in Appendix A, which entails removing redshift, Savitz–Golay filtering, and normalizing the spectral flux (Savitzky & Golay 1964). The resulting spectra are divided into nine small wavelength windows as shown in Figure 2. The wavelength windows include the major spectral lines in SN identification as shown in Table 2. (Marcaide et al. 2002; Maguire et al. 2010; Smith et al. 2010; Taddia et al. 2016; Anderson et al. 2018; de Jaeger et al. 2018; Gutiérrez et al. 2018; Singh et al. 2018). We carefully select the H_α and H_β windows to include all possible absorption and emission lines, because hydrogen lines are the key lines in defining SNe II. Except for the wave window of "Gap," all other wave windows are denoted by the dominating spectral lines, such as FeMg, H_β, FeOMgSi, S, Na, H_α, NaMg, and Ca. As the "Gap" window may be contaminated by the telluric H_2O spectral line at 7165 Å. However, some SNe II may embody calcium spectral lines (Ca(II)7291,7323) in this region (Maguire et al. 2010; Faran et al. 2014; de Jaeger et al. 2018). But we exclude the wave window of 6800–7000 Å and 7400–7700 Å due to the telluric $O_2$6867 and $O_2$7620 contamination. The numbers of the available spectra in each of the wave windows is listed in Table 2. Every individual spectra in different wave windows are normalized with its average value being equal to zero.

2.2. FPCA Algorithm

Now, we can use FPCA algorithm to extract a series of principal components from every single spectrum. The principal components are a set of orthogonal functions that could recover the spectra via their linear combination (Jones & Rice 1992; Hall et al. 2006; Peng & Paul 2007). A specific spectrum $X_n(\lambda)$ can be represented as

$$X_n(\lambda) = \mu(\lambda) + \sum_{m=1}^{\infty} \beta_{m,n}\phi_m(\lambda), \tag{1}$$

where $\phi_m(\lambda)$ is the m-th basis functions, β_m is the m-th functional principal components scores (Scores), and $\mu(\lambda)$ is the average of the spectra. The bilinear orthogonal form of the basis functions is the integral of the product of functions over the interval of the wave window $[\lambda_{\min}, \lambda_{\max}]$:

$$\int_{\lambda_{\min}}^{\lambda_{\max}} \phi_j(\lambda)\phi_k(\lambda)d\lambda = \delta_{jk}. \tag{2}$$

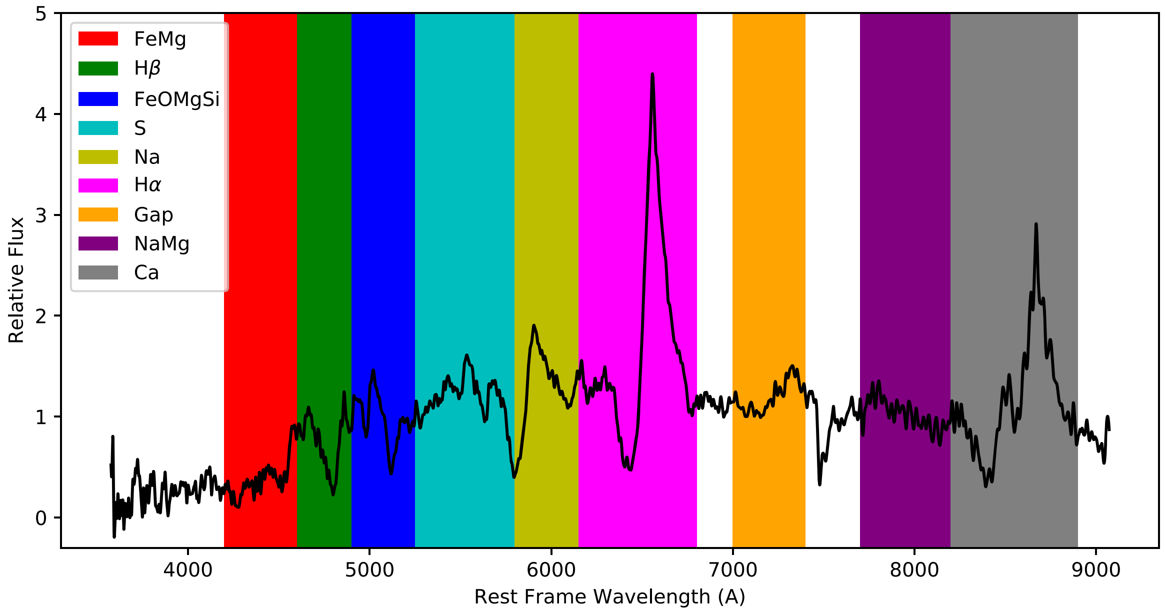

Figure 2. Nine wave windows shown in the spectrum of SN2018bl (Type II, WISeREP Id: 45232).

Table 2
Wave-window Name, Wavelength Ranges, Major Spectral Lines, and the Number of the Available SNe Spectra in Each Window

Name	Wavelength Range (Å)	Elements and Spectral Lines	Available SNe IIL/IIP Spectra
FeMg	4200–4600	H_γ4340, H_δ4102, Ba(II)4554	170/667
H_β	4600–4900	H_β4861	181/717
FeOMgSi	4900–5250	Fe(II)4924,5018,5108,5169. O(III)4959,5007.	176/733
S	5250–5800	S(II)5454,5433. O(V)5597.	154/700
Na	5800–6150	Na(I)5876,5896. He(I)5876. Ba(II)6142.	168/722
H_α	6150–6800	H_α 6563. He(I)6678, O(I)6300,6364. N(II)6548. Sc(II)6247	177/735
Gap	7000–7400	Fe(II)7155. He(I)7065. O(II)7319,7330. Ca(II)7291,7323	159/672
NaMg	7700–8200	Na(I)8183,8195	111/491
Ca	8200–8900	Ca(II)8498,8542,8662. O(I)7774.	113/440
All 9	4200–8900	All of the abovementioned lines	73/337

The convolution form of the m-th Score is

$$\beta_{m,n} = \int_{\lambda_{\min}}^{\lambda_{\max}} \phi_m(\lambda)(X_n(\lambda) - \mu(\lambda)d\lambda, \qquad (3)$$

which is sorted by its variance among all spectra:

$$\forall\, m\colon Var_n(\beta_{m,n}) > Var_n(\beta_{m+1,n}). \qquad (4)$$

The basis function is selected by the following equation:

$$\begin{aligned}\phi_m(\lambda) &= \underset{\phi_m(\lambda)}{\arg\max}\, \{Var_n(\beta_{m,n})\}\\ &= \left\{\phi'_m(x) | Var_n\left(\int_{\lambda_{\min}}^{\lambda_{\max}} (X(\lambda) - \mu(\lambda)\phi'_n(\lambda)d\lambda\right)\right\},\end{aligned} \qquad (5)$$

which ensures that the most information is extracted from the spectrum $X(\lambda)$.

In this paper, we use the R language package `fpca` to get the best set of basis functions (Peng & Paul 2007). Because of the limitation of our computational resource, we extract only 30 basis functions from each wave window of every single spectra. Then, we use these basis functions to reconstruct the spectra of SN1990E (SNe IIP, WISeREP ID: 1194) and SN2014G (SNe IIL, WISeREP ID: 30101), the shape of which match their normalized spectra quite well, as shown in Figure 3. The variance distribution and the fidelity of the basis functions are discussed in Appendix B.

2.3. Classification

Then, a series of FPCA Scores extracted from spectra of SNe II are ready for classification. In Figure 4, we show the correlations of the first and second Scores in the H_α wave window of different type of SNe IIP/IIL/IIb/IIn. Although different patterns appear, it is still impossible to tell the boundaries between different SNe by using only two Scores. To use all of the Scores information, we apply SVM (Chen et al. 2005; Cristianini & Ricci 2008; Chang & Lin 2011) and ANN (LeCun et al. 2015) for the classification of SNe IIP/IIL.

2.3.1. Support Vector Machine

SVM is a popular machine-learning method for classification that aims to find the optimal hyperplane to separate two sets of dots with largest margin distance, as shown in Figure 5. This method was first proposed by Boser et al. (1992).

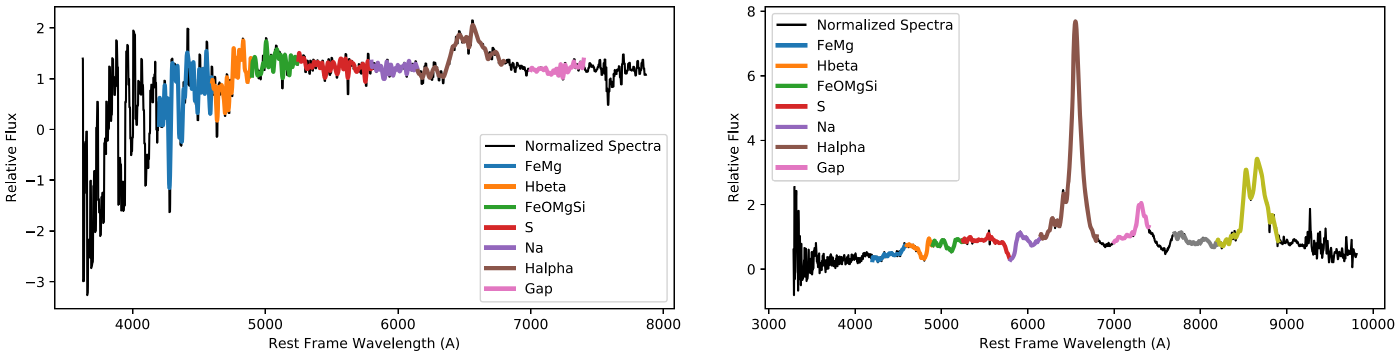

Figure 3. Normalized spectra and the 30-order FPCA reconstructed results of SN1990E (SNe IIP, WISeREP ID: 1194) and SN2014G (SNe IIL, WISeREP ID: 30101).

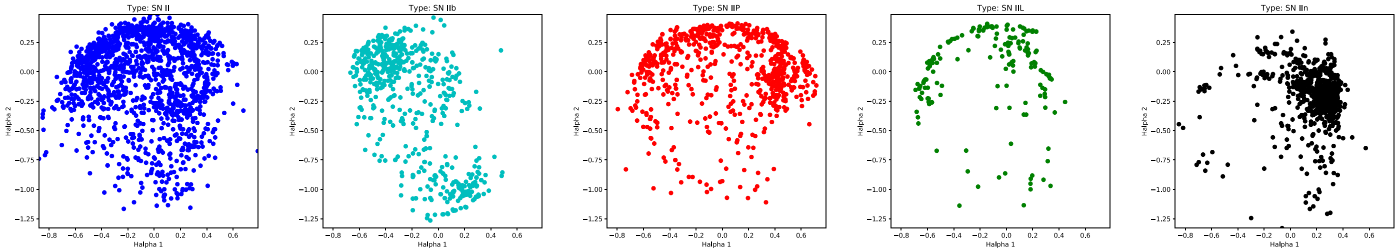

Figure 4. First and second FPCA Scores in the H_α wave window for different types of SNe.

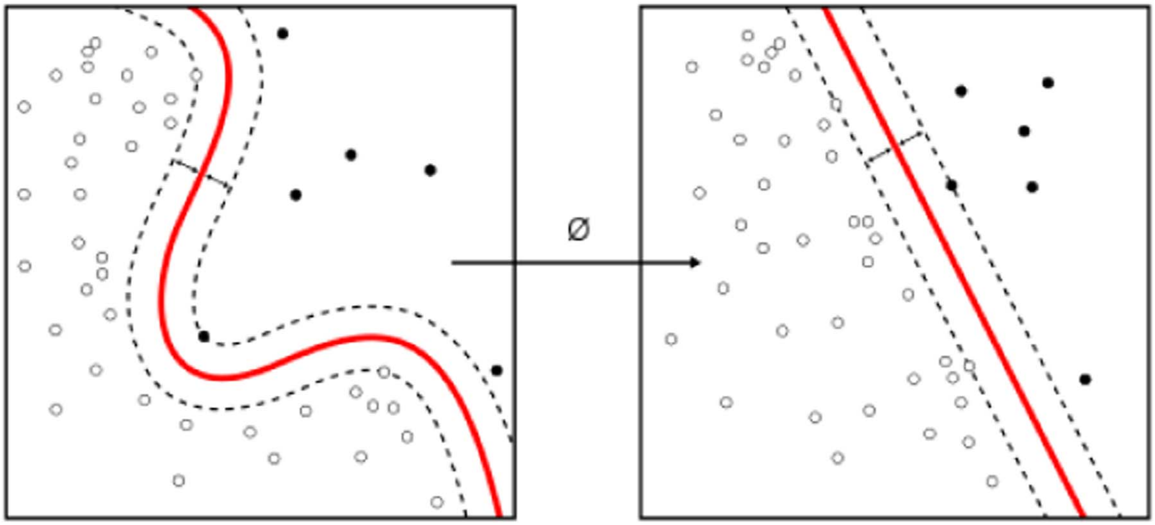

Figure 5. Illustration of how Kernel SVM works. Alisneaky: https://commons.wikimedia.org/w/index.php?curid=47868867.

Consider two kinds of nodes in a hyperspace; the target of SVM is to find a hyperplane that could separate the data, which is written in the form

$$0 = X \cdot \omega + b, \tag{6}$$

where X is the coordination on the hyperplane, ω is the weight of each dimension, and b is the bias. The dots closest to the hyperplane are denoted as support vectors; the distance between the support vector and the hyperplane is

$$\mathrm{Min}_i\left(\frac{\omega \cdot X_i + b}{||\omega||}\right), \tag{7}$$

where X_i is the coordination of the nodes. The minimal is reached if the X_i is the support vector. To construct an optimized hyperplane, the distances between the support vectors and the hyperplane should be as large as possible; the optimization target is

$$\mathrm{Max}_{\omega,b}\left[\mathrm{Min}_i\left(\frac{\omega \cdot X_i + b}{||\omega||}\right)\right]. \tag{8}$$

If the distance between the support vector and the hyperplane is denoted to 1, then the optimization target is

$$\mathrm{Max}\left(\frac{1}{||\omega||}\right), \text{ subject to: } y_i(\omega \cdot X_i + b > 1), \tag{9}$$

where $y_i = \pm 1$ is the sign of the nodes, which marks their tags (IIP or IIL). Because the nodes may not been intrinsically

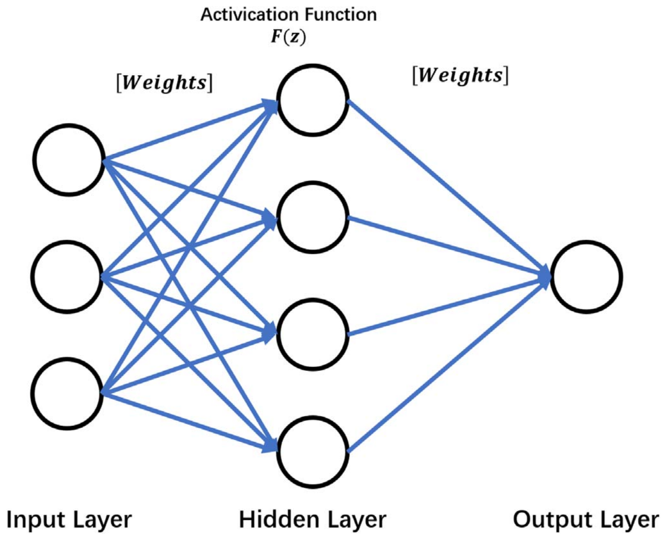

Figure 6. Structure of the artificial neural network.

separated, an slack variable is introduced to gauge the misclassification,

$$1 - \xi_i = y_i(\omega \cdot X_i + b),\ \xi_i >= 0. \tag{10}$$

The cost function for the optimization is written as

$$L(\omega, b, \xi) = \frac{1}{2}||\omega||^2 + C\sum_i \xi_i, \tag{11}$$

where C is a penalty parameter which defined before. In this paper, we use `python.sklearn.svm` for SVM classification, radial basis function is used for nonlinear classification, and the kernel function and the distance are written as

$$K(x, x') = \exp\left(-\gamma\frac{||x - x'||^2}{\sigma^2}\right) \tag{12}$$

and

$$K(\omega, X) + b = \exp\left(-\gamma\frac{||x - x'||^2}{\sigma^2}\right) + b. \tag{13}$$

2.4. Artificial Neural Network

A simple ANN is composed of three parts: the input layer, hidden layers, and the output layer, as shown in Figure 6 (LeCun et al. 2015). Nodes in adjacent layers are linked with different weights. In the ANN, data are propagated from the input layer to the output layer with the equations

$$z_i = W_i \cdot y_i + b_i \tag{14}$$

and

$$y_{i+1} = f_{i+1}(z_i), \tag{15}$$

where W_i is the weight matrix, b_i is the bias, and $f_{i+1}(z_i)$ is the activation function within the i-th layer. The output layer will give the probabilistic classification upon a certain input data. In this paper, we use the rectified linear unit (ReLU) function $f(z) = \max(0,\ z)$ as the hidden layer's activation function, which performs well in many supervised deep-learning paradigms. Because it is a dichotomy problem, we choose sigmoid function ($f(z) = 1/(1 + \exp(-x))$) rather than ReLU as the output layer's activation function, as it constrains the output in the region (0, 1) and allows us to predict the positive and negative. Unlike SVM, we set 1 and 0 for SNe IIP and IIL separately for the application of ANN.

The weight and bias parameters are trained using the back propagation procedure. In this paper, we choose binary cross entropy as the loss function, which is written as

$$L(y_r, y_p) = -\sum_i [y_r \ln(y_p) + (1 - y_r)\ln(1 - y_p)], \tag{16}$$

where y_r is the true probability of 0 or 1, and y_p is the probability given by the neural network (Shannon 2001). The training target of ANN is to modify the weights and bias to minimize the loss function, which make the prediction of "positive" data close to 1 and "negative" data close to 0. In the basic stochastic gradient descent algorithm, the update of weights and bias are following the equation

$$\theta_{\mathrm{new}} = \theta_{\mathrm{old}} - \alpha g_\theta, \tag{17}$$

where α is the learning rate, and θ is the weights or bias. In each training epoch, the update of weights and biases are uploaded upon the gradient of the loss functions: $g_\omega = \frac{\partial L}{\partial \omega}$ and $g_b = \frac{\partial L}{\partial b}$. In this paper, we use the Adam algorithm (Kingma & Ba 2014) for optimization, in which the weights and biases are updated with

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_\theta, \tag{18}$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_\theta, \tag{19}$$

$$\hat{m}_t = m_t/(1 + \beta_1^t), \tag{20}$$

$$\hat{v}_t = v_t/(1 + \beta_2^t), \tag{21}$$

$$\theta_{\mathrm{new}} = \theta_{\mathrm{old}} - \alpha\hat{m}_t/(\sqrt{\hat{v}_t} + \epsilon), \tag{22}$$

where m_t and v_t are the t-th epoch first and second momentum, $\epsilon = 10^{-8}$ is set to avoid overflow. In this paper, we choose the decay parameters $\beta_1 = 0.9$, $\beta_2 = 0.999$ and the learning rate $\alpha = 0.001$, following the original paper of Adam (Kingma & Ba 2014). To avoid the over fitting problem, we add $L1$ and $L2$ penalties into the loss function which is written as

$$\begin{aligned} L(y_r, y_p) = &-\sum_i [y_r \ln(y_p) + (1 - y_r)\ln(1 - y_p)] \\ &+ l_1(\sum|\theta|) + l_2(\sum|\theta|^2)^{0.5}. \end{aligned} \tag{23}$$

2.5. Performance Evaluations

Because the number of the available spectra of SNe IIL/IIP are small, as is shown in Table 2 and Figure 1, we adopt a cross validation method to evaluate the performance. The whole data set is divided into a training set (80%) and a testing set (20%). We use the training set to train the classifier. We evaluate the performances of our classifiers using the F1_Score, which is defined as

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}, \tag{24}$$

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}}, \tag{25}$$

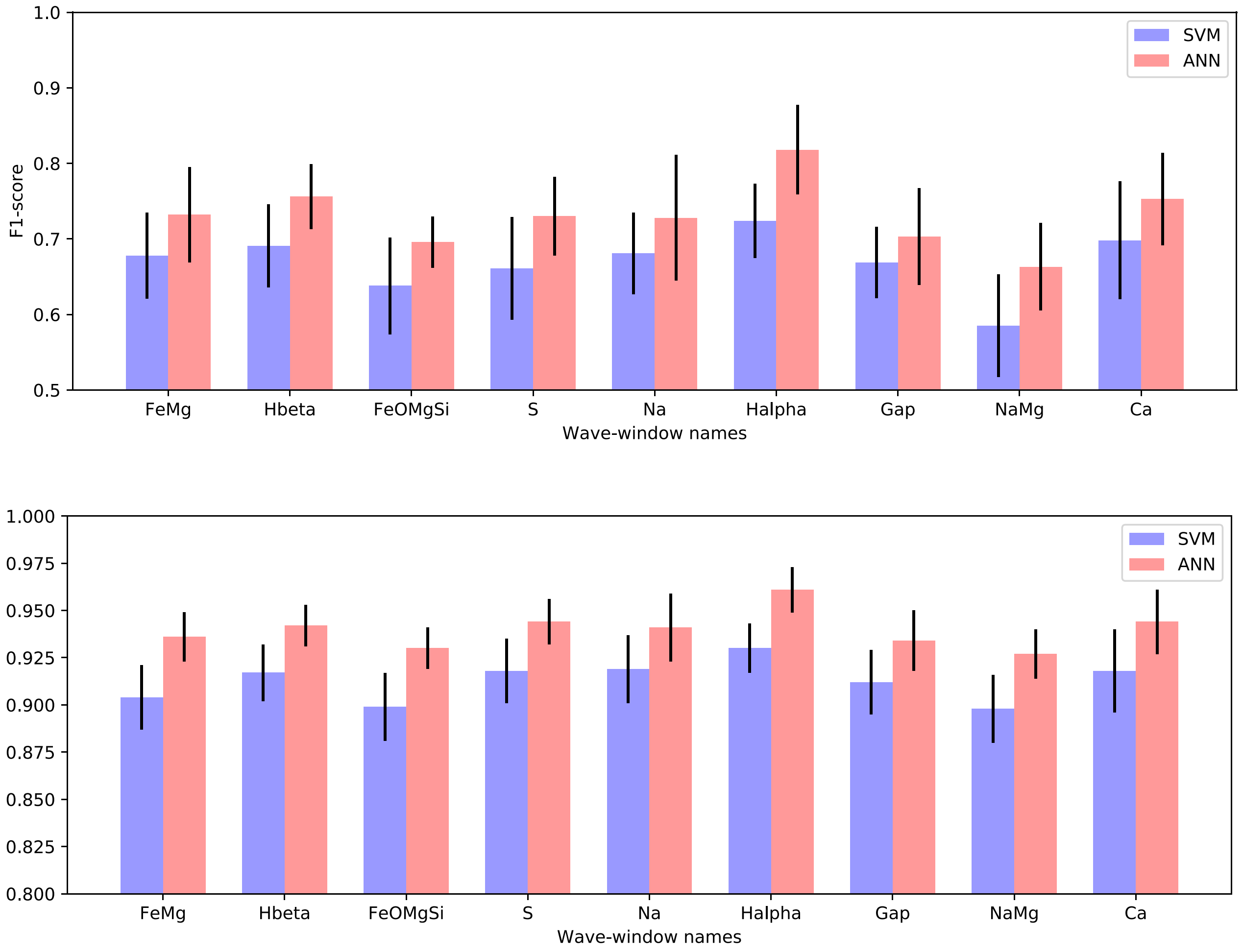

Figure 7. Upper panel: the F1_Score of SVM and ANN for SNe IIL. Lower panel: the F1_Score of SVM and ANN for SNe IIP. Red bars are the F1_Score from ANN and blue bars are the F1_Score from SVM. The standard error are plotted with black lines on top of the bar.

$$\text{F1\_Score} = \frac{2}{\frac{1}{\text{Precision}} + \frac{1}{\text{Recall}}}, \tag{26}$$

where True Positive is the number of "positive" spectra retrieved by the classifier; True Negative is the number of "negative" spectra that did not retrieved by the classifier; False Positive is the number of "negative" spectra wrongly retrieved by the classifier; and False Negative is the number of "positive" spectra did not retrieved by the classifier (Olson & Delen 2008).

The training and testing processes will be repeated 200 times for SVM algorithm and 20 times for ANN algorithm, and the data set is separated randomly in each iteration. When the training and testing processes are finished, we calculated the average F1_Score to evaluate the classifier. In addition, the standard errors of F1_Score are also calculated to evaluate the stability of our classifiers.

3. Results and Discussion

First, we only use one wave-window information for the classification including 30 FPCA Scores and two normalization factors. We use `keras` to construct a three-layered ANN, and apply Adam algorithm as the optimizer. We set the number of nodes in the middle layer to be 40, 100, 400, 600, 800, 1000, 2000, 4000, 6000, and 8000, respectively. Initially, we tried the model with 40, 400, and 4000 hidden nodes, then chose the number of nodes around the best-performance model. When the best number of middle layer nodes that produces the most optimal F1_Score is determined, we then add the $L1$ and $L2$ penalty into the models. The parameters are shown in Table 3. In the SVM, the parameter γ is automatically generated by the program. We set the penalty parameter C to be 300, 1000, 3000, 10,000, 30,000, 100,000, 300,000, 1,000,000, and 3,000,000, respectively. Initially, we calculated the F1_Score when C is 3000, 30,000, and 300,000, then choose the C around the best-performed model to further increase the F1_Score. The highest F1_Scores and their standard errors are plotted in Figure 7. The parameters of the best-performed

Table 3
ANN Parameters for One-window Classification

Items	Keys
Input layer	32
Hidden layer	See Table 5
Activation function	ReLU
Output layer	1
Activation function	Sigmoid
Optimization method	Adam
Regularization	None

Table 4
SVM Performance in a Single Wave Window

Wave Window	C	IIL F1_Score	IIP F1_Score	IIL F1_Score 1σ	IIP F1_Score 1σ
FeMg	3000	0.678	0.904	0.057	0.017
H_β	3,000,000	0.691	0.917	0.055	0.015
FeOMgSi	100,000	0.638	0.899	0.064	0.018
S	3,000,000	0.661	0.918	0.068	0.017
Na	300,000	0.681	0.919	0.054	0.018
H_α	300,000	0.724	0.930	0.049	0.013
Gap	10,000	0.669	0.912	0.047	0.017
NaMg	300,000	0.585	0.898	0.068	0.018
Ca	3000	0.698	0.918	0.078	0.022

Table 5
ANN Performance in a Single Wave Window

Wave Window	Middle Layer Units	IIL F1_Score	IIP F1_Score	IIL F1_Score 1σ	IIP F1_Score 1σ
FeMg	1000	0.732	0.936	0.063	0.013
H_β	400	0.756	0.942	0.043	0.011
FeOMgSi	2000	0.696	0.930	0.034	0.011
S	4000	0.730	0.944	0.052	0.012
Na	400	0.728	0.941	0.083	0.018
H_α	800	0.818	0.961	0.059	0.016
Gap	4000	0.703	0.934	0.064	0.013
NaMg	2000	0.663	0.927	0.058	0.012
Ca	4000	0.753	0.944	0.061	0.017

Table 6
ANN Performance in the H_α Wave Window with Different $L1$ and $L2$ Penalties

$L1$ Penalty	$L2$ Penalty	IIL F1_Score	IIP F1_Score	IIL F1_Score 1σ	IIP F1_Score 1σ
0	**0**	**0.818**	**0.961**	**0.059**	**0.016**
0.1	0	0.500	0.920	0.092	0.016
0.01	0	0.696	0.930	0.034	0.011
0.001	0	0.787	0.955	0.038	0.009
0	0.1	0.687	0.939	0.050	0.013
0	0.01	0.818	0.961	0.059	0.016
0	0.001	0.779	0.950	0.041	0.010
0.1	0.1	0.745	0.941	0.069	0.015
0.01	0.01	0.766	0.949	0.055	0.011
0.001	0.001	0.793	0.953	0.066	0.015

Note. The bold parameters obtain the best performance.

model are shown in Tables 4 and 5. We notice that the H_α wave window has the highest F1_Score in all of the wave windows, which reaches 0.961(IIP) and 0.818 (IIL) when using ANN, while reaching 0.930 (IIP) and 0.724 (IIL) when using SVM. Moreover, ANN performs better than SVM in every wave window. It is also found that the F1_Score of SNe IIP is higher than that of SNe IIL by $0.1 \sim 0.2$, which is probably due to the data size of SNe IIP is larger than SNe IIL.

Because the ANN performs betters in the H_α wave window than in other wave windows, we consider adding the penalty to further increase the F1_Scores in this H_α window. We try different combinations of $L1$ and $L2$: (0.1, 0), (0, 0.1), (0.01, 0), (0, 0.01), (0.001, 0), (0, 0.001), (0.1, 0.1), (0.01, 0.01), and (0.001, 0.001), but none of them performs better than the model without regularization as shown in Table 6.

Now, to maximize the performance, we want to use the overall data with 288 dimensions: 30 FPCA Scores and two normalization factors in each wave window. We applied the same strategies to increase the model performance by modifying the hyperparameters (for example, C in SVM, $L1$ penalty, $L2$ penalty, and number of middle layer nodes in ANN). In Table 7, we show the F1_Score of the best-performance SVM and ANN with different parameters. In contrast to the trials in the H_α wave window, adding $L2$ penalty in all of the nine wave windows increases the performance of the ANN significantly. We obtain the best performance to our knowledge with the F1_Score of 0.871 (IIL) and 0.974 (IIP).

Whether more FPCA Scores in the ANN will increase the accuracy remains a question. We calculate F1_Scores for different number of FPCA Scores with 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 14, 17, 20, and 25 as shown in Table 8. We adopt 800 nodes in the middle layer for the H_α window, 2000 nodes for the nine wave windows, and $L2 = 0.01$ for penalty. As shown in Figure 8, the F1_Score increases quickly when the dimension of FPCA Scores is less than 10 for H_α window, while it does not change too much for the overall nine wave

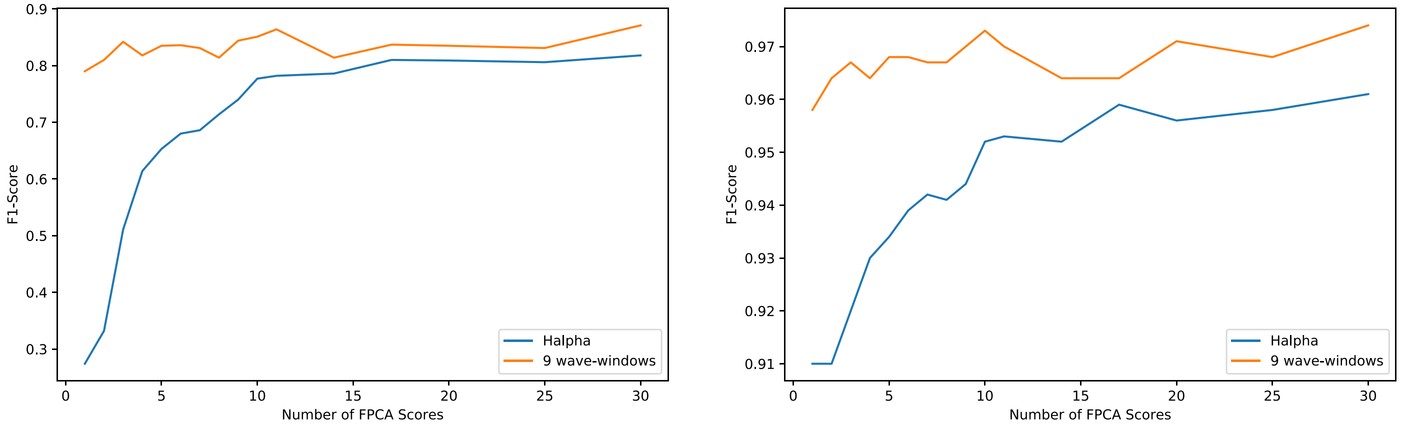

Figure 8. Relation between the F1_Score and the number of FPCA Scores. Left panel: SNe IIL. Right panel: SNe IIP.

Table 7
The Performance of SVM and ANN in the H_α and All Nine Wave Windows

Classifier	Wave Window	C (SVM)	Middle Layer	$L1$ (ANN)	$L2$ (ANN)	IIL F1	IIP F1	IIL F1 1σ	IIP F1 1σ
SVM	H_α	300,000	/	/	/	0.724	0.930	0.049	0.013
SVM	9	3000,000	/	/	/	0.810	0.962	0.073	0.015
ANN	H_α	/	**800**	0	0	**0.818**	**0.961**	0.059	0.012
ANN	9	/	2000	0	0	0.836	0.968	0.090	0.015
ANN	9	/	**2000**	0	0.01	**0.871**	**0.974**	0.066	0.013

Note. The bold parameters obtain the best performance.

Table 8
Performance in Different Dimension of FPCA

Basis Functions	IIL F1 H_α	IIP F1 H_α	IIL F1 Nine Wave Windows	IIP F1 Nine Wave Windows
30	0.818	0.961	0.871	0.974
25	0.806	0.958	0.831	0.968
20	0.809	0.956	0.835	0.971
17	0.810	0.959	0.837	0.964
14	0.786	0.952	0.814	0.964
11	0.782	0.953	0.864	0.970
10	0.777	0.954	0.851	0.973
9	0.740	0.944	0.844	0.970
8	0.714	0.941	0.814	0.967
7	0.686	0.942	0.831	0.967
6	0.680	0.939	0.836	0.968
5	0.653	0.934	0.835	0.968
4	0.614	0.930	0.818	0.964
3	0.511	0.920	0.842	0.967
2	0.332	0.910	0.810	0.964
1	0.274	0.910	0.790	0.958

Note. For the H_α wave window, the input dimension is the number of basis function plus two normalization factors. For the nine wave-window situation, the input dimension is the number of basis function multiplied by nine and plus two normalization factors for each nine wave windows. Columns from left to right: the number of basis functions in each wave window; the F1_Score of Type IIL when using H_α for classification; the F1_score of Type IIP when using H_α for classification; the F1_Score of Type IIL when using nine wave windows for classification; the F1_Score of Type IIP when using nine wave windows for classification.

windows. Therefore, 30 FPCA Scores is enough for our analysis.

4. Summary

We decompose SNe IIP/IIL spectra into different basis functions via FPCA algorithm. Based on the FPCA scores, we have trained SVM and ANN to classify SNe IIP and SNe IIL, two types of supernovae that are believed to have no adequate spectroscopic discrepancies in previous research. The three-layered ANN has a better performance for the classification. The spectra are divided into nine small-size wave windows for FPCA analysis; the best F1_Score we got is 0.871 for SNe IIL, and 0.974 for SNe IIP. It is the first quantitative classification of SNe IIP/IIL through spectral data. We show that compared with other eight wave windows, a better score is obtained in the narrow wave window at 6150–6800 Å, which covers the P-Cygni profile of H_α line. Thus we suggest that FPCA analysis directly on spectral line of a specific element could be a new approach in SNe classification.

In a future study, we will investigate the relationship between the principal components in spectra and the explosion profile of SNe, and expand the classification method to include other Type of SNe.

X.C. would like to thank Prof. Avishay Gal-Yam (Weizmann Institute of Science) and Prof. Lifan Wang (Purple Mountain Observatory) for supportive discussions. We thank Lei Hu (Purple Mountain Observatory) and Tianrui Sun (Purple Mountain Observatory) for support on calculations. We thank Weizmann Interactive Supernova data REPository (WISeREP) https://wiserep.weizmann.ac.il/ and Transient Name Server (TNS) https://wis-tns.weizmann.ac.il/ for the data.

Software: python,keras,scikit-learn,R-fpca (Peng & Paul 2007).

Appendix A
Preprocessing

We first remove the redshift of the raw spectrum by using the equation $\lambda_{\rm RF} = \lambda_{\rm obs}/(1+z)$ to transform the wavelength $\lambda_{\rm obs}$ in the observer frame to the rest frame of $\lambda_{\rm RF}$. Then the spectra in the rest frame are normalized by dividing the average flux, as the absolute flux of a spectrum in each pixel strongly depends on the telescope and the apparent magnitude. Finally, we use a Savitzky–Golay filter that has been widely used to reduce the noise in SNe spectra (Sun & Gal-Yam 2017). Tucker et al. (2019) smoothed the spectra of SNe Ia by using a two-order Savitzky–Golay filter with a smoothing width of 3000 km s^{-1}, which is broader than the noise. By considering the narrow-line feature in SNe II, we adopt a smaller width of 1000 km s^{-1} for our two-order Savitzky–Golay filter. In the next step, each spectrum in the selected wave window is subtracted with its average flux, and divided by the standard deviation. Such a process removes the blackbody component of the spectra. Finally, we resample the spectra to the resolution of 1 Å using linear interpolation method. The whole process is illustrated in Figure 9.

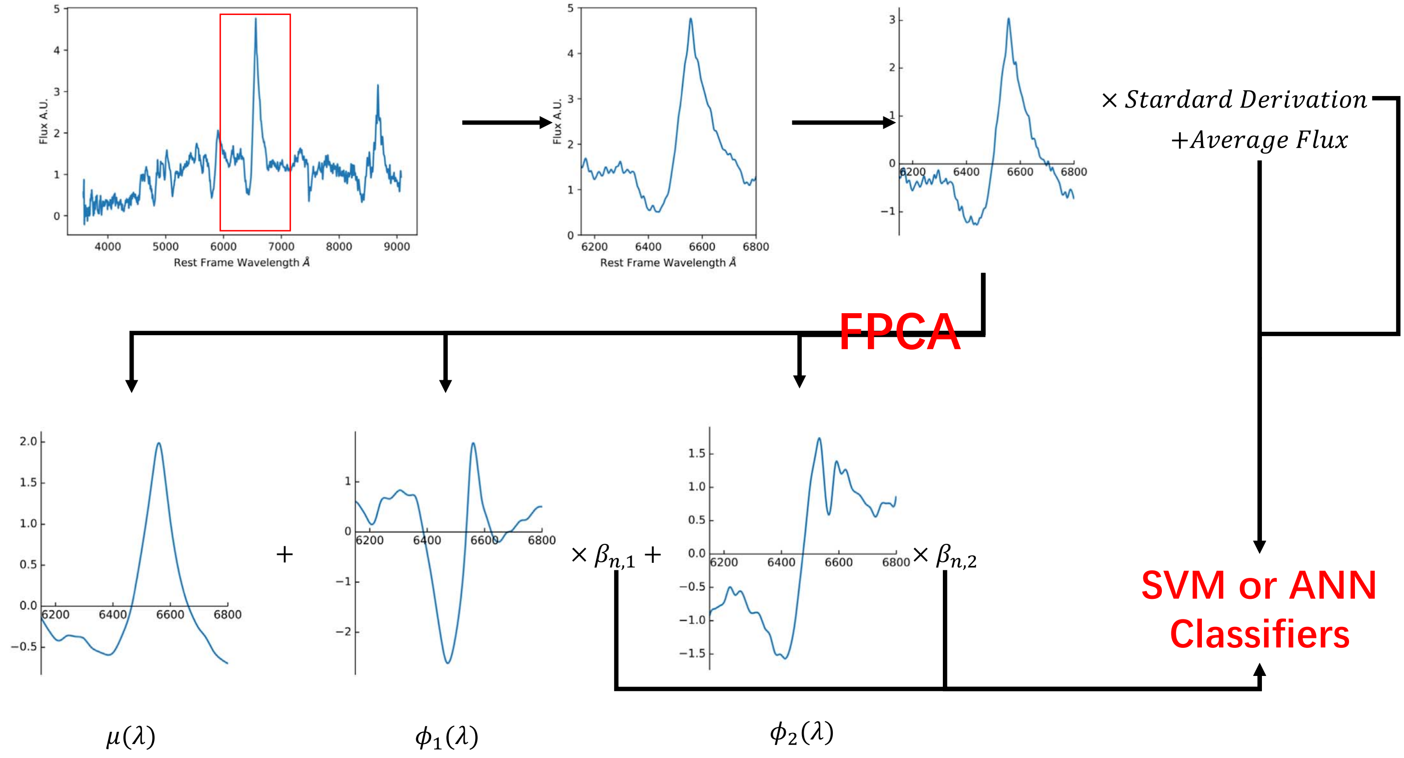

Figure 9. Illustrative flow chart of the data-processing steps.

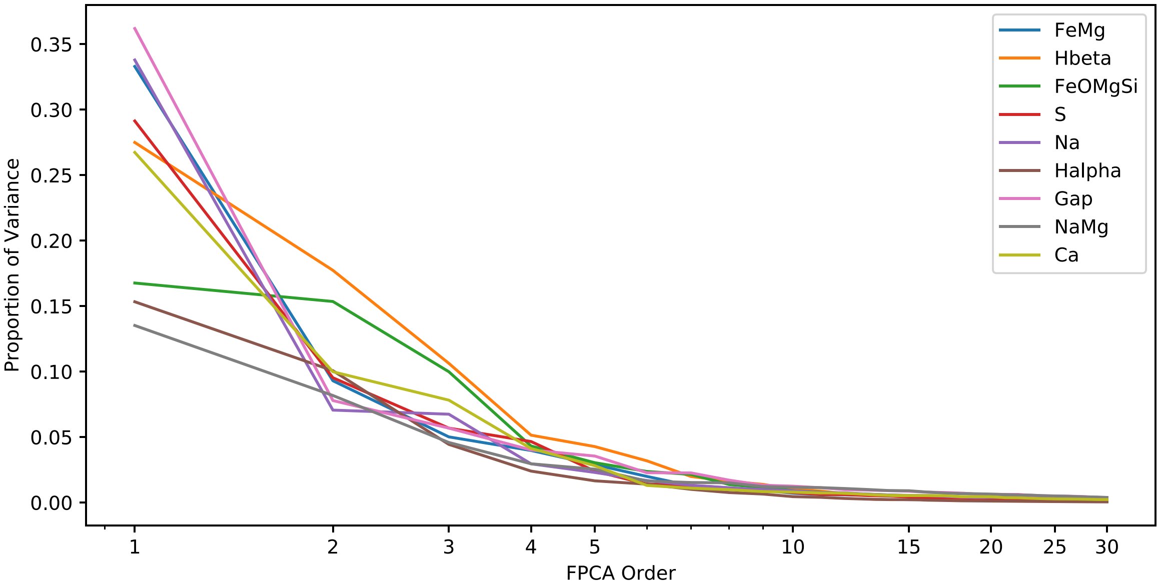

Figure 10. Proportion of variance of each FPCA Score in different wave windows.

Appendix B
FPCA Fidelity Discussion

We can use the function principal component, average flux, and standard derivative to reconstruct the spectra by the equation

$$F_n(\lambda) = X_n(\lambda) \times S_n + A_n \approx \left(\mu(\lambda) + \sum_{m=1}^{m_{\max}} \beta_{m,n}\phi_m(\lambda)\right) \times S_n + A_n, \quad (27)$$

where A_n is the average flux, and S_n is the standard derivation. We choose all SNe IIP/IIL spectra for FPCA calculation in the H_α, Ca, S, Na, NaMg, and “Gap” wave windows, but only 20% spectra in the H_β, FeOMgSi, and FeMg wave windows, due to our computational capability. To evaluate the information loss, we calculate the variance of every FPCA basis function by using the equation

$$V_m = \frac{\sum_n (\beta_{m,n} - \bar{\beta}_m)^2}{\sum_n \left(\int_{\lambda_{\min}}^{\lambda_{\max}} X_n(\lambda)^2 d\lambda\right)} \quad (28)$$

where V_m is the variance of the m-th order basis function, $\beta_{m,n}$ is the m-th Score of the n-th spectra, $\bar{\beta}_m$ is the average of the m-th order Score of all spectra. We show the contribution of different wave windows to the variance in Figure 10. The first 10-order FPCA contribute more than 99% to the variance.

Appendix C
Possible Data Leak Hazard

Our data set includes 73 spectra of SNe IIL and 337 spectra of SNe IIP, in which 39 are from SN2013ej and 36 are from SN2004et. Because some SNe have more spectra, while other SNe have fewer, we split the total 73 or 337 spectra into the training set (80%) and the testing set (20%). This strategy will lead to the situation where some spectra of a specific SN are in the training set, while the others are in the testing set. Because the spectra from the same SN share similar features, it may lead the deep-learning model to favor the Type of SNe with more spectra. This effect is called ”data leak” in the deep-learning field. To investigate this effect our result, we divide all SNe into two groups with roughly same number of spectra: the frequently observed group and the less-observed group. For SN IIP, the frequently observed group contain 165 spectra from nine SNe that have at least 12 spectra in 4200–8900 Å: SN2004et, SN2009N, SN2005cs, SN2013am, SN2013K, iPTF13dqy, SN2012aw, SN2006bp, and SN1999em. The rest of the 172 type IIP spectra are in the less-observed group. For SN IIL, 39 spectra of SN2013ej constitute the frequently observed group, and rest IIL SNe are in less-observed group. We then apply the neural network with the best performance to train and test the ANN for 200 times; the accuracy of each groups are shown in Table 9. This result indicates that more data will improve this effect.

Table 9
The Accuracy of Frequently and Less-observed SNe

SN Type	Observation Frequency	Accuracy
SN IIP	Frequently observed	0.991
SN IIP	Less observed	0.983
SN IIP	All	0.987
SN IIL	Frequently observed	0.888
SN IIL	Less observed	0.669
SN IIL	All	0.785

References

Abbott, T., Abdalla, F. B., Aleksic, J., et al. 2016, MNRAS, 460, 1270
Anderson, J. P., Dessart, L., Gutiérrez, C. P., et al. 2018, NatAs, 2, 574
Anderson, J. P., Habergham, S. M., James, P. A., & Hamuy, M. 2012, MNRAS, 424, 1372
Arcavi, I. 2017, Hydrogen-Rich Core-Collapse Supernovae (Cham: Springer)
Bose, S., Kumar, B., Misra, K., et al. 2016, MNRAS, 455, 2712
Boser, B. E., Guyon, I. M., & Vapnik, V. N. 1992, in Proc. the Fifth Annual Workshop on Computational Learning Theory. COLT ’92, A Training Algorithm for Optimal Margin Classifiers (New York: ACM)
Chambers, K. C. & Pan-STARRS Team 2017, AAS Meeting Abstracts, 229, 223.03
Chang, C.-C., & Lin, Chih-Jen. 2011, ACM Trans. Intell. Syst. Technol., 2, 27:1

Chen, P.-H., Lin, C.-J., & Schölkopf, B. 2005, Appl. Stoch. Models Bus. Ind., 21, 111
Cristianini, N., & Ricci, Elisa. 2008, Encyclopedia of Algorithms (Springer)
de Jaeger, T., Galbany, L., Gutiérrez, C. P., et al. 2018, MNRAS, 478, 3776
Eldridge, J. J., & Tout, C. A. 2004, MmSAI, 75, 694
Elias-Rosa, N., Van Dyk, S. D., Li, W., et al. 2011, ApJ, 742, 6
Faran, T., Poznanski, D., Filippenko, A. V., et al. 2014, MNRAS, 445, 554
Fraser, M., Takáts, K., Pastorello, A., et al. 2010, ApJL, 714, L280
Gal-Yam, A. 2017, Observational and Physical Classification of Supernovae (Cham: Springer), 195
Gutiérrez, C. P., Anderson, J. P., Hamuy, M., et al. 2014, ApJL, 786, L15
Gutiérrez, C. P., Anderson, J. P., Sullivan, M., et al. 2018, MNRAS, 479, 3232
Hall, P., Müller, H.-G., & Wang, J.-L. 2006, AnSta, 34, 1493
Ivezic, Z., Kahn, S. M., Tyson, J. A., et al. 2019, ApJ, 873, 111
Jones, M. C., & Rice, J. A. 1992, The American Statistician, 46, 140
Kessler, R., Bassett, B., Belov, P., et al. 2010, PASP, 122, 1415
Kingma, D. P., & Ba, J. 2014, arXiv:1412.6980
Kochanek, C. S., Shappee, B. J., Stanek, K. Z., et al. 2017, PASP, 129, 104502
LeCun, Y., Bengio, Y., & Hinton, G. E. 2015, Natur, 521, 436
Li, W., Leaman, J., Chornock, R., et al. 2011, MNRAS, 412, 1441
Lochner, M., McEwen, J. D., Peiris, H. V., Lahav, O., & Winter, M. K. 2016, ApJS, 225, 31
Maguire, K., Di Carlo, E., Smartt, S. J., et al. 2010, MNRAS, 404, 981
Marcaide, J. M., Pérez-Torres, M. A., Ros, E., et al. 2002, A&A, 384, 408
McCray, R. 2017, The Physics of Supernova 1987A (Cham: Springer), 2181
Muthukrishna, D., Parkinson, D., & Tucker, Brad. 2019, ApJ, 885, 85
Olson, D. L., & Delen, Dursun. 2008, Advanced Data Mining Techniques (1st ed.; Berlin: Springer)
Patat, F., Barbon, R., Cappellaro, E., & Turatto, M. 1994, A&A, 282, 731
Peng, J., & Paul, D. 2007, arXiv:0710.5343
Ray, A., Petre, R., & Schlegel, E. M. 2001, AJ, 122, 966
Richmond, M. W., Treffers, R. R., Filippenko, A. V., et al. 1994, AJ, 107, 1022
Sasdelli, M., et al. 2016, MNRAS, 461, 2044
Savitzky, A., & Golay, M. J. E. 1964, AnaCh, 36, 1627
Schlegel, E. M. 1990, MNRAS, 244, 269
Shannon, C. E. 2001, SIGMOBILE Mob. Comput. Commun. Rev., 5, 3
Singh, A., Srivastav, S., Kumar, B., Anupama, G. C., & Sahu, D. K. 2018, MNRAS, 480, 2475
Smartt, S. J. 2015, PASA, 32, e016
Smith, N., Chornock, R., Silverman, J. M., Filippenko, A. V., & Foley, R. J. 2010, ApJ, 709, 856
Sun, F., & Gal-Yam, A. 2017, arXiv:1707.02543
Taddia, F., Moquist, P., Sollerman, J., et al. 2016, A&A, 587, L7
Tucker, M. A., Shappee, B. J., & Wisniewski, J. P. 2019, ApJL, 872, L22
Valenti, S., Sand, D., Stritzinger, M., et al. 2015, MNRAS, 448, 2608
Wozniak, P. R. et al. (ed.) 2014, in The Third Hot-wiring the Transient Universe Workshop (HTU-III)
Yaron, O., & Gal-Yam, A. 2012, PASP, 124, 668
Yuan, F., Jerkstrand, A., Valenti, S., et al. 2016, MNRAS, 461, 2003