A comparative view of machine learning models used to find Brown Dwarfs

Nik Mohsen

A senior thesis submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree of

Bachelor of Science

Denise Stephens, Advisors

Department of Physics and Astronomy

Brigham Young University

ABSTRACT

A comparative view of machine learning models used to find Brown Dwarfs

Nik Mohsen
Department of Physics and Astronomy, BYU
Bachelor of Science

Brown Dwarfs are ultra cold objects that are hard to see. As such, it is hard to find them without the correct instruments. New space telescopes will be launched soon that will be able to get thousands of new data points on new Brown Dwarfs. With all that new Data, it will be necessary to have efficient ways of picking out Brown Dwarf Candidates. In this paper we compare two machine learning algorithms and discuss the efficiency and accuracy of using them. The two algorithms are Kernel Localized Linear Regression and Red Dragon. We find that KLLR can produce fits similar to Red Dragon, but only with careful kernel-width selection, and this becomes harder in sparse regions of the brown dwarf data.

Keywords: Brown Dwarf, Machine Learning, Kernel Localized Linear Regression, Gaussian Mixture Model

ACKNOWLEDGMENTS

This research has made use of the SVO Filter Profile Service "Carlos Rodrigo", funded by MCIN/AEI/10.13039/501100011033/ through grant PID2023-146210NB-I00

This work has benefitted from The UltracoolSheet at http://bit.ly/UltracoolSheet, maintained by Will Best, Trent Dupuy, Michael Liu, Aniket Sanghi, Rob Siverd, and Zhoujian Zhang, and developed from compilations by Dupuy & Liu (2012, ApJS, 201, 19), Dupuy & Kraus (2013, Science, 341, 1492), Deacon et al. (2014, ApJ, 792, 119), Liu et al. (2016, ApJ, 833, 96), Best et al. (2018, ApJS, 234, 1), Best et al. (2021, AJ, 161, 42), Sanghi et al. (2023, ApJ, 959, 63), and Schneider et al. (2023, AJ, 166, 103).

I would like to thank the BYU Department of Physics and Astronomy for allowing me the opportunity to participate in research as part of my education and for funding my research endeavors.

I would also like to thank Denise Stephens, William Black, and Zachary Shakespeare for their continued mentorship on this project, without which I would not have a senior thesis to write.

Finally, I would like to thank my wife, Jessica, for her continued support and encouragement throughout my academic career. I could not do this without her.

Contents

Chapter 1

Introduction

Brown dwarfs are low-mass, substellar objects that cannot sustain fusion and therefore have no means of self-heating, making them extremely difficult to detect in the wavelength regimes normally used for stellar classification. However, as telescope technology improves, it is becoming increasingly possible to gather photometric data on cooler objects. The Euclid preliminary data release has already produced promising results in the search for brown dwarfs, with Kiwy et al. 2026 discovering 15 new objects in a targeted small section of sky used as a preliminary testing ground for Euclid [1]. In addition, Honaker and Gizis 2025 predict that there may be up to 750,000 brown dwarfs in our solar neighborhood alone [2]. With the launch of Euclid and the upcoming Nancy Roman Space Telescope, the volume of data on cool objects is expected to increase dramatically. This creates strong potential for new discoveries, but also a growing need for efficient methods to identify the objects most worthy of further study.

Machine learning methods for understanding astronomical phenomena are not uncommon. Red Dragon, seen in Black and Evrard 2022, is an algorithm that uses a Gaussian mixture model to identify qualities of galaxies that help properly identify them [3]. This model has recently been adjusted to help predict the likelihood of an object being a Brown Dwarf based on numerical spectral type and color. Local Linear Regression (LLR) was a model used in Farahi et al. 2018 to compare

the mass of baryonic matter vs. dark matter to understand how Dark Matter Halos evolved [4]. Now known as KLLR from Farahi et al. 2022, it too has been modified to predict the likelihood of an object being a brown dwarf [5].

This project uses data from The UltracoolSheet, a catalog of known brown dwarf photometry, spectroscopy, and astrometry, curated by Best et al. [6]. From this catalog, we used photometric data of brown dwarfs from Gaia, WISE, and 2MASS to create the colors that KLLR and Red Dragon then use for brown dwarf prediction. After selecting the optimal kernel width for KLLR in each individual color space, we then compared the results to Red Dragon to determine whether our methodology is sound and to determine where the discrepancies between the two models were.

Chapter 2

Methods

2.1 The Data

The UltracoolSheet is a compilation of known data for over 4000 brown dwarfs. The data are taken from multiple different sources and compiled by Dr. William Best as a resource for people who want to study these objects [6]. For this project, we use specifically photometric data on brown dwarfs that have known data across the Gaia, 2MASS, and WISE surveys. These surveys were chosen because they cover a wide range of the infrared spectrum; thus, we were able to ensure that we can catch brown dwarfs across multiple filters. Of the 3891 entries in the UltracoolSheet, 1817 fit the criteria for this project.

2.1.1 GAIA Photometry

In Figure 2.1 we see that the Gaia space telescope is equipped with three main filters: the blue part, the green, and the red part. Together these filters cover a range of just under 3300 Å to just over 10 500 Å. The blue part isolates the first half of that range, coming from about 3300 Å to 6700 Å, and thus is specifically equipped for handling bluer brown dwarfs. Likewise, the red part isolates

the latter half of the *Gaia* filters, from about 6200 Å to about 10 500 Å. Its increased effectiveness in the redder half of this wavelength band actually makes it more effective for finding redder, dimmer stars than the regular *G* filter. And so, we use all three filters to construct two of the colors used in our research.

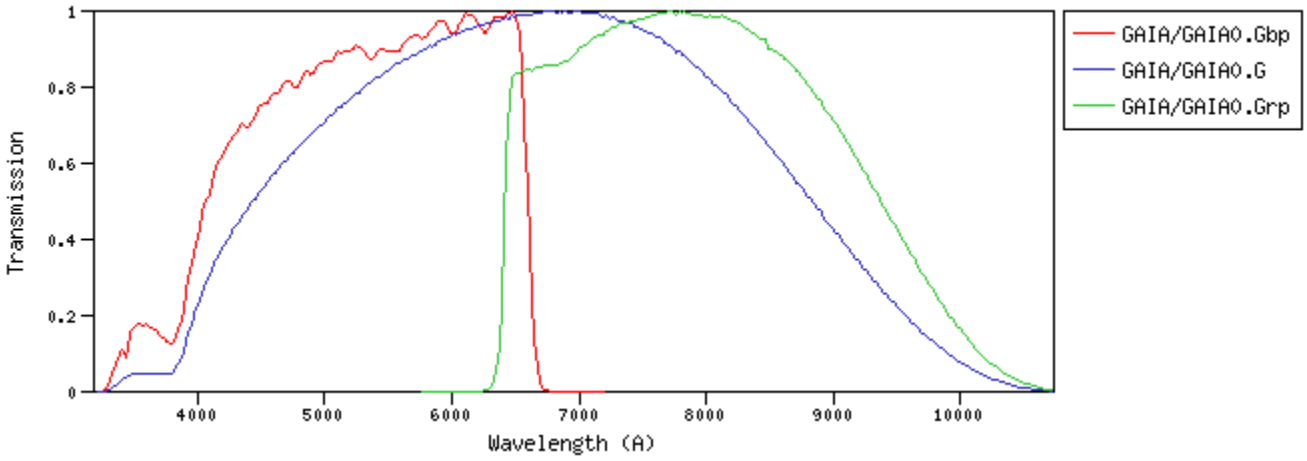


Figure 2.1 Gaia filter profiles showing wavelength range (measured in Angstroms) for Gaia instruments and transparency of each filter. Provided by SVO Filter Profile Service [7]

2.1.2 2MASS Photometry

Unlike Gaia, 2MASS has three filters that cover separate wavelength bands in the IR. Figure 2.2 shows 2MASS *J* covering from about 10 700 Å to 14 200 Å. 2MASS *H* covers from about 14 400 Å to about 18 200 Å. 2MASS K_s covers from about 19 400 Å to 23 800 Å, which covers a sizeable portion of the mid-infrared. This is helpful for us, as we can cover a wide range of magnitudes across known brown dwarfs.

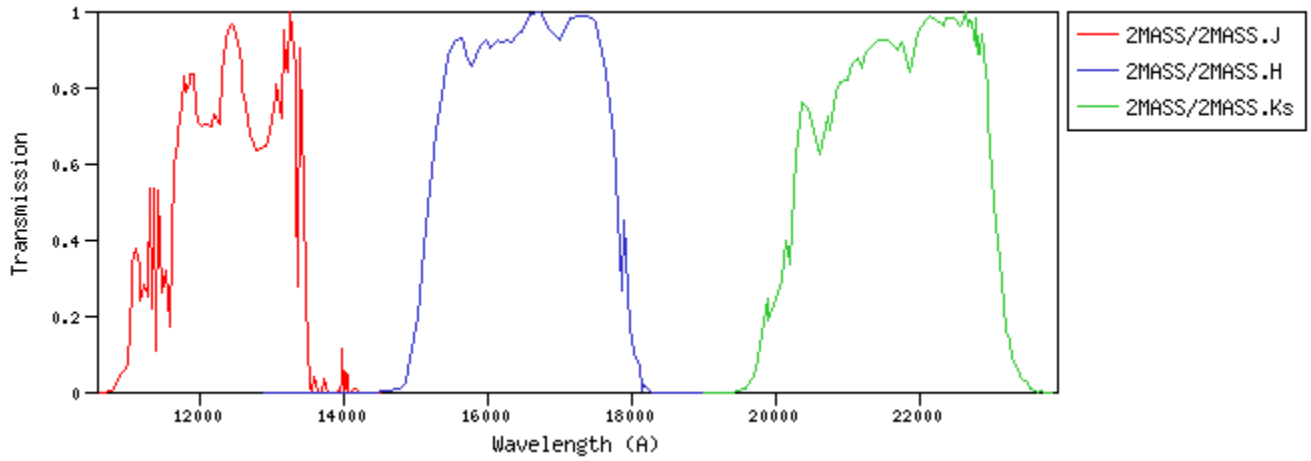


Figure 2.2 2MASS filter profiles showing wavelength range (measured in Angstroms) for 2MASS instruments and transparency of each filter. Provided by SVO Filter Profile Service [7]

2.1.3 WISE Photometry

Lastly, we took data from WISE, which covers wavelength bands in the latter half of the mid-infrared. Note that WISE 3 and WISE 4 were excluded because they are considered in the far-infrared regime. Again, WISE 1 and WISE 2, like 2MASS, are two distinct filters that have no overlap. Figure 2.3 shows WISE 1 ranges from about 27 500 Å to about 38 700 Å, and WISE 2 ranges from about 39 600 Å to 53 400 Å.

2.2 Color

The measurements from the UltracoolSheet that we use are the apparent magnitudes of each brown dwarf in the respective filter. In other words, this is a measurement of how bright you would see that star when looking through the filter. However, the metric that we are actually comparing to spectral type is the color. Color is a term astronomers use to describe a difference in magnitude between two filters. Apparent magnitude is a function of distance:

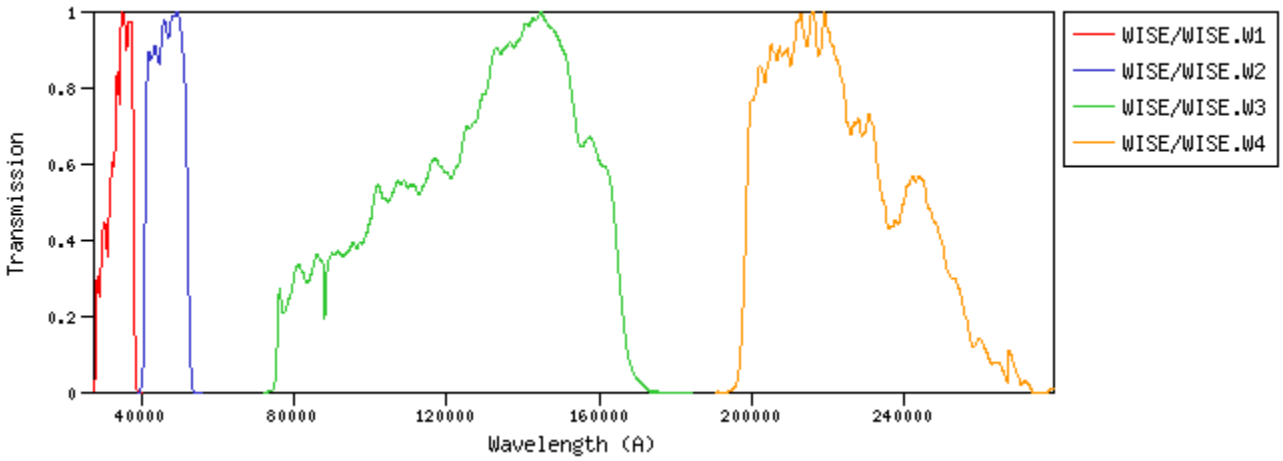


Figure 2.3 Wise filter profiles showing wavelength range (measured in Angstroms) for Wise instruments and transparency of each filter. Provided by SVO Filter Profile Service [7]

$$m = 5\log_{10}\left(\frac{d}{10}\right) + M, \tag{2.1}$$

where m is the apparent magnitude, d is the distance in parsecs, and M is the absolute magnitude.

If color is the difference between the apparent magnitudes of two different filters, then we can use it to analyze each star without having to know the distance or the apparent magnitude. Furthermore, since we no longer care about the distance, we have a valuable tool for comparison to other brown dwarfs.

The colors that we built and are using for this project are *Gaia* BP−*Gaia* G, *Gaia* G−*Gaia* RP, *Gaia* RP−2MASS J, 2MASS J−2MASS H, 2MASS H−2MASS K_s, 2MASS K_s−WISE 1, and WISE 1−WISE 2. This gives us seven unique colors for each brown dwarf across the whole near- to mid-infrared range, which we will compare in each color space to numerical spectral type to identify the brown dwarfs.

2.3 Numeric Spectral Type

One way stellar objects have historically been classified is by spectral type. Spectral type is a classification system that groups stars by how similar their spectra are, giving a letter assignment to stars with the same or similar spectra [8]. Traditionally, these classifications are O, B, A, F, G, K, and M, with subcategories 0 to 9 [8]. Because the composition and ionization of elements in a star are intrinsically related to the temperature of the star [9], spectral type is also a good way to classify stars via temperature. According to this convention, the hottest main-sequence stellar objects are classified as O0 and the coldest main-sequence stellar objects are classified as M9. Though brown dwarfs are not considered main-sequence objects, this convention is extended to them as well. In Kirkpatrick et al. 1999, objects cooler than M were assigned the classification of L type [10], and an even colder spectral type of T was proposed for brown dwarfs cooler than L type. Later, in Burgasser et al. 2006, T dwarfs were adopted as the official classification for brown dwarfs colder than L [11].

The data we are using in this project consist of cool dwarf-type objects ranging from early-M to late-T dwarfs. Rather than representing these objects with their letter/number classification, we use the strictly numeric classification provided by the UltracoolSheet. This is done so that it is easier to bin objects for a more continuous data set rather than using discrete spectral classifications. Additionally, it is easy for the code to read in a numerical value rather than a letter-number combination.

In the convention Best uses for the UltracoolSheet, 0 corresponds to a spectral type of M0, 10 corresponds to L0, and 20 corresponds to T0 [6]. Of the 1817 Brown Dwarfs taken from the UltracoolSheet, 790 are M dwarfs (numerical value of 0 to 9.9), 1009 are L dwarfs (numerical value of 10 to 19.9), and 18 are T dwarfs (numerical value greater than 20). The sparsity of data on T dwarfs is due to the selection criteria of the data set. As discussed previously, Gaia's filters are only sensitive out to about 10 500 Å, which is the tail end of the optical band. As such, only the brightest

T dwarfs are visible to Gaia. The rest are too faint to detect. This sparsity of data in the later end of the fit has implications which will be discussed later in this paper.

2.4 Kernel Localized Linear Regression

The Kernel Localized Linear Regression model (KLLR) is a machine learning model that uses linear regression on individual bins to produce a non-linear fit to a given set of data [5]. In this context, the term "bin" refers to an evaluation point (a location on a fixed grid in the x variable) where a local regression is performed. The data are not partitioned into disjoint bins; instead, the kernel defines an overlapping neighborhood around each bin center, so individual data points can influence multiple bins with different weights.

Whereas a normal linear regression factors in every data point, giving all of them equal weight and producing a linear best fit, KLLR looks at a specific bin and produces a linear best fit locally, hence the "local" aspect of Kernel Localized Linear Regression. A kernelin the context of KLLR is a weighted window that controls which data points are considered in the regression and how much weight is given to each data point. Intuitively, it acts like a spotlight that slides across the x axis highlighting each point. Data points close to the center are given more weight than points towards the edges. KLLR applies the kernel to each bin, the regression is run, and the collection of the fits are combined to make a smooth trend line across all data points.

In practice, the way the “spotlight” fades from the center to the edges depends on the kernel function. KLLR offers two kernel options: a top-hat kernel and a Gaussian kernel. A top-hat kernel assigns equal weight to all data points within the kernel window (and zero weight outside it), so the “spotlight” is just as bright at the edges as it is at the center. In this project we use a Gaussian kernel, which downweights points smoothly as their distance from the bin center increases: the spotlight is

brightest at the center and gradually dims outward. The Gaussian kernel weight is

$$w_i \propto \exp\left[-\frac{1}{2}\left(\frac{x_i - x_\star}{h}\right)^2\right], \tag{2.2}$$

where the proportionality indicates that only the relative weighting between points matters for the fit. Here $x_\star$ is the bin center (evaluation point), x_i is the x value of a given data point being evaluated as part of the regression, and h is the kernel width [5]. The kernel width is a user-specified parameter that scales the weights in Eq. 2.2. The input value of the kernel can be a single number, which applies the same kernel width across all bins, or it can be an array of numbers that create a dynamic kernel width that changes from bin to bin based on the corresponding value in that element in the Array. That is to say, the kernel width for the nth bin will be the value of the nth element of the array. In this way, KLLR allows the user to control the smoothness of the regression based on the data available in and around a given bin. Because KLLR runs regression based on the weighted priority of the data points in the kernel, it is important to pick a kernel width that realistically fits the raw data.

Mathematically speaking, the optimal kernel width is

$$h_n^\star = \left(\frac{mC_2}{4C_1 n}\right)^{\frac{1}{m+4}} \tag{2.3}$$

where h_n^* is the optimal kernel width, m is the number of input dimensions being smoothed over, n is the number of data points in the local sample, and C_1 and C_2 are constants that describe how the bias and variance of the fit change with kernel width [12]. In practice, C_1 and C_2 are not known ahead of time, so they are usually estimated from the data, or the kernel width is chosen by testing several values. However, if we assume C_1 and C_2 are constants, and since we are only smoothing over 1 dimension, we can take optimal kernel width to be

$$h_n^\star \propto n^{-\frac{1}{5}}, \tag{2.4}$$

which provides a useful rule to follow. In Eq. 2.4, h_n^* decreases as n increases, meaning that in the data-dense regions, it's better to use a smaller kernel width, and in the data-sparse regions, it is better to use a bigger kernel width. To estimate uncertainty in each fit, KLLR uses bootstrapping. To do this, KLLR builds a pseudo dataset by randomly selecting data points from the original set with replacement until the new set has the same amount of data points as the original set. KLLR will then rerun the regression on the pseudo dataset and then repeat the process up to the specified amount of Bootstraps. This produces a distribution of fitted values at every bin, which can be used to estimate confidence intervals on the trend. Throughout this work we use the spread of the bootstrap fits to estimate the uncertainty in the trend, and we plot this uncertainty as the shaded region around the mean fit.

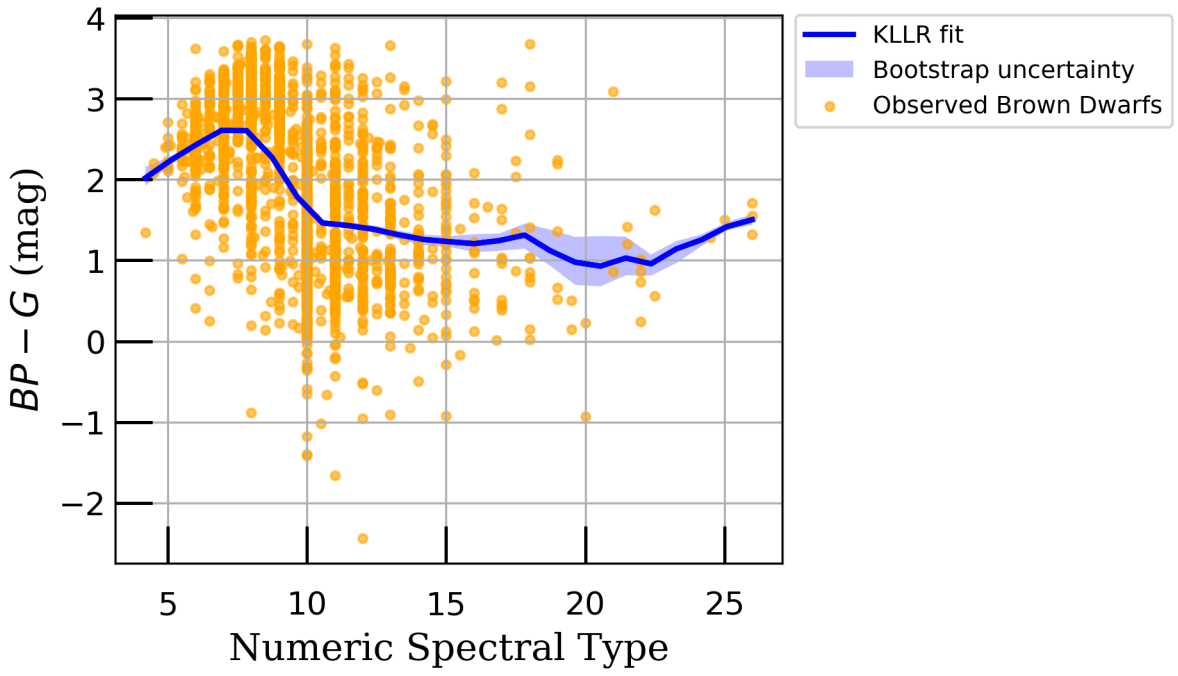


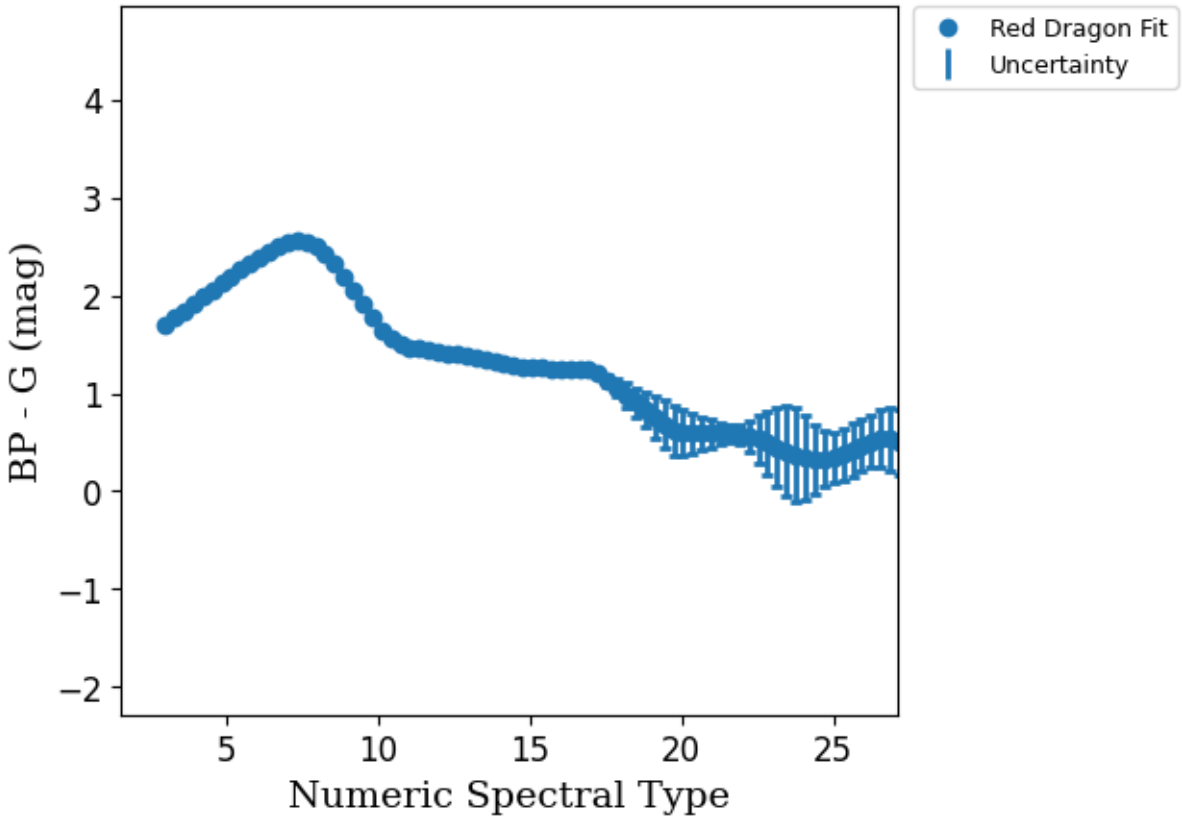
Figure 2.4 Example KLLR output showcasing the raw Brown Dwarf Color data and the KLLR fit in the *GaiaBP* − *GaiaG* color space. This fit was run using 25 bins, 50 bootstraps, and a kernel width of 1.

Figure 2.4 is a representation of the full KLLR output for a given kernel width. The output shows the Brown Dwarf $GaiaBP - GaiaG$ color data and the KLLR fit using a single-value kernel width of 1 evaluated across 25 bins, and bootstrapped 50 times. The bootstrap uncertainty is represented by the shaded area around the KLLR fit. In the data-dense region there is hardly any uncertainty, as opposed to the data-sparse regions where we start to see more apparent uncertainty. This pattern is indicative of the need for more data in the 17 to 25 spectral type region and will be common across all colors.

2.5 Red Dragon

Red Dragon takes a different approach to fitting the data than KLLR [3, 13]. Red Dragon uses a Gaussian mixture model to fit the data. Unlike KLLR, which is a one-dimensional fit with respect to numerical spectral type, a Gaussian mixture model can be multi-dimensional. Basically what Red Dragon does is group the data by bin across our seven-dimensional color space and run a minimization algorithm such that the groupings are the smallest, most approximate fit that they can be. It then runs KLLR on the seven-dimensional groupings to give continuity to the discrete bins. The benefit of using Red Dragon is that because it is in a multi-dimensional space, it factors in the covariance between different colors in addition to the individual colors themselves, as opposed to KLLR, which can only factor in one color at a time. In doing this we can capture more of the relation between different color spaces; however, when we take our one-dimensional slice and look at the fit for one color, we can expect some discrepancies between Red Dragon and KLLR. That is not to say that one is better than the other. It just depends on how you want to view it.

Figure 2.5 is a raw Red Dragon output. As mentioned above, Red Dragon is taking a one-dimensional slice out of a seven-dimensional fit. Red Dragon is not parameterized by kernel width; as such, the fit that you would get for a particular color is the only fit you would expect to get from



[b]

Figure 2.5 Example Red Dragon Fit for *GaiaBP* − *GaiaG* color data and uncertainty. No Brown Dwarf data is innately plotted with Red Dragon

Red Dragon. Red Dragon does not innately show the raw data; however, in the next section we will show the Red Dragon fit against the raw data and the KLLR fit. The Red Dragon output shows a best-fit point per bin as well as error bars for that given point.

Chapter 3

Results

In this section, we analyze the effects that kernel width has on the KLLR fits, as well as look at comparisons between the optimal-fit kernel width and scaled-up and scaled-down comparison kernel widths. We then look at direct comparison of Red Dragon overlaid with KLLR optimal fits and analyze similarities between the two as well as their discrepancies.

3.1 Kernel Width Optimization

As predicted, KLLR kernel widths have an impactful effect on the quality of the fit produced. Figure 3.1 shows a fit conducted on *Gaia* BP−G color versus numerical spectral type with a kernel width of 0.1. In Figure 3.1, we see a lot of sharp curves indicative of possible noise in the data from overrepresenting nearby data points and underrepresenting farther data points. This is particularly visible from spectral-type value 15 to 25, where the data are a lot sparser and the fit is a lot more jagged. The overall fit between 10 and 15 is alright, but we see a decently large margin for error as well as a very sharp indent in the fit where we should expect to see a smoother curve. We also see a spike in residual and scatter at values higher than 10, which is indicative of a poor match in

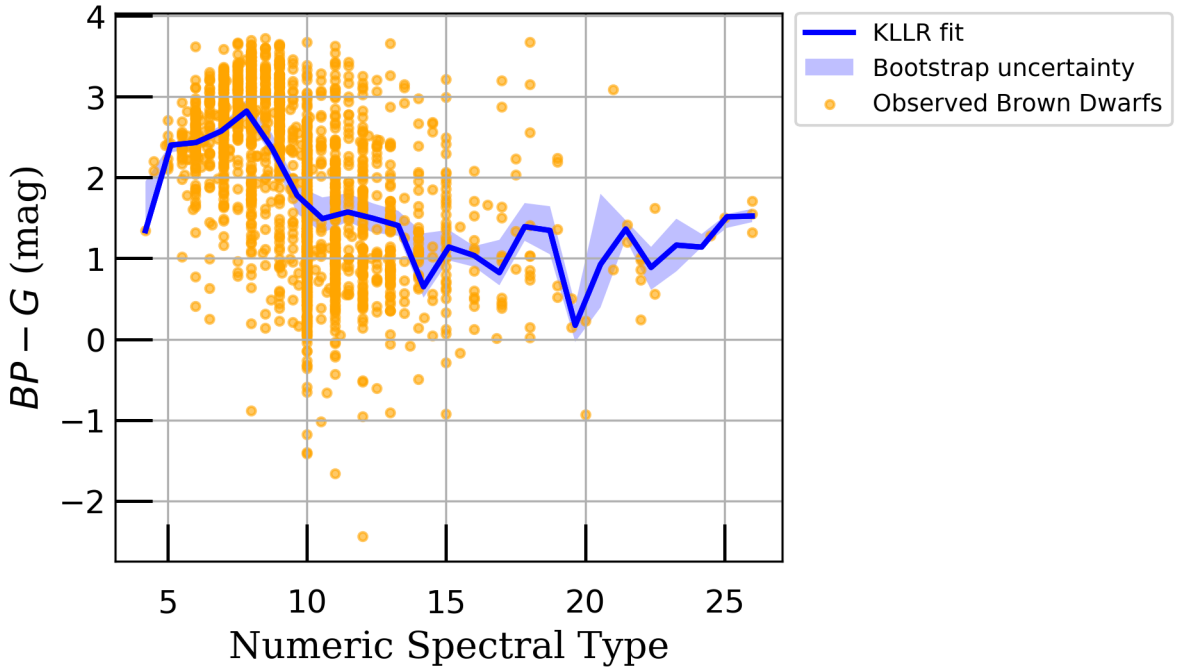


Figure 3.1 KLLR $BP_{\text{Gaia}} - G_{\text{Gaia}}$ fit with kernel width of 0.1.

those areas. The 1 to 10 area is fine but could use a bit more smoothness for ease of observation and statistical accuracy.

Figure 3.2 shows the same Gaia BP−G color versus numerical spectral type, but now with a kernel width of 10. In this fit we see, in the 1 to 15 range of spectral types, a really smooth downward linear curve. Again, as we expected, the algorithm is underprioritizing local data points and overprioritizing farther data points, causing the general area to be smoothed out and dampening important features in that range. From 15 to 25, we see a smooth downward transition as well, with a little bit of ambiguity toward the tail end, which we would expect at least at the very end, but there are still some features that have been dampened by the large kernel width.

Figure 3.3 shows the same *Gaia* BP−G color versus numeric spectral type, but now with a kernel width of 1. In Figure 3.3, we see a very smooth beginning curve from spectral types 1 to 10. We see the same characteristics and shape as in Figure 3.1 (kernel width 0.1) but without the

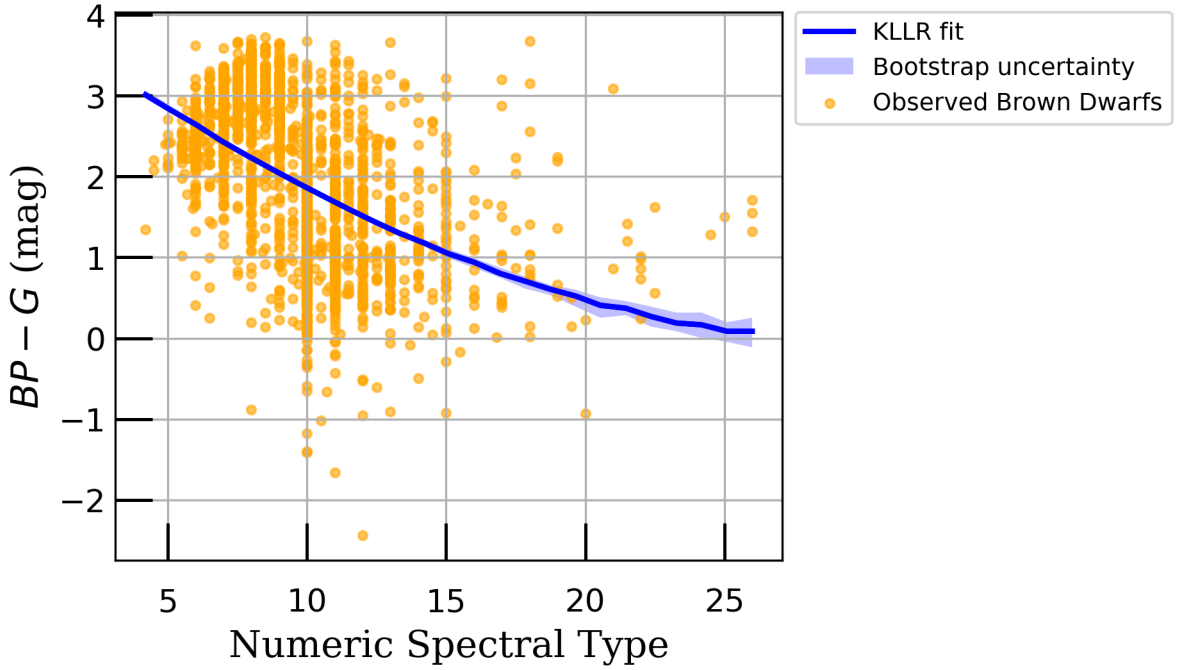


Figure 3.2 KLLR $BP_{\text{Gaia}} - G_{\text{Gaia}}$ fit with kernel width of 10.

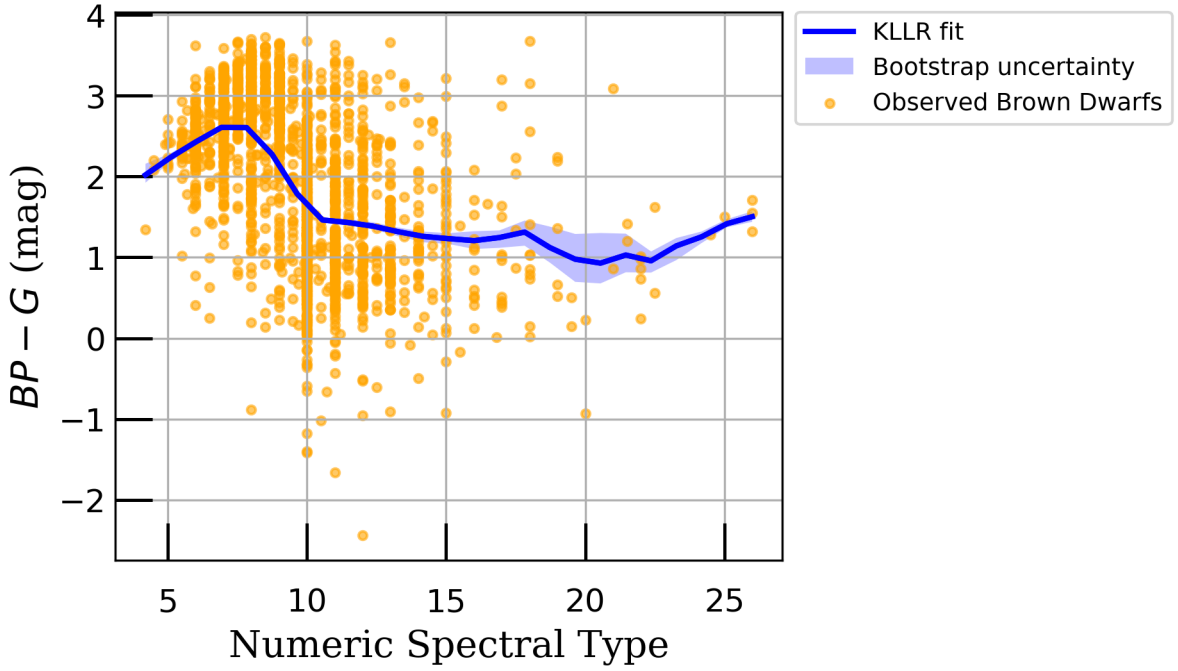


Figure 3.3 KLLR $BP_{\text{Gaia}} - G_{\text{Gaia}}$ fit with kernel width of 1.

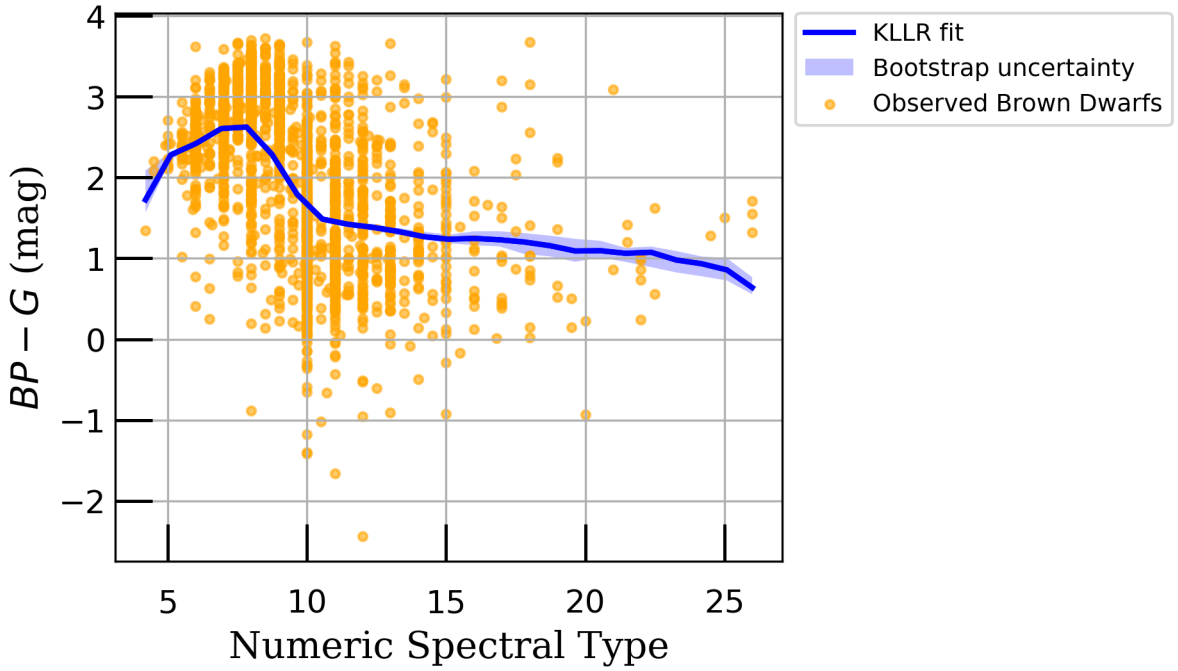


Figure 3.4 KLLR $BP_{\text{Gaia}} - G_{\text{Gaia}}$ fit with a dynamic kernel width.

jaggedness of a low kernel width. Between points 10 and 15 we see a linear decline which, while still accurate, is a little bit too rigid as seen in the residual and scatter plots. And as we observe data points above 15, we see that there is a bit too much noise in those data points, indicating that we still need a larger kernel width to handle those sparse data points.

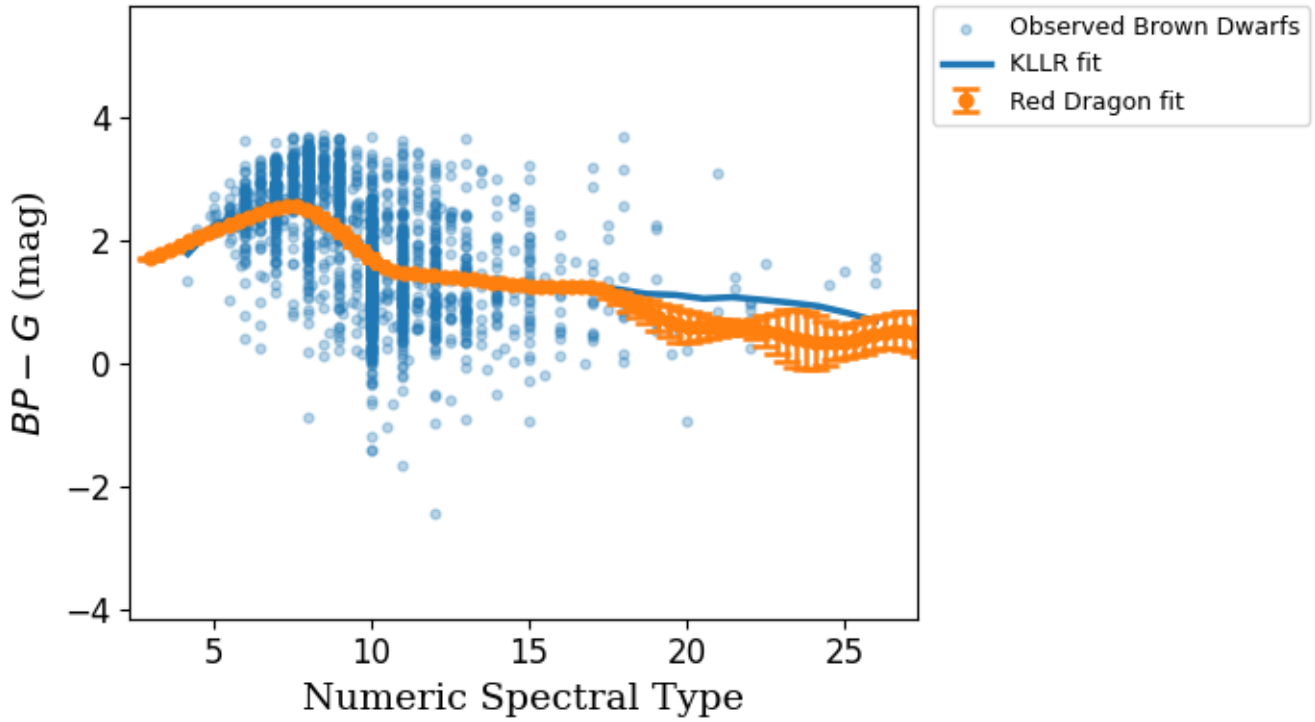
In order to find the optimal kernel width, we applied dynamic scaling to the fit where we manually altered the width of the kernel on a per-bin basis. This allows us to apply finer detailed fits to areas that are data-heavy while also allowing us to cut out the noise of data-sparse regions. We then iteratively changed each element in the array to find a dynamic kernel width that fit the data as best as possible. Figure 3.4 shows BP−G using a dynamic kernel width where the current fit starts at a kernel width of 0.5 and increases to the tail end at a kernel width of 7. Looking at the data, we see that it fits quite well with this dynamic kernel width in the data-dense region. In the data-sparse region, you do see a little bit of a divergence between the fit and the data, but that is to

be expected since we do not have a full enough data set of observed brown dwarfs to show the trend for sure. Thus, we see in the data some uncertainty around the end points. Appendix A contains the optimal KLLR fits for all seven colors.

3.2 Red Dragon Comparison

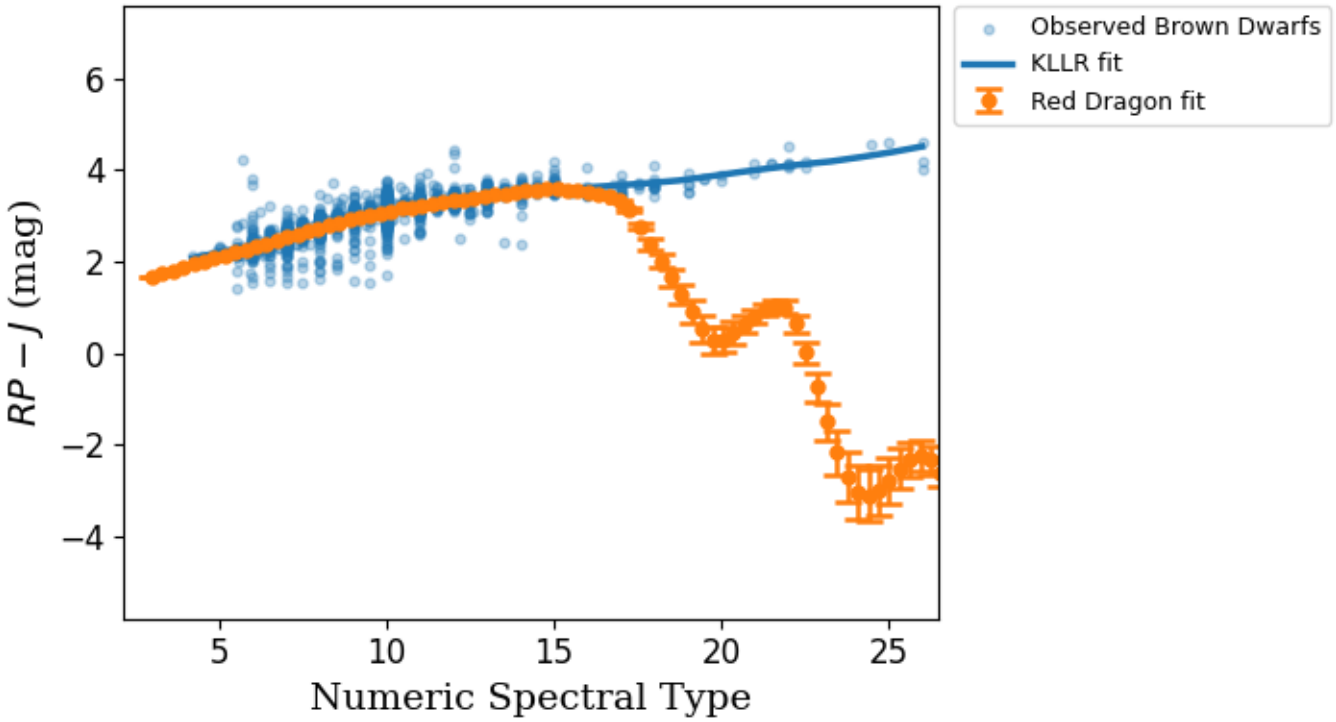
Our results show that we can get an accurate representation of the Red Dragon plot using the KLLR algorithm. Figure 3.5 shows the Red Dragon fit for *Gaia* BP−G overlaid onto the KLLR optimal-kernel-width fit and the raw data. The overlay shows that, for the most part, in the 0 to approximately 17 spectral-type range, there is heavy agreement between KLLR and Red Dragon. Both have an initial increase in curvature and then a decay into a flat line. The disagreement starting at around 17 shows Red Dragon steadily declining while KLLR continues a sort of flat trend with a minor dropoff. Note, however, that error bars at the end of the Red Dragon fit suggest that the KLLR fit is not too far off from the possible predicted values by Red Dragon.

Up until now we have used $GaiaBP - GaiaG$ for each of our example figure. In order to contrast a case where Red Dragon and KLLR do not agree well, we are going to introduce GaiaRP-J in Figure 3.6. In this figure we see KLLR follow the data even after the 17 mark where Red Dragon begins this sort of erratic decline. The error bars for Red Dragon now do not support the KLLR model as much. By inspection, we can see that Red Dragon is predicting bluer brown dwarfs out to spectral type 25, while KLLR is predicting a steady increase in redness, which we take to be more physically accurate since we do not expect objects to become bluer as they get colder. A possible explanation for the discrepancy could be the sparsity in the data. We have already made the claim that these models are not so accurate past spectral type 17 because of the lack of data. Red Dragon could also be interpreting data from the covariance of other colors since it approaches the problem from a seven-dimensional point of view. The last possibility is some missing factor that we have not



[b]

Figure 3.5 Red Dragon $BP_{\text{Gaia}} - G_{\text{Gaia}}$ overlaid on the KLLR fit and brown dwarf data as a function of numeric spectral type.



[b]

Figure 3.6 Red Dragon $RP_{\text{Gaia}} - J_{\text{2MASS}}$ overlaid on the KLLR fit and brown dwarf data as a function of numeric spectral type.

accounted for in the data, the KLLR model, or the Red Dragon model. Further research can be done to explain this discrepancy.

Chapter 4

Conclusion

In this paper, we set out to answer the question, "Does Red Dragon produce a reasonable output, and what differences are there between methods?" To do this, we applied a different machine learning algorithm, KLLR, to the raw Brown Dwarf data. After obtaining best fits for each color, we then compared the fits to Red Dragon to see if we get a reasonable match and to explain the differences between the models.

Our results showed that, given the right parameters, KLLR will produce a fit that closely matches the fits produced by Red Dragon. The caveat is that kernel width needs to be custom-tailored for the data set being used. The Best Ultra Cool Sheet is complete enough for numeric spectral types less than approximately 17 that we were able to estimate a good kernel width. However, the region greater than numeric spectral type 17 is a lot more data sparse, and therefore it was harder to determine an optimal kernel width. Red Dragon also interpolates the data differently than KLLR (taking a slice of a 7-D fit, rather than building an inherent 2-D fit), and thus the end behavior doesn't always agree. The other caveat is that all our KLLR fits were produced using 25 bins, and little to no research was done in measuring the effects of altering bin size.

In terms of direct comparison between Red Dragon and KLLR, both produced similar results across all seven colors, again disregarding certain caveats in data sparsity and optimal kernel width.

It then falls to the user to decide what the optimal model should be. Red Dragon is a more detailed model than just KLLR in that you can build a multidimensional color space and see how each color is dependent on each other. In other words, you can build a complete covariance matrix. KLLR might be simpler to implement, but requires either a more complete data set or user-defined kernel widths for each of the seven different colors we examined, meaning that KLLR, while simpler to implement, does require more work on the user's part. At the end of the day, this project will be used as a check against Red Dragon for a larger project involving Brown Dwarf identification, and in that respect, we say that Red Dragon produces favorable results and KLLR confirms that to be the case.

Future research on this topic could include bringing in more data to complete the data set. As we discover more Brown Dwarfs, naturally we will be able to bring in more objects and thus define the process of using Red Dragon and KLLR. One could also put more research into finding an optimal kernel width across our seven color spaces in order to find a more statistically accurate best-fit prediction. In addition to optimizing kernel widths, future research could go into examining the effects of higher or lower bin sizes on the data. Twenty-five bins is the standard, but it can easily be surmised that you might find more smoothness in using a lower number or more accuracy in using a higher number, depending on how accurate one would want to be. Lastly, it is worth noting that KLLR and Red Dragon are just two of many machine learning models that are available, and though KLLR and Red Dragon fulfill the purpose of this project, one might also find it worth it to study the implementation of other machine learning models on this data set should they desire further accuracy.

Appendix A

Optimal KLLR Fit Figures

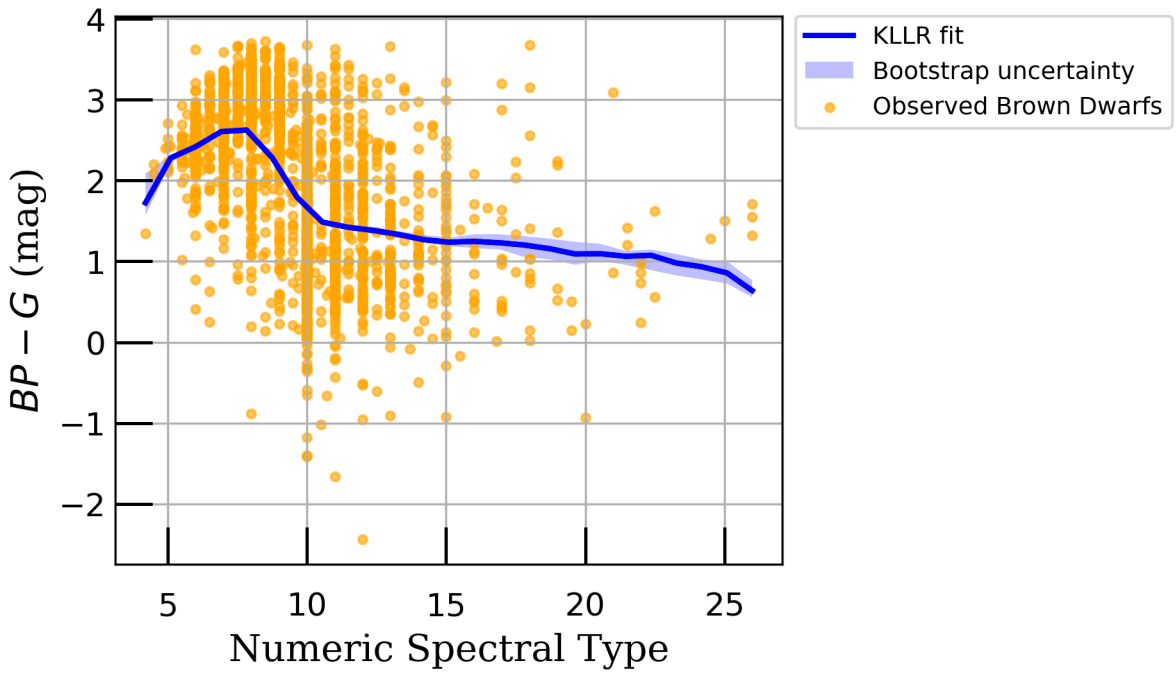


Figure A.1 KLLR $BP_{\mathrm{Gaia}} - G_{\mathrm{Gaia}}$ fit as a function of numeric spectral type, using the optimal kernel width.

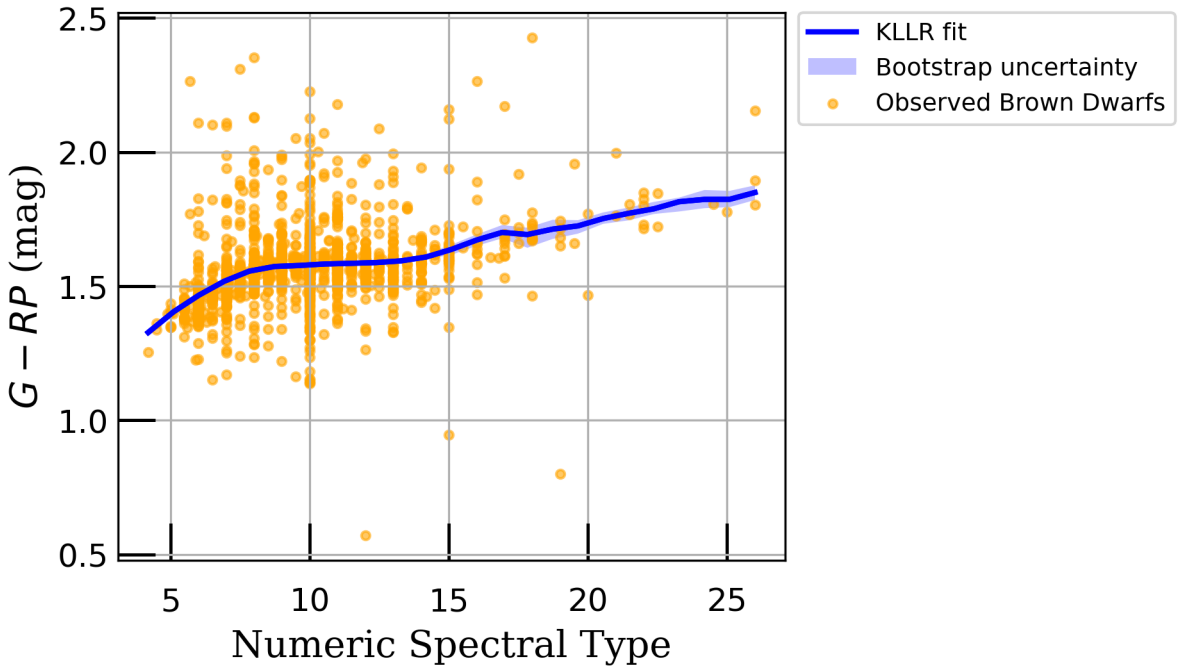


Figure A.2 KLLR $G_{\mathrm{Gaia}} - RP_{\mathrm{Gaia}}$ fit as a function of numeric spectral type, using the optimal kernel width.

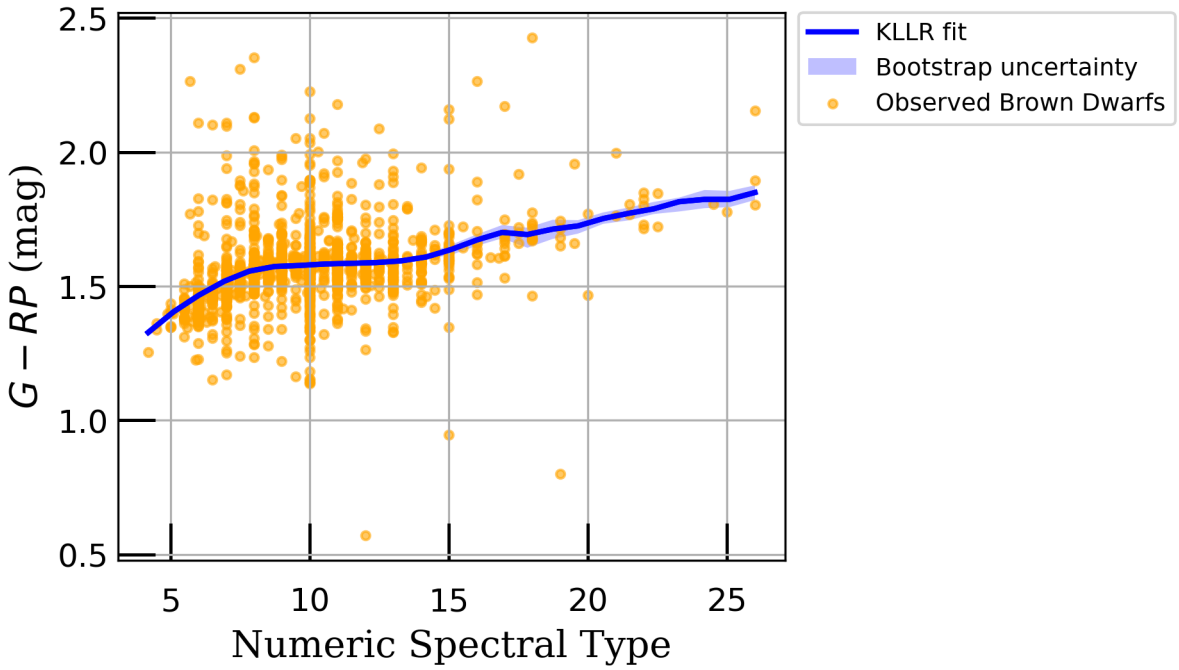


Figure A.3 KLLR $G_{\mathrm{Gaia}} - RP_{\mathrm{Gaia}}$ fit as a function of numeric spectral type, using the optimal kernel width.

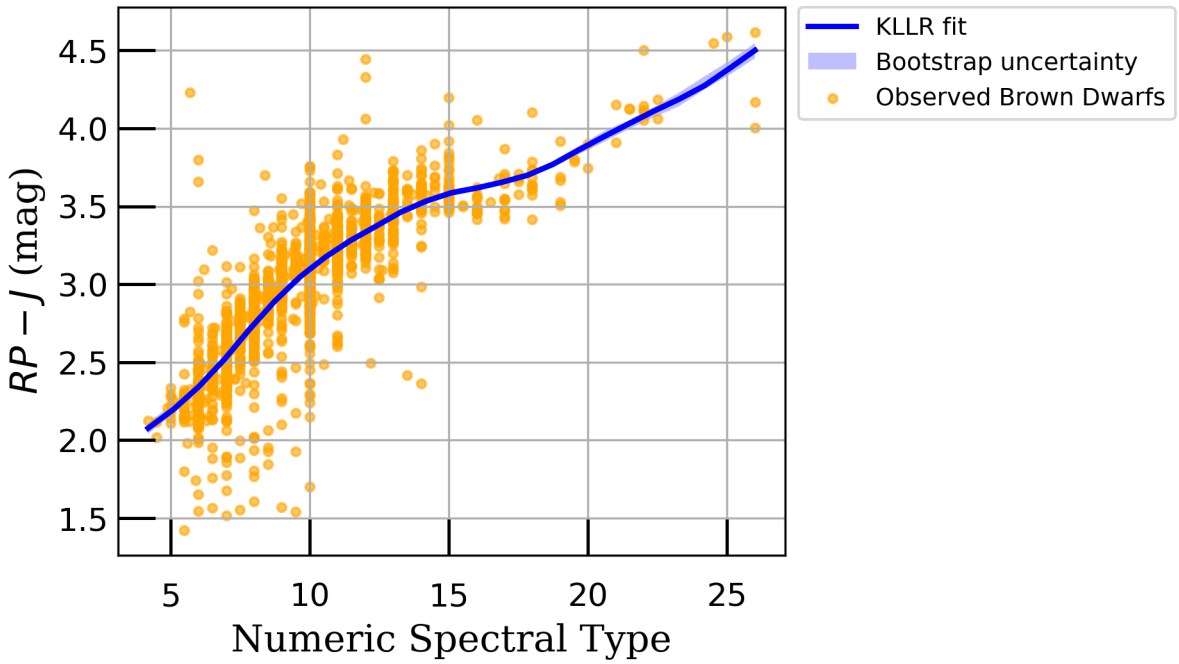


Figure A.4 KLLR $RP_{\mathrm{Gaia}} - J_{\mathrm{2MASS}}$ fit as a function of numeric spectral type, using the optimal kernel width.

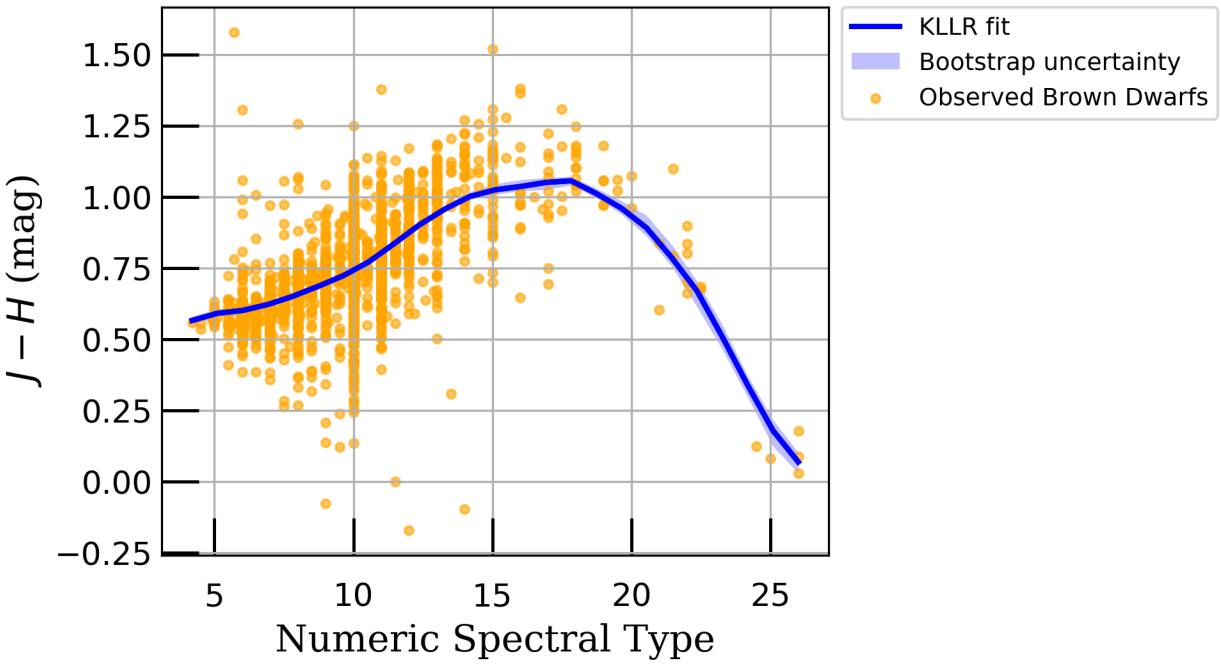


Figure A.5 KLLR $J_{2\mathrm{MASS}} - H_{2\mathrm{MASS}}$ fit as a function of numeric spectral type, using the optimal kernel width.

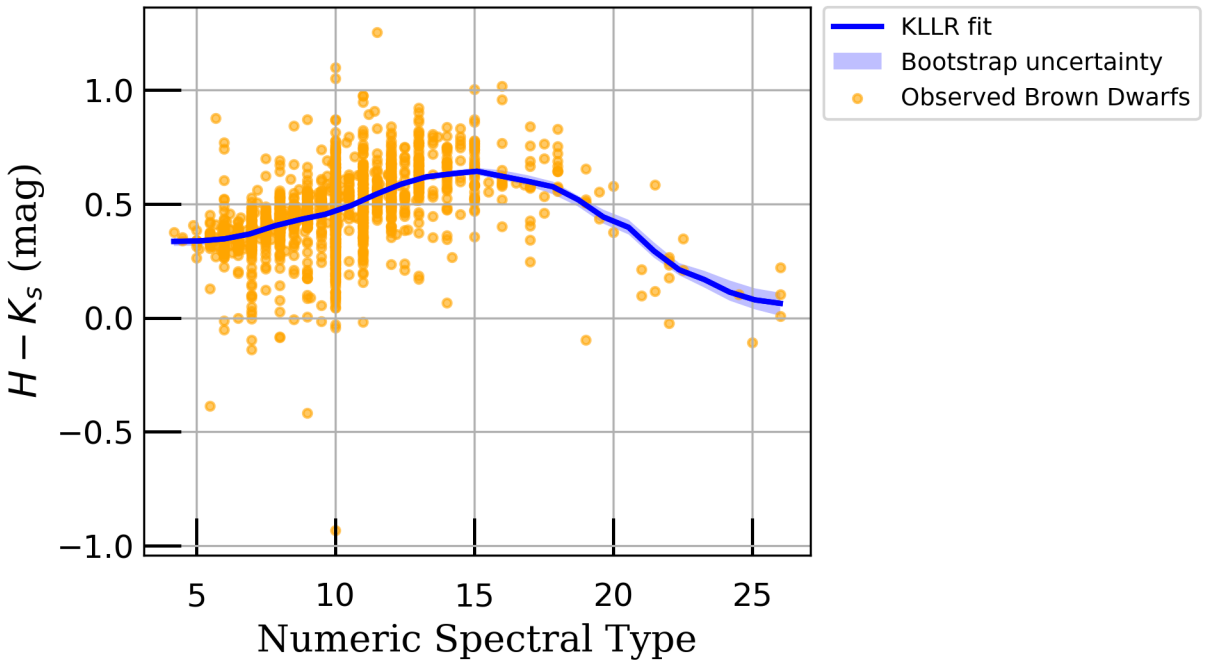


Figure A.6 KLLR $H_{2\mathrm{MASS}} - K_{s,2\mathrm{MASS}}$ fit as a function of numeric spectral type, using the optimal kernel width.

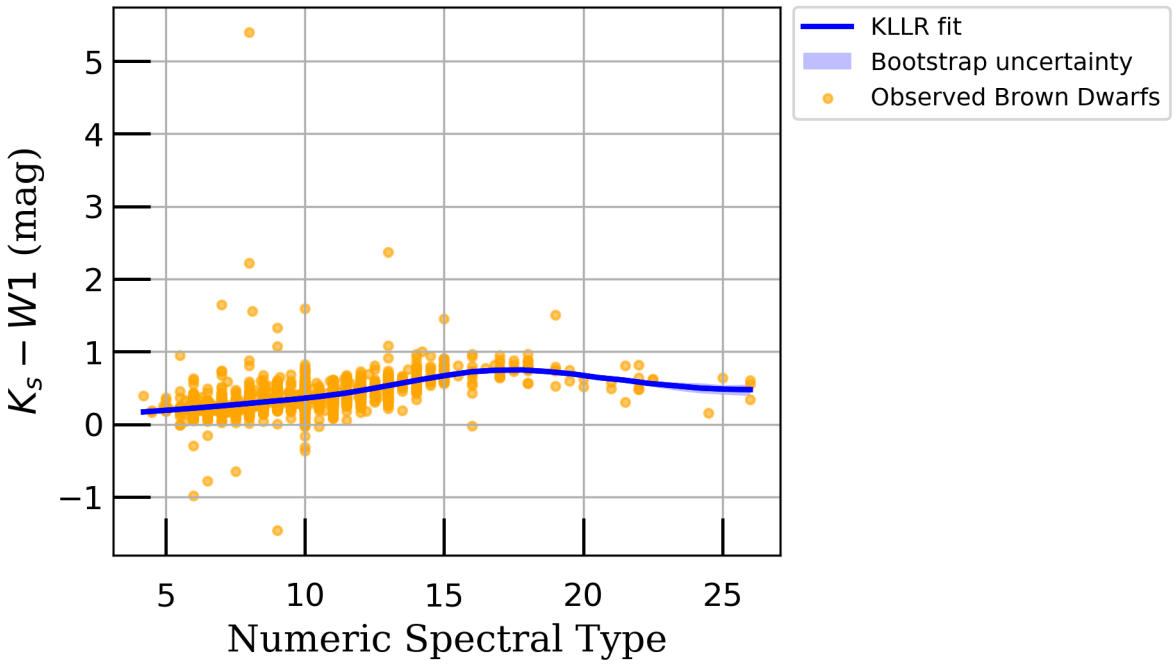


Figure A.7 KLLR $K_{s,\mathrm{2MASS}} - W1_{\mathrm{WISE}}$ fit as a function of numeric spectral type, using the optimal kernel width.

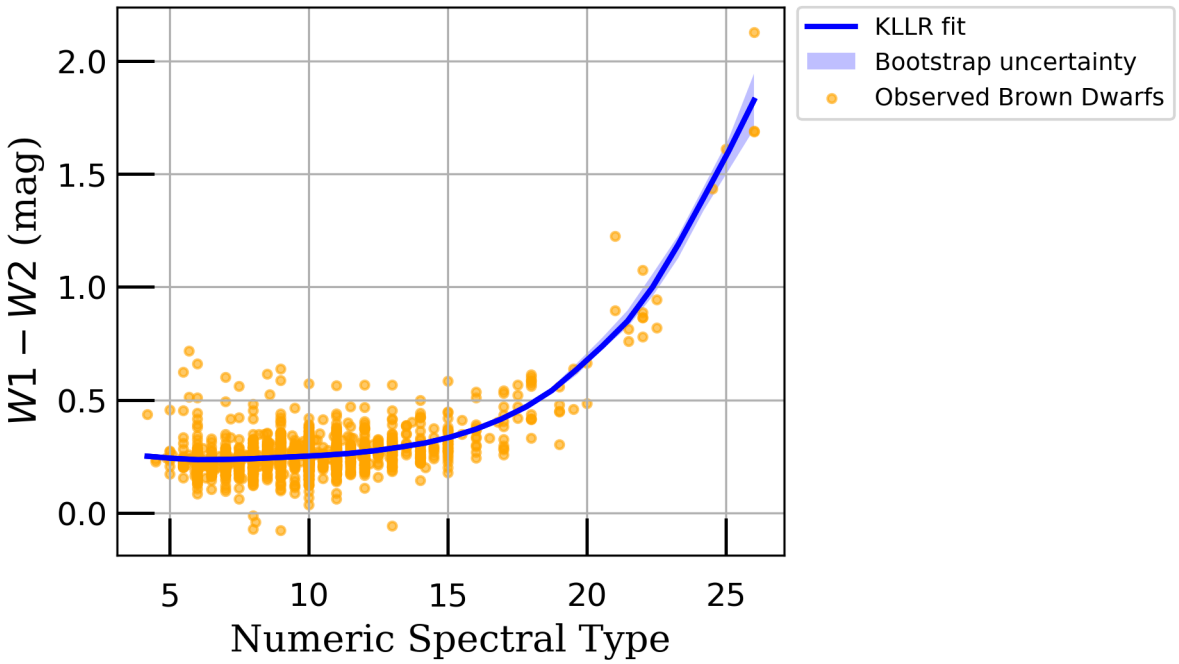


Figure A.8 KLLR $W1_{\mathrm{WISE}} - W2_{\mathrm{WISE}}$ fit as a function of numeric spectral type, using the optimal kernel width..

Appendix B

KLLR vs Red Dragon Comparison

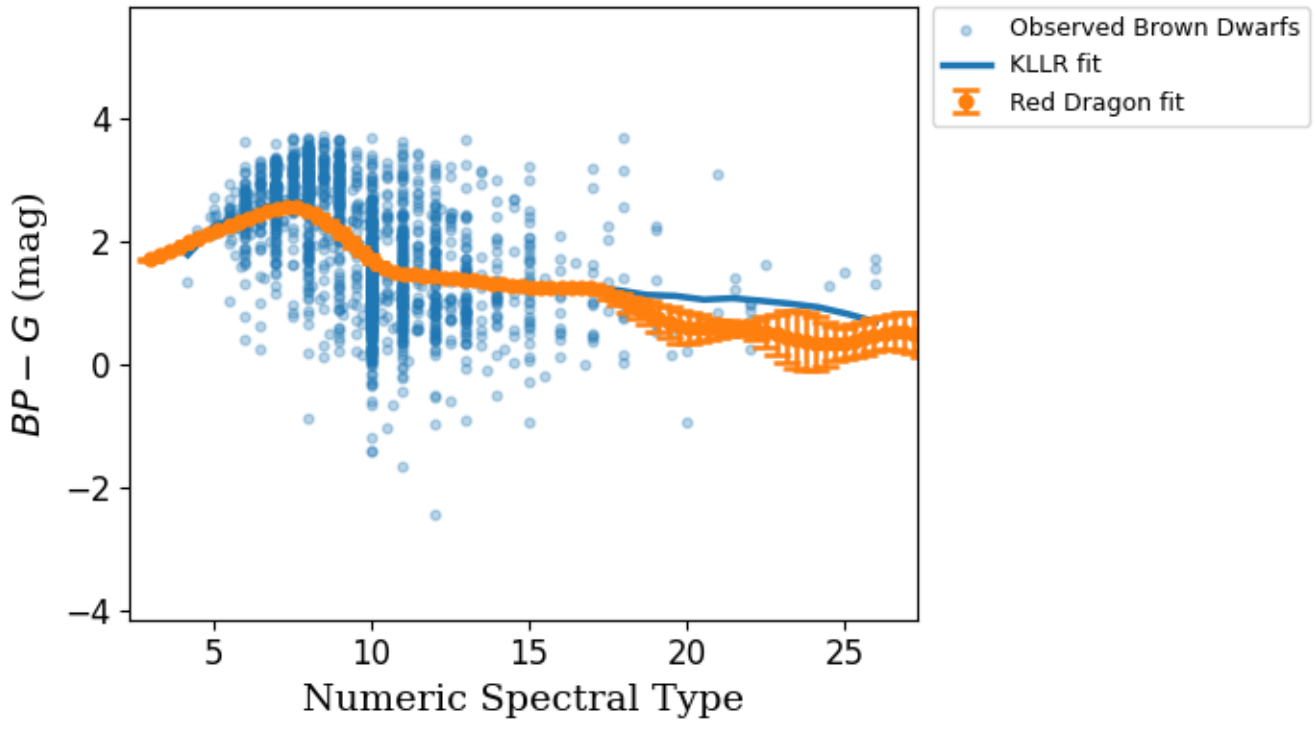


Figure B.1 Red Dragon $BP_{\mathrm{Gaia}} - G_{\mathrm{Gaia}}$ overlaid on the KLLR fit and brown dwarf data as a function of numeric spectral type.

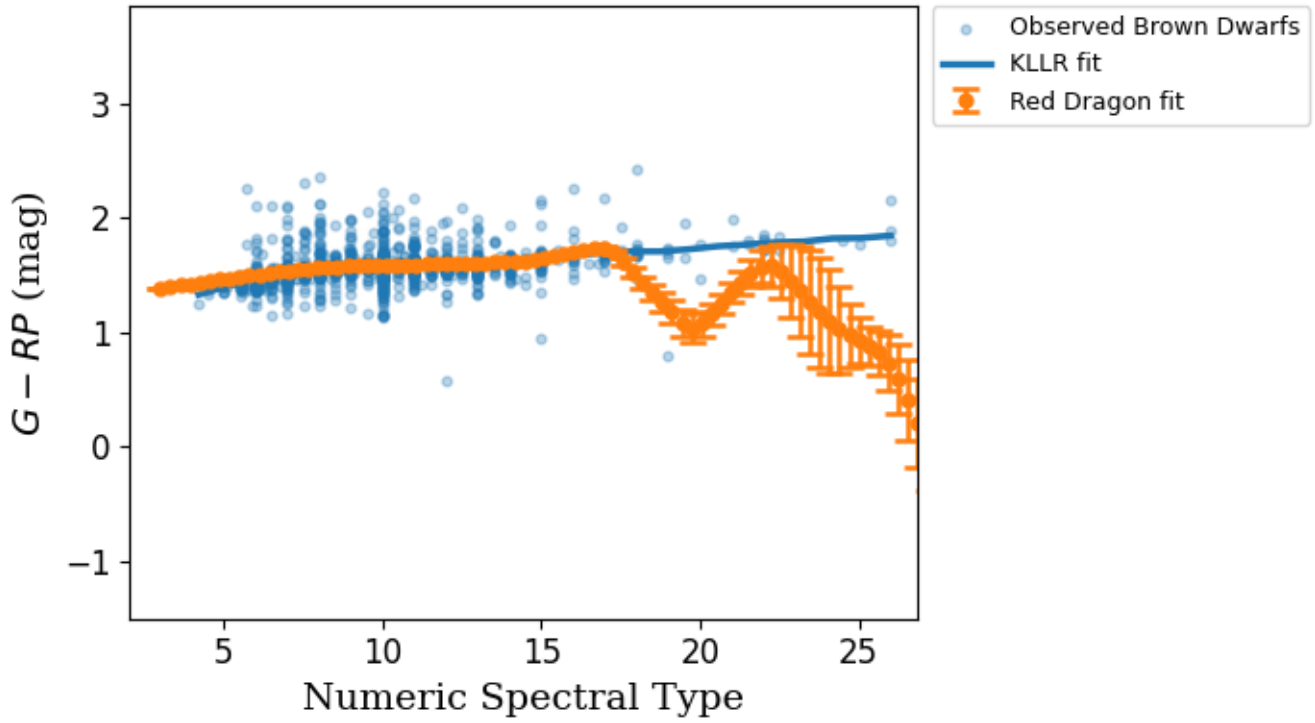


Figure B.2 Red Dragon $G_{\mathrm{Gaia}} - RP_{\mathrm{Gaia}}$ overlaid on the KLLR fit and brown dwarf data as a function of numeric spectral type.

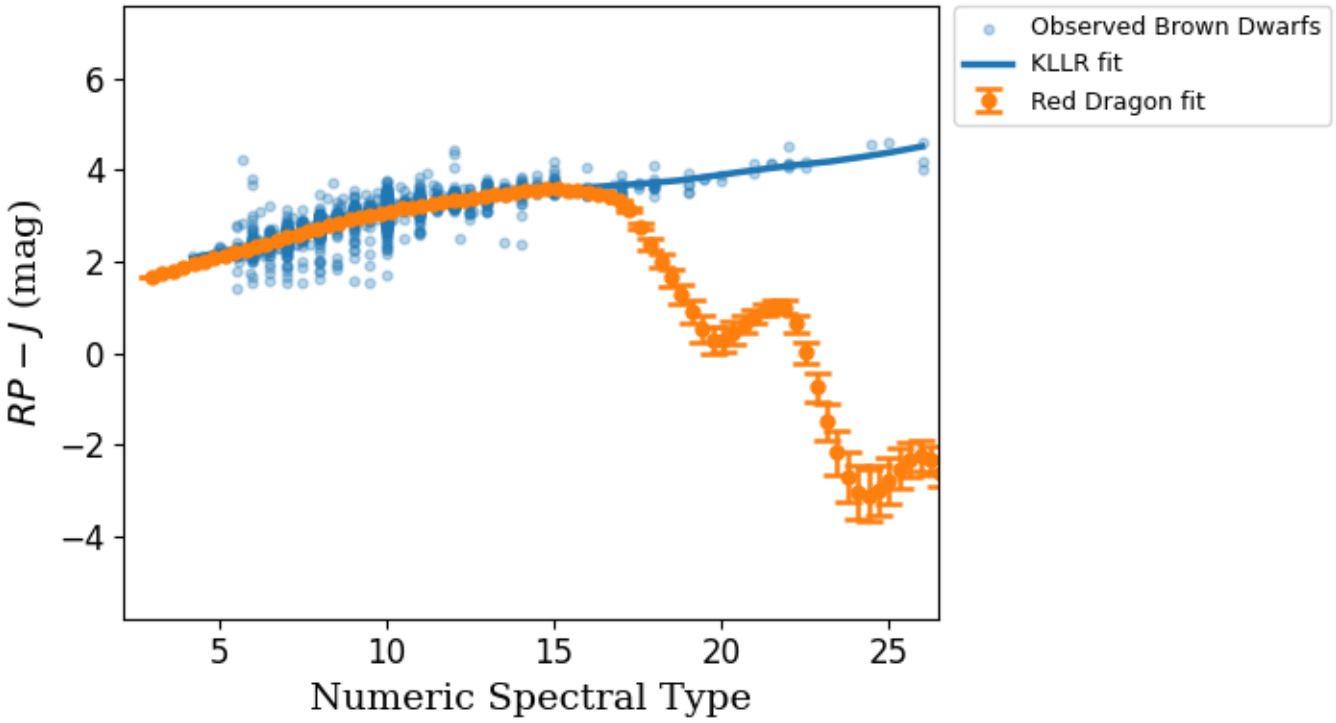


Figure B.3 Red Dragon $RP_{\mathrm{Gaia}} - J_{\mathrm{2MASS}}$ overlaid on the KLLR fit and brown dwarf data as a function of numeric spectral type.

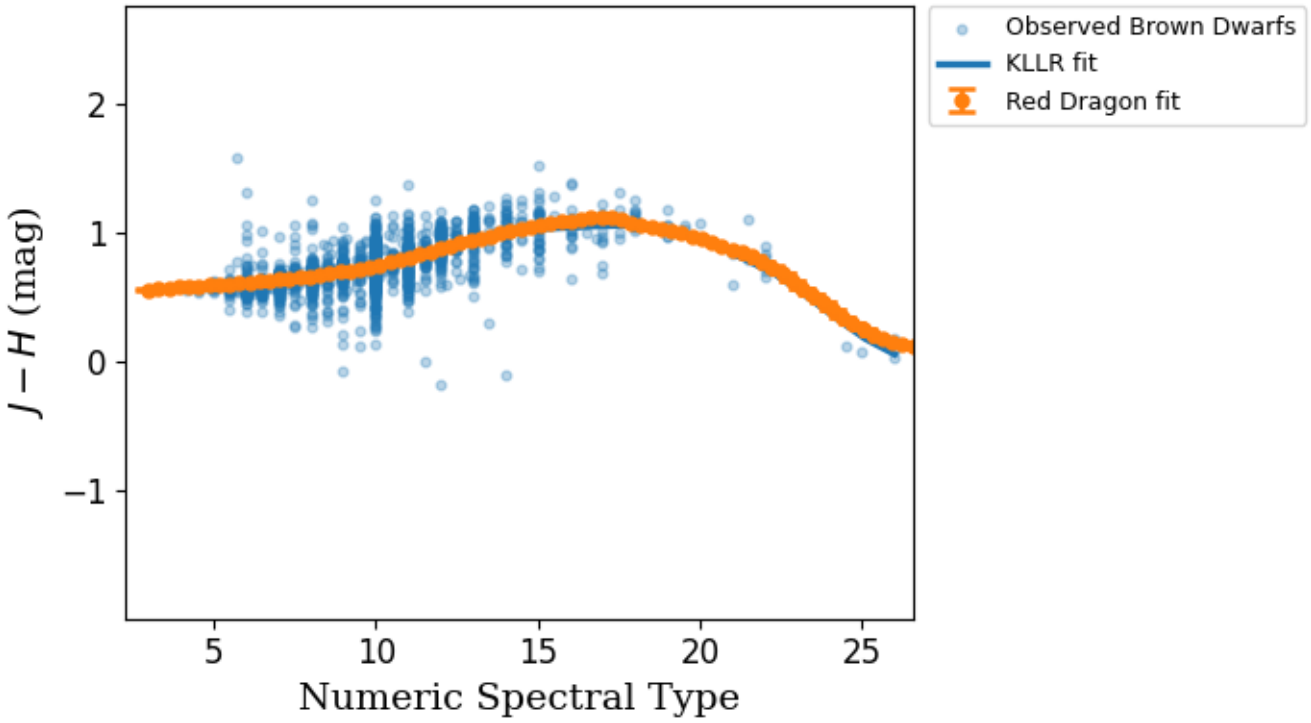


Figure B.4 Red Dragon $J_{2\mathrm{MASS}} - H_{2\mathrm{MASS}}$ overlaid on the KLLR fit and brown dwarf data as a function of numeric spectral type.

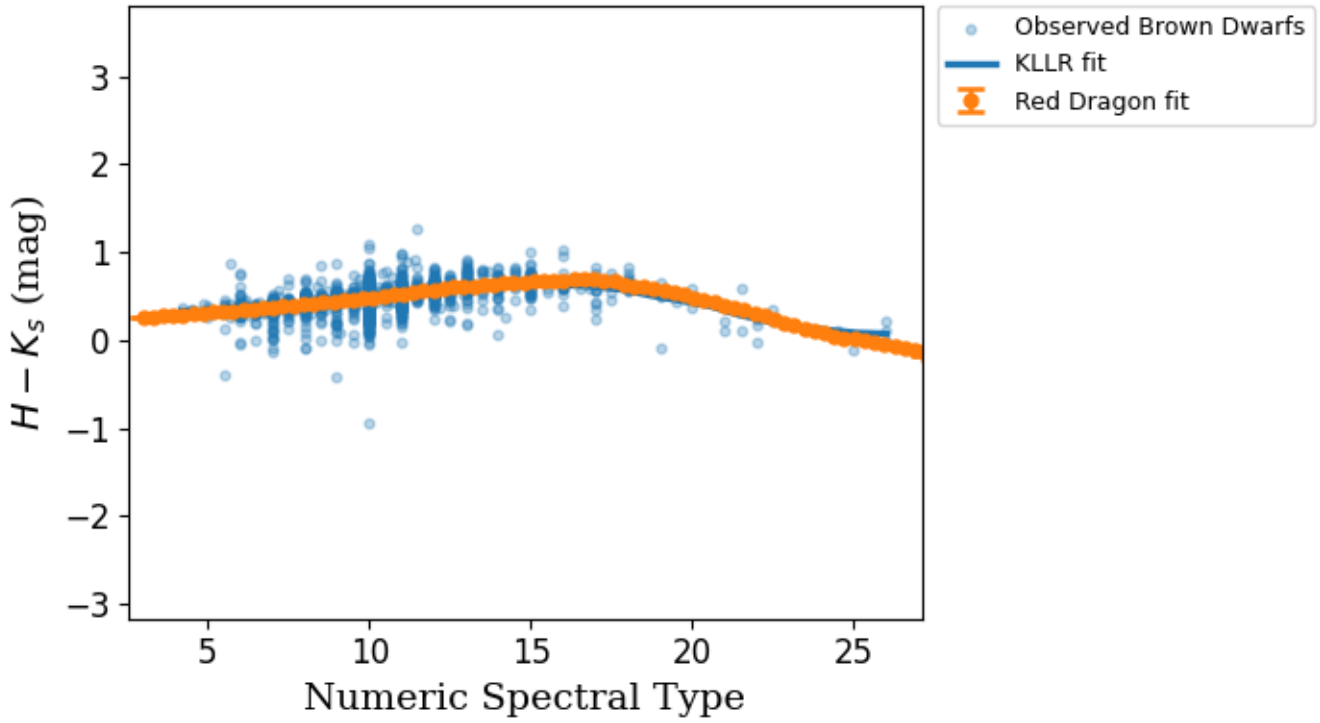


Figure B.5 Red Dragon $H_{\mathrm{2MASS}} - K_{s,\mathrm{2MASS}}$ overlaid on the KLLR fit and brown dwarf data as a function of numeric spectral type.

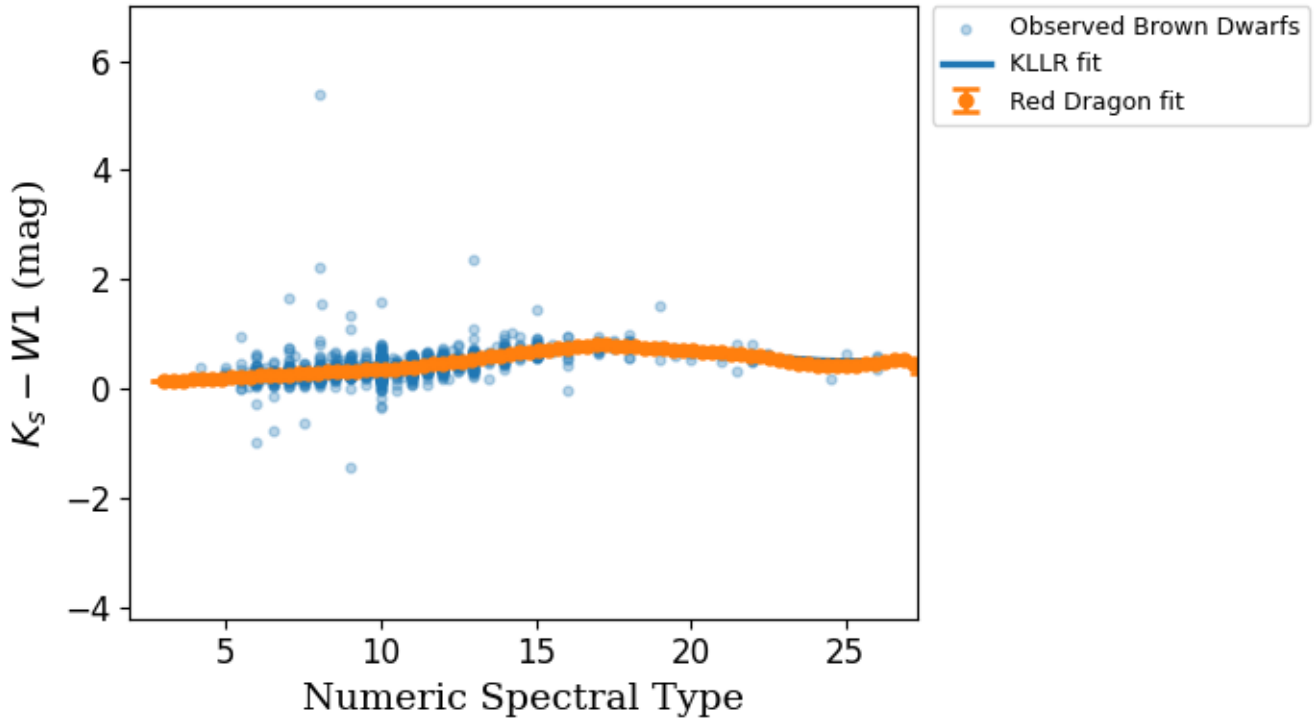


Figure B.6 Red Dragon $K_{s,\mathrm{2MASS}} - W1_{\mathrm{WISE}}$ overlaid on the KLLR fit and brown dwarf data as a function of numeric spectral type.

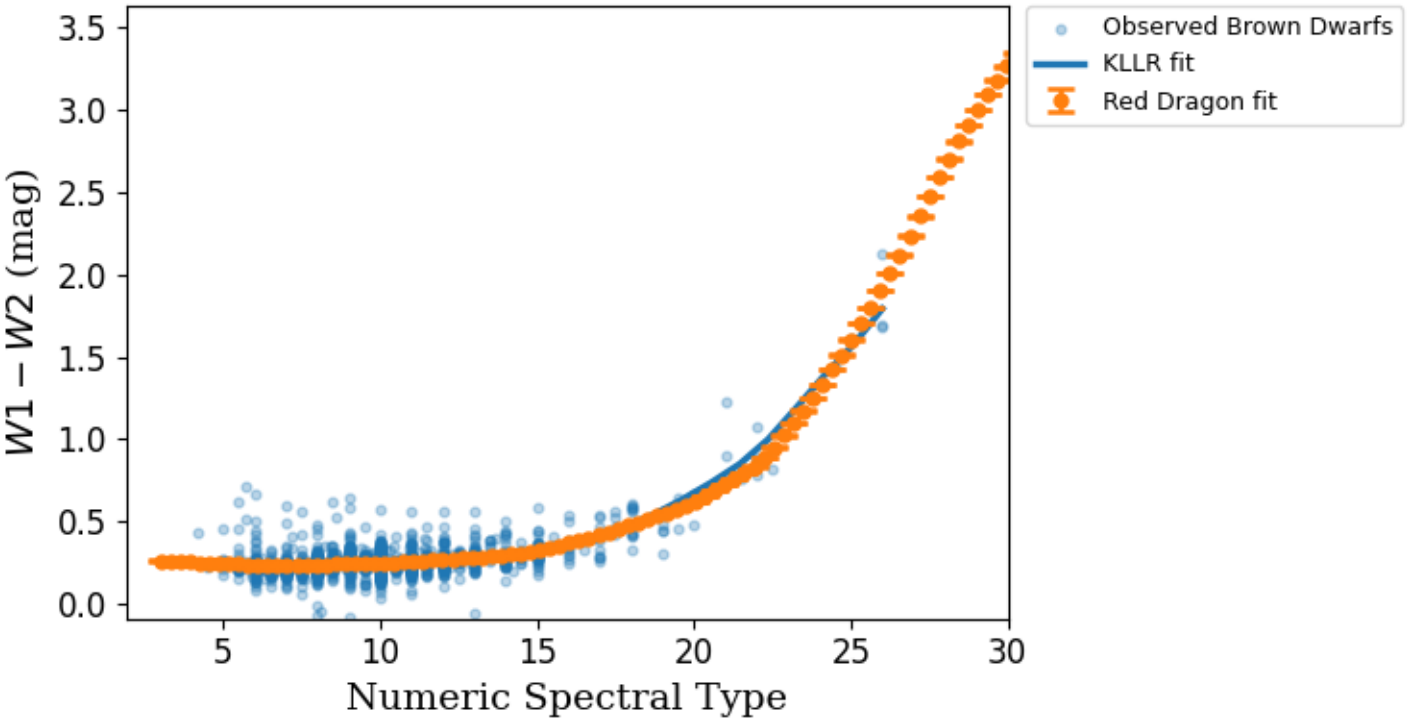


Figure B.7 Red Dragon $W1_{\mathrm{WISE}} - W2_{\mathrm{WISE}}$ overlaid on the KLLR fit and brown dwarf data as a function of numeric spectral type.

List of Figures

Bibliography

[1] F. Kiwy, J. D. Kirkpatrick, A. C. Schneider, A. M. Meisner, J. K. Faherty, M. J. Kuchner, D. Bardalez Gagliuffi, S. L. Casewell, T. P. Bickle, and The Backyard Worlds: Planet 9 Collaboration, "A Search for Late-type Brown Dwarfs in the Euclid Quick Data Release 1," **171,** 108 (2026).

[2] E. J. Honaker and J. E. Gizis, "Simulating Solar Neighborhood Brown Dwarfs. I. The Luminosity Function above and below the Galactic Plane," The Astrophysical Journal **985,** 48 (2025).

[3] W. K. Black and A. Evrard, "Red Dragon: a redshift-evolving Gaussian mixture model for galaxies," Mon. Not. R. Astron. Soc. **516,** 1170–1182 (2022).

[4] A. Farahi, A. E. Evrard, I. McCarthy, D. J. Barnes, and S. T. Kay, "Localized massive halo properties in bahamas and MACSIS simulations: scalings, lognormality, and covariance," Mon. Not. R. Astron. Soc. **478,** 2618–2632 (2018).

[5] A. Farahi, D. Anbajagane, and A. E. Evrard, "KLLR: A Scale-dependent, Multivariate Model Class for Regression Analysis," The Astrophysical Journal **931,** 166 (2022).

[6] W. M. J. Best, T. J. Dupuy, M. C. Liu, A. Sanghi, R. J. Siverd, and Z. Zhang, "The UltracoolSheet: Photometry, Astrometry, Spectroscopy, and Multiplicity for 4000+ Ultracool Dwarfs and Imaged Exoplanets,", 2025.

[7] J. Maíz Apellániz, I. Negueruela, and J. A. Caballero, "Spectral classification," In *Encyclopedia of Astrophysics, Volume 2*, **2,** 43–84 (2026).

[8] A. J. Cannon and E. C. Pickering, "Spectra of bright southern stars photographed with the 13-inch Boyden telescope as part of the Henry Draper Memorial," Annals of Harvard College Observatory **28,** 129–P.6 (1901).

[9] M. N. Saha, "On a Physical Theory of Stellar Spectra," Proceedings of the Royal Society of London Series A **99,** 135–153 (1921).

[10] J. D. Kirkpatrick, I. N. Reid, J. Liebert, R. M. Cutri, B. Nelson, C. A. Beichman, C. C. Dahn, D. G. Monet, J. E. Gizis, and M. F. Skrutskie, "Dwarfs Cooler than "M": The Definition of Spectral Type "L" Using Discoveries from the 2 Micron All-Sky Survey (2MASS)," **519,** 802–833 (1999).

[11] A. J. Burgasser, T. R. Geballe, S. K. Leggett, J. D. Kirkpatrick, and D. A. Golimowski, "A Unified Near-Infrared Spectral Classification Scheme for T Dwarfs," **637,** 1067–1093 (2006).

[12] A. Farahi, "Joint Estimation of Local Linear Parameters and Kernel Bandwidth in KLLR," In *Proceedings of the 28th International Conference on Artificial Intelligence and Statistics (AISTATS)*, Proceedings of Machine Learning Research 258 (PMLR, Mai Khao, Thailand, 2025).

[13] W. K. Black and A. E. Evrard, "Cosmic Dragons: A Two-Component Mixture Model of COSMOS Galaxies," The Open Journal of Astrophysics 7 (2024).

[14] W. K. Black and A. E. Evrard, "Cosmic dragons: A two-component mixture model of COSMOS galaxies," The Open Journal of Astrophysics 7 (2024).

[15] A. Farahi, D. Anbajagane, and A. Evrard, "KLLR: A scale-dependent, multivariate model class for regression analysis," (2022).

[16] W. M. J. Best, T. J. Dupuy, M. C. Liu, A. Sanghi, R. J. Siverd, and Z. Zhang, "The Ultracool-Sheet: Photometry, Astrometry, Spectroscopy, and Multiplicity for 4000+ Ultracool Dwarfs and Imaged Exoplanets,", 2025.

[17] C. Rodrigo *et al.*, "Photometric segregation of dwarf and giant FGK stars using the SVO Filter Profile Service and photometric tools," **689,** A93 (2024).

[18] C. Rodrigo, E. Solano, and A. Bayo, "SVO Filter Profile Service Version 1.0,", IVOA Working Draft 15 October 2012, 2012.

[19] C. Rodrigo and E. Solano, "The SVO Filter Profile Service," In *XIV.0 Scientific Meeting (virtual) of the Spanish Astronomical Society*, p. 182 (2020).

Index