Computer Assisted Spectroscopic Inspection of Gravitational Lensing Objects (CASIGLO)

Joseph E. Summers,* Joel R. Brownstein,† and Jeff M. Phillips

(Dated: Submitted: April 30, 2023)

Gravitational lensing is an important phenomenon in astrophysics that provides insight into the distribution of matter in the universe. One very effective way to discover lensed obejcts is through spectroscopic detection; despite its significance, traditional methods for detecting and characterizing gravitational lenses spectroscopically are limited and rely heavily on the manual examination of spectra. In this paper, we explore the potential of supervised machine learning in detecting and analyzing gravitational lenses from Sloan Digital Sky Survey (SDSS) spectra. The study is a proof of concept for a neural network trained on synthetic data to reliably recognize 12 different emission lines and their source redshift. This is the first study to demonstrate the potential of machine learning in spectroscopic detection of lensed objects, and the results are promising, but the accuracy must be improved to make the CASIGLO project viable for implementation into the data pipeline for a survey, such as SDSS.

I. INTRODUCTION

Many of the scientific challenges once faced by teams of astronomical data scientists are now being overcome by strategically applying deep learning methods [see for e.g.; Farsal et al. 2018, Salakhutdinov 2014], providing greater insight into the tremendous volume of astronomical data that has been collected over the span of many decades. The understanding of the composition of galaxies and the search for dark matter continues to be considered one of the biggest open questions remaining in our model of the universe [National Academies Press 2021]. Studying gravitational lensing is a great tool for learning about galaxies' initial mass functions (IMFs), galactic evolution, the expansion of the universe, dark matter, and even modified theories of gravity. However, currently in the field of gravitational lensing, finding lenses to study requires a ton of manual inspection, either of images or of galactic spectra.

We leverage TensorFlow [Abadi et al. 2016], a machine learning framework for Python, alongside SDSS's massive dataset of galactic spectra, in the hopes of achieving accurate and efficient automated inspection of galactic spectra. This will not only speed up the process of finding lensed objects to study but also has the possibility of finding lenses overlooked in past surveys. While we will be using SDSS data, the process detailed in this paper should be generalizable to other datasets, as should the weights, biases, and filters in the trained machine-learning model.

The main goals of this project are to use the results of the computer-assisted inspection to identify candidates that would benefit most from follow-up space telescope imaging and to add refinement to the detection algorithms to minimize false positives. We have also applied physics as much as possible inside the model to limit the deep learning technical debt [Sculley et al. 2015].

Before jumping into the process, let us first discuss some of the important background needed to understand what the CASIGLO project is and why it is designed the way it is. The rest of §1 discusses important background information. §2 will cover the computational methods used in the CASIGLO model. §3 will look at the results of training the model and steps that were taken to improve results. §4 concludes the paper by giving an overview of the process and discussing future work that needs to be done.

A. Machine Learning Background

Machine learning is a field of artificial intelligence that involves training computers to effectively 'learn' from data and make predictions or decisions based on that learning. The field has a much longer history than most would assume, dating back to 1959 with Arthur Samuel and his work at IBM [Samuel 1959]. In essence, the algorithm he created allowed a computer to play the game of checkers. It combined information such as the number of pieces on both sides (differentiating between regular pieces and 'kings') and how close pieces were to becoming 'kings'. Using this information, the computer could calculate the probability of winning in the current state, as well as the future probability after making different sets of moves. While computing power was limited, Samuel continued refining his scoring function until the computer could beat an amateur player by simply choosing the moves that maximized the function. This method, called "minimax", is the basic idea many machine learning algorithms still use today.

Over the decades, machine learning has advanced significantly, driven by the development of more robust algorithms, the availability of large datasets, and the growth of computing power. In recent years, machine learning has become increasingly important in various fields, in-

* E-mail: joe.summers@utah.edu
† E-mail: joelbrownstein@physics.utah.edu

cluding healthcare, finance, transportation, and more. For example, machine learning has benefited tasks such as object recognition and image classification in computer vision. The applications of machine learning are not limited to computer vision, however. Machine learning can predict stock prices, detect fraud, and improve risk management systems in finance. In healthcare, machine learning is used for drug discovery [Koh et al. 2021], diagnosing diseases, and analyzing medical images. The now (in)famous "ChatGPT", a natural language processing chatbot, was also built on these principles.

The importance of machine learning lies in its ability to automate increasingly complex tasks as our understanding of the field deepens. By learning from large datasets, machine learning algorithms can identify patterns, make accurate predictions, and provide valuable insights that would be difficult to obtain using traditional methods; for example, Srinivasan et al. [2022] at Argonne National Lab have automated the process of discovering possible new materials, accelerating the process of material discovery. Other branches of physics are already using machine learning to their advantage; our goal is to do the same. Machine learning can also discover new knowledge and relationships in data that may have gone unnoticed by human experts; in this proof of study concept, we use a specific type of machine learning, called "supervised learning", in hopes of doing just that.

Supervised machine learning is a type of artificial intelligence that involves training a model to make predictions or classifications based on a set of labeled data. In this approach, one provides the model with input data and its corresponding outputs (or labels) for each data point. The model then learns to generalize from this training data and make predictions on new data the model has not seen previously.

The use of supervised machine learning has the potential to revolutionize the field of gravitational lensing by allowing for the automated detection and characterization of lenses from large datasets. A well-trained AI could detect many of the lensed objects overlooked by experts, significantly increasing the number of lensed objects available to be studied. Increasing the amount of data could provide us with a more complete picture of the distribution of matter in the universe and advance our understanding of the evolution of galaxies and even the nature of dark matter. However, first, let us cover some astronomy and lensing background so we can understand how to apply machine learning.

B. Astronomy / Spectra Background

Just like a prism separates white light into the rainbow of colors in the electromagnetic spectrum, astronomers use a device called a spectrograph to split up the light from a galaxy into its component 'colors', giving us a graph that shows the intensity of light emitted by the galaxy at different wavelengths. The resulting spectrum can reveal a lot of important information about the galaxy's composition (temperature, motion, etc).

Galaxy spectra typically show a continuous spectrum, which is a smooth curve that represents the intensity of light emitted by the galaxy at all wavelengths. This is caused by the thermal radiation of stars and gas in the galaxy. However, galaxy spectra also show distinct spectral lines, which are sharp peaks or valleys at specific wavelengths that correspond to the emission or absorption of light by atoms or molecules in the galaxy.

When an atom or ion in a galaxy transitions from an "excited", higher energy state to a lower energy state, it emits a photon with a very specific wavelength. The wavelengths of the emission lines are determined by the energy difference between the two energy levels involved in the transition, which in turn depends on the chemical element producing the line. So different chemicals produce different sets of emission lines, and these same sets of lines can be seen in galaxy spectra. By measuring the differences in shape and width of these lines, we can deduce information about the temperature, density, and velocity of the gases and objects in the galaxy. We can also use these emission lines to measure the distance to a galaxy.

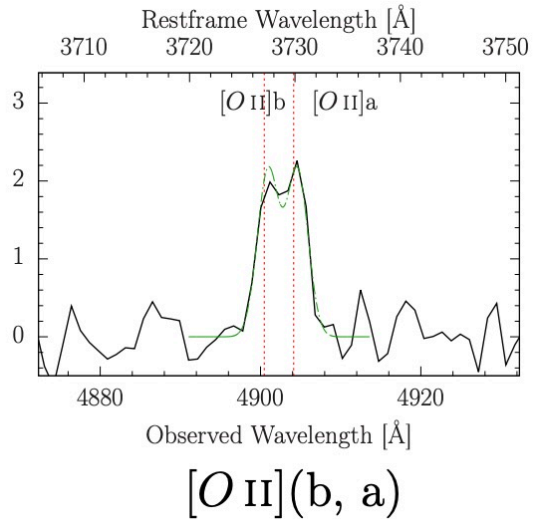

FIG. 1. Two emission lines created by O_2 detected in the MaNGA Survey using the BOSS spectrograph (discussed in §1.D) [Talbot et al. 2022]. They appear almost as a single line due to the detectors sensitivity and the lines being relatively close to one another.

However, as emission line photons travel across the universe from a galaxy to our detectors, the expansion of the universe stretches out the wavelength of the photon and pushes the emission line to a longer, redder wavelength, moving the position that the lines that appear in spectra. The ratio of the shift in wavelength to its emission wavelength is called the redshift, and it is directly related to its distance from Earth (Higher redshift = longer distance).

The most commonly used spectral lines for redshift measurements are the Balmer series of hydrogen emission lines, which occur at specific wavelengths in the visible part of the spectrum, but there are 12 standard emission lines that are commonly detectable in galaxy spectra shown in Table 1.

TABLE I. Detectable Emission-line. The listed emission lines were used to detect background galaxy candidates. Column one lists the name of each emission line. Column two lists the wavelength in a vacuum of a restframe. Column three lists the maximum redshift each emission line can be detected by the BOSS spectrograph.[Bolton et al. 2012]

Emission Line (1)	Restframe Wavelength [Å] (2)	$z_{\max}$ (3)
[O II]b	3727.09	1.78
[O II]a	3729.88	1.78
Hδ	4102.89	1.52
Hγ	4341.68	1.38
Hβ	4862.68	1.13
[O III]b	4960.30	1.09
[O III]a	5008.24	1.07
[N II]b	6549.86	0.58
Hα	6564.61	0.58
[N II]a	6585.27	0.57
[S II]b	6718.29	0.54
[S II]a	6732.68	0.54

The science behind spectral emission lines and photon redshift allows us to learn many things about stars, nebulae, and galaxies, however, it can only tell us so much. Simply knowing the chemical makeup can not tell us any information on the structure of the galaxy, such as its shape or mass, both of which are needed to study dark matter halos. To get this kind of information, we need another form of observation: how the galaxy itself affects the light around it.

C. Gravitational Lensing Background

Gravitational lensing is a phenomenon that occurs when light from a distant source passes close to a massive object, such as a galaxy or a galaxy cluster. The gravitational pull of the massive object bends the path of the light, causing it to be deflected and magnified, creating either multiple images of a point source or bending an extended source into a "banana-shaped" arc. The bending of light due to gravity was first predicted by Albert Einstein in his theory of general relativity in 1919. Sir Arthur Eddington confirmed it the same year during a total solar eclipse in Russia amid the First World War [Dyson et al. 1920]. However, Einstein never really pursued the idea further. It was not until the 1930s that Rudi W. Mandl persuaded Einstein to consider the implications of gravitational lensing seriously, but even then, he only considered the lensing due to stars; Fritz Zwicky was the first person to discuss the possibility of lensing due to galaxies (which were called Nebulae at the time), noting that the likelihood of measuring a "double image" of other galaxies would be much more significant than that for stars due to their larger diameters [Zwicky 1937].

Gravitational lensing has since become a significant tool in astrophysics, providing a unique way to study the distribution of matter in the universe. The lensing effect is sensitive to the mass distribution of the lensing object, which includes both the visible and dark matter components, making it a powerful tool for exploring the dark matter content of galaxies and galaxy clusters or testing theories of modified gravity (e.g., MOND Sanders and McGaugh [2002]). Scientists used lensing calculations on the Bullet Cluster, one of the most famous pieces of evidence for dark matter, to determine the total mass distribution in the cluster apart from the baryonic mass, which is detected through x-rays [Clowe et al. 2006]. Lenses, however, also *focus* light; this causes lensed objects to appear brighter in the sky, allowing us to see more distant objects than normally possible.

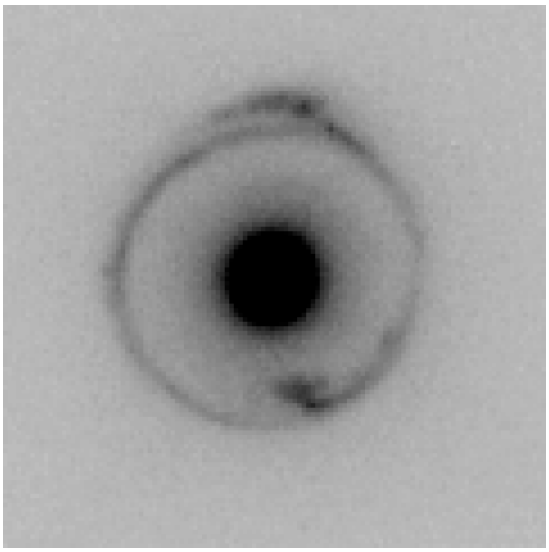

FIG. 2. An example of a gravitational lens imaged by the Hubble Space Telescope for the SLACS survey. The dark circle in the middle is the "lens" galaxy, which is bending the light of a galaxy behind it into a ring.[Brownstein et al. 2012]

Standard methods for detecting and characterizing gravitational lenses rely on the manual identification of characteristic features either in galaxy spectra or images. However, these methods are time-consuming and require expert knowledge, limiting the number of lenses that can be identified and characterized. Spectral methods have also restricted positive detections to rely on emission lines with relatively large signal-to-noise ratios (SNRs) to limit false positives, further limiting the number of lenses we detect. With a properly trained model, this SNR limit

could no longer be an issue, meaning that many lensed objects previously missed by experts, could be detected.

Today, gravitational lenses are being used to constrain specific *types* of dark matter. A recent paper published in Nature by Amruth et al. [2023] uses gravitational lensing to provide evidence that axions (very light bosons) are a stronger candidate for dark matter than weakly-interacting massive particles (WIMPs) due to their wave-like properties.

D. Past Surveys (SLACS, SWELLS, BELLS, SILO)

Now that we have sufficiently discussed the history of gravitational lensing let us look into more recent searches for gravitational lenses. From least to most recent, the most prominent projects are SLACS, SWELLS, BELLS, and SILO, all of which use the Sloan Digital Sky Survey's database of galaxy spectra for object selection. We will also discuss the MaNGA project, which focused specifically on nearby galaxies but did not focus on lensed objects. Firstly, the Sloan Lens ACS (SLACS) Survey [Bolton et al. 2005] used the SDSS database of galaxy spectra and searched for spectra that contained at least three emission lines at a singular redshift, z_FG, much greater than the target redshift. The survey was designed to study galaxy evolution and galactic structure formation, such as spiral arms, by detecting early-type galaxies at large redshifts. (Early-type galaxies have little to no disk component and low star-formation rates). The paper also presented a method for subtracting the foreground galaxy from the observed spectra to more easily identify features like emission lines. After selecting a subset of spectra with the highest likelihood of being lensed objects, the SLACS surveys followed up with imaging using the Advanced Camera for Surveys (ACS) aboard the Hubble Space Telescope, allowing for more detailed modeling to be done. This paper, published in 2006, found 19 newly discovered lensed objects. The survey eventually published 13 separate papers by the end of 2017 and discovered over 100 strongly lensed galaxies, making it one of the most significant collections of lensed galaxies to date.

On the other hand, in 2011, the Sloan WFC Edge-on Late-type Lens Survey (SWELLS) looked specifically for late-type, edge-on galaxies to observe both their lensing and kinematic components to calculate the relative contributions of baryons to dark matter in the centers of these galaxies [Treu et al. 2011]. However, to do this, one needs to separate the disk and halo structures of galaxies, which requires an independent measurement of the mass-to-light ratio of the galaxy. Often, this ratio is assumed to be the constant $\gamma_\odot = 5133$ kg/W, which is taken from the sun. However, as much of the mass in the galaxy lies in its dark matter, this value would be inaccurate for anything on a galactic scale. This survey used two sub-samples of data: one from the SLACS survey and the second found by the SWELLS team using a selection algorithm. The SWELLS algorithm identified over 200 lens candidates, and the SLACS survey had 85 lenses and 13 candidates. All 298 of these images were manually inspected to pick out only edge-on, late-type galaxies, and this resulted in only 27 lenses used in the survey.

Building off both the techniques of the Baryon Oscillation Spectroscopic Survey (BOSS), which allowed for the detection of higher-redshift emission lines, and the SLACS survey's success, the BOSS Emission-line Lens Survey (BELLS) spectroscopically built a catalog of 25 definite and 11 high-likelihood strong galaxy-galaxy lenses with redshifts between 0.4 and 0.7 [Brownstein et al. 2012]. These 44 lens candidates were found in only the first six months of data from an approximately five-year-long project. While much of the data was narrowed down by automated selection procedures, 1303 multi-line hits, and 741 single-line hits (2 thousand spectra total) had to be manually inspected to confirm the presence/likelihood of it being a lensed object. Shu et al. [2016] later expanded the survey in the fourth project paper by finding 21 more lens candidates identified through Lyman-Alpha ($Ly-\alpha$) emissions and confirming them with Hubble Space Telescope imaging.

While not specifically a lensing survey, the MaNGA (Mapping Nearby Galaxies at Apache Point Observatory) survey is also relevant to the CASIGLO project. The MaNGA survey was a significant component of SDSS-IV, obtaining not just a single spectrum for a galaxy, but as many as 127 spectra across different locations on the galaxy, depending on its angular size [Bundy et al. 2014]. MaNGA alone did not directly identify any lensed galaxies. However, a project that began a few years later during SDSS-IV combed MaNGA spectra to pick out strong lenses.

That project was the *Spectroscopic Identification of Lensing Objects* project by [Talbot et al. 2020]. They published a complete catalog of lensed objects from both the MaNGA and eBOSS surveys, which contains "838 likely, 448 probable, and 265 possible strong lens candidates within $\approx$ 2 million galaxy spectra", all of which were found spectroscopically. That is a total of $1,551$ possible lensed objects. If every candidate turned out to be a lens, then out of 2 million spectra, $\ll 1\%$ of the data would be an object of interest; again, many of these spectra had to be visually inspected to confirm their value for follow-up imaging by HST or their inclusion in the SILO database.

The chances of a gravitational lens occurring in the first place are already meager; adding in the additional reduction in the likelihood due to instrumental factors making it even more challenging to detect spectroscopically creates a very daunting task for anyone – even professionals – to attempt manually. This problem is precisely what we have set out to solve.

II. COMPUTATIONAL METHODS

Now that we have discussed who and what works we are building off of, let us look at where we started

A. Model Input & Pre-Process Questions

The SDSS spectrograph, used in all the prominent surveys discussed in the previous section, gives continuous spectra between $3,600–10,300\mathring{A}$ with $4,563$ individual flux read-outs. To anticipate training on SDSS spectra, we will use an input array of length $4,563$.

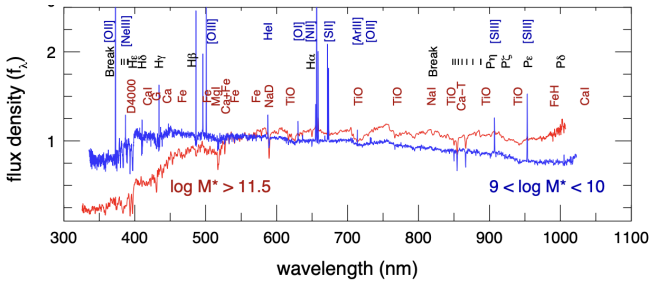

FIG. 3. An example of two galaxy spectra taken during BOSS survey [Dawson et al. 2012]. Many emission line features are labeled for both spectra.

However, when attempting to train a neural network to detect faint features from a galaxy in the background (the source), we want to remove these prominent features from the foreground (the lens), so that the model can focus on the features it wants to predict. Luckily, the SLACS pipeline already created a way of looking for lensed objects spectroscopically: by subtracting a best-fit model for the foreground galaxy, one can effectively see *just the background* of the galaxy, which has much fewer features and makes searching for emission lines much more manageable.

The size of the spectrum's array is not changed in the foreground-subtraction process and results in essentially pure noise arrays unless objects are present in the background. This is ideal for training a machine learning model, as features that *are* present stand out much more.

So, the original plan for our training dataset was to use SILO's database of lensed galaxies and modeled foregrounds to attain enough background spectra to train the CASIGLO model to reliably detect any of the 12 emission lines detailed in Table 1. However, we soon realized that there were a few key issues that we had to address first:

1. Low Lensing Probability / Emission Line Rarity
2. Dealing with sky-lines and other anomalous features
3. Deciding the details of the model's output

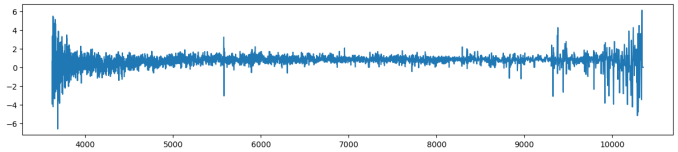

FIG. 4. An example of a foreground-subtracted spectra from SILO's database without any emission lines. Flux is on the y-axis and the x-axis is the wavelength Angstroms ($\mathring{A}$). While a few features are present – such as the flares on both sides and a spike near $5,600\mathring{A}$– with the foreground subtracted, the spectra is almost pure noise.

Firstly, as discussed in Arneson et al. [2012], the probability of a galaxy spectra containing a lensed system is less than 1%. So, while SDSS has millions of spectra available, SILO only has about 900 spectra available that would be useful for training our model. However, spectra vary wildly due to the 2^{12} different possible combinations of emission lines, redshifts ranging between 0.01 and 1 in the SDSS survey, and the vast differences in the general shape of galaxy spectra due to chemical makeups and types of galaxies. Thus, to properly train a machine learning model to recognize the presence of a lensed object in a galaxy's spectra, we need much more data than will likely ever be available. Instead, we shall generate an extensive database of *synthetic* spectra.

It may seem like a bad idea to train a neural network on data from a different source than the one on which the model will make predictions. However, as shown Kaisti et al. [2021], models trained using synthetic data can produce results that outperform models trained on real data. This has also been shown to be true for models that take higher-dimensional inputs such as images (see Seib et al. [2020]). Using ideas from both papers referenced above, the plan for the CASIGLO project's database became clear: Using SDSS's plethora of foreground-subtracted spectra that are incredibly unlikely to have any emission lines, we can then add synthetic emission lines at any redshift and SNR we desire.

To build this database, we use all the spectra in the SDSS database, subtract the foreground and save the backgrounds. As this is done, the foreground redshifts of each spectrum are recorded so that the distribution of redshifts can be analyzed. Since the initial goal is to test this model on spectra in the MaNGA database, which only contains galaxies with $0.03 < z < 0.10$, we want most of our data to fall into that range.

After the backgrounds are saved into their respective bins, we can start adding in our synthetic emission lines. But this is were our second problem starts to stand out. In figure 4, no emission lines are present, yet there still appears to be a line at $\approx 5,600\mathring{A}$. If used as training data, this could be confused for an emission line.

These features are our second problem, and they come from the fact that the SDSS/BOSS spectrographs are located on Earth. This means that the instruments have

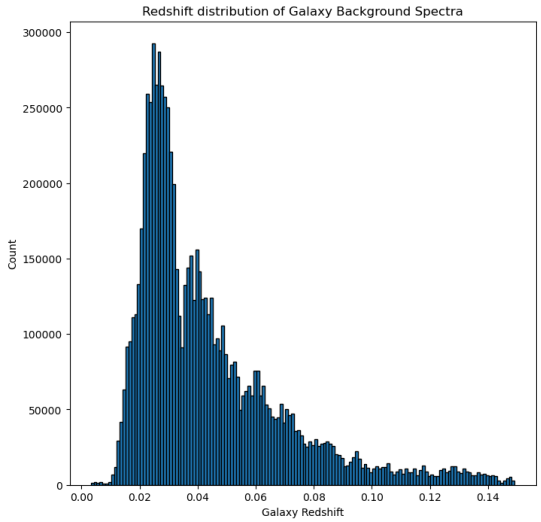

FIG. 5. The redshift (z) distribution of foreground galaxies used to generate synthetic spectra for the CASIGLO training dataset. The majority of spectra had a foreground redshift < 0.10, matching the MaNGA dataset.

to look through Earth's atmosphere, resulting in a signal, *most* of which is from our sky, leaving only $\approx 1\%$ of the signal corresponding to the galaxy and background. How one handles the sky determines what the leftover spectra look like[1]. The atmosphere is removed from these spectra so we can focus on the galaxies; however, this subtraction is not always ideal. Often, "sky-lines" appear that look like emission lines but have a much larger SNR (seen in Figure 4). If any sky-lines are present in the synthetic spectra we generate, it could lead to slower training and worse performance. To deal with these extra spikes in the spectra, we slide a rolling window across our spectra of width $200\mathring{A}$ and find the standard deviation, σ, of flux readings across it. If any flux value in that window has more than 3σ is removed and replaced with a random value between $\pm\sigma$. This ensures that no sky-lines or other anomalous features are present in the spectra before adding the details we want our model to recognize.

Lastly, before we can start training, we have to decide the output of our model (i.e., what our predictions will tell us). If you were training a model to detect handwriting based on images, its output could be a set of probabilities corresponding to different letters and numbers, with the highest probability being the model's "guess". Alternatively, if you wanted a model that could guess a person's age based on health information, you could have a single output node that returns a number in a specific range (say 1-100 years old). For CASIGLO, we want our model to recognize emission lines behind the target galaxy, so it needs to know what redshift the emission lines are at. Thus, we will have an output node that tries to approximate the redshift of emission lines. It would also be beneficial to know precisely which emission lines are present (of which there are 12) and how strong they appear (their signal-to-noise ratio). Therefore, we will design our output to be 13 nodes, one for each emission line's SNR and one more for the approximate redshift value of said emission lines.

B. Synthetic Data Generation

With a clear plan of data structure, data cleaning, and our inputs and outputs, let us now look into the details of how we generated our data and the decisions we made to make them physically accurate. As already discussed, we use real backgrounds to generate our synthetic data, and anomalous features are removed and replaced with random, non-significant noise. First, SDSS's `speczall.fits` file is used to get unique ID numbers[2] for each spectrum in the database. These IDs are then used to retrieve the raw spectra, as well as the corresponding foreground model. The foreground is then subtracted, leaving only the background spectra which are binned into folders based on the foreground galaxy's redshift value and saved for later use. As discussed previously, during this process the foreground redshift of each spectrum is saved into an array so that we can see the redshift distribution shown in Figure 5.

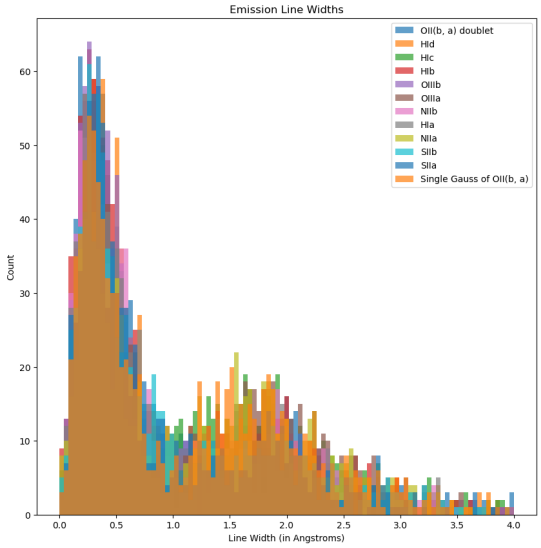

FIG. 6. The distribution of standard deviations for all emission lines detected by SILO. Most emission lines detected have $\sigma = 0.4\mathring{A}$ and the rest group around $1.7\mathring{A}$. These two values, along with their respective probabilities, are used in generating synthetic emission lines.

[1] This is one issue that would affect how well the CASIGLO model could be applied to other spectrographs.

[2] plate number, and Integral Field Unit (IFU) design

Once all the backgrounds are saved, we can start adding in our synthetic emission lines, which can be approximated as Gaussians. However, in order to create physically accurate emission lines we need to use width parameters (σ) that match the widths of observed emission lines. When plotting the width of all emission lines detected by SILO, we can see two main values stand out: one around 0.4 Angstroms and another around 1.7 Angstroms. These 2 widths show up with relative probabilities of about 6-to-1.

So, these two most common values were used when generating synthetic emission lines and were generated in their relative probabilities of 6/7 and 1/7 respectively. After deciding the width, the SNR of each emission line must also be set. The SNR of any given emission line needs to be set and should be randomized to give variation to the model data to aid training. This value can be arbitrarily large, but in order to match our data, we need to limit this value to more reasonable values. The SILO project limited their minimum requirement for emission line candidates to a SNR of $\tilde{4}$, however many emission lines are likely much less significant than that, and few may be much more significant. Thus, when generating our random SNRs, we want to generate their values with a distribution that peaks around 4, has a higher likelihood of being less than 4, and drops off quickly past about 5. After analyzing many different distributions, we settled on a Moyal distribution with $\mu = 4$ and $\sigma = 1.5$.

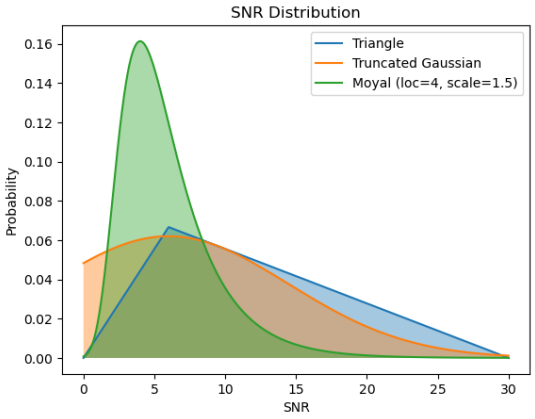

FIG. 7. Different SNR distributions that were considered for synthetic data generation. We settled on the Moyal distribution (shown in green) due to its lower probability of generating large SNRs.

We now have random generation of the width and SNR of our synthetic Gaussian emission lines ready. There is only one more step before we can begin generating realistic synthetic spectra: redshifting the emission lines to the proper wavelengths.

Since the definition of redshift is simply the difference in wavelength over the rest-frame wavelength, we can do some simple algebra to get from a redshift value to the new location of our emission line, since we know the rest frame wavelength.

$$\begin{aligned} z &= \frac{\lambda_{obs} - \lambda_{rest}}{\lambda_{rest}} \\ &= \frac{\lambda_{obs}}{\lambda_{rest}} - 1 \\ z + 1 &= \frac{\lambda_{obs}}{\lambda_{rest}} \\ \lambda_{obs} &= \lambda_{rest}(z+1) \end{aligned} \tag{1}$$

With this simple relation, we only need a redshift value for each spectra we generate and we can easily shift each of our emission lines to a simulated redshift. But, just as we wanted randomization in our SNRs to help the training process, we also want to randomize our redshifts. Since the source objects will always be behind the lens itself, we only want to generate redshift values *larger* than the foreground redshift of the background we are using. *How much* larger is a more difficult question, however.

Looking at SILO's detected lenses and separating them into the SLACS, BELLS, and MaNGA surveys, we can see how the redshift of each source object relates to the redshift of its lensing counterpart.

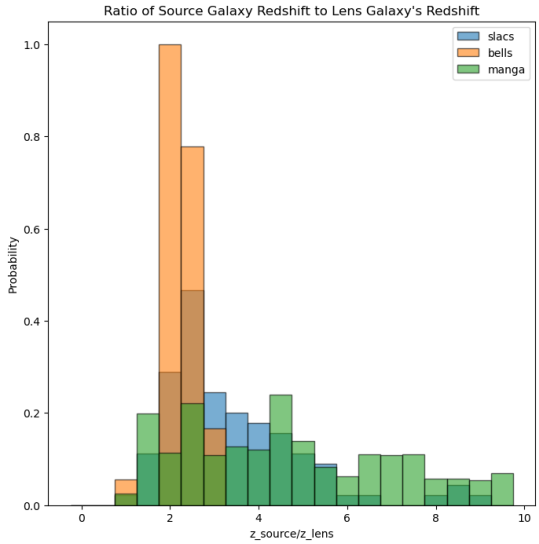

FIG. 8. The ratio of source redshift to lens redshift. For BELLS we can see a very clear peak at 2, indicating that the source objects are very likely to be at twice the redshift as the lens. Moving to MaNGA, this distribution really flattens out to almost be uniform between 2-10. This is due to all the lens galaxies in the MaNGA survey being at low redshifts.

All the preparations are now complete and all that is left to do is generate spectra. We will iterate over all foreground redshift bins and generate 20 spectra for each combination of emission lines. These spectra are then saved as a parquet file to be loaded later and fed into our model for training. This is quite a hefty calculation though, as we have 146 redshift bins spanning

$z = 0.00370$ to $z = 0.14970$. With 12 emission lines, taking every combination results in $2^{12} = 4,096$ combinations and totaling in 598,016 HDF5 files. With 20 spectra in each file, a total of 11,960,320 spectra are being generated. Being written in python without using Numba or any other python compiler packages, generating even a single file takes a few seconds. Generating nearly 12 million files would be unreasonable without parellelization. We use 64 cores and divide our 146 redshift bins evenly amongst each core. Doing so lets us generate all 12 million files in approximately 4 days.

The final spectra generated can be compared to real spectra and results are almost indiscernible by eye.

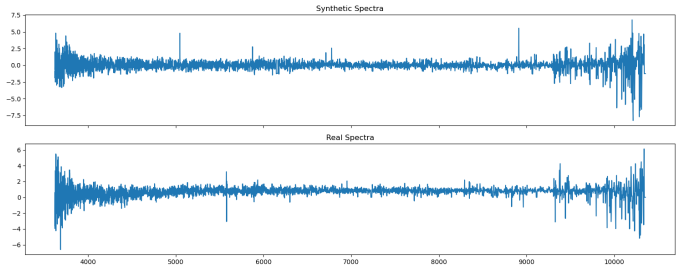

FIG. 9. Top: One of the synthetic spectra generated for the CASIGLO database with synthetic emission lines present. Bottom: A real foreground-subtracted spectra from a lens galaxy found in the MaNGA survey without emission lines. (Plate-IFU: 8606-6102). While the presence of emission lines identifies one from the other, the noise and general appearances have very similar forms, making them ideal for training.

Now that we have a database of realistic synthetic spectra with enough variation to train our model, and have our inputs and outputs planned, we just have to build the structure of our model before we can begin training.

C. Expand on Machine Learning Process

To start building our model architecture we employ the use of TensorFlow [Abadi et al. 2015]: An open-source platform that allows users to quickly build, train, and evaluate machine learning models and AI. TensorFlow has a python library that makes incorporating a model into your existing python workflow incredibly easy. However, there is not just *one kind* of machine learning, and before discussing what we built, one should first understand why we chose to build our model the way we did.

There are three main learning problems in machine learning:

Supervised Learning:: The model attempts to learn the best mapping between inputs and outputs. This is referred to as "supervised" as the output that is supplied is compared to the model's predictions and the mapping is corrected based on the differences.

[Ex:]Estimating the price of a house based on data from other houses in the same city

Unsupervised Learning:: The model attempts to find relationships from input data. As no outputs are fed into the model, is simply tried to group the data or estimate the data's distribution.

[Ex:]Kernel Density Estimations

Reinforcement Learning:: The model is given a set of operations that it can execute. Supplied with some numerical "reward", the model tries to maximize its reward by changing the order and set of actions that it executes.

[Ex:]An AI that learns to play the game "Snake". The model can move in all 4 cardinal directions and the length of the player is the reward.

The problem CASIGLO is trying to solve is falls into the category of supervised learning as we want to learn the mapping from input spectra to an output of emission lines' SNRs and the source redshift. But supervised learning itself can be broken down into two main situations: Classification & Regression.

In both cases, the model is attempting to predict the labels in whatever form we supply them (as strings, integers, Booleans, etc.). Classification would be predicting a *class label* for our input whereas regression would predict a *numerical value*. As we are hoping to output the SNRs of each emission line and the source redshift, our problem falls into the regression category. Simple supervised learning neural network architecture is composed of a set of input nodes and output nodes, with "hidden layers" in between. Most of the time, these nodes are "fully connected" to the next layer, meaning each node can have an effect on every other node in the next layer. The effect that each node has on a connected node is decided by the connection's weights and the bias of the hidden note: The output of a node in a hidden layer is related to all of its inputs by the following.

$$z = b + \sum_{i=1}^{i=N} a_i w_i$$

Where z is the output, a_i is the input of each connected node (of which, there are N), w_i is the weight of each connection, and b is the bias of the hidden node.

The training process for a neural network simply updates these weights & biases every time a new input-output pair is supplied. How these values are updated is all based on your model's *loss function*. For regression cases, this loss function could be the accuracy, precision, or simply the absolute difference from the expected output. Were we to lower our data dimensionality to one input and one output, one can see that minimizing this

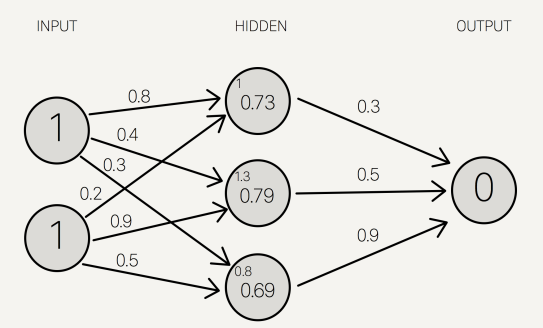

FIG. 10. A simple, fully connected neural network with labeled weights and biases. Weights are listed above each line and biases are the difference between the number in the top left of each hidden node and the node's value.[stack exchange link]

loss function would be equivalent to fitting a single best-fit line to a set of points. A model with 2 inputs and 1 output would be finding a best-fit surface to match the data. With our input of 4,563 flux readings and an output of 13 values, however, we are fitting a much more complicated, high-dimensional hyper-plane to our input spectra and output labels. This is a very difficult problem and trying to fit this hyperplane would require an immense amount of computing power and time. This is where convolutional neural networks can help us. Convolutional neural networks are commonly used in computer vision problems such as object detection and image classification. Convolution is a process that lowers the dimensionality of your data while preserving the relative locations of objects by passing a number of filters across your data. The filters have the same dimensionality as the data and you can have multiple filters in each layer. For example, instead of a classifier using every pixel in an image to try and discern the difference between a human and a cat, it can learn to create two-dimensional filters that can instead group pixels together into objects such as "Head", "Tail", "arms", etc. By looking at the relationships between the few objects the filters detect, the model can learn much faster. We intend to use convolution to pick out peaks in our spectra and more quickly train our model.

Now that we understand the concepts used to build our model, let us discuss the specific architecture of the CASIGLO model and what changes were made during the training process.

III. OUTCOME & CHALLENGES

The architecture of the CASIGLO convolutional neural network was initially based on the work of a previous graduate student in the School of Computing (a special thanks to Salvatore Stone Mele) who was trying to predict just the redshift of spectra. The model was only accurate to around ± 0.01. Since redshift values are expected to be scientifically accurate $\pm 10^{-5}$, the model was not feasible and was abandoned. However, the architecture could be easily modified to output 13 values instead of one.

The model consisted of four convolutional layers, each with a successively decreasing number of filters. After the fourth convolutional layer applied its filters, the results were then flattened out into a single set of nodes. These nodes were then output into four fully-connected layers (not convolutional layers). These fully-connected layers decreased in size until the final output was a single node which predicted the redshift. As stated above, we replace this last single node with 13 nodes.

However, just having the layers themselves is not sufficient for effective training. As discussed in Huang et al. [2020], normalization of input data removes statistical difference in magnitudes between features, improving learning results. Thus, we add batch normalization to every layer in our model. Other statistical tools can also be used to help improve the results of training: Both "max pooling" and "Dropout" can reduce prediction errors as well. Max-pooling reduces the size of your layer by passing forward only the maximum value in a patch of some size (specified by the user). Dropout layers make it so that random nodes are ignored during the training process, helping reduce the possibility that nodes become co-dependant on one another. The effect of each of these layers is discussed in Wu and Gu [2015].

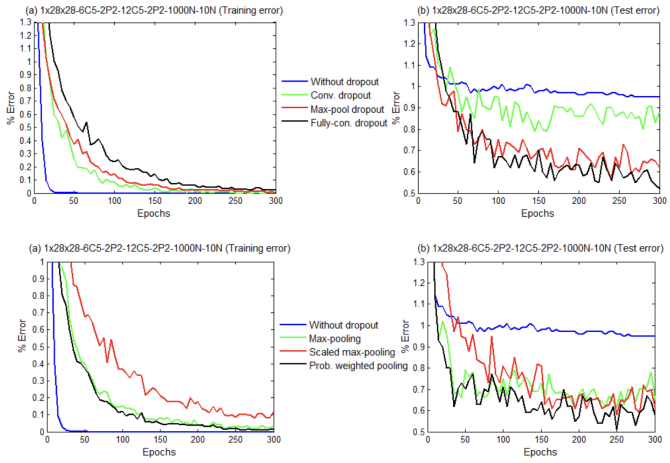

FIG. 11. The impact of dropout and max-pooling layers on a neural networks training and validation stages. The top two plots are for dropout while the bottom plots are for max-pooling. Similarly, the left plots correspond to the training phase while the right plots correspond to the validation (or testing) phases. Notice that the presence of both layers slows down the learning rate during the training stage, but this lower learning rate tends to reduce error faster during the validation stage.

We thus include both max-pooling on all layers and dropout layers just on convolution. We do not include

dropout in the convolutional layers to avoid dropping out emission lines. With our model architecture completed, we can view how many parameters are being trained by calling TensorFlow's `summary` function. Doing so, we see the following parameter counts.

$$\begin{aligned} \text{Total parameters}: &\ 4,558,073 \\ \text{Trainable parameters}: &\ 4,551,169 \\ \text{Non-trainable parameters}: &\ 6,904 \end{aligned}$$

Meaning that training our model requires fitting over 4.5 million parameters.

All that is left to do now, before we can start training is to decide a loss function to minimize. For our first run, we naively used accuracy, as well as keeping track of the mean-squared-error along the way.

With a model created, a database of synthetic spectra at the ready, and a loss function determined, we could begin training the model. When first beginning training, the model was instructed to simply iterate over every file in every sub-folder of every redshift bin of data that was generated. Each file contained 20 spectra and those spectra were each passed into the model only once. That means that the weights, biases, and convolutional filters were only updated once every time a spectra was input.

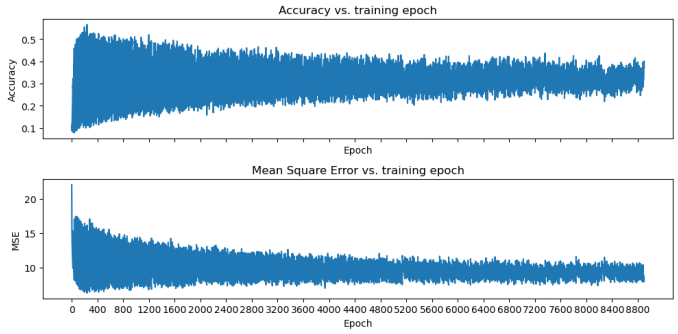

FIG. 12. The results of our first attempt to train the CASIGLO neural network. The top plot shows the model's prediction accuracy as a function of the training epoch (the number of times the model updates its weights, biases, and filters). The top plot shows the mean-squared error (MSE) as a function of epoch. The Accuracy seems to be converging to ~ 0.35, meaning that the model made the correct predictions of SNR and redshift 35% of the time. The MSE appears to converge around 10, meaning that the SNRs and redshift had an average error of ~ 3.16.

While not unpromising, these results have a fatal flaw. The accuracy loss function calculates how often the model predicts the correct class result (both true positives & true negatives), meaning it is reserved for *classification* problems. Therefore, our results were fundamentally flawed and should not be taken to have any meaning. The MSE, however, *is* used for regression problems. But the results converged to such a high value in relation to the redshift and SNR that the model is useless.

Realizing these errors, the model's loss function was changed to measure the mean absolute error (MAE) and the precision was monitored as well. As the precision was not set as our loss function, only a quantity to monitor, the fact that it is a classification metric had no effect on how the model was learning.

We soon realized a few more mistakes made in the training process. In order for a model to learn more effectively, the inputs supplied to the model should be varied in their outputs. This would ensure that the model is not simply learning a single set of emission lines for every file passed in, then immediately un-learning due to a different set of emission lines being passed. Initially, simply using all 20 spectra in a file meant that all the same emission lines were present and the only variation was in the SNRs. The process was changed to grab 20 random files and use all 20 spectra in each to train the model[3]. Doing so ensures that each batch has a mix of emission lines present. Since SNRs are still being randomized, each batch should sufficiently represent our dataset.

Restarting the training process, our results appeared pretty similar with only slightly better errors.

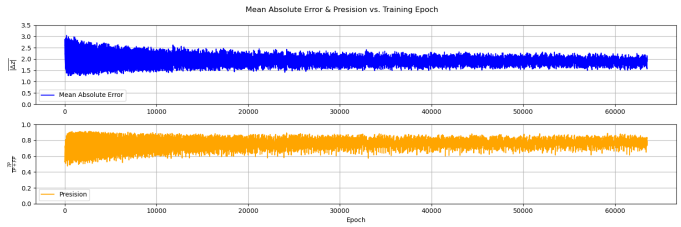

FIG. 13. The results of a later training run after fixing errors and slightly streamlining the training process. The top plot shows the MAE of SNRs and redshift as a function of epoch. The bottom plot shows the precision. While precision seems to converge around 80%, this is only due to the model learning to predict all zero values. This specific training was done before the batch size was increased, so the training speed per batch is not visible as it is in the next figure.

While results seemed better due to the precision converging to almost 80%, when looking at the actual predictions, it became clear that the seemingly high performance was due to the model simply predicting zero for all emission lines no matter what spectra it was predicting on. This state of predicting all zeros is due to the loss function getting stuck in one of its local minima; this particular local minimum happens to be quite large due

[3]The "Batch size" is the number of files times the number of spectra in each file; 400 in this case.

to the number of emission lines and the low likelihood of emission lines being present.[4]

At this point in the project, many more training attempts were made, and not much was progressing. The training was slow, and the results kept tending toward all zero predictions. The decision was made to simplify the model in order to both speed up the training process as well as more easily identify what was going wrong.

The model was simplified to only two convolutional layers – one with large filters and one with small filters – followed by only a single dense layer before outputting the 13 values. Another training attempt reveals that not only is a larger batch size helping our training results, but the smaller model seems to be producing more accurate predictions.

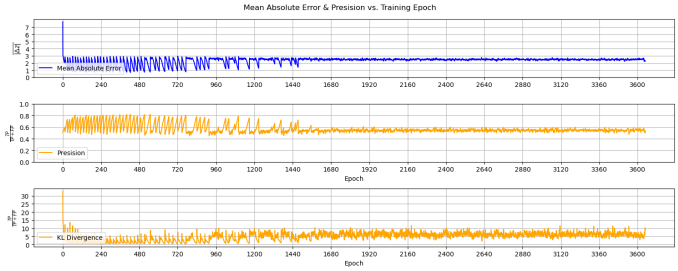

FIG. 14. The results of a training run with the simplified convolution and dense layers. For this training, another metric was also monitored: the KL divergence. The KL divergence is a measure of how close two distributions are, so this metric can clue us in as to how often the model is guessing the correct set of emission lines.
Notice that eventually the model seems to stop learning anything and gets stuck at a higher error value than what it achieves through training on each batch. This is the result of over-fitting the model.

The model continues to get stuck in the local minima of all-zero predictions, and training results alone did not clue us in to the issue. Our next step to identify the issue was something that should have been done much sooner: plotting the results of the loss and other metrics from the *validation phase* and comparing it to the training phase.

When a batch of files is passed into a model for training, it is standard practice to reserve around 20% of your data for testing or validating how well the model learned from the other 80% of the data. The model first tries to learn what to predict in the training phase, then the validation phase (also called the testing phase) is just for making predictions; weights and biases are not updated during this phase. However, after each epoch, the model should still make better predictions on the validation set due to the training phase updates to the model parameters.

[4]This issue persisted for many different iterations of our model and is still a major sticking point for this project.

When doing so for both the original CASIGLO model as well as the smaller model, we see that while the MAE for training decreases nicely over each batch, the MAE during validation either stays constant or tends to increase. This is another sign of over-fitting.

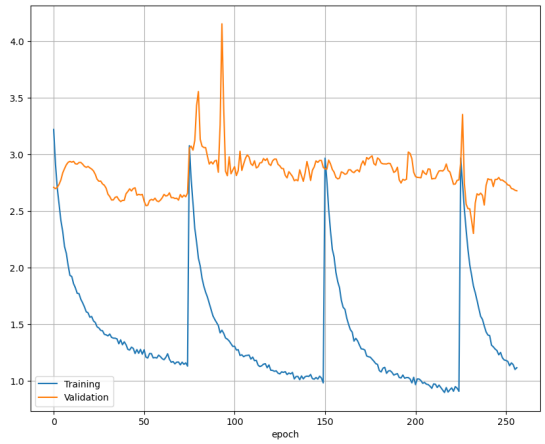

FIG. 15. Mean-Absolute-Error (MAE) vs epoch for both the training phase (shown in blue) and validation phase (shown in orange). Four batches of files are present and are identified by the sharp increase in error between them due to a new set of spectra being introduced.
While the training error decreases at a promising rate, validation error stays mostly the same over each batch, indicating that the model is only learning the training data, not learning the problem in general. This is another sign of over-fitting

It was becoming clear that trying to fine-tune the loss function, batch size, learning rate, and other basic model parameters was not going to fix the issue. Some extra steps are needed to dissuade the model from falling into the trap of over-fitting. This is when *regularization* was added.

Regularization is a process that alters your loss function by adding extra terms and can help reduce the training impact of noise in your data and fight over-fitting. There are two main types of regularization used in machine learning: Lasso (L1) regression and Ridge (L2) regression. Ridge (L2) regression works by adding a penalization term to the loss function and effectively puts constraints on the weights so that no single node can contribute too much to the output (e.g. no zero values). Lasso (L1) regression is very similar but it *does* allow weights to reach zero and can thus help with feature selection (i.e. choosing which input nodes/set of nodes contributes most to the output).

At the time of learning about regression, the end of the semester was rapidly approaching and little time was left to make improvements. I tested a few combinations of both L1 and L2 regression parameters and was able to find that using just L1 regularization (L2=0) with a value $L1 \approx 0.04$ was enough to allow the validation error to start following the training error.

Other combinations of L1 & L2 regularization that

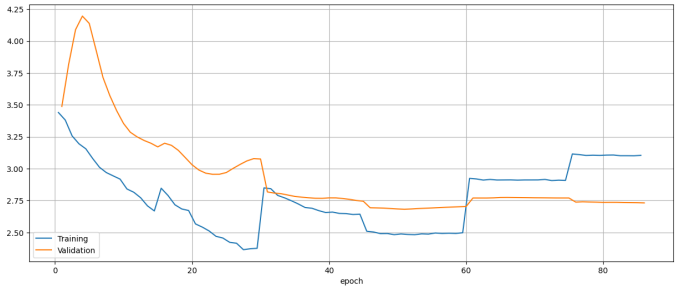

FIG. 16. Training and validation MAE values plotted against epoch. Unlike earlier the validation line begins to track the training line, implying that the model is learning to actually predict emission lines, not just learning the data set. Unfortunately, the model stopped learning during training after just two batches.

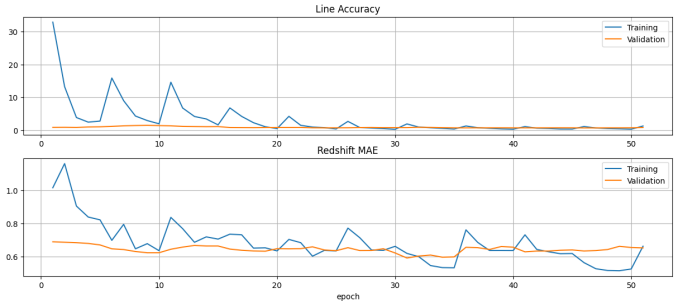

FIG. 17. Results of the last training ran on the newly updated model. The top plot shows the accuracy of predicted emission line presence and the bottom plot shows the MAE in redshift predictions.
This model seems to be performing the best out of all models attempted so far. Decreasing peaks illustrate the potential success of this architecture.

were tested did not produce any noticeable progress.

A few weeks before research progress came to a halt, a complete redesign of the model was created with some promising results, however, not enough time was left to explore the extent of the new model's capabilities.

Both models previously used were completely linear, with the output of each layer being passed into the next. The idea for the model redesign was that the redshift estimate should have a very large impact on which input nodes correspond to relative output nodes as the redshift changes where emission lines are located. In order to have the model account for this, a non-linear model was created. The input spectra are first passed through two convolutional layers which are then flattened into a dense layer which is then used to make a single prediction for the redshift. However, unlike the linear models, the input spectra are also passed into a parallel branch consisting of two of convolutional layers which are again flattened. This parallel flattened layer is then combined with the initial estimate for the redshift, re-joining the parallel branches. Then the model has one more dens layer before returning two outputs: one for the *presence* of each emission line, and another for an updated redshift estimate.

To help ease the model into the emission line problem, instead of jumping right into the deep end and detecting the SNR of each emission line, we take a step back and just try to have the model predict whether an emission line is present or not. Emission line output nodes return a probability that an emission line is present, and the redshift output continues to return a decimal value. L1 and L2 regression parameters are still present along with batch normalization, max-pooling, and dropout layers.

With this updated model, we can see that training results not only learn but continue to build upon what was learned in previous epochs. This can be seen in the successively decreasing spikes at the start of each epoch. However, the validation loss continues to stay stagnant.

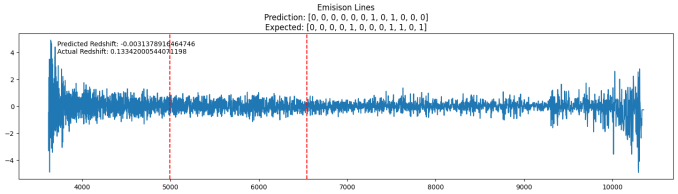

FIG. 18. Prediction results from the last training run. The input spectrum is plotted in blue and predicted locations of emission lines are present.
Predicted vs. expected redshift is shown as a plot annotation. Predicted vs expected emission lines are shown in the plot's title; each 0 or 1 is representative of the respective emission line detailed in Table 1. $H\alpha$ was correctly detected.

IV. CONCLUSION

The CASIGLO project still has a long way to go. These results end my work, and while the model is not reliable and cannot be implemented into SDSS workflows, they should form a great backbone for anyone else willing to continue the project.

We have learned a great deal about what it takes to apply machine learning to a process such as this, and huge steps have been taken to achieve the goal of automated detection of lensed galaxies. However, there is still a lot of work that has to be done. The model should be properly trained to greater than 90% accuracy in emission line detection and be redshift accurate to $\sim 10^{-5}$. This would ensure that if follow-up is done using Hubble Space Telescope imaging or any other method, that time, materials, and funding is not wasted on false positives.

With the mistakes made along the way, updated knowledge of the machine learning process, and hindsight, I will use the next section to help guide anyone who deems this project worthy of pursuing further.

A. Future Steps

The interconnected model was the single biggest progress jump over the span of my work on the project. I believe that a similar structure is key to the success of this project; however, there are a few improvements that I would have made myself had time permitted.

Firstly, the disk space requirements of the backgrounds is quite large at the moment. If no statistical difference is found between backgrounds in different redshift bins, PCA could be used to generate physically accurate backgrounds on the fly and would require very little disk space.

Also, the redshift estimates of the new, interconnected model can still predict negative values. This is an obvious error that needs to be fixed either with regularization parameters or a change of data type in the model output layer. I believe that fixing this would improve the model's overall performance, both with redshift estimates as well as emission line predictions.

Another key element that could be introduced to the redshift predictions is by using SDSS's pre-existing measure of the foreground redshift. Setting this value as a floor for the initial redshift prediction could drastically reduce the error for the model's output as it would give the model a "starting point" of sorts.

Next, regularization parameters should be fine-tuned to produce the best combination of penalization and feature selection. This could be done through multiple-regression with the training speed as the dependant variable and L1 & L2 parameters as the independent variables. By training a few batches on different combinations of L1 and L2 and using the validation learning rate as a predictor variable, fine-tuning these parameters could be a quick task if training can be executed fast enough.

This is where my last suggestion comes in; training, both on CPU and GPU, can be quite cumbersome. Due to the way spectra were saved, reading the files, combining their spectra into the proper format for TensorFlow to accept, and finally passing it into the model slows down the training process immensely. Files were saved in parquet format as discussed previously but were done so using the Pandas Python package [pandas development team 2020]. Pandas works great for saving and loading files, however, it does not allow generators to handle the files. This became an issue during the training process as a generator would be a much faster and easier way to shuffle, format, and feed in data, but it had to be done manually instead. Switching away from Pandas to another DataFrame handler such as h5py could allow for the use of generators, creating faster training. This would even open up the possibility to train the model on multiple batches at once through distributed training/multi-threading.

I hope this work and the suggestions provided create a solid ground for future researchers to excel in the advancement of lensed object detection. All my code is available here and data used to generate spectra are available through SDSS databases.

V. ACKNOWLEDGEMENTS

A big thanks to Joel Brownstein, Kyle Dawson, and Jeff Phillips for being a part of this project and for offering support whenever I needed it. I could not have come this far alone.

Funding for the Sloan Digital Sky Survey IV has been provided by the Alfred P. Sloan Foundation, the U.S.

SDSS-IV acknowledges support and resources from the Center for High-Performance Computing at the University of Utah.

The SDSS website is www.sdss.org.

SDSS-IV is managed by the Astrophysical Research Consortium for the Participating Institutions of the SDSS Collaboration including the Brazilian Participation Group, the Carnegie Institution for Science, Carnegie Mellon University, Center for Astrophysics — Harvard & Smithsonian, the Chilean Participation Group, the French Participation Group, Instituto de Astrofísica de Canarias, The Johns Hopkins University, Kavli Institute for the Physics and Mathematics of the Universe (IPMU) / University of Tokyo, the Korean Participation Group, Lawrence Berkeley National Laboratory, Leibniz Institut für Astrophysik Potsdam (AIP), Max-Planck-Institut für Astronomie (MPIA Heidelberg), Max-Planck-Institut für Astrophysik (MPA Garching), Max-Planck-Institut für Extraterrestrische Physik (MPE), National Astronomical Observatories of China, New Mexico State University, New York University, University of Notre Dame, Observatário Nacional / MCTI, The Ohio State University, Pennsylvania State University, Shanghai Astronomical Observatory, United Kingdom Participation Group, Universidad Nacional Autónoma de México, University of Arizona, University of Colorado Boulder, University of Oxford, University of Portsmouth, University of Utah, University of Virginia, University of Washington, University of Wisconsin, Vanderbilt University, and Yale University. Department of Energy Office of Science, and the Participating Institutions.

References

[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[2] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, and X. Zheng. TensorFlow: A system for large-scale machine learning. *arXiv e-prints*, page arXiv:1605.08695, May 2016.

[3] A. Amruth, T. Broadhurst, J. Lim, M. Oguri, G. F. Smoot, J. M. Diego, E. Leung, R. Emami, J. Li, T. Chiueh, H.-Y. Schive, M. C. H. Yeung, and S. K. Li. Anomalies in gravitational-lensed images revealing einstein rings modulated by wavelike dark matter. 2023.

[4] R. A. Arneson, J. R. Brownstein, and A. S. Bolton. Quantifying the biases of spectroscopically selected gravitational lenses. 2012.

[5] A. S. Bolton, S. Burles, L. V. E. Koopmans, T. Treu, and L. A. Moustakas. The sloan lens acs survey. i. a large spectroscopically selected sample of massive early-type lens galaxies. 2005.

[6] A. S. Bolton, D. J. Schlegel, E. Aubourg, S. Bailey, V. Bhardwaj, J. R. Brownstein, S. Burles, Y.-M. Chen, K. Dawson, D. J. Eisenstein, J. E. Gunn, G. R. Knapp, C. P. Loomis, R. H. Lupton, C. Maraston, D. Muna, A. D. Myers, M. D. Olmstead, N. Padmanabhan, I. Paris, W. J. Percival, P. Petitjean, C. M. Rockosi, N. P. Ross, D. P. Schneider, Y. Shu, M. A. Strauss, D. Thomas, C. A. Tremonti, D. A. Wake, B. A. Weaver, and W. M. Wood-Vasey. Spectral classification and redshift measurement for the sdss-iii baryon oscillation spectroscopic survey. 2012.

[7] J. R. Brownstein, A. S. Bolton, D. J. Schlegel, D. J. Eisenstein, C. S. Kochanek, N. Connolly, C. Maraston, P. Pandey, S. Seitz, D. A. Wake, W. M. Wood-Vasey, J. Brinkmann, D. P. Schneider, and B. A. Weaver. The BOSS Emission-Line Lens Survey (BELLS). I. A Large Spectroscopically Selected Sample of Lens Galaxies at Redshift ~0.5. , 744(1):41, Jan. 2012.

[8] K. Bundy, M. A. Bershady, D. R. Law, R. Yan, N. Drory, N. MacDonald, D. A. Wake, B. Cherinka, J. R. Sánchez-Gallego, A.-M. Weijmans, D. Thomas, C. Tremonti, K. Masters, L. Coccato, A. M. Diamond-Stanic, A. Aragón-Salamanca, V. Avila-Reese, C. Badenes, J. Falcón-Barroso,

F. Belfiore, D. Bizyaev, G. A. Blanc, J. Bland-Hawthorn, M. R. Blanton, J. R. Brownstein, N. Byler, M. Cappellari, C. Conroy, A. A. Dutton, E. Emsellem, J. Etherington, P. M. Frinchaboy, H. Fu, J. E. Gunn, P. Harding, E. J. Johnston, G. Kauffmann, K. Kinemuchi, M. A. Klaene, J. H. Knapen, A. Leauthaud, C. Li, L. Lin, R. Maiolino, V. Malanushenko, E. Malanushenko, S. Mao, C. Maraston, R. M. McDermid, M. R. Merrifield, R. C. Nichol, D. Oravetz, K. Pan, J. K. Parejko, S. F. Sanchez, D. Schlegel, A. Simmons, O. Steele, M. Steinmetz, K. Thanjavur, B. A. Thompson, J. L. Tinker, R. C. E. van den Bosch, K. B. Westfall, D. Wilkinson, S. Wright, T. Xiao, and K. Zhang. Overview of the sdss-iv manga survey: Mapping nearby galaxies at apache point observatory. 2014.

[9] D. Clowe, M. Bradac, A. H. Gonzalez, M. Markevitch, S. W. Randall, C. Jones, and D. Zaritsky. A direct empirical proof of the existence of dark matter. 2006.

[10] K. S. Dawson, D. J. Schlegel, C. P. Ahn, S. F. Anderson, Éric Aubourg, S. Bailey, R. H. Barkhouser, J. E. Bautista, A. Beifiori, A. A. Berlind, V. Bhardwaj, D. Bizyaev, C. H. Blake, M. R. Blanton, M. Blomqvist, A. S. Bolton, A. Borde, J. Bovy, W. N. Brandt, H. Brewington, J. Brinkmann, P. J. Brown, J. R. Brownstein, K. Bundy, N. G. Busca, W. Carithers, A. R. Carnero, M. A. Carr, Y. Chen, J. Comparat, N. Connolly, F. Cope, R. A. C. Croft, A. J. Cuesta, L. N. da Costa, J. R. A. Davenport, T. Delubac, R. de Putter, S. Dhital, A. Ealet, G. L. Ebelke, D. J. Eisenstein, S. Escoffier, X. Fan, N. F. Ak, H. Finley, A. Font-Ribera, R. Génova-Santos, J. E. Gunn, H. Guo, D. Haggard, P. B. Hall, J.-C. Hamilton, B. Harris, D. W. Harris, S. Ho, D. W. Hogg, D. Holder, K. Honscheid, J. Huehnerhoff, B. Jordan, W. P. Jordan, G. Kauffmann, E. A. Kazin, D. Kirkby, M. A. Klaene, J.-P. Kneib, J.-M. L. Goff, K.-G. Lee, D. C. Long, C. P. Loomis, B. Lundgren, R. H. Lupton, M. A. G. Maia, M. Makler, E. Malanushenko, V. Malanushenko, R. Mandelbaum, M. Manera, C. Maraston, D. Margala, K. L. Masters, C. K. McBride, P. McDonald, I. D. McGreer, R. McMahon, O. Mena, J. Miralda-Escudé, A. D. Montero-Dorta, F. Montesano, D. Muna, A. D. Myers, T. Naugle, R. C. Nichol, P. Noterdaeme, S. E. Nuza, M. D. Olmstead, A. Oravetz, D. J. Oravetz, R. Owen, N. Padmanabhan, N. Palanque-Delabrouille, K. Pan, J. K. Parejko, I. Pâris, W. J. Percival, I. Pérez-Fournon, I. Pérez-Ràfols, P. Petitjean, R. Pfaffenberger, J. Pforr, M. M. Pieri, F. Prada, A. M. Price-Whelan, M. J. Raddick, R. Rebolo, J. Rich, G. T. Richards, C. M. Rockosi, N. A. Roe, A. J. Ross, N. P. Ross, G. Rossi, J. A. Rubiño-Martin, L. Samushia, A. G. Sánchez, C. Sayres, S. J. Schmidt, D. P. Schneider, C. G. Scóccola, H.-J. Seo, A. Shelden, E. Sheldon, Y. Shen, Y. Shu, A. Slosar, S. A. Smee, S. A. Snedden, F. Stauffer, O. Steele, M. A. Strauss, A. Streblyanska, N. Suzuki, M. E. C. Swanson, T. Tal, M. Tanaka, D. Thomas, J. L. Tinker, R. Tojeiro, C. A. Tremonti, M. V. Magana, L. Verde, M. Viel, D. A. Wake, M. Watson, B. A. Weaver, D. H. Weinberg, B. J. Weiner, A. A. West, M. White, W. M. Wood-Vasey,

C. Yeche, I. Zehavi, G.-B. Zhao, and Z. Zheng. The baryon oscillation spectroscopic survey of sdss-iii. 2012.

[11] F. W. Dyson, A. S. Eddington, and C. Davidson. IX. a determination of the deflection of light by the sun's gravitational field, from observations made at the total eclipse of may 29, 1919. *Philos. Trans. R. Soc. Lond.*, 220(571-581):291–333, Jan. 1920.

[12] W. Farsal, S. Anter, and M. Ramdani. Deep learning: An overview. In *Proceedings of the 12th International Conference on Intelligent Systems: Theories and Applications*, SITA'18, New York, NY, USA, 2018. Association for Computing Machinery.

[13] P. W. Koh, S. Sagawa, H. Marklund, S. M. Xie, M. Zhang, A. Balsubramani, W. Hu, M. Yasunaga, R. L. Phillips, I. Gao, T. Lee, E. David, I. Stavness, W. Guo, B. Earnshaw, I. Haque, S. M. Beery, J. Leskovec, A. Kundaje, E. Pierson, S. Levine, C. Finn, and P. Liang. Wilds: A benchmark of in-the-wild distribution shifts. In M. Meila and T. Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 5637–5664. PMLR, 18–24 Jul 2021.

[14] National Academies Press. *Pathways to Discovery in Astronomy and Astrophysics for the 2020s*. Washington, DC, 2021.

[15] T. pandas development team. pandas-dev/pandas: Pandas, Feb. 2020.

[16] R. Salakhutdinov. Deep learning. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, page 1973, New York, NY, USA, 2014. Association for Computing Machinery.

[17] A. L. Samuel. Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development*, 1959.

[18] R. H. Sanders and S. S. McGaugh. Modified newtonian dynamics as an alternative to dark matter. 2002.

[19] D. Sculley, G. Holt, D. Golovin, E. Davydov, T. Phillips, D. Ebner, V. Chaudhary, M. Young, J.-F. Crespo, and D. Dennison. Hidden technical debt in machine learning systems. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015.

[20] V. Seib, B. Lange, and S. Wirtz. Mixing real and synthetic data to enhance neural network training – a review of current approaches, 2020.

[21] Y. Shu, A. S. Bolton, S. Mao, C. S. Kochanek, I. Pérez-Fournon, M. Oguri, A. D. Montero-Dorta, M. A. Cornachione, R. Marques-Chaves, Z. Zheng, J. R. Brownstein, and B. Ménard. The boss emission-line lens survey. iv. : Smooth lens models for the bells gallery sample. 2016.

[22] S. Srinivasan, R. Batra, D. Luo, T. Loeffler, S. Manna, H. Chan, L. Yang, W. Yang, J. Wen, P. Darancet, and S. K R S Sankaranarayanan. Machine learning the metastable phase diagram of covalently bonded carbon. *Nat. Commun.*, 13(1):3251, June 2022.

[23] M. S. Talbot, J. R. Brownstein, K. S. Dawson, J.-P. Kneib, and J. Bautista. The completed sdss-iv extended baryon oscillation spectroscopic survey: A catalogue of strong galaxy-galaxy lens candidates. 2020.

[24] M. S. Talbot, J. R. Brownstein, J. Neumann, D. Thomas, C. Maraston, and N. Drory. Sdss-iv manga: A catalogue of spectroscopically detected strong galaxy-galaxy lens candidates. 2022.

[25] T. Treu, A. A. Dutton, M. W. Auger, P. J. Marshall, A. S. Bolton, B. J. Brewer, D. Koo, and L. V. E. Koopmans. The swells survey. i. a large spectroscopically selected sample of edge-on late-type lens galaxies. 2011.

[26] H. Wu and X. Gu. Towards dropout training for convolutional neural networks. 2015.

[27] F. Zwicky. Nebulae as gravitational lenses. *Phys. Rev.*, 51(4):290–290, Feb. 1937.