Cycle-StarNet: Bridging the Gap between Theory and Data by Leveraging Large Data Sets

Teaghan O’Briain, Yuan-Sen Ting (丁原森), Sébastien Fabbro, Kwang M. Yi, Kim Venn, and Spencer Bialek

1 Department of Physics and Astronomy, University of Victoria, Victoria, BC V9E 2E7, Canada; obriain@uvic.ca
2 Institute for Advanced Study, Princeton, NJ 08540, USA
3 Department of Astrophysical Sciences, Princeton University, Princeton, NJ 08540, USA
4 Observatories of the Carnegie Institution of Washington, 813 Santa Barbara Street, Pasadena, CA 91101, USA
5 Research School of Astronomy & Astrophysics, Australian National University, Cotter Road, Weston, ACT 2611, Australia
6 National Research Council Herzberg Astronomy & Astrophysics, 4071 West Saanich Road, Victoria, BC, Canada
7 Department of Computer Science, University of British Columbia, Vancouver, BC V6T 1Z4, Canada

Received 2020 July 6; revised 2020 September 7; accepted 2020 November 12; published 2021 January 18

Abstract

Advancements in stellar spectroscopy data acquisition have made it necessary to accomplish similar improvements in efficient data analysis techniques. Current automated methods for analyzing spectra are either (a) data driven, which requires prior knowledge of stellar parameters and elemental abundances, or (b) based on theoretical synthetic models that are susceptible to the gap between theory and practice. In this study, we present a hybrid generative domain-adaptation method that turns simulated stellar spectra into realistic spectra by applying unsupervised learning to large spectroscopic surveys. We apply our technique to the APOGEE H-band spectra at $R = 22,500$ and the Kurucz synthetic models. As a proof of concept, two case studies are presented. The first is the calibration of synthetic data to become consistent with observations. To accomplish this, synthetic models are morphed into spectra that resemble observations, thereby reducing the gap between theory and observations. Fitting the observed spectra shows an improved average $R^2$ reduced from 1.97 to 1.22, along with a mean residual reduced from 0.16 to $-0.01$ in normalized flux. The second case study is the identification of the elemental source of missing spectral lines in the synthetic modeling. A mock data set is used to show that absorption lines can be recovered when they are absent in one of the domains. This method can be applied to other fields that use large data sets and are currently limited by modeling accuracy. The code used in this study is made publicly available on GitHub (https://github.com/teaghan/Cycle_SN).

Unified Astronomy Thesaurus concepts: Computational methods (1965); Neural networks (1933); Spectroscopy (1558); Spectral line identification (2073)

1. Introduction

Using theoretical models to decipher stellar spectra in terms of stellar properties is difficult. It requires detailed modeling of the photospheric surface layers, understanding a myriad of atomic and plasma processes, and calculating the radiative transfer through complex stellar atmospheres. Nevertheless, stellar spectra are one of the most important data sources that we have to understand stars.

Classically, methods that compare theoretical stellar spectra directly to observations have been used for decades. The comparison is typically performed manually (e.g., Sneden et al. 2008; Aoki et al. 2013; Venn et al. 2020), but massively multiplexed and higher resolution stellar spectroscopic surveys have been launched in the past few years (Gilmore et al. 2012; Dalton et al. 2014; Buder et al. 2018; Holtzman et al. 2018), where the data analysis approaches have started to become more automatic (Yanny et al. 2009; Ting et al. 2017, 2019; Fabbro et al. 2018; Zhang et al. 2019; S. Bialek et al. 2020, in preparation; Guiglion et al. 2020).

Unfortunately, methods that employ synthetic spectral models typically suffer from the differences between theory and practice referred to as the “synthetic gap.” The synthetic gap can be induced by both theoretical systematics and instrumental factors. In terms of theoretical systematics, many assumptions are made during spectral modeling; for example, stellar atmospheres are often assumed to be one-dimensional, in hydrostatic equilibrium, and in local thermodynamic equilibrium (LTE). These assumptions often fail, causing systematic offsets between theoretical spectra and actual measurements. Instrumental factors can further introduce signatures that are not reproduced by theoretical modeling, for example, telluric lines imposed by the Earth’s atmosphere and the image formation on the detector due to the light path through the telescope. Modern spectroscopic pipelines (e.g., Ballester et al. 2000; Martioli et al. 2012) often need to make accurate assumptions about the instrumental signatures in order to reproduce realistic and consistent spectra. These assumptions can contribute to the synthetic gap and limit the capabilities of the methods that use synthetic spectra directly.

Previous work has been done to attempt to overcome the synthetic gap between theory and observations. For instance, efforts have been made toward incorporating non-LTE and 3D hydrodynamic effects in the model atmospheres (e.g., Amarsi 2015; Amarsi et al. 2016; Kovalev et al. 2019). Other methods have been proposed to isolate spectral regions, only using spectral regions where the astrophysics and instrumental effects are better understood (e.g., Jahandar et al. 2017; Ting et al. 2019). On top of that, Coelho et al. (2020) compared the use of synthetic versus empirical libraries and studied the impact of spectral coverage versus missing physics. Furthermore, some have attempted to reduce this gap by augmenting the synthetic data through sampling and adding noise to make the spectra look more realistic (e.g., S. Bialek et al. 2020, in preparation). Despite these efforts, there is still a large amount of room for improvement.
In contrast, methods that use the empirical observed data directly for the spectral templates have been proposed (Prugnien et al. 2011; Ness et al. 2015; Sharma et al. 2015; Ting et al. 2017; Fabbro et al. 2018; Leung & Bovy 2019; Xiang et al. 2019). These “data-driven” methods skip the direct use of synthetic spectra, but depend on a priori knowledge of stellar parameters and elemental abundances for a large training set. Accordingly, these methods learn a model that directly translates spectra into physical characteristics, or vice versa. Due to the increasing number of amassed spectra, training these data-driven methods has become more tangible.

However, the stellar labels determined from data-driven models are limited in accuracy. By training on previously derived stellar parameters and abundances, the errors in the original spectra and their parameters are propagated through to the final results. Therefore, the systematic errors are difficult to uncover when comparing the predictions to the original pipeline results. Naturally, this raises doubts on whether the model is learning actual physics or simply inheriting the biases from the original pipeline. Lastly, building data-driven models requires high-quality data for training, often in the form of spectra with high signal-to-noise ratio. In most cases, collecting a sufficient number of high-quality empirical templates that span the full range of stellar parameter and elemental abundance ranges can be difficult, if not impractical.

In this study, we propose a novel solution, CYCLE-STARNET, that overcomes the synthetic gap without suffering from the shortcomings of data-driven methods. At its core, CYCLE-STARNET is trained to learn to transform data from the synthetic domain to the observed spectral domain. To accomplish this, the network leverages recent advancements in machine-learning methodologies, specifically, domain adaptation (Liu et al. 2017; Zhu et al. 2017). Furthermore, the method can work directly with noisy training spectra and implicitly denoise them. In essence, the domain adaptation is accomplished by forcing the two domains to share an abstract representation, which we use to exploit the connections found between the domains.

CYCLE-STARNET shows how unlabeled observed data can be used to calibrate synthetic spectra to become consistent with the observations, while also correcting for any improperly modeled spectral features. We refer to this as autocalibration. In other words, our approach alleviates the need to know stellar labels before training (as is the case for data-driven methods) and, at the same time, bridges the synthetic gap that plagues existing spectral analysis methods, which use synthetic spectra directly.

This paper is organized as follows. In Section 2, we detail the critical insights of CYCLE-STARNET. In Section 3, the technical details of CYCLE-STARNET are described. In Section 4, CYCLE-STARNET is used in two case studies: correcting the systematics in theoretical models through domain adaptation, and identifying missing spectral features in the synthetic models. In Section 5, the advantages of CYCLE-STARNET compared to other spectral analysis techniques are discussed, as well as its limitations. We conclude the study in Section 6.

2. Motivation and Overview of CYCLE-STARNET

The main goal of CYCLE-STARNET is to learn the connection between two sets of unlabeled spectra and how to “morph” from one domain to another. In other words—adopting terminology from the area of domain adaptation in machine learning—CYCLE-STARNET can transfer spectral models from the synthetic domain to the observed domain, and by doing so, it corrects for the systematic errors in the synthetic modeling.

Domain adaptation has a long history in the field of machine learning. Additionally, with the advent of the generative adversarial network (GAN), GAN-based domain adaptation has seen numerous successes. For example, Zhu et al. (2017) built a domain transfer model that was capable of translating photoimages from daytime settings to nighttime settings (and vice versa), while keeping the content in the photos the same. In our work, we apply a similar method to translate between two spectral domains. As an analogy, one can think of the robust spectral features as the content in the photoimages (which both domains share), whereas the day/night “contexts” are the systematics that we want to correct for.

Liu et al. (2017) accomplished their domain adaptation through the combined use of variational autoencoders and adversarial learning (i.e., the use of GANs). In short, autoencoders accomplish the task of representation learning by creating a bottleneck through the use of a neural network. This forms a compressed representation of the samples, which can make it easier to find correlations within the data set. Adversarial learning, however, is typically accomplished through the use of a generator and a discriminator; the generator aims to produce realistic samples, while the discriminator acts as the teacher to help the generator achieve this goal. The method presented in Liu et al. (2017) utilizes autoencoders to transfer samples from one domain to the other, while the adversarial learning ensures that the transferred samples are accurate.

An important aspect of these domain-adaptation methods (including ours) is that the data are unpaired across the two domains. In other words, we are provided with data from each domain (e.g., the synthetic and observed spectra), but we do not have samples from one domain that correspond to samples in the opposite domain. This is unlike the other proposed data-driven models (e.g., Ness et al. 2015; Ting et al. 2017; Xiang et al. 2019), which assume that the corresponding stellar labels of the observed spectra are known beforehand, and then we learn the label–spectra translation through supervised learning. Instead, for unsupervised domain adaptation, this mapping can be built with unpaired data. Not requiring paired samples is ideal for future uses in stellar spectroscopy because, when obtaining newly observed spectra, no prior assumptions or knowledge are needed regarding the stellar parameters and elemental abundances of the stars.

We will elaborate on the technical details of CYCLE-STARNET in Section 3; here, we focus on the insights. The critical insight is that, while the samples in the two domains are unlabeled, they should occupy the same regions of the stellar label space if they represent the same underlying objects. Therefore, a common representation can be constructed for these spectra. In CYCLE-STARNET, this common representation is formed in what we call the shared latent space.

In order to achieve this goal, CYCLE-STARNET has two mission-critical components: (1) a method to extract the underlying hidden abstractions (or latent space) of each domain, essentially performing a dimensionality reduction of...
the dimensionality reduction component. Common space is created via autoencoders, where \( E \) represents the encoder (or the dimensionality reduction component) and \( D \) is the decoder (or the spectral generation component).

In order to provide a physically interpretable connection between the synthetic and observed spectra, the first step is to create a link between the physical parameters and synthetic spectra. In CYCLE-STARNET, we propose to accomplish this by including a synthetic emulator (see Figure 2). The synthetic emulator maps stellar labels, \( \gamma \), to the synthetic spectra, \( \lambda_{\text{synth}} \). Subsequently, the synthetic spectra can then be morphed into the observed spectra through the use of the decoder networks, \( D_{\text{obs}} \). These processes allow for spectra to be reconstructed within the same domain or transferred from one domain to another. Combining the emulator with the domain transfer network makes it possible to trace the latent representations back to the physical stellar labels. The individual components discussed in Section 3 are annotated in the plot.

3. Methodology

In this section, we lay out the details of CYCLE-STARNET, which consists of two key components. The first component is a spectral emulator that produces synthetic spectra from a set of physical parameters. The second component is an unsupervised domain-adaptation algorithm that transforms spectra from one spectral domain to another. We detail the two components in Sections 3.1 and 3.2, respectively.

3.1. Spectral Emulator

In order to provide a physically interpretable connection between the synthetic and observed spectra, the first step is to create a link between the physical parameters and synthetic spectra. In CYCLE-STARNET, we propose to accomplish this by including a synthetic emulator (see Figure 2). The synthetic emulator maps stellar labels, \( \gamma \), to the synthetic spectra, \( \lambda_{\text{synth}} \). Subsequently, the synthetic spectra can then be morphed into the observed spectra through the use of the decoder networks, \( D_{\text{obs}} \). These processes allow for spectra to be reconstructed within the same domain or transferred from one domain to another. Combining the emulator with the domain transfer network makes it possible to trace the latent representations back to the physical stellar labels. The individual components discussed in Section 3 are annotated in the plot.

3.2. Domain Adaptation

Adaptation does not involve the training of the emulator. In CYCLE-STARNET, we adopt a domain-adaptation algorithm that transforms spectra from one domain to another. This concept is visualized in Figure 1. On top of this framework, inspired by Gonzalez-Garcia et al. (2018), we apply a twist that tightens our method: we leave room for a nonshared latent space. In more detail, the synthetic domain is only able to show variations that are a subset of what happens in real data. For example, the observed data might have instrumental variations that are not seen in the synthetic spectra. As a result, without leaving room for a nonshared latent space, it is impossible for the framework to model phenomena such as instrumental variations. To mitigate this issue, we provide the framework with the freedom to use additional latent variables, the split latent variables, which are unique to the observed domain and not connected to the synthetic domain.

In this study, we adopt an improved version by considering a single large MLP to emulate the synthetic spectra as a whole. We found that using a single network facilitates the extraction of information from adjacent pixels, thus improving the emulation precision. This version of THE PAYNE can be found in the latest THE PAYNE GitHub: https://github.com/tingyuansen/The_Payne.
3.2. Domain Adaptation

The primary motivation for CYCLE-STARNET is the capacity to bridge the gap between observed and synthetic data sets using unsupervised methods, that is, by learning hidden aspects of the two domains automatically, without any human supervision. For ease in explanation, we will refer to the domain of synthetic data as the “synthetic domain” and the observed data as the “observed domain.” We further denote data from these two domains as $\mathcal{X}_{\text{synth}}$ and $\mathcal{X}_{\text{obs}}$, respectively. Note, however, that CYCLE-STARNET is highly flexible and can be used to transform between any two domains, and it is not restricted to transferring between synthetic and real observations. For example, one could also transfer between spectra obtained from different spectrographs, or between two different synthetic models, which we leave as future applications.

3.2.1. Architecture

In order to learn the mapping from one set of spectra to the other, we propose a method based on the UNsupervised Image-to-image Translation Networks (UNIT, Liu et al. 2017). This framework is constructed out of encoders and decoders, as shown in Figure 2. One unique aspect of this architectural design is that we impose a hierarchical structure on the encoder–decoder pairs, which is used to facilitate training. Namely, two low-level encoder–decoder pairs ($E_{\text{synth}}$–$D_{\text{synth}}$ and $E_{\text{obs}}$–$D_{\text{obs}}$) are implemented that are dedicated to capturing domain-specific changes within spectra. We then utilize a high-level encoder–decoder pair ($E_{\text{sh}}$–$D_{\text{sh}}$) that further abstracts the latent space; this pair is shared between both domains. Lastly, to implement the split latent space, we also make use of a second high-level encoder–decoder pair ($E_{\text{sp}}$–$D_{\text{sp}}$) for the observed domain. The shared and split latent spaces will be explained more thoroughly next. Note that the dimensionality of the data gets reduced as it goes through the encoders—thus the data is abstracted—while the opposite happens when it goes through the decoders.

This architectural design allows low-level and data-related information to be learned by the domain-specific encoder–decoders. At the same time, the shared encoder–decoder learns to abstract high-level physical concepts that are shared among both domains. While this design choice was motivated by the architecture used for UNIT (Liu et al. 2017), our framework is unique because of the use of the split encoder–decoder, which was found to provide improved convergence. Another key difference between CYCLE-STARNET and UNIT is that we implement deterministic autoencoders instead of variational autoencoders in this study. The exact architectural designs of these networks are outlined in Appendix A.

3.2.2. The Latent Space

**Shared Latent Space**—Using the framework outlined above, both data sets are encoded independently down to a shared representation. We denote this shared latent space as $\mathcal{Z}_{\text{sh}}$ and illustrate this concept in Figure 1. Once the shared space is created, a synthetic spectrum can be mapped to the latent space, and then the latent representation can be used to create the corresponding spectrum in the observed domain. Furthermore, as a byproduct of this shared latent space, we create four domain mappings, which can be written as (1) $\mathcal{X}_{\text{synth}} \rightarrow \mathcal{X}_{\text{obs}}$, (2) $\mathcal{X}_{\text{obs}} \rightarrow \mathcal{X}_{\text{sh}}$, (3) $\mathcal{X}_{\text{sh}} \rightarrow \mathcal{X}_{\text{obs}}$, and (4) $\mathcal{X}_{\text{sh}} \rightarrow \mathcal{X}_{\text{synth}}$. Of these, the translation of $\mathcal{X}_{\text{obs}} \rightarrow \mathcal{X}_{\text{synth}}$ (the mapping of imperfect synthetic models to the observed domain) is the main focus of this paper.

**Split Latent Space**—While the transfer of spectra between domains requires a shared latent space, not all of the information in one domain is present in the other. This is especially true when relating synthetic and observed spectra. For example, two stars in the observed domain may have the same set of stellar labels, yet vary in other characteristics. In other words, the two domains could have “shared” characteristics but also unique properties of their own. The latent representation of the synthetic spectra can therefore be considered as a subset of the observed spectra, since the latter have other defining features from instrumental profiles (e.g., line spread function) and observational affects (e.g., telluric features) that might not be fully captured in the synthetic models. To account for this, in addition to the shared latent variables (that are common to both domains), we introduce a split latent space, $\mathcal{Z}_{\text{sp}}$, which represents information that is unique to the observed domain.

In order to ensure that the shared and split information is contained in the appropriate latent space, adversarial learning is applied to the latent space, which is explained in Section 3.2.5. In short, if the unique observed information is encoded into the shared latent space, the adversarial training process will identify this, and the encoded information will be adjusted.

3.2.3. Training the Network

In order for the network to be able to translate from one domain to the other, it should be able to perform—simultaneously—the following tasks:

1. The encoder–decoders should be able to abstract and deabstract spectra, meaning they should be able to map spectra to latent representations, then back to the spectra within each domain.
2. They should also be able to transfer spectra from one domain to the other. Once transferred, the spectra produced should look as if they are from the resulting domain.
3. They should retain physical meaning within the transferred spectra. Therefore, when a spectrum is transferred from one domain to the other, it is once again transferred back to the original domain (thus forming a cycle). This cycled spectrum should be identical to the original spectrum.

These three objectives are formed as three different loss functions, which we combine into a single loss function for optimization. We denote (1) the loss related to the within-domain reconstruction as $\mathcal{L}_{\text{rec}}$, (2) the domain transfer loss as $\mathcal{L}_{\text{adv}}$, and (3) the cycle-reconstruction loss as $\mathcal{L}_{\text{cr}}$. Since these losses are replicated for both domains, the overall loss used during training can be written as

$$
\mathcal{L} = \lambda_{\text{synth}} (\mathcal{L}_{\text{rec,synth}} + \mathcal{L}_{\text{cr,synth}}) + \lambda_{\text{obs}} (\mathcal{L}_{\text{rec,obs}} + \mathcal{L}_{\text{cr,obs}}) + \lambda_{\text{adv}} (\mathcal{L}_{\text{adv,synth}} + \mathcal{L}_{\text{adv,obs}}),
$$

where the $\lambda$s are hyperparameters that control the influence of each term.

This loss formulation is similar to that in Liu et al. (2017), except we adopt a mean squared distance instead of a mean
absolute distance. Each term is described below; for specific training details, see Appendix B.

3.2.4. Within-domain Reconstruction, $\mathcal{L}_{\text{rec}}$

The first objective is to ensure that the encoder-decoders are able to reconstruct data within each domain. If we denote the transfer of spectra from domain $a$ to domain $b$ to be $T_{a\rightarrow b}$, we can introduce the shorthand notations

\[
\begin{align*}
\mathcal{X}_{\text{synth} \rightarrow \text{synth}} &\equiv T_{\text{synth} \rightarrow \text{synth}}(\mathcal{X}_{\text{synth}}), \\
\mathcal{X}_{\text{obs} \rightarrow \text{obs}} &\equiv T_{\text{obs} \rightarrow \text{obs}}(\mathcal{X}_{\text{obs}}), \\
\mathcal{X}_{\text{sh} \rightarrow \text{obs}} &\equiv D_{\text{sh/obs}}E_{\text{sh}}E_{\text{synth}}(\mathcal{X}_{\text{synth}}), \\
\mathcal{X}_{\text{obs} \rightarrow \text{sh}} &\equiv D_{\text{obs}}D_{\text{sh/obs}}E_{\text{sh/obs}}E_{\text{obs}}(\mathcal{X}_{\text{obs}}).
\end{align*}
\]  

Here, for simplicity, we have denoted using both encoder-decoders, $E_{\text{sh}}-D_{\text{sh}}$ and $E_{\text{sp}}-D_{\text{sp}}$, as $E_{\text{sh/obs}}-D_{\text{sh/obs}}$. With this notation, the within-domain reconstruction loss function can be written as

\[
\mathcal{L}_{\text{rec}} = d(\mathcal{X}_{\text{synth}}, \mathcal{X}_{\text{synth} \rightarrow \text{synth}}) + d(\mathcal{X}_{\text{obs}}, \mathcal{X}_{\text{obs} \rightarrow \text{obs}}),
\]  

where $d$ is the distance function. For the synthetic domain, a standard mean squared error (MSE) loss is minimized, while for the observed domain, an MSE with samples weighted by the spectrum uncertainties (and bad pixels masked) is minimized.

3.2.5. Cross-domain Translation, $\mathcal{L}_{\text{adv}}$

For the cross-domain translation, recall that there is no direct pairing between spectra in the two domains. To overcome this lack of direct pairing, as in UNIT (Liu et al. 2017), we employ a generative adversarial network (Goodfellow et al. 2014). In other words, the translated spectra are required to “look” as though they belong in the translated domain. Thus, we train critics (also referred to as discriminators) to distinguish between the actual spectra from each domain and the translated spectra.

A core idea behind GANs is that—unlike typical deep network training where a fixed criterion exists—the critic is also a deep network that is trained at the same time as the original generative network (here, our encoder-decoders). Furthermore, since the generative network is trained to fool, whereas the critic is trained to discriminate, the training process is an adversarial game; the critic minimizes a loss function, while the generative network maximizes it.

In the case of CYCLE-STARNET, as shown in Figure 3, we use one critic for each domain: $C_{\text{synth}}$ and $C_{\text{obs}}$. For both critics, they take in reconstructed $^1$ and cross-domain mapped spectra as inputs. The critics then predict a confidence value of whether or not each sample is “real” (i.e., resembling the reconstructed spectra from the cross domain) or “fake” (transferred from the opposite domain). Additionally, to help constrain the shared latent representations to be the same for both synthetic and observed spectra, the critic networks act on the latent space as well. More explicitly, the critics receive paired samples

\[
\begin{align*}
\mathcal{X}_{\text{synth} \rightarrow \text{obs}} &\equiv T_{\text{obs} \rightarrow \text{synth}}(\mathcal{X}_{\text{synth}}), \\
\mathcal{X}_{\text{obs} \rightarrow \text{synth}} &\equiv T_{\text{synth} \rightarrow \text{obs}}(\mathcal{X}_{\text{obs}}), \\
\mathcal{X}_{\text{sh} \rightarrow \text{obs}} &\equiv D_{\text{sh/obs}}E_{\text{sh}}E_{\text{synth}}(\mathcal{X}_{\text{synth}}), \\
\mathcal{X}_{\text{obs} \rightarrow \text{sh}} &\equiv D_{\text{obs}}D_{\text{sh/obs}}E_{\text{sh/obs}}E_{\text{obs}}(\mathcal{X}_{\text{obs}}).
\end{align*}
\]

Note that the $D_{\text{sh/obs}}$ also requires the split latent variables, $Z_{\text{sp}}$. Therefore, for each spectrum transferred from the synthetic to the observed domain, we choose a random observed spectrum, $\mathcal{X}_{\text{obs}}$, encode it with $E_{\text{sp}}E_{\text{obs}}(\mathcal{X}_{\text{obs}})$, and use its split latent values for this process. Since we do not aim for a “target” observed spectrum—but rather, to construct a realistic “observed” spectrum—for the adversarial training, any $\mathcal{X}_{\text{obs}}$ could be used and this would not harm the generality of $\mathcal{L}_{\text{adv}}$. The spectra and their latent representations and predict a confidence value for each pair.

Mathematically, the training objective is defined with a binary cross-entropy function. Binary cross entropy assigns 0 and 1 values for the two groups (real versus fake). In our notation, we assign the value 1 for objects that truly belong to the group, and the value 0 otherwise. In short, the critic would want to assign value 1 for all objects that are reconstructed in a domain-specific manner, and 0 for objects that are passed through the cross-domain translation. If we denote the binary cross-entropy function as $H$, the loss for the critics can be summarized as

\[
\mathcal{L}_{\text{adv}} = H(1, C_{\text{synth}}(\mathcal{X}_{\text{synth} \rightarrow \text{synth}}, Z_{\text{sh} \rightarrow \text{obs}})) + H(0, C_{\text{obs}}(\mathcal{X}_{\text{obs} \rightarrow \text{obs}}, Z_{\text{obs}})) + H(1, C_{\text{obs}}(\mathcal{X}_{\text{obs} \rightarrow \text{obs}}, Z_{\text{sh} \rightarrow \text{obs}}, Z_{\text{sp} \rightarrow \text{obs}})) + H(0, C_{\text{obs}}(\mathcal{X}_{\text{synth} \rightarrow \text{obs}}, Z_{\text{sh} \rightarrow \text{obs}}, Z_{\text{sp} \rightarrow \text{obs}})),
\]

where $Z_{\text{sp} \rightarrow \text{obs}}$ are the split latent representations produced when spectra are cycled from the synthetic domain, to the observed domain, and back to the synthetic domain. Here, we have again used the shorthand notation for the transfer functions:

\[
\begin{align*}
\mathcal{X}_{\text{synth} \rightarrow \text{obs}} &\equiv T_{\text{obs} \rightarrow \text{synth}}(\mathcal{X}_{\text{synth}}), \\
\mathcal{X}_{\text{obs} \rightarrow \text{synth}} &\equiv T_{\text{synth} \rightarrow \text{obs}}(\mathcal{X}_{\text{obs}}), \\
\mathcal{X}_{\text{sh} \rightarrow \text{obs}} &\equiv D_{\text{sh/obs}}E_{\text{sh}}E_{\text{synth}}(\mathcal{X}_{\text{synth}}), \\
\mathcal{X}_{\text{obs} \rightarrow \text{sh}} &\equiv D_{\text{obs}}D_{\text{sh/obs}}E_{\text{sh/obs}}E_{\text{obs}}(\mathcal{X}_{\text{obs}}).
\end{align*}
\]

10 In this study, we use the words “translated” or “transferred” interchangeably. Both words designate the result of our domain adaptations, $\mathcal{X}_{\text{synth} \rightarrow \text{obs}}$ and $\mathcal{X}_{\text{obs} \rightarrow \text{synth}}$, which transform spectra from one domain to the opposite domain.
11 The critics are asked to discern the reconstructed spectra and the cross-domain spectra (instead of the original spectra and the cross-domain spectra) because this will facilitate the translation to denoise the cross-domain transferred spectra. Denoising will be discussed in Section 5.
While the task of the critics is to minimize this particular loss function, the task for the cross-domain autoencoders is the complete opposite: to fool the critics. Therefore, the adversarial objective for the generative processes is to maximize $\mathcal{L}_{\text{adv}}$, instead of minimizing. In order to accomplish this, we simply switch the target class for the domain transferred spectra. The reconstructed “true” spectra are not used for this particular optimization process. Training of the critic network is alternated with the training of the rest of the framework, forming a min–max training setup as in typical GANs. For more details on the critic training, we refer readers to Goodfellow et al. (2014).

### 3.2.6. Cycle Reconstruction, $\mathcal{L}_{\text{cr}}$

Accomplishing the within-domain reconstruction and cross-domain adversarial objectives would result in having a model that can—not surprisingly—reconstruct and cross-domain transfer spectra. In addition, the constraint of applying the critics on the latent representations further provides the basis for a shared latent space. However, there is no guarantee that critics on the latent representations further provides the basis for a given spectrum, $\mathcal{X}_{\text{synth}}$, the cross-domain generated spectrum, $T_{\text{synth} \rightarrow \text{obs}}(\mathcal{X}_{\text{synth}})$, is the correct corresponding spectrum in the $\mathcal{X}_{\text{obs}}$ domain. Therefore, as described in Liu et al. (2017), we enforce that the physical meaning is preserved throughout the transfer by introducing a cycle-consistency constraint. In other words, we require that a spectrum from $\mathcal{X}_{\text{synth}}$ can be mapped to the observed domain and then back to the synthetic domain accurately. The same applies to spectra in the $\mathcal{X}_{\text{obs}}$ domain.

Mathematically, we introduce the shorthand notations

$$\begin{align*}
\mathcal{X}_{\text{synth} \rightarrow \text{obs} \rightarrow \text{synth}} &= T_{\text{obs} \rightarrow \text{synth}} \circ T_{\text{synth} \rightarrow \text{obs}}(\mathcal{X}_{\text{synth}}) \\
\mathcal{X}_{\text{obs} \rightarrow \text{synth} \rightarrow \text{obs}} &= T_{\text{synth} \rightarrow \text{obs}} \circ T_{\text{obs} \rightarrow \text{synth}}(\mathcal{X}_{\text{obs}}).
\end{align*}$$

(6)

Importantly, when transferring an observed spectrum to the synthetic domain, the information in the split latent space is lost. Therefore, to accurately cycle-reconstruct this spectrum, the originally encoded split latent variables are used when mapping this spectrum back to the observed domain.

We can write the cycle-reconstruction loss as

$$\begin{align*}
\mathcal{L}_{\text{cr}} &= d(\mathcal{X}_{\text{synth}}, \mathcal{X}_{\text{synth} \rightarrow \text{obs} \rightarrow \text{synth}}) \\
&\quad + d(\mathcal{X}_{\text{obs}}, \mathcal{X}_{\text{obs} \rightarrow \text{synth} \rightarrow \text{obs}}),
\end{align*}$$

(7)

which is similar to $\mathcal{L}_{\text{rec}}$, but with the cycled spectra.

### 3.2.7. Extracting Spectral Feature–Label Correlations

With the synthetic spectral emulator included, CYCLE-STARNET provides a closed connection between stellar labels and the systematics-corrected translated spectra. This can be done by evaluating the continuous flow of $\mathcal{X}_{\text{synth} \rightarrow \text{obs}} = T_{\text{synth} \rightarrow \text{obs}} \circ \text{Payne}(\mathcal{Y})$, as shown in Figure 4. Moreover, since each mapping—including the synthetic emulator and autoencoder—is continuous, by interpreting the differential relations that the network has found, we can identify spectral features and associate them with individual stellar labels.

Spectral features can be identified by calculating how individual input elemental abundances impact the output pixels in the observed domain. For example, if $\mathcal{X}_{\text{synth} \rightarrow \text{obs}}$ is the systematic-corrected model, then the partial derivative of the individual pixels with respect to a particular element, $\mathcal{Y}_j$, shows the spectral response to that element, which can be written as

$$\frac{\partial \mathcal{X}_{\text{synth} \rightarrow \text{obs}}}{\partial \mathcal{Y}_j} = \left( \frac{\partial \mathcal{X}_{\text{synth} \rightarrow \text{obs}}}{\partial \mathcal{Y}_j}, \ldots, \frac{\partial \mathcal{X}_{\text{synth} \rightarrow \text{obs}}}{\partial \mathcal{Y}_j} \right).$$

(8)

The derivative spectrum for the synthetic domain emulator, $\mathcal{X}_{\text{synth}} = \text{Payne}(\mathcal{Y})$, can also be calculated, which provides the information held within our theoretical models. Therefore, by taking the difference between the two “response functions,” the additional information that is not contained in the synthetic models can be revealed. This method is tested in Section 4.2.

### 4. Experiments and Results

In this section, we present two case studies to show that CYCLE-STARNET can generate systematic-corrected models from unlabeled observed spectra. In Section 4.1, we show how Kurucz synthetic models translate into APOGEE-observed spectra (spectra taken from the Apache Point Observatory Galactic Evolution Experiment). We quantify the agreement between spectra in terms of the residuals and via a t-distributed stochastic neighbor embeddings analysis. Admittedly, the accurate agreement of spectra does not necessarily guarantee that CYCLE-STARNET has learned actual physics. Therefore, we further investigate the derivatives of CYCLE-STARNET in Section 4.2 to determine how the network derivatives have aligned with stellar labels. This second case study shows that the network has learned the actual physics behind the spectra. Specifically, we show that CYCLE-STARNET can associate...
missing spectral features in the synthetic spectra with their correct corresponding elemental abundances.

4.1. Mitigating the Synthetic Gap with CYCLE-STARNET

In this section, we show how to mitigate the synthetic gap between the Kurucz models and APOGEE observations with CYCLE-STARNET.

4.1.1. Experimental Setup

For this case study, ATLAS12/SYNTHETE models (Kurucz & Avrett 1981; Kurucz 1993, 2005, 2013) are adopted for the synthetic domain, and THE PAYNE is used as the synthetic emulator (Section 3.1). Similar to Ting et al. (2019), instead of using the default Kurucz line list, we utilized a calibrated line list by P. Cargile et al. (2020, in preparation), which was tuned to better match the solar and Arcturus Fourier transform spectrum (FTS). As for the observed domain, to be consistent with Ting et al. (2019), APOGEE DR14 spectra are adopted, which have been wavelength calibrated to vacuum to be consistent with the Kurucz models. These spectra are constructed by coadding multiple velocity-corrected visits of the same object. All spectra are continuum normalized using the same routine as in Ting et al. (2019).

For simplicity, the APOGEE-Payne catalog (the revised APOGEE catalog of stellar labels determined by using THE PAYNE) is adopted as our reference. To reduce the effects of outlier spectra, only APOGEE spectra that have decent fits in the APOGEE-Payne catalog are included, that is, those with a reduced χ² < 50 and a total broadening (which consists of both macroturbulence and rotation) of less than 10 km s⁻¹. Finally, only APOGEE spectra with a median signal-to-noise ratio (S/N) of 30 < S/Nsynth < 500 are included to eliminate noisy or saturated spectra.

Furthermore, the APOGEE-Payne catalog is randomized to showcase how CYCLE-STARNET can perform domain adaptation with unpaired spectra. In particular, for the observed domain, we adopt APOGEE spectra from half of the objects that meet the above criterion. For the other half of the objects, the APOGEE-Payne stellar labels are used to generate the synthetic Kurucz spectra. While the APOGEE-Payne labels are used to generate the Kurucz models and remove outlier spectra, we emphasize that CYCLE-STARNET never “sees” the stellar labels of either domain; the training is entirely unsupervised.

The 25 stellar labels from APOGEE-Payne, which are inherited, include Teff, log g, microturbulence, additional broadening, 20 elemental abundances in [X/H], namely, C, N, O, Na, Mg, Al, Si, P, S, K, Ca, Ti, V, Cr, Mn, Fe, Co, Ni, Cu, and Ge, and the isotopic ratio C12/C13. The above procedure yields a set of 97,000 spectra in each domain. From these, we adopt 80,000 spectra as the training set, adopt 10,000 spectra as the validation set, and withhold 7000 spectra as the test set. Unless stated otherwise, all of the results shown below are based on the test set.

4.1.2. Within-domain Reconstruction

As discussed in Sections 3.2.4 and 3.2.6, CYCLE-STARNET provides two approaches for reconstructing spectra within the same domain: direct reconstruction and cycle reconstruction. We first demonstrate that these within-domain reconstructions work, which are a necessary condition for the cross-domain translations.

In Figure 5, we show the accuracy of these mappings in the synthetic domain. The autoencoded, $\chi_{\text{synth} \rightarrow \text{synth}}^2$, and cycle-reconstructed spectra, $\chi_{\text{synth} \rightarrow \text{obs} \rightarrow \text{synth}}^2$, are compared to the original spectra, $\chi_{\text{synth}}^2$. To show the relative residual, we normalize the difference with the original spectra, and we display the 16–84 percentile as the $1\sigma$ range to estimate the normalized scatter. The red line demonstrates the pixel-by-pixel mean, and the orange line the absolute mean. As demonstrated, CYCLE-STARNET can reconstruct the synthetic domain with negligible bias (< 0.1%) and a scatter of ~0.4%. This applies to both the direct reconstruction and the cycle reconstruction. Recall that the cycle reconstruction is performed by passing information first to the opposite domain (here, the observed domain) and then back to the original domain (the synthetic domain). The fact that this cycle reconstruction is accurate demonstrates that the latent space has learned not only information within the domain but also information from the opposite domain.

Figure 6 shows similar results for the observed domain. However, since the observations are noisy, we do not expect the “denoised” reconstructions to be the exactly the same as the input spectra. Therefore, we normalize the residuals with the uncertainties of the observed spectra as reported by APOGEE, essentially calculating the reduced $\chi_{\text{R}}^2$. The deviations for both reconstructions are minimal when compared to the original spectra, providing a negligible bias and a normalized scatter of $\sim 1$. This demonstrates that CYCLE-STARNET is also able to reconstruct spectra in the noisy observed domain. Finally, the fact that the normalized scatter is close to 1 demonstrates that the reconstructions are implicitly denoised.

Nevertheless, outliers do exist and some pixels have more substantial deviations. There are ~3% of the pixels that have a normalized deviation greater than 3$\sigma$ ($s > 3$). The exact reason for these substantial variations is unclear, but we suspect some mischaracterizations of the APOGEE uncertainties may be a cause. Here, the uncertainties provided by APOGEE are assumed to be calibrated and uncorrelated between pixels, which might not be strictly true, especially for the resampled and coadded spectra. This also motivates us to take the 16–84 percentiles of the residuals, instead of the simple standard deviation, to calculate the reduced $\chi^2_{\text{R}}$ and normalized scatter.

4.1.3. Domain Adaptation with CYCLE-STARNET

In order to evaluate the accuracy of the domain-adaptation procedure, we perform a least-squares fitting on the observed spectra to determine how closely the best-fit spectra match the observations. This provides a direct method to evaluate the improvements in model calibrations, as well as estimates for the associated stellar labels (see Section 4.1.5). In more detail, as done in Ting et al. (2019), the fitting can be achieved by using the emulator directly, along with the error spectrum, $\sigma_{\text{obs}}$:

$$\hat{y} = \arg\min_y \left\| \frac{1}{\sigma_{\text{obs}}} (\chi_{\text{obs} \rightarrow \text{ Payne}}(y)) \right\|_2^2$$

With CYCLE-STARNET, there are two alternate ways to fit the spectra. The first is achieved by mapping the observed spectra to the synthetic domain and again using the emulator directly for the fitting:

$$\hat{y} = \arg\min_y \left\| \chi_{\text{obs} \rightarrow \text{synth} \rightarrow \text{ Payne}}(y) \right\|_2^2$$

Alternatively, the Kurucz models can be transferred to the observed domain, having their observed representations
A comparison between the cycle-reconstructed spectra, these autoencoders. Shown are the relative residuals for the test set of 7000 synthetic spectra whose stellar labels are drawn randomly from the APOGEE-Payne catalog. Overplotted in each panel are the pixelwise mean residual clarity. The top panel shows the relative residuals between the reconstructed spectra compared to the observations. In other words, CYCLE-STARNET can be used to perform the least-squares fitting in the observed domain by forward modeling the synthetic spectra:

$$\hat{y} = \arg \min_{y} \left\| \frac{1}{\sigma_{\text{obs}}} (\chi_{\text{obs}} - T_{\text{synth}}(\chi_{\text{obs}})) \right\|_2^2 \quad (11)$$

Unless stated otherwise, we refer to Equation (9) as the Kurucz fitting and Equation (11) as the default CYCLE-STARNET fitting. As we will discuss, for CYCLE-STARNET, it is better to forward model the synthetic spectra to the data space, instead of transforming the data to the synthetic space.

In Figures 7 and 8, we show examples of how CYCLE-STARNET is capable of adapting spectra from one domain to the other, which is a key objective of this study. Figure 7 demonstrates this procedure for a typical M-giant with solar abundances in APOGEE. The upper panel compares the APOGEE spectrum to the corresponding best-fit Kurucz model. It is clear that, even though the two spectra are normalized with the same procedure, the synthetic gap persists. In contrast, the second panel of Figure 7 compares the same APOGEE spectrum to the best-fit CYCLE-STARNET model. The transferred model illustrates how CYCLE-STARNET can correct for the improperly modeled spectral features. Furthermore, CYCLE-STARNET has learned to understand the imperfect continuum normalization in the data and produces transferred models that are consistent with such normalization. The last two panels of Figure 7 show the residuals for these comparisons across the entire wavelength region and normalized by the reported pixel uncertainties in the APOGEE spectrum.

To further illustrate this point, in Figure 8, we analyze the residuals for all 7000 test spectra. The top panel shows the residuals between the APOGEE spectra and the best-fit Kurucz models, whereas the bottom panel shows these same residuals with the transferred synthetic spectra produced by CYCLE-STARNET. The residuals are normalized by the reported APOGEE pixelwise uncertainties in the observed spectra. Similar to Figure 7, after the domain adaptation, the spectra exhibit much better agreement with the observations, reducing the synthetic gap. Evidently, when transferring the spectra with CYCLE-STARNET, the sample bias is 15 times smaller, and the sample reduced $\chi^2$ is 1.6 times smaller, reaching almost the same precision as the within-domain reconstructions, shown in Figure 6. Since the residuals are normalized by the uncertainties in the observed spectra, a normalized scatter (or reduced $\chi^2$) of $\sim 1$ demonstrates that the deviations are consistent with the observational uncertainties. As previously mentioned, the Kurucz models used in this study have been generated with the line list improved by P. Cargile et al. (2019, in preparation), as described in Ting et al. (2019). Consequently, the synthetic gap would be even larger if we were to use the original Kurucz models rather than those with the improved line list.

To determine how these residuals are dependent on the stellar parameters, Figure 9 shows a Kiel diagram colored by the average reduced $\chi^2$ in the best-fit spectra. The left panel shows the results using the Kurucz models, and the right panel shows the results using the CYCLE-STARNET calibrated models. While the Kurucz models exhibit larger deviations from the observations (particularly the cool M-giants), the best-fit spectra produced by CYCLE-STARNET are in much better agreement and are consistent with the APOGEE observational uncertainties for all stellar types.

#### 4.1.4. Visualization of Domains via t-SNE

An intuitive visualization of the synthetic gap—and how we reduce it—is provided by using t-distributed stochastic neighbor embeddings (t-SNE; van der Maaten & Hinton 2008).
The t-SNE is a dimensionality-reduction technique that is widely used to visualize high-dimensional spaces. Within the context of this paper, the t-SNE projects spectra from an ~7000-dimensional “spectral pixel” space (or a 600-dimensional latent representation) to a compressed, two-dimensional representation. This allows one to illustrate the distribution of a high-dimensional data set in a 2D space, where each sample is represented by a single point. Furthermore, the proximity of two points in the 2D space demonstrates the similarity of those two samples. Since the t-SNE projected space is a lower-dimensional representation with arbitrary units, there are no explicit dimensions in the axes of the plots.

In Figure 10, using three separate t-SNE analyses, we show how our method mitigates the synthetic gap. The left panel shows the comparison between the Kurucz synthetic models, $X_{\text{synth}}$, and the APOGEE spectra, $X_{\text{obs}}$. Consistent with Figure 8, the two domains are distinct in the t-SNE projection, which is further evidence of the synthetic gap. By contrast, the middle panel compares the domain transferred spectra, $X_{\text{synth} \rightarrow \text{obs}}$, to the autoencoded version of the original observed spectra, $X_{\text{obs} \rightarrow \text{obs}}$. Evidently, the domain-adapted synthetic spectra show close agreement with the observed spectra, demonstrating that CYCLE-STARNET can generate accurate synthetic spectra that are almost indistinguishable from those in the observed domain. In these first two analyses, since CYCLE-STARNET implicitly denoises spectra, we adopted the reconstructed versions of the observed spectra, $X_{\text{obs} \rightarrow \text{obs}}$, as the reference for our comparisons instead of the original noisy spectra, $X_{\text{obs}}$. The denoised versions were chosen to demonstrate that the synthetic gap is inherently due to imperfect modeling and reduction, not the observational noise. Therefore, the autoencoded $X_{\text{obs} \rightarrow \text{obs}}$ is a more direct comparison with $X_{\text{synth}}$ and $X_{\text{synth} \rightarrow \text{obs}}$ because these are also noiseless.

Finally, the right panel shows the two latent representations of $X_{\text{synth}}$ and $X_{\text{obs}}$ produced by their respective encoders. The agreement between the latent representations demonstrates that CYCLE-STARNET has indeed created a shared latent space that extracts common information from both domains.

4.1.5. Deriving Stellar Parameters for APOGEE

In this section, we study the recovery of stellar parameters from APOGEE spectra with the better-calibrated Kurucz models from CYCLE-STARNET. In particular, we fit 100,000 random APOGEE spectra taken from the observed domain and compare the results to the original stellar parameters inferred with THE PAYNE, which uses the same Kurucz models. A more complete inference framework with a detailed comparison to other pipelines—as well as the study of elemental abundances—is deferred to future work.

As mentioned in Section 4.1.3, there are several methods that can be used to infer stellar labels from observed spectra. In Ting et al. (2019), the emulator is used directly as a surrogate for the physical model (Equation 9). Naturally, this procedure is limited by the accuracy of the synthetic models. A second approach is to accommodate these inaccuracies by producing synthetic counterparts for the observed spectra (using CYCLE-STARNET) and, again, performing the fitting with the emulator (Equation 10). Finally, the domain transfer—achieved through CYCLE-STARNET—can be used to fit observed spectra directly in the observed domain (Equation 11) by forward modeling the Kurucz models into the observed domain. In the following example, each of the objective functions (Equations 9, 10, and 11) is minimized using a
Figure 7. Mitigating the synthetic gap with CYCLE-STARNET. The top panel shows a portion of the best-fit Kurucz model for an M-giant APOGEE spectrum with solar-like abundances. While the Kurucz model is broadly consistent with the APOGEE observation, a minor synthetic gap persists. The second panel shows the CYCLE-STARNET spectrum generated via domain transfer. Evidently, the transferred model exhibits better consistency with the observation, especially for cool stars like M-giants whose spectral models were not well calibrated. On top of that, CYCLE-STARNET also learns the imperfect continuum normalization in the data (as illustrated by the fact that some normalized values are >1) and produces transferred synthetic models that are normalized in a self-consistent manner. The last two panels show the normalized residuals for these same two comparisons, but across the entire wavelength range.
second-order method (the limited-memory Broyden–Fletcher–Goldfarb–Shanno algorithm, or L-BFGS).

Figure 11 illustrates the differences between these three methods by plotting the respective estimates of surface gravity against effective temperature for the 100,000 APOGEE spectra. To provide a visual guide for the general expected trend, Modules for Experiments in Stellar Astrophysics (MESA) Isochrones and Stellar Tracks (MIST) isochrones at 7 Gyr old are also plotted. In the top left panel, the results from Ting et al. (2019) are shown. As discussed in their paper, to omit poorly modeled spectral regions, they constructed a spectroscopic mask that is specific to the data set. In the top right panel, the results produced by THE PAYNE without using this spectroscopic mask are shown. Consequently, due to the imperfection of the Kurucz models, fitting spectra without the spectroscopic mask exacerbates the systematic biases, for example, the cool M-giants.

The results from performing the CYCLE-STARNET fitting in the synthetic space (using Equation (10)) are shown in the bottom left panel of Figure 11. These estimates exemplify a tighter distribution of parameters (perhaps even overly tight compared to the isochrones), yet some metal-poor dwarf stars are inconsistent with the expected trend. While transferring observed spectra to the noiseless synthetic domain—and performing a least-squares fitting on the synthetic spectra—corrects for several issues (e.g., noise, outliers, and instrumental effects), it also limits the predicting power of the network. For instance, if an object is observed that is outside of the distribution of our synthetic training set, the observed spectrum may be moved to fit within this distribution when transferred to the synthetic domain. Alternatively, if we use the original observed spectrum and instead transfer the synthetic spectra to the observed domain during the fitting procedure, we mitigate the potential for the underlying object to be altered. Furthermore, performing the fitting process in the observed domain can be cast as a likelihood estimation—by taking into account the observed noise properly—and is therefore more interpretable. Due to these considerations, we choose to adopt CYCLE-STARNET to transfer synthetic models to the observed domain as our preferred fitting routine throughout this paper.

The bottom right panel shows the results obtained by fitting with CYCLE-STARNET in the observed domain, that is, with the better-calibrated Kurucz models produced through domain adaptation (Equation (11)). Not surprisingly—and consistent with Figure 8—the domain transferred models still provide a reduced amount of scatter in the estimated stellar parameters. Using CYCLE-STARNET to perform the fitting in the observed domain also shows improvements compared to fitting in the synthetic space. For instance, the distribution of $T_{\text{eff}}$–$\log g$ is more in line with the isochrones, and the problematic fits for the metal-poor dwarfs in the synthetic space are alleviated. Importantly, CYCLE-STARNET allows the fitting to be performed on the entire spectrum, using all of the information collected, without the need for a spectroscopic mask.

4.2. Learning Stellar Physics with CYCLE-STARNET

While we demonstrate in Section 4.1 that CYCLE-STARNET successfully morphs the Kurucz models to the APOGEE observations—closing the synthetic gap—it does not guarantee that CYCLE-STARNET has extracted useful physics. In this section, we show a possible physical interpretation of CYCLE-STARNET using the network’s derivatives, in particular, the derivatives of individual flux intensities with respect to elemental abundances.
4.2.1. Experimental Setup

As discussed in Section 3.2.7, the derivatives (or “flux responses”) of elemental abundances help determine which spectral features are associated with a particular element. However, for the APOGEE observed spectra, it is impossible to know the ground truth; we simply do not know what may or may not be missing in our line list. Therefore, as a proof of concept, we consider a mock “observed” data set drawn from the Kurucz models instead, which allows us to know the ground truth derivatives. Applications to real spectra are deferred to future studies.

The training of this version of CYCLE-STARNET is similar to the one in Section 4.1, but this time we create a controlled observed domain. In more detail, instead of adopting the APOGEE spectra, an “observed” data set is synthesized with Kurucz models using the APOGEE-Payne labels that correspond to our observed training set used in Section 4.1. Noise is added to these mock observed spectra to mimic a more realistic observed training set.

Additionally, in the synthetic domain, we randomly masked approximately 30% of the strong absorption features by setting them to the continuum level. In order to mask features that are associated with particular elemental sources, the gradients of the emulator network were investigated. In more detail, for a solar-metallicity K-giant (Teff = 4750 K, log g = 2.5), the gradient of each pixel in the simulated spectrum was calculated with respect to each input elemental abundance. Absorption lines with at least one pixel that had a gradient (in normalized flux) less than −0.1 dex−1 were considered “strong features” of that element, and surrounding pixels with a gradient less than −0.02 dex−1 were included as part of the feature. Furthermore, only elements with more than three strong features were included in the masking procedure.

To summarize, in this controlled experiment, two sets of unpaired Kurucz models are utilized. The synthetic domain is composed of noiseless Kurucz models where 30% of the strong spectral features (from prominent elements) are missing. The observed spectra are the original Kurucz models without missing features, but with added noise, mimicking the real APOGEE spectra. This will demonstrate that CYCLE-STARNET can learn actual physics. In particular, by learning from the data alone, CYCLE-STARNET can correctly identify missing spectral features in the “synthetic” models and associate them with the correct corresponding elements.

The flux derivatives of the synthetic emulator, ∂Payne( ylabel) /∂ylabel, and the derivatives of the domain transferred spectra, ∂T synth—obs(Payne( ylabel)) /∂ylabel, are calculated. The former informs us of the original input line list in the “synthetic” domain. The latter reveals the “true,” complete line list learned from the “observed” data. The differences between the two are used to identify the missing spectral features.

4.2.2. Recovering Missing Spectral Features

An example of the results of this domain-adaptation problem is shown in Figure 12. A K-giant (Teff = 4750 K, log g = 2.5) with solar abundances is used as our set of reference labels, ylabel. The top panel shows the spectrum in the synthetic domain, Payne( ylabel), and the second panel shows the transferred synthetic spectrum, T synth—obs(Payne( ylabel)), as well as an actual observed spectrum with the same stellar labels. In this second panel, when mapped to the observed domain via CYCLE-STARNET, the missing features in the synthetic domain are correctly filled in. Similar to Figure 7, this provides evidence that CYCLE-STARNET is able to bridge the synthetic gap.

The last two panels in Figure 12 demonstrate that, not only can CYCLE-STARNET accurately fill in the missing features, but it also associates them with their corresponding elements. In more detail, the third panel shows the difference between the transferred spectrum and the synthetic spectrum, which illustrates the missing lines and their corresponding elemental abundances. To make this more clear, we color code the masked features with their associated elements, focusing on the four elements that have a prominent presence in the APOGEE.

---

12 This is not strictly true because some elemental abundances, especially for prolific electron donors, can also substantially change the stellar atmospheric structure, which indirectly affects all spectral features. As a result, spectral features that vary with a particular element do not necessarily imply that those features are due to the direct atomic or molecular transitions of that specific element.
Figure 10. The t-SNE projections of 7000 pairs of test spectra. The left panel illustrates the synthetic gap between the Kurucz models and the APOGEE spectra; without domain adaptation, spectra from the two domains span different regions in the t-SNE projection. The middle panel shows the CYCLE-STARNET results, which exemplify the effectiveness of the generative aspect of CYCLE-STARNET. After domain adaptation, CYCLE-STARNET successfully morphs the synthetic domain to the observed domain. In both panels, to have a more robust comparison, CYCLE-STARNET will only be able to limit the potential sources that mimic the APOGEE observations. Similar results are found for most other missing lines associated with other elemental abundances. Nonetheless, as occasionally seen in Figure 12, the gradients can be nonzero for unassociated elements, signaling that the unsupervised learning can still be improved in future studies.

In order to assess how well the network is able to recover missing lines for other objects, we performed a similar analysis on 100,000 different sets of stellar labels. Figure 13 exemplifies that the accurate recovery is consistent across the parameter space of the three atmospheric stellar parameters. In this figure, the varying areas of the parameter space are colored by the percentage error in the pixel recovery for the missing lines. The percentage error is calculated by taking the absolute residual between the recovered masked pixels and the ground truth flux (normalizing by the ground truth). The only regions that see a substantial discrepancy (∼1%) between the recovered lines and the ground truth are the coolest M-giants. This is presumed to be due to a lower number of training samples in this region.

While this experiment produces encouraging results for line identifications, it also highlights some limitations. Specifically, CYCLE-STARNET can struggle to identify the exact sources of features when multiple elements contribute to the feature, either directly or indirectly. This is especially true for the CNO molecular features (via the molecular balance of CNO, e.g., Ting et al. 2018). For instance, Figure 14 shows a few of the carbon and nitrogen features that were masked in the synthetic domain. As illustrated, CYCLE-STARNET can assign the C- and N-related features to either C or N or both. In these cases, CYCLE-STARNET will only be able to limit the potential sources rather than identify them precisely through a gradient analysis. However, such molecular features are usually very prominent in the spectrum, with many neighboring transitions within the wavelength region. As a result, incorporating prior domain knowledge of stellar spectroscopy could readily resolve this limitation.

Lastly, for the sake of completeness, an experiment was conducted to analyze the extent of missing information in the synthetic domain that can be recovered. In more detail, nine instances of training CYCLE-STARNET were completed while the percentage of masked absorption features was increased from 10% to 90% (in increments of 10%), and the percentage error in pixel recovery for these masked regions was calculated. Table 1 summarizes these results by comparing the percentage error for the pixel recovery for each training instance. Interestingly, in terms of flux recovery, the network only appears to see a significant decrease in performance once 80% of the strong features are masked.

5. Discussion

In this study, we showed how unsupervised domain adaptation is a promising method for autocalibrating for model inaccuracies by exploiting large data sets. To illustrate this, we presented a case study using stellar spectroscopy surveys. In particular, we demonstrated that domain adaptation is a powerful idea that can harness the strengths of large data sets and synthetic modeling while mitigating their limitations. In the following, we first discuss the advantages of CYCLE-STARNET when compared with standard spectral calibration and fitting methods. We then discuss the limitations of CYCLE-STARNET.

5.1. Advantages of Applying CYCLE-STARNET

A key advantage of CYCLE-STARNET compared to existing data-driven models (e.g., Ness et al. 2015; Fabbro et al. 2018; Leung & Bovy 2019) is that the training does not require any labels for the observed data. This is important as supervised training often inherits biases in the training labels. These biases are a result of obtaining training labels derived from other pipelines, which adopt their own set of models and fitting techniques that are susceptible to systematics. In contrast, CYCLE-STARNET learns a common abstraction of both the synthetic and observed domain—purely from the data—and directly adapts data in the synthetic domain to the observed domain and vice versa. Therefore, unlike other approaches,
CYCLE-STARNET is not subject to biases in the training labels. Thus, CYCLE-STARNET provides an entirely new idea for constructing data-driven models through unlabeled, large data sets. Furthermore, one can interpret the shared representation as a multidimensional match, alleviating the need for spatial cross-matching between surveys. This would ultimately mitigate the selection biases seen in purely data-driven analyses.

Moreover, CYCLE-STARNET can yield robust and denoised data-driven models, even when trained with noisy spectra. We attribute this “effective denoising” of observed spectra to the following subtleties related to our method. First, the reconstruction and cycle-reconstruction loss functions are weighted by the uncertainties in the spectra. As a result, this weighting forces the network to focus more on the cleaner portions of the data set and downweight the noisier spectral fluxes. In addition, when provided with enough data, autoencoders learn to reconstruct the common information within a data set (i.e., spectral features) and therefore ignore information that is highly specific to a given sample (i.e., the noise). Lastly, to facilitate convergence and training, we use the reconstructed spectra as our input samples for the critic networks (see Equation (4)). Therefore, the transfer of spectra from the synthetic to the observed domain is implicitly primed to produce noiseless spectra. Being able to learn from noisy spectra is particularly useful as we can train with the bulk of the survey data instead of restricting it to only a small subset of high-S/N spectra.

Since the network incorporates synthetic data in the training and is applied to real observations, CYCLE-STARNET can also be regarded as an autocalibration of the synthetic models. However, unlike standard calibration methods that are often based on a few standard stars (e.g., the Sun and Arcturus, Shetrone et al. 2015), we effectively calibrate based on all of the stars in the data set. The critical insight here is that spectra are fundamentally low-dimensional objects that lie on a manifold once transferred to an abstract latent space. The latent space corresponds to the astrophysical properties of stars and has a finite and small number of degrees of freedom. Benefiting from a large amount of unlabeled data, we apply machine learning to discover this manifold that both the synthetic and observed spectra lie on. By enforcing the commonality of the manifold, autocalibration is attained. This is a drastically different philosophy for calibrating spectral models.
models, a calibration that relies on the redundancy in large data sets rather than standard "ground truths."

As demonstrated in this study, our new insight can lead to a better calibration of spectral models than standard calibration techniques. This is perhaps not surprising because the standard calibration focuses only on a few stars, which span a limited range of stellar labels. Consequently, calibrating with relatively hot stars like the Sun and Arcturus requires extrapolation when considering cooler stars like the M-giants (see Figures 7 and 9). In contrast, CYCLE-STARNET calibrates the models using all of the available stars, which span the entire stellar parameter space of interest. Furthermore, since the network learns from the existing data, it can capture the variations in the instrumental and observational effects. This allows the method to correct for things that are challenging to model with preexisting methods (e.g., the line-spread function variation), alleviating a key roadblock when comparing models to observations.

Finally, while we adopted stellar spectra as a case study, we emphasize that unsupervised domain adaptation could be broadly applied in astronomy as well as other fields. For instance, our method may be suitable for studies where computationally expensive simulations could be emulated and large data sets exist to learn the corrections. It can be especially useful when, while the initial physics is simple, accurate modeling is infeasible because of the complexity of modeling subtle variations. For example, improved modeling of spectral energy distributions of stellar populations could be accomplished with CYCLE-STARNET. As another example, relating N-body simulations of dark matter distributions to large-scale galaxy data is an ideal candidate. For fields outside astronomy,
medical imaging provides interesting opportunities for CYCLE-STARNET because the data are highly dependent on the particular medical device that is used for data acquisition. These instrumental effects could be corrected either by combining simulations and clinical data or by learning the translation of data from one system to another. Last but not least, climate modeling is an area where efficiency could be improved by relating low-resolution models to high-resolution data, a domain-transfer problem where the difference between the domains is resolution.

5.2. Limitations and Future Implementations

While CYCLE-STARNET has many attractive properties, the training of CYCLE-STARNET can be delicate and may still require some fine-tuning. The difficulty mainly arises from the adversarial training setup. Specifically, adversarial training provides competition between the autoencoder networks and the critic networks, making convergence not as straightforward as other machine-learning tasks such as supervised regression. We note that the other proposed methods, including THE PAYNE (Ting et al. 2019), THE CANNON (Ness et al. 2015), AstroNN (Leung & Bovy 2019), and StarNet (S. Bialek et al. 2020, in preparation), are examples of supervised regression, and therefore are technically much easier to train.

To train CYCLE-STARNET, we have extensively explored numerous architectural choices and training details before deciding upon the final model. For instance, removing the adversarial losses leads to a smoother convergence, but without it, the latent representations are no longer shared, causing cross-domain translations to fail. Nevertheless, adversarial training is an active field of research in machine learning, with many new ideas proposed (Deshpande et al. 2018; Miyato et al. 2018; Donahue & Simonyan 2019) since the original UNIT paper was published. Consequently, we expect that some of these advancements would benefit and stabilize the training of future versions of CYCLE-STARNET.

Besides the difficulty in training in an adversarial setting, we have made a few simplifying assumptions regarding the training spectra data sets. First, we assumed that the uncertainties provided by APOGEE are uncorrelated and accurately determined. How a mischaracterization of the noise may skew the training is yet to be studied.

Additionally, the labels from both domains in this study span the same ranges. Early experiments suggest that having the two domains span a similar label space is an essential aspect of CYCLE-STARNET. For a new survey, we might not know beforehand the range of the stellar labels. Knowing how to resolve this problem, as well as efficient sampling of the $\geq 25$ parameter space dimensions, requires a more thorough investigation. While in principle, the training of CYCLE-STARNET does not require paired samples in the two domains—and is therefore unsupervised—here we accelerated the training by matching distributions from previously estimated stellar parameters from APOGEE-Payne and then generating synthetic spectra by drawing from these distributions. The use of the APOGEE-Payne distributions to generate the synthetic domain implies that we are not entirely blind when training the domain-adaptation process. Perhaps weakly supervised is a more appropriate description of the framework.

Furthermore, since CYCLE-STARNET is trained on the “bulk” of the observed data, the network might not extrapolate well for out-of-distribution samples and exotic stellar objects. To mitigate the potential impact of outliers during training, we eliminated APOGEE spectra by only considering spectra that produced a decent fit in the original APOGEE-Payne (reduced $\chi^2 < 50$). We do not expect a small number of odd samples to dominate the training, but, when adopting this method for other applications, it may be useful to remove outliers from the training process (e.g., through “simpler” unsupervised clustering methods).

Recall that the domain-adaptation process of CYCLE-STARNET conducts a shared abstraction of both domains. The abstraction is then traced back to the stellar labels via a physics surrogate emulator. As a result, CYCLE-STARNET autocalibrates spectra primarily based on the original models, with which the emulator is trained. As demonstrated in Figure 10, some systematics can persist even after the autocalibration. For example, the $T_{\text{eff}}$ and log $g$ of the red clump stars are still systematically lower than the isochrones, similar to the results from the original Kurucz models. Additionally, the stellar parameters for the coolest dwarfs remain problematic. These results illustrate that the autocalibration of CYCLE-STARNET only works to a certain extent and may not correct for significant “zero-point” biases in the models. Including external physical priors (e.g., an isochrone prior) could constrain the zero-point biases.

With regards to the case study presented in Section 4.2, where the flux responses of CYCLE-STARNET were investigated to identify missing information in the synthetic modeling, the application to real observations can present new challenges. For instance, the difference in the flux responses will no longer be strictly due to missing spectral lines; inaccurate oscillator strengths or damping, for example, can cause discrepancies between the flux responses. This is by design and exemplifies how CYCLE-STARNET is correcting for these effects, but this adds more complexity to the analysis. However, since it is impossible to synthesize all possible unknowns, in practice, the flux responses should be examined on a case-by-case basis, and domain knowledge should be utilized.

As a final remark, CYCLE-STARNET is a project that began 2 yr ago. During the development of CYCLE-STARNET,
multiple unsupervised domain-adaptation methods have emerged with the potential to alleviate some of the mentioned caveats. In particular, the UNIT method has been adapted to multiple domain adaptation (Huang et al. 2018) and to smaller sample requirements (Liu et al. 2019). Normalizing flows for domain adaptation has also been shown to enforce cycle consistency and provide more tractable gradients and likelihoods by construction (Grover et al. 2020).

5.3. Open Source

For future reproducibility and to assist with applications to other projects, we have made the code for CYCLE-STARNET publicly available on GitHub, along with in-depth explanations on the code itself and the training details. As noted in Section 5.2, training CYCLE-STARNET can be sensitive to the choices of the network architecture and training hyperparameters. Therefore, we emphasize that GitHub aims to only serve as a starting point for other applications.

6. Conclusion

Maximally extracting information from stellar spectra requires that we have perfect knowledge of spectral synthesis,

---

Table 1
Average Percentage Error of Pixel Recovery for the Masked Absorption Lines When Varying the Amount of Masked Features in the Synthetic Domain

| % of Features Masked | Avg. Error (%) |
|----------------------|----------------|
| 10                   | 0.3835         |
| 20                   | 0.4136         |
| 30                   | 0.3547         |
| 40                   | 0.3750         |
| 50                   | 0.3862         |
| 60                   | 0.3903         |
| 70                   | 0.3794         |
| 80                   | 4.4961         |
| 90                   | 4.5906         |

---

13 https://github.com/teaghan/Cycle_SN
an idea that has remained elusive despite decades of studies. In this paper, we present a new methodology, CYCLE-STARNET, to tackle this problem. CYCLE-STARNET adopts ideas from domain adaptation in machine learning to mitigate model systematics. Our results are summarized below:

1. CYCLE-STARNET autocalibrates for deficiencies in spectral modeling and develops a common abstraction of both synthetic and observed data through an unsupervised network. This abstraction is related to physical stellar labels via a physics surrogate emulator network.

2. CYCLE-STARNET can build data-driven models via a set of unlabeled training spectra, without knowing the stellar labels of the training spectra a priori.

3. Through the use of a split latent space, CYCLE-STARNET can distinguish the actual astrophysical information from the spectral variations due to instrumental and observational factors. This reduces the reliance of the modeling accuracy on instrumental factors, such as the fiber-dependent line-spread functions and telluric feature modeling.

4. By fitting the APOGEE spectra, we demonstrated that the autocalibrated models produced by CYCLE-STARNET yield more precise stellar parameters than the original model, even without adopting any spectroscopic mask or spectral windows.

5. CYCLE-STARNET can understand stellar astrophysics and uncover unknown spectral features. Testing on a mock data set, we illustrated that CYCLE-STARNET can recover the missing features in the Kurucz models and associate the missing features with the correct corresponding elements.

The philosophy of autocalibrating models with domain adaptation, exemplified by CYCLE-STARNET, is generic and can be applied to many other fields. Our results provide an entirely new path for extracting information from large, unlabeled data sets, harnessing advancements in machine learning to redefine what big-data astronomy can mean for stellar spectroscopy and beyond.

T.O. and S.B. acknowledge the support provided for a portion of this research by the Natural Sciences and Engineering Research Council of Canada (NSERC) Undergraduate Student Research Awards (USRA). Y.S.T. is supported by the NASA Hubble Fellowship grant HST-HF2-51425.001 awarded by the Space Telescope Science Institute. K.V. and S.B. acknowledge funding from the National Science and Engineering Research Council Discovery Grants program and the CREATE training program on New Technologies for Canadian Observatories.

Appendix A

**CYCLE-STARNET Architecture**

We summarize the architecture of each of the subnetworks in Table 2. Each row within the table shows a subsequent network layer, as well as the activation function (LeakyReLU, Sigmoid, InstanceNorm) applied after the layer operation. The CONV layers are the standard 1D-convolutional layers, the DCONV layers are deconvolutional (or transposed convolutional) layers, and FC denotes the fully connected layers. For each layer, N, K, and S represent, respectively, the number of filters (or nodes), the size of the filters (or kernel size), and the stride length of the convolutional operations.

Furthermore, instead of using the standard 7214 pixels from the ASPCAP reduction, we discard 47 pixels in the red chip to easily accommodate symmetrical down- and up-sampling in the convolutional networks. Thus, each spectrum consists of 7167 pixels. When processed through either $E_{\text{syn}}$ or $E_{\text{obs}}$, the intermediate representation is $64 \times 447$ dimensional. This intermediate representation is down-sampled by $E_{\text{sh}}$ to produce the shared latent representation, which is $25 \times 24$ dimensional. Lastly, $E_{\text{sp}}$ down-samples the intermediate representation to the split latent space, which has the shape of either $4 \times 24$ (Section 4.1) or $1 \times 24$ (Section 4.2).

On this note, the only difference between the architectures used in Sections 4.1 and 4.2 is that the split latent space has four filters in the former and one filter in the latter. Note that because Section 4.2 uses a mock data set for the observed

| Layer | $E_{\text{syn}}$ and $E_{\text{obs}}$ |
|-------|----------------------------------|
| 1     | CONV-(N32,K7,S4), LeakyReLU      |
| 2     | CONV-(N64,K7,S4), LeakyReLU      |

| Layer | $E_{\text{sh}}$ |
|-------|-----------------|
| 1     | CONV-(NI28,K7,S4), LeakyReLU |
| 2     | CONV-(N256,K7,S2), LeakyReLU |
| 3     | CONV-(N512,K7,S2), LeakyReLU |
| 4     | CONV-(N25,K1,S1), InstanceNorm |

| Layer | $D_{\text{sh}}$ |
|-------|-----------------|
| 1     | DCONV-(N32,K7,S2), LeakyReLU |
| 2     | DCONV-(N32,K7,S2), LeakyReLU |
| 3     | DCONV-(N32,K7,S4), LeakyReLU |

| Layer | $D_{\text{syn}}$ and $D_{\text{obs}}$ |
|-------|----------------------------------|
| 1     | DCONV-(N64,K7,S4), LeakyReLU      |
| 2     | DCONV-(N32,K7,S4), LeakyReLU      |
| 3     | CONV-(N1,K1,S1)                  |

| Layer | $C_{\text{syn}}$ and $C_{\text{obs}}$ |
|-------|----------------------------------|
| 1a    | CONV-(N16,K7,S4), LeakyReLU      |
| 2a    | CONV-(N32,K7,S4), LeakyReLU      |
| 3a    | CONV-(N64,K7,S4), LeakyReLU      |
| 4a    | CONV-(N128,K7,S4), LeakyReLU     |
| 5a    | CONV-(N256,K7,S4), LeakyReLU     |
| 1b    | CONV-(N32,K1,S1), LeakyReLU      |
| 2b    | CONV-(N64,K1,S1), LeakyReLU      |
| 3b    | CONV-(N128,K1,S1), LeakyReLU     |
| 4b    | CONV-(N256,K1,S1), LeakyReLU     |
| 5b    | CONV-(N512,K1,S1), LeakyReLU     |
| 6     | FC-(N1), Sigmoid                 |
domain, there are no other observation-specific variables. Therefore, in principle, there is no need for a split latent space. However, to keep the two architectures relatively consistent, we simply used fewer filters.

Appendix B
CYCLE-STARNET Training Details

CYCLE-STARNET is trained by optimizing the loss outlined in Equation (1). An essential aspect of having the network converge correctly is determining the correct combination of the $\lambda$ values in this equation, which control the influence of each loss term. Through various empirical tests, we determined that the values $\lambda_{\text{synth}} = 3$, $\lambda_{\text{obs}} = 9$, and $\lambda_{\text{adv}} = 1$ lead to satisfactory performance.

As discussed in Section 3.2.5, training is done iteratively; one iteration of training the autoencoders is computed, including the adversarial loss for the transfer mappings, followed by an iteration of training the critic networks. Each iteration used a batch of eight spectra from each domain. If we consider a single batch iteration as one optimization step for both processes, the network was trained for 350,000 batch iterations. The training was done with the Adam optimizer using a learning rate of 0.0001. Furthermore, this learning rate was decreased by a factor of 0.7 at 50,000, 100,000, 150,000, and 200,000 batch iterations. Figure B1 shows the training progress for the case study outlined in Section 4.1.

ORCID iDs

Yuan-Sen Ting (丁源 森) @ https://orcid.org/0000-0001-5082-9536

Sébastien Fabbro @ https://orcid.org/0000-0003-2239-7988

Figure B1. Training progress for the case study outlined in Section 4.1. The top panel reflects the distance losses in both the synthetic and observed domains for the training reconstructions and cycle reconstructions. The bottom panel shows the evaluation of the adversarial losses on the training set for both the generative processes ($X_{\text{synth}} \rightarrow \text{obs}$ and $X_{\text{obs}} \rightarrow \text{synth}$) and the critics (represented simply as $C_{\text{synth/obs}}(X_{\text{true/obs}})$).

Kim Venn @ https://orcid.org/0000-0003-4134-2042

References

Amarsi, A. M. 2015, MNRAS, 452, 1612

Amarsi, A. M., Asplund, M., Collet, R., & Leenaarts, J. 2016, MNRAS, 455, 3735

Aoki, W., Beers, T. C., Lee, Y. S., et al. 2013, AJ, 145, 13

Ballester, P., Modigliani, A., Boitquin, O., et al. 2000, Msgr, 101, 31

Bader, S., Asplund, M., Duong, L., et al. 2018, MNRAS, 478, 4513

Coelho, P. R. T., Brazuul, G., & Charlot, S. 2020, MNRAS, 491, 2025

Dalton, G., Trager, S., Abrams, D. C., et al. 2014, Proc. SPIE, 9147, 91470L

Deshpande, I., Zhang, Z., & Schwing, A. G. 2018, in Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (Salt Lake City, UT: IEEE Computer Society), 3483

Donahue, J., & Simonnyan, K. 2019, in Advances in Neural Information Processing Systems 32 (NeurIPS 2019), ed. H. Wallach et al. (Red Hook, NY: Curran Associates, Inc.), 10542

Fabbro, S., Venn, K. A., O’Brien, T., et al. 2018, MNRAS, 475, 2978

Gilmore, G., Randich, S., Asplund, M., et al. 2012, Msgr, 147, 25

Gonzalez-García, A., van de Weijer, J., & Bengio, Y. 2018, in Advances in Neural Information Processing Systems 31 (NeurIPS 2018), ed. H. Bengio et al. (Red Hook, NY: Curran Associates, Inc.), 1287

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., et al. 2014, in Advances in Neural Information Processing Systems 27 (NIPS 2014), ed. Z. Ghahramani et al. (Red Hook, NY: Curran Associates, Inc.), 2672

Grover, A., Chute, C., Shu, R., Cao, Z., & Ermon, S. 2020, in AAAI Conf. on Artificial Intelligence (Palo Alto, CA: AAAI Press), 4028

Guiglion, G., Matijevic, G., Queiroz, A. B. A., et al. 2020, A&A, 644, A168

Holtzman, J. A., Hasselquist, S., Shetrone, M., et al. 2018, AJ, 156, 125

Huang, X., Liu, M.-Y., Belongie, S., & Kautz, J. 2018, in Proc. European Conference on Computer Vision (ECCV) (Cham: Springer), 172

Jahandar, F., Venn, K. A., Shetrone, M. D., et al. 2017, MNRAS, 470, 4782

Kovalev, M., Bergemann, M., Ting, Y.-S., & Rix, H.-W. 2019, A&A, 628, A54

Kurucz, R. L. 1970, SAOSR, 309, 291

Kurucz, R. L. 1993, SYNTHE Spectrum Synthesis Programs and Line Data (Cambridge, MA: Smithsonian Astrophysical Observatory)

Kurucz, R. L. 2005, MSAIS, 8, 14

Kurucz, R. L. 2013, ATLAS10: Oacity Sampling Model Atmosphere Program, Astrophysics Source Code Library, ascl:1303.024

Kurucz, R. L., & Avrett, E. H. 1981, SAOSR, 391, 139

Leung, H. W., & Bovy, J. 2019, MNRAS, 483, 3255

Liu, M.-Y., Breuel, T., & Kautz, J. 2017, in Advances in Neural Information Processing Systems 30 (NIPS 2017), ed. I. Guyon et al. (Red Hook, NY: Curran Associates, Inc.), 700

Liu, M.-Y., Huang, X., Mallya, A., et al. 2019, in 2019 IEEE/CVF International Conf. on Computer Vision (ICCV) (New York: IEEE), 10551

Martíoli, E., Teppe, D., Manset, N., et al. 2012, Proc. SPIE, 8451, 84512B

Miyato, T., Kataoka, T., Koyama, M., & Yoshida, Y. 2018, arXiv:1802.05957

Ness, M., Hogg, D. W., Rix, H.-W., Ho, A. Y., & Zasowski, G. 2015, ApJ, 808, 16

Prugniel, P., Vauglin, I., & Koleva, M. 2011, A&A, 531, A165

Sharma, K., Prugniel, P., & Singh, H. P. 2015, A&A, 585, A64

Shetrone, M., Bizyaev, D., Lawler, J., et al. 2015, ApJS, 221, 24

Snedden, C., Cowan, J. J., & Gallino, R. 2008, ARA&A, 46, 241

Ting, Y.-S., Conroy, C., Rix, H.-W., & Asplund, M. 2018, ApJ, 860, 159

Ting, Y.-S., Conroy, C., Rix, H.-W., & Cargile, P. 2019, ApJ, 879, 69

Ting, Y.-S., Rix, H.-W., Conroy, C., Ho, A. Y., & Lin, J. 2017, ApJL, 849, L9

van der Maaten, L., & Hinton, G. 2008, J. Mach. Learn. Res., 9, 2579, http://neuralnets/krogh/9/van.dermaaten08a.html

Venn, K. A., Kielty, C. L., Sestito, F., et al. 2020, MNRAS, 492, 3241

Xiang, M., Ting, Y.-S., Rix, H.-W., et al. 2019, ApJS, 245, 34

Yanny, B., Rockosi, C., Newberg, H. J., et al. 2009, AJ, 137, 4377

Zhang, X., Zhao, G., Yang, C. Q., Wang, Q. X., & Zuo, W. B. 2019, PASP, 131, 094202

Zhang, X.-Y., Park, T., Isola, P., & Efros, A. A. 2017, in 2017 IEEE International Conference on Computer Vision (ICCV) (New York: IEEE), 2223