MIRKWOOD: Fast and Accurate SED Modeling Using Machine Learning

Sankalp Gilda1, Sidney Lower1, and Desika Narayanan1,2,3

1 Department of Astronomy, University of Florida, 211 Bryant Space Science Center, Gainesville, FL 32611, USA
2 University of Florida Informatics Institute, 432 Newell Drive, CISE Bldg E251 Gainesville, FL 32611, USA
3 Cosmic Dawn Centre at the Niels Bohr Institute, University of Copenhagen and DTU-Space, Technical University of Denmark, Denmark

Received 2021 January 7; revised 2021 April 30; accepted 2021 May 4; published 2021 July 23

Abstract

Traditional spectral energy distribution (SED) fitting codes used to derive galaxy physical properties are often uncertain at the factor of a few level owing to uncertainties in galaxy star formation histories and dust attenuation curves. Beyond this, Bayesian fitting (which is typically used in SED fitting software) is an intrinsically compute-intensive task, often requiring access to expensive hardware for long periods of time. To overcome these shortcomings, we have developed MIRKWOOD: a user-friendly tool comprising an ensemble of supervised machine-learning-based models capable of nonlinearly mapping galaxy fluxes to their properties. By stacking multiple models, we marginalize against any individual model’s poor performance in a given region of the parameter space. We demonstrate MIRKWOOD’s significantly improved performance over traditional techniques by training it on a combined data set of mock photometry of $z = 0$ galaxies from the SIMBA, EAGLE, and ILLUSTRISTNG cosmological simulations, and comparing the derived results with those obtained from traditional SED fitting techniques. MIRKWOOD is also able to account for uncertainties arising both from intrinsic noise in observations, and from finite training data and incorrect modeling assumptions. To increase the added value to the observational community, we use Shapley value explanations to fairly evaluate the relative importance of different bands to understand why particular predictions were reached. We envisage MIRKWOOD to be an evolving, open-source framework that will provide highly accurate physical properties from observations of galaxies as compared to traditional SED fitting.

Unified Astronomy Thesaurus concepts: Astronomy data analysis (1858); Astrostatistics (1882); Algorithms (1883); Extragalactic astronomy (506)

1. Introduction

1.1. Traditional SED Fitting

Spectral energy distributions (SEDs) of galaxies are crucial in informing our understanding of the most fundamental physical properties, such as their redshifts, stellar population masses, ages, and dust content. The task of extracting these properties by fitting models of the stellar photometry and dust attenuation to observations of galaxies is commonly known as SED fitting, and is one of the most common ways of deriving physical properties of galaxies. In SED fitting, stellar populations are evolved over an assumed star formation history (SFH), with an assumed stellar initial mass function (IMF) and through an assumed dust attenuation curve, to produce a composite spectrum. This resulting spectrum is then compared to the galaxy photometric observation under consideration, and the input assumptions are varied until there is a reasonable match between the two. The best-fit model is then used to infer the physical properties of the concerned galaxy (see recent reviews by Walcher et al. 2011; Conroy 2013).

This technique has been transformational in turning observed data into inferred galaxy physical properties. For instance, we now understand the evolution of the galaxy stellar mass function through $z \sim 7$ (Tomczak et al. 2014; Grazian et al. 2015; Leja et al. 2019b), the diversity in shapes of galaxy SFHs (Papovich et al. 2011; Leja et al. 2020), the nature of the star formation rate–stellar mass (SFR–$M^\star$) relation through $z = 2$ (Mobasher et al. 2008; Iyer et al. 2018), the process of inside-out quenching (Jung et al. 2017), and the general shapes of galaxy dust attenuation curves at high-$z$ (Salmon et al. 2016), all thanks to a combination of high-quality data with SED fitting techniques. These developments have spurred—and in turn have been accelerated—by the development of modern SED fitting codes such as CIGALE (Boquien et al. 2019), PROSPECTOR (Leja et al. 2017), FAST (Kriek et al. 2009), BAGPIPES (Carnall et al. 2018), and MAGPHYS (Da Cunha et al. 2008). These codes rely on Bayesian optimization and Markov Chain Monte Carlo sampling methods to explore the large input base of computer-simulated SED templates for close matches with an observed SED, and consequently output its physical properties.

At the same time, this traditional method of fitting SEDs has been known to introduce potential uncertainties and biases in the derived physical properties of galaxies. At the core of the issue are assumptions regarding the model stellar isochrones, spectral libraries, the IMF, the SFH, and the dust attenuation law. Each of these choices can dramatically impact the derived physical properties. For example, Acquaviva et al. (2015) evaluated the effects that different modeling assumptions have on the recovered SED parameters, and found that the incorrect parameterizations of the SFH can significantly impact the derived physical properties. This finding has been confirmed by the works of Wyots et al. (2009), Michałowski et al. (2014), Sobral et al. (2014), and Simha et al. (2014), who have all found significant impact of SFH on the derived galaxy stellar mass in various observational data sets. Pacifici et al. (2015) found that the simple assumption of exponentially declining SFH, simple dust law, and no emission-line contribution fails to recover the true (SFR–$M^\star$) relationship for star-forming galaxies. Similarly, Iyer & Gawiser (2017) and Lower et al. (2020) found that fitting the SFH using simple functional forms leads to a bias of up to 70% in the recovered total stellar mass.
In addition, Carnall et al. (2019) demonstrated that such simple functional forms for SFH also impose strong priors on other physical parameters, which in turn bias our inference for fundamental cosmic relations, such as the stellar mass function (SMF) and the cosmic star formation rate density (CSFRD; Ciesla et al. 2017; Leja et al. 2019a), critical for answering crucial questions in the study of galaxy formation and evolution. Even with more complex SFH functions which have been able to better reproduce the CSFRD (Gladders et al. 2013), individual stellar age estimates have been biased (Carnall et al. 2019; Leja et al. 2019b). In a similar vein to SFH modeling, in their recent review Salim & Narayanan (2020) found that incorrect assumptions of the dust attenuation law, even within the commonly accepted family of locally calibrated curves, can drive major uncertainties in the derived galaxy SFRs.

To overcome some of these limitations of traditional SED fitting, two broad approaches have been introduced. The first has been to modify the SFH prior employed during SED fitting. This has been accomplished by either creating new functional, parametric forms for the SFH that result in lower biases in derived posteriors in physical parameters, for instance by being less subject to the "outshining bias" (Behroozi et al. 2013; Simha et al. 2014), or by developing parameter-free (nonparametric) descriptions of the SFH (Tojeiro et al. 2007; Iyer & Gawiser 2017; Iyer et al. 2019; Leja et al. 2019a). The latter are those that allow for a number of flexible bins of star formation to vary in the SED modeling procedure in order to model the more complex star formation histories that likely represent real galaxies (Iyer et al. 2018; Leja et al. 2020). In Lower et al. (2020), the authors used a subset of the hydrodynamical simulations used in this work to study the biases introduced by various commonly used parametric forms of SFH on the derived stellar masses, and found that some of the most commonly used formulations—"constant," "delayed-τ," and "delayed-τ + burst"—dramatically under-predict the true stellar mass. They then compared the performance against that obtained by using a nonparametric SFH model (Leja et al. 2017), and found significantly better results. Even so, there were a large number of catastrophic outliers, with errors of up to 40% (see Figure 3 in their work). This is because SED fitting is inherently a complex task involving several moving parts, and SFH is only one of several model assumptions that go in it. As already mentioned, for example, the assumed dust attenuation law is another factor that can significantly impact the derived galaxy properties (see Figure 2 in Salim & Narayanan 2020).

The upshot is that the uncertainties in the galaxy SFH and star-dust geometry (as manifested in their dust attenuation curves) translate to significant uncertainties in the derived properties from traditional SED fitting techniques (Leja et al. 2017, 2019a; Lower et al. 2020; Salim & Narayanan 2020). An alternative to SED fitting is to instead derive a mapping between the photometry of a galaxy and the underlying physical properties. In this regard, machine learning (ML) techniques can serve as a potential alternative to traditional SED fitting. In this paper, we demonstrate that ML is not only a suitable alternative, but potentially a more accurate one.

1.2. Machine Learning and SED Fitting

To explore the complex and degenerate parameter space more efficiently, ML has emerged as an alternative tool for deriving the physical parameters of galaxies. Unlike traditional SED fitting procedures, including those utilizing novel nonparametric SFH priors, ML algorithms are able to directly learn the relationships between the input observations (either photometric of spectroscopic) and the galaxy properties of interest without the need for human-specified priors. With ML, the priors themselves become learnable. This is possible because ML algorithms learn from the entire population ensemble, learning not just the mapping between individual galaxy observations and physical properties, but also the distribution of properties.

Lovell et al. (2019) trained convolutional neural networks (CNNs; Krizhevsky et al. 2012) to learn the relationship between synthetic spectral galaxy and high-resolution SFHs from the EAGLE (Schaye et al. 2015) and ILLUSTRIS (Vogelsberger et al. 2014) simulations. Stenbos-Smidt et al. (2017) estimated specific SFRs and redshifts using broadband photometry from the Sloan Digital Sky Survey (SDSS; Eisenstein et al. 2011). Surana et al. (2020) used CNNs with multiband flux measurements from the Galaxy and Mass Assembly (GAMA; Driver et al. 2009) survey to predict galaxy stellar mass, SFR, and dust luminosity. Simet et al. (2019) used neural networks trained on semi-analytic catalogs tuned to the Cosmic Assembly Near-infrared Deep Extragalactic Legacy Survey (Grogin et al. 2011) survey to predict stellar mass, metallicity, and average SFR. Owing to their superior ability in capturing nonlinearity in data, ML-based models have seen applications in almost every field of astronomical study, including identification of supernovae (D’Isanto et al. 2016), classification of galaxy images (Barchi et al. 2020; Cheng et al. 2020, 2021; Ćiprjanović et al. 2020; Hausen & Robertson 2020), and categorization of signals observed in a radio SETI experiment (Harp et al. 2019). With the exponential growth of astronomical data sets, ML-based models are slowly becoming an integral part of all major data-processing pipelines (Siemiginowska et al. 2019). Baron (2019) provides a comprehensive and current overview of the application of ML methodologies to astronomical problems.

1.3. This Paper

In this paper we introduce MIRKWOOD, a fully ML-based tool for deriving galaxy physical parameters from multiband photometry, while robustly accounting for various sources of uncertainty. We compare MIRKWOOD’s performance with that of the current state-of-the-art Bayesian SED fitting tool PROSPECTOR, by using data from galaxy formation simulations for which we have ground-truth values. To do this, we first create mock SEDs from three different cosmological hydrodynamical simulations (via the procedure discussed in detail in Section 2 and illustrated in Figure 1)—SIMBA, EAGLE, and ILLUSTRIS-TNG. We then fit GALEX, HST, Spitzer, Herschel, and JWST photometry of these mock SEDs to extract stellar masses, dust masses, stellar metallicities, and instantaneous SFRs (defined as the mean SFR in the past 100 Myr from the time of observation) using the publicly available SED fitting software PROSPECTOR (Leja et al. 2017) with the state-of-the-art nonparametric SFH model from Leja et al. (2017, 2019b) (Section 2). We repeat this exercise using MIRKWOOD, and

---

4 While one could consider using observed photometric data of real galaxies instead, a glaring lack of well-defined ground-truth physical parameters—indeed, any inference tool employed—would make comparisons between results from MIRKWOOD and any traditional SED fitting code difficult.
compare both sets of derived galaxy properties to their true values from simulations. The rest of this paper is organized as follows. In Section 2 we describe both our data and the traditional SED fitting technique in significant detail, including the three hydrodynamical simulations (Section 2.1), the method to select galaxies of interest from these simulations (Section 2.2), the dust radiative transfer process used to create SEDs (Section 2.3), and finally the Bayesian code used to fit these SEDs to derive galaxy properties (Section 2.4). In Section 3 we describe in detail the SED modeling procedure using our proposed model MIRKWOOD. We describe how MIRKWOOD robustly estimates uncertainties in its predictions (uncertainty quantification, Section 3.1), and introduce metrics to quantify performance gains with respect to traditional SED fitting (Section 3.2). In Section 4, we investigate the improvement in the determination of the four galaxy properties. Finally, in Section 5 we summarize our findings, and suggest possible improvements to both the proposed model and to areas of application. Throughout we assume a Planck 2013 cosmology with the following parameters: $\Omega_m = 0.30$, $\Omega_{\Lambda} = 0.70$, $\Omega_b = 0.048$, $h = 0.68$, $\sigma_8 = 0.82$, and $n_s = 0.97$.

2. Training on Cosmological Galaxy Formation Simulations

The first step in a supervised ML code is to train the software on known results. To do this, we employ three state-of-the-art cosmological galaxy formation simulations where we know the true physical properties. We then generate mock SEDs from the galaxies in these simulations using stellar population synthesis models and dust radiative transfer; these mock SEDs allow our ML algorithms to develop a highly accurate mapping between input photometry and galaxy physical properties.

We train on a set of three galaxy formation simulations: SIMBA (Davé et al. 2019), EAGLE (Schaller et al. 2015; Schaye et al. 2015; McAlpine et al. 2016), and ILLUSTRIS-TNG (Vogelsberger et al. 2014). These models provide a large and diverse sample of galaxies with realistic growth histories, and free us from being dependent on any one set of galaxy model assumptions. In the following subsections, we describe each simulation and the galaxy formation models included in each suite as well as our galaxy selection method and computational techniques for generating mock SEDs for each galaxy.

2.1. Cosmological Galaxy Formation Simulations

SIMBA (Davé et al. 2019): The SIMBA galaxy formation model is a descendant of the MUFASA model, and uses the GIZMO gravity and hydrodynamics code (Hopkins 2015). SIMBA uses an H$_2$-based SFR, given by the density of H$_2$ divided by the local dynamical timescale. The H$_2$ density itself relies on the metallicity and local column density (Krumholz 2014). The chemical enrichment model tracks 11 elements from from Type II supernovae (SNe II), SNe Ia, and asymptotic giant branch (AGB) stars with yields following Nomoto et al. (2006), Iwamoto et al. (1999), and Oppenheimer & Davé (2006), respectively. Sub-resolution models for stellar feedback include contributions from SNe II, radiation pressure, and stellar winds. The two-component stellar winds adopt the mass-loading factor scaling from FIRE (Anglés-Alcázar et al. 2017) with wind velocities given by Muratov et al. (2015). Metal-loaded winds are also included, which extract metals from nearby particles to represent the local enrichment by the SNe driving the wind. Feedback via active galactic nuclei (AGNs) is implemented as a two-phase jet (high accretion rate) and radiative (low accretion rate) model, similar to the model used in ILLUSTRIS-TNG described below. Finally, dust is modeled self-consistently, produced by condensation of metals ejected from SNe and AGB stars, and is allowed to grow and erode depending on temperature, gas density, and shocks from local SNe II (Li et al. 2019, 2020). The tunable parameters in SIMBA were chosen to reproduce the $M_{\text{BH}}-\sigma$ relation and the galaxy SMF at redshift $z = 0$. For our analysis, we use a box with

Figure 1. Data pipeline for each simulated galaxy. Once galaxies are selected from each simulation, we perform 3D dust radiative transfer with POWDERDAY, which utilizes FSPS to generate the stellar spectral energy distributions (SEDs) and HYPERION to propagate the stellar SEDs through a dusty medium. We then sample mock photometry in 33 filters from the POWDERDAY SED.
$25 h^{-1}$ Mpc side length with 512$^3$ particles, resulting in a baryon mass resolution of $1.4 \times 10^9 M_\odot$.

EAGLE (Crain et al. 2015; Schaller et al. 2015; Schaye et al. 2015; McAlpine et al. 2016): The EAGLE suite of cosmological simulations is based on a modified version of the smoothed particle hydrodynamics code GADGET 3 (Springel 2005). Star formation occurs via gas particles that are converted into star particles at a rate that is pressure dependent and gas temperature and density limited, following Schaye (2004) and Schaye & Dalla Vecchia (2008). The chemical enrichment model tracks nine independent elements generated via winds from AGB stars, winds from massive stars, and SNe II following Wiersma et al. (2009), Portinari et al. (1998), and Marigo (2001), respectively. Stellar feedback from SNe II heats the local interstellar medium (ISM), delivering fixed jumps in temperature and a fraction of energy from the SNe, and similarly, a single feedback mode for AGN injects thermal energy proportional to the black hole (BH) accretion rate at a fixed efficiency into the surrounding gas. Like SIMBA, the parameters in EAGLE have been tuned to reproduce the $z = 0.1$ galaxy SMF. For our analysis, we use a $50 h^{-1}$ Mpc box size with 752$^3$ particles, resulting in a baryon mass resolution of $1.81 \times 10^9 M_\odot$.

IllustrisTNG (Weinberger et al. 2017; Pillepich et al. 2018a, 2018b): ILLUSTRISTNG is an updated version of the ILLUSTRIS project, based on the AREPO (Springel 2010) magnetohydrodynamics code. Star formation occurs in gas above a given density threshold, at a rate following the Kennicutt–Schmidt relation (Schmidt 1959; Kennicutt 1989). As stars evolve and die, nine elements are tracked via enrichment from SNe II and AGB stars following the models and yields of Wiersma et al. (2009), Portinari et al. (1998), and Nomoto et al. (2006). Star formation also drives galactic-scale winds. These hydrodynamically decoupled wind particles are injected isotropically with an initial speed that scales with the local 1D dark matter velocity dispersion. The galactic winds carry additional thermal energy to avoid spurious artifacts where the wind particles hydrodynamically recouple with the gas. Like SIMBA, ILLUSTRISTNG features a two-mode feedback model for AGNs: thermal energy is injected into the surrounding ISM at high accretion rates, while BH-driven winds are produced at low accretion rates. Tunable parameters were chosen in ILLUSTRISTNG to match observations including the $z = 0$ SMF, the SFR density, and the stellar mass–halo mass relation at $z = 0$. We use the TNG100 box with a baryonic mass resolution of $1.4 \times 10^9 M_\odot$.

2.2. Galaxy Selection

Galaxies from each simulation were identified with their respective friends-of-friends algorithms. We use the publicly available galaxy catalogs for EAGLE (McAlpine et al. 2016) and ILLUSTRISTNG (Nelson et al. 2019) in which subhalo structures are identified by a minimum of 32 particles with the SUBFIND algorithm (Springel et al. 2001; Dolag et al. 2009). For SIMBA, we have employed CAESAR5 (Thompson 2014), which identifies halos and galaxies in snapshots based on the number of bound stellar particles (a minimum of 32 particles defines a galaxy). To ensure our galaxy sample is robust across the three simulations, we select galaxies from the full subhalo populations that have (i) a stellar mass above the minimum mass found in the $z = 0$ SIMBA snapshot ($4.4 \times 10^7 M_\odot$) and (ii) have a nonzero gas mass. Due to computational limits, we randomly sample $\sim 10,000$ galaxies from the $\sim 85,000$ previously identified ILLUSTRISTNG galaxy sample, though we ensure that this subsample matches the intrinsic stellar mass function of the entire sample. After these selections, we have a total of 1797, 4697, and 10000 from the SIMBA, EAGLE, and ILLUSTRISTNG simulations, respectively, for redshift $z = 0$.

2.3. Mock SEDs from 3D Radiative Transfer

Our analysis relies on realistic mock galaxy SEDs as input to predict physical properties including stellar mass and SFR. To produce these synthetic SEDs, we perform full 3D dust radiative transfer (RT) for each galaxy in our subsamples. We use the RT code POWDERDAY6 (Narayanan et al. 2021) to construct the synthetic SEDs by first generating with FSPS (Conroy et al. 2009; Conroy & Gunn 2010) the dust-free SEDs for the star particles within each cell using the stellar ages and metallicities as returned from the cosmological simulations. For these, we assume a Kroupa (2002) stellar IMF and the MIST stellar isochrones (Choi et al. 2016; Dotter 2016). These FSPS stellar SEDs are then propagated through the dusty ISM. Here, differences in the RT model arise as dust is implemented differently in each simulation. EAGLE and ILLUSTRISTNG do not have a native dust model and so we assume a constant dust mass to metals mass ratio of 0.4 (Dwek 1998; Vladilo 1998; Watson 2011). For SIMBA, the diffuse dust content is described from the on-the-fly self-consistent model of Li et al. (2019). From there, the dust is treated identically between simulations: it is assumed to have extinction properties following the carbonaceous and silicate mix of Draine & Li (2007), that follows the Weingartner & Draine (2001) size distribution and the Draine (2003) renormalization relative to hydrogen. We assume $R_V = A_V/E(B-V) = 3.15$. We do not assume further extinction from sub-resolution birth clouds. Polycyclic aromatic hydrocarbons (PAHs) are included following the Robitaille et al. (2012) model in which PAHs are assumed to occupy a constant fraction of the dust mass (here, modeled as grains with size $a < 20 \, \mu m$) and occupying 5.86% of the dust mass. The dust emissivities follow the Draine & Li (2007) model, though are parameterized in terms of the mean intensity absorbed by grains, rather than the average interstellar radiation field as in the original Draine & Li model. The RT propagates through the dusty ISM in a Monte Carlo fashion using HYPERION (Robitaille 2011), which follows the Lucy (1999) algorithm in order to determine the equilibrium dust temperature in each cell. We iterate until the energy absorbed by 99% of the cells has changed by less than 1%. Note: a result of these calculations is that while we assume the Draine & Li (2007) extinction curve in every cell, the effective attenuation curve is a function of the star–dust geometry, and therefore varies from galaxy to galaxy (Salim & Narayan 2020).

The result of the POWDERDAY radiative transfer is the UV–FIR spectrum for each galaxy. From there, we sample broadband photometry in 33 filters, ensuring robust SED coverage. These filters are shown in Table 2. Our training set for MIRKWOOD includes photometry at signal-to-noise ratios (S/Ns) of 2, 10, and 20.

5 https://github.com/dnarayanan/caesar

6 https://github.com/dnarayanan/powderday
2.4. Prospector SED Fitting

We compare our results from MIRKWOOD to traditional galaxy SED fitting using the state-of-the-art SED fitting code PROSPECTOR (Leja et al. 2017, 2019a; Johnson et al. 2019). PROSPECTOR wraps FSPS stellar modeling with DYNESTY (Speagle 2020) Bayesian inference to provide estimates of galaxy properties given models forms for the galaxy SFH, metallicity, and dust content. We match the models for stellar isochrones, spectra, and IMF between the input POWDERDAY SEDs and PROSPECTOR as both rely on FSPS to model stellar populations and generate spectra. For the galaxy SFH, we use the nonparametric linear piece-wise function as described in Leja et al. (2019a) with six time bins spaced equally in logarithmic time. The stellar metallicity is tied to the inferred stellar mass of the galaxy, constrained by the Gallazzi et al. (2005) $M^* - Z$ relation from SDSS DR7 data. The dust attenuation model follows Kriek & Conroy (2013), in which

---

https://github.com/bd-j/prospector
the strength of the 2175 Å bump is tied to the powerlaw slope. Dust emission follows the Draine & Li (2007) model. Prior distributions for model components approximately follow those outlined in Lower et al. (2020) and have been shown to reasonably reproduce SIMBA galaxy properties including stellar mass and SFR.

We provide the same mock POWDERDAY broadband SEDs to PROSPECTOR as MIRKWOOD with the same S/Ns. In all comparisons to MIRKWOOD, we report the median value and 1σ width of the posterior distribution from the PROSPECTOR fit for each galaxy property.

3. Proposed Method

We now present implementation details of our proposed method. MIRKWOOD is a supervised learning algorithm, meaning it requires a training set consisting of pairs of inputs and outputs, to be able to learn the mapping from the former to the latter. Each input consists of a set of discrete features, or independent variables, whereas the outputs—also referred to as labels—are the dependent variables. The algorithm learns this mapping by minimizing a user-specified loss function. This is similar to the minimization of the mean-squared error commonly employed when fitting data in linear regression analysis. In this work, each input \( x \) is a mock SED vector consisting of flux values in 35 bands, and the associated galaxy properties (redshift, stellar mass, dust mass, stellar metallicity, and instantaneous SFR) form a vector output of length 5. While certain types of ML algorithms are able to map each input vector to an output vector consisting of multiple labels (for example, neural networks), this is not possible with MIRKWOOD. Hence we associate with each input only one galaxy property at a time, and use the model five times. Let us denote by \( x \) an input SED vector, by \( y \) a galaxy property, by \( n \) the number of SED samples, and by \( d \) the number of bands/filters. Then the data set on hand can be mathematically specified as \( D = \{ (x_i, y_i) ; i = 1, ... , n \} \). Our aim is to learn a data-driven mapping from the set of all \( x \), to the set of all \( y \), to be able to predict \( y_{\text{new}} \) for a new, unseen observation \( x_{\text{new}} \). To solve this regression problem, we employ the NGBOOST algorithm (Duan et al. 2019). Unlike most commonly used ML algorithms and libraries such as Random Forests (Breiman 2001), Randomized Trees (Geurts et al. 2006), XGBoost (Chen & Guestrin 2016), and Gradient Boosting Machines (Ke et al. 2017), NGBOOST enables us to easily work in a probabilistic setting and, corresponding to every input galaxy SED, output both a measure of the central tendency (i.e., \( \mu \)), and a measure of dispersion (i.e., \( \sigma \)) for the galaxy property of interest. We thus use NGBOOST as the backbone of MIRKWOOD, and use the hyperparameters suggested by Ren et al. (2019).8 We make the assumption that samples are drawn from

8. Classification tasks are those where the goal is to predict discrete labels, e.g., identifying whether an object is a hotdog or not. In regression problems we aim to prediction continuous labels, e.g., predicting the price of a stock tomorrow given its price over the past year.

9. Ren et al. (2019) found that their new set of hyperparameters result in faster convergence than the default values supplied by the developers at https://github.com/stanfordmlgroup/ngboost/blob/master/ngboost/ngboost.py.
Gaussian distributions. Our loss function is the negative likelihood function (Duan et al. 2019).

Consider the case where we have samples from all three simulations. As a reminder, we have 1797 samples from SIMBA, 4697 from EAGLE, and 10,073 from ILLUSTRIS. To predict any given galaxy property for each of the \( n = 1797 \) SIMBA galaxies, we proceed as follows. First, we fix the parameters of the NGBOST model to those recommended by Ren et al. (2019). Then, we divide the \( n \) samples into \( N_{\text{CV},1} = 5 \) distinct subsets (or folds) based on stratified sampling—\( y_i \) in each subset are chosen such that their distribution closely matches that of the parent sample \( y_i \). The first four subsets taken together form the training set, \( \mathcal{D}_{\text{TRAIN}} \); while the fifth, left-out fold for which we assume output labels are unavailable, is called the test set, \( \mathcal{D}_{\text{TEST}} \). This is the first step shown in Figure 2. We then sub-divide \( \mathcal{D}_{\text{TRAIN}} \) into \( N_{\text{CV},2} = 5 \) subsets, again in a stratified fashion. The model is trained cyclically on the combination of any distinct four sub-subsets—referred to as \( \mathcal{D}_{\text{TRAIN}_{\text{N}_{\text{bags}}}} \), while its predictions on the samples in the left-out sub-subset, \( \mathcal{D}_{\text{TRAIN}_{\text{N}}_{\text{bags}}} \), are recorded and stored. Instead of training directly on \( \mathcal{D}_{\text{TRAIN}_{\text{N}_{\text{bags}}}} \), we create \( N_{\text{bags}} = 48 \) “bags” by randomly shuffling the data and choosing the same number of samples with replacement. The NGBOST model initialized earlier is trained on all bags, and the respective predictions on \( \mathcal{D}_{\text{TRAIN}_{\text{N}_{\text{bags}}}} \) are averaged. The exact method of averaging is expanded upon in Section 3.1. After \( N_{\text{CV},2} \) rounds we have \( N_{\text{CV},2} \) non-overlapping \( \mathcal{D}_{\text{TRAIN}_{\text{N}_{\text{bags}}}} \) data sets containing distinct samples, which taken together constitute \( \mathcal{D}_{\text{TRAIN}} \). These together constitute the second and third steps in the pipeline. For each of the 1797 samples, we compare the predicted galaxy properties with their ground-truth values, and note the average negative-log likelihood, \( \mathcal{NLL}_{\text{VAL}} \). We repeat this exercise multiple times, on each iteration tuning NGBOST’s hyperparameters so that the \( \mathcal{NLL}_{\text{VAL}} \) is as low as possible. With the optimized set of NGBOST hyperparameters thus obtained, we train MIRKWOOD on the entire training set \( \mathcal{D}_{\text{TRAIN}} \); and record its predictions on \( \mathcal{D}_{\text{TEST}} \). This process of hyperparameter optimization is the fourth step in the pipeline. This process is repeated \( N_{\text{CV},1} = 1 \) more times, to obtain predictions on all \( n \) samples—this is the fifth and final step in the pipeline. The entire process is carried out in a chained fashion—the model is first trained to use galaxy flux values to predict stellar mass, then the predicted masses in conjunction with the original flux values are used to predict dust mass, and so on; this is explained pictorially in Figure 3. We know that galaxy properties can be strongly degenerate (for instance, the well-known age–reddening–metallicity degeneracy); by chaining, we “hack” our model to leverage the inter-dependencies between these parameters. To predict \( \text{SFR}_{100} \), the galaxy SFR average over the last 100 Myr, for an unseen sample \( x_{\text{new}} \); for instance, we will use NGBOST to first predict galaxy stellar mass, then galaxy dust mass, then stellar metallicity, and finally \( \text{SFR}_{100} \).

3.1. Uncertainty Quantification

Real observations differ from simulations in a number of ways.

1. It is rarely the case that observers have access to all filters in a survey for every observed galaxy. For example, one might encounter a situation where they are interested in determining a galaxy’s mass from its photometry, but do not have access to \( K \)- and \( H \)-band information, thus making their task difficult.

2. Different filters have differing sensitivities and noise properties, resulting in non-uniform measurement noises across observation bands. Even in an ideal case where observations in UV, optical, and IR are equally robust, there is an irreducible Poisson measurement error.

3. While modern hydrodynamical simulations have demonstrated a number of successes, there is an inherent, irreducible domain gap between their approximation of reality and actual reality. Imperfect modeling assumptions both about known physical processes, and arising from our sheer lack of knowledge of them, are primary contributors to this gap. Some specific reasons are modeling star formation with a sub-resolution model, divergence of modeled stellar mass functions from observed ones, and incomplete understanding of AGN feedback processes.

4. Any supervised ML algorithm is only as good as the data on which it is trained. It might well be the case that the observed galaxy is in a completely different parameter space than any of the galaxies in our training set.

5. Even if some particular model, say NGBOST, with a given set of hyperparameters, is able to faithfully and satisfactorily predict galaxy properties, it is very possible that there exist other models which can also make predictions that are just as accurate. It is also possible that for our chosen model/set of models, a different data preprocessing scheme, choice of hyperparameters (as mentioned in Section 3), the chosen set of hyperparameters is locally optimal but is not guaranteed to be globally optimal, or choice of scheme for balancing classes might result in better predictions.

Owing to these factors, a highly desirable behavior of any predictive model would be to not only return a point prediction (per galaxy property under consideration), but also some quantity conveying the model’s belief or confidence in its output (Kendall & Gal 2017; Choi et al. 2018). We can break down such predictive uncertainty into two parts—aleatoric and epistemic (Kendall & Gal 2017).10 Aleatoric uncertainty captures the uncertainty in the data-generating process—for example, measurement noise in filters, or the inherent randomness of a coin-flipping experiment. It cannot be reduced by collecting more training data,11 however, it can be reduced by collecting more informative features.12 On the other hand, epistemic uncertainty captures the ignorance of the predictive model, and can indeed be explained away given a sufficiently large training data set.13 Aleatoric uncertainty is thus irreducible or statistical, epistemic being its reducible or

10 The word epistemic comes from the Greek “episteme,” meaning “knowledge”; epistemic uncertainty is “knowledge uncertainty.” Aleatoric comes from the Latin “aleator,” meaning “dice player”; aleatoric uncertainty is the “dice player’s” uncertainty (Gal 2016).

11 We know that a fair coin has a 0.5 probability of landing heads, yet we cannot assert with 100% certainty the outcome of the next flip, no matter how many times we have already flipped the coin.

12 Uncertainty in galaxy mass estimation in the absence of \( H \)- and \( K \)-band information can be drastically reduced by collecting high-quality data in these bands.

13 If we are given a biased coin where we do not know the probability \( p \) of turning up heads, we can determine it to arbitrary precision by flipping the coin an arbitrarily large number of times and counting the number of times it turns up heads.
systematic cousin. Aleatoric uncertainty covers the first three of the five issues above, while epistemic uncertainty subsumes the last two points. High aleatoric uncertainty can be indicative of noisy measurements or missing informative features, while high epistemic uncertainty could be a clue that the observed galaxy is very different from any of the galaxies the model was trained on (such a galaxy is referred to as being out of distribution).

As previously discussed, the aim for a supervised ML algorithm is to learn the latent data-generating mechanism that maps each input sample \( x_i \) to its corresponding output label \( y_i \). This can be represented as follows:

\[
y_i = f(x_i) + \eta_i
\]

where \( f(\cdot) \) is the unknown latent function and a measurement error \( \eta \) follows a standard normal distribution with a variance \( \sigma_{\eta,i}^2 \), i.e., \( \eta_i \sim \mathcal{N}(0, \sigma_{\eta,i}^2) \). The variance of \( \eta_i \) corresponds to the aleatoric uncertainty, i.e., \( \sigma_{\eta,i}^2 = \sigma_{a,i}^2 \) where we will denote \( \sigma_{a,i}^2 \) as aleatoric uncertainty associated with the observation \( x_i \). Similarly, epistemic uncertainty will be denoted as \( \sigma_{e,i}^2 \). Since \( f(\cdot) \) is unknown, by training on the training set the model attempts to approximate, or guess, this function as closely as possible. Suppose that we train \( \hat{f}(x) \) to approximate \( f(x) \) from the samples in the input data set, \( \mathcal{D} \). Then, we can see that

\[
\mathbb{E}[||y_i - \hat{f}(x_i)||^2] = \mathbb{E}[||y_i - f(x_i) + f(x_i) - \hat{f}(x_i)||^2]
\]

\[
= \mathbb{E}[||y_i - f(x_i)||^2] + \mathbb{E}[||f(x_i) - \hat{f}(x_i)||^2]
\]

\[
= \sigma_{a,i}^2 + \sigma_{e,i}^2
\]

where \( \mathbb{E}||\cdot||^2 \) is the expectation operator. This indicates the total predictive variance is the sum of aleatoric uncertainty and epistemic uncertainty.
The NGBOOST algorithm that forms the backbone of MIRKWOOD is naturally able to output both the mean $\mu_i$ and the aleatoric uncertainty $\sigma_{ai}^2$ for a given observation $x_i$. To obtain the epistemic uncertainty $\sigma_{ei}^2$, we use bootstrapping. This involves creating multiples copies of the same training data set, each time picking samples with replacement, training the same model on all copies, and averaging the results. The goal is to simulate a scenario where a model only has access to limited samples—and hence limited parameter space—and to study how this impacts its predictive ability (see Figure 2, especially Step 3). That is, given an NGBOOST model, the predicted label for an observation $x_i$ is represented as

$$p(y|x_i) = N(\mu_i, \sigma_{ai}^2).$$

Creating $N_{\text{bags}}$ bootstrapped bags essentially creates $N_{\text{bags}}$ copies of the above equation. To obtain final summary statistics, we combine them as

$$\hat{\mu}_i = \sum_{j=1}^{N_{\text{bags}}} \mu_{ij}; \quad \hat{\sigma}_{ai}^2 = \sum_{j=1}^{N_{\text{bags}}} \sigma_{ai,j}^2; \quad \hat{\sigma}_{ei}^2 = \sum_{j=1}^{N_{\text{bags}}} (\mu_{ij} - \hat{\mu}_i)^2. \quad (1)$$

The upshot of this exercise is that the errors modeled by MIRKWOOD encapsulate both the propagated noise due in the input samples, and the uncertainty arising due to the finite size of the training set. The inclusion of the latter source of uncertainty represents a significant advancement over the current state-of-the-art (e.g., Acquaviva et al. 2015; Lovell et al. 2019), and helps mitigate the over-confidence in predictions that is typical of ML models.

### 3.2. Performance Metrics

As we have discussed, for each input sample MIRKWOOD outputs three values: $\mu$, $\sigma_{ai}$, and $\sigma_{ei}$. As the same time, for each input, PROSPECTOR outputs two values: $\mu$, and $\sigma$. We quantify the performance of each model by comparing the predicted galaxy properties against the true ones known from simulations. We divide this task into two parts—comparing just the predicted mean deterministic predictions of each galaxy property with its ground-truth value, and judging the quality of the entire predicted probability distribution function (probabilistic predictions) by comparing it against the ground-truth value.

#### 3.2.1. Metrics for Deterministic Predictions

For determining the “goodness” of the predicted means for the galaxy properties, we adopt the following three performance criteria:

(a) Predicted dust mass from MIRKWOOD vs. PROSPECTOR: as in Figure 4, we see that predictions from MIRKWOOD are a significant step-up from traditional SED-based results. (b) Predicted aleatoric uncertainty in dust mass from MIRKWOOD: unlike in Figure 4, we only plot the calibration curve for aleatoric uncertainties from MIRKWOOD, since in this work we do not have access to uncertainties in dust mass from PROSPECTOR. (c) Importance of different bands in predicting dust mass from MIRKWOOD: We see that the NIR bands are the most predictive of dust mass, with only a tiny contribution from predicted mass. This tracks with our understanding of dust mass properties.

The Astrophysical Journal, 916:43 (22pp), 2021 July 20 Gilda, Lower, & Narayanan
1. Normalized root-mean-square error (RMSE):

$$\text{NRMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (\hat{Y}(i) - Y(i))^2}.$$

2. Normalized mean absolute error (NMAE):

$$\text{NMAE} = \frac{1}{n} \sum_{i=1}^{n} |\hat{Y}(i) - Y(i)|.$$

3. Normalized bias error:

$$\text{NBE} = \frac{1}{n} \sum_{i=1}^{n} (\hat{Y}(i) - Y(i))$$

where $Y(i)$ and $\hat{Y}(i)$ are the true and predicted values of any of the five galaxy properties for the $i$th sample, while $n$ is the total number of samples. A perfect prediction would result in NRMSE, NMAE, and NBE of 0. These values can be seen in the top-left plots in Figures 4 through 7.

### 3.2.2. Metrics for Probabilistic Predictions

Probabilistic predictions are assessed by constructing confidence intervals (CIs) and using them in evaluation criteria. We note that our CIs are akin to the Bayesian credible intervals, and approximation of the frequentist confidence intervals. In this work, we adopt average coverage error (ACE) and interval sharpness (IS) as our metrics. $\alpha$ is the significance level—the probability that one will mistakenly reject the null hypothesis when in fact it is true. Specifically, since in this paper we are interested in $\pm 1\sigma$ predictions (CI = 68.2%), we choose $\alpha = 1 - \text{CI} = 0.318$. For MIRKWOOD results, $\sigma = \sqrt{\sigma_{\mu}^2 + \sigma_{\sigma}^2}$, while for PROSPECTOR results, there is only one $\sigma$ per prediction. For a given sample $i$, if the prediction of any galaxy property, from either MIRKWOOD or PROSPECTOR, is $\{\mu(i); \sigma(i)\}$, then the lower and upper confidence intervals $L_{\mu}(i)$ and $U_{\mu}(i)$ are given by $\mu(i) - \sigma(i)$ and $\mu(i) + \sigma(i)$, respectively. $L_{\sigma}(i)$ and $U_{\sigma}(i)$ correspond to the $u$-values for these: $l_{\sigma}(i) = 0.5 - \text{CI}/2 = 0.159$, $u_{\sigma}(i) = 0.5 + \text{CI}/2 = 0.841$. $y(i)$ corresponds to the $u$ – value of the ground-truth value $Y(i)$. If the predicted mean $\hat{Y}(i)$ for a galaxy property exactly matches $Y(i)$, then $y_{\alpha}(i) = 0.5$; if the predicted mean overshoots or undershoots the ground-truth value, then $y_{\alpha}(i) > 0.5$ or $y_{\alpha}(i) < 0.5$, respectively.

1. Average coverage error:

$$\text{ACE}_\alpha = \frac{1}{n} \sum_{i=1}^{n} c_{\alpha}(i) \times 100\% - 100 \times (1 - \alpha)\%,$$

where $c_{\alpha}$ is the indicative function of coverage:

$$c_{\alpha}(i) = \begin{cases} 1, & Y(i) \in [L_{\alpha}(i), U_{\alpha}(i)] \\ 0, & Y(i) \notin [L_{\alpha}(i), U_{\alpha}(i)]. \end{cases}$$

ACE is effectively the ratio of target values falling within the confidence interval to the total number of predicted samples.
2. Interval sharpness:

\[ \Delta \alpha(i) = \alpha(i) - \alpha(i) \]

where \( \Delta \alpha(i) = u_i(i) - l_i(i) \) is the uncertainty. As a reminder, corresponding to \( \pm 1\sigma \) intervals, \( \alpha = 0.318 \), \( l_i(i) = 0.159 \), and \( u_i(i) = 0.841 \).

From the equation of ACE, a value close to zero denotes a high reliability of prediction interval. As an infinitely wide prediction interval is meaningless, IS is another indicator contrary to the coverage rate of interval, which measures the accuracy of probabilistic forecasting. IS is always < 0; a large prediction interval obtains high reliability with a narrow width, which has a small absolute value of IS. For ease of reference, we summarize several acronyms and machine learning terms in Table 3.

4. Results

In order to test our proposed model for SED fitting, we compare against fits from the Bayesian SED fitting software PROSPECTOR. PROSPECTOR and MIRKWOOD are given the same information in order to infer galaxy properties. This includes broadband photometry in 33 bands with 5%, 10%, and 20% Gaussian uncertainties (respectively \( S/N = 20, 10, \) and 5). We present here the results from both methods for several galaxy properties including stellar mass, dust mass, SFR averaged over the last 100 Myr (SFR\(_{100}\)), and stellar metallicity.

In Figures 4 through 7, we present MIRKWOOD’S predictions when trained on the full data set comprising of simulations from SIMBA, EAGLE, and ILLUSTRITNG, for \( S/N = 5 \). The training set contains all 10,073 samples from ILLUSTRITNG, 4697 samples from EAGLE, and 359 samples from SIMBA selected by stratified five-fold cross-validation (CV) (see Section 3 for details on implementation), while the test set contains the remaining 1438 samples from SIMBA. After making inference on all test splits, we collate the results, thus successfully predicting all four galaxy properties for all 1797 samples from SIMBA. Each predicted output for a physical property contains three values— the mean \( \mu \), the aleatoric uncertainty \( \sigma_a \), and the epistemic uncertainty \( \sigma_e \). The first type of uncertainty quantifies the inherent noise in the data, while the second quantifies the confidence of the model in the training data (see Section 3.1 for a detailed description). The results from traditional SED fitting are a subset of the total training data. We present a brief overview of the data sets in Table 1.

In subplots (a) of Figures 4 through 7, we plot the recovered physical properties from MIRKWOOD and PROSPECTOR against the true values from the simulations for each of the four galaxy properties including stellar mass, dust mass, SFR averaged over the last 100 Myr (SFR\(_{100}\)), and stellar metallicity.

Corresponding plots for \( S/N = 10 \) and \( S/N = 20 \), along with our complete code, are available online at https://github.com/astrogilda/mirkwood.
modeled properties: stellar mass, dust mass, metallicity, and SFR. In each case, the blue contours represent the physical properties derived from traditional SED fitting, while the orange points denote the results from MIRKWOOD. Without exception, MIRKWOOD provides superior predictions, as evidenced both by visual inspection, and via a more quantitative approach by using three different metrics (detailed in Section 3.2).

In subplots (b), we plot “calibration diagrams” (Zelikman & Healy 2020). On the y-axis are the inverse cumulative distribution functions for predictions from both MIRKWOOD and PROSPECTOR. These are commonly used in ML literature—the plotting function takes as input a vector consisting of the ground-truth value, predicted mean, and predicted standard deviation, and outputs a value between 0 and 1. The plotting function takes as input a vector consisting of the ground-truth value, predicted mean, and predicted standard deviation, and outputs a value between 0 and 1. The 1:1 line on this curve indicates a perfectly calibrated set of predictions, with deviations farther from it indicative of poorer calibration. Both visually and quantitatively by using the two probabilistic metric (Section 3.2) ACE and IS, we ascertain that MIRKWOOD outperforms traditional SED fitting.

In subplots (c) and (d), we plot the histograms for predicted means, and Shapley value (SHAP) summary plots for samples in $D_{\text{TEST}}$, respectively. The histograms in subplots (c) enable us to visualize both bias and variance in predictions from MIRKWOOD, and compare the results to those from PROSPECTOR. As is readily apparent, both these quantities are significantly smaller for predictions from our ML-based model than from traditional SED fitting, for all four galaxy properties. We explain in detail how SHAP values are calculated in the Appendix. In brief, for a given filter (dotted vertical lines in subplots (d)), an absolute value farther from 0 denotes a larger impact of that feature in affecting the model outcome. SHAP values can thus be thought of as a type of sensitivity—they control how reactive the output is to changes in the input (here flux in the filter). Red and blue indicate high and low ends of values per feature/filter. Thus for a given feature, SHAP values >0 with a red hue indicate that the galaxy property increases as that feature increases. Similarly, SHAP values >0 with a blue hue indicate that the galaxy property decreases as the feature value decreases. As an example, in panel (d) of Figure 4, the largest correlations with the $M^*$ of a galaxy are, quite naturally, the ~1–2 $\mu$m range.

In Table 4, we delineate all five metrics from Section 3.2 to quantitatively compare MIRKWOOD’s performance with that of traditional SED fitting, across different levels of noise. As we would expect, the results from both models improve with increasing S/N. Across every metric and galaxy property, MIRKWOOD demonstrates excellent performance. We also note that the difference in their relative performances increases with

---

### Table 1
Summary Statistics of the Five Galaxy Properties in This Paper, for All Three Hydrodynamical Simulations

| Property                      | SIMBA | EAGLE | ILLUSTRATING |
|-------------------------------|-------|-------|--------------|
| $\log_{10}(M^*)$              |       |       |              |
| Min                           | 7.64  | 7.64  | 7.70         |
| Max                           | 12.15 | 12.01 | 12.19        |
| Mean                          | 8.91  | 8.71  | 8.82         |
| Median                       | 8.78  | 8.50  | 8.65         |
| Std. Dev.                    | 0.86  | 0.83  | 0.85         |
| $\log_{10}(1 + \frac{M_{\text{dust}}}{M})$ |       |       |              |
| Min                           | 0.0   | 2.81  | 4.72         |
| Max                           | 8.63  | 10.5  | 10.26        |
| Mean                          | 5.39  | 6.25  | 6.82         |
| Median                       | 5.33  | 6.06  | 6.72         |
| Std. Dev.                    | 1.47  | 0.86  | 0.75         |
| $\log_{10}(Z^*/Z_0)$         |       |       |              |
| Min                           | -1.61 | -1.05 | -1.27        |
| Max                           | 0.08  | 0.30  | 0.17         |
| Mean                          | -0.86 | -0.25 | -0.49        |
| Median                       | -0.87 | -0.24 | -0.50        |
| Std. Dev.                    | 0.35  | 0.18  | 0.28         |
| $\log_{10}(1 + \text{SFR}_{100})$ |       |       |              |
| Min                           | 0.0   | 0.0   | 0.0          |
| Max                           | 1.35  | 1.14  | 1.57         |
| Mean                          | 0.10  | 0.05  | 0.09         |
| Median                       | 0.02  | 0.01  | 0.03         |
| Std. Dev.                    | 0.18  | 0.11  | 0.15         |

#Samples                  | 1797  | 4697  | 10,073       |

---

### Table 2
Table of the 33 Filters Used to Extract Photometry from the Synthetic POWDERDAY Spectra

| Instrument | Filter | Effective Wavelength (AA) |
|------------|--------|---------------------------|
| GALEX      | FUV    | 1549                      |
|            | NUV    | 2304                      |
| HST/WFC3   | F814W  | 7977                      |
|            | F105W  | 10431                     |
|            | F125W  | 12364                     |
|            | F140W  | 13735                     |
|            | F160W  | 15279                     |
| Spitzer/IRAC | Ch1   | 35075                     |
| Herschel/PACS | Blue  | 689247                    |
|            | Green  | 979036                    |
|            | Red    | 1539451                   |
| JWST/NIRCam | F070W | 7007                      |
|            | F090W  | 8980                      |
|            | F115W  | 11487                     |
|            | F150W  | 14942                     |
|            | F200W  | 19791                     |
|            | F277W  | 27466                     |
| Spitzer/MIPS | 24 $\mu$m | 232096                   |
| Herschel/SPIRE | PSW   | 2428393                   |
|            | PMW    | 3408992                   |
|            | PLW    | 4822635                   |
increasing noise level. Below a certain S/N, traditional SED fitting stops producing meaningful results (as can also be seen clearly from subplots (a) in Figures 4 through 7), whereas our ML-based approach is still able to extract signal from the noisy data.

Finally, in Figures 8 through 11, we demonstrate the impact of different training sets on predictions from MIRKWOOD, for all three S/Ns of 5, 10, and 20. In all figures, “P” stands for results from PROSPECTOR, “SET” for results from MIRKWOOD when the training data set contains samples from ILLUSTRATING, EAGLE, and SIMBA, “ET” for EAGLE and ILLUSTRATING, “T” for only ILLUSTRATING, “E” for only EAGLE, and “S” for only SIMBA. For “SET” and “S,” we use stratified five-fold cross validation to select non-overlapping samples from SIMBA in the training set. As expected, we obtain the best predictions, as quantified by various deterministic and probabilistic metrics, when the training set consists of samples from all three simulations. This ensures the most diverse coverage in parameter space, and hence offers the highest generalization power. We also notice that prediction quality improves as the S/N improves, something that one would intuitively expect.

5. Discussion and Future Work

Our work shows that it is possible to reproduce galaxy property measurements with high accuracy in a purely data-driven way by using robust ML algorithms that do not have any knowledge about the physics of galaxy formation or evolution. The biggest caveat here is that the training set must be closely representative of the data of interest, i.e., observations of real

| Term | Meaning |
|------|---------|
| Bagging | Bootstrapped aggregation. A technique of building several models at a time by randomly sampling with replacement, or bootstrapping, from the original data set. Each bag may or may not be the same size as the parent training data set. |
| **HPO** | Hyperparameter optimization. The task of finding a well performing set of hyperparameters for any given ML algorithm. |
| **Chaining** | Given multiple outputs of interest (dependent variables), the technique of running the regression model in sequence to exploit correlations among them. |
| **Cross-validation** | The task of dividing the training set into non-overlapping “folds” for the purpose of hyperparameter optimization. |
| **D** | Input training set comprising of all samples from SIMBA, EAGLE, and ILLUSTRATION. Each sample consists of input (independent variables)-output (dependent variables) pairs. |
| **D_{TRAIN}** | Training set, derived from D. MIRKWOOD is trained on this, and used for inference on D_{TEST}. |
| **D_{TRAIN+TEST}** | Training set, derived from D_{TRAIN}. Used for HPO. |
| **D_{TRAIN+VAL}** | Validation set, derived from D_{TRAIN}. Used for HPO. |
| **D_{TEST}** | Test set. |
| **N_{CV,1}** | Number of CV folds used to produce D_{TRAIN+VAL} and D_{TEST} from D. |
| **N_{CV,2}** | Number of CV folds used to produce D_{TRAIN+TRAIN} and D_{TRAIN+VAL} from D_{TRAIN}. |
| **\( \sigma_e \)** | Epistemic uncertainty. |
| **\( \sigma_a \)** | Aleatoric uncertainty. |
| **\( f(\cdot) \)** | Functional representation of MIRKWOOD. This acts on a given input vector \( x \) to produce the output \( y \). |
| **\( \mu \)** | The mean of a predicted galaxy property, using one bag of training data. |
| **\( \tilde{\mu} \)** | The mean of \( \mu \) predicted by all \( N_{bags} \) bags. |

**Table 3**

Summary of Machine Learning Terms and Variables Appearing in This Work

**Table 4**

Comparative Performance of MIRKWOOD vs. PROSPECTOR across Different Noise Regimes

| S/N | NRMSE | NMAE | NBE | ACE | IS |
|-----|-------|------|-----|-----|----|
| Mass |
| 5 | (0.198, 1.003) | (0.123, 1.091) | (−0.042, −0.528) | (−0.002, −0.497) | (0.001, 0.005) |
| 10 | (0.165, 1.0) | (0.118, 1.088) | (−0.035, −0.518) | (−0.021, −0.502) | (0.001, 0.004) |
| 20 | (0.155, 1.002) | (0.115, 1.117) | (−0.041, −0.479) | (−0.066, −0.482) | (0.001, 0.033) |
| Dust Mass |
| 5 | (0.48, 0.996) | (0.339, 0.998) | (−0.219, −0.905) | (0.003, nan) | (0.001, nan) |
| 10 | (0.456, 0.996) | (0.332, 0.998) | (−0.209, −0.905) | (0.003, nan) | (0.001, nan) |
| 20 | (0.475, 1.263) | (0.336, 1.121) | (−0.215, −0.679) | (−0.076, nan) | (0.001, nan) |
| Metallicity |
| 5 | (0.062, 0.544) | (0.06, 0.478) | (−0.011, −0.297) | (−0.024, 0.046) | (0.041, 0.301) |
| 10 | (0.058, 0.534) | (0.055, 0.464) | (−0.01, −0.275) | (−0.032, 0.041) | (0.036, 0.295) |
| 20 | (0.056, 0.547) | (0.052, 0.487) | (−0.01, −0.229) | (−0.063, 0.036) | (0.032, 0.302) |
| SFR |
| 5 | (0.241, 0.907) | (0.205, 0.99) | (−0.069, −0.687) | (0.074, −0.557) | (7.314, 0.001) |
| 10 | (0.329, 0.91) | (0.226, 0.992) | (−0.09, −0.686) | (0.048, −0.564) | (1.937, 0.001) |
| 20 | (0.277, 1.988) | (0.215, 2.911) | (−0.078, 1.437) | (0.035, −0.547) | (0.006, 0.2) |

Note. The five metrics are the normalized root mean squared error (NRMSE), normalized mean absolute error (NMAE), normalized bias error (NBE), average coverage error (ACE), and interval sharpness (IS). In any given cell, the tuple of values consists of the metric of interest from MIRKWOOD and PROSPECTOR, respectively. A value of “nan” represents lack of predictions from PROSPECTOR. We do not have predicted error bars from PROSPECTOR for dust mass, hence ACE and IS values corresponding to this property are “nans”. For each tuple, the better value (i.e., lower error metric) is bolded.
We also demonstrate the necessity of carefully accounting for both the noise in the data and sparsity of the training set in any given parameter space of interest. By combining galaxy simulations with disparate physics, we offer a way to better train our model for a real observation. Across all S/Ns, we show a considerable improvement over state-of-the-art...
SED fitting using PROSPECTOR and a nonparametric SFH model, both in terms of prediction quality and compute requirements. For comparison, MIRKWOOD ran for $\sim 24$ hours on the first author’s home workstation with 12 cores, to derive all four galaxy parameters for $\sim 1700$ SIMBA galaxies; PROSPECTOR required about the same time, but 3000 cores on the University of Florida HiPerGator supercomputing system. We emphasize that for MIRKWOOD, the approximate 24 hour period included the time to train our ensemble of models, in addition to inference time for extracting galaxy properties of the out-of-fold samples; the inference time, by itself, was on the order of 1 minute for all four properties. Another notable result is that we are able to extract accurate galaxy properties from photometries with $S/N_s$ as low as 5. All these contributions considered together imply that scientifically interesting galaxy properties can be accurately and quickly
measured on future data sets from large-scale photometric surveys.

Several avenues for future research remain open. The crux of our follow-up efforts will focus on development of MIRKWOOD to a higher level of complexity and precision. We identify the following areas for improvement.

1. Extensive optimization. In this work we use NGBOOST models with hyperparameters only slightly modified from the prescription of Ren et al. (2019). This is an easy avenue for gaining performance boost with respect to reconstruction of galaxy properties; in future work we will leverage modern Bayesian hyperparameter optimization libraries to conduct extensive HPO to eke out maximum performance from our models.

2. Managing missing observations. Most real-life observations of all galaxies do not contain observations in the same set of filters, nor is the S/N of the data in each filter the same. Currently, MIRKWOOD cannot easily accommodate these complications, since NGBoost is unable to natively handle missing feature values. In future work we

Figure 10. Same as Figures 8 and 9, but for metallicity Z.
will explore other boosted tree models, such as CatBoost (Prokhorenkova et al. 2018), to extract galactic properties from real observations.

3. Synergistic spectroscopy. While spectroscopic data are considerably more expensive to obtain, there exist targets where photometry and spectra are available simultaneously (Carnall et al. 2019). Modeling both modes of data simultaneously can increase the S/N available to any model, and help significantly in reducing uncertainties in derived galaxy properties, thus enabling accurate reconstruction of higher-resolution ones such as a galaxy’s entire SFR. We will explore inclusion of a convolutional neural network to this end, similar to the work of Lovell et al. (2019).

4. Probability calibration. This is a post-processing step that has been shown to improve predictions, measured both by deterministic and probabilistic metrics. ML algorithms are often over-confident, and unable to assess the degree of their fallacy; if given a completely novel input, most commonly used ML algorithms are likely to predict a
wrong label for it, instead of just saying, “I do not know the correct answer.” Probability calibration is widely used in the ML community (Lakshminarayanan et al. 2017; Kuleshov et al. 2018; Nixon et al. 2019; Zelikman & Healy 2020), albeit its adoption in astronomy has been slow. We expect probability calibration would result in more sensible uncertainties in the second through fourth columns of Figures 8 through 11. Specifically, we expect that calibrated aleatoric, epistemic, and total uncertainties would mean the lowest ACE and IS values for the “SET” column, which is not consistently the case in this paper.

5. Randomized chaining. Here we have considered one particular way of chaining galaxy properties: only fluxes are used when predicting stellar mass, fluxes and predicted stellar mass are synergistically used to predict dust mass, and so on all the way until SFR_{100} (see Figure 3 for details). However, given the interdependence of all properties, a more justified approach would be to consider multiple such orderings and average the result. For instance, to predict stellar mass, model_1 will be trained on just the galaxy fluxes, model_2 on fluxes + metallicity, model_3 on fluxes + dust mass + SFR_{100}, and all such permutations. Finally, the predicted stellar mass results will be averaged. We expect this would average out any biases introduced by chaining in any particular order, and we plan to explore this line of research in more detail.

6. Redshifts as both input and output. We plan on expanding the training set considerably by incorporating photometries from galaxies over a range of redshifts to enable extraction of properties of real galaxies. In this sense, redshift would be the fifth output variable. On the other hand, it might very well be the case that for a given set of samples, high-confidence spectroscopic redshift is available, but photometry data are present only in a few bands. In such a case, it would be sensible to use redshift as an input rather than an output. We plan on upgrading our network so that MIRKWOOD can easily take a redshift as either a fixed input, or take a user-generated prior for redshift which it will then aim to improve.

7. Semi-supervised learning. In an effort to reduce the “domain gap”—the difference between the training set(s) and real-life data—several semi-supervised algorithms have been developed and used. These can smoothly combine a computer-simulated set of data with high-confidence real-life observations, such as spectra, to more closely approximate reality.

8. Outlier detection and elimination. In its current implementation, MIRKWOOD is unable to identify and reject outlying inputs. These can result from any of the several data-processing pipelines that are used to create scientifically interesting data products, which can fail and output nonsensical numbers, for example a flux absurdly high as 10^{52}. This issue becomes even more urgent when dealing with semi-supervised learning, where data from real-life observations form part of the training set. In future we plan on exploring algorithms such as variational auto-encoders (Zellner 1988; Ghosh & Basu 2016; Gilda et al. 2020) to this end.

5.1. On SED Modeling Priors and Assumptions

A fundamental aspect of SED modeling is the choice of model for each component (e.g., SFH or dust attenuation) and the subsequent priors used to constrain the models. In this work, we have used PROSPECTOR to fit the synthetic SEDs of galaxies from three cosmological simulations, following the model assumptions outlined in Lower et al. (2020). There, the model components, namely the nonparametric SFHs, were optimized to result in the best possible estimates for stellar mass, mass-weighted stellar age, and recent SFR for the SIMBA simulation at \( z = 0 \). However, it is conceivable that these optimizations do not hold true for EAGLE or ILLUSTRATION. For instance, in recent work by Iyer et al. (2020), the above cosmological simulations produce significantly different SFHs depending on the different subgrid models for, e.g., star formation and stellar feedback, which modulate the short- and long-timescale variations of a galaxy’s SFH. Thus, using the same SFH model to describe the galaxy SFHs from these simulations may not be justified, but it is ultimately outside the scope of this paper to fully tune the SFH models for each simulation. Similarly, in future work, we are working to implement physically motivated priors to further improve the results from traditional SED fitting by including variable dust attenuation curves.

Combining the input from three cosmological simulations as a training set for MIRKWOOD removes the simplifications of modeling galaxy SEDs with relatively simple analytic functions as in traditional SED modeling. This method of inferring galaxy properties can also overcome the biases that affect traditional SED fitting, such as the outshining of older stellar populations by younger, massive stars which generally causes stellar masses to be underestimated for star-forming galaxies (Papovich et al. 2001) and the generally poor reconstruction of early SFHs due to the non-uniform sensitivity to variations in star formation as a function of time (i.e., that SED fitting is more sensitive to recent star formation and much information about star formation beyond a few Gyr before the observation is lost) (Iyer et al. 2019).

Lastly, because the stellar models (isochrone, spectra, and IMF) in PROSPECTOR match the input SEDs from POWDER-DAY, it is important to note that if comparisons between the performance of MIRKWOOD and PROSPECTOR were made using real observations, a major uncertainty—the modeling of stellar populations—would be introduced into the results from PROSPECTOR. PROSPECTOR currently has the advantage in being able to choose from a variety of stellar models as well as numerous models for SED components including SFH and dust attenuation. Though MIRKWOOD has shown significant improvements in inferring galaxy physical properties, we discuss some of the caveats to our model related to the reliance on one particular set of models in the following section.

5.2. Caveats to the Mirkwood Training Set

Though MIRKWOOD is trained on a wealth of data from three moderate-volume cosmological simulations, we are not yet free from broad assumptions pertaining to stellar population synthesis modeling. MIRKWOOD relies on synthetic galaxy SEDs generated by post-process 3D dust RT modeling via HYPERION and FSPS. In practice, we have assumed a universal IMF for all galaxies as well as one library of stellar spectra and
isochrones. In contrast to traditional SED fitting, which inevitably suffers from uncertainties related to stellar modeling, our training set is tied to these particular models we have chosen above. To understand the magnitude of importance of these assumptions on inferring galaxy properties with MIRKWOOD, we would need to regenerate the synthetic galaxy SEDs for each combination of IMFs, spectral libraries, and isochrones, representing millions of hours of CPU time in addition to the time required to retrain MIRKWOOD on each set of SEDs for each redshift. Given the scale this problem requires, this retraining is ultimately outside the scope of this work.

S.G. is grateful to Sabine Bellstedt (GAMA collaboration, University of Western Australia) for several fruitful discussions, Viviana Aquaviva (Department of Physics, CUNY NYC College of Technology) for an in-depth review of the first draft, and Joshua Speagle (University of Toronto) for his helpful comments. This material is based upon work supported by the National Science Foundation under grant No. AST-1715206.

Appendix

Shapley Values

Consider a data set $X$ with $N$ features and $S$ samples $X = \{X_1^{(1)}, \ldots, X_N^{(S)}\}$, and a continuous, real-valued target variable vector $Y_{1:S}$. Given an instance (observation) $x \in X \in R^{1:N}$, a model $v$ forecasts the target as $v(x)$. In addition to this forecasting, we would also like to know if we can attribute the departure of the prediction $v(x)$ from the average value of the target variable $y = \bar{y}_{1:S}$ to each of the features (predictor variables) in the observed instance $x$. In a linear model, the answer is trivial: $\forall x \in X \in R^{1:N}, v(x) - \bar{y}_{1:S} = \beta_1 x_1^{(1)} + \ldots + \beta_N x_N^{(N)}$ (see Slide 13 in López de Prado 2020). We would like to do a similar local decomposition for any (nonlinear) ML model. Shapley values answer this attribution problem through game theory (Shapley 1953). Specifically, they originated with the purpose of resolving the following scenario: a group of differently skilled participants are all cooperating with each other for a collective reward; how should this reward be fairly divided among the group? In the context of ML, the participants are the features of our input data set $X$ and the collective payout is the excess model prediction (i.e., $v(x) - \bar{y}$). Here, we use Shapley values to calculate how much each individual feature marginally contributes to the model output—both for the data set as a whole (global explainability) and for each instance (local explainability).

While a goal of this paper is to explain the contributions of narrow and broad bands to various galaxy properties, for the sake of simplicity we choose to explain Shapley values using a toy example. Let us say that we operate a small hedge fund, and our team consists of three people: Alice, Bob, and Carol. On average, they manage to make approximately $100,000$ everyday ($\bar{y} = 100$ (thousand)). At the end of the year, we would like to distribute bonuses to our team members in direct proportion to their contribution to beating the average daily profit. Let us consider one day in particular, when owing partly to market volatility and partly to luck, today they make $120,000$ (i.e., excess income = 20 (thousand) dollars). Below, we delineate the process of determining each person’s fair contribution to this number.

1. First, we consider the $2^{N-3}$ possible coalitions (interactions) between the players (features), where some players may not participate (i.e., remain at their average value), and other may participate (depart from their average value). See Table 5.

2. Second, we use the coalitions table to compute the marginal contribution of each player conditional on the contribution of every other player. The marginal impact of changing $A$ after changing $B$ may differ from the marginal impact of changing $B$ after changing $A$. Accordingly, we must account for all possible $N! = 6$ sequences of conditional effects. See Table 6.

3. Finally, the Shapley value of a player is calculated as the average conditional marginal contribution of that player across all the possible $N!$ ways of conditioning the predictive model $v$. See the last row of Table 6 and Equation (A1).

Eliminating redundancy. As we can appreciate from the numerical exercise in Tables 5 and 6, some calculations are redundant. For example, the marginal contribution of $C$ on sequence $(A, B, C)$ must be the same as the marginal contribution of $C$ on sequence $(B, A, C)$, because (a) in both cases $C$ comes in third position, and (b) the permutations of $(A, B)$ do not alter that marginal contribution. The Shapley value of a feature $i$ can be derived as the average contribution of $i$ across all possible coalitions $S$, where $S$ does not include feature $i$:

$$\phi_i(v) = \sum_{S \subseteq \mathcal{N} \setminus \{i\}} \frac{|S|!(|N| - |S| - 1)!}{|N|!} \left(v(S \cup \{i\}) - v(S)\right).$$

(A1)

The above equation describes a coalitional game (the scenario described previously) with a set $\mathcal{N}$ of $|N|$ players. The function $v$ gives the value (payout) for any subset of those players. For example, if $S$ be a subset of $\mathcal{N}$, then $v(S)$ gives us the value of that subset. Thus for a coalitional game $(\mathcal{N}, v)$ Equation (A1) outputs the value for feature $i$, i.e., its Shapley value.\footnote{Equation (A1) computes the exact Shapley values by grouping the marginal conditional contributions in terms of coalitions ($S$). For large $N$, Lundberg & Lee (2017) and Lundberg et al. (2020) have developed fast algorithms for the estimation of $\phi_i$.} For ease of explanation, we focus our attention on calculating how many of the excess 20 (thousand) dollars can be attributed to Carol, i.e., calculating the Shapley value for Carol. We then examine the various components of Equation (A1) and study their significance.

1. If we relate this back to the parameters of the Shapley value formula we have $\mathcal{N} = \{A, B, C\}$ and $i = C$. The highlighted section in Equation (A2) (derived from Equation (A1) by rearranging some terms) says that we need to take our group of people and exclude the person that we are focusing on now. Then, we need to consider all of the possible subsets that can be formed. So if we exclude C from the group we are left with A, B. From this remaining group we can form the following subsets (i.e., these are the sets that C can take on): $\emptyset, A, B, AB$. In total, we can construct four unique subsets from the
We can use the interactions table to derive the marginal contribution of each feature in each sequence. The Shapley values are the averages per column.

**Table 5**

| Alice ≠ Alice | Bob ≠ Bob | Carol ≠ Carol | v(...) = v(x...) − y |
|---------------|-----------|---------------|----------------------|
| 0             | 0         | 0             | v(∅) = v(x[000]) − y = 0 |
| 0             | 0         | 1             | v(C) = v(x[001]) − y = 10 |
| 0             | 1         | 0             | v(B) = v(x[010]) − y = 5 |
| 0             | 1         | 1             | v(B, C) = v(x[011]) − y = 7 |
| 1             | 0         | 0             | v(A) = v(x[100]) − y = 2 |
| 1             | 0         | 1             | v(A, C) = v(x[101]) − y = 8 |
| 1             | 1         | 0             | v(A, B) = v(x[110]) − y = 10 |
| 1             | 1         | 1             | v(A, B, C) = v(x[111]) − y = 20 |

**Note.** For each interaction, we compute the departure of the model’s forecast from its baseline (average value). We encode as “1” a feature that is not at its average value (it forms part of a coalition), and “0” a feature that is at its average value. Thus v(A, B) = v(B, A), v(B, C) = v(C, A), v(A, C) = v(C, A), and v(A, B, C) = v(B, A, C) = v(A, C, B).

**Table 6**

| Combination | Alice | Bob | Carol | Total |
|-------------|-------|-----|-------|-------|
| A, B, C     | v(A)  | v(A)−v(∅) = 2 | v(A, B)−v(A) = 8 | v(A, B, C)−v(A, B) = 10 | 20 |
| A, C, B     | v(A)  | v(A)−v(∅) = 2 | v(A, C)−v(A, C) = 12 | v(A, C)−v(A) = 6 | 20 |
| A, B, C     | v(B, A)−v(B) = 5 | v(B)−v(∅) = 5 | v(B, C)−v(B) = 2 | 20 |
| B, C, A     | v(C, A)−v(C) = 3 | v(C, A)−v(C, A) = 12 | v(C)−v(∅) = 10 | 20 |
| C, B, A     | v(C, B)−v(C) = 3 | v(C)−v(∅) = 10 | 20 |
| Average     | 6.3   | 6.5 | 8     | 20 |

**Note.** We can use the interactions table to derive the marginal contribution of each feature in each sequence. The Shapley values are the averages per column.

remaining team members, including the null set

\[ \phi_1(v) = \frac{1}{|N|} \sum_{S \subseteq N \setminus \{i\}} \left( \frac{|N| - 1}{|S|} \right)^{-1} (v(S \cup \{i\}) - v(S)). \]  

(2.2)

2. Next, we focus on the highlighted term in Equation (A3)

\[ \phi_1(v) = \frac{1}{|N|} \sum_{S \subseteq N \setminus \{i\}} \left( \frac{|N| - 1}{|S|} \right)^{-1} (v(S \cup \{i\}) - v(S)). \]  

(A3)

This refers to the marginal value of adding player i to the game. Specifically, we want to see the difference in the money earned daily if we add C to each of our four subsets. We denote these four marginal values as: \( \Delta V_{C, B}, \Delta V_{A, C, B}, \Delta V_{B, C, A}, \Delta V_{C, B, A} \). Each of these is a different scenario that we need to observe in order to fairly assess how much C contributes to the overall profit. This means that we need to observe how much excess money is produced if everyone is working at their average levels (i.e., the empty set ) and compare it to what happens if we only have C working at above- or below-average productivity. We also need to observe how much profit is earned by A and B working simultaneously and compare that to the profit earned by A and B together with C, and so on.

3. Next, the summation in the Shapley value equation is telling us that we need to add all them together. However, we also need to scale each marginal value before we do that, for which we are provided this prescription by the highlighted part in Equation (A4)

\[ \phi_1(v) = \frac{1}{|N|} \sum_{S \subseteq N \setminus \{i\}} \left( \frac{|N| - 1}{|S|} \right)^{-1} (v(S \cup \{i\}) - v(S)). \]  

(A4)

This calculates how many permutations of each subset size we can have when constructing it out of all remaining team members excluding feature (player) i. In other words, given \( |N| - 1 \) features (players), how many groups of size \( |S| \) can one form with them? We then use this number to divide the marginal contribution of feature (player) i to all groups of size \( |S| \). For our scenario, we have that \( |N| - 1 = 2 \), i.e., these are the remaining team members when we are left with when calculating the Shapley value for C. In our case we will use that part of the equation to calculate how many groups we can form of size 0, 1, and 2, since those are only group sizes we can construct with the remaining players. So, for example, if we have that \(|S| = 1\) then we get that we can construct two different groups of this size: A and B. This means that we should apply the following scaling factors to each of our four marginal values: \( \Delta V_{C, B}, \Delta V_{A, C, B}, \Delta V_{B, C, A}, \Delta V_{C, B, A} \). By adding this scaling factor we are averaging out the effect that the rest of the team members have for each subset size. This means that we are able to capture the average marginal contribution of C when added to a team of size 0, 1, and 2 regardless of the composition of these teams.

4. Next, we focus on the remaining term, highlighted in Equation (A5)

\[ \phi_1(v) = \frac{1}{|N|} \sum_{S \subseteq N \setminus \{i\}} \left( \frac{|N| - 1}{|S|} \right)^{-1} (v(S \cup \{i\}) - v(S)). \]  

So far we have averaged out the effects of the other team members for each subset size, allowing us to express how much C contributes to groups of size 0, 1, and 2. The final piece of the puzzle is to average out the effect of the
group size as well, i.e., how much does $C$ contribute regardless of the size of the team (number of explanatory features in the data set). For our scenario we do this by dividing by 3 since that is the number of different group sizes that we can consider. With this final step, we arrive at the point where can finally compute the Shapley value for $C$. We have observed how much she marginally contributes to all different coalitions the team that can be formed. We have also averaged out the effects of both team member composition as well as team size which finally allows us to compute the Shapley value for $C$:

$$
\phi_C(v) = \frac{1}{3} \sum \left( \frac{1}{2} \Delta v_{C,A} + \frac{1}{2} \Delta v_{BC,A} + 1 \Delta v_{ABC,AB} \right)
$$

$$
= \frac{1}{3} \left( 1 \times 10 + 2 \times 6 + 2 \times 2 + 1 \times 1 \right)
$$

$$
= 8.
$$

(A6)

Comparing Equation (A6) with the Shapley value for $C$ in the last row of Table 6, we see that they match exactly.

After calculating the Shapley values for the rest of the players, we can then determine their individual contributions to the excess 20 (thousand) dollars earned today, allowing us to fairly divide the bonus among all team members:

$$
20 = v([A, B, C]) = \phi_A(v) + \phi_B(v) + \phi_C(v).
$$

(A7)

This is also evident from the last row of Table 6, where the Shapley values for $A$, $B$, and $C$ add to the total of 20.

Properties. As they apply to local explanations of predictions from ML models, Shapley values have some nice uniqueness guarantees (Shapley 1953). As described above, Shapley values are computed by introducing each feature, one at a time, into a conditional expectation function of the model’s output, $v(S) \approx E[v(x)|x_S]$, and attributing the change produced at each step to the feature that was introduced, then averaging this process over all possible feature orderings. Shapley values are provably the only possible method in the broad class of additive feature attribution methods (Lundberg & Lee 2017) that simultaneously satisfy four important properties of fair credit-assignment: efficiency, symmetry, missingness, and additivity.

1. Efficiency/Local accuracy. The sum of the Shapley values of all features (players) equals the value of the total coalition. That is, the value of 20 in the bottom-rightmost corner in Table 5 is the same as the 20 in the bottom-rightmost cell in Table 6.

2. Missingness. If a feature (player) never adds any marginal value regardless of the coalition, its payoff (Shapley value $\phi_i(v)$) is 0. That is, if: $v(S \cup \{i\}) = v(S \setminus i \forall$ subsets of features $S$, then $\phi_i(f) = 0$.

3. Symmetry. All players have a fair chance to join the game. That is why Table 5 lists all the permutations of the players. If two players always add the same marginal value to any subset to which they are added, their payoff portion should be the same. In other words, $\phi_i = \phi_j$ iff feature $i$ and feature $j$ contribute equally to all possible coalitions.

4. Additivity. A function with combined outputs has as Shapley values the sum of the constituent ones. For any pair of games $v$, $w$, $\phi(v+w) = \phi(v) + \phi(w)$, where $(v+w)(S) = v(S) + w(S)v/S$. This property allows us to combine Shapley values from $n$ different models, thus enabling interpretability for ensembles.

References

Acquaviva, V., Raichoor, A., & Gaviser, E. 2015, ApJ, 804, 8
Anglés-Alcázar, D., Faucher-Giguère, C.-A., Quataert, E., et al. 2017, MNRAS, 472, L109
Barchi, P., de Carvalho, R., Rosa, R., et al. 2020, A&C, 30, 100334
Baron, D. 2019, arXiv:1904.07248
Behroozi, P. S., Wechsler, R. H., & Conroy, C. 2013, ApJ, 770, 57
Boquien, M., Burgarella, D., Roehlly, Y., et al. 2019, A&A, 622, A103
Breiman, L. 2001, Mach. Learn., 45, 5
Carnall, A., McLure, R., Dunlop, J., & Davé, R. 2018, MNRAS, 480, 4379
Carnall, A., McLure, R., Dunlop, J., et al. 2019, MNRAS, 490, 417
Chen, T., & Guestrin, C. 2016, Proc. 22nd ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (New York, NY: Association for Computing Machinery), 785
Cheng, T.-Y., Conselice, C. J., Aragón-Salamanca, A., et al. 2020, MNRAS, 493, 4209
Cheng, T.-Y., Huertas-Company, M., Conselice, C. J., et al. 2021, MNRAS, 503, 4446
Choi, J., Dotter, A., Conroy, C., et al. 2016, ApJ, 823, 102
Choi, S., Lee, K., Lim, S., & Oh, S. 2018, in 2018 IEEE Int. Conf. on Robotics and Automation (ICRA), IEEE (Piscataway, NJ: IEEE), 6915
Ciesla, L., Elbaz, D., & Fensch, J. 2017, A&A, 608, A41
Ciprijanović, A., Snyder, G., Nord, B., & Peek, J. E. 2020, A&C, 32, 100390
Conroy, C. 2013, ARA&A, 51, 393
Conroy, C., & Gunn, J. E. 2010, ApJ, 712, 833
Conroy, C., Gunn, J. E., & White, M. 2009, ApJ, 699, 486
Crain, R. A., Schaye, J., Bower, R. G., et al. 2015, MNRAS, 450, 1937
Da Cunha, E., Charlot, S., & Elbaz, D. 2008, MNRAS, 388, 1595
Davé, R., Anglés-Alcázar, D., Narayanan, D., et al. 2019, MNRAS, 486, 2827
D’Isanto, A., Cavuoti, S., Brescia, M., et al. 2016, MNRAS, 457, 3119
Dolag, K., Borgani, S., Murante, G., & Springel, V. 2009, MNRAS, 399, 497
Dotter, A. 2016, ApJS, 222, 8
Draine, B. T. 2003, ARA&A, 41, 241
Draine, B. T., & Li, A. 2007, ApJ, 657, 810
Driver, S. P., Norberg, P., Baldry, I. K., et al. 2009, A&G, 50, 512
Duan, T., Avati, A., Ding, D., et al. 2019, arXiv:1910.03225
Dwek, E. 1998, ApJ, 501, 643
Eisenstein, D. J., Weinberg, D. H., Agol, E., et al. 2011, AJ, 142, 72
Gal, Y. 2016, PhD Dissertation, University of Cambridge
Gallazzi, A., Charlot, S., Brinchmann, J., White, S. D. M., & Tremonti, C. A. 2005, MNRAS, 362, 41
Geurts, P., Ernst, D., & Wehenkel, L. 2006, Mach. Learn., 63, 3
Ghosh, A., & Bauu, A. 2016, ASMM, 68, 413
Gilda, S., Ting, Y.-S., Wittington, K., et al. 2020, arXiv:2011.03132
Gladders, M. D., Oemler, A., Dressler, A., et al. 2013, ApJ, 770, 64
Grazian, A., Fontana, A., Santini, P., et al. 2015, A&A, 575, A96
Grogin, N. A., Kocevski, D. D., Faber, S., et al. 2011, ApJS, 197, 35
Harp, G. R., Richards, J., Tarter, S. S. J., et al. 2019, arXiv:1902.02426
Hausen, R., & Robertson, B. E. 2020, ApJS, 248, 20
Hepburn, C., Furlanetto, S. R., & Itagaki, Y. 2019, MNRAS, 490, 430
Johnson, B. D., Leja, J. L., Conroy, C., & Speagle, J. S. 2019, Prospector: Stellar Population Inference from Spectra and SEDs, Astrophysics Source Code Library, ascl:1905.025
Ke, G., Meng, Q., Finley, T., et al. 2017, in Proc. of the 31st Int. Conf. on Neural Information Processing Systems (Red Hook, NY: Curran Associates, Inc.), 3146

ORCID iDs

Sankalp Gilda @ https://orcid.org/0000-0002-3645-4501
Sidney Lower @ https://orcid.org/0000-0003-4422-8595
Desika Narayanan @ https://orcid.org/0000-0002-7064-4309
