ArtPop: A Stellar Population and Image Simulation Python Package

JOHNNY P. GRECO1,∗ AND SHANY DANIELI2,3,†

1Center for Cosmology and AstroParticle Physics (CCAPP), The Ohio State University, Columbus, OH 43210, USA
2Department of Astrophysical Sciences, 4 Ivy Lane, Princeton University, Princeton, NJ 08544
3Institute for Advanced Study, 1 Einstein Drive, Princeton, NJ 08540, USA

Submitted to ApJ

ABSTRACT

We present Artificial Stellar Populations (ArtPop), an open-source Python package for synthesizing stellar populations and generating artificial images of fully populated stellar systems. The code is designed to be intuitive to use and as modular as possible, making it possible to use each of its functionalities independently or together. ArtPop has a wide range of scientific and pedagogical use cases, including the measurement of detection efficiencies in current and future imaging surveys, the calculation of integrated stellar population parameters, quantitative comparisons of isochrone models, and the development and validation of astronomical image processing algorithms. In this paper, we give an overview of the ArtPop package, provide simple coding examples to demonstrate its implementation, and present results from some potential applications of the code. We provide links to the source code that created each example and figure throughout the paper. ArtPop is under active development, and we welcome bug reports, feature requests, and code contributions from the community.

Keywords: methods: observational, methods: statistical, stars: general, galaxies: stellar content, galaxies: dwarf, galaxies: distances and redshifts, techniques: image processing

1. INTRODUCTION

The physical properties of galaxies are almost entirely derived from observations of their light. At distances too large for resolving galaxies into individual stars, stellar population synthesis modeling of the spectral energy distribution may be used to infer integrated properties such as the total stellar mass, star formation rate, and metal content (Walcher et al. 2011; Conroy 2013, and references therein). In the nearby universe, on the other hand, it is possible to study galaxies using observations of individual stars. Star formation histories (SFHs), for instance, can be measured from Hubble Space Telescope (HST)-based resolved star color-magnitude diagrams (CMDs; e.g., Weisz et al. 2008), though such measurements are resource intensive and are only possible for a very small subset of the galaxy population.

In between these two extremes lies the semi-resolved regime. The statistical “lumpiness” observed in relatively nearby galaxies, which are unresolvable due to stellar crowding, contains a plethora of information about the underlying stellar populations. For instance, the sparse number of red giants relative to main sequence stars produces surface brightness fluctuations (SBFs) between pixels, which are well-known as an accurate extragalactic distance indicator (Tonry & Schneider 1988), as well as a sensitive probe of stellar populations (e.g., Buzzoni 1993; Ajhar & Tonry 1994; Cantiello et al. 2003; Raimondo et al. 2005). Taking this idea even further, fluctuation spectroscopy (van Dokkum & Conroy 2014) provides a powerful method for studying metal-rich giants in massive elliptical galaxies, and pixel CMDs (Conroy & van Dokkum 2016) generalize the SBF technique and are sensitive to SFHs, distances, and metallicities, as demonstrated in Cook et al. (2019, 2020).

Current and future wide-field imaging surveys will revolutionize our view of extragalactic stellar systems by producing large sets of deep and high resolution images. Indeed, surveys like the Hyper Suprime-Cam (HSC) Subaru Strategic Program (Aihara et al. 2018) and the Dark Energy Survey (DES; Abbott et al. 2018) are already producing large catalogs of low surface brightness galaxies over large areas of the sky (Greco et al. 2018;
Tanoglídís et al. 2021), an effort that will be continued in earnest by the Vera Rubin Observatory Legacy Survey of Space and Time (LSST; Ivezić et al. 2019). From space, the Nancy Grace Roman Space Telescope (Spergel et al. 2015) will deliver HST-quality imaging over wide areas, which will significantly increase the number of systems for which it will be possible to study stellar populations in both the resolved and semi-resolved regimes.

Artificial image simulations have long been an invaluable tool in the preparation for and exploitation of unprecedented data sets. To demonstrate how the then-hypothetical HST might supplement ground-based studies, Bahcall & Schneider (1988) simulated HST observations of a nearby globular cluster. The same simulation software was later used in Tonry & Schneider (1988) to quantify the feasibility and limitations of the SBF distance measurement method. Today, similar star-by-star image simulations have become the gold standard for measuring point source completeness and modeling resolved stellar populations (e.g., Stetson 1987; Dolphin 2000; Dalcanton et al. 2009).

In this work, we present Artificial Stellar Populations (ArtPop), an open-source Python package for modeling stellar populations and generating their corresponding images. ArtPop was conceptually introduced in Danieli et al. (2018), and it soon proved useful in helping to confirm the distance to the “galaxy lacking dark matter” (van Dokkum et al. 2018). ArtPop was then generalized, expanded upon, and used in Greco et al. (2021) to study SBFs in low-luminosity stellar systems.

The goal of ArtPop is to provide the community with a user-friendly tool for generating artificial images of resolved and semi-resolved stellar systems. It complements more comprehensive packages like GalSim (Rowe et al. 2015), which tend to have steep learning curves and have most often been applied to studies of unresolved galaxies. It is our hope that ArtPop will be useful both for scientific applications and in the classroom, where it can provide students with a unique perspective of stellar populations and their image formation.

The paper is organized as follows. In Section 2, we present our methods for synthesizing stellar systems and generating artificial images. In Section 3, we describe the ArtPop software package. We include links (indicated by the icon) to the Python code that created each example and figure. In Section 4, we provide a diverse set of scientific and pedagogical example applications. Finally, we conclude with a summary in Section 5.

2. SYNTHESIZING STELLAR SYSTEMS

There are three main components that are necessary for generating artificial images of fully-populated stellar systems. The first is modeling stellar populations and the associated stellar fluxes. To synthesize complex star formation histories, multiple single-burst populations may be combined. The second is spatial information, including the distance to the system and image coordinates for all its stars. Finally, image processing tools are required to inject the stellar fluxes into an image, convolve with a point-spread function (PSF), and add noise according to a set of instrumental and observational parameters. In this section, we describe how we have implemented each of these components.

2.1. Stellar Populations

The basic building block of stellar population synthesis is the simple stellar population (SSP), which consists of a population of stars born at the same time with a single metallicity and abundance pattern. To build SSPs, ArtPop starts from pre-calculated stellar isochrones and synthetic photometry. While the code can work with any set of models, here we use the Modules for Experiments in Stellar Astrophysics (MESA; Paxton et al. 2011, 2013, 2015) Isochrones and Stellar Tracks (MIST) project (Dotter 2016; Choi et al. 2016). In particular, we use synthetic photometry generated from the rotating models with $v/v_{\text{crit}} = 0.4$ from MIST version 1.2.

Given a set of stellar isochrones, we populate an SSP by sampling stellar masses from the initial mass function, $\Phi(M_i)$, using inverse transform sampling. The form of $\Phi(M_i)$ is an optional parameter in ArtPop. Unless noted otherwise in this work, we assume the initial mass function from Kroupa (2001), with a minimum mass of $M_{\text{min}} = 0.1 \, M_\odot$. The maximum mass of sampled stars is set by the stellar isochrone at a given age and metallicity. The total mass of the SSP, including both stars and stellar remnants, is given by

$$M_{\text{tot}}(t) = \int_{M_{\text{min}}}^{M(t)} M_{\text{act}}(M_i) \Phi(M_i) \, dM_i + M_{\text{rem}},$$

where $t$ is the stellar population age, $M_{\text{act}}(M_i)$ is the actual mass of a star that had an initial mass $M_i$, and $M_{\text{rem}}$ is the mass contained in stellar remnants including white dwarfs, neutron stars, and black holes. For the normalization of the initial mass function, we use a maximum mass of $M_{\text{max}} = 120 \, M_\odot$.

We assign masses to stellar remnants using the prescription of Renzini & Ciotti (1993). For stars with initial masses $M_i < 8.5 \, M_\odot$, the remnants are assumed to be white dwarfs of mass $M_{\text{rem}} = 0.077 \, M_i + 0.48$. Stars with $8.5 \, M_\odot \leq M_i < 40 \, M_\odot$ leave behind 1.4 $M_\odot$ neutron stars. Finally, the most massive stars with initial
masses $M_i \geq 40 M_\odot$ produce black holes of mass $0.5 M_i$. These initial-final mass relations are of course rough approximations, but they are standard in the field of stellar population synthesis (e.g., Bruzual & Charlot 2003; Maraston 1998; Conroy et al. 2009), making it possible to interpret and compare ArtPop models in this context.

In practice, ArtPop only samples “live” stars, which have masses that are included in the isochrone. To account for stellar remnants in the total mass that is reported by the ArtPop model, we apply a correction factor given by the ratio of the mass—as defined in Equation (1)—with and without remnants. This factor generally ranges from total masses (live stars and remnants) that are $\sim$5%-50% larger than the sum of the sampled stellar masses, depending on the age and metallicity of the SSP. Of course, older SSPs, in which stars have more time to evolve, have more mass locked up in stellar remnants than their younger counterparts.

### 2.2. Spatial Information

In addition to the stellar population synthesis described in the previous section, we must specify the spatial properties of the stars to completely describe the system. In particular, we need to know the stellar density distribution in physical units, the system distance to convert relative positions to angular units and stellar luminosities to brightnesses, and the pixel scale of the mock imaging system to convert the stellar positions to image coordinates. ArtPop can inject stars into images using arbitrary user-provided image coordinates, but we provide functions for sampling random positions from common spatial distributions.

Currently, ArtPop has sampling functions for a uniform surface brightness distribution, the Plummer distribution (Plummer 1911), which provides a good description of the distribution of mass in globular clusters, and the more general S´ersic distribution (S´ersic 1968), which spans the range of observed concentrations in stellar density distributions. An important note about sampling stellar positions is that the maximum radius out to which stars are sampled should be at least several times the scale length of the spatial distribution, even if such stars will fall outside of the image. Otherwise, too many stars will be sampled near the center of the galaxy, leading to a surface brightness distribution that is inconsistent with the input parameters.

### 2.3. Artificial Images

Given a simulated stellar population, including the positions and fluxes of every star, ArtPop generates an artificial image by injecting the individual stellar fluxes into an image array and convolving with a point-spread function (PSF). The current implementation of ArtPop assumes each star falls within the center of a pixel. For most purposes, this simplification will have a negligible effect, provided the PSF is well-sampled (e.g., Olsen et al. 2003); to the extent that spatial sampling matters for a given application, its impact will be most acute in rare stellar phases for which the density is such that we expect few stars per pixel. We plan to include optional subpixel star injection in a future version of ArtPop.

There are several options for adding noise to an ArtPop simulation, spanning the noiseless “ideal” case to fully artificial images with read noise and Poisson noise from the sky and artificial sources. In the latter case, the user must provide the necessary instrumental and observational parameters, including the aperture diameter, detector and photometric filter properties, exposure time, and sky surface brightness.

To convert stellar magnitudes in bandpass $x$ to photon counts, we use the analytic expression

$$C_x = A \cdot \varepsilon_x \cdot \frac{t_{\text{exp}} \Delta \lambda_x}{h} \frac{1}{\lambda_{\text{eff}, x}} 10^{-0.4 (m_x + 48.6)},$$

where $A$ is the effective collecting area of the telescope, $\varepsilon_x$ is the efficiency in band $x$ (set to unity by default), $t_{\text{exp}}$ is the exposure time, $h$ is the Planck constant, $\Delta \lambda_x$ and $\lambda_{\text{eff}, x}$ are the width and effective wavelength of bandpass $x$, and $m_x$ is the stellar AB magnitude (Oke & Gunn 1983) in bandpass $x$.

After convolution with the PSF, Poisson noise is generated from the combined counts of the source and sky, and the read noise is assumed to be Gaussian. If the galaxy is injected into a real image—which will already have detector and sky noise—Poisson noise is optionally generated from only the source counts before converting into the image flux units, provided the necessary parameters for Equation (2) are given as input.

### 3. THE SOFTWARE PACKAGE

In this section, we give an overview of the ArtPop software package and provide coding examples to demonstrate the code implementation and basic usage. The code is written in the Python programming language and is entirely open source. We refer the reader to the project website\textsuperscript{2} for a detailed description of the code, installation instructions, and a growing list of tutorials. We are actively developing ArtPop in a public GitHub repository\textsuperscript{3} and welcome bug reports, feature requests, and code contributions from the community. Through-

\textsuperscript{2} https://artpop.readthedocs.io
\textsuperscript{3} https://github.com/ArtificialStellarPopulations/ArtPop
Figure 1. A schematic overview of the procedure we follow to generate artificial images of fully-populated stellar systems. In (a), we sample stellar masses from an initial mass function (weights indicated by the color bar) and interpolate the associated stellar fluxes from an isochrone of the indicated age and metallicity. In (b), we assign spatial positions to each star using random draws from a two-dimensional Plummer distribution. In (c), the stars are injected into an image array, which is then convolved with the point-spread function (PSF). Finally, noise from the sources, sky, and detector are added to the image. In (d), we show a grI-composite image of an ArtPop simulation of a metal poor globular cluster at 5 kpc. In this example, the mock observations used the filter set, PSFs, and pixel scale of HST ACS/WFC. 

out the paper, we provide links (as icons) to the Python code that created each figure and example.

3.1. Code Structure

An important feature of the code design is that it is highly modular and extensible. This makes it possible for each of ArtPop’s functionalities to be used independently, together, or in combination with independently-generated input data (e.g., stellar positions and/or fluxes). ArtPop is divided into three primary modules—one for each of the components necessary for generating artificial images of fully-populated stellar systems, as described in Section 2. In particular, the core of the code is composed of the artpop.stars, artpop.space, and artpop.image modules, which are used for building stellar populations, sampling spatial distributions, and generating artificial images, respectively.

In Figure 1, we show a schematic overview of the procedure we follow to generate an artificial HST-like image of a globular cluster. For each step, we indicate the ArtPop module in which the corresponding code is implemented. In (a), we use the artpop.stars module to sample stellar masses from a user-specified initial mass function (indicated by the color bar) and interpolate the associated fluxes from a stellar isochrone. In (b), each star is assigned a spatial position in image coordinates using the artpop.space module. The stellar fluxes and positions are stored in an artpop.Source object, which in (c), is “observed” using the artpop.image module. Finally, the ArtPop image simulation is shown as a grI-composite image in (d).

3.2. Coding Example: Stellar Populations

A useful example of ArtPop’s modularity is using the artpop.stars module to generate simple and composite stellar populations. This capability is independent from making images and may be used to calculate integrated population parameters such as total magnitude, color, and the surviving stellar mass. Given a user-specified initial mass function and isochrone model, ArtPop can calculate such parameters either using numerical integration or by sampling a finite number of stars. The latter method introduces stochasticity at low stellar mass, which may be desired in certain applications.

Calculations involving stellar isochrones are carried out in the flexible Isochrone class. Three input arguments are required to initialize an Isochrone object: SSP initial stellar masses (mini), the actual stellar mass
after accounting for mass loss (mact), and a table of the associated stellar magnitudes (mags). Importantly, the code implementation is independent from how these parameters were generated, provided they are given in the correct format. Assuming these arguments have been defined, the code may be implemented as follows [4]:

```python
from artpop.stars import Isochrone

iso = Isochrone(mini, mact, mags)
```

The iso object has methods for performing real-time calculations of integrated SSP parameters. For example, if the magnitude table contains LSST magnitudes, the IMF-weighted $g - i$ color and surviving stellar mass (assuming a Salpeter initial mass function) of the SSP may be calculated using

```python
g_i = iso.ssp_color(
    blue="LSST_g", # blue filter
    red="LSST_i", # red filter
    imf="salpeter" # initial mass function
)
m_survive = iso.ssp_surviving_mass("salpeter")
```

For convenience, we have implemented a helper MIST-specific class, MISTIsochrone, which inherits all the methods from Isochrone. The user provides the desired SSP age, metallicity, and photometric system, and MISTIsochrone loads the required input parameters using the MIST synthetic photometry grids[4], interpolating over metallicity if necessary.

In Figure 2, we show the results from ArtPop calculations of the time evolution of SSPs mass-to-light ratios (top left) and integrated $V - I$ color (right). The top row shows the $V$- and $I$-band mass-to-light ratios (left) and integrated $V - I$ color (right). The bottom row shows the $V$- and $I$-band absolute SBF magnitudes (left) and $V - T$ SBF color (right). The SSPs were generated using the Kroupa (2001) initial mass function. Solar metallicity SSPs are shown as dashed lines, and metal poor SSPs with $[\text{Fe/H}] = -1.5$ are shown as solid lines[2].

```python
from artpop.stars import SSP

ssp = SSP(
    isochrone=iso, # Isochrone object
    num_stars=1e5, # number of stars
    imf="salpeter", # initial mass function
)
```

Similar to the MISTIsochrone class, there is a MISTSSP helper class, which loads a MIST isochrone for a given set of SSP parameters. The above ssp object has various methods for calculating integrated properties. For example, the $i$-band magnitude and $g - i$ color are calculated using the total_mag and integrated_color methods:

```python
i = ssp.total_mag("LSST_i")
g_i = ssp.integrated_color("LSST_g", "LSST_i")
```

The distance of the population is set to 10 pc by default, so the above magnitude is in absolute units. To build composite stellar populations (CSPs) in ArtPop, ssp objects may be intuitively added together using the + operator. For example, suppose we have created two SSPs, one old (ssp_old) and the other young (ssp_young). Then, we may combine them into a single composite population as follows:

```python
csp = ssp_old + ssp_young
```

The new csp object is a composite of the old and young SSPs, inheriting all the same methods for calculating integrated properties.

[4] http://waps.cfa.harvard.edu/MIST/model_grids.html
We emphasize that, since SSP objects contain a finite number of stars, the integrated properties will be stochastic due to incomplete sampling of the mass function (e.g., Santos & Frogel 1997; Greco et al. 2021). The number of stars required for the calculations to converge is a function of stellar population parameters (e.g., due to the frequency of rare, luminous stars) and photometric bandpass, but in general it takes a total stellar mass of $>10^6 \, M_\odot$ to approach a fully sampled mass function. However, if the goal is to fully sample the IMF, the Isochrone class should be used.

### 3.3. Coding Example: Image Simulations

To create artificial images, ArtPop uses Imager objects, which are implemented in the artpop.image module. There are two types of Imager objects: IdealImager and ArtImager. The former generates noiseless images, and the latter generates fully artificial images that include simulated noise from the sky, source, and detector. Initializing an ArtImager object therefore requires both instrumental (e.g., mirror diameter and read noise) and observational (e.g., exposure time and the sky surface brightness) parameters as input.

Similar to real observatories, a single Imager object is designed to “observe” any number of sources. This is convenient when the goal is to mock observe many sources using the same imaging setup. In ArtPop, sources are stored in Source objects, which are containers that hold the positions and magnitudes of the stars. A primary purpose of ArtPop is to generate these positions and magnitudes, though this is not necessary to create images using Imager and Source objects. As a simple example, let us create a mock observation using the SSP created in Section 3.2.

First, we create a Source object using the magnitudes from the previously created SSP. For this example, we use the artpop.plummer.xy function to sample positions from a Plummer distribution:

```python
from astropy import units as u
from artpop import Source
from artpop.space import plummer_xy

# image coordinates
xy = plummer_xy(
    num_stars=ssp.num_stars,
    scale_radius=500*u.pc,
    distance=8*u.Mpc,
    xy_dim=xy_dim,
    pixel_scale=pixel_scale
)
```

The above src object holds the stellar positions and magnitudes for a system of $10^6$ stars (the number of stars in ssp) at a distance of 8 Mpc, with a spatial distribution that follows a Plummer profile of scale radius 500 pc. We also specify the image dimensions and pixel scale in order to convert the positions to image coordinates and flag stars that fall within the image. Stars that fall outside the image contribute to the total mass but are masked within the array of stellar positions, since they will not be injected into the image.

For simplicity, we will observe the source using an IdealImager, which may be initialized without any input parameters:

```python
from artpop.image import IdealImager

imager = IdealImager()
```

Mock observations are carried out using the observe method. Here, we will observe the artificial source in the $i$ band, assuming 0".6 seeing, which we will model as a Moffat profile using the moffat_psf function:

```python
from artpop.image import moffat_psf

psf = moffat_psf(
    fwhm=0.6*u.arcsec,
    pixel_scale=pixel_scale
)
```

The returned object, obs, is an IdealObservation object, which is a container that holds the PSF-convolved image, as well as metadata such as the zero point and observation bandpass.

The above examples show that multiple steps are required to create stellar positions and magnitudes, which are required to create a Source object. For convenience, ArtPop provides helper classes for creating complete Source objects in a single step. For example, a Source object composed of an SSP with a Sérsic spatial distribution and synthetic photometry generated using the MIST isochrones can be created using the MISTSersicSSP class.

### 4. EXAMPLE APPLICATIONS

There is a wide range of use cases for ArtPop. From visualizing the age-metallicity degeneracy to measuring survey detection efficiencies to generating synthetic data for machine learning algorithms, its potential applications span both scientific and pedagogical projects. In
ArtPop simulations of an old dwarf spheroidal of stellar mass $3 \times 10^6 \, M_\odot$, placed at a distance of 5 Mpc. The galaxy is composed of an SSP of age 12.6 Gyr and metallicity $[\text{Fe}/\text{H}] = -1.8$. We show $gri$-composite images of mock observations of varying image resolution (FWHM values indicated at the top of each column) and exposure times (indicated on the left of each row). The image resolutions were chosen to be similar to HST/ACS (left column), HSC (middle column), and SDSS (right column). All images assume a mirror diameter of 8 m and sky brightnesses of 22, 21, and 20 mag arcsec$^{-2}$ in $g$, $r$, and $i$, respectively. In each panel, the green line indicates the scale of 20″.

4.1. Image resolution and exposure time

The ArtImager class generates fully artificial images, adding noise from the detector, sky, and artificial source according to the user-provided instrumental and observational parameters. For a fixed artificial source, this is particularly useful for visually and quantitatively exploring the interplay between observational parameters such as exposure time and image resolution.

In Figure 3, we show $gri$-composite images of the same dwarf galaxy placed at 5 Mpc, with the pixel scales and resolutions similar to HST/ACS (left column), HSC (middle column), and SDSS (right column). The dwarf galaxy has a Plummer mass distribution with a scale radius of 400 pc and a total stellar mass of $3 \times 10^6 \, M_\odot$. The bottom row shows mock observations with an exposure time of 3 min, and the top row shows mock observations with a factor of 10 longer exposure time. Other than the resolutions and pixel scales, the mock observation setups are identical.

Comparing the top and bottom rows, the increase in exposure time leads to the expected increase in the signal-to-noise ratio. Stars in the outskirts of the galaxy disappear below the noise level in the short exposure panels. The comparisons become more interesting when we also vary the image resolution (both seeing and pixel scale). At the highest resolution, the RGB is resolved in the longer 30 min exposure, but the small pixels (0.05 arcsec pixel$^{-1}$) make the source appear as a diffuse object. As the resolution is decreased, stars blend into brighter point sources and the full galaxy becomes more easily detected as a single coherent object.
$\mu_I = 26$ mag arcsec$^{-2}$  

8.0 Mpc

$\mu_I = 23$ mag arcsec$^{-2}$

2.0 Mpc

$\mu_I = 20$ mag arcsec$^{-2}$

D = 0.5 Mpc

4.2. Surface brightness and distance

In the nearby universe, surface brightness is independent of distance. This important result can be visualized using ArtPop’s MISTUniformSSP class, which generates a uniformly distributed SSP using the MIST isochrone models and a user-specified average surface brightness and distance. Fixing the SSP parameters and field of view, the number of stars in an image increases as a function of distance, exactly compensating for dimming due to the inverse square law to ensure the surface brightness remains constant.

In Figure 4, we show image simulations of uniformly distributed SSPs with mean I-band surface brightnesses of 26 (left column), 23 (middle column), and 20 mag arcsec$^{-2}$ (right column)—spanning the stellar density range from low-mass dwarf galaxies to globular clusters (e.g., Muñoz et al. 2015). We place the SSPs at distances of 0.5 (bottom row), 2 (middle row), and 8 Mpc (top row). In all panels, the stellar population age and metallicity are 10 Gyr and [Fe/H] = −1.6, respectively. Each image is $35'' \times 25''$, with a pixel scale of 0.05 arcsec pixel$^{-1}$. For the mock observations, we used the ArtImager with 90 minute exposures in each of the HST ACS/WFC $I_{814}$, $r_{606}$, and $g_{475}$ filters. The PSFs were generated using Tiny Tim (Krist 1995).

As expected, we see that fainter stars become increasingly resolved at fainter surface brightnesses (lower stellar density) and closer distances. At the nearest distance in the 23 mag arcsec$^{-2}$ column (middle panel of the bottom row), there are $\sim 3 \times 10^5$ stars, with the blue main sequence resolved into individual stars and only a handful of luminous giants in the frame. As the stars are placed to larger distances, their numbers increase as the square of the distance—there are $\sim 5 \times 10^6$ stars in the 2 Mpc
Figure 5. Simulated gri-composite images of dwarf galaxies of stellar mass $10^6$ M$_\odot$ at a distance of 5 Mpc. The galaxies are SSPs with $[\text{Fe/H}] = -1.5$, assuming the MIST model isochrones. Each row shows a single galaxy of fixed age, which is indicated on the left. For each galaxy, the leftmost panel shows the full SSP, and the remaining five panels show stars that are on the main sequence (MS), red giant branch (RGB), core-helium burning (CHeB) stars, and the early and thermally pulsating asymptotic branches (E-AGB and TP-AGB, respectively). The simulations were tuned to resemble an LSST-like observatory and observing conditions with exposure times of 90 minutes in $i$ and 45 minutes in $g$ and $r$. Each panel is 2 kpc on a side. As noted in the main text, the phases are defined according to the MIST primary equivalent evolutionary phases.

More than 200 million stars are required to generate the high surface brightness 20 mag arcsec$^{-2}$ population at 8 Mpc, which is memory intensive. For such situations, ArtPop provides an option so set a magnitude limit, fainter than which individual stars will not be sampled. Instead, the flux from the stars is combined into a smooth model, which is added to the image along with the brighter individual stars.

4.3. Dwarf galaxies and stellar populations

ArtPop makes it easy to visualize stellar systems as a function of astrophysical (e.g., distance, stellar mass, and SSP age) and observational (e.g., exposure time, bandpass, aperture diameter, and sky surface brightness) parameters. Moreover, since stars are injected individually, it is possible to visually compare how...
Figure 6. DECam $g - i$ color distribution of 1000 realizations of an ArtPop dwarf galaxy with fixed model parameters. The galaxy is composed of an ancient, metal poor SSP with an age of 12.6 Gyr and metallicity of $[\text{Fe}/\text{H}] = -2$. Given its low stellar mass of $10^5 \, M_\odot$, stochasticity in the numbers of luminous evolved stars leads to a wide range of integrated and visual properties. The $gri$-composite images on the top show the bluest (left), median (middle), and reddest (right) dwarf galaxy in the sample. A DECam-like observatory and observing conditions were assumed, with $\sim 1''$ seeing and 2 hr (1 hr) exposures in $i$ ($g$ and $r$). The galaxy is placed at a distance of 2.5 Mpc, and the images are 800 pc on a side.

Different phases of stellar evolution contribute to the (semi)resolved appearance and integrated properties of the system, which helps build intuition for interpreting images of similar systems in real data.

To demonstrate the pedagogical utility of isolating stellar phases in artificial images, Figure 5 shows $gri$-composite images of simulated dwarf galaxies at a distance of 5 Mpc. The distribution of stars in each galaxy follows a Sérsic distribution, with a total stellar mass of $10^6 \, M_\odot$. Each galaxy is composed of a metal-poor SSP with $[\text{Fe}/\text{H}] = -1.5$ and an age ranging from 10 Gyr (top row) to 100 Myr (bottom row), assuming the MIST model isochrones. Each row shows the same galaxy realization, with the full SSP shown in leftmost panel. The remaining five panels show stars that are in the evolutionary phase indicated at the top of each column. From left to right, the phases are the main sequence (MS), red giant branch (RGB), core-helium burning (CHeB) stars, and the early and thermally pulsating asymptotic branches (E-AGB and TP-AGB, respectively). Similar to Greco et al. (2021), we label phases of stellar evolution according to the MIST primary equivalent evolutionary phases (EEP; Choi et al. 2016; Dotter 2016), which are useful computationally but in some cases lead to terminology that differs from standard nomenclature (e.g., an RGB phase associated with high-mass stars).

When the mass of a stellar system is $\lesssim 10^5 \, M_\odot$ (Greco et al. 2021), the IMF becomes significantly undersampled. At such low masses, the numbers of the most luminous stars range from zero to a dozen or so, leading to
ArtPop

Figure 7. Visual comparison of artificial dwarf galaxies generated using stellar fluxes from the MIST (left panel) and PARSEC (right panel) isochrones. The isochrones were generated consistently using the Flexible Stellar Population Synthesis software package. Other than the choice of isochrone, the model parameters are identical, including the individual stellar positions and masses. The images are HST ACS/WFC $g-r-I$-composites, with the integrated $g_{475} - I_{814}$ galaxy color indicated below each image. The middle panel shows the color-magnitude diagram of the constituent MIST (red) and PARSEC (blue) stars. The galaxies are SSPs composed of $5 \times 10^6$ stars with the population parameters indicated in the middle panel. With new imaging surveys like LSST and ultimately the Roman Space Telescope, there will soon be a vast increase in the number of low-mass galaxies with high-quality imaging and (semi-)resolved stellar populations. These systems will span a large range of stellar population parameters, potentially providing powerful benchmarks for stellar population synthesis models.

In Figure 6, we show the DECam $g - i$ color distribution of 1000 realizations of an ArtPop dwarf galaxy with fixed model parameters. The galaxy has an age of 12.6 Gyr, metallicity of $[Fe/H] = -2$, and low stellar mass of $10^5 M_{\odot}$. Stochasticity in the numbers of evolved stars—particularly AGB stars—results in a standard deviation of $\sim 0.05$ mag and range of $\sim 0.25$ mag in $g - i$ color. In the top three panels, we show $gri$-composite images of the bluest (left), median color (middle), and reddest (right) dwarf galaxy realization in the sample.

4.4. Comparing isochrone models

With new imaging surveys like LSST and ultimately using the Roman Space Telescope, there will soon be a vast increase in the number of low-mass galaxies with high-quality imaging and (semi-)resolved stellar populations. These systems will span a large range of stellar population parameters, potentially providing powerful benchmarks for stellar population synthesis models. While model comparisons using color-magnitude diagrams are common, ArtPop’s modular design makes it straightforward to additionally generate artificial galaxy images based on the different models.

In Figure 7, we show a visual comparison of dwarf galaxies generated using stellar fluxes from the MIST (left panel) and PARSEC (right panel; Bressan et al. 2012) isochrone models. The shown HST ACS/WFC $g-r-I$-composite images are based on mock observations with an HST-like observatory using 180 min exposures. To ensure the stellar fluxes were calculated consistently, we used the Flexible Stellar Population Synthesis software package (Conroy et al. 2009; Conroy & Gunn 2010). The center panel shows the MIST (blue points) and PARSEC (red points) color-magnitude diagrams associated with the stars in the mock images.

Other than the isochrone model, all model parameters are identical—including the stellar positions and masses. The galaxies are composed of SSPs with $5 \times 10^6$ stars of age $\sim 355$ Myr and metallicity $[Fe/H] = -1.5$. To exactly match the stellar masses, we restricted the mass range from $0.1 M_{\odot}$ to $2.8305 M_{\odot}$. The minimum stellar mass is set by the MIST isochrones, whereas the maximum stellar mass is set by the PARSEC isochrones.

4.5. Injecting into real images

A complementary approach to using the ArtImager class for generating fully artificial images (as shown in Section 4.1-4.4) is to inject artificial sources into real astronomical images. Using this approach, ArtPop has been proven an effective tool in estimating imaging survey depth and detection completeness (Greene et al. 2021). Particularly, the dual-mode functionality of ArtPop, namely generating stellar population models and artificial images, can be used simultaneously in the following way.

Using the MISTSersicSSP class, we generate an artificial dwarf galaxy source, placed at 1 Mpc, assuming an old SSP with $M_* = 5 \cdot 10^6 M_{\odot}$, $log(Age) = 10.1$ Gyr, $[Fe/H] = -2$, and a Sérsic surface brightness distribution. We then use the IdealImager to generate a noise-
model, $M_\star = 5 \cdot 10^5 M_\odot$, 1 Mpc

model injected to DES data

Figure 8. ArtPop model of an ultra-faint dwarf galaxy with a stellar mass of $M_\star = 5 \cdot 10^5 M_\odot$, placed at a distance of 1 Mpc. The left panel shows the gri-composite color image and the middle panel shows the same model injected into a Dark Energy Survey image. In the right panel we show the $gr$ color-magnitude diagram, where stars are color-coded according to their stellar phase. The dashed horizontal line marks the $g$-band limiting magnitude reported by DES DR1 of $m_{g,\text{lim}} = 24.33$ mag. Ignoring star-galaxy separation issues, stars that are brighter than this limit should, in theory, be detected in the image.

For understanding the detection of low stellar density systems in imaging surveys, models of dwarf galaxies, spanning a range of stellar masses, chemical compositions, ages, and morphologies can be generated. As demonstrated, ArtPop provides both catalogs of stars that can be injected into existing survey catalogs, accounting for noise and detection limits, and realistic images with photometry in the appropriate photometric system that can be then injected into survey images.

5. SUMMARY

In this paper we have presented ArtPop, a public software package for synthesizing of stellar populations and simulating realistic images of stellar systems. The code is modular and designed to allow maximal user flexibility. ArtPop is under active development and currently provides the following capabilities:

1. Stellar population synthesis: The artpop.stars module builds simple and composite stellar populations by sampling a user-specified initial mass function. Stellar fluxes are calculated by interpolating pre-calculated magnitude grids from a stellar isochrone model, which the user is free to choose.

2. Sampling spatial distributions: The artpop.space module samples two-dimensional positions in image coordinates. Currently, we have implemented samplers for uniform, Plummer, and Sersic distributions. Grid sampling for arbitrary two-dimensional functions is also possible using Astropy model objects.

3. Image simulations: The artpop.image module generates artificial images of Source objects. The simulations can be fully artificial images with realistic noise or ideal noiseless images, which may be injected into real imaging data.

These three functionalities can be used independent or collectively to generate catalogs and images of different stellar systems such as galaxies, globular clusters, and stellar streams.

We encourage the reader to install ArtPop, go through the tutorials on the website, and run some of the ex-
artpopenotes

ArtPop

to ArtPop. More information about contributing to ArtPop can be found at https://artpop.readthedocs.io/en/latest/getting_started/contribute.html.



ACKNOWLEDGMENTS

We are deeply indebted to the open-source astronomy and scientific Python communities. Indeed, we learned almost everything we know about developing open-source software from poking around public GitHub repositories. We thank Adrian Price-Whelan and Pieter van Dokkum for providing useful comments, and we thank Ava Polzin for her help generating HST PSFs.

J.P.G. was supported by an NSF Astronomy and Astrophysics Postdoctoral Fellowship under award AST-1801921. S.D. is supported by NASA through Hubble Fellowship grant HST-HF2-51454.001-A awarded by the Space Telescope Science Institute, which is operated by the Association of Universities for Research in Astronomy, Incorporated, under NASA contract NAS5-26555.

REFERENCES

Abbott, T. M. C., Abdalla, F. B., Allam, S., et al. 2018, ApJS, 239, 18, doi: 10.3847/1538-4365/aae90

Aihara, H., Arimoto, N., Armstrong, R., et al. 2018, PASJ, 70, S4, doi: 10.1093/pasj/psx066

Ajhar, E. A., & Tonry, J. L. 1994, ApJ, 429, 557, doi: 10.1086/174344

Astropy Collaboration, Robitaille, T. P., Tollerud, E. J., et al. 2013, A&A, 558, A33, doi: 10.1051/0004-6361/201322068

Bahcall, J. N., & Schneider, D. P. 1988, in The Harlow-Shapley Symposium on Globular Cluster Systems in Galaxies, ed. J. E. Grindlay & A. G. D. Philip, Vol. 126, 455–463

Bressan, A., Marigo, P., Girardi, L., et al. 2012, MNRAS, 427, 127, doi: 10.1111/j.1365-2966.2012.21948.x

Bruzual, G., & Charlot, S. 2003, MNRAS, 344, 1000, doi: 10.1046/j.1365-8711.2003.06897.x

Buzzoni, A. 1993, A&A, 275, 433

Cantiello, M., Raimondo, G., Brocato, E., & Capaccioli, M. 2003, AJ, 125, 2783, doi: 10.1086/375322

Choi, J., Dotter, A., Conroy, C., et al. 2016, ApJ, 823, 102, doi: 10.3847/0004-637X/823/2/102

Conroy, C. 2013, ARA&A, 51, 393, doi: 10.1146/annurev-astro-082812-141017

Conroy, C., & Gunn, J. E. 2010, ApJ, 712, 833, doi: 10.1088/0004-637X/712/2/833

Conroy, C., Gunn, J. E., & White, M. 2009, ApJ, 699, 486, doi: 10.1088/0004-637X/699/1/486

Cook, B. A., Conroy, C., & van Dokkum, P. 2020, ApJ, 893, 160, doi: 10.3847/1538-4357/ab83ea

Cook, B. A., Conroy, C., van Dokkum, P., & Speagle, J. S. 2019, ApJ, 876, 78, doi: 10.3847/1538-4357/ab1e65

Dalcanton, J. J., Williams, B. F., Seth, A. C., et al. 2009, ApJS, 183, 67, doi: 10.1088/0067-0049/183/1/67

Danieli, S., van Dokkum, P., & Conroy, C. 2018, ApJ, 856, 69, doi: 10.3847/1538-4357/aaadfb

Dolphin, A. E. 2000, PASJ, 112, 1383, doi: 10.1086/316630

Dotter, A. 2016, ApJS, 222, 8, doi: 10.3847/0067-0049/222/1/8

Greco, J. P., van Dokkum, P., Danieli, S., Carlsten, S. G., & Conroy, C. 2021, ApJ, 908, 24, doi: 10.3847/1538-4357/abd030

Greco, J. P., Greene, J. E., Strauss, M. A., et al. 2018, ApJ, 857, 104, doi: 10.3847/1538-4357/aab842

Greene, J. E., Lancaster, L., Ting, Y.-S., et al. 2021, arXiv e-prints, arXiv:2105.04581.

Ivezić, Ž., Kahn, S. M., Tyson, J. A., et al. 2019, ApJ, 873, 111, doi: 10.3847/1538-4357/ab042c

Krist, J. 1995, in Astronomical Society of the Pacific Conference Series, Vol. 77, Astronomical Data Analysis Software and Systems IV, ed. R. A. Shaw, H. E. Payne, & J. J. E. Hayes, 349

Kroupa, P., 2001, MNRAS, 322, 231, doi: 10.1046/j.1365-8711.2000.04022.x

Maraston, C. 1998, MNRAS, 300, 872, doi: 10.1046/j.1365-8711.1998.01947.x

Muñoz, R. P., Eigenthaler, P., Puzia, T. H., et al. 2015, ApJL, 813, L15, doi: 10.1088/2041-8205/813/1/L15

Software: astropy (Astropy Collaboration et al. 2013), Flexible Stellar Population Synthesis (Conroy et al. 2009; Conroy & Gunn 2010), matplotlib (Hunter 2007), numpy (Van der Walt et al. 2011), scipy (https://www.scipy.org)
