UNIVERSITY OF CALIFORNIA
SANTA CRUZ

LARGE SCALE ANALYSIS OF ASTRONOMICAL DATA USING MACHINE LEARNING AND VISUALIZATION TECHNIQUES

A dissertation submitted in partial satisfaction of the
requirements for the degree of

DOCTOR OF PHILOSOPHY

in

COMPUTER SCIENCE

by

Ryan David Hausen

June 2022

The Dissertation of Ryan David Hausen
is approved:

Roberto Manduchi, Chair

Brant Robertson

J. Xavier Prochaska

Narges Norouzi

Peter Biehl
Vice Provost and Dean of Graduate Studies

Table of Contents

List of Figures **vi**

List of Tables **xxx**

Abstract **xxxviii**

Dedication **xxxix**

Acknowledgments **xl**

1 Introduction **1**

2 Morpheus: A Deep Learning Framework For Pixel-Level Analysis of Astronomical Image Data **4**

2.1 Introduction 4
2.2 *Morpheus* Deep Learning Framework 10
2.2.1 Input Data 11
2.2.2 Neural Network 12
2.2.3 Parallelization for Large Images 20
2.3 Model Training 21
2.3.1 Training Data 22
2.3.2 Data Augmentation 23
2.3.3 Loss Function 24
2.3.4 Optimization Method 29
2.3.5 Model Evaluation 30
2.4 Segmentation and Deblending 32
2.4.1 Segmentation 33
2.4.2 Deblending 34
2.5 Object-Level Classification 35
2.6 *Morpheus* Data Products 40
2.6.1 Morphological Images for GOODS South 41
2.7 *Morpheus* Performance 49

2.7.1 Object-Level Morphological Classifications 49
2.7.2 Simulated Detection Tests 60
2.7.3 Morphological Classification vs. Surface Brightness Profile 64
2.7.4 Source Detection and Completeness 65
2.7.5 Morphological Classification vs. Source Magnitude 71
2.7.6 False Positives in GOODS South 72
2.8 Value Added Catalog for 3D-HST Sources with *Morpheus* Morphologies 75
2.9 Discussion 78
2.9.1 Pixel-Level Morphology 78
2.9.2 Morphological Deblending 79
2.9.3 Classifications Beyond Morphology 82
2.9.4 Deep Learning and Astronomical Imagery 83
2.10 Summary and Conclusions 83

3 Partial-Attribution Instance Segmentation for Astronomical Source Detection and Deblending 87
3.1 Introduction 87
3.1.1 Related Work 88
3.2 Partial-Attribution Instance Segmentation 89
3.3 Our Approach 91
3.3.1 Dataset 91
3.3.2 Model 92
3.3.3 Training 93
3.4 Discussion and Future Work 94

4 FitsMap: A Simple, Lightweight Tool For Displaying Interactive Astronomical Image and Catalog Data 95
4.1 Introduction 95
4.2 Methods 98
4.2.1 *Map Generator* 99
4.2.2 *Map Viewer* 100
4.3 Performance 102
4.3.1 *Map Generator* Performance 102
4.3.2 *Map Viewer* Performance 103
4.4 Conclusion and Future Work 106

5 Revealing the Galaxy-Halo Connection Through Machine Learning 116
5.1 Introduction 116
5.2 Methods 119
5.2.1 Explainable Boosting Machines 120
5.2.2 Simulated Galaxy Catalog Training Set 121

5.2.3 Training Procedure 124
5.3 Results 127
5.3.1 Average Contribution 127
5.4 Composite EBMs for Restricted Parameter Sets 132
5.4.1 CEBM Feature and Interaction Functions 134
5.4.2 CEBM Average Contribution 134
5.4.3 Visualizing CEBM Feature and Interaction Functions . . . 135
5.5 Composite EBM Model for Star Formation Rate 136
5.6 Composite EBM Model for Stellar Mass 138
5.7 Discussion 142
5.8 Summary 144

6 Conclusion **165**

A Deep Learning & Morpheus Code/Data Products **167**
A.1 Deep Learning 167
A.1.1 Artificial Neuron 167
A.1.2 Neural Networks 169
A.1.3 Convolutional Neural Networks 170
A.1.4 Other Functions in Neural Networks 171
A.1.5 U-Net Architecture 174
A.2 code release 174
A.3 code tutorial 175
A.4 Data Release 175

List of Figures

2.1 Diagram of a single block in the *Morpheus* neural network architecture (Figure 2.2). Panel (c) shows a single block from the architecture, parameterized by the number P (black) of block operations and the number Q (purple) of convolutional artificial neurons (CANs; Section A.1.3) in all of the convolutional layers within the block. Panel (b) shows an example zoom-in where there are $P = 2$ groups of $Q = 4$ block operations. Panel (a) shows a zoom-in on a block operation, which consists of batch normalization, $Q = 4$ CANs, and a ReLU operation. In the notation of Equation 2.1, this block operation would be written as $\mathrm{OP}_4(\mathbf{X})$. 15

2.2 Neural network architecture of the *Morpheus* deep learning framework, following a U-Net [224] configuration. The input to the model *Morpheus* consists of astronomical FITS images in n_b bands (upper left). These images are processed through a series of computational blocks (sky blue rectangles), each of which applies P (black numbers) block operations consisting of a batch normalization and multiple convolutional layers producing Q (purple numbers) feature maps. The blocks are described in more detail in Figure 2.1. During the contraction phase of the model, max-pooling layers (salmon rectangles) are applied to the data to reduce the pixel size of the images by taking local maxima of 2×2 regions. The contraction phase is followed by an expansion phase where the output feature maps from each block are expanded by a 2×2 factor via bicubic interpolation (green rectangles) and concatenated with the output from the corresponding block in the contraction phase. The output from the last block is processed through a set of convolutional layers (light blue box with $Q = 5$) that result in a feature map for each classification in the model. These "classification images" are normalized to sum to unity pixel-by-pixel. In this chapter, the classification images are *spheroid*, *disk*, *irregular*, *point source/compact*, and *background*. 18

2.3 Data augmentation pipeline used during neural network training. Each training image is processed by the data augmentation pipeline before being presented to the neural network during training. The pipeline can be described in 7 stages (annotated '(a)-(g)' above). First,an image from the training set is selected (Panel a). A number of augmentation operations are then applied to the image. The image is rotated by a random angle $\theta \in [0, 2\pi]$ (Panel b), flipped horizontally with 50% probability (Panel c), and flipped vertically with a 50% probability (Panel d). The centermost 60×60 subset of the resulting image is cropped (Panel e), and then a random 40×40 subset is selected from the cropped image (Panel f). The output 40×40 rotated, flipped, and cropped image is then used for training. This procedure increases the available images for training by a factor of $\sim 574,400$. Using this process helps reduce overfitting, particularly in cases of datasets with limited training sample sizes. 25

2.4 Segmentation and deblending process used by *Morpheus*, illustrating Algorithms 1 and 2. The *background* image (Panel a) output from the *Morpheus* neural network is used as input to a Sobel-filtered image (Panel b) and a discretized map marking regions of high and low *background* (Panel c). These two images are input to a watershed algorithm to identify and label distinct, connected regions of low *background* that serve as the highest-level *Morpheus* segmentation map (Panel e) This segmentation map represents the output of Algorithm 1. A flux image and a list of object locations (Panel d) are combined with the high-level segmentation map to deblend multicomponent objects using an additional watershed algorithm by using the source locations in the flux image as generating points. The end result is a deblended segmentation map (Panel f), corresponding to the output of Algorithm 2. 37

2.5 *Morpheus* morphological classification results for a region of the GOODS South field. The far left panel shows a three-color composite VzH image. The scale bar indicates 1.5". The V, z, J, and H FITS images are supplied to the *Morpheus* framework, which then returns images for the *spheroid* (red-black panel), *disk* (blue-black panel), *irregular* (green-black panel), *point source/compact* (yellow-black panel), and *background* (white-black panel) classifications. The pixel values of these images indicate the local dominant *Morpheus* classification, normalized to sum to one across all five classifications. The panel labeled "Segmentation Map" is also generated by *Morpheus*, using the 3D-HST survey sources as generating locations for the segmentation Algorithm 1. The regions in the segmentation map are color-coded by their flux-weighted dominant class computed from the *Morpheus* classification values. The far right panel shows the *Morpheus* "classification color" image, where the pixel hues indicate the dominant morphological classification, and the intensity indicates $1-$*background*. The saturation of the *Morpheus* color image indicates the difference between the dominant classification value and the second most dominant classification, such that white regions indicate pixels where *Morpheus* returns a comparable result for multiple classes. See Section 2.6.1 for more details. 38

2.6 *Morpheus background* classification image for the Hubble Legacy Fields [118] reduction of the CANDELS survey data [95, 144] in GOODS South. Shown are the normalized model estimates that each of the $\sim 10^8$ pixels belongs to the *background* class. The scale bar indicates 1.5 arcmin. The color bar indicates the *background*$\in[0, 1]$, increasing from white to black. Correspondingly, the bright areas indicate regions of low background where sources were detected by *Morpheus*. 43

2.7 *Morpheus spheroid* classification image for the Hubble Legacy Fields [118] reduction of the CANDELS survey data [95, 144] in GOODS South. Shown are the normalized model estimates that each of the $\sim 10^8$ pixels belongs to the *spheroid* class. The scale bar indicates 1.5 arcmin. The color bar indicates the *spheroid*$\in[0,1]$, increasing from black to red. Correspondingly, the bright red areas indicate pixels where *Morpheus* identified *spheroid* objects. 44

2.8 *Morpheus disk* classification image for the Hubble Legacy Fields [118] reduction of the CANDELS survey data [95, 144] in GOODS South. Shown are the normalized model estimates that each of the $\sim 10^8$ pixels belongs to the *disk* class. The scale bar indicates 1.5 arcmin. The color bar indicates the *disk*$\in[0,1]$, increasing from black to blue. Correspondingly, the bright blue areas indicate pixels where *Morpheus* identified *disk* objects. 45

2.9 *Morpheus irregular* classification image for the Hubble Legacy Fields [118] reduction of the CANDELS survey data [95, 144] in GOODS South. Shown are the normalized model estimates that each of the $\sim 10^8$ pixels belongs to the *irregular* class. The scale bar indicates 1.5 arcmin. The color bar indicates the *irregular*$\in[0,1]$, increasing from black to green. Correspondingly, the bright green areas indicate pixels where *Morpheus* identified *irregular* objects. 47

2.10 *Morpheus point source/compact* classification image for the Hubble Legacy Fields [118] reduction of the CANDELS survey data [95, 144] in GOODS South. Shown are the normalized model estimates that each of the $\sim 10^8$ pixels belongs to the *point source/-compact* class. The scale bar indicates 1.5 arcmin. The color bar indicates the *point source/compact*$\in[0,1]$, increasing from black to yellow. Correspondingly, the bright yellow areas indicate pixels where *Morpheus* identified *point source/compact* objects. 48

2.11 *Morpheus* morphological color image for the Hubble Legacy Fields [118] reduction of the CANDELS survey data [95, 144] in GOODS South. The image intensity is set proportional to $1-$*background* for each pixel, such that regions of high background are black and regions with low background containing source pixels identified by *Morpheus* appear bright. The hue of each source pixel indicates its dominant classification, with *spheroid* shown as red, *disk* as blue, *irregular* as green, and *point source/compact* as yellow. The color saturation of each pixel is set to the difference between the first and second most dominant class values, such that regions with indeterminate morphologies as determined as *Morpheus* appear as white and regions with strongly determined classifications appear as deep colors. 50

2.12 Distribution of morphological classifications in the Kartaltepe et al. [132] sample, which serve as a training sample for *Morpheus*. Shown are histograms of the fraction of sources with a non-zero probability of belonging to the *spheroid* (upper left), *disk* (upper right), *irregular* (lower left), or *point source/compact* classes, as determined visual classification by expert astronomers. The histograms have been normalized to show the distribution of classification probabilities for each class, and consist of $\approx 7,600$ sources. 52

2.13 Histograms (purple) and cumulative distribution (blue lines) of agreement $a(\boldsymbol{p})$ for the Kartaltepe et al. [132, K15] visual morphological classifications, for objects with *spheroid* (upper left), *disk* (upper right), *irregular* (lower left), and *point source/compact* (lower right) as their dominant classification. Agreement $a(\boldsymbol{p})$ (see Equation 2.17 for a definition) characterizes the breadth of the distribution of morphological classes assigned by the K15 classifiers for each object, with $a(\boldsymbol{p})$=1 indicating perfect agreement of a single class and $a(\boldsymbol{p})$=0 corresponding to perfect disagreement with equal probability among classes. The distribution of agreement in the K15 training classifications is roughly bimodal, with a strong peak near-perfect agreement and a broader peak near $a(\boldsymbol{p})\approx0.5$, close to the agreement value for an even split between two classes. 54

2.14 Confusion matrix for the distribution of K15 morphological classifications. Shown is the distribution of morphologies assigned by K15 visual classifiers for objects of a given dominant classification. Objects with a dominant *spheroid* class show the most variation, with frequent additional *disk* and *point source/compact* morphologies assigned. The most distinctive dominant class is *point source/-compact*, which also receives a *spheroid* classification in 14% of objects. The off-diagonal components of the confusion matrix indicate imperfect agreement among the K15 classifiers, consistent with the distributions of the agreement statistic shown in Figure 2.13. . . . 55

2.15 Confusion matrix showing the spread in *Morpheus* dominant classifications for objects with a given K15 dominant classifications. The *Morpheus* framework is trained to reproduce the input K15 distributions, and this confusion matrix should, therefore, largely match Figure 2.14. The relative agreement between the two confusion matrices demonstrates that the *Morpheus* output can approximate the input K15 classification distributions. 56

2.16 Confusion matrix quantifying the spread in *Morpheus* dominant classifications for K15 objects with a distinctive morphology. Shown are the output *Morpheus* classification distributions for K15 objects where all visual classifiers agreed on the input classification. The *Morpheus* pixel-by-pixel classifications computed for the HLF GOODS South images were aggregated into flux-weighted object-by-object classifications following Equation 2.16 using the K15 segmentation maps. The results demonstrate that *Morpheus* can reproduce the results of the dominant K15 visual classifications for objects with distinct morphologies, even as the *Morpheus* classifications were computed from per-pixel classifications using different FITS images of the same region of the sky. 57

2.17 Morphology as a function of stellar mass and redshift for 54,000 sources in the five CANDELS fields. Sources included in the plot are those where $H < 24.5$AB and the *Morpheus* confidence for *spheroid*, *disk*, or *irregular* is greater than 0.7. See Section 2.7.1. . 60

2.18 False negative test for the *Morpheus* source detection scheme. Simulated sources with different signal-to-noise ratios ($SNRs$) were inserted into a noise image and then recovered by *Morpheus*, which assigns a low *background* value to regions it identifies as containing source flux (see Section 2.7.2). Shown are lines corresponding to the number of pixels assigned to sources of different SNR, as a function of the *background* threshold. As trained on the K15 sample, *Morpheus* becomes incomplete for objects with $SNR \lesssim 15$, and is more complete if the threshold for identifying sources is made more permissive (i.e., at a higher *background* value). 63

2.19 Morphological classifications as a function of simulated source surface brightness profile Sersic index. Shown are the *Morpheus* classification distributions for simulated $SNR = 20$ objects with circular Sersic [232] profiles, as a function of the Sersic index $\eta \in [1, 9]$. The experiment was repeated on objects with effective radii of three (upper left panel), five (upper right panel), seven (lower left panel), and nine (lower right panel) pixels. Objects with $\eta = 1$ were dominantly classified as *disk* or *spheroid*. Intermediate Sersic profiles ($\eta \sim 2 - 3$) were mostly classified as *spheroid*. Objects with high Sersic index ($\eta \geq 4$) were classified as *point source/compact*. These simulated objects with azimuthally symmetrical surface brightness profiles were assigned almost no *irregular* classifications by *Morpheus*. 66

2.20 Dominant morphological classification as a function of simulated source surface brightness profile Sersic index η and effective radius R_e in pixels. Each element of the matrix is color-coded to indicate the dominant *Morpheus* classification assigned for each $[\eta, R_e]$ pair, with the saturation of the color corresponding to the difference between the dominant and second *Morpheus* classification values. Large objects with low Sersic index are classified as *disk* (blue). Large objects with high Sersic index are classified as *spheroid* (red). Small objects with high Sersic index are classified as *point source/-compact* (yellow). None of the symmetrical objects in the test were classified as *irregular* (green). 67

2.21 Two-dimensional histogram of *Morpheus background* values and 3D-HST source flux in GOODS South. Shown is the distribution of *background* at the location of 3D-HST sources [239, 185] in GOODS South of various H-band magnitudes, along with the marginal histograms for both quantities (side panels). For reference, the K15 completeness (green line) and 3D-HST 90% completeness (red line) flux limits are also shown. The 3D-HST sources most frequently have *background*=0, and the majority of 3D-HST sources of any flux $H < 29$ have *background*<0.5. The *background* values for objects where K15 and 3D-HST are complete is frequently zero. The *Morpheus background* values increase for many objects at flux levels $H > 26$AB. 68

2.22 Completeness of *Morpheus* in source detection relative to 3D-HST [239, 185] in GOODS South. Shown is the fraction of 3D-HST sources in GOODS South detected by *Morpheus* brighter than some H-band source magnitude, for different *background* thresholds defining a detection (purple lines). The inset shows the *Morpheus* completeness for the brightest objects where 3D-HST (red line and arrow) and K15 (green line and arrow) are both highly complete. The completeness of *Morpheus* relative to 3D-HST is >90% where 3D-HST is highly complete. The completeness of *Morpheus* declines rapidly at faint magnitudes ($H \gtrsim 26.5$), but some objects are detected to $H \sim 29$, about 100× fainter than objects in the training set. 69

2.23 Source detection completeness as a function of color for sources with an H-band (F160W) AB magnitude of H <24.5. Sources that had a V band flux less than the V band error, had their flux replaced with three times the error value to limit unrealistically large V-H values. *Morpheus* does not show bias in the detection of objects with respect to color. There is a dip in completeness at $V - H \sim 0.2$, where the completeness is ∼75%. However, this bin only has four sources, indicating *Morpheus* only missed one source at this color. 71

2.24 Morphological classification as a function of object flux in GOODS South. Shown are the fraction of 3D-HST objects (see left axis) with *Morpheus* dominant, flux-weighted classifications of *spheroid* (red line), *disk* (blue line), *irregular* (green line), and *point source/-compact* (yellow line), each as a function of their H-band ($F160W$) AB magnitude. The brightest objects in the image are stars that are classified as *point source/compact*. The faintest objects in the image are compact faint galaxies classified as *point source/compact* or irregular. At intermediate fluxes, the objects are primarily classified as *disk* and *spheroid*. Also shown as a gray histogram (see right axis) is the number of 3D-HST objects detected and classified by *Morpheus* with source magnitude. 73

2.25 Example automated morphological decomposition by *Morpheus*. The left panel shows the VzH multi-color image of a galaxy in GOODS South from the Hubble Legacy Fields. The disk galaxy, 3D-HST ID 46386, has a prominent central bulge. The right panel shows the *Morpheus* classification color image, with pixels displaying *spheroid*, *disk*, *irregular*, or *point source/compact* dominant morphologies shown in red, blue, green, and yellow, respectively. The figure demonstrates that *Morpheus* correctly classifies the spheroid and disk structural components of the galaxy correctly, even though the training process for *Morpheus* does not involve spatially-varying morphologies for galaxy interiors. I note that there is a large-scale image artifact in $F850LP$ that appears as green in the left image, but does not strongly affect the *Morpheus* pixel-level classifications. 80

2.26 Example of morphological deblending by *Morpheus*. The leftmost panel shows the VzH image of a star-galaxy blend in GOODS South from the Hubble Legacy Fields. The star, 3D-HST ID 601, overlaps with a spheroidal galaxy 3D-HST ID 543. The center panel shows the *Morpheus* classification color image, with pixels displaying *spheroid*, *disk*, *irregular*, or *point source/compact* dominant morphologies shown in red, blue, green, and yellow, respectively. The pixel regions dominated by the star or spheroid are correctly classified by *Morpheus*. The right panel shows the resulting *Morpheus* segmentation map, illustrating that the dominant object classification in each segmentation region is also correct. The pixel-level classifications could be used to refine the segmentation to more precisely include only pixels that contained a single dominant class. The green feature in the left panel is an image artifact in $F850LP$. 81

3.1 End-to-end example using our method to detect and deblend sources. Starting from the left: A flux image is input to the Model (see Section 3.3.2). The Model outputs the deblended image in the *Partial Claim Representation* (PCR; see Section 3.2).The output from the model is then decoded using the non-learned PCR Decoder algorithm into separate deblended source images. The deblended source images have their total flux within r_{90} annotated. The deblended source images are then added together to generate the reconstructed image which has an L_1 total flux difference of 1.52×10^{-5} with the original input image. 93

4.1 An overview of the *Map Generator* in the *FitsMap* architecture. The *Map Generator*, described in Section 4.2.1, processes the input image and catalog files and converts them into a format that can be rendered by the *Map Viewer* (Section 4.2.2). The *Map Generator* can process multiple files in parallel and further leverage parallelism when processing each file. The *Map Generator* builds a directory containing the tiled image and catalog data along with the *Map Viewer* data and code (gray). The output website can be viewed using a simple web server. 108

4.2 An illustration of tiling image and catalog data at three zoom levels (0, 1, 2). (left) Illustration of tiling image data only. (right) Illustration of tiling image and catalog data. For simplicity, not all sources in the image are labeled with a marker. *FitsMap* computes the cluster assignments at every zoom level and stores them as markers in their respective tile. When the *Map Viewer* requests the catalog data associated with a particular tile, it will retrieve source markers and/or cluster markers and renders them appropriately. Source image credit: NASA. 109

4.3 An example of the *FitsMap* interface. (a) The search function button searches catalogs by `id` (see Section 4.2.2). (b) A marker popup displaying catalog data associated with the indicated source. (c) The `Leaflet` layer control allows users to switch between display images and catalog overlays. Image Credit: [75]. 110

4.4 Viewing the catalog clustering levels as a function of zoom level. Starting from the upper-left panel and following the arrows until the upper right panel are progressively increasing zoomed-in views of the data released by Hausen and Robertson [104]. Two catalogs are displayed simultaneously, indicated in blue and orange. The final bottom right panel shows the same image as the upper right panel, but with the popup for one of the catalogs. 111

4.5 FitsMap performance parsing and tiling image and catalog data. (a) The time, measured in seconds, to tile and generate a FitsMap consisting of a single image. The image size is indicated along the x-axis, where the image size is $\sqrt{n_{\text{pixels}}} \times \sqrt{n_{\text{pixels}}}$. The primary y-axis (blue) indicates the total amount of time to tile the image and generate the output website directory. Every image is generated five times with the mean plotted as a blue dashed line, and a single standard deviation shaded as light blue around the mean line. The secondary y-axis (purple) indicates the tot number of tiles generated in the output tiled image directory. (b) The time, measured in seconds, to parse a tile catalog file. Each catalog was scaled to fit an image size of $65,536 \times 65,536$, resulting in approximately $87,381$ tiles. Each catalog was parsed and tiled fives times with the mean plotted as the blue dotted line, and a single standard deviation shaded around the mean in light blue. 112

4.6 Storage comparison of catalog data and its CBOR-encoding used by *FitsMap*. The primary axis (black) shows the total storage use of the catalog data and the CBOR-encoded version (see Section 4.2.1 for more information on CBOR encoding). The secondary axis (purple) shows the ratio of the storage requirements of the CBOR encoded version of the catalog data to the original catalog data. 113

4.7 The ratio of the total storage requirements of the tiled catalog data to the original catalog size. The x-axis indicates the image size, and the y-axis indicates the catalog size. The color in the image indicates the value of the ratio, where purple indicates a value less than one, white indicates a value near or equal to one, and blue indicates a value greater than one. 113

4.8 *FitsMap* page speed metrics for mobile browsers. *Page Load* indicates the amount of time required for the page to load fully after sending the request to the server. *FCP* (First Contentful Paint) indicates the amount of time from page loading to seeing something rendered on the page. *DOM Processing* indicates the amount of time after the browser starts processing the page until the content is loaded. Each test was performed three times with the mean and a standard deviation plotted. 114

4.9 *FitsMap* page speed metrics for desktop browsers. *Page Load* indicates the amount of time it takes for the page to load fully. *FCP* (First Contentful Paint) indicates the amount of time from page loading to seeing something rendered on the page. *DOM Processing* indicates the amount of time after the browser starts processing the page until the content is loaded. Each test was performed three times with the mean and a standard deviation plotted. 115

4.10 *Map Viewer Session Memory Consumption* during a typical interaction on a test *FitsMap*. The website was generated according to the procedure described in Section 4.3.2. The steps taken during the test are indicated in order along the x-axis. The memory consumption is indicated on the y-axis in megabytes. The tests were run using the Selenium browser automation tool and run on the BrowserStack test framework. Each test was run three times with the mean and the standard deviation reported above. Memory consumption reporting is reported for Chrome browsers which support reporting memory consumption via JavaScript. 115

5.1 Top seven features with the highest average contribution in the EBM $\gamma(SFR|\boldsymbol{\theta})$ targeting the star formation rate *SFR*. In order of decreasing importance, these features include peak circular velocity v_{peak}, virial mass M_{vir}, environmental density ρ_1, redshift z, environmental temperature T_1, the mass ratio of nearby halos $\Upsilon_{0.1}$, and the interaction between virial mass M_{vir} and $\Upsilon_{0.1}$. Average contribution is calculated using the average of the absolute value of the feature functions weighted by the number of samples in each bin (see Equation 5.5). 147

5.2 Learned univariate feature functions f_y^i for the EBM $\gamma(SFR|\boldsymbol{\theta})$ trained to predict the star formation rate *SFR*. Shown (left to right) are the feature functions for peak circular velocity v_{peak}, virial mass M_{vir}, environmental density ρ_1, redshift z, environmental temperature T_1, and nearby halo mass ratio $\Upsilon_{0.1}$. Light blue areas indicate regions where $f_y^i > 0$ and dark blue areas indicate regions where $f_y^i < 0$. The shaded areas show the variation in f_y^i between the k-fold iterations. 148

5.3 Most important learned interaction functions f_y^{ij} for the EBM model $\gamma(SFR|\boldsymbol{\theta})$ targeting the star formation rate *SFR*, as a function of their parameter pairs. Each panel shows the contribution of the bivariate interaction terms, normalized such that the color map ranges between plus or minus the maximum of the norm of each function $||f||_{\rm max}$. Light blue areas indicate regions of joint parameter space where the feature interactions contribute positively to the star formation rate, while dark blue areas indicate regions with negative contributions. The table lists $||f||_{\rm max}$ for the interaction functions, each with units $\log_{10} M_\odot$ yr^{-1}. In absolute terms, the largest interaction occurs for halos with large peak circular velocity $v_{\rm peak}$ and no large neighboring halos ($\Upsilon_{0.1} \approx 0$). The other interaction functions are relatively weak, and contribute changes to $\log SFR \lesssim 0.05$. 149

5.4 Summary of the EBM model $\gamma(SFR|\boldsymbol{\theta})$ targeting star formation rate (*SFR*) as a function of virial mass. The upper left panel shows the two-dimensional distribution of *SFR* with $M_{\rm vir}$ for galaxies in the CROC simulations, with the color scale showing the number of simulated galaxies at each $[SFR, M_{\rm vir}]$ location. The lower left panel shows the EBM model results for the distribution of *SFR* with $M_{\rm vir}$, where the *SFR* is computed from the EBM using the parameters $\boldsymbol{\theta} = [M_{\rm vir}, v_{\rm peak}, z, \rho_1, T_1, \Upsilon_{0.1}]$. The upper right panel shows the residuals between the simulated CROC galaxy *SFR*s and the EBM model results. The lower right panel shows the simulated CROC galaxy *SFR*s that lie outside the EBM model predictions. These outliers represent $\lesssim 3\%$ of simulated CROC galaxies. 150

5.5 Summary of the EBM model $\gamma(M_\star|\boldsymbol{\theta})$ targeting stellar mass $M_\star$ as a function of virial mass. The upper left panel shows the distribution of $M_\star$ with virial mass $M_{\rm vir}$ in the CROC simulated galaxy catalogs, with the coloration indicating the number of galaxies at each $[M_\star, M_{\rm vir}]$ location. The lower left panel shows the EBM model prediction of the stellar mass distribution with virial mass given in the input parameters $\boldsymbol{\theta} = [M_{\rm vir}, z, v_{\rm peak}, \rho_1, T_1, \Upsilon_{0.1}]$. The upper right panel shows the residuals between the simulated and predicted $M_\star$ vs. $M_{\rm vir}$ distribution, and the lower right panel shows the outliers in the simulated distribution not captured by the EBM model $\gamma(M_\star|\boldsymbol{\theta})$. The fraction of outliers is $\lesssim 1\%$. 151

5.6 Learned univariate feature functions, f_y^i in Equation 5.1, for the EBM $\gamma(M_\star|\boldsymbol{\theta})$ trained to predict $M_\star$. Areas highlighted in orange indicate portions of the function that contribute positively to the predicted $M_\star$ and areas in red contribute negatively. Stellar mass increases with peak circular velocity and virial mass, increases with decreasing redshift, and increases with environmental density. Temperature correlates positively with stellar mass, with a strong feature near $T_1 \approx 10^4$ K where hydrogen ionizes. Stellar mass also increases with the mass ratio of neighboring halos. 152

5.7 Features with the highest average contribution for the EBM $\gamma(M_\star|\boldsymbol{\theta})$ trained to predict $M_\star$. Average contribution is calculated using the average of the absolute value of the learned functions weighted by the number of samples in each bin (see Equation 5.5). The features with the largest contributon are $v_{\rm peak}$ and $M_{\rm vir}$, followed by redshift z, environmental density ρ_1, environmental temperature T_1, and mass ratio of nearby halos $\Upsilon_{0.1}$. The interaction with the largest average contribution involves [z,$v_{\rm peak}$]. 153

5.8 Learned bivariate interaction functions f_y^{ij} for the EBM $\gamma(M_\star|\boldsymbol{\theta})$ trained to predict $M_\star$. Areas highlighted in orange indicate portions of the functions that contribute positively to the predicted $M_\star$ while areas in red contribute negatively. Halos with large environmental temperatures T_1 at high redshift z show enhanced stellar mass. The stellar masses of halos with low environmental temperature $T_1 < 10^4$K correlate with environmental density, increasing with increasing ρ_1. Massive halos with no comparable large neighboring halos ($\Upsilon_{0.1} \approx 0$) also show enhanced stellar mass. 154

5.9 Details for the base EBM model $\gamma(SFR|\boldsymbol{\theta}')$ component of the CEBM $\Gamma(SFR|\boldsymbol{\theta}')$ trained to predict *SFR*. Panel a) displays the average contribution of features. Panel b) shows the feature functions contributing to the base EBM model. Panel c) presents the interaction functions for the base EBM $\gamma(SFR|\boldsymbol{\theta}')$. Each panel shows the contribution of the bivariate interaction terms, normalized such that the color map ranges between plus or minus the maximum of the norm of each function $||f||_{\max}$. Purple indicates negative contributions and blue indicates positive contributions. The table lists $||f||_{\max}$ for the interaction functions, each with units $\log_{10} M_\odot$ yr^{-1}. 155

5.10 Details for the outlier EBM model $\delta(SFR|\boldsymbol{\theta}')$ component of the CEBM $\Gamma(SFR|\boldsymbol{\theta}')$ trained to predict *SFR*. Panel a) displays the average contribution of features. Panel b) shows the feature functions for the outlier EBM $\delta(SFR|\boldsymbol{\theta}')$. Panel c) presents the interaction functions for the outlier EBM $\delta(SFR|\boldsymbol{\theta}')$. Each panel shows the contribution of the bivariate interaction terms, normalized such that the color map ranges between plus or minus the maximum of the norm of each function $||f||_{\max}$. Purple indicates negative contributions and blue indicates positive contributions. The table lists $||f||_{\max}$ for the interaction functions, each with units $\log_{10} M_\odot$ yr^{-1}. 156

5.11 Details for the classification EBM model $\phi_{SFR}(\boldsymbol{\theta}')$ that interpolates between the base EBM $\gamma(SFR|\boldsymbol{\theta}')$ and the outlier EBM $\delta(SFR|\boldsymbol{\theta}')$ for creating the CEBM $\Gamma(SFR|\boldsymbol{\theta}')$. Panel a) displays the average contribution of features to the classification EBM model $\phi_{SFR}(\boldsymbol{\theta}')$. Panel b) shows the feature functions contributing to the classifier EBM $\phi_{SFR}(\boldsymbol{\theta}')$. Panel c) presents the interaction functions for the classifier EBM $\phi_{SFR}(\boldsymbol{\theta}')$. Each panel shows the contributions of the interaction terms, normalized such that the color map ranges between plus or minus the maximum of the norm of each function $||f||_{\max}$. Purple indicates negative log odds and blue indicates positive log odds that a given galaxy is an outlier in SFR. The table lists $||f||_{\max}$ for the interaction functions, listed as the corresponding change in log odds. 157

5.12 Details for the CEBM model $\Gamma(SFR|\boldsymbol{\theta}')$ trained to predict *SFR*. Panel a) displays the average contribution of features to the CEBM. Virial mass M_{vir} provides the largest average contribution to the star formation rate. Panel b) shows the feature functions contributing to the CEBM $\Gamma(SFR|\boldsymbol{\theta}')$. Panel c) presents the interaction functions for the CEBM $\Gamma(SFR|\boldsymbol{\theta}')$. Each panel shows the contribution of the interaction terms, normalized such that the color map ranges between plus or minus the maximum of the norm of each function $||f||_{\max}$. Purple indicates negative contributions and blue indicates positive contributions. The table lists $||f||_{\max}$ for the interaction functions, each with units $\log_{10} M_{\odot}\ \text{yr}^{-1}$. 158

5.13 Summary of the CEBM model $\Gamma(SFR|\boldsymbol{\theta'})$ targeting star formation rate (SFR) as a function of virial mass. The upper left panel shows the two-dimensional distribution of SFR with $M_{\rm vir}$ for galaxies in the CROC simulations, with the color scale showing the number of simulated galaxies at each $[SFR, M_{\rm vir}]$ location. The lower left panel shows the CEBM model results for the distribution of SFR with $M_{\rm vir}$, where the SFR is computed from the CEBM using the parameters $\boldsymbol{\theta'} = [M_{\rm vir}, z, \rho_1, T_1, \Upsilon_{0.1}]$. The upper right panel shows the residuals between the simulated CROC galaxy SFRs and the CEBM model results. The lower right panel shows the simulated CROC galaxy SFRs that lie outside the CEBM model predictions. These outliers represent $\approx 5\%$ of simulated CROC galaxies. 159

5.14 Summary of the CEBM model $\Gamma(M_\star|\boldsymbol{\theta'})$ targeting stellar mass $M_\star$ as a function of virial mass. The upper left panel shows the distribution of $M_\star$ with virial mass $M_{\rm vir}$ in the CROC simulated galaxy catalogs, with the coloration indicating the number of galaxies at each $[M_\star, M_{\rm vir}]$ location. The lower left panel shows the CEBM model prediction of the stellar mass distribution with virial mass given in the input parameters $\boldsymbol{\theta'} = [M_{\rm vir}, z, \rho_1, T_1, \Upsilon_{0.1}]$. The upper right panel shows the residuals between the simulated and predicted $M_\star$ vs. $M_{\rm vir}$ distribution, and the lower right panel shows the outliers in the simulated distribution not captured by the CEBM model $\Gamma(M_\star|\boldsymbol{\theta'})$. The fraction of outliers is $\lesssim 2\%$. 160

5.15 Details for the base EBM model $\gamma(M_\star|\boldsymbol{\theta}')$ component of the CEBM $\Gamma(M_\star|\boldsymbol{\theta}')$ trained to predict stellar mass $M_\star$. Panel a) displays the average contribution of features to the base EBM model $\gamma(M_\star|\boldsymbol{\theta}')$. Panel b) shows the feature functions contributing to the base EBM model $\gamma(M_\star|\boldsymbol{\theta}')$. Panel c) presents the interaction functions for the base EBM $\gamma(M_\star|\boldsymbol{\theta}')$. Each panel shows the contribution of the bivariate interaction terms, normalized such that the color map ranges between plus or minus the maximum of the norm of each function $||f||_{\max}$. Teal indicates negative contributions and green indicates positive contributions. The table lists $||f||_{\max}$ for the interaction functions, each with units $\log_{10} M_\odot$. 161

5.16 Details for the outlier EBM model $\delta(M_\star|\boldsymbol{\theta}')$ component of the CEBM $\Gamma(M_\star|\boldsymbol{\theta}')$ trained to predict $M_\star$. Panel a) displays the average contribution of features to the outlier EBM model $\delta(M_\star|\boldsymbol{\theta}')$. Panel b) shows the feature functions for the outlier EBM $\delta(M_\star|\boldsymbol{\theta}')$. qPanel c) presents the interaction functions for the outlier EBM $\delta(M_\star|\boldsymbol{\theta}')$. Each panel shows the contribution of the interaction terms, normalized such that the color map ranges between plus or minus the maximum of the norm of each function $||f||_{\max}$. Teal indicates negative contributions and green indicates positive contributions. The table lists $||f||_{\max}$ for the interaction functions, each with units $\log_{10} M_\odot$. 162

5.17 Details for the classification EBM model $\phi_{M_\star}(\boldsymbol{\theta'})$ that interpolates between the base EBM $\gamma(M_\star|\boldsymbol{\theta'})$ and the outlier EBM $\delta(M_\star|\boldsymbol{\theta'})$ for creating the CEBM $\Gamma(M_\star|\boldsymbol{\theta'})$. Panel a) displays the average contribution of features to the classification EBM model $\phi_{M_\star}(\boldsymbol{\theta'})$. Panel b) shows the feature functions contributing to the classifier EBM $\phi_{M_\star}(\boldsymbol{\theta'})$. These feature functions represent the change in log odds that a given galaxy will be an outlier in $M_\star$. Panel c) presents the interaction functions for the classifier EBM $\phi_{M_\star}(\boldsymbol{\theta'})$. Each panel shows the contributions of the interaction terms, normalized such that the color map ranges between plus or minus the maximum of the norm of each function $||f||_{\max}$. Teal indicates negative log odds and green indicates positive log odds that a given galaxy is an outlier in stellar mass. The table lists $||f||_{\max}$ for the interaction functions, listed as the corresponding change in log odds. 163

5.18 Details for the CEBM model $\Gamma(M_\star|\boldsymbol{\theta'})$ trained to predict stellar mass $M_\star$. Panel a) displays the average contribution of features to the CEBM model $\Gamma(M_\star|\boldsymbol{\theta'})$. Panel b) shows the feature functions contributing to the CEBM $\Gamma(M_\star|\boldsymbol{\theta'})$. Panel c) presents the interaction functions for the CEBM $\Gamma(M_\star|\boldsymbol{\theta'})$. Each panel shows the contribution of the interaction terms, normalized such that the color map ranges between plus or minus the maximum of the norm of each function $||f||_{\max}$. Teal indicates negative contributions and green indicates positive contributions. The table lists $||f||_{\max}$ for the interaction functions, each with units $\log_{10} M_\odot$. 164

A.1 Schematic of a simple neural network. Given an input vector **x**, the neural network applies a series of reductions and nonlinear transformations through a collection of layers **L** to produce an output **o**. Each layer L consists of a set of artificial neurons **AN** that perform a linear rescaling of their input data, followed by a nonlinear transformation via the application of an activation function (see Equation A.2). The activation function may vary across layers. . . 169

A.2 Schematic of a convolutional neural network (CNN). Shown is a simplified CNN consisting of a convolutional layer feeding a fully connected layer. Each artificial neuron (AN) in the convolutional layer outputs a feature map as described by Equation A.7. Each output feature map is flattened and concatenated into a single vector. This vector is processed by each AN in the fully connected layer (see Equation A.2). The curly brace represents connections from all elements of the vector input. 171

A.3 Comparison of max and average pooling layers. Pooling layers perform reductions on subsets of feature maps, providing a local average or maximum of data elements in a window (2×2 in this schematic). Shown are cells of an input feature map (left), color-coded within a window to match the corresponding regions of the output feature map (right). The pooling layers perform a simple reduction with these windows, taking either a maximum (upper branch) or average (lower branch). 172

A.4 Color composite of the HR2020 morphological classifications for the COSMOS field from the CANDELS survey [95, 144]. 177

A.5 Color composite of the HR2020 morphological classifications for the EGS field from the CANDELS survey [95, 144]. 178

A.6 Color composite of the HR2020 morphological classifications for the GOODS North field from the CANDELS survey [95, 144]. 179

A.7 Color composite of the HR2020 morphological classifications for the UDS field from the CANDELS survey [95, 144]. 180

List of Tables

2.1 Computational steps in the *Morpheus* deep learning framework. For each Layer (left column), I list its Input (center column), and the Output Shape of its data (right column). The model takes as its starting input a set of images in n_b bands, each with $N \times M$ pixels. The final output of the model is a set of n_c classification images, each with $N \times M$ pixels. The *Morpheus* block structures are illustrated in Figure 2.1. The "+" symbol denotes a concatenation between two layer outputs, as shown in Figure 2.2. 19

2.2 *Adam* optimizer [140] hyper-parameter values used during the training of the neural network used in *Morpheus*. See the text for definitions of the hyper-parameters. 30

2.3 *Morpheus* training and test results for accuracy $\mathcal{A}$, and intersection-over-union $\mathcal{I}_{\mathcal{U}}$ as a function of *background* threshold B. 31

2.4 Summary of sources identified by *Morpheus* in GOODS-S that were absent in the CANDELS or 3D-HST catalogs. Of the 19,481 sources identified by *Morpheus* in a subregion of GOODS-S, 123 sources did not have CANDELS or 3D-HST counterparts. Upon visual inspection, these objects could be categorized as either *image artifacts*, *poor deblends* where *Morpheus* had shredded sources, *missed sources* corresponding to real objects missed by CANDELS and 3D-HST, or *actual false positives* incorrectly identified as *Morpheus* as real sources. The false positive rate for the *Morpheus* algorithm is only roughly 0.09%, defined relative to the CANDELS and 3D-HST catalogs. See Section 2.7.6 for more discussion. . . . 75
2.5 *Morpheus* + 3D-HST Value Added Catalog for GOODS South . . 77

3.1 Detection and deblending method categorization 89
3.2 *Partial Claim Representation* encoding efficacy 92
3.3 Training metric results 94

4.1 *Map Viewer* testing environments. Test environments were provided by and executed using BrowserStack. All mobile environments have a network speed throttled to model 4G speeds (9Mbps), and all desktop environments were throttled to have a network speed throttled to model broadband speeds (40Mbps). 106

5.1 Hyperparameters used to train the InterpretML [196] implementation of the EBM. All other model hyperparameters were set to the default values for InterpretML version 0.2.7. 126
5.2 Training results for the EBM using k-fold cross validation. See Section 5.2.3 for more information on the training process. Reported are values for the variance metric r^2, the outlier fraction ζ, and the mean absolute error (MAE). Uncertainties are computed from the variation among the k-fold trials. 126

5.3 Summary of the EBM model trained to predict *SFR*. The first entry, $\beta_{\log_{10} SFR}$, is the baseline value learned model (see Section 5.2.1). The next seven entries are the average contributions of the most important feature functions listed in descending order (see Equation 5.5). 129

5.4 Summary of the EBM model $\gamma(M_\star|\boldsymbol{\theta})$ trained to predict $M_\star$ as a function of the full parameter set $\boldsymbol{\theta}$. The first entry, $\beta_{\log_{10} M_\star}$, is the learned baseline value of the model (see Section 5.2.1). The next seven entries are the feature functions with the highest average contribution in descending order. Average contribution is calculated using the average of the absolute value of the feature functions weighted by the number of samples in each bin (see Equation 5.5). 131

5.5 Training results for CEBM models for *SFR* and $M_\star$ using k-fold cross validation. See Section 5.2.3 for more information on the training process. Reported are values for the variance metric r^2, the outlier fraction ζ, and the mean absolute error (MAE). Uncertainties are computed from the variation among the k-fold trials. . 138

5.6 Average contribution to the CEBM model $\Gamma(SFR|\boldsymbol{\theta}')$ trained to predict *SFR* from the parameter set $\boldsymbol{\theta}'$. The first entry, $\beta_{\log_{10} SFR}$, is the learned baseline of the model. The next seven entries are the feature functions with the highest average contribution listed in descending order. The average contribution is calculated using the average of the absolute value of the base EBM function values weighted by the number of samples in each bin and the output of the classification EBM for each sample (see Section 5.4.2 for more details). 139

5.7 Summary of the CEBM model $\Gamma(M_\star|\boldsymbol{\theta}')$ trained to predict $M_\star$ using the restricted parameter set $\boldsymbol{\theta}'$. The first entry, $\beta_{\log_{10} M_\star}$, is the learned baseline of the model. The next seven entries are the learned functions with the highest average contribution in descending order. The average contribution is computed via Equation 5.8 (see Section 5.4.2 for more details). 141

A.1 Data release files generated by Morpheus and associated URLs for the COSMOS CANDELS field. The data release files for each field are organized into three groups: *pixel-level morphological classifications*, *segmentation maps*, and *3D-HST value-added catalogs*. The *pixel-level morphological classification* files are named according to the following scheme `morpheus_COSMOS_[morphology].v1.0.fits`, where `[morphology]` can be one of the morphological classes (*spheroid*, *disk*, *irregular*, *ps_compact*, *background*) or *mask*, a binary image mask indicating which pixels in the image were classified by HR2020. The *segmentation map* files are named according to the following scheme `morpheus_COSMOS_[segmap_type].v1.0.fits` , where `[segmap_type]` can be *3dhst-segmap* (indicating the 3D-HST informed segmap) or *segmap* (indicating a segmap based only on background class/flux values). Finally, the 3D-HST value-added catalog files are named according to the following scheme `morpheus_-COSMOS_3dhst-catalog.v1.0.[file_type]`, where `[file_type]` can be *csv* for a comma-separated-value version of the value-added catalog and *txt* for the machine-readable table version described in Table 2.5. Additionally, a link to an archive containing all of the files associated with the COSMOS field is available in an additional section called *All Files*. See Appendix A.4 for details. 181

A.2 Data release files generated by Morpheus and associated URLs for the EGS CANDELS field. The data release files for each field are organized into three groups: *pixel-level morphological classifications*, *segmentation maps*, and *3D-HST value-added catalogs*. The *pixel-level morphological classification* files are named according to the following scheme `morpheus_EGS_[morphology].v1.0.fits`, where `[morphology]` can be one of the morphological classes (*spheroid*, *disk*, *irregular*, *ps_compact*, *background*) or *mask*, a binary image mask indicating which pixels in the image were classified by HR2020. The *segmentation map* files are named according to the following scheme `morpheus_EGS_[segmap_type].v1.0.fits` , where `[segmap_type]` can be *3dhst-segmap* (indicating the 3D-HST informed segmap) or *segmap* (indicating a segmap based only on background class/flux values). Finally, the 3D-HST value-added catalog files are named according to the following scheme `morpheus_-EGS_3dhst-catalog.v1.0.[file_type]`, where `[file_type]` can be *csv* for a comma-separated-value version of the value-added catalog and *txt* for the machine-readable table version described in Table 2.5. Additionally, a link to an archive containing all of the files associated with the EGS field is available in an additional section called *All Files*. See Appendix A.4 for details. 182

A.3 Data release files generated by Morpheus and associated URLs for the GOODS North CANDELS field. The data release files for each field are organized into three groups: *pixel-level morphological classifications*, *segmentation maps*, and *3D-HST value-added catalogs*. The *pixel-level morphological classification* files are named according to the following scheme `morpheus_GOODS-N_-[morphology].v1.0.fits`, where `[morphology]` can be one of the morphological classes (*spheroid*, *disk*, *irregular*, *ps_compact*, *background*) or *mask*, a binary image mask indicating which pixels in the image were classified by HR2020. The *segmentation map* files are named according to the following scheme `morpheus_GOODS-N_-[segmap_type].v1.0.fits` , where `[segmap_type]` can be *3dhst-segmap* (indicating the 3D-HST informed segmap) or *segmap* (indicating a segmap based only on background class/flux values). Finally, the 3D-HST value-added catalog files are named according to the following scheme `morpheus_GOODS-N_3dhst-catalog.v1.0.[file_-type]`, where `[file_type]` can be *csv* for a comma-separated-value version of the value-added catalog and *txt* for the machine-readable table version described in Table 2.5. Additionally, a link to an archive containing all of the files associated with the GOODS North field is available in an additional section called *All Files*. See Appendix A.4 for details. 183

A.4 Data release files generated by Morpheus and associated URLs for the GOODS South CANDELS field. The data release files for each field are organized into three groups: *pixel-level morphological classifications*, *segmentation maps*, and *3D-HST value-added catalogs*. The *pixel-level morphological classification* files are named according to the following scheme `morpheus_GOODS-S_-[morphology].v1.0.fits`, where `[morphology]` can be one of the morphological classes (*spheroid*, *disk*, *irregular*, *ps_compact*, *background*) or *mask*, a binary image mask indicating which pixels in the image were classified by HR2020. The *segmentation map* files are named according to the following scheme `morpheus_GOODS-S_-[segmap_type].v1.0.fits` , where `[segmap_type]` can be *3dhst-segmap* (indicating the 3D-HST informed segmap) or *segmap* (indicating a segmap based only on background class/flux values). Finally, the 3D-HST value-added catalog files are named according to the following scheme `morpheus_GOODS-S_3dhst-catalog.v1.0.[file_-type]`, where `[file_type]` can be *csv* for a comma-separated-value version of the value-added catalog and *txt* for the machine-readable table version described in Table 2.5. Additionally, a link to an archive containing all of the files associated with the GOODS South field is available in an additional section called *All Files*. See Appendix A.4 for details. 184

A.5 Data release files generated by Morpheus and associated URLs for the UDS CANDELS field. The data release files for each field are organized into three groups: *pixel-level morphological classifications*, *segmentation maps*, and *3D-HST value-added catalogs*. The *pixel-level morphological classification* files are named according to the following scheme `morpheus_UDS_[morphology].v1.0.fits`, where `[morphology]` can be one of the morphological classes (*spheroid*, *disk*, *irregular*, *ps_compact*, *background*) or *mask*, a binary image mask indicating which pixels in the image were classified by HR2020. The *segmentation map* files are named according to the following scheme `morpheus_UDS_[segmap_type].v1.0.fits` , where `[segmap_type]` can be *3dhst-segmap* (indicating the 3D-HST informed segmap) or *segmap* (indicating a segmap based only on background class/flux values). Finally, the 3D-HST value-added catalog files are named according to the following scheme `morpheus_-UDS_3dhst-catalog.v1.0.[file_type]`, where `[file_type]` can be *csv* for a comma-separated-value version of the value-added catalog and *txt* for the machine-readable table version described in Table 2.5. Additionally, a link to an archive containing all of the files associated with the UDS field is available in an additional section called *All Files*. See Appendix A.4 for details. 185

Abstract

Large Scale Analysis of Astronomical Data Using Machine Learning and Visualization Techniques

by

Ryan David Hausen

The nature and volume of astronomical data present significant challenges in applying off-the-shelf machine learning and visualization methods. In this work, I present new machine learning and visualization techniques motivated by the needs of astronomical research. Specifically, this work presents novel approaches to source detection, deblending, and morphological classification that leverage recent advances in computer vision. Further, this work introduces FitsMap, a new tool for displaying image and catalog data that scales to large volumes of data and is performant on mobile devices. Finally, the relationships between the physical properties of simulated galaxies and their stellar mass and star formation rate are modeled using Explainable Boosting Machines, an interpretable machine learning model.

This work is dedicated to God, whose love and grace sustain me daily. It is also dedicated to my family, whose continual support and confidence gave me the strength to push through the difficult times.

Acknowledgments

I am incredibly grateful to Professor Brant Robertson, without whom this would not have been possible. His support, knowledge, and guidance have been invaluable in my development as a researcher. I am also deeply indebted to Professor Roberto Manduchi for his feedback and expertise. Further, I would like to thank Professor X. Prochaska and Professor Narges Norouzi for serving on my defense committee and for the opportunities to collaborate. Additionally, I would like to thank the University of California and NASA for funding my research.

A special thanks to the Computational Astrophysics Research Group and the Computer Vision Lab members for their help and feedback. I am also grateful to the Computer Science & Engineering and Astronomy & Astrophysics departments for working in tandem to make this interdisciplinary work possible.

I would like to thank Professor Bin Tang for allowing me to pursue research as an undergraduate, an opportunity that left a profound impact on me. Lastly, I would like to thank my family for believing in me and supporting me.

The text of this dissertation includes reprints of the following previously published material:

Hausen, R. and Robertson, B. E., *"Morpheus: A Deep Learning Framework for the Pixel-level Analysis of Astronomical Image Data"*, The Astrophysical Journal Supplement Series, vol. 248, no. 1, pp. 1-37, 2020.

Hausen, R. and Robertson, B. E., *"Partial-Attribution Instance Segmentation for Astronomical Source Detection and Deblending"*, Fourth Workshop on Machine Learning and the Physical Sciences, NeurIPS 2021.

Hausen, R. and Robertson, B. E., *"FitsMap: A Simple, Lightweight Tool For Displaying Interactive Astronomical Image and Catalog Data"*, Astronomy and

Computing, vol. 39, pp. 100586, 2022.

Hausen, R., Robertson B. E., Zhu, H., Gnedin, N. Y., Madau, P., Schneider, E. E., Villasenor, B., Drakos, N. E., *"Revealing the Galaxy-Halo Connection Through Machine Learning"*, Submitted to AAS Journals. arXiv:2204.10332, 2022.

The co-author, Brant E. Robertson, listed in the publications: "Morpheus: A Deep Learning Framework for the Pixel-level Analysis of Astronomical Image Data," "Partial-Attribution Instance Segmentation for Astronomical Source Detection and Deblending," and "FitsMap: A Simple, Lightweight Tool For Displaying Interactive Astronomical Image and Catalog Data," directed and supervised the research.

I contributed to the multiauthored publication, "Revealing the Galaxy-Halo Connection Through Machine Learning," in the following ways. I suggested the use of EBMs, came up with the idea and formalized the use of the Composite EBM, wrote the model training code, generated the manuscript figures, and contributed to manuscript writing.

Approval for reprinting the contents of the multiauthored publication "Revealing the Galaxy-Halo Connection Through Machine Learning" by the authors is listed in the supplemental files for this dissertation.

Chapter 1

Introduction

Astronomical data is richly complex, with galaxies and stars of various sizes, luminosities, opacities, and degrees of interaction and overlap. Compounding the complexity of the data is the fact that astronomy is in the midst of a big data revolution. The Vera Rubin Observatory (VRO) [122, 123], currently scheduled to come online in August 2023, will produce 20 terabytes of imaging data each night. The data volume produced by the VRO nightly is equivalent to the space required to store ≈ 66 copies of the ImageNet [68] dataset. The recently launched James Webb Space Telescope [JWST; for a review, see Robertson [221]] and the Nancy Grace Roman Space Telescope [RST; [245, 9]], currently scheduled for launch in May 2027, will capture the deepest images of space ever recorded. These images will see further back in time and increase the number and variety of galaxies to be analyzed. Beyond observations made with telescopes, astronomers study the universe using cosmological simulations [275, 274, 131, 209, 230, 276] that emulate how the modern universe formed. Large-scale cosmological simulations can use thousands of CPUs and GPUs with tens of terabytes of data products per simulation. Techniques to analyze and augment these cosmological simulations are an open area of research.

The vast volume of upcoming observational and simulation data is impossible to analyze using human means alone. However, machine learning and artificial intelligence are in an excellent position to meet the challenge. Inspired by the challenges and beauty of astronomical data, this work presents novel techniques for analyzing astronomical image and simulation data, representing an advance in state-of-the-art astronomical data analysis. Further, this work demonstrates that developing techniques for applications within astronomy can advance the state-of-the-art in computer science by revealing assumptions and corner cases that may not be apparent in working exclusively with the standard benchmark datasets. Ideally, the work shown here will motivate further interdisciplinary work mutually advancing the state-of-the-art in both computer science and astronomy.

Chapter 2 introduces the Morpheus deep learning framework. Deep learning has had profound success in source detection and the morphological classification of galaxies and stars. In particular, deep learning has been applied to detect faint x-ray sources [156], point sources [263, 253, 31, 45], radio sources [54, 87, 169, 255], galaxies [40, 34], point sources and galaxies [216, 295], point sources in gamma-rays [202], fast-moving near-earth objects [79], transient objects [89, 42, 231, 124, 186, 110, 171, 298], quasars [203], and low surface brightness galaxies [48]. Deep learning has also been applied with great success to the morphological classification of galaxies [70, 73, 113, 51, 246, 130, 88, 50, 277, 116, 206, 133, 301, 44, 150, 93, 5, 261, 97, 294, 30, 43, 218, 157, 215, 46, 47, 86] and radio galaxies [289, 11, 175, 18, 228, 250, 182, 36, 193, 84]. Further, classifiers have been trained to distinguish between stars and galaxies [139, 41, 40] and early and late-type galaxies [272, 252]. Morpheus makes two key contributions to detection and morphological classification. First, Morpheus introduces morphological classification at the pixel level. Second, Morpheus presents a scalable approach for applying pixel-level

machine learning techniques to native astronomical data.

Deblending, the process of separating the signal of two or more overlapping sources, is a well-known and studied problem but remains an open area of research. The most popular methods are SExtractor [22] and SCARLET [179]. More recently, deep learning approaches have been proposed for limited use cases [214, 34, 40], but a general method has yet to be developed. Chapter 3 introduces one such way, a novel computer vision problem setting with a direct application to astronomical source deblending called Partial-Attribution Instance Segmentation. Additionally, a neural network architecture is released as a proof of concept for the method.

Visualizing large-scale astronomical image and catalog data will become increasingly crucial as imaging data grows. Chapter 4 introduces FitsMap, a new tool for generating interactive visualizations of astronomical image and catalog data. FitsMap is a lightweight server/client architecture that can scale to arbitrarily large image and catalog data by leveraging a novel approach to storing catalog data.

In Chapter 5, Explainable Boosting Machines (EBM) are used to examine how stellar mass and star formation rate are related to the physical properties of simulated galaxies. Additionally, I propose the Composite Explainable Boosting Machine as an ensemble EBM method for using multiple EBMs to capture outlier trends in the data while maintaining the interpretability of the EBM. This work will help simulations that do not resolve some physical features of the data to more accurately predict the stellar mass and star formation rate of simulated galaxies.

The large volume of imaging, catalog, and simulation data present a unique opportunity for innovation. More accurate and faster models will accelerate our understanding of the greater universe and generalize machine learning methods.

Chapter 2

Morpheus: A Deep Learning Framework For Pixel-Level Analysis of Astronomical Image Data

2.1 Introduction

Morphology represents the structural end state of the galaxy formation process. Since at least Hubble [112], astronomers have connected the morphological character of galaxies to the physics governing their formation. Morphology can reflect the initial conditions of galaxy formation, dissipation, cosmic environment and large-scale tidal fields, merger and accretion history, internal dynamics, star formation, the influence of supermassive black holes, and a range of other physics [e.g., 26, 76, 27, 72, 77, 21, 257]. The development of morphological measures for galaxies, therefore, comprises an important task in observational astronomy. To

help realize the potential of current and future surveys for understanding galaxy formation through morphology, this chapter presents *Morpheus*, a deep learning-based model for the simultaneous detection and morphological classification of objects through the pixel-level semantic segmentation of large astronomical image datasets.

The established connections between morphology and the physics of galaxy formation run deep, and the way these connections manifest themselves observationally depends on the measures of morphology used. Galaxy size and surface brightness profile shape have served as common proxies for morphology, as quantitatively measured from the light distribution of objects [271, 232, 205]. Size, radial profile, and isophotal shape or ellipticity vary with stellar mass and luminosity [e.g., 145, 219, 233, 235, 38, 267, 268, 189, 113, 10, 125, 180, 297]. When controlled for other variables, these measures of galaxy morphology may show variations with cosmic environment [78, 240, 60, 114, 135], redshift [4, 258, 58, 80, 259, 164, 269, 204, 236], color [82, 293], star formation rate or quiescence [254, 303, 290, 20, 154, 283], internal dynamics [25], the presence of active galactic nuclei [143, 39, 211], and stellar age [287]. The presence and size of bulge, disk, and bar components also vary with mass and redshift [234, 237, 173, 71], and provide information about the merger rate [e.g., 162, 281]. Galaxy morphology encodes a rich spectrum of physical processes and can augment what we learn from other galaxy properties.

While complex galaxy morphologies may be easily summarized with qualitative descriptions (e.g., "disky", "spheroidal", "irregular"), providing quantitative descriptions of this complexity represents a long-standing goal for the field of galaxy formation and has motivated ingenuitive analysis methods including measures of galaxy asymmetry, concentration, flux distribution [e.g., 2, 3, 57, 56, 163],

shapelet decompositions [136, 137], morphological principal component analyses [208], and unsupervised morphological hierarchical classifications [109]. These measures provide well-defined characterizations of the surface brightness distribution of galaxies and can be connected to their underlying physical state by, e.g., calibration through numerical simulation [115]. The complementarity between these quantitative measures and qualitative morphological descriptions of galaxies means that developing both classes of characterizations further can continue to improve our knowledge of galaxy formation physics.

Characterizing large numbers of galaxies with descriptive classifications simultaneously requires domain knowledge of galaxy morphology ("expertise"), the capability to evaluate quickly each galaxy ("efficiency"), a capacity to work on significant galaxy populations ("scalability"), some analysis of the data to identify galaxy candidates for classification ("pre-processing"), a presentation of galaxy images in a format that enables the characteristic structures to be recognized ("data model"), and an output production of reliable classifications ("accuracy"). Methods for the descriptive classification of galaxy morphology have addressed these challenges in complementary ways.

Perhaps the most important and influential framework for galaxy morphological classification to date has been the Galaxy Zoo project [161, 285, 286], which enrolls the public in the analysis of astronomical data including morphological classification. This project has addressed the expertise challenge by training users in the classification of galaxies and statistically accounting for the distribution of users' accuracies. The efficiency of users varies, but by leveraging the power of the public interest and enthusiasm, and now machine learning [17, 277], the project can use scalability to offset variability in the performance of individual users. The pre-processing and delivery of suitable images to the users has required significant

investment and programming, but has led to a robust data model for both the astronomical data and the data provided by user input. Science applications of Galaxy Zoo include quantitative morphological descriptions of ∼50,000 galaxies [238] in the CANDELS survey [95, 144], probes of the connection between star formation rate and morphology in spiral galaxies [284], and measuring galaxy merger rates [281].

Other efforts have emphasized different dimensions of the morphological classification task. Kartaltepe et al. [132] organized the visual classification of ∼10,000 galaxies in CANDELS by a team of dozens of professional astronomers. This important effort performed object detection and source extraction on the CANDELS science data, assessed their completeness, and provided detailed segmentation maps of the regions corresponding to classified objects. The use of high expertise human classifiers leads to high accuracy, but poses a challenge for scalability to larger samples. The work of Kartaltepe et al. [132] also leveraged a significant investment in the pre-processing and presentation of the data to their users with a custom interface with a high-quality data model for the results.

Leveraging human classifiers, be they highly expert teams or well-calibrated legions, to provide descriptive morphologies for forthcoming datasets will prove challenging. These challenges motivate a consideration of other approaches, and I present two salient examples in *James Webb Space Telescope* [*JWST*; 85] and the Large Synoptic Survey Telescope [*LSST*; 123, 168].

JWST enables both sensitive infrared imaging with *NIRCam* and multi-object spectroscopy with *NIRSpec* free of atmospheric attenuation. The galaxy population discovered by *JWST* will show a rich range of morphologies, star formation histories, stellar masses, and angular sizes [288], which makes identifying *NIRCam*-selected samples for spectroscopic follow-up with *NIRSpec* challenging. The

efficiency gain of parallel observations with *NIRCam* and *NIRSpec* will lead to programs where the timescale for constructing *NIRCam*-selected samples will be very short (∼2 months) to enable well-designed parallel survey geometries. For this application, the ability to generate quick morphological classifications for thousands of candidate sources will enhance the spectroscopic target selection in valuable space-based observations.

LSST presents a challenge of scale, with an estimated 30 billion astronomical sources, including billions of galaxies over ∼17,000 deg^2 [168]. The morphological classification of these galaxies will require the development of significant analysis methods that can both scale to the enormity of the *LSST* dataset and perform well enough to allow imaging data to be reprocessed in pace with the *LSST* data releases. Indeed, morphological classification methods have been identified as keystone preparatory science tasks by in the *LSST* Galaxies Science Roadmap [220, see also Robertson et al. 222.].

Recently, advances in the field of machine learning called deep learning have enjoyed success in morphological classification. Dieleman et al. [70] (D15) and Dai and Tong [63] used deep learning to classify the Galaxy Zoo Survey. Huertas-Company et al. [113] used a deep learning model derived from D15 and the classifications from K15 to classify the CANDELS survey. González et al. [94] used deep learning to perform galaxy detection and morphological classification, an approach that has also been used to characterize Dark Energy Survey galaxy morphologies [251]. Deep learning models have been further applied to infer the surface brightness profiles of galaxies [262] and measure their fluxes [34], and now to simulate entire surveys [241].

Here, I extend previous efforts by applying a semantic segmentation algorithm to both classify pixels and identify objects in astronomical images using our deep

learning framework called *Morpheus*. The software architecture of the *Morpheus* framework is described in Section 2.2, with the essential convolutional neural network and deep learning components reviewed in Appendix A.1. The *Morpheus* framework has been engineered by using TensorFlow [1] implementations of these components to perform convolutions and tensorial operations, and is not a port of existing deep learning frameworks or generated via "transfer learning" [e.g., 212] of existing frameworks pre-trained on non-astronomical data such as ImageNet [68].

I train *Morpheus* using multi-band Flexible Image Transport System [FITS; 282] images of CANDELS galaxies visually classified by Kartaltepe et al. [132] and their segmentation maps derived from standard *sextractor* analyses [22]. The training procedure is described in Section 2.3, including the loss function used to optimize the *Morpheus* framework. Since *Morpheus* provides local estimates of whether image pixels contain source flux, the *Morpheus* output can be used to perform source segmentation and deblending. I present fiducial segmentation and deblending algorithms for *Morpheus* in Section 2.4.

I then apply *Morpheus* to the Hubble Legacy Fields [118] reduction of the CANDELS and GOODS data in the GOODS South region, the v1.0 data release [95, 144] for the other four CANDELS regions, and generate FITS data files of the same pixel format as the input FITS images, each containing the pixel-by-pixel model classifications of the image data into *spheroid*, *disk*, *irregular*, *point source/compact*, and *background* classes, as described in Section 2.6. I release publicly these *Morpheus* pixel-level classification data products and detailed them in Appendix A.4. I evaluate the performance of *Morpheus* in Section 2.7, including tests that use the catalog of 3D-HST photometric sources [239, 185] to measure the completeness of *Morpheus* in recovering sources as a function of source magnitude.

I find that *Morpheus* is highly complete (>90%) for sources up to one magnitude fainter than objects used to train the model. Using the *Morpheus* results, I provide estimates of the morphological classification of 3D-HST sources as a public value-added catalog, described in Section 2.8. In Section 2.9, I discuss applications of *Morpheus* and semantic segmentation, which extend well beyond morphological classification, and connect the capabilities of *Morpheus* to other research areas in astronomical data analysis. I publicly release the *Morpheus* code, provide online tutorials for using the framework via Jupyter notebooks, and present an interactive website to visualize the *Morpheus* classifications and segmentation maps in the context of the HLF images and 3D-HST catalog. These software and data releases are described in Appendices A.2, A.3, and A.4. A summary of the contributions for this chatper are presented with some conclusions in Section 2.10. Throughout the chapter, I have used the AB magnitude system [200] and assumed a flat ΛCDM universe ($\Omega_m = 0.3$, $\Omega_\Lambda = 0.7$) with a Hubble parameter $H_0 = 70$ km/s/Mpc when necessary.

2.2 *Morpheus* Deep Learning Framework

Morpheus provides a deep learning framework for analyzing astronomical images at the pixel level. Using a semantic segmentation algorithm, *Morpheus* identifies which pixels in an image are likely to contain source flux and separates them from "background" or sky pixels. *Morpheus*, therefore, allows for the definition of corresponding segmentation regions or "segmentation maps" by finding contiguous regions of source pixels distinct from the sky. Within the same framework, *Morpheus* enables for further classification of the source pixels into additional "classes". In this chapter, I have trained *Morpheus* to classify the source pixels into morphological categories (*spheroid*, *disk*, *irregular*, *point source/compact*,

and *background*) approximating the visual classifications performed by the CANDELS collaboration in K15. These source pixel classes identified by *Morpheus* could, in principle, be trained to reproduce other properties of the galaxies, such as, e.g., photometric redshift, provided a sufficient training dataset is available. In the sections below, I describe the architecture of the *Morpheus* deep learning framework. Readers unfamiliar with the primary computational elements of deep learning architectures may refer to Appendix A.1 where more details are provided.

2.2.1 Input Data

I engineered the *Morpheus* deep learning framework to accept astronomical image data as direct input for pixel-level analysis. *Morpheus* operates on science-quality FITS images, with sufficient pipeline processing (e.g., flat fielding, background subtraction, etc.) to enable photometric analysis. *Morpheus* accepts multi-band imaging data, with a FITS file for each of the n_b bands used to train the model (see Section 2.3). The pixel format of the input FITS images (or image region) matches the format of FITS images used to perform training, reflecting the size of the convolutional layers of the neural network determined before training. *Morpheus* allows for arbitrarily large images to be analyzed by subdividing them into regions that the model processes in parallel, as described in Section 2.2.3 below.

For the example application of morphological classification presented in this chapter, I use the $F606W(V)$, $F850LP(z)$, $F125W(J)$, and $F160W(H)$ band images from Hubble Space Telescope for training, testing, and our final analysis. Our training and testing images were FITS thumbnails and segmentation maps provided by Kartaltepe et al. [132]. Once trained, *Morpheus* can be applied to arbitrarily large images via a parallelization scheme described below in Section

2.2.3. I have used the CANDELS public release data [95, 144] in additional performance tests and the Hubble Legacy Fields v2.0 data [118] for our *Morpheus* data release.

I note that the approach taken by *Morpheus* differs from deep learning models that use traditional image formats, e.g., three-color Portable Network Graphics (PNG) or Joint Photographic Experts Group (JPEG) images as input. Using PNG or JPEG files as input is convenient because deep learning models trained on existing PNG or JPEG datasets, such as ImageNet [68, 227], can be retrained via transfer learning to classify galaxies. However, the use of these inputs requires additional pre-processing beyond the science pipeline, including arbitrary decisions about how to weight the FITS images to represent the channels of the multi-color PNG or JPEG. With the goal of including *Morpheus* framework analyses as part of astronomical pipelines, I have instead used FITS images directly as input to the neural network.

2.2.2 Neural Network

Morpheus uses a neural network inspired by the U-Net architecture [224, See Section A.1.5] and is implemented using Python 3 [270] and the TensorFlow library [1]. I construct *Morpheus* from a series of "blocks" that combine multiple operations used repeatedly by the model. Each block performs a sequence of "block operations". Figure 2.1 provides an illustration of a *Morpheus* block and its block operations. Block operations are parameterized by the number Q of convolved output images, or feature maps, they produce, one for each convolutional artificial neuron in the layer. I describe this process in more detail below.

Consider input data $\mathbf{X}$, consisting of K layers of images with $N \times M$ pixels. I define a block operation on $\mathbf{X}$ as

$$\mathrm{OP}_Q(\mathbf{X}) = \mathrm{ReLU}(\mathrm{CONV}_Q(\mathrm{BN}(\mathbf{X})), \tag{2.1}$$

where ReLU is the Rectified Linear Unit activation function [ReLU; 98, 152, See also Appendix A.1.1], CONV_Q is a convolutional layer (see Appendix A.1.3) with a number Q convolutional artificial neurons (see Appendix A.1.3), and BN is the batch normalization procedure [121, and Appendix A.1.4]. Note that the values of Q appearing in OP_Q and CONV_Q are equal. For example, OP_4 would indicate that the convolutional layer within the OP_4 function has 4 convolutional artificial neurons. Unless stated otherwise, all inputs into a convolutional layer are zero-padded to preserve the width and height of the input, and all convolutional artificial neurons have kernel dimensions 3×3. Given Equation 2.1, for an input $\mathbf{X}$ with dimensions $N \times M \times K$ the output of the function $\mathrm{OP}_4(\mathbf{X})$ would have dimensions $N \times M \times 4$.

Equation 2.1 allows for a recursive definition of a function describing a series of block operations, where the input data to one block operation consist of the output from a previous block operation. This recursion can be written as

$$\mathrm{OP}_Q^P(\mathbf{X}) = \begin{cases} \mathbf{X}, & \text{if } P = 0 \\ \mathrm{ReLU}(\mathrm{CONV}_Q(\mathrm{BN}(\mathrm{OP}_Q^{P-1}(\mathbf{X}))) & \text{if } P > 0 \end{cases}. \tag{2.2}$$

Equation 2.2 introduces a new parameter P, shown with a superscript in OP_Q^P. The parameter P establishes the conditions of a base case for the recursion. Note that in Equation 2.2 the input $\mathbf{X}$ is processed directly when $P = 1$, and when $P > 1$ the input to the OP_Q^P function is the output from OP_Q^{P-1}. It can be seen from the formulation of Equations 2.1 and 2.2 that $OP_Q(\mathbf{X}) = OP_Q^1(\mathbf{X})$.

Since a block performs a number P block operations, a block can be defined

mathematically as

$$\text{BLOCK}(Q, P, \mathbf{X}) = \text{OP}_Q^P(\mathbf{X}). \tag{2.3}$$

An example block and its block operations can be seen diagrammatically in Figure 2.1. With these definitions, I can present the neural network architecture used in *Morpheus*.

Like the U-Net architecture, the *Morpheus* architecture consists of a contraction phase and an expansion phase. The contraction phase consists of three blocks with parameters $(P = 4, Q = 8)$, $(P = 4, Q = 16)$, and $(P = 4, Q = 32)$. Each block is followed by a max-pooling operation with size=(2×2) (see Section A.1.4), halving the width and height of its input. After the contraction phase there is a single intermediary block preceding the expansion phase with the parameters $(P = 1, Q = 16)$. The expansion phase consists of three blocks with the parameters $(P = 2, Q = 8)$, $(P = 2, Q = 16)$, $(P = 2, Q = 32)$. Each block is preceded by a bicubic interpolation operation that doubles the width and the height of its input. Importantly, the output from each block in the contraction phase is concatenated (see Section A.1.4) with the output from the bicubic interpolation operation in the expansion phase whose output matches its width and height (see Figure 2.2). The output from the final block in the expansion phase is passed through a single convolutional layer with 5 convolutional artificial neurons. A softmax operation (see Equation 2.4) is performed on the values in each pixel, ensuring the values sum to unity. The final output is a matrix with the same width and height as the input into the network, but where the last dimension, 5, now represents a classification distribution describing the confidence the corresponding pixel from the input belongs to one of the 5 specified morphological classes.

The blocks in *Morpheus* are organized into the U-Net structure, shown in Figure 2.2. The model proceeds clockwise, starting from "Input" on the upper

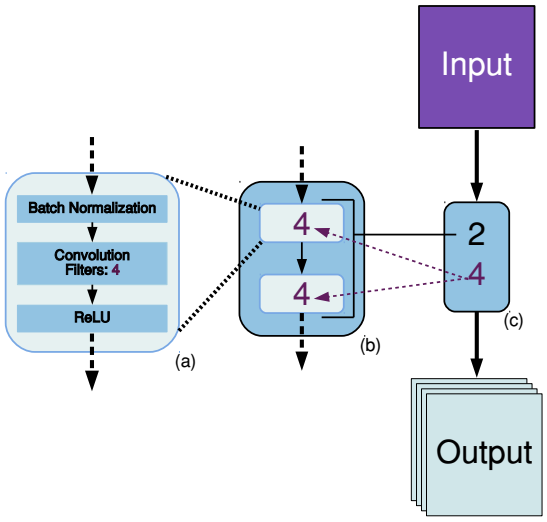

Figure 2.1: Diagram of a single block in the *Morpheus* neural network architecture (Figure 2.2). Panel (c) shows a single block from the architecture, parameterized by the number P (black) of block operations and the number Q (purple) of convolutional artificial neurons (CANs; Section A.1.3) in all of the convolutional layers within the block. Panel (b) shows an example zoom-in where there are $P = 2$ groups of $Q = 4$ block operations. Panel (a) shows a zoom-in on a block operation, which consists of batch normalization, $Q = 4$ CANs, and a ReLU operation. In the notation of Equation 2.1, this block operation would be written as $\mathrm{OP}_4(\mathbf{X})$.

left through to "Output" on the lower left. The very first step involves the insertion of the input FITS images into the model. Each FITS image is normalized to have a mean of 0 and unit variance before processing by *Morpheus*. I will refer to the number of input bands as n_b, and in the application presented here, I take $n_b = 4$ (i.e., $VzJH$). The input images each have pixel dimensions $N \times M$, and I can, therefore, consider the astronomical input data to have dimensions $N \times M \times n_b$. Only the first block operation takes the FITS images as input, and every subsequent block operation in the model takes the output from previous blocks as input.

The first convolution in the first block operation convolves the normalized $N \times M \times n_b$ astronomical data with three-dimensional kernels of size $n_k^2 \times n_b$, and each element of the kernel is a variable parameter of the model to be optimized. The convolutions operate only in the two pixel dimensions, such that n_b convolutions are performed, one for each $N \times M$ pixel image, using a different $n_k \times n_k$ kernel for each convolution. The n_b convolved images are then summed pixel by pixel to create an output feature map of size $N \times M$. The convolutional layer repeats this process Q times with different kernels, generating Q output feature maps and an output dataset of size $N \times M \times Q$. For the first block in *Morpheus* I use $Q = 8$ (see Figure 2.2). After the first convolution on the astronomical data, every subsequent convolution in the first block has both input and output data of size $N \times M \times Q$.

Each block performs a number P block operations, resulting in output data with dimensions of $N \times M \times Q$ emerging from the block. The number of feature maps Q changes with each block. For a block producing Q filters, if the data incoming into the block has size $N \times M \times Q'$ with $Q' \neq Q$, then the first convolutional layer in the first block operation will have Q kernels of size $n_k^2 \times Q'$. All

subsequent convolutional layers in the block will then ingest and produce data of size $N \times M \times Q$ by using kernels of size $n_k^2 \times Q$.

I can apply further operations on the data in between the blocks, and the character of these operations can affect the dimensions of the data. The first half of the model is a contraction phase, where each block is followed by a max-pooling operation [53, and Appendix A.1.4]. The max-pooling is applied to each feature map output by the block, taking the local maximum over small areas within each feature map (in the version of *Morpheus* presented here, a 2×2 pixel region) and reducing the size of the data input to the next block by the same factor. For this chapter, the contraction phase in the *Morpheus* framework uses three pairs of blocks plus max-pooling layers.

After the contraction phase, the model uses a series of blocks, bicubic interpolation layers, and data concatenations in an expansion phase to grow the data back to the original format. Following each block in the expansion phase, a bicubic interpolation layer expands the feature maps by the same areal factor as the max-pooling layers applied in the contraction phase (2×2 in the version of *Morpheus* presented here). The output feature maps from the interpolation layers are concatenated with the output feature maps from the contraction phase blocks where the data have the same format. Finally, the output from the last block in the expansion phase is input into a convolutional layer that produces the final output images that I call "*Morpheus* classification images", one image for each class. The pixel values in these images contain the model estimates for their classification, normalized such that the element-wise sum of the classification images equals unity. For this chapter, where I am performing galaxy morphological classification, there are five classification images (*spheroid*, *disk*, *irregular*, *point source /compact*, and *background*).

As the data progresses through the model, the number of feature maps and their shapes change owing to the max-pooling and interpolation layers. For reference, in Table 2.1, I list the dimensions of the data at each stage in the model, assuming input images in n_b bands, each with $N \times M$ pixels, and a total of n_c classification images produced by the model.

2.2.3 Parallelization for Large Images

While the *Morpheus* neural network performs semantic segmentation on pixels in FITS images with a size determined by the training images, the model can process and classify pixels in arbitrarily large images. To process large images, *Morpheus* uses a sliding window strategy by breaking the input FITS files into thumbnails of size $N \times M$ (the size of the training images) and classifying them individually. *Morpheus* proceeds through the large format image, first column by column, and then row by row, shifting the active $N \times M$ window by a unit pixel stride and then recomputing the classification for each pixel.

As the classification process continues with unit pixel shifts, each pixel is deliberately classified many times. I noticed heuristically that the output *Morpheus* classification of pixels depended on their location within the image, and that the pixel classifications were more accurate relative to our training data when they resided in the inner $n_p = (N - B) \times (M - B)$ region of the classification area, where the lesser accuracy region consisted of a border about $B \sim 5$ pixels wide on each side. Outside of the very outer B pixels in the large format image, *Morpheus* classifies each pixel n_p times. For the large FITS data images used in this chapter, this repetition corresponds to $n_p = 900$ separate classifications per pixel per output class, where each classification occurs when the pixel lies at a different location within the active window. This substantial additional information can be

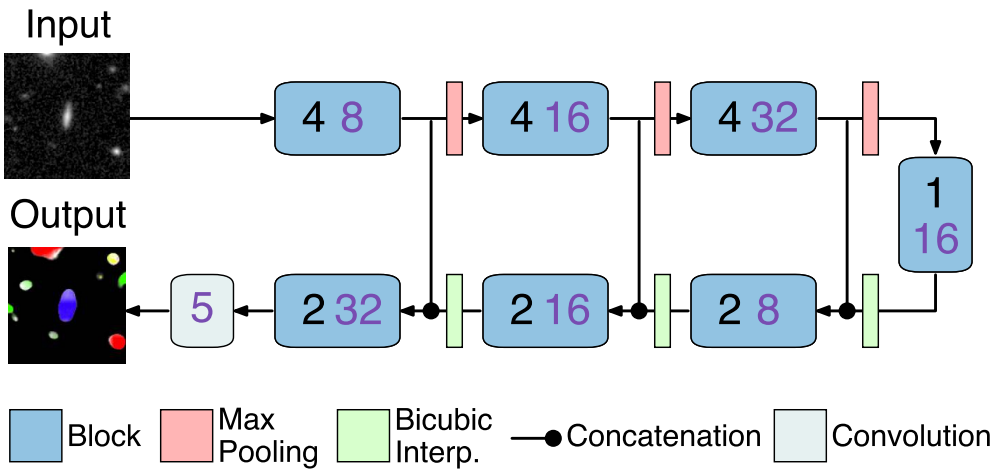

Figure 2.2: Neural network architecture of the *Morpheus* deep learning framework, following a U-Net [224] configuration. The input to the model *Morpheus* consists of astronomical FITS images in n_b bands (upper left). These images are processed through a series of computational blocks (sky blue rectangles), each of which applies P (black numbers) block operations consisting of a batch normalization and multiple convolutional layers producing Q (purple numbers) feature maps. The blocks are described in more detail in Figure 2.1. During the contraction phase of the model, max-pooling layers (salmon rectangles) are applied to the data to reduce the pixel size of the images by taking local maxima of 2×2 regions. The contraction phase is followed by an expansion phase where the output feature maps from each block are expanded by a 2×2 factor via bicubic interpolation (green rectangles) and concatenated with the output from the corresponding block in the contraction phase. The output from the last block is processed through a set of convolutional layers (light blue box with $Q = 5$) that result in a feature map for each classification in the model. These "classification images" are normalized to sum to unity pixel-by-pixel. In this chapter, the classification images are *spheroid*, *disk*, *irregular*, *point source/compact*, and *background*.

Layer	Input	Output Dimensions
Input Images	n_b Bands, $N \times M$ Pixels	[N, M, n_b]
Block 1a	Input Images	[N, M, 8]
Block 1b	Block 1a	[N, M, 8]
Block 1c	Block 1b	[N, M, 8]
Block 1d	Block 1c	[N, M, 8]
Max Pooling 1	Block 1d	[$N/2$, $M/2$, 8]
Block 2a	Max Pooling 1	[$N/2$, $M/2$, 16]
Block 2b	Block 2a	[$N/2$, $M/2$, 16]
Block 2c	Block 2b	[$N/2$, $M/2$, 16]
Block 2d	Block 2c	[$N/2$, $M/2$, 16]
Max Pooling 2	Block 2d	[$N/4$, $M/4$, 16]
Block 3a	Max Pooling 2	[$N/4$, $M/4$, 32]
Block 3b	Block 3a	[$N/4$, $M/4$, 32]
Block 3c	Block 3b	[$N/4$, $M/4$, 32]
Block 3d	Block 3c	[$N/4$, $M/4$, 32]
Max Pooling 3	Block 3d	[$N/8$, $M/8$, 32]
Block 4a	Max Pooling 3	[$N/8$, $M/8$, 16]
Interpolation 1	Block 4a	[$N/4$, $M/4$, 16]
Block 5a	Interp. 1 + Block 3d	[$N/4$, $M/4$, 8]
Block 5b	Block 5a	[$N/4$, $M/4$, 8]
Interpolation 2	Block 5b	[$N/2$, $M/2$, 8]
Block 6a	Interp. 2 + Block 2d	[$N/2$, $M/2$, 16]
Block 6b	Block 6a	[$N/2$, $M/2$, 16]
Interpolation 3	Block 6b	[N, M, 16]
Block 7a	Interp. 3 + Block 1d	[N, M, 32]
Block 7b	Block 7a	[N, M, 32]
Convolution	Block 7b	[N, M, n_c]

Table 2.1: Computational steps in the *Morpheus* deep learning framework. For each Layer (left column), I list its Input (center column), and the Output Shape of its data (right column). The model takes as its starting input a set of images in n_b bands, each with $N \times M$ pixels. The final output of the model is a set of n_c classification images, each with $N \times M$ pixels. The *Morpheus* block structures are illustrated in Figure 2.1. The "+" symbol denotes a concatenation between two layer outputs, as shown in Figure 2.2.

leveraged to improve the model, but storing the full distribution of classifications produced by this method would increase our data volume by roughly three orders of magnitude.

While *Morpheus* would enable full use of these distributions, for practical considerations, I instead record some statistical information as the computation proceeds and do not store the entire set of n_p samples. To avoid storing the full distribution, I track running estimates of the mean and variance of the distribution[1]. Once the mean for each class for each pixel is computed, I normalize the means across classes to sum to unity. I further record a statistic I call *rank voting*, which is a tally of the number of times each output class was computed by the model to be the top class for each pixel. The sum of rank votes across classes for a single pixel equals the number of times *Morpheus* processed the pixels (i.e., n_p for most pixels). After the computation, the rank votes are normalized to sum to unity across the classes for each pixel.

The strips of classified regions produce fifteen output images, containing the mean and variance estimators for the classification distribution and normalized rank votes for each class. This striped processing of the image can be performed in parallel across multiple *Morpheus* instances and then stitched back together. The weak scaling of this processing is, in principle, trivial and is limited only by the number of available GPUs and the total memory of the computer used to perform the calculation.

[1]See, e.g., http://people.ds.cam.ac.uk/fanf2/hermes/doc/antiforgery/stats.pdf for an example of running mean and variance estimators.

2.3 Model Training

The training of deep learning frameworks involves important decisions about the training data, the metrics used to optimize the network, numerical parameters of the model, and the length of training. I provide some rationale for these choices below.

2.3.1 Training Data

To train a model to perform semantic segmentation, I require a dataset that provides both information on the segmentation of regions of interest and classifications associated with those regions. For galaxy morphological classification, I use 7,629 galaxies sampled from the K15 dataset. Their 2-epoch CANDELS data provide an excellent combination of multi-band FITS thumbnails, segmentation maps in FITS format, and visually-classified morphologies in tabulated form. The K15 classifications consisted of votes by expert astronomers, between $3 - 60$ per object, who inspected images of galaxies and then selected from several morphological categories to assign to the object. The number of votes for each category for each object are provided, allowing *Morpheus* to use the distribution of votes across classifications for each object when training. I downloaded and used the publicly available K15 thumbnail FITS files for the $F606W$, $F850LP$, $F125W$, and $F160W$ bands as input into the model for training and testing. In training Morpheus to reproduce the K15 classifications, multi-band data approximates the information provided to the astronomers who performed the K15 classifications. Morpheus is trained using the same V, z, J, and H-band image thumbnails used in the K15 classification process. Other bands or different numbers of bands could be used for training as necessary, and *Morpheus* allows for reconfiguration and retraining depending on the available training images. Of the K15 dataset, I used

80% of the objects to form our training sample and 20% to form our test sample. Various statistical properties of the test and training samples are described throughout the rest of the chapter.

The primary K15 classifications *spheroid, disk, irregular*, and *point source/-compact* were used in the example *Morpheus* application presented here. I added one additional class, *background*, to represent sky pixels absent significant source flux. I classify pixels as belonging to the background category if those pixels fell outside the K15 segmentation maps. Pixels inside the segmentation maps were assigned the distribution of classifications provided by the K15 experts.

The K15 classification scheme also included an *unknown* class for objects. Since *Morpheus* works at the pixel level and could provide individual pixel classifications that were locally accurate within a source but that collectively could sum to an object whose morphology expert astronomers might classify as *unknown*, I were posed with the challenge of how to treat the K15 *unknown* class. Given our addition of the *background* class constructed from large image regions dominated by sky, one might expect overlap in the features of regions that are mostly noise and amorphous regions classified as *unknown*. Since one might also expect overlap between *unknown* and *irregular* classifications, I wanted to preserve some distinction in the object classes. I, therefore, removed the *unknown* class by removing any sources that had *unknown* as their primary classification from the training sample (213 sources). For any sources where the non-dominant K15 classifications included *unknown*, I redistributed the *unknown* votes proportionally to the other classes.

2.3.2 Data Augmentation

To increase the effective size of the training dataset, *Morpheus* uses a data augmentation method. Augmentation supplements the input training dataset by performing transformations on the training images to alter them with the intent of adding similar but not identical images with known classifications. Augmentation has been used successfully in the context of galaxy morphological classification [e.g., 70], and *Morpheus* adopts a comparable approach to previous implementations.

During training, *Morpheus* produces a series of 40×40 pixel augmented versions of the training images. The augmentation approach is illustrated in Figure 2.3. For each band in the original training image, the image is collectively rotated by a random angle $\phi \in [0, 2\pi]$, flipped horizontally with a random 50% probability, and then flipped vertically with a random 50% probability. A crop of the inner 60×60 pixels of the resulting image is produced, and then a random 40×40 pixel subset of the image is selected and passed to the model for training. This method allows us to increase the effective number of images available for training by a factor of $\sim 574,400$ and helps ameliorate over-training on the original training image set.

2.3.3 Loss Function

A standard method for training deep learning frameworks is to define a loss function that provides a statistic based on the output classifications to optimize via stochastic gradient descent with gradients computed using back-propagation [226]. Here, I describe how the *Morpheus* loss function is constructed.

The first task is to assign a distribution of input classifications on a per-pixel basis, choosing between the n_c classes available to the *Morpheus* model. For this

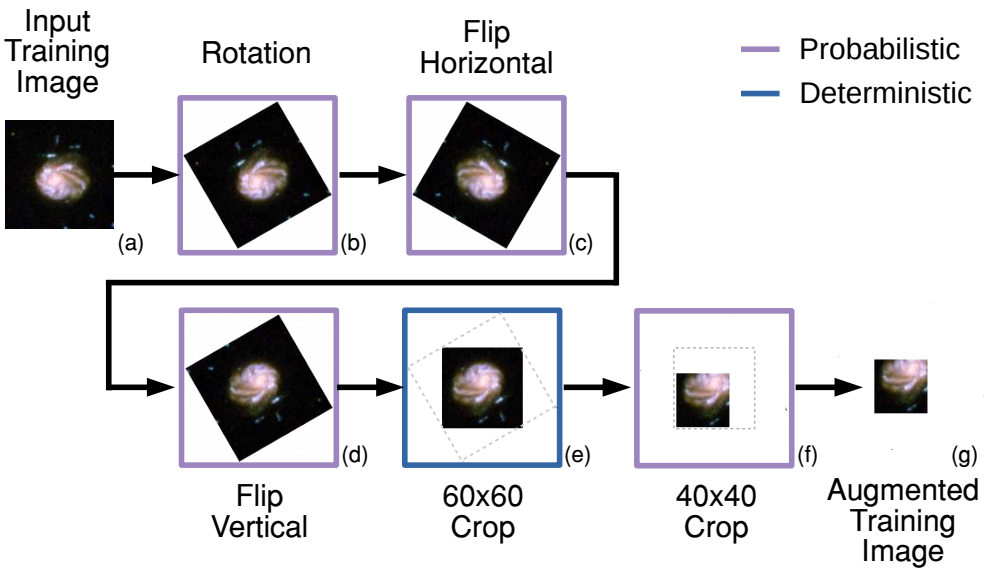

Figure 2.3: Data augmentation pipeline used during neural network training. Each training image is processed by the data augmentation pipeline before being presented to the neural network during training. The pipeline can be described in 7 stages (annotated '(a)-(g)' above). First,an image from the training set is selected (Panel a). A number of augmentation operations are then applied to the image. The image is rotated by a random angle $\theta \in [0, 2\pi]$ (Panel b), flipped horizontally with 50% probability (Panel c), and flipped vertically with a 50% probability (Panel d). The centermost 60×60 subset of the resulting image is cropped (Panel e), and then a random 40×40 subset is selected from the cropped image (Panel f). The output 40×40 rotated, flipped, and cropped image is then used for training. This procedure increases the available images for training by a factor of $\sim 574,400$. Using this process helps reduce overfitting, particularly in cases of datasets with limited training sample sizes.

chapter, I choose $n_c = 5$ (*background*, *disk*, *spheroid*, *irregular*, and *point source/-compact*), but *Morpheus* can adopt an arbitrary number of classes. I use the index k to indicate a given class, with $k \in [1, n_c]$. Consider an $N \times M$ image of an astronomical object that has been visually classified by a collection of experts, and a segmentation map defining the extent of the object in the image. Outside the segmentation map of the object, the pixels are assumed to belong to the sky and are assigned the *background* class. Inside the segmentation map, pixels are assigned the distribution of *disk*, *spheroid*, *irregular*, and *point source/compact* classifications determined by the experts for the entire object. For each pixel ij, with $i \in [1, N]$ rows and $j \in [1, M]$ columns, I then have the vector q_{ij} whose elements q_{ijk} contain the input distribution of classifications. Here, the index k runs over the number of classes n_c and $\sum_k q_{ijk} = 1$ for each pixel with indices ij. The goal of the model is to reproduce this normalized distribution q_{ij} of discrete classes for each pixel of the training images. I wish to define a *total loss function* L_{tot} that provides a single per-image statistic for the model to optimize when attempting to reproduce q_{ij}. *Morpheus* combines a weighted *cross entropy* loss function with a Dice loss [197, 181] for its optimization statistic, which I describe below.

At the end of the *Morpheus* data flow, as outlined in Figure 2.2, the raw output of the model consists of $N \times M$ vectors x_{ij} with n_c elements per-pixel estimates that represent unnormalized approximations to the input per-pixel distributions q_{ij}. The model outputs x_{ij} for each pixel are then normalized to form a probability distribution p_{ij} using the *softmax* function

$$p_{ij} = \frac{\exp(x_{ij})}{\sum_{k=1}^{n_c} \exp(x_{ijk})}, \text{ for } k \in [1, n_c]. \tag{2.4}$$

The distribution p_{ij} then represents the pixel-by-pixel classifications computed by

Morpheus for each of the $k \in [1, n_c]$ classes. For a pixel with indices ij, I can define the per-pixel cross entropy loss function as

$$L_{ij}(p_{ij}, q_{ij}) = -\sum_{k=1}^{n_c} p_{ijk} \log(q_{ijk}) \tag{2.5}$$

where p_{ij} and q_{ij} are again the two per-pixel probability distributions, with q_{ij} representing the true distribution of the input classifications for the pixel ij and p_{ij} representing the model output.

Equation 2.5 provides the per-pixel contribution to the entropy loss function. However, for many images, the majority of pixels lie outside the segmentation maps of sources identified in the training data and are therefore labeled as *background*. To overcome this imbalance and disincentivize the model from erroneously learning to classify pixels containing source flux as *background*, I apply a weighting to the per-pixel loss. I define an index $k_{ij}^{max,q}$ that indicates which class is the maximum of the input classification distribution for each pixel, written as

$$k_{ij}^{max,q} = \text{argmax } q_{ij} \tag{2.6}$$

with $1 \leq k_{ij}^{max,q} \leq n_c$. For each class k, I then define a weight w_k that is inversely proportional to the number of pixels with $k_{ij}^{max,q} = k$. I can write

$$w_k = \left[\sum_{i=1}^{N} \sum_{j=1}^{M} \max(q_{ij}) \delta_{k,k_{ij}^{max,q}} \right]^{-1}. \tag{2.7}$$

Here, $\delta_{i,j}$ is the Kronecker delta function. The vector w has size n_c and each of its elements w_k contain the inverse of the sum of $\max(q_{ij})$ for pixels with $k_{ij}^{max,q} = k$. In a given image, I ignore any classes that do not appear in the input classification distribution (i.e., any class k for which $\sum_i \sum_j q_{ijk} = 0$).

Using w, I define a weighted cross entropy loss for each pixel as

$$L_{ij}^{w} = w_{k_{ij}^{max}} L_{ij}(p_{ij}, q_{ij}). \tag{2.8}$$

A mean weighted loss function is then computed by averaging Equation 2.8 over all pixels as

$$\bar{L}^{w} = \frac{1}{N \times M} \sum_{i=1}^{N} \sum_{j=1}^{M} L_{ij}^{w}. \tag{2.9}$$

This mean weighted loss function serves as a summary statistic of the cross entropy between the output of *Morpheus* and the input classification distribution.

When segmenting images primarily comprised of *background* pixels, the classification distributions of the output pixels should be highly unbalanced, with the majority having *background*≈ 1. In this case, the mean loss function statistic defined by Equation 2.9 will be strongly influenced by a single class. A common approach to handle unbalanced segmentations is to employ a Dice loss function to supplement the entropy loss function [e.g., 181, 248]. The Dice loss function used by *Morpheus* is written as

$$L^{D}(\boldsymbol{b}, \boldsymbol{m}) = 1 - 2\frac{\sum_i \sum_j (S(\boldsymbol{b}) \circ \boldsymbol{m})_{ij}}{\sum_i \sum_j (S(\boldsymbol{b}) + \boldsymbol{m})_{ij}}. \tag{2.10}$$

Here, $S(\boldsymbol{b}) = (1 + \exp(-\boldsymbol{b}))^{-1}$ is the sigmoid function (see Equation A.3) applied pixel-wise to the *background* classification image output by the model. The image $\boldsymbol{m}$ is the input mask with values m=1 denoting *background* pixels and m=0 indicating source pixels, defined, e.g., by a segmentation map generated using *sextractor*. The $\circ$ symbol indicates a Hadamard or element-wise product of the matrices $S(\boldsymbol{b})$ and $\boldsymbol{m}$. Note that the output *background* matrix $\boldsymbol{b}$ has not yet been normalized using a softmax function, and so $b_{ij} \in [-\infty, \infty]$ and $S(b_{ij}) \in [0, 1]$. The Dice loss then ranges from $L^{D} = 0$ if $S(\boldsymbol{b}) \approx \boldsymbol{m}$ and $L^{D} \sim 1$ when $S(\boldsymbol{b})$ and $\boldsymbol{m}$

differ substantially. The addition of this loss function helps to maximize the spatial coincidence of the output *background* pixels assigned $b_{ij} \approx 1$ with the non-zero elements of the input segmentation mask $\boldsymbol{m}$.

To define the total loss function optimized during the training of *Morpheus*, the cross entropy and Dice losses are combined as a sum weighted by two parameters λ_w and λ_D. The total loss function is written as

$$L_{tot} = \lambda_w L^w + \lambda_D L^D \tag{2.11}$$

For the implementation of *Morpheus* used in this chapter, the entropy and Dice loss functions are weighted equally by setting λ_w=1 and λ_D=1.

2.3.4 Optimization Method

To optimize the model parameters, the *Adam* stochastic gradient descent method [140] was used. The *Adam* algorithm uses the first and second moments of first-order gradients computed via backpropagation to find the minimum of a stochastic function (in this case, our loss function, see Section 2.3.3, which depends on the many parameters of the neural network). The *Adam* optimizer, in turn, depends on hyper-parameters that determine how the algorithm iteratively finds a minimum. Since the loss function is stochastic, the gradients change each iteration, and *Adam* uses an exponential moving average of the gradients ($\hat{m}$) and squared gradients ($\hat{v}$) when searching for a minimum. Two dimensionless hyper-parameters (β_1 and β_2) set the decay rates of these exponential averages [see Algorithm 1 of 140]. As the parameters θ of the function being optimized are iterated between steps $t-1$ and t, they are updated according to

$$\theta_t \leftarrow \theta_{t-1} - \alpha \cdot \hat{m}_t / (\sqrt{\hat{v}_t} + \epsilon). \tag{2.12}$$

Adam Optimizer Hyper-parameters	
Hyper-parameter	Value
β_1	0.9
β_2	0.999
ϵ	10^{-8}
α	9.929×10^{-5}

Table 2.2: *Adam* optimizer [140] hyper-parameter values used during the training of the neural network used in *Morpheus*. See the text for definitions of the hyper-parameters.

Here, ϵ is a small, dimensionless safety hyper-parameter that prevents division by zero, and α is a small, dimensionless hyper-parameter that determines the magnitude of the iteration step. Table 2.2 lists the numerical values of the *Adam* optimizer hyper-parameters used by *Morpheus*. I use the default suggested values for β_1, β_2, and ϵ. After some experimentation, I adopted a more conservative step size for α than used by Kingma and Ba [140].

2.3.5 Model Evaluation

As training proceeds, the performance of the model can be quantified using various metrics and monitored to determine when training has effectively completed. The actual performance of *Morpheus* will vary depending on the classification scheme used, and here I report the performance of the model relative to the CANDELS images morphologically classified in K15. Performance metrics reported in this Section refer to pixel-level quantities, and I discuss object-level comparisons of morphological classifications relative to K15 in Section 2.5.

While the model training proceeds by optimizing the loss function defined in Section 2.3.3, I want to quantify the accuracy of the model in recovering the per-pixel classification and the overlap of contiguous regions with the same classification. First, I will need to define the index k_{ij}^{max} with maximum probability

Morpheus Training and Test Results

Metric	Training	Test
Accuracy $\mathcal{A}$		
Background	91.5%	91.4%
Disk	74.9%	75.1%
Irregular	80.6%	68.6%
Point source/compact	91.0%	83.8%
Spheroid	72.3%	71.4%
All Classes	86.8%	85.7%
Intersection-Over-Union $\mathcal{I}_{\mathcal{U}}$		
B>0.5	0.899	0.888
B>0.6	0.900	0.891
B>0.7	0.902	0.893
B>0.8	0.902	0.895
B>0.9	0.900	0.896

Table 2.3: *Morpheus* training and test results for accuracy $\mathcal{A}$, and intersection-over-union $\mathcal{I}_{\mathcal{U}}$ as a function of *background* threshold B.

to reflect either the input classification q_{ij} or the output classification p_{ij}. I define an equivalent of Equation 2.6 for p_{ij} as

$$k_{ij}^{max,p} = \text{argmax } p_{ij}. \tag{2.13}$$

I can then define a percentage accuracy

$$\mathcal{A} = \frac{100}{N \times M} \sum_{i=1}^{N} \sum_{j=1}^{M} \delta_{k_{ij}^{max,p},k_{ij}^{max,q}}. \tag{2.14}$$

The accuracy $\mathcal{A}$ then provides the percentage of pixels for which the maximum probability classes of the input and output distributions match.

In addition to accuracy, the intersection-over-union $\mathcal{I}_{\mathcal{U}}$ of pixels with *background* probabilities above some threshold is computed between the input q_{ij} and output p_{ij} distributions. If I define the index b to represent the *background* class, I can express the input *background* probabilities as q_b=q_{ijb} for i∈$[1, N]$ and j∈$[1, M]$,

and the equivalent for the output *background* probabilities p_b. I can refer to q_b and p_b as the input and output *background* images, and the regions of these images with values above some threshold B as $q_b(>B)$ and $p_b(>B)$, respectively. Note that the input q_b only contains values of zero or one, whereas the output p_b has continuous values between zero and one. I can then define the $\mathcal{I}_\mathcal{U}$ metric for threshold B as

$$\mathcal{I}_\mathcal{U}(B) = \frac{p_b(>B) \cap q_b(>B)}{p_b(>B) \cup q_b(>B)}. \tag{2.15}$$

Intuitively, this $\mathcal{I}_\mathcal{U}$ metric describes how well the pixels assigned by *Morpheus* as belonging to a source match up with the input source segmentation maps. A value of $\mathcal{I}_\mathcal{U} = 1$ indicates a perfect match between source pixels identified by *Morpheus* and the input segmentation maps, while a value of $\mathcal{I}_\mathcal{U} = 0$ would indicate no pixels in common between the two sets.

As training proceeds, the accuracy $\mathcal{A}$ and intersection-over-union $\mathcal{I}_\mathcal{U}$ are monitored until they plateau with small variations. For the K15 training data, the model plateaued after about 400 epochs. The training then continues for another 100 epochs to find a local maximum in $\mathcal{A}$ and $\mathcal{I}_\mathcal{U}$, and the model parameters at this local maximum adopted for testing. Table 2.3 summarizes the per-pixel performance of *Morpheus* in terms of $\mathcal{A}$ for each class separately, $\mathcal{A}$ for all classes, and $\mathcal{I}_\mathcal{U}(B)$ for B=[0.5, 0.6, 0.7, 0.8, 0.9]. I also report the performance of the training and testing samples separately. The pixel-level classifications are $70 - 90\%$ accurate depending on the class, and the intersection-over-union is $\mathcal{I}_\mathcal{U}{\sim}0.9$ for all thresholds $B{\geq}0.5$. The model shows some evidence for overfitting as accuracy declines slightly from the training to test sets for most classes.

2.4 Segmentation and Deblending

To evaluate the completeness of *Morpheus* in object detection and to compute an object-level classification, segmentation maps must be constructed and then deblended from the *Morpheus* pixel-level output. *Morpheus* uses the *background* class from the output of the neural network described in Section 2.2.2 to create a segmentation map. The segmentation algorithm uses a watershed transform to separate *background* pixels from source pixels and then assigns contiguous source pixels a unique label. The deblending algorithm uses the flux from the input science images and the output of the segmentation algorithm to deblend under-segmented regions containing multiple sources. I summarize these procedures as Algorithms 1 and 2. Figure 2.4 illustrates the process for generating and deblending segmentation maps.

2.4.1 Segmentation

The segmentation algorithm operates on the output *background* classification image and identifies contiguous regions of low background as sources. The algorithm begins with the *background* image $b \equiv p_b$ defined in Section 2.3.5 and an initially empty mask $m = \mathbf{0}$ of the same size. For every pixel in the image, if $b_{ij}=1$ I set $m_{ij} = 1$ and if $b_{ij} = 0$ I set $m_{ij} = 2$. The *background* mask m then indicates extreme regions of b. The [242] algorithm is applied to the *background* image b to produce a Sobel edge image s. *Morpheus* then applies the watershed algorithm of Couprie and Bertrand [61], using the Sobel image s as the "input image" and the *background* mask m as the "marker set". I refer the reader to Couprie and Bertrand [61] for more details on the watershed algorithm, but in summary, the watershed algorithm collects together regions with the same marker set value within basins in the input image. The Sobel image s provide these basins

Algorithm 1: Segmentation

Input: Background probability map b, Specified marker set p (optional, same size as b)
Output: Labelled segmentation map sm
$m \leftarrow$ zero matrix same size as b
for m_{ij} in m **do**
  if $b_{ij} = 1$ **then**
    $m_{ij} \leftarrow 1$
  end
  else if $b_{ij} = 0$ *or* $p_{ij} > 0$ **then**
    $m_{ij} \leftarrow 2$
  end
end
$s \leftarrow$ SOBEL(b)
$sm \leftarrow$ WATERSHED(s, m)
$id \leftarrow 1$
for *each contiguous set of pixels* $y > 0$ *in* sm **do**
  for *pixel* y_{ij} *in* y **do**
    $y_{ij} \leftarrow id$
  end
  $id \leftarrow id + 1$
end
return sm

Where SOBEL is the Sobel algorithm [242] and WATERSHED is the watershed algorithm [61]. Optional parameter p allows for pixel locations to be specified, such as the locations of known sources, and used as generating points for the watersheding operation.

by identifying edges in the background, and the *background* mask m provides the marker locations for generating the individual sheds. The output of the watershed algorithm is then an image sm containing distinct regions generated from areas of low *background* that are bounded by edges where the *background* is changing quickly. The algorithm then visits each of the distinct regions in sm and assigns them a unique id, creating the segmentation map sm before deblending.

2.4.2 Deblending

The algorithm described in Section 2.4.1 provides a collection of segmented regions of contiguous areas, each with a unique index. Since this algorithm iden-

tifies contiguous regions of low *background*, neighboring sources with overlapping flux in the science images will be blended by the segmentation algorithm. The deblending algorithm used in *Morpheus* is *ad hoc* and is primarily designed to separate the segmented regions into distinct subregions containing a single predefined object. The locations of these objects may be externally specified, such as catalog entries from a source catalog (e.g., 3D-HST sources), or they may be internally derived from the science images themselves (e.g., local flux maxima).

The deblending algorithm I use applies another round of the watershed operation on each of the distinct regions identified by the segmentation algorithm, using the local flux distributions from the negative of a science image (e.g., $F160W$) as the basins to fill and object locations as the marker set. I assign the resulting subdivided segmentations a distinct *subid* in addition to their shared *id*, allowing us to keep track of adjacent deblended regions that share the same parent segmentation region. The *subid* of deblended sources is indicated by decimal values and the parent *id* is indicated by the whole number of the *id*. For example, if a source with $id = 8$ was actually two sources, after deblending the two deblended sources would have id values 8.1 and 8.2.

In testing *Morpheus*, I find that the deblending algorithm may shred extended sources like large disks or point source diffraction spikes. However, the Morpheus algorithm successfully deblends some small or faint sources proximate to bright sources that are missing from the 3D-HST catalog.

2.5 Object-Level Classification

While *Morpheus* uses a semantic segmentation model to enable pixel-level classification of astronomical images using a deep learning framework, some applications, like the morphological classification of galaxies, additionally require

Algorithm 2: Deblending

Input: Segmentation map sm, flux image h, minimum radius between flux peaks r, maximum number of deblended subregions nd_{max}, Specified marker set p (optional, same size as sm)

Output: Deblended segmentation map db

if *p is not specified* **then**
  $idc \leftarrow 10^{\lceil \log_{10} nd_{max} \rceil}$ ($\lceil \rceil$ indicate ceiling operation)
  $sm \leftarrow idc \times sm$
end
for *each contiguous set of source pixels $s > 0$ in sm* **do**
  $h_{local} \leftarrow$ subset of h corresponding to s
  if *p is specified* **then**
    $p_{local} \leftarrow$ subset of p corresponding to s
    if *p_{local} contains more than one id* **then**
      $s \leftarrow \textsc{Watershed}(-h_{local}, p_{local})$
    end
    else
      $s \leftarrow \textsc{Max}(p_{local})$
    end
  end
  else
    $idx \leftarrow \textsc{PeakLocalMaxima}(h_{local}, r, c)$
    if *$\textsc{Count}(idx) > 1$* **then**
      $subid \leftarrow 1$
      $m \leftarrow$ a zero matrix same size as s
      for *indices i, j in idx* **do**
        $m_{ij} \leftarrow subid$
        $subid \leftarrow subid + 1$
      end
      $s \leftarrow \textsc{Watershed}(-h_{local}, m)$
    end
  end
end
if *p is not specified* **then**
  $db \leftarrow idc^{-1} \times sm$
end
else
  $db \leftarrow sm$
end
return *db*

Where Watershed is the watershed algorithm [61]. PeakLocalMaxima(x, y, z) returns a list of tuples marking the pixel locations of at most z local maxima in x that lie at least $2y$ pixels apart, as implemented by van der Walt et al. [266]. Count returns the number of elements in a collection. Max returns the maximum element from a matrix. Optional parameter p allows for pixel locations of known sources to used for generating points in the watershed algorithm.

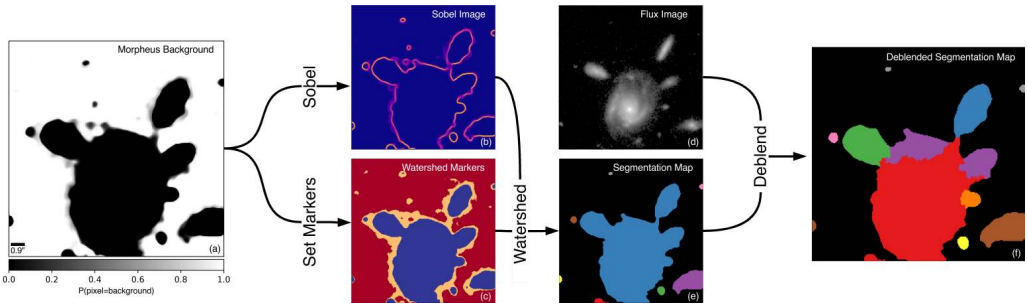

Figure 2.4: Segmentation and deblending process used by *Morpheus*, illustrating Algorithms 1 and 2. The *background* image (Panel a) output from the *Morpheus* neural network is used as input to a Sobel-filtered image (Panel b) and a discretized map marking regions of high and low *background* (Panel c). These two images are input to a watershed algorithm to identify and label distinct, connected regions of low *background* that serve as the highest-level *Morpheus* segmentation map (Panel e) This segmentation map represents the output of Algorithm 1. A flux image and a list of object locations (Panel d) are combined with the high-level segmentation map to deblend multicomponent objects using an additional watershed algorithm by using the source locations in the flux image as generating points. The end result is a deblended segmentation map (Panel f), corresponding to the output of Algorithm 2.

object-level classification. *Morpheus* aggregates pixel-level classifications into an object-level classification by using a flux-weighted average.

Figure 2.5 shows the results of the *Morpheus* pixel-level classification for an example area of the CANDELS region of GOODS South. The leftmost panel shows a three-color VzH composite of the example area for reference, though *Morpheus* operates directly on the science-quality $VzJH$ FITS images. The central panels show the output pixel classifications (i.e., q from Section 2.3.3) for the *background*, *spheroid*, *disk*, *irregular*, and *point source/compact* classes, with the intensity of each pixel indicating the normalized probability $q_{ijk} \in [0, 1]$. The segmentation map resulting from the algorithms described in Section 2.4 is also shown in as a central panel. The rightmost panel shows a color composite of the *Morpheus* pixel-level classification, with the color of each pixel indicating its the dominant class and the saturation of the pixel being proportional to the difference Δq between the dominant and second most dominant class. White pixels then indicate regions where the model did not strongly distinguish between two classes, such as in transition regions in the image between two objects with different morphological classes. The pixel intensities in the pixel-level classification image are set to 1-*background* and are not flux-weighted. The dominant classification for each object, as determined by *Morpheus*, is often clear visually. The brightest objects are well-classified and agree with the intuitive morphological classifications an astronomer might assign based on the VzH color composite image. Faint objects in the image have less morphological information available and are typically classified as *point source/compact*, in rough agreement with their classifications in the K15 training set. However, these visual comparisons are qualitative, and I now turn to quantifying the object-level classification from the pixel values.

Consider a deblended object y containing a total of n_o contiguous pixels of

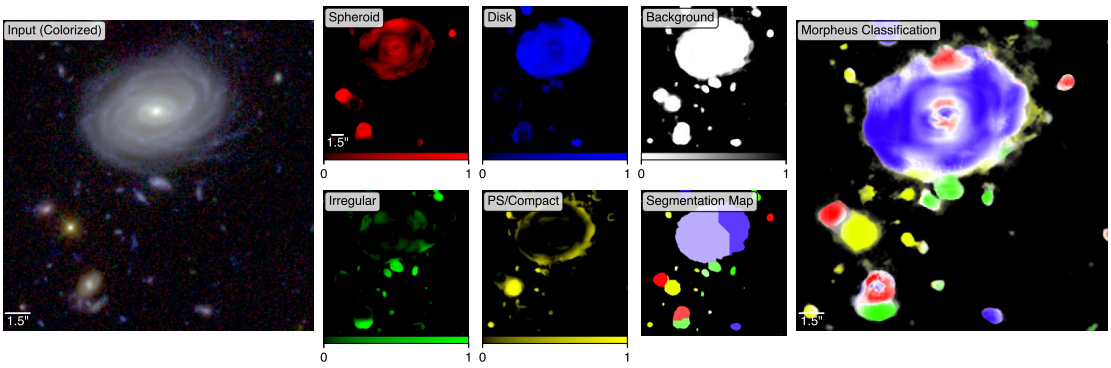

Figure 2.5: *Morpheus* morphological classification results for a region of the GOODS South field. The far left panel shows a three-color composite VzH image. The scale bar indicates 1.5". The V, z, J, and H FITS images are supplied to the *Morpheus* framework, which then returns images for the *spheroid* (red-black panel), *disk* (blue-black panel), *irregular* (green-black panel), *point source/-compact* (yellow-black panel), and *background* (white-black panel) classifications. The pixel values of these images indicate the local dominant *Morpheus* classification, normalized to sum to one across all five classifications. The panel labeled "Segmentation Map" is also generated by *Morpheus*, using the 3D-HST survey sources as generating locations for the segmentation Algorithm 1. The regions in the segmentation map are color-coded by their flux-weighted dominant class computed from the *Morpheus* classification values. The far right panel shows the *Morpheus* "classification color" image, where the pixel hues indicate the dominant morphological classification, and the intensity indicates $1-$*background*. The saturation of the *Morpheus* color image indicates the difference between the dominant classification value and the second most dominant classification, such that white regions indicate pixels where *Morpheus* returns a comparable result for multiple classes. See Section 2.6.1 for more details.

arbitrary shape within a flux image, and a single index i=$[1, n_o]$ scanning through the pixels in y. Each class k∈$[1, n_c]$ in the distribution of classification probabilities $\boldsymbol{Q}$ for the object is computed as

$$Q_k = \frac{\sum_{i=1}^{n_o} y_i q_{ik}}{\sum_{i=1}^{n_o} y_i}. \tag{2.16}$$

Here, y represents the pixel region in a science image assigned to the object, and y_i is the flux in the ith pixel of the object. The quantity q_{ik} is the kth classification probability of the ith pixel in y. Equation 2.16 represents object-level classification computed as the flux-weighted average of the pixel-level classifications in the object.

2.6 *Morpheus* Data Products

Before turning the quantifications of the object-level performance, I provide a brief overview of the derived data products produced by *Morpheus*. A more detailed description of the data products is presented in Appendix A.4, where I describe a release of pixel-level morphologies for the 5 CANDELS fields and 3D-HST value-added catalog, including object-level morphologies. The Hubble Legacy Fields [118] GOODS South v2.0 release and 3D-HST survey [185] are the primary focus of the analysis of the *Morpheus*' performance owing to their depth and completeness.

As described in Section 2.5, *Morpheus* produces a set of n_c "classification images" that correspond to the pixel-by-pixel model estimates q_{ij} for each class, normalized across classes such that $\sum_k q_{ijk} = 1$. The value of each pixel is, therefore, bounded (q_{ijk}∈$[0, 1]$). The classification images are stored in FITS format, and inherit the same ($N \times M$) pixel dimensions as the input FITS science im-

ages provided to *Morpheus*. When presenting classification images used in this chapter, I represent *background* images in negative grayscale, *spheroid* images in black-red, *disk* images in black-blue, *irregular* images in black-green, and *point source/compact* images in black-yellow color scales. Figure 2.5 shows *spheroid*, *disk*, *irregular*, *point source/compact*, and *background* images (central panels) for a region of CANDELS GOODS South.

Given the separate classification images, I can construct what I deem a "*Morpheus* morphological color image" that indicates the local dominant class for each pixel. To produce a Red-Blue-Green false color image to represent the morphological classes visually, I use the Hue-Saturation-Value (HSV) color space and convert from HSV to RGB via standard conversions. In the HSV color space, the Hue image indicates a hue on the color wheel, Saturation provides the richness of the color (from white or black to a deep color), and Value sets the brightness of a pixel (from dark to bright). On a color wheel of hues, $\mathcal{H}\in[0, 360]$ ranges from red ($\mathcal{H}$=0) to red ($\mathcal{H}$=360) through yellow ($\mathcal{H}$=120), green ($\mathcal{H}$=180), and blue ($\mathcal{H}$=240), I can assign Hue pixel values corresponding to the dominant morphological class (*spheroid* as red, *disk* as blue, *irregular* as green, and *point source/compact* as yellow). I set the Saturation of the image to be the Δq_{ijk} between the dominant class and the second most dominant class, such that cleanly classified pixels ($q_{ijk_{ij}^{max}}\approx1$, $\Delta q_{ijk}\approx1$) appear as deep red, blue, green, or yellow, and pixels where *Morpheus* produces an indeterminate classification ($\Delta q_{ijk}\approx0$) appear as white or desaturated. The Value channel is set equal to $1-$*background*, such that regions of low background containing sources are bright, and regions with high background are dark. Figure 2.5 also shows the *Morpheus* morphological color image (far right panel) for a region of CANDELS GOODS South.

2.6.1 Morphological Images for GOODS South

As part of our data products, I have produced *Morpheus* morphological images of the Hubble Legacy Fields [HLF v2.0; 118] reduction of GOODS South. These data products are used in Section 2.7 to quantify the performance of *Morpheus* relative to standard astronomical analyses, and I, therefore, introduce them here. The *Morpheus* morphological classification images for the HLF were computed as described in Section 2.2.3, feeding *Morpheus* subregions of the HLF $VzJH$ images for processing and then tracking the distribution of output pixel classifications to select the best classification for each. The $\sim 10^8$ pixels in each classification image are then stitched back together to produce contiguous *background*, *spheroid*, *disk*, *irregular*, and *point source/compact* images for the entire HLF GOODS South.

Background Image

Figure 2.6 shows the *background* image for the *Morpheus* analysis of the HLF reduction of GOODS South. The *background* classification for each pixel is shown in negative gray scale, with black corresponding to *background*=1 and white regions corresponding to *background*=0. The *background* image is used throughout Section 2.7 to quantify the performance of *Morpheus* in object detection.

spheroid Image

Figure 2.7 shows the *spheroid* image for the *Morpheus* analysis of the HLF reduction of GOODS South. The *spheroid* classification for each pixel is shown on a black-to-red colormap, with black corresponding to *spheroid*=0 and red regions corresponding to *spheroid*=1.

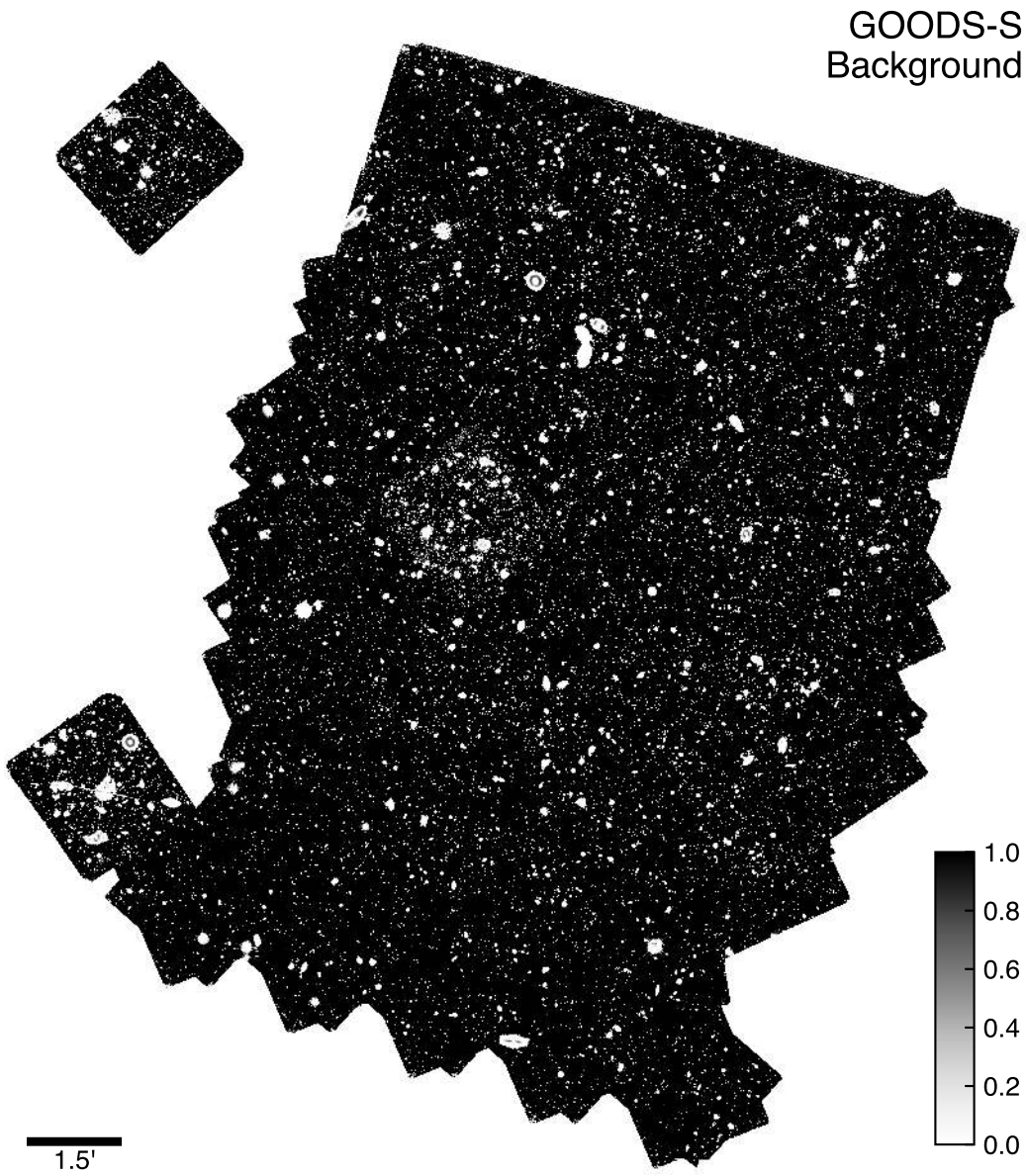

Figure 2.6: *Morpheus background* classification image for the Hubble Legacy Fields [118] reduction of the CANDELS survey data [95, 144] in GOODS South. Shown are the normalized model estimates that each of the $\sim 10^8$ pixels belongs to the *background* class. The scale bar indicates 1.5 arcmin. The color bar indicates the *background*$\in[0, 1]$, increasing from white to black. Correspondingly, the bright areas indicate regions of low background where sources were detected by *Morpheus*.

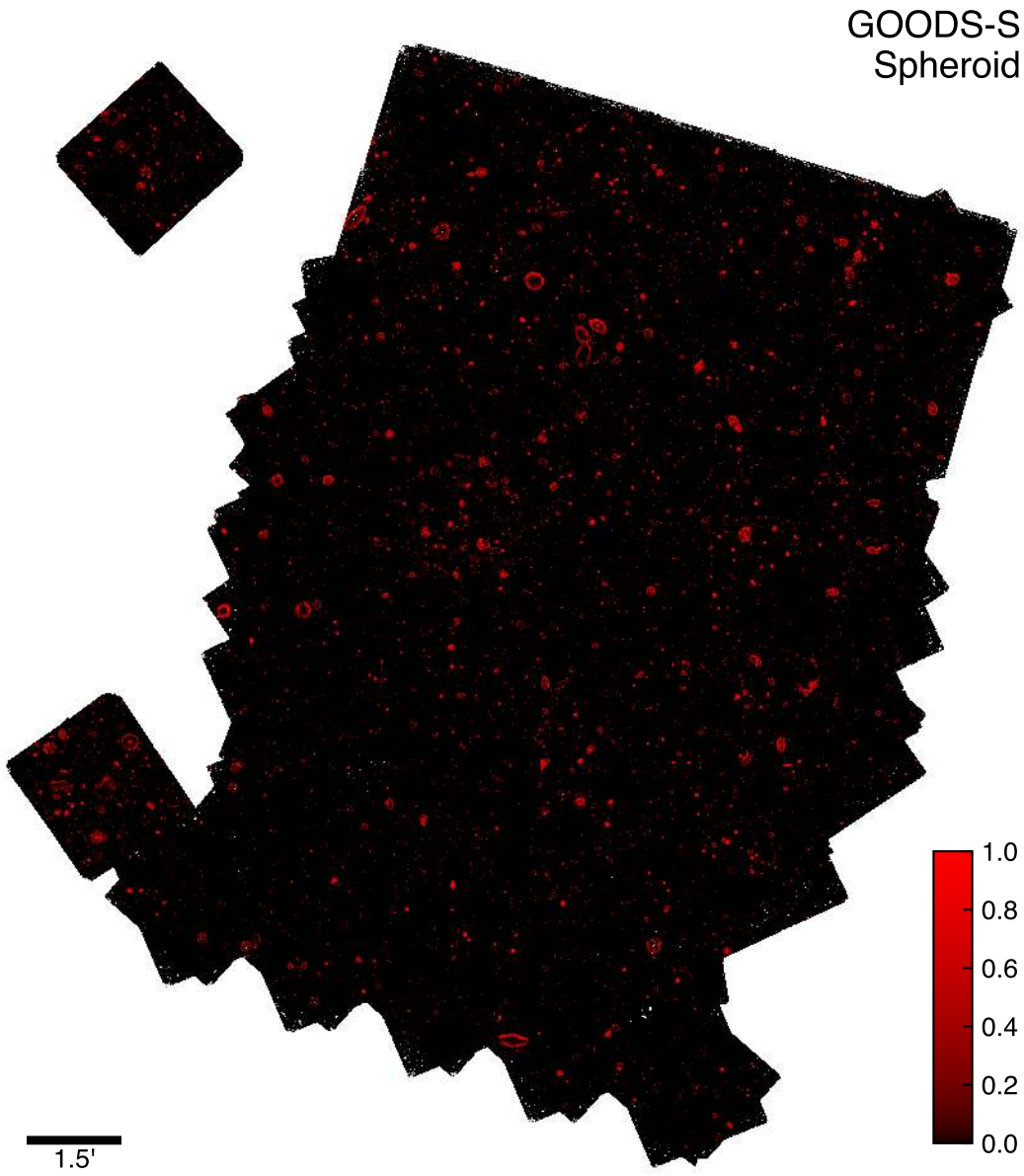

Figure 2.7: *Morpheus spheroid* classification image for the Hubble Legacy Fields [118] reduction of the CANDELS survey data [95, 144] in GOODS South. Shown are the normalized model estimates that each of the $\sim 10^8$ pixels belongs to the *spheroid* class. The scale bar indicates 1.5 arcmin. The color bar indicates the *spheroid*$\in[0, 1]$, increasing from black to red. Correspondingly, the bright red areas indicate pixels where *Morpheus* identified *spheroid* objects.

Disk Image

Figure 2.8 shows the *disk* image for the *Morpheus* analysis of the HLF reduction of GOODS South. The *disk* classification for each pixel is shown on a black-to-blue colormap, with black corresponding to *disk*=0 and blue regions corresponding to *disk*=1.

Irregular Image

Figure 2.9 shows the *disk* image for the *Morpheus* analysis of the HLF reduction of GOODS South. The *irregular* classification for each pixel is shown on a black-to-green colormap, with black corresponding to *irregular*=0 and green regions corresponding to *irregular*=1.

Point source/compact Image

Figure 2.10 shows the *point source/compact* image for the *Morpheus* analysis of the HLF reduction of GOODS South. The *point source/compact* classification for each pixel is shown on a black-to-yellow colormap, with black corresponding to *point source/compact*=0 and yellow regions corresponding to *point source/-compact*=1.

Morphological Color Image

Figure 2.10 shows the *morphological color* image for the *Morpheus* analysis of the HLF reduction of GOODS South. The false color image is constructed following Section 2.6, with the pixel intensities scaling with $1-$*background*, the pixel hues set according to the dominant class, and the saturation indicating the indeterminacy of the pixel classification. Pixels with a single dominant class appear as bright red, blue, green, or yellow for *spheroid*, *disk*, *irregular*, or *point*

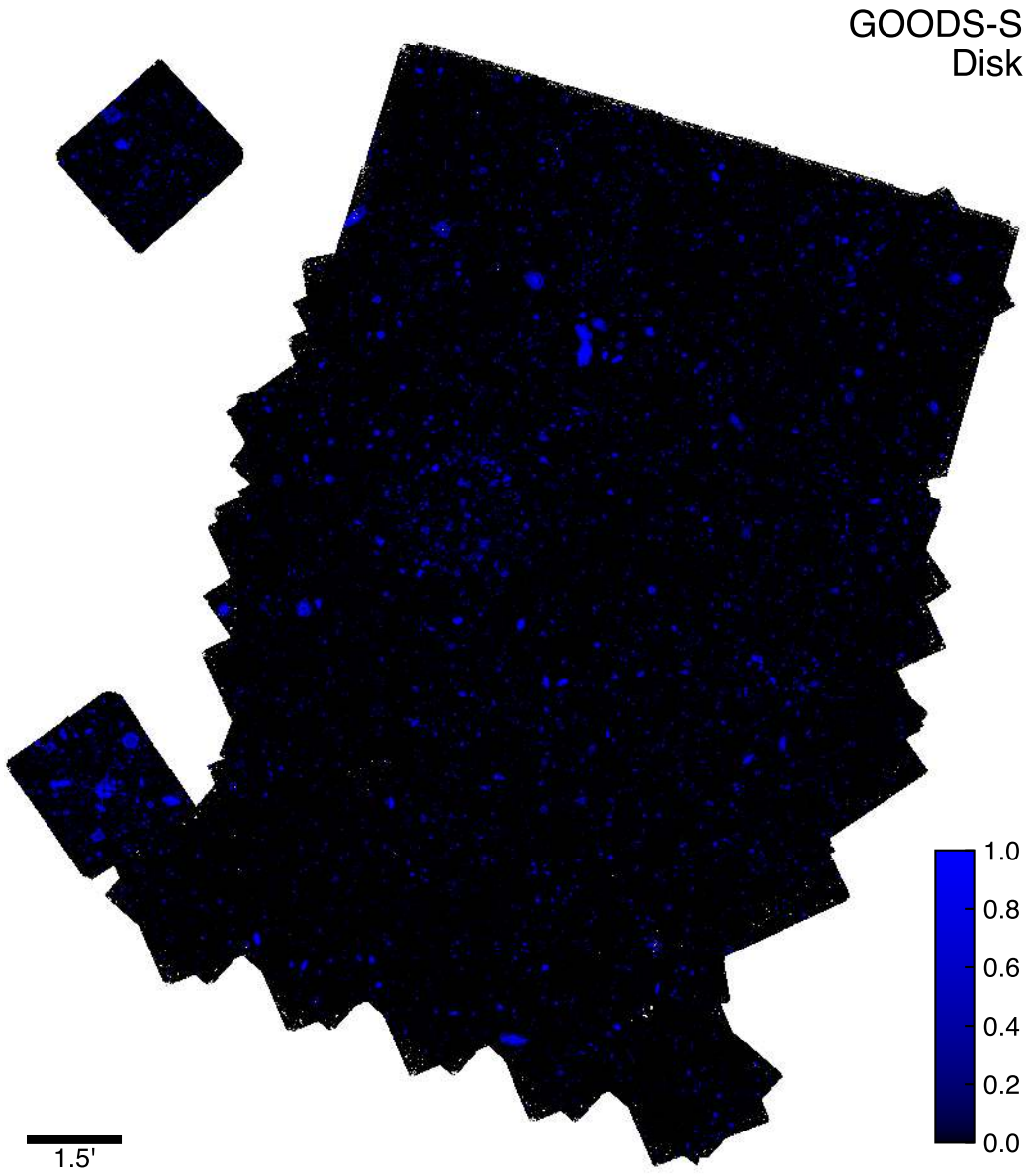

Figure 2.8: *Morpheus disk* classification image for the Hubble Legacy Fields [118] reduction of the CANDELS survey data [95, 144] in GOODS South. Shown are the normalized model estimates that each of the $\sim 10^8$ pixels belongs to the *disk* class. The scale bar indicates 1.5 arcmin. The color bar indicates the $disk \in [0, 1]$, increasing from black to blue. Correspondingly, the bright blue areas indicate pixels where *Morpheus* identified *disk* objects.

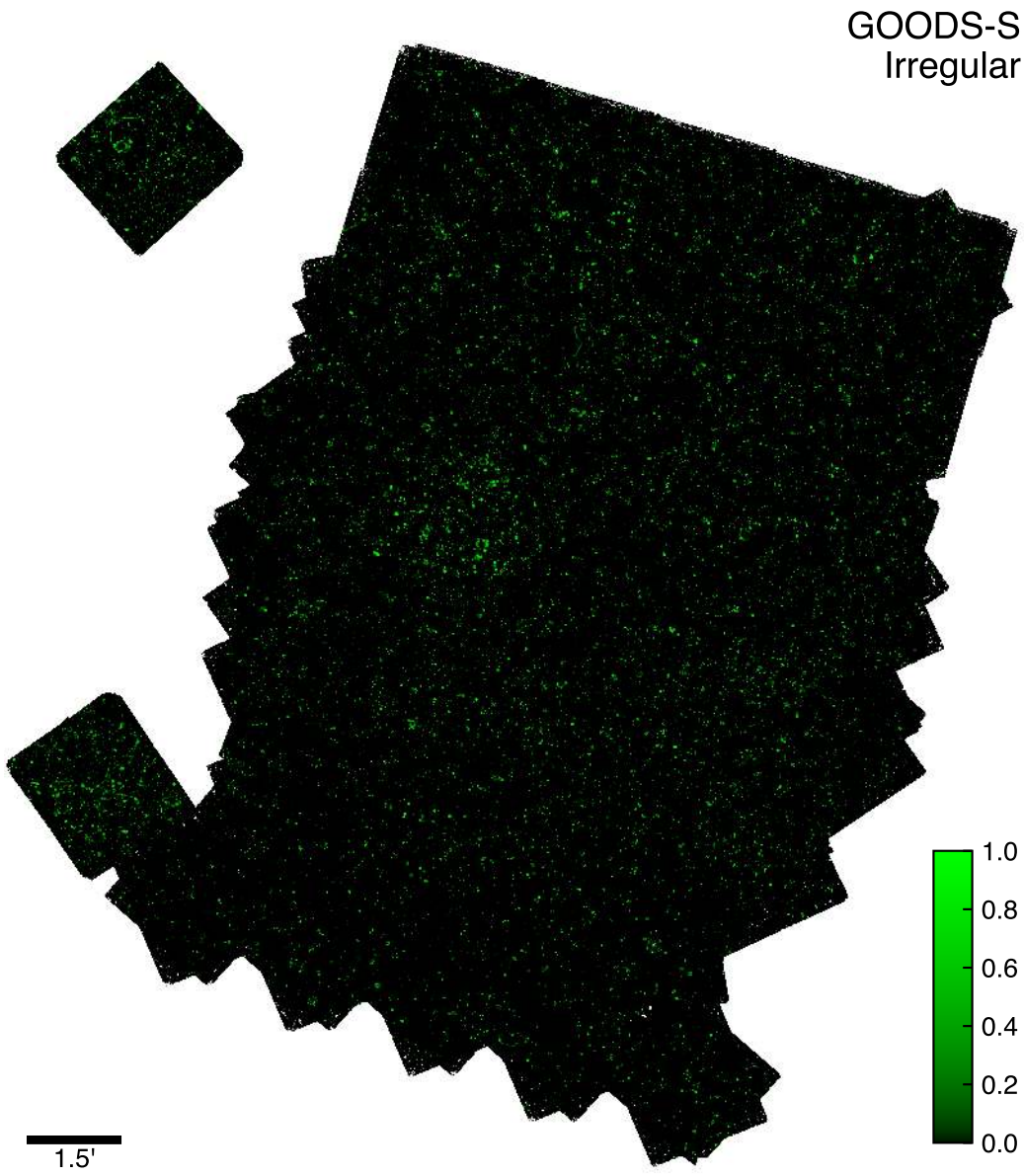

Figure 2.9: *Morpheus irregular* classification image for the Hubble Legacy Fields [118] reduction of the CANDELS survey data [95, 144] in GOODS South. Shown are the normalized model estimates that each of the $\sim 10^8$ pixels belongs to the *irregular* class. The scale bar indicates 1.5 arcmin. The color bar indicates the *irregular*$\in[0, 1]$, increasing from black to green. Correspondingly, the bright green areas indicate pixels where *Morpheus* identified *irregular* objects.

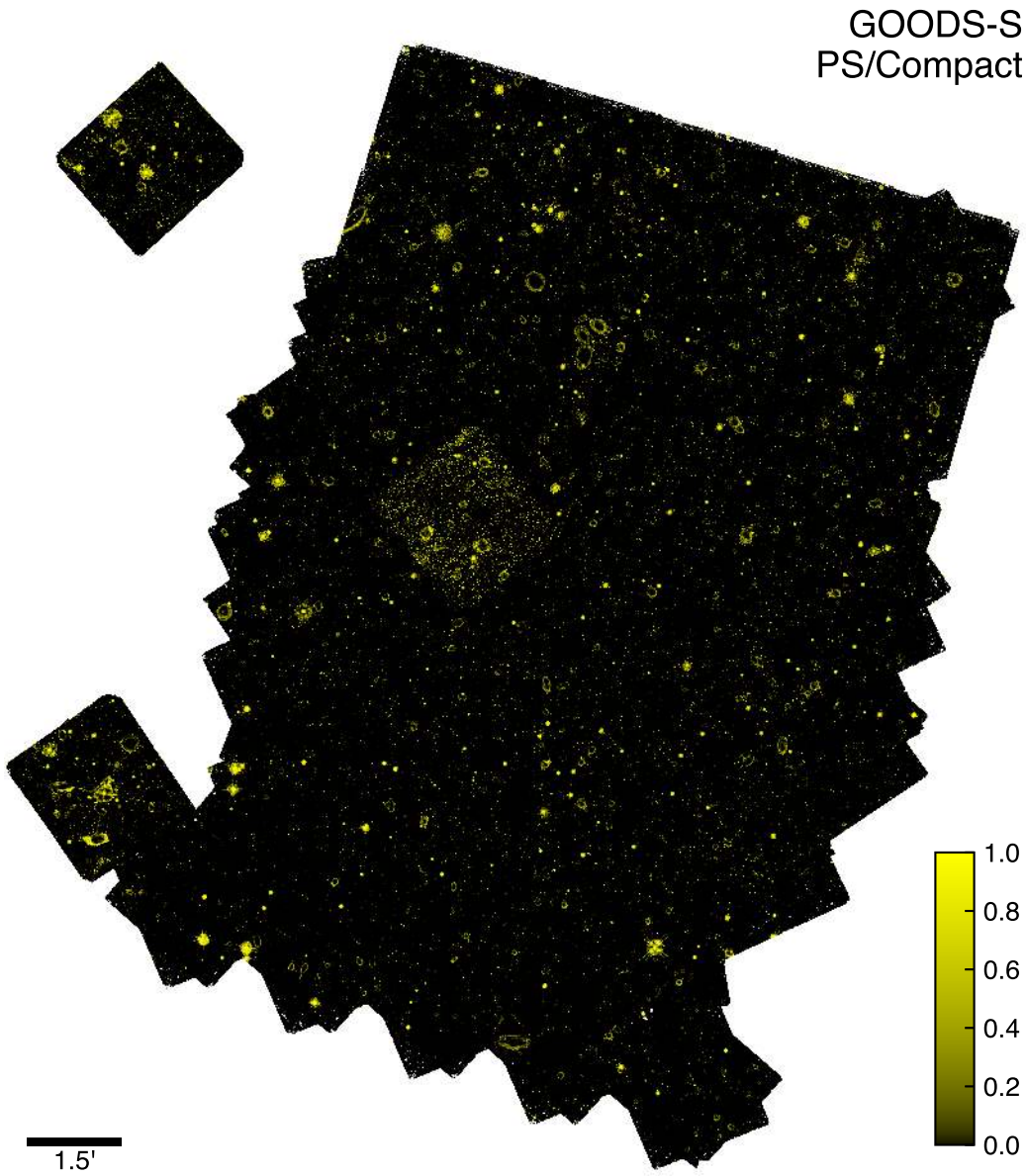

Figure 2.10: *Morpheus point source/compact* classification image for the Hubble Legacy Fields [118] reduction of the CANDELS survey data [95, 144] in GOODS South. Shown are the normalized model estimates that each of the $\sim 10^8$ pixels belongs to the *point source/compact* class. The scale bar indicates 1.5 arcmin. The color bar indicates the *point source/compact*$\in[0, 1]$, increasing from black to yellow. Correspondingly, the bright yellow areas indicate pixels where *Morpheus* identified *point source/compact* objects.

source/compact classifications, respectively. Bright white pixels indicate regions of the image where the model results were indeterminate in selecting a dominant class. Dark regions represent pixels the model classified as *background*. I note that the pixel intensities are not scaled with the flux in the image, and the per-object classifications require a local flux weighting following Equation 2.16 and the process described in Section 2.5. This flux weighting usually results in a distinctive class for each object, since the bright regions of objects often have a dominant shared pixel classification. The outer regions of objects with low flux show more substantial variation in the per-pixel classifications, but these regions often do not contribute strongly to the flux-weighted per-object classifications computed from this morphological color image.

2.7 *Morpheus* Performance

Given the data products generated by *Morpheus*, I can perform a variety of tests to quantify the performance of the model. There are basic performance metrics relevant to how the model is optimized, reflecting the relative agreement between the output of the model and the training data classifications. However, given the semantic segmentation approach of *Morpheus* and the pixel-level classification it provides, there are additional performance metrics that can be constructed to mirror widely-used performance metrics in more standard astronomical analyses including the completeness of sources detected by *Morpheus* as regions of low background. In what follows, I attempt to address both kinds of metrics and provide some ancillary quantifications to enable translations between the performance of *Morpheus* as a deep learning framework and as an astronomical analysis tool. In particular, I focus our analysis on the 3D-HST catalog and HLF reduction of the GOODS South region in the CANDELS Survey.

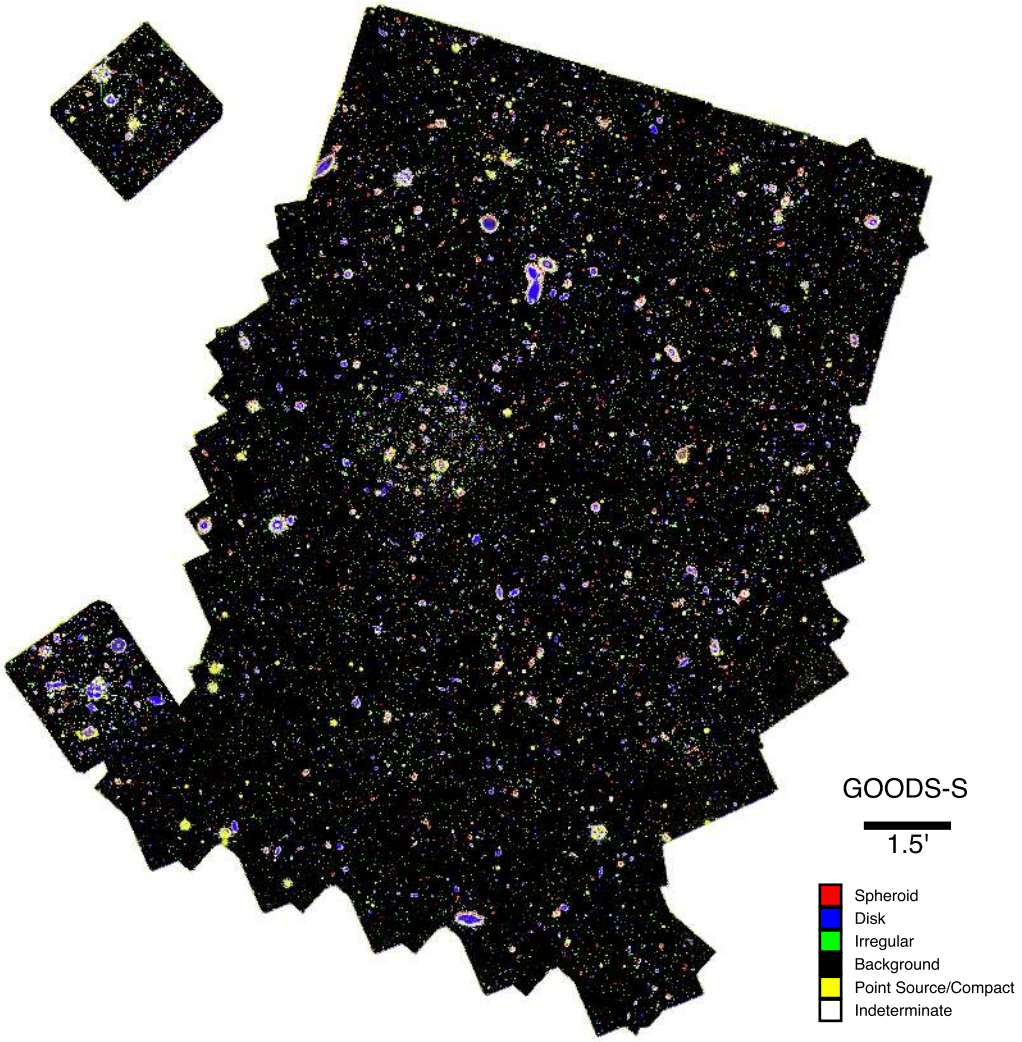

Figure 2.11: *Morpheus* morphological color image for the Hubble Legacy Fields [118] reduction of the CANDELS survey data [95, 144] in GOODS South. The image intensity is set proportional to $1-$*background* for each pixel, such that regions of high background are black and regions with low background containing source pixels identified by *Morpheus* appear bright. The hue of each source pixel indicates its dominant classification, with *spheroid* shown as red, *disk* as blue, *irregular* as green, and *point source/compact* as yellow. The color saturation of each pixel is set to the difference between the first and second most dominant class values, such that regions with indeterminate morphologies as determined as *Morpheus* appear as white and regions with strongly determined classifications appear as deep colors.

2.7.1 Object-Level Morphological Classifications

The semantic segmentation approach of *Morpheus* provides classifications for each pixel in an astronomical image. These pixel-level classifications can then be combined into object-level classifications $\boldsymbol{p}$ using the flux-weighted average described by Equation 2.16. The *Morpheus* object-level classifications can then be compared directly with a test set of visually-classified object morphologies provided by Kartaltepe et al. [132].

To understand the performance of *Morpheus* relative to the K15 visual classifications, I present some summary statistics of the training and test sets pulled from the K15 samples. During training, the loss function used by *Morpheus* is computed relative to the distribution of input K15 classifications for each object and not only their dominant classification. The goal is to retain a measure of the uncertainty in visual classifications for cases where the morphology of an object is not distinct.

Distribution of Training Sample Classifications

Galaxies in the K15 training set have been visually classified by multiple experts, providing a distribution of possible classifications for each object in the sample. Figure 2.12 presents histograms of the fraction of K15 classifiers recording votes for *spheroid*, *disk*, *irregular*, and *point source/compact* classes for each object. Only classes with more than one vote are plotted.

Classification Agreement in Training Sample

To aid these comparisons, I introduce the *agreement* statistic

$$a(\boldsymbol{p}) = 1 - \frac{H(\boldsymbol{p})}{\log(n_c)} \tag{2.17}$$

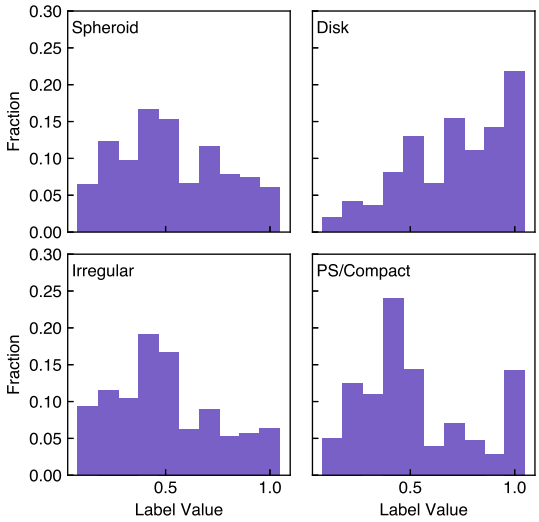

Figure 2.12: Distribution of morphological classifications in the Kartaltepe et al. [132] sample, which serve as a training sample for *Morpheus*. Shown are histograms of the fraction of sources with a non-zero probability of belonging to the *spheroid* (upper left), *disk* (upper right), *irregular* (lower left), or *point source/-compact* classes, as determined visual classification by expert astronomers. The histograms have been normalized to show the distribution of classification probabilities for each class, and consist of $\approx 7,600$ sources.

where $\boldsymbol{p}$ is the distribution of classifications and n_c is the number of classes. The quantity

$$H(\boldsymbol{p}) \equiv -\sum_{k=1}^{n_c} p_k \log p_k \tag{2.18}$$

is the self entropy. According to these definitions, $H(\boldsymbol{p}) \in [0, \log n_c]$ and $a(\boldsymbol{p}) \in [0, 1]$. The agreement $a(\boldsymbol{p}) \to 1$ when the distribution of classifications $\boldsymbol{p}$ is concentrated in a single class, and $a(\boldsymbol{p}) \to 0$ when the classifications are equally distributed. For reference, $a(\boldsymbol{p}) \approx 0.57$ for two equal classes and $a(\boldsymbol{p}) \approx 0.8$ for a 90% / 10% split between two classes for $n_c = 5$ possible classes.

Training and Test Set Statistics

The K15 classifications have substantial variation in their agreement $a(\boldsymbol{p})$. Figure 2.13 shows histograms and the cumulative distribution of $a(\boldsymbol{p})$ for objects with *spheroid*, *disk*, *irregular*, and *point source/compact* dominant classes. These distributions of $a(\boldsymbol{p})$ are roughly bimodal, consisting of a single peak near $a(\boldsymbol{p})=1$ and a broader peak near $a(\boldsymbol{p}) \approx 0.5$ with a tail to larger $a(\boldsymbol{p})$. As the cumulative distributions indicate, roughly 20%-60% of objects in the K15 sample had perfect agreement in their morphological classification, with *disk* and *point source/compact* being the most distinctive classes.

The breadth in the agreement statistic for the input K15 data indicates substantial variation in how expert astronomers would visually classify individual objects. As these data are used to train *Morpheus*, understanding exactly what *Morpheus* should reproduce requires further analysis of the K15 data. An important characterization of the input K15 data is the confusion matrix of object classifications. This matrix describes the typical classification distribution for objects of a given dominant class. Figure 2.14 presents the confusion matrix for the K15 classifications, showing the typical spread in classifications for objects

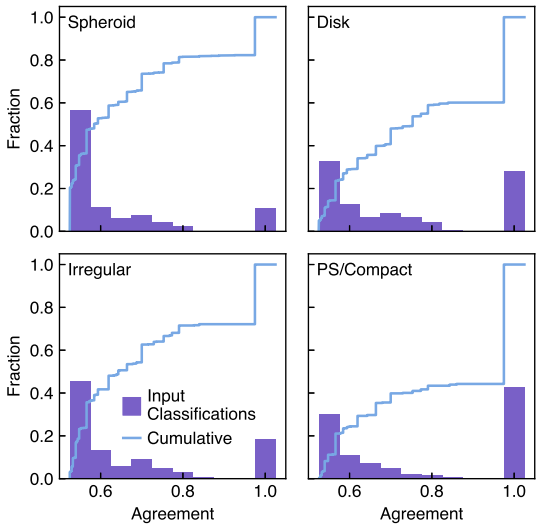

Figure 2.13: Histograms (purple) and cumulative distribution (blue lines) of agreement $a(\boldsymbol{p})$ for the Kartaltepe et al. [132, K15] visual morphological classifications, for objects with *spheroid* (upper left), *disk* (upper right), *irregular* (lower left), and *point source/compact* (lower right) as their dominant classification. Agreement $a(\boldsymbol{p})$ (see Equation 2.17 for a definition) characterizes the breadth of the distribution of morphological classes assigned by the K15 classifiers for each object, with $a(\boldsymbol{p})$=1 indicating perfect agreement of a single class and $a(\boldsymbol{p})$=0 corresponding to perfect disagreement with equal probability among classes. The distribution of agreement in the K15 training classifications is roughly bimodal, with a strong peak near-perfect agreement and a broader peak near $a(\boldsymbol{p})\approx0.5$, close to the agreement value for an even split between two classes.

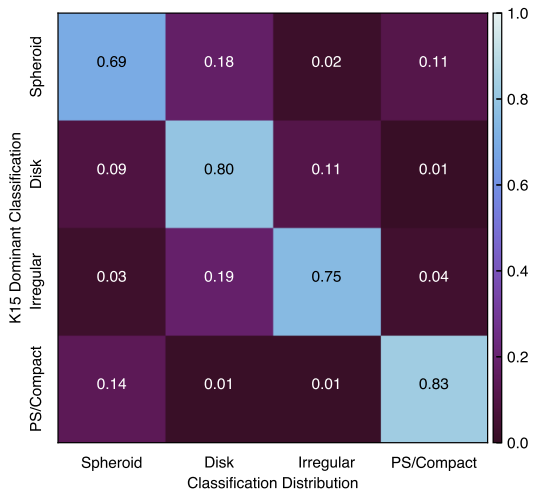

Figure 2.14: Confusion matrix for the distribution of K15 morphological classifications. Shown is the distribution of morphologies assigned by K15 visual classifiers for objects of a given dominant classification. Objects with a dominant *spheroid* class show the most variation, with frequent additional *disk* and *point source/compact* morphologies assigned. The most distinctive dominant class is *point source/compact*, which also receives a *spheroid* classification in 14% of objects. The off-diagonal components of the confusion matrix indicate imperfect agreement among the K15 classifiers, consistent with the distributions of the agreement statistic shown in Figure 2.13.

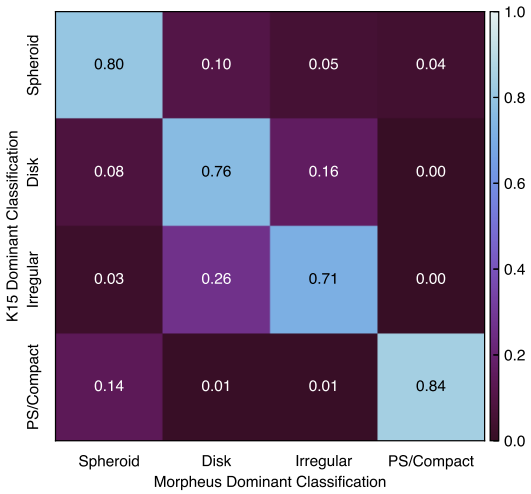

Figure 2.15: Confusion matrix showing the spread in *Morpheus* dominant classifications for objects with a given K15 dominant classifications. The *Morpheus* framework is trained to reproduce the input K15 distributions, and this confusion matrix should, therefore, largely match Figure 2.14. The relative agreement between the two confusion matrices demonstrates that the *Morpheus* output can approximate the input K15 classification distributions.

assigned *spheroid*, *disk*, *irregular*, or *point source/compact* dominant morphologies. For reference, a confusion matrix for a distribution with perfect agreement is the identity matrix. Figure 2.14 provides some insight into the natural degeneracies present in visually-classified morphologies. Objects with a dominant *disk* classification are partially classified as *spheroid* (10%) and *irregular* (11%). The *irregular* objects frequently receive an alternative *disk* classification (19%). The *point source/compact* objects also are assigned *spheroid* classifications (14%). Objects with a dominant *spheroid* class have the highest variation and receive substantial *disk* (18%) and *point source/compact* (11%) classifications. This result is consistent with Figure 2.13, which shows a relatively large disagreement for objects with a dominant *spheroid* classification.

Since *Morpheus* is trained to reproduce the distribution of K15 classifications,

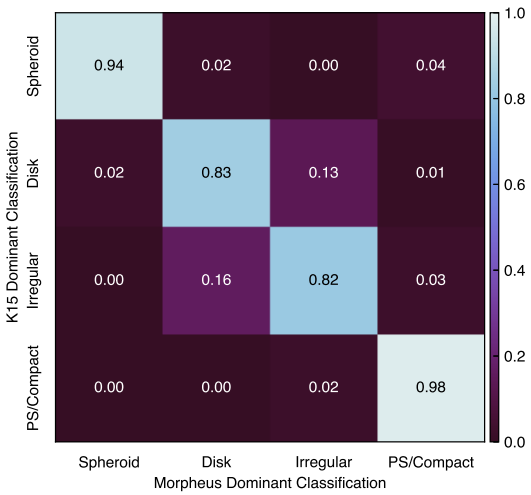

Figure 2.16: Confusion matrix quantifying the spread in *Morpheus* dominant classifications for K15 objects with a distinctive morphology. Shown are the output *Morpheus* classification distributions for K15 objects where all visual classifiers agreed on the input classification. The *Morpheus* pixel-by-pixel classifications computed for the HLF GOODS South images were aggregated into flux-weighted object-by-object classifications following Equation 2.16 using the K15 segmentation maps. The results demonstrate that *Morpheus* can reproduce the results of the dominant K15 visual classifications for objects with distinct morphologies, even as the *Morpheus* classifications were computed from per-pixel classifications using different FITS images of the same region of the sky.

the confusion matrix between the dominant *Morpheus* classifications and the K15 classification distributions should be similar to Figure 2.14. Indeed, Figure 2.15 shows the distribution of K15 classifications for objects with a given dominant *Morpheus* classification agrees well with the input K15 distributions shown in Figure 2.14. This result demonstrates *Morpheus* reproduces well the intrinsic uncertainty in the K15 classifications, as measured by the distribution of morphologies, recovered for a given K15 dominant classification.

The ability of *Morpheus* to reproduce the distribution of K15 classifications is not the only metric of interest, as it does not indicate whether the object-by-object *Morpheus* classifications agree with the K15 classifications for objects with distinctive morphologies. Figure 2.13 shows that 20-60% of objects in the K15 classifications have an agreement $a(\boldsymbol{p})$=1, meaning that all K15 visual classifiers agreed on the object morphology. The confusion matrix for these distinctive objects constructed from the K15 data is diagonal, and the confusion matrix for these objects constructed from the *Morpheus* output should also be diagonal if *Morpheus* perfectly reproduced the object-by-object K15 classifications. Further, to ensure that *Morpheus* captures the distribution of the K15 morphologies, the cumulative distribution of dominant K15 morphologies and dominant *Morpheus* morphologies as a function of color were compared using a two-sample Kolmogorov–Smirnov test. For each morphology, the p-values ($p = 0.3 - 0.99$) indicate consistency between the Morpheus and K15 distributions as a function of color. These results suggest that *Morpheus* accurately captures the K15 representation of morphology without significant color bias.

To characterize the performance of *Morpheus* for the $a(\boldsymbol{p})$=1 K15 subsample, I used the *Morpheus* output classification images computed from the HLF GOODS South images. The flux-weighted *Morpheus* morphological classifications were

computed following Equation 2.16 and using the K15 segmentation maps to ensure the same pixels were being evaluated. Figure 2.16 presents the resulting confusion matrix showing the *Morpheus* dominant classification for each object's dominant classification determined by K15. As Figure 2.16 demonstrates, *Morpheus* achieves extremely high agreement ($\geq$ 90%) with K15 for *spheroid* and *point source/-compact* objects, and good agreement ($\geq$ 80%) for *disk* and *irregular* objects with some mixing $\sim$15% between them. This performance is comparable to other object-by-object morphological classifications in the literature [e.g., 113], but is constructed directly from a flux-weighted average of pixel-by-pixel classifications by *Morpheus* using real FITS image data of differing formats and depth.

Redshift Evolution of Morphology in CANDELS Galaxies

To illustrate the scientific applications of *Morpheus*, I examine the morphological distribution of $\sim$54,000 3D-HST sources in the five CANDELS fields as a function of redshift and stellar mass (Figure 2.17). I combine together the flux-weighted *Morpheus* classifications of galaxies identified in CANDELS with the 3D-HST stellar masses and redshift, dividing the sample into coarse redshift bins. The fraction of objects N/N_{tot} with a flux-weighted classification of *spheroid* (red), *disk* (blue), or *irregular* (green) are shown as a function of stellar mass for each redshift bin, along with Poisson uncertainties on the binned values. The well-established trends of increasing fractions of irregular objects at small masses and high redshifts are correctly reproduced by *Morpheus*, as well as the growth of the disk population at low redshifts. These results can be compared with the results reported in Figure 3 of Huertas-Company et al. [114, HC16]. To ensure comparable samples between HC16 and this chapter, the *Morpheus*-classified samples in Figure 2.17 are limited to objects with H <24.5AB. Since HC16 and

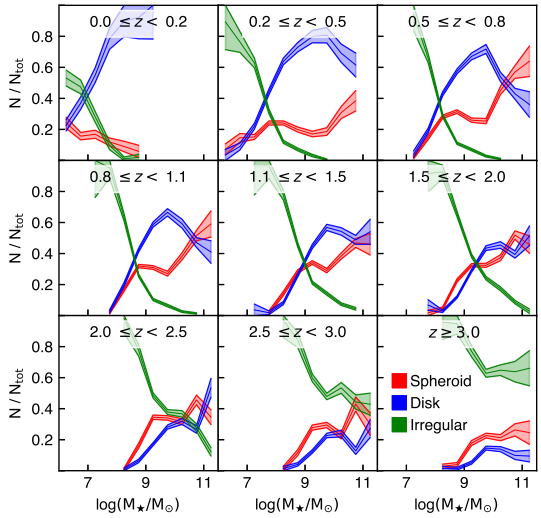

Figure 2.17: Morphology as a function of stellar mass and redshift for 54,000 sources in the five CANDELS fields. Sources included in the plot are those where $H < 24.5$AB and the *Morpheus* confidence for *spheroid*, *disk*, or *irregular* is greater than 0.7. See Section 2.7.1.

Morpheus use similar but not identical morphological classifications, I adapt the sample definitions used by HC16 to the *Morpheus* classification scheme. To be counted as a part of a morphological class, each galaxy's flux-weighted confidence value assigned by *Morpheus* must be greater than 0.7. This threshold ensures each classification is mutually exclusive, but low enough to ensure a comparable sample size to HC16.

The trends in Figure 2.17 agree with those found by HC16 in two important aspects. First, at lower redshifts, disks tend to dominate spheroids, and as redshift increases, spheroids tend to dominate disks. Second, irregular sources are a larger portion of the population than spheroids and disks at lower stellar masses and more become less abundant than spheroids and disks as stellar mass increases. The agreement between *Morpheus* and the results of HC16, which were based on object-level classifications, confirms the ability of *Morpheus* to capture source-level morphologies by aggregating pixel-level classifications.

2.7.2 Simulated Detection Tests

The *Morpheus* framework enables the detection of astronomical objects by producing a *background* classification image, with source locations corresponding to regions where *background*<1. If generating points in the form of a source catalog are not supplied, the segmentation algorithm of *Morpheus* uses an even more restrictive condition that regions near sources must contain pixels with *background*=0. Given that the semantic segmentation algorithm of *Morpheus* was trained on the K15 sample that has a completeness limit, whether the regions identified by *Morpheus* to have *background*=0 correspond to an approximate flux limit should be tested. Similarly, whether noise fluctuations lead to regions assigned *background*≈0 in error should also be evaluated.

Below, I summarize detection tests for *Morpheus* using simulated images. For these tests, a simulated sky background was generated using Gaussian random noise with RMS scatter measured in 0.5″ apertures after convolving with a model HST PSF and scaled to that measured from the K15 training images. The Tiny Tim software [149] was used to produce the PSF models appropriate for each band.

Simulated False Positive Test

Provided a large enough image of the sky, random sampling of the noise could produce regions with local fluctuation some factor f above the RMS background σ and lead to a false positive detection. A classical extraction technique using aperture flux thresholds would typically identify such regions as a $SNR = f$ source. Here, I evaluate whether *Morpheus* behaves similarly.

Using the Gaussian random noise field, single-pixel fluctuations were added to the H-band only such that the local flux measured in a 0.5″ aperture after

convolving with Tiny Tim corresponded to SNR=[0.5, 1, 2, 3, 4, 5, 6, 7, 10]. The false signals were inserted at well-separated locations such that *Morpheus* evaluated them independently. The V, z, and J images were left as blank noise, and then all four images were supplied to *Morpheus*. I find that *Morpheus* assigns none of these fake signals pixels with *background*=0. However, the SNR=7 and SNR=10 regions have some *background*<1 pixels, and while in the default algorithm, *Morpheus* would not assign these regions segmentation maps, a more permissive version of the algorithm could. An alternative test was performed by replacing the $SNR = 10$ noise fluctuation in the H-band image with a Tiny Tim H-band PSF, added after the convolution step with an amplitude corresponding to $SNR = 10$ measured in a $0.5''$ aperture. This test evaluates whether the shape of flux distribution influences the detection of single-band noise fluctuations. In this case, the minimum pixel values decreased to *background*≈0.05 for a single band SNR=10 fluctuation shaped like an H-band PSF, but did not lead to a detection. I conclude that *Morpheus* is robust to false positives arising from relatively large ($SNR \lesssim 7$) noise fluctuations.

False Negative Test

Given that *Morpheus* seems insensitive to false positives from noise fluctuations, it may also miss real but low SNR sources. By performing a similar test to that presented in Section 2.7.2 but with sources inserted in all bands rather than noise fluctuations inserted in a single band, the typical SNR where *Morpheus* becomes incomplete for real objects can be estimated.

Noise images were generated to have the same RMS noise as the K15 images by convolving Gaussian random variates with the Tiny Tim [149] model for the HST PSF. An array of well-separated point sources modeled by the PSF were then

inserted with a range of $SNR \in [1, 25]$ into all four input band images. The *Morpheus* model was then applied to the images, and the output *background* image analyzed to find regions with *background* below some threshold value. Figure 2.18 shows the number of pixels below various *background* threshold values assigned to objects with different SNR. Below about $SNR \sim 15$, the number of pixels identified as low *background* begins to decline rapidly. I therefore expect *Morpheus* to show incompleteness in real data for $SNR \lesssim 15$ sources. However, I emphasize that this limitation likely depends on the training sample used. Indeed, the K15 training dataset is complete to $H = 24.5$AB in images with 5σ source sensitivities of $H \approx 27$AB. If trained on deeper samples, *Morpheus* may prove more complete to fainter magnitudes. I revisit this issue in Section 2.7.4 below, but will explore training *Morpheus* on deeper training sets in future work.

2.7.3 Morphological Classification vs. Surface Brightness Profile

In this chapter, the *Morpheus* framework is trained on the K15 visual classifications to provide pixel-level morphologies for galaxies. The K15 galaxies are real astronomical objects with a range of surface brightness profiles for a given dominant morphology. Correspondingly, the typical classification that *Morpheus* would assign to idealized objects with a specified surface brightness profile is difficult to anticipate without computing it directly. Understanding how *Morpheus* would classify idealized galaxy models can provide some intuition about how the deep learning framework operates and what image features are related to output *Morpheus* classifications.

Figure 2.19 shows the output *Morpheus* classification distribution for simulated objects with circular Sersic [232] surface brightness profiles, for objects with

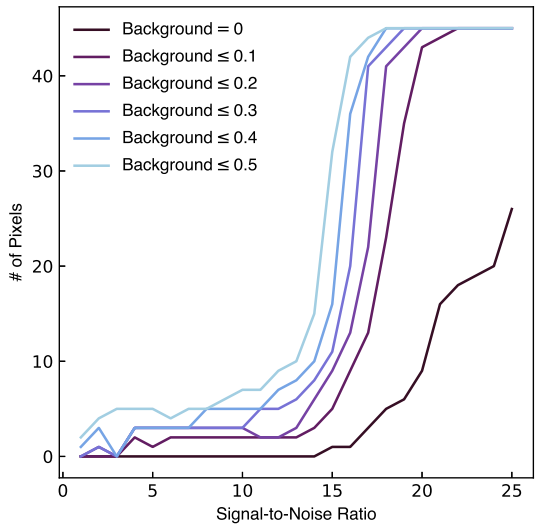

Figure 2.18: False negative test for the *Morpheus* source detection scheme. Simulated sources with different signal-to-noise ratios ($SNRs$) were inserted into a noise image and then recovered by *Morpheus*, which assigns a low *background* value to regions it identifies as containing source flux (see Section 2.7.2). Shown are lines corresponding to the number of pixels assigned to sources of different SNR, as a function of the *background* threshold. As trained on the K15 sample, *Morpheus* becomes incomplete for objects with $SNR \lesssim 15$, and is more complete if the threshold for identifying sources is made more permissive (i.e., at a higher *background* value).

$SNR = 20$, Sersic indices $\eta \in [1, 9]$, and effective radii ranging from three to nine pixels. Synthetic FITS images for each object in each band were constructed by assuming zero color gradients and a flat f_ν spectrum, populating the image with a Sersic profile object and noise consistent with the K15 images, and then convolving the images with a Tiny Tim point spread function model appropriate for each input HST filter.

The results from *Morpheus* reflect common expectations for the typical Sersic profile of morphological classes. Objects with $\eta = 1$ were typically classified as *disk* or *spheroid*, while intermediate Sersic index objects (e.g., $\eta \approx 2-3$) were classified as *spheroid*. More compact objects, with Sersic indices $\eta \geq 4$, were dominantly classified as *point source/compact*. Also, as expected for azimuthally-symmetric surface brightness profiles, *Morpheus* did not significantly classify any objects as *irregular*. Figure 2.20 provides a complementary summary of the *Morpheus* classification of Sersic profile objects, showing a matrix indicating the dominant classification assigned for each pair of $[\eta, R_e]$ values. The *Morpheus* model classifies large objects with low η as *disk*, large objects with high η as *spheroid*, and small objects with high η as *point source/compact*.

Overall, this test indicates that for objects with circular Sersic profiles, *Morpheus* reproduces the expected morphological classifications and that asymmetries in the surface brightness are needed for *Morpheus* to return an *irregular* morphological classification.

2.7.4 Source Detection and Completeness

The semantic segmentation capability of *Morpheus* allows for the detection of astronomical objects directly from the pixel classifications. In its simplest form, this object detection corresponds to regions of the output *Morpheus* classification

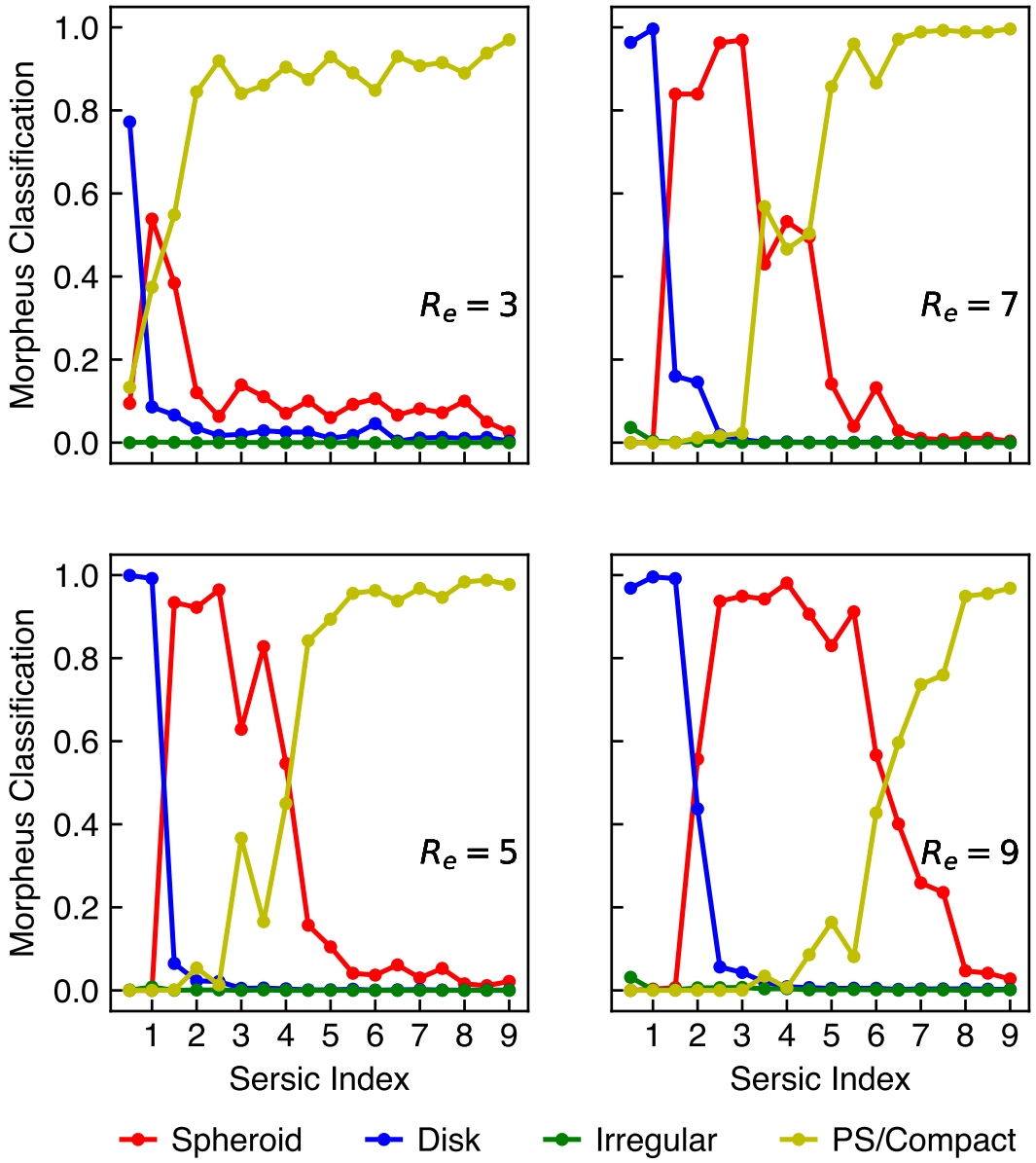

Figure 2.19: Morphological classifications as a function of simulated source surface brightness profile Sersic index. Shown are the *Morpheus* classification distributions for simulated $SNR = 20$ objects with circular Sersic [232] profiles, as a function of the Sersic index $\eta \in [1, 9]$. The experiment was repeated on objects with effective radii of three (upper left panel), five (upper right panel), seven (lower left panel), and nine (lower right panel) pixels. Objects with $\eta = 1$ were dominantly classified as *disk* or *spheroid*. Intermediate Sersic profiles ($\eta \sim 2 - 3$) were mostly classified as *spheroid*. Objects with high Sersic index ($\eta \geq 4$) were classified as *point source/compact*. These simulated objects with azimuthally symmetrical surface brightness profiles were assigned almost no *irregular* classifications by *Morpheus*.

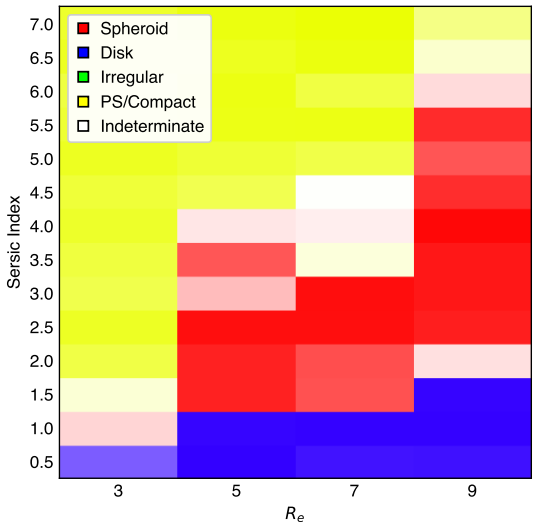

Figure 2.20: Dominant morphological classification as a function of simulated source surface brightness profile Sersic index η and effective radius R_e in pixels. Each element of the matrix is color-coded to indicate the dominant *Morpheus* classification assigned for each $[\eta, R_e]$ pair, with the saturation of the color corresponding to the difference between the dominant and second *Morpheus* classification values. Large objects with low Sersic index are classified as *disk* (blue). Large objects with high Sersic index are classified as *spheroid* (red). Small objects with high Sersic index are classified as *point source/compact* (yellow). None of the symmetrical objects in the test were classified as *irregular* (green).

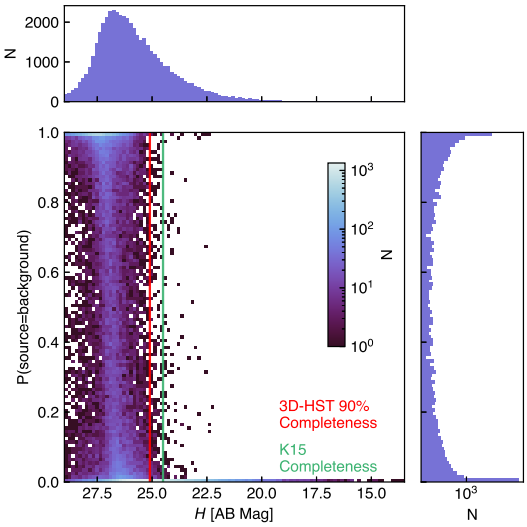

Figure 2.21: Two-dimensional histogram of *Morpheus background* values and 3D-HST source flux in GOODS South. Shown is the distribution of *background* at the location of 3D-HST sources [239, 185] in GOODS South of various H-band magnitudes, along with the marginal histograms for both quantities (side panels). For reference, the K15 completeness (green line) and 3D-HST 90% completeness (red line) flux limits are also shown. The 3D-HST sources most frequently have *background*=0, and the majority of 3D-HST sources of any flux $H < 29$ have *background*<0.5. The *background* values for objects where K15 and 3D-HST are complete is frequently zero. The *Morpheus background* values increase for many objects at flux levels $H > 26$AB.

images with low *background* class values. However, the *Morpheus* object detection capability raises several questions. The model was trained on the K15 sample, which has a reported completeness of $H = 24.5$AB, and given the pixel-by-pixel *background* classifications computed by *Morpheus*, it is unclear whether the object-level detection of sources in images would match the K15 completeness. In regions of low *background*, the transition to regions of high *background* likely depends on the individual pixel fluxes, but this transition should be characterized.

In what follows below, I provide some quantification of the *Morpheus* performance for identifying objects with different fluxes. To do this, I use results from the 3D-HST catalog of sources for the GOODS South [239, 185]. Given the output *Morpheus background* classification images computed from the HLF GOODS South FITS images in $F606W$, $F850LP$, $F125W$, and $F160W$, I can report the pixel-by-pixel *background* values and typical *background* values aggregated for objects. These measurements can be compared directly with sources in the Momcheva et al. [185] catalog to characterize how *Morpheus* detects objects and the corresponding completeness relative to 3D-HST.

In a first test, I can locate the Momcheva et al. [185] catalog objects based on their reported coordinates in the *Morpheus background* image, and then record the *background* pixel values at those locations. Figure 2.21 shows the two-dimensional histogram of *Morpheus background* value and 3D-HST source H-band AB magnitude, along with the marginal distributions of both quantities. The figure also indicates the reported K15 sample and 3D-HST 90% completeness flux levels. The results demonstrate that for the majority of 3D-HST sources and for the vast majority of bright 3D-HST sources with $H < 25$, the local *Morpheus background*=0. The low *background* values computed by *Morpheus* extend to extremely faint magnitudes (e.g., $H \approx 29$), indicating that for some faint sources, *Morpheus* reports

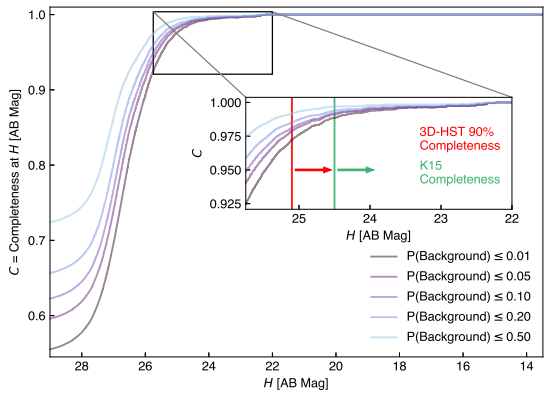

Figure 2.22: Completeness of *Morpheus* in source detection relative to 3D-HST [239, 185] in GOODS South. Shown is the fraction of 3D-HST sources in GOODS South detected by *Morpheus* brighter than some H-band source magnitude, for different *background* thresholds defining a detection (purple lines). The inset shows the *Morpheus* completeness for the brightest objects where 3D-HST (red line and arrow) and K15 (green line and arrow) are both highly complete. The completeness of *Morpheus* relative to 3D-HST is >90% where 3D-HST is highly complete. The completeness of *Morpheus* declines rapidly at faint magnitudes ($H \gtrsim 26.5$), but some objects are detected to $H \sim 29$, about 100× fainter than objects in the training set.

background=0 and that *background* is not a simple function of the local SNR of an object. For many objects with fluxes below the 3D-HST completeness, the *Morpheus background* value does increase with decreasing flux, and there is a rapid transition between detected sources at $H{\approx}26.5$ to undetected sources at $H{\leq}27.5$.

Owing to this transition in *background* with decreasing flux, the completeness of *Morpheus* relative to 3D-HST will depend on a threshold in *background* used to define a detection. Figure 2.22 shows the completeness of *Morpheus* in recovering 3D-HST objects as a function of H-band source flux for different *background* levels defining a *Morpheus* detection. The completeness flux limits for K15 and 3D-HST are indicated for reference. For magnitudes $H < 25$AB, where 3D-HST and K15 are complete, *Morpheus* is highly complete and recovers more than 99% of all 3D-HST sources. The *Morpheus* completeness declines rapidly at fluxes $H > 26.5$AB,

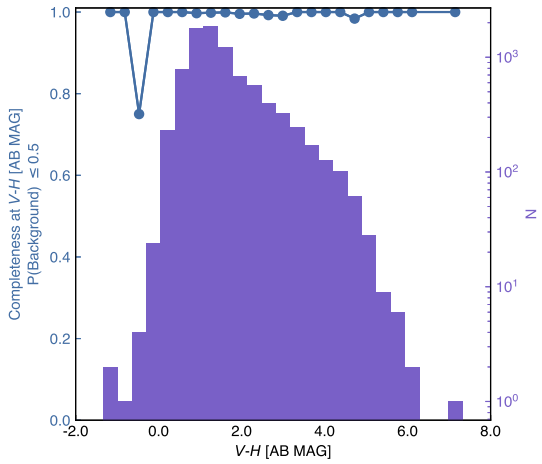

Figure 2.23: Source detection completeness as a function of color for sources with an H-band (F160W) AB magnitude of H <24.5. Sources that had a V band flux less than the V band error, had their flux replaced with three times the error value to limit unrealistically large V-H values. *Morpheus* does not show bias in the detection of objects with respect to color. There is a dip in completeness at $V - H \sim 0.2$, where the completeness is ∼75%. However, this bin only has four sources, indicating *Morpheus* only missed one source at this color.

where *Morpheus* is 90% relative to 3D-HST for *background* thresholds of $P \leq 0.5$. Perhaps remarkably, for all *background* thresholds $P \leq 0.01 - 0.5$ *Morpheus* detects some objects as faint as $H \approx 29$, about 100× fainter in flux than the training set objects.

I further examined the detection of 3D-HST sources as a function of color (V-H) to evaluate bias that may have been inherited as result of the training dataset. In our tests, I found that Morpheus is not biased with respect to color for those sources which are brighter than the K15 magnitude limit (Figure 2.23). When considering all sources within the 3D-HST catalog, Morpheus detects sources well, with a slight bias for bluer sources, but performs less well for very red ($(V - H) \geq 9$) and ($(V - H) < 0$) sources. However, it should be noted that there are very few such sources in the training set, and with a more extensive training sample, Morpheus could be more complete.

2.7.5 Morphological Classification vs. Source Magnitude

The tests of *Morpheus* on simulated Sersic objects of different effective radii and the completeness study suggest that the ability of *Morpheus* to provide informative morphological information about astronomical sources will depend on the size and signal-to-noise of the object. While these are intuitive limitations on any morphological classification method, the distribution of morphological classifications with source flux determined by *Morpheus* should be quantified.

Figure 2.24 shows the fraction of 3D-HST objects detected and classified by *Morpheus* as *spheroid*, *disk*, *irregular*, and *point source/compact* as a function of their H-band magnitude. Most of the brightest objects in the image are nearby stars, classified as *point source/compact*. At intermediate magnitudes, *Morpheus* classifies the objects as primarily a mix of *disk* ($\sim$50%) and *spheroid* ($\sim$30%), with contributions from *irregular* ($\sim 10-30\%$) and *point source/compact* ($\sim 5-15\%$). For fainter objects, below the completeness limit of the K15 training sample, *Morpheus* increasingly classifies objects as *irregular* and *point source/compact*. This behavior is in part physical, in that many low mass galaxies are irregular and distant galaxies are physically compact. In part, it reflects a natural bias in how the morphologies are defined during training. In K15, the class *point/source compact* can describe bright stars and compact unresolved sources (see Section 3.1 of K15). However, the trend also reflects how *Morpheus* becomes less effective at distinguishing morphologies in small, faint objects and returns either *point source/compact* and *irregular* for low SNR and compact sources. While training *Morpheus* on fainter objects with well-defined morphologies could enhance the ability of *Morpheus* to distinguish the features of faint sources, the results of this test make sense in the context of the completeness limit of the K15 training sample used.

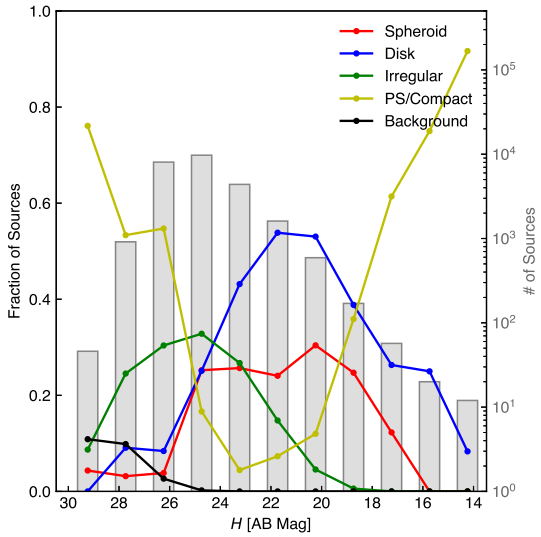

Figure 2.24: Morphological classification as a function of object flux in GOODS South. Shown are the fraction of 3D-HST objects (see left axis) with *Morpheus* dominant, flux-weighted classifications of *spheroid* (red line), *disk* (blue line), *irregular* (green line), and *point source/compact* (yellow line), each as a function of their H-band ($F160W$) AB magnitude. The brightest objects in the image are stars that are classified as *point source/compact*. The faintest objects in the image are compact faint galaxies classified as *point source/compact* or irregular. At intermediate fluxes, the objects are primarily classified as *disk* and *spheroid*. Also shown as a gray histogram (see right axis) is the number of 3D-HST objects detected and classified by *Morpheus* with source magnitude.

2.7.6 False Positives in GOODS South

The segmentation and deblending of real astronomical datasets are challenging tasks. An important test of the efficacy of the *Morpheus* segmentation and deblending algorithms is to examine false positives generated when *Morpheus* is applied to a real image. To quantify the propensity for *Morpheus* to generate false positives, the segmentation and deblending algorithms were run on the HLF GOODS South image without the specified marker set parameter p (See Algorithms 1 and 2). For the purposes of this test, a false positive is then defined as a set of pixels classified by the *Morpheus* segmentation and deblending algorithms as a source but that does not contain a source from the 3DHST and CANDELS [96] catalogs. Additionally, since the edges of the GOODS South classified image are a frayed mix of pixels, to minimize the effects of data artifacts sources less than 20 pixels from the edge of the classified area were excluded from the analysis. Further, I conservatively use the "default" *Morpheus* algorithms that identify sources with $background = 0$, i.e., when *Morpheus* indicates a source detection with high confidence. With these choices, the sample used for the false positive analysis was a total of 19,481 sources.

Among the objects classified by the segmentation and deblending algorithms, 123 sources were not present in the CANDELS or 3D-HST catalogs. Upon visual inspection of these sources, each can be categorized as an image artifact, a poor deblend, a missed source, or an actual false positive. I list the number of sources in each category in Table 2.4.

Sources in the *image artifact* category are false positives caused by image artifacts. The *poor deblend* category represents false positives caused by the *Morpheus* deblending algorithm, where single sources in the CANDELS or 3D-HST catalogs were shredded into multiple *Morpheus* sources. The *missed sources* are

False Positives in GOODS South

Category	Count	% of False Positives	% of All Sources
Image Artifact	27	21.95%	0.139%
Poor Deblend	31	25.20%	0.159%
Missed Source	47	38.21%	0.241%
Actual False Positive	18	14.64%	0.092%
Total	**123**	**100%**	**0.631%**

Table 2.4: Summary of sources identified by *Morpheus* in GOODS-S that were absent in the CANDELS or 3D-HST catalogs. Of the 19,481 sources identified by *Morpheus* in a subregion of GOODS-S, 123 sources did not have CANDELS or 3D-HST counterparts. Upon visual inspection, these objects could be categorized as either *image artifacts*, *poor deblends* where *Morpheus* had shredded sources, *missed sources* corresponding to real objects missed by CANDELS and 3D-HST, or *actual false positives* incorrectly identified as *Morpheus* as real sources. The false positive rate for the *Morpheus* algorithm is only roughly 0.09%, defined relative to the CANDELS and 3D-HST catalogs. See Section 2.7.6 for more discussion.

Morpheus sources that upon visual inspection correspond to real objects missed by the 3D-HST or CANDELS catalogs. Sources in the *actual false positive* category are false positives not associated with any image artifact or real source after visual inspection.

As Table 2.4 shows, *Morpheus* can identify real sources that other methods used to generate catalogs can miss, although the algorithms used by *Morpheus* can very rarely cause actual false positives (at roughly the 0.1% rate). Given the delicate nature of deblending, this analysis suggests that the *Morpheus* deblending algorithm could be integrated with other methods to generate more robust segmentation maps.

2.8 Value Added Catalog for 3D-HST Sources with *Morpheus* Morphologies

The *Morpheus* framework provides a system for performing the pixel-level analysis of astronomical images and has been engineered to allow for the processing of large-format scientific FITS data. As described in Section 2.6.1, *Morpheus* was applied to the Hubble Legacy Fields [HLF; 118] reduction of HST imaging in GOODS South[2] and a suite of morphological classification images produced. Using the *Morpheus background* in GOODS South, the detection efficiency of *Morpheus* relative to the Momcheva et al. [185] 3D-HST catalog was computed (see Section 2.7.4) and a high level of completeness demonstrated for objects comparably bright to the Kartaltepe et al. [132] galaxy sample used to train the model. By segmenting and deblending the HLF images, *Morpheus* can then compute flux-weighted morphologies for all the 3D-HST sources.

Table 2.5 provides the *Morpheus* morphological classifications for 50,506 sources from the 3D-HST catalog of Momcheva et al. [185]. This value-added catalog lists the 3D-HST ID, the source right ascension and declination, the $F160W$-band AB magnitude (or -1 for negative flux objects), and properties for the sources computed by *Morpheus*. The value-added properties include a flag denoting whether and how *Morpheus* detected the object, the area in pixels assigned to each source, and the *spheroid*, *disk*, *irregular*, *point source/compact*, and *background* flux-weighted classifications determined by *Morpheus*. The size of the segmentation regions assigned to each 3D-HST object following Algorithms 1 and 2 is reported for all objects. If the segmentation region assigned to an object was smaller than a circle with a 0.36" radius, or the object was undetected, instead, use a 0.36" radius

[2]Some bright pixels in the released HLF images are censored with zeros. For the purpose of computing the segmentation maps only, I replaced these censored pixels with nearby flux values.

aperture (about 109 pixels) to measure flux-weighted quantities. Only objects with joint coverage in the HLF V, z, J, and H FITS images are classified and receive an assigned pixel area. The full results for the *Morpheus* morphological classifications of 3D-HST objects are released as a machine-readable table accompanying this chapter. Appendix A.4 describes the *Morpheus* Data Release associated with this chapter, including FITS images of the classification images, the value-added catalog, and segmentation maps generated by *Morpheus* for the 3D-HST sources used to compute flux-weighted morphologies. Additionally, I release an interactive online map at https://morpheus-project.github.io/morpheus/, which provides an interface to examine the data and overlay the 3D-HST catalog on the *Morpheus* classification images, morphological color images, and segmentation maps.

2.9 Discussion

The analysis of astronomical imagery necessarily involves pixel-level information to be used to characterize sources. The semantic segmentation approach of *Morpheus* delivers pixel-level separation between sources and the background sky, and provides an automated classification of the source pixels. In this chapter, I trained *Morpheus* with the visual morphological classifications from Kartaltepe et al. [132]. I then characterized the performance of *Morpheus* in reproducing the object-level classifications of K15 after aggregating the pixel information through flux-weighted averages of pixels in *Morpheus*-derived segmentation maps, and in detecting objects via completeness measured relative to the 3D-HST catalog [185]. The potential applications of *Morpheus* extend well beyond object-level morphological classification. Below, I discuss some applications of the pixel-level information to understanding the complexities of galaxy morphology and future applications of the semantic segmentation approach of *Morpheus* in areas besides

Table 2.5: *Morpheus* + 3D-HST Value Added Catalog for GOODS South

ID	RA [deg]	Dec [deg]	$H160$ [AB mag]	Detection Flag	Area [pixels]	*sph*	*dsk*	*irr*	*psc*	*bkg*	min(*bkg*)
1	53.093012	-27.954546	19.54	1	4408	0.092	0.797	0.106	0.003	0.003	0.000
2	53.089613	-27.959742	25.49	0	–	–	–	–	–	–	–
3	53.102913	-27.959642	25.37	1	121	0.013	0.033	0.894	0.025	0.034	0.000
4	53.101709	-27.958481	21.41	1	725	0.001	0.874	0.120	0.004	0.001	0.000
5	53.102277	-27.958683	24.62	1	144	0.098	0.003	0.020	0.746	0.133	0.000
6	53.090577	-27.958515	25.07	2	109	0.000	0.831	0.034	0.000	0.134	0.001
7	53.099964	-27.958278	23.73	1	266	0.000	0.712	0.284	0.000	0.003	0.000
8	53.096144	-27.957583	21.41	1	1322	0.001	0.752	0.238	0.003	0.006	0.000
9	53.091572	-27.958367	25.90	2	109	0.000	0.044	0.083	0.081	0.792	0.431
10	53.091852	-27.958181	25.88	2	109	0.000	0.000	0.038	0.186	0.776	0.570

Column 1 provides the 3D-HST source ID. Columns 2 and 3 list the right ascension and declination in degrees. Column 4 shows the $F160W$ AB magnitude of the 3D-HST source, with -1 indicating a negative flux reported by 3D-HST. Column 5 lists the detection flag, with 0 indicating the object was not within the region of GOODS South classified by *Morpheus*, 1 indicating a detection with *background*=0 at the source location, 2 indicating a possible detection with 0<*background*<1 at the source location, and 3 indicating a non-detection with *background*=1 at the source location. Column 6 reports the area in pixels for the object determined by the *Morpheus* segmentation algorithm. For non-detections and objects with very small segmentation regions, I instead use a 0.36" radius circle (about 109 pixels) for their segmentation region. Columns 7-11 list the flux-weighted *Morpheus* morphological classifications of the objects within their assigned area. These columns are normalized such that the classifications sum to one for objects where the detection flag $!= 2$, and *sph* indicates *spheroid*, *dsk* indicates *disk*, *irr* indicates *irregular*, *psc* indicates *point source/compact* and *bkg* indicates *background*. Column 12 reports the minimum *background* value within the segmentation region. Table 2.5 is published in its entirety in the machine-readable format along with comparable tables for the other CANDELS fields. A portion is shown here for guidance regarding its form and content.

morphological classification. I also comment on some features of *Morpheus* specific to its application on astronomical images.

2.9.1 Pixel-Level Morphology

The complex morphologies of astronomical objects have been described by both visual classification schemes and quantitative morphological measures for many years. Both Hubble [112] and Vaucouleurs [271] sought to subdivide broad morphological classifications into more descriptive categories. Quantitative morphological decompositions of galaxies [e.g., 205] also characterize the relative strength of bulge and disk components in galaxies, and quantitative morphological classifications often measure the degree of object asymmetry [e.g., 2, 57, 163].

The object-level classifications computed by *Morpheus* provide a mixture of the pixel-level morphologies from the *Morpheus* classification images. The classification distributions reported in the *Morpheus* value-added catalog in GOODS South provide many examples of flux-weighted measures of morphological type. However, more information is available in the pixel-level classifications than flux-weighted summaries provide.

Figure 2.25 shows an example object for which the *Morpheus* pixel-level classifications provide direct information about its complex morphology. The figure shows a disk galaxy with a prominent central bulge. The pixel-level classifications capture both the central bulge and the extended disk, with the pixels in each structural component receiving dominant bulge or disk classifications from *Morpheus*. Note that *Morpheus* was not trained to perform this automated bulge–disk decomposition, as in the training process, all pixels in a given object are assigned the same distribution of classifications as determined by the K15 visual classifiers. As the use of pixel-level morphological classifications becomes wide-spread,

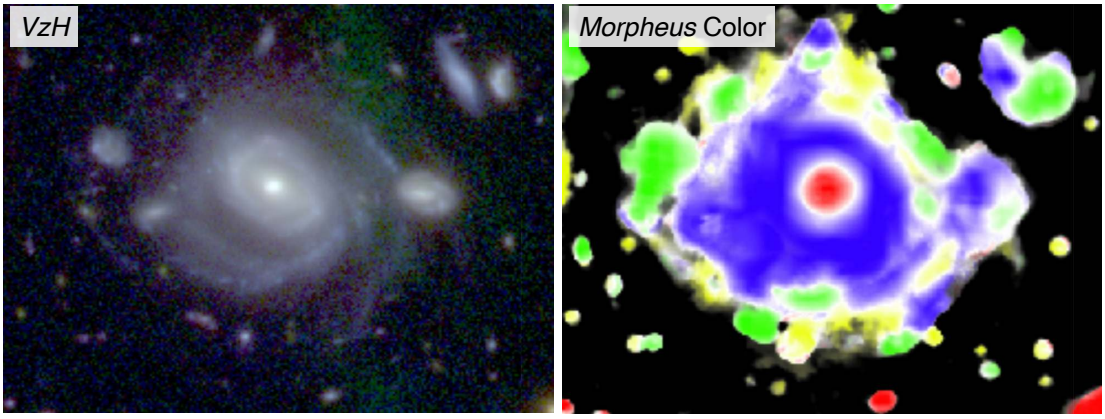

Figure 2.25: Example automated morphological decomposition by *Morpheus*. The left panel shows the VzH multi-color image of a galaxy in GOODS South from the Hubble Legacy Fields. The disk galaxy, 3D-HST ID 46386, has a prominent central bulge. The right panel shows the *Morpheus* classification color image, with pixels displaying *spheroid*, *disk*, *irregular*, or *point source/compact* dominant morphologies shown in red, blue, green, and yellow, respectively. The figure demonstrates that *Morpheus* correctly classifies the spheroid and disk structural components of the galaxy correctly, even though the training process for *Morpheus* does not involve spatially-varying morphologies for galaxy interiors. I note that there is a large-scale image artifact in $F850LP$ that appears as green in the left image, but does not strongly affect the *Morpheus* pixel-level classifications.

the development of standard datasets that include labels at the pixel-level will be needed to evaluate the efficacy of classifiers. Simulations of galaxy formation may be useful for generating such training datasets [e.g., 116]. I leave a more thorough analysis of automated morphological decompositions with *Morpheus* to future work.

2.9.2 Morphological Deblending

The ability of *Morpheus* to provide pixel-level morphological classifications has applications beyond the bulk categorization of objects. One potential additional application is the morphological deblending of overlapping objects, where the pixel-level classifications are used to augment the deblending process. Figure 2.26

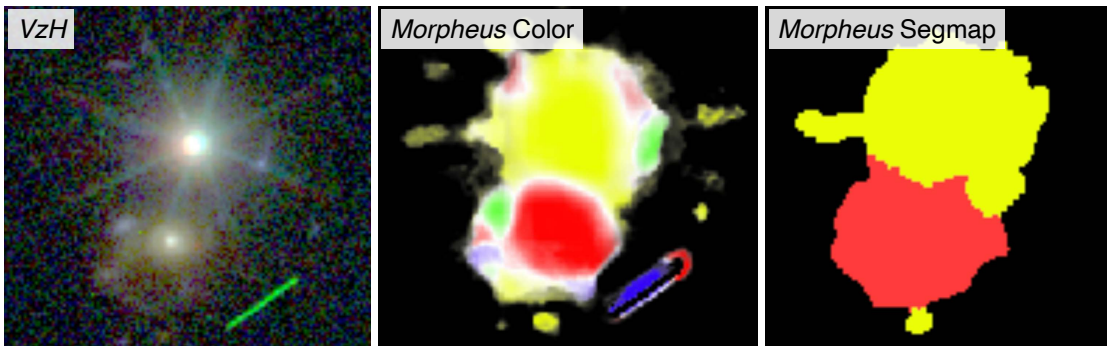

Figure 2.26: Example of morphological deblending by *Morpheus*. The leftmost panel shows the VzH image of a star-galaxy blend in GOODS South from the Hubble Legacy Fields. The star, 3D-HST ID 601, overlaps with a spheroidal galaxy 3D-HST ID 543. The center panel shows the *Morpheus* classification color image, with pixels displaying *spheroid*, *disk*, *irregular*, or *point source/compact* dominant morphologies shown in red, blue, green, and yellow, respectively. The pixel regions dominated by the star or spheroid are correctly classified by *Morpheus*. The right panel shows the resulting *Morpheus* segmentation map, illustrating that the dominant object classification in each segmentation region is also correct. The pixel-level classifications could be used to refine the segmentation to more precisely include only pixels that contained a single dominant class. The green feature in the left panel is an image artifact in $F850LP$.

shows an example of two blended objects, 3D-HST IDs 543 and 601, where the *Morpheus* pixel-level classifications could be used to perform or augment star-galaxy separation. As the figure makes clear, when *Morpheus* correctly assigns dominant classifications to pixels, there exists an interface region between regions with distinctive morphologies (in this case, *spheroid* and *point source/compact*) that could serve as an interface between segmented regions in the image. The deblending algorithm used in this chapter could include other forms of machine learning [e.g., 176, 107] information in the deblending process. If *Morpheus* was trained on information other than morphology, such as photometric redshift, those pixel-level classifications could be used in the deblending process as well. I plan to explore this idea in future applications of *Morpheus*.

2.9.3 Classifications Beyond Morphology

The semantic segmentation approach of *Morpheus* allows for complex features of astronomical objects to be learned from the data, as long as those features can be spatially localized by other means. In this chapter, I used the segmentation maps of K15 to separate source pixels from the sky, and then assigned pixels within the segmentation maps the morphological classification determined by K15 on an object-by-object basis. In principle, this approach can be extended to identify regions of pixels that contain a wide variety of features. For instance, *Morpheus* could be trained to identify image artifacts, spurious cosmic rays, or other instrumental or data effects that lead to distinctive pixel-level features in images. Of course, real features in images could also be identified, such as the pixels containing arcs in gravitational lenses, or perhaps low-surface brightness features in interacting systems and stellar halos. These pixel-level applications of *Morpheus* complement machine learning-based methods already deployed, such as those that

discover and model gravitational lenses [6, 108, 190, 191]. Pixel-level photometric redshift estimates could also be adopted by *Morpheus* and compared with existing methods based on SED fitting or other forms of machine learning [e.g., 176, 107].

2.9.4 Deep Learning and Astronomical Imagery

An important difference in the approach of *Morpheus*, where a purpose-built framework was constructed from TensorFlow primitives, compared with the adaptation and retraining of existing frameworks like Inception [e.g., 249] is the use of astronomical FITS images as training, test, and input data rather than pre-processed PNG or JPG files. The incorporation of deep learning into astronomical pipelines will benefit from the consistency of the data format. The output data of *Morpheus* are also FITS classification images, allowing pixel-by-pixel information to be easily referenced between the astronomical science images and the *Morpheus* model images. As indicated in Section 2.2.2, the *Morpheus* framework is extensible and allows for any number of astronomical filter images to be used, as opposed to a fixed red-blue-green set of layers in PNG or JPG files. The *Morpheus* framework has been engineered to allow for the classification of arbitrarily-sized astronomical images. The same approach also provides *Morpheus* a measure of the dispersion of the classifications of individual pixels, allowing the user to choose a metric for the "best" pixel-by-pixel classification. The combination of these features allows for immense flexibility in adapting the *Morpheus* framework to problems in astronomical image classification.

2.10 Summary and Conclusions

In this chapter, I presented *Morpheus*, a deep learning framework for the pixel-level analysis of astronomical images. The architecture of *Morpheus* consists of our original implementation of a U-Net [224] convolutional neural network. *Morpheus* applies the semantic segmentation technique adopted from computer vision to enable pixel-by-pixel classifications, and by separately identifying background and source pixels *Morpheus* combines object detection and classification into a single analysis. *Morpheus* represents a new approach to astronomical data analysis, with wide applicability in enabling per-pixel classification of images where suitable training datasets exist. Important results from this chapter include:

- *Morpheus* provides pixel-level classifications of astronomical FITS images. By using user-supplied segmentation maps during training, the model learns to distinguish *background* pixels from pixels containing source flux. The pixels associated with astronomical objects are then classified according to the classification scheme of the training dataset. The entire *Morpheus* source code has been publicly released, and a Python package installer for *Morpheus* provided. Further, I have a citable "frozen" version of code available through Zenodo [104].

- As a salient application, I trained *Morpheus* to provide pixel-level classifications of galaxy morphology by using the Kartaltepe et al. [132] visual morphological classifications of galaxies in the CANDELS dataset [95, 144] as our training sample.

- Applying *Morpheus* to the Hubble Legacy Fields [118] v2.0 reduction of the CANDELS data in GOODS South and the v1.0 data [95, 144] for COSMOS, EGS, GOODS North and UDS, I generated morphological classifications for

every pixel in the HLF mosaics. The resulting *Morpheus* morphological classification images have been publicly released.

- The pixel-level morphological classifications in GOODS South were then used to compute and publicly release a "value-added" catalog of morphologies for all objects in the public 3D-HST source catalog [239, 185].

- The CANDELS HLF and 3D-HST data were used to quantify the performance of *Morpheus*, both for morphological classification and its completeness in object detection. As trained, the *Morpheus* code shows high completeness at magnitudes $H \lesssim 26.5$AB. I demonstrate that *Morpheus* can detect objects in astronomical images at flux levels up to 100× fainter than the completeness limit of its training sample ($H \sim 29$AB).

- Tutorials for using the *Morpheus* deep learning framework have been created and publicly released as Jupyter notebooks.

- An interactive visualization of the *Morpheus* model results for GOODS South, including the *Morpheus* segmentation maps and pixel-level morphological classifications of 3D-HST sources, has been publicly released.

I expect that semantic segmentation will be increasingly used in astronomical applications of deep learning, and *Morpheus* serves as an example framework that leverages this technique to identify and classify objects in astronomical images. I caution that *Morpheus* may be most effective at wavelengths similar to the data on which the model was trained (i.e., the $F606W$, $F850LP$, $F125W$, and $F160W$ bands). However, Domínguez Sánchez et al. [74] have shown recent success in applying transfer learning on astronomical datasets with morphological labels. With the advent of large imaging datasets such those provided by Dark Energy Survey [64] and Hyper Suprime-Cam [7, 8], and next-generation surveys

to be conducted by Large Synoptic Survey Telescope [123, 222], Euclid [151, 217], and the Wide Field Infrared Survey Telescope [9], pixel-level analysis of massive imaging datasets with deep learning will find many applications. While the details of the *Morpheus* neural network architecture will likely change and possibly improve, I expect the approach of using semantic segmentation to provide pixel-level analyses of astronomical images with deep learning models will be broadly useful. The public release of the *Morpheus* code, tutorials, and example data products should provide a basis for future applications of deep learning for astronomical datasets.

Chapter 3

Partial-Attribution Instance Segmentation for Astronomical Source Detection and Deblending

3.1 Introduction

Astronomical images can contain tens of thousands of stars and galaxies (sources). Forthcoming telescopes including the Vera Rubin Observatory [122, 123], James Webb Space Telescope [288], and Nancy Grace Roman Space Telescope [244, 245] will push the current limits of observational astronomy and dramatically increase the number of sources to analyze. To measure accurate properties for these sources, we must *detect* sources by identifying statistically significant local maxima in an image and *deblend* sources by isolating the potentially overlapping flux distributions of each object. Consider a background-subtracted astronomical image $\mathcal{I} \in \mathbb{R}^{h \times w \times b}$ in which n sources are observed, where h is the height, w is the width, and b indicates the number of astronomical passbands. The image $\mathcal{I}$ can

be decomposed into a sum of individual object contributions as

$$\mathcal{I} = \sum_{i=1}^{N} S_i + \epsilon \tag{3.1}$$

where $S_i \in \mathbb{R}^{h \times w \times b}$ represents the flux contributed to $\mathcal{I}$ by source i, and $\epsilon \in \mathcal{N}(0, \sigma)$ is the approximate noise distribution in the image. The process of decomposing an image into the form of Equation 3.1 represents the core challenge of source deblending. In this chapter, I present a deep learning-based method to perform detection and deblending on astronomical images.

3.1.1 Related Work

Source detection and deblending are well-studied problems in astronomy, and many approaches have been developed. Below, I highlight some popular and recent methods for source detection and deblending and point the interested reader to the review by Masias et al. [174].

Detection and deblending methods can be characterized by their *detection capacity* and *deblend type*. The detection capacity represents the number of sources a method can detect within a single image. For Equation 3.1, a detection capacity of N would indicate that a method could detect all sources appearing in an image. The deblend type indicates how the flux in a single pixel may be split between overlapping (blended) sources. A *disjoint* deblender assigns all flux in a pixel to a single source exclusively. An *intersecting/discrete* deblender can assign the flux to more than one source with uniform weighting across pixels. Finally, an *intersecting/continuous* deblender can assign the flux to more than one source with variable weighting across pixels.

Astronomical analysis methods vary in their detection and deblending methods. Bertin and Arnouts [22] introduced SExtractor that uses a convolution and

Table 3.1: Detection and deblending method categorization

Name	Detection Capacity	Deblend Type
SExtractor[22]	N	Disjoint
Morpheus[104]	N	Disjoint
Mask R-CNN[40]	N	Intersecting/Discrete
`blend2mask2flux`[34]	2	Intersecting/Discrete
Modified SRGAN[214]	0	Intersecting/Continuous
SCARLET[179]	0	Intersecting/Continuous
This Chapter	N	Intersecting/Continuous

thresholding approach for detection, and an isophotal analysis using binned pixel intensity for deblending. Hausen and Robertson [104] introduced Morpheus, a U-Net [224] style convolutional neural network (CNN) model that filters out background pixels, uses a thresholding approach for detection, and combines watershed and peak finding algorithms for deblending. Another U-Net based model called `blend2mask` [34] performs detection and deblending using the U-Net alone. Reiman and Göhre [214] use a modified Super-Resolution GAN (SRGAN) [153] to deblend overlapping sources. Burke et al. [40] trained a Mask R-CNN [105] model to detect and deblend sources. SCARLET [179] deblends sources using constrained matrix factorization.

Table 3.1 summarizes the features of these previous methods, none of which have a detection capacity of N and an *intersecting/continuous* deblend type. I now present a deep learning-based *intersecting/continuous* deblending algorithm with a detection capacity of N.

3.2 Partial-Attribution Instance Segmentation

Partial-Attribution Instance Segmentation (PAIS) is a new extension of the instance segmentation paradigm that allows for weighted, overlapping segmenta-

tion maps. PAIS differs from other segmentation schemes like cell segmentation [299], interacting surface segmentation [291], and amodal instance segmentation [159]. PAIS aims to isolate objects appearing in an image while preserving their measurable quantities within areas of overlap. For PAIS, we can approximate Equation 3.1 as

$$\tilde{\mathcal{I}} = \sum_{i=1}^{N} M_i \odot \mathcal{I} \tag{3.2}$$

where $\tilde{\mathcal{I}} \in \mathbb{R}^{h \times w \times b}$ estimates the background-subtracted flux image $\mathcal{I}$ in Equation 3.1, $M_i \in [0,1]^{h \times w \times b} st. \sum_i^N M_{i,jkl} = 1$ constitutes the pixel-level fractional contribution of source i to $\mathcal{I}$, and $\odot$ symbolizes the Hadamard or element-wise product. Equation 3.2 is tractable for deep learning models, allowing the model to learn the bounded quantities M_i rather than the unbounded source images S_i. The N number of sources setting the upper limit of the sum in Equation 3.2 can differ for each image.

To construct a PAIS format that can be represented by a CNN, I have to construct an encoding for the M_i in Equation 3.2. Inspired by Cheng et al. [49] and Kendall et al. [138], I propose an encoding for the M_i components called *Partial Claim Representation* (PCR). The goal of PCR is to encode, for any single pixel (j,k,l), the fractional contribution to its intensity from the closest n sources. Using PCR, a variable number N of sources can be encoded per image. PCR consists of three tensors: the Center-of-mass $C^c \in \{0,1\}^{h \times w}$, Contribution-vectors $C^v \in \mathbb{R}^{h \times w \times n \times 2}$ and Contribution-maps $C^m \in [0,1]^{h \times w \times b \times n}$. The center-of-mass encodes the locations of all the sources in an image. For any pixel, I set $C^c_{jk} = 1$ if that location indicates the center of a source and $C^c_{jk} = 0$ otherwise. The contribution-vector C^v_{jk} encodes the Cartesian offset to the closest n sources. The contribution-map C^m_{jkl} connects the fractional contribution of the n sources

with the associated contribution-vectors C^v_{jk}. The fixed dimensionality of C^c, C^v, and C^m make PCR tractable for deep learning algorithms.

3.3 Our Approach

Our approach consists of making a PAIS dataset leveraging PCR and is implemented using a novel neural network architecture. I summarize our dataset, model, and training method below.

3.3.1 Dataset

To generate the PAIS input samples, I used the Hubble Legacy Fields (HLF) GOODS-South F160W (1.6μm) flux images [118], along with the 3D-HST source catalog [185]. The HLF images were split into training and test sets of 256 $\times$ 256 pixel subregions, with 2,000 training samples and 500 test samples. The input labels, as described in Section 3.2, consist of the center-of-mass images C^c, the contribution-vectors C^v, and the contribution-maps C^m. The center-of-mass training images are generated in a manner similar to Cheng et al. [49], by placing pixelated 2D Gaussians with standard deviation $\sigma = 8$(pixels) at the locations of sources in the 3D-HST catalog. The contribution-vectors, an extension to the method by Cheng et al. [49], are generated from the Cartesian offset to the nearest $n = 3$ sources to each pixel. The value n=3 is used because it is the smallest value and produces the most compact representation that can generally reproduce the true image values. The contribution-maps require the M_i values from Equation 3.2. To determine M_i, I use SCARLET [179] with the F125W, F160W, F606W, and F850LP flux and weight images from the HLF GOODS-South data and the TinyTim point-spread functions [148] to deblend the sources

Table 3.2: *Partial Claim Representation* encoding efficacy

Test	Value
Total Source Flux [e/s] (MAE)	1.97 ± 15.43
Two-Sample KS Test p-value	0.93 ± 0.22

from the 3D-HST catalog. I then use PCR to encode the M_i from SCARLET. The complete dataset generation routine can be found in our project repository (`https://github.com/ryanhausen/morpheus-deblend/`).

To evaluate the efficacy of PCR to encode M_i, I define two metrics. I use the mean difference between the total flux determined by the SCARLET encoded M_i for each input source and that recovered by our encoding. I also use a two-sample Kolmogorov–Smirnov (KS) test to compare the normalized cumulative surface brightness profile within the radius encompassing 90% of the total flux of each source to evaluate the encoding of the spatial flux distribution. Table 3.2 reports the results and demonstrates that PCR encoding approximately preserves both the total flux and the spatial flux distribution for each source. With this verification, I can train a network to recover the PCR encoding for each input HST F160W image.

3.3.2 Model

To recover the PCR for training images, I developed a novel neural network architecture inspired by Cheng et al. [49], based on the Fast Attention Network [111] and implemented in TensorFlow [1]. The model features two decoders that share a single encoder. The first decoder, called the *spatial decoder*, predicts values for C^c and C^v. The second decoder, called the *attribution decoder*, predicts values for C^m. The complete model code can be found in the repository for this project (`https://github.com/ryanhausen/morpheus-deblend/`). An end-

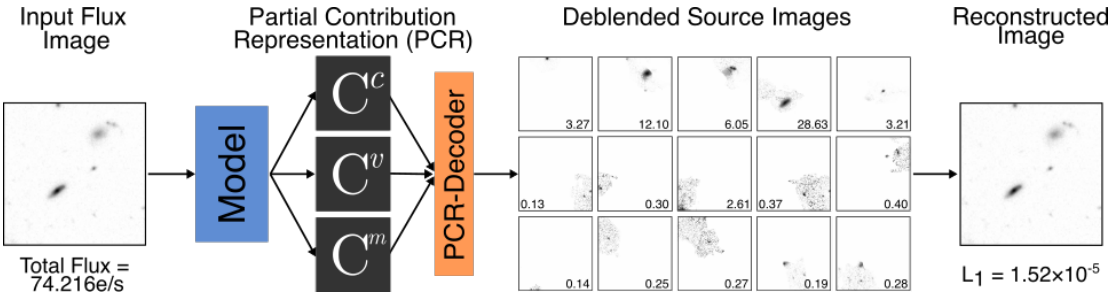

Figure 3.1: End-to-end example using our method to detect and deblend sources. Starting from the left: A flux image is input to the Model (see Section 3.3.2). The Model outputs the deblended image in the *Partial Claim Representation* (PCR; see Section 3.2).The output from the model is then decoded using the non-learned PCR Decoder algorithm into separate deblended source images. The deblended source images have their total flux within r_{90} annotated. The deblended source images are then added together to generate the reconstructed image which has an L_1 total flux difference of 1.52×10^{-5} with the original input image.

to-end example of the model can be seen in Figure 3.1.

3.3.3 Training

To train the model to recover the PCR of the input images, I use the Adam Optimizer [140] with a learning rate of 5×10^{-5}, $\beta_1 = 0.9$, $\beta_2 = 0.9999$, $\epsilon = 1\times10^{-7}$, and a batch size of 100. The model was trained for 1000 epochs using an NVIDIA V100 32GB GPU, taking 31 hours. The loss function for the model is composed of three functions. The *spatial decoder* outputs for C^c and C^v are penalized according to mean squared error (MSE) and the mean absolute error (MAE), respectively. The *attribution decoder* output C^m is penalized using cross-entropy loss with an additional entropy regularization term. In practice, I found that the additional entropy regularization helped incentivize the network to learn information about multiple sources in C^m. Each loss term is weighted and combined into a single loss function described by

Table 3.3: Training metric results

Metric	Training	Test
MAE	27.0183 ± 1.0658	28.5090 ± 0.3386
MSE	0.0114 ± 0.0001	0.0124 ± 0.0006
cross-entropy	0.9485 ± 0.0069	1.0806 ± 0.0098

$$\mathcal{L}_{\text{total}} = \lambda_{C^c}\mathcal{L}_{C^c} + \lambda_{C^v}\mathcal{L}_{C^v} + \lambda_{C^m}\mathcal{L}_{C^m} + \lambda_S\mathcal{L}_S, \tag{3.3}$$

where $\mathcal{L}_{C^c}$ is the MSE loss calculated between the model output and input label with $\lambda_{C^c} = 15$, $\mathcal{L}_{C^v}$ is the MAE loss calculated between the model output and input label with $\lambda_{C^v} = 0.06$, $\mathcal{L}_{C^m}$ is a cross-entropy loss calculated between the model output and input label with $\lambda_{C^m} = 4$, and $\mathcal{L}_S$ is the entropy regularization on the model C^m output with $\lambda_S = 2$. See Table 3.3 for a summary of the training results, demonstrating a good balance between test and training error. A complete log of training experiments is available at (`https://www.comet.ml/ryanhausen/morpheus-deblend/`).

3.4 Discussion and Future Work

In this chapter, I introduced the Partial Attribution Instance Segmentation (PAIS) scheme for astronomical source debelending. I presented *Partial Claim Representation* (PCR) as a method for implementing PAIS within deep learning-based models. I demonstrated the efficacy of PCR for encoding the results of existing astronomical deblenders, and developed a novel neural network architecture to recover the PCR from input flux images. While I demonstrated deblending for single band (F160W) images, PCR can be extended to multi-band images. As with many supervised methods, our model requires labeled training data. To ap-

ply this method on other survey datasets may require the use of transfer learning [74, 212] or retraining.

Chapter 4

FitsMap: A Simple, Lightweight Tool For Displaying Interactive Astronomical Image and Catalog Data

4.1 Introduction

Astronomical image data is inherently visual, and visual inspection and interpretation remain vital tools in the scientific process within astronomy. Upcoming telescopes like the James Webb Space Telescope [JWST; for a review, see Robertson [221]], Nancy Grace Roman Space Telescope [245, 9], and Vera Rubin Observatory [122, 123] will produce larger and deeper images of space than ever before. Specialized tools for visualizing and interacting with large-scale astronomical images will enable a more rapid transition from observation to analysis for these facilities.

Visualizing large astronomical images is not a new problem, and several tools have been developed to meet this need. There are desktop tools like `DS9` [127], `AstroVis` [207], `Fips` [146], `FITS3D/2D` [183], and `Aladin` [32], which are *thick* clients that render local images or fetch images from a remote server. The term *thick client* refers to programs with little to no reliance on a remote server to perform the computational tasks associated with the software. Advances in web technologies have also enabled the development of *thin* clients for astronomical image data visualization. *Thin* clients are dependent on a separate server to perform their computational tasks. An advantage to using thin clients is that by offloading heavier computational tasks to a remote server, the client hardware requirements are far less powerful than the server requirements and can even include mobile devices. Some examples of thin clients developed for astronomical image, and catalog visualization include `VisiOmatic` [23, 24], `Aladin Lite` [29], `RCSED` [52, 141], the GAIA archive visualization service [184], `Toyz` [187], WorldWide Telescope [225], `ASTRODEEP` [69, 279], `ESASky` [90], Virtual Desktops [178], and Hassan et al. [100]. More recently, the popularity of the Python [270] programming language has inspired a series of new *thick* and *thin* clients that leverage Python and the Jupyter Lab/Notebook [iPython; 142] ecosystem. Some examples include `Vizic` [296], `LSSGalPy` [13], Astro Data Lab [128], `Jovial` [12], and `Jdaviz` [201].

This chapter introduces a new tool called *FitsMap*[103]. *FitsMap* is designed to work for a simple use case where the user has images including `FITS` [282], `PNG` [120], or `JPEG` [119] and catalogs associated with the image data, and would like to view and possibly share that data interactively. A user can generate a website that displays image and catalog information using a single *FitsMap* function or command-line interface call. *FitsMap* improves on previous methods with two

innovations:

1. Catalog data are preclustered and tiled, similar to how image data are treated.

2. All image and catalog pre-processing is handled internally by *FitsMap*, including the website directory structure and supporting files.

Preclustering and tiling the catalog data offer the same performance advantages as tiling the image data, as only the portion of the catalog data currently in view needs to be loaded. Tiling the catalog also removes the need for an additional process or database to serve the marker and cluster data and can scale to arbitrarily sized catalogs (see Section 4.2). The ability of *FitsMap* to process the image and catalog files and generate the website directory structure is novel. Some visualization tools such as `VisiOmatic`, `Aladin Lite`, and WorldWide Telescope expect the image data to be pre-tiled and direct users to separate, affiliated software to perform image tiling. Some existing visualization methods also typically require catalog data to be pre-processed by another tool. For example, `Alaldin lite` requires that users convert their catalog data using `HiPS-cat` before it can be visualized.

Popular non-Jupyter web-based tools, including `VisiOmatic`, `Aladin Lite`, and WorldWide Telescope, require familiarity with JavaScript and HTML to write the webpage that requests the data to display. In contrast, *FitsMap* does not require the user to write any HTML/JavaScript or, if the user employs the command-line interface, perform any programming at all. *FitsMap* has already been deployed to visualize large simulated JWST [288] and Roman [75] datasets and machine-learning galaxy morphological analyses of large Hubble Space Telescope surveys [104].

The remainder of this chapter is structured as follows. Section 4.2 describes the details of the design and methods of *FitsMap*. Section 4.3 describes the performance of *FitsMap*, and Section 4.4 reviews the contributions and future directions of *FitsMap*.

4.2 Methods

The design philosophy behind *FitsMap* is to render visualizing image and catalog data simple by minimizing the number of steps and technical knowledge required to go from image and catalog files to an interactive display. The only requirements to generate and view the output from *FitsMap* are *FitsMap* itself and a web server. If viewing the output locally, *FitsMap* can also run the web server. If *FitsMap* is used to view image data alone, a web server is not required, as the user can simply open the output `index.html` file using a browser, and the processed images will render.

Catalog data present a unique challenge in the effort to minimize server requirements. Small catalogs with low storage requirements can be transferred from the server to be clustered and rendered on the client, but large catalogs practically cannot. Large catalogs, those with high storage requirements, may either take too long to transfer to the client from a user experience perspective or exceed the memory capabilities of the client. Managing the catalog data on the server-side requires a more sophisticated web server that clusters and serves catalog data. Leveraging the continuing decline in the cost of storage [177], *FitsMap* separates the catalog data into a set of tiles at different zoom levels. *FitsMap* then precomputes clusters of the catalog data at every zoom level for every tile (Section 4.2.1). Precomputing the clustering and tiling the data significantly reduces the computational requirements of both the server and client by never requiring the

entire catalog to be loaded into memory.

The *FitsMap* architecture consists of two main components: the *Map Generator* and the *Map Viewer*. The *Map Generator*, implemented in Python, builds the website, including the *Map Viewer* files. The *Map Viewer*, implemented in Javascript, renders the webpage and fetches the image and catalog data.

4.2.1 *Map Generator*

The *Map Generator* parses a list of the input image and catalog file locations and converts them into a format that the *Map Viewer* can render. The output of the *Map Generator* is a directory that contains the processed input image and catalog data and the HyperText Markup Language (`HTML`), Cascading Style Sheets (`CSS`), JavaScript, and images. The *Map Generator* can process multiple files in parallel and further parallelize each file's processing. For example, image and catalog tiles can be generated in parallel. See Figure 4.1 for a graphical representation of the *Map Generator*. The *Map Generator* performs two significant tasks when generating an interactive web website: parsing image files and parsing catalog files. *FitsMap* dynamically generates the *Map Viewer* files using the image and catalog data information. FitsMap sites can be updated when image or catalog data has been changed by deleting the directory in the output website associated with the image or catalog and rerunning FitsMap. FitsMap will then only generate the missing catalog or image data.

Parsing Image Data

The first step in parsing the image data is to convert the image data into a `numpy` [99] array. The *Map Generator* converts `FITS` files using `astropy` [14, 15]

and converts traditional image files (`PNG`, `JPG`, etc.) using `pillow` [1]. The *Map Generator* builds tiled representations, in parallel if desired, of the array data at each zoom level in a structured set of directories compatible with the *Map Viewer*. The tiled representations of each processed image can be viewed as separate layers in the *Map Viewer*.

Parsing Catalog Data

The *Map Generator* can parse plain text files that are delimited (`CSV`, `TSV`, etc.) and where the first line contains the column names. Catalog source locations can be stored in either $[x, y]$ (image pixel coordinates) or $[\alpha, \delta]$ (right ascension and declination). If the coordinates are given in $[\alpha, \delta]$, then a `FITS` file containing a reference World Coordinate System (`WCS`) to translate the sky coordinates to image coordinates must be provided. After reading the catalog data, the *Map Generator* clusters the data at every zoom level using a python port[2] of the `supercluster`[3] JavaScript library. Then the *Map Generator* builds a tiled representation of the catalog source and cluster locations at every zoom level. The tiled catalog source locations are encoded and stored using the MapBox Vector Tile Format[4], a compact binary representation of structured data representing geometric objects in map tiles. Other source attributes (effective radius, magnitude, etc.) are stored in a separate Concise Binary Object Representation [CBOR; 33] file for each source in a dedicated directory for each catalog.

[1] `https://python-pillow.org/`
[2] `https://github.com/ryanhausen/supercluster-py`
[3] `https://github.com/mapbox/supercluster`
[4] `https://github.com/mapbox/vector-tile-spec`

4.2.2 *Map Viewer*

The *Map Viewer* consists of the `HTML`, `CSS`, JavaScript, and image files that render the image and catalog data. The *Map Viewer* is built using `Leaflet`[5], an open-source and flexible framework for displaying large amounts of visual information. A custom Leaflet layer `L.GridLayer.TiledMarkers`, packaged with FitsMap, enables Leaflet to render catalog markers and clusters generated by the Map Generator. The *Map Viewer* fetches and renders markers (sources and clusters) using the data stored in each of the encoded tile files; see Figures 4.2 and 4.4 for an illustration and an example of image and catalog tiling.

The *Map Viewer* supports displaying both circular and elliptical markers. The *Map Viewer* will render the catalog marker as an ellipse if the catalog data contains columns for semi-major axis `a`, semi-minor `b`, and position angle `theta`. Otherwise, the marker will be rendered as a circle. Fitsmap requires `a` and `b` to be in pixel units and `theta` in radians.

When a source marker is selected, the *Map Viewer* fetches additional information about the source from its corresponding `CBOR` file (see Section 4.2.1 and Figure 4.1) and renders it in a pop-up above the marker (see Figure 4.3). The marker pop-ups support `HTML` to enable the inclusion of images and other valid HTML.

The *Map Viewer* includes a search function that searches catalog entries by the `id` column. The search functionality is implemented using a custom search backend to the popular Leaflet Control Search plugin[6] packaged with *FitsMap*. To search by `id`, the *Map Viewer* sends a `GET` request for a `CBOR` file named `{id}.cbor` in each catalog's extra `CBOR` source file directory in `catalog_assets` (see Figure 4.2). If the request returns `404` (file not found), the `id` does not exist.

[5] `https://leafletjs.com/`
[6] `https://opengeo.tech/maps/leaflet-search`

If the request returns `200` (OK), the desired object exists, and the *Map Viewer* will extract location information stored in the file and pan to and highlight the source.

FitsMap does not support changing marker colors or catalog popup appearance from the Python or CLI interfaces. However, the *Map Viewer*, being a website, is highly customizable to users familiar with HTML, CSS, or JavaScript.

4.3 Performance

As described in Section 4.2, to reduce the computational requirements of both the server and the client, *FitsMap* tiles the marker data such that the client only renders the markers that are currently in view and unclustered at the current zoom level. In this section, the performance of *FitsMap* is evaluated in two ways. First, the *Map Generator* is evaluated with respect to computational and storage requirements. Second, the performance of the *Map Viewer* is evaluated with respect to speed and memory on both desktop and mobile environments.

4.3.1 *Map Generator* Performance

The *Map Generator* performance is empirically evaluated with respect to the time needed to tile various sized FITS images and the time and storage requirements for parsing and tiling catalog data. Synthetic image and catalog data were used to perform the evaluations. The images consisted of $n \times n$ arrays of Normally distributed values ($\mu = 1, \sigma^2 = 1$) with 32-bit floating-point precision, where $n \in \{2^{10}, 2^{11}, 2^{12}, 2^{13}, 2^{14}, 2^{15}, 2^{16}\}$. The synthetic catalog consisted of m rows of data, where $m \in \{1 \times 10^3, 5 \times 10^3, 1 \times 10^4, 5 \times 10^4, 1 \times 10^5, 5 \times 10^5, 1 \times 10^6\}$, with 43 columns of data. Here the first column is a unique integer column (`id`), the next

two columns are randomly sampled floating-point values between $[0, 1)$ representing the x and y coordinates, the following twenty columns are randomly generated integer values between $[0, 100)$, and the final twenty columns are randomly generated floating-point values between $[0, 1000)$. When parsing the catalog, the $[x, y]$ coordinates are rescaled by the pixel size of the corresponding image to fill the entire scene. The performance tests are evaluated using an Intel Xeon E5-2698 system with 232 GB of RAM and six workers for parallel processing.

The time required for *FitsMap* to process the synthetic image and catalog data is reported in Figure 4.5. The results indicate that image parsing and tiling times scale linearly with the total number of tiles generated. Note that the numbers of tiles generated scale geometrically with image size. Catalog parsing and tiling times also scale linearly with the total number of sources. Note that the image size is kept fixed for the catalog parsing and tiling to prevent the number of tiles generated from influencing the run time. The storage cost for catalog data is composed of the cost to store the catalog source data in the CBOR format and the cost to store the catalog source location information as clustered tiles (See Section 4.2.1). The storage cost for encoding the data into the CBOR format as a function of the catalog size is shown in Figure 4.6, which indicates that the total data volume of the CBOR-encoded version of the catalog converges to approximately 115% of the original catalog size. Figure 4.7 shows the tiled catalog data storage requirements as a function of catalog size and image size. The storage cost for the tiled catalog location information is generally less than the input catalog size.

4.3.2 *Map Viewer* Performance

The *Map Viewer* is written entirely in JavaScript and only requires a web server to render the output from the *Map Generator*. The *Map Viewer* performance was measured for a *FitsMap* generated using the catalog and image data described in Section 4.3.1. Specifically, the $65,536^2$-pixel image (17.2 GB) is paired with the $1,000,000$ sample catalog (467.4 MB). The *FitsMap* was hosted on a server at the University of California, Santa Cruz at the time of testing and can be accessed publicly [7]. Two types of tests were run against the example *FitsMap* described above: page loading speeds and session memory usage. Both sets of tests were executed using BrowserStack[8], a service that performs speed and automated browser interaction tests free of charge to open source projects.

The speed of the *FitsMap Map Viewer* was evaluated using seven different mobile devices and three different desktop web browsers. Table 4.1 contains the details of the testing environments. Each test is run three times, with the mean and standard deviation for each benchmark reported in Figures 4.8 and 4.9. The speed tests are comprised of opening the URL for the *FitsMap* to be tested in each testing environment and recording the following performance metrics: *Page Load*, *First Contentful Paint* (FCP), and *Document Object Model Processing* (DOM Processing).

Page Load indicates the amount of time it takes for the page to load fully. Figures 4.8 and 4.9 indicate that *FitsMap* is performant with respect to speed, with page load times of around 2 seconds. *FCP* tests the time elapsed from the initial page request to perceived page rendering and corresponds to when the user witnesses the first part of the site being rendered on the screen. Walton [278] indicates that websites should aim for an FCP of less than 1.8 seconds. Our

[7] `https://purl.org/fitsmap/performance-testing`
[8] `https://browserstack.com`

testing finds that *FitsMap*, excluding a single run on Desktop Google Chrome, has FCP times well below 1.8 seconds, as seen in Figures 4.8 and 4.9. *DOM Processing* indicates the amount of time to parse the retrieved HTML document and build the DOM. *DOM Processing* times can indicate website performance issues in at least two ways. First, if *DOM Processing* times are long, a website may have too many thread-blocking resources preventing the page from loading. Second, if there is a large disparity between the *DOM Processing* and *Page Load* times, it can indicate poor optimization in static assets, like JavaScript, that delay page rendering. Figures 4.8 and 4.9 show that *Page Load* and *DOM Processing* times are nearly identical, indicating that the page is ready to be rendered as soon as the DOM is built.

Another benchmark used to evaluate the *Map Viewer* performance was *Session Memory Consumption*. Session memory consumption consists of opening the test *FitsMap* page and performing navigation and interaction tasks while monitoring memory usage and includes the following tasks in order:

1. Open web browser
2. Navigate to the test page
3. Add the catalog markers to the map
4. Zoom in (7x) to display unclustered catalog markers
5. Select a marker that displays a catalog popup
6. Close the catalog popup
7. Zoom out (7x)

Device	Browser
Google Pixel 3	Google Chrome
Samsung Note 10	Google Chrome
Samsung Galaxy S10	Google Chrome
Samsung Galaxy A10	Google Chrome
Apple iPhone XS	Safari
Apple iPhone 12	Safari
Apple iPad Air 4	Safari
Desktop OSX	Microsoft Edge
Desktop OSX	Google Chrome
Desktop OSX	Mozilla Firefox

Table 4.1: *Map Viewer* testing environments. Test environments were provided by and executed using BrowserStack. All mobile environments have a network speed throttled to model 4G speeds (9Mbps), and all desktop environments were throttled to have a network speed throttled to model broadband speeds (40Mbps).

The tests were scripted using the Selenium[9] browser automation framework and executed using the BrowserStack testing framework. Memory consumption was monitored using Google Chrome's proprietary JavaScript Application Programming Interface (API), and so the results are limited to testing environments using Chrome. Each test was executed three times, and the mean and standard deviation of the tests are reported in Figure 4.10. The results in Figure 4.10 show that memory usage remains low for both mobile and desktop environments.

4.4 Conclusion and Future Work

This chapter introduced *FitsMap*, a simple, lightweight tool for generating interactive web-based visualizations of astronomical images and catalogs. *FitsMap* uses a novel approach to serve catalog data that precomputes cluster information for the entire image at all zoom levels and displays the cluster and source information based on visible image tiles. Advantages of precomputing

[9] `https://www.selenium.dev`

and tiling the catalog data include 1) the visualization only requires a simple web server and 2) the produced website loads quickly and proves responsive for large catalogs, even on mobile devices. The techniques developed in this work have already been used on public-facing websites. Both Hausen and Robertson [104] and Drakos et al. [75] leverage *FitsMap* to generate interactive visualizations displaying their data products (See `https://purl.org/fitsmap/morpheus` and `https://purl.org/fitsmap/dream`, respectively). Potential drawbacks to *FitsMap* include the increased storage requirements above the input image and catalog data sizes that owe to the tiling method and CBOR encoding. The *FitsMap* clustering and tiling require pre-processing of the catalog data, but the pre-processing only needs to be performed once per website. The development of *FitsMap* is active and ongoing. Future work includes allowing for more complex shapes in catalog data such as polygons, allowing users to change `FITS` image scaling on the fly in the *Map Viewer*, minimizing storage usage, and supporting other catalog formats.

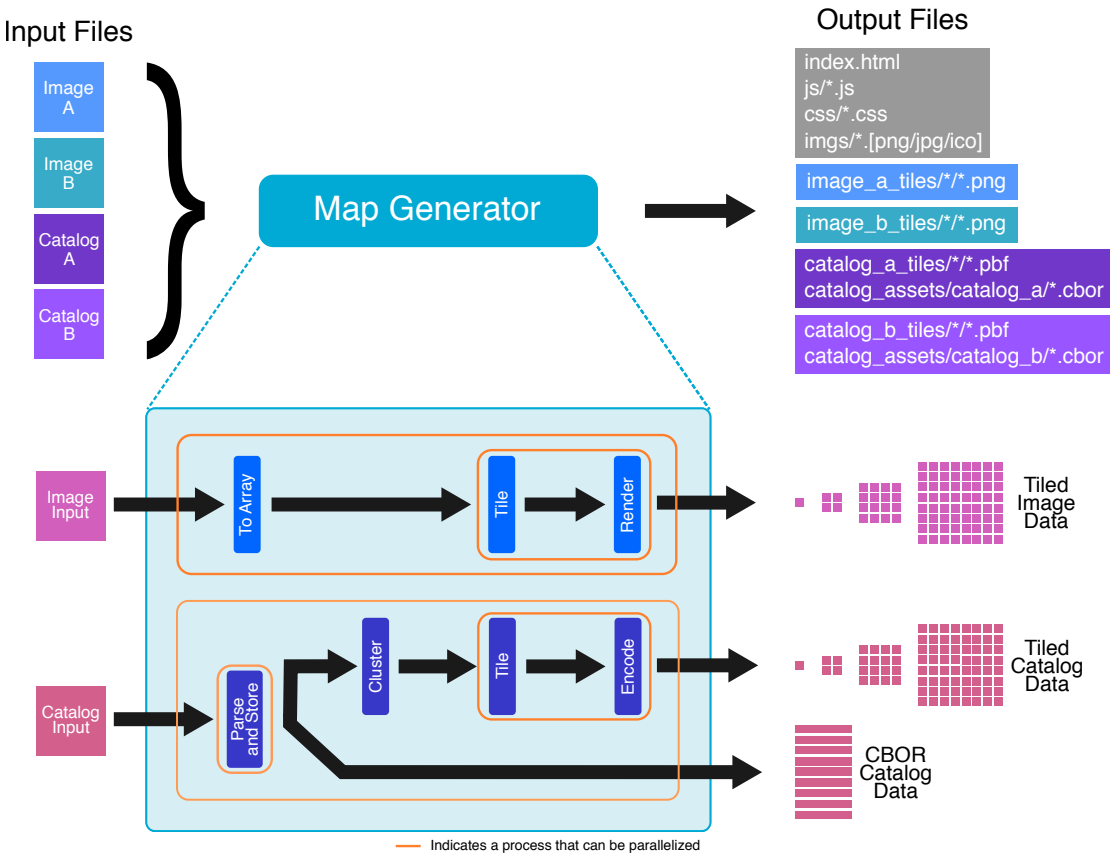

Figure 4.1: An overview of the *Map Generator* in the *FitsMap* architecture. The *Map Generator*, described in Section 4.2.1, processes the input image and catalog files and converts them into a format that can be rendered by the *Map Viewer* (Section 4.2.2). The *Map Generator* can process multiple files in parallel and further leverage parallelism when processing each file. The *Map Generator* builds a directory containing the tiled image and catalog data along with the *Map Viewer* data and code (gray). The output website can be viewed using a simple web server.

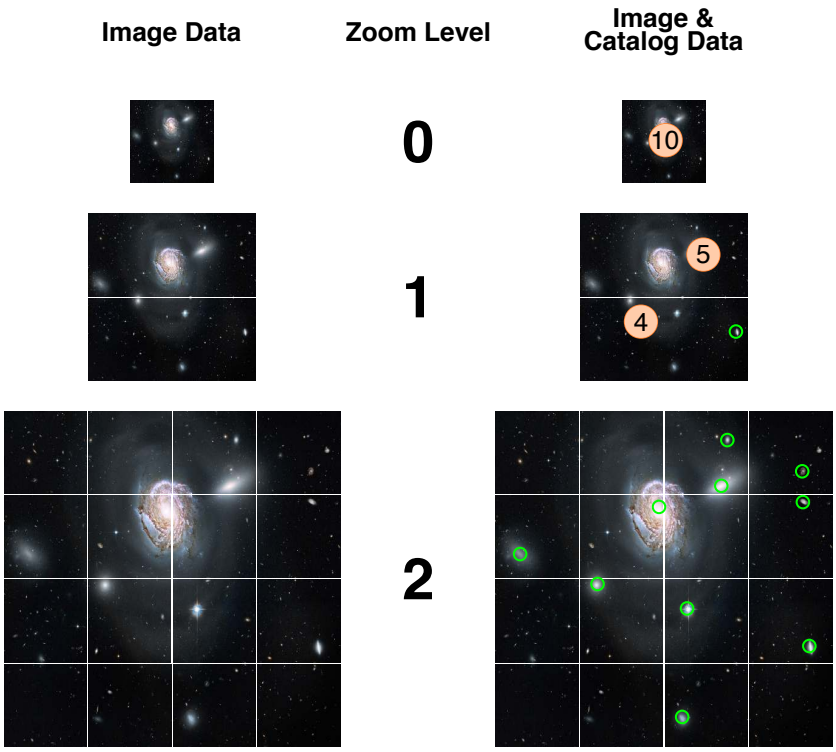

Figure 4.2: An illustration of tiling image and catalog data at three zoom levels (0, 1, 2). (left) Illustration of tiling image data only. (right) Illustration of tiling image and catalog data. For simplicity, not all sources in the image are labeled with a marker. *FitsMap* computes the cluster assignments at every zoom level and stores them as markers in their respective tile. When the *Map Viewer* requests the catalog data associated with a particular tile, it will retrieve source markers and/or cluster markers and renders them appropriately. Source image credit: NASA.

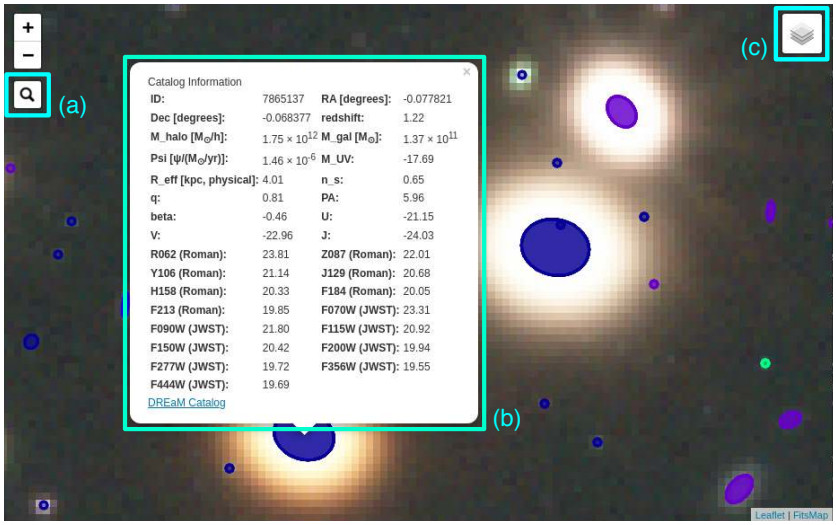

Figure 4.3: An example of the *FitsMap* interface. (a) The search function button searches catalogs by `id` (see Section 4.2.2). (b) A marker pop-up displaying catalog data associated with the indicated source. (c) The `Leaflet` layer control allows users to switch between display images and catalog overlays. Image Credit: [75].

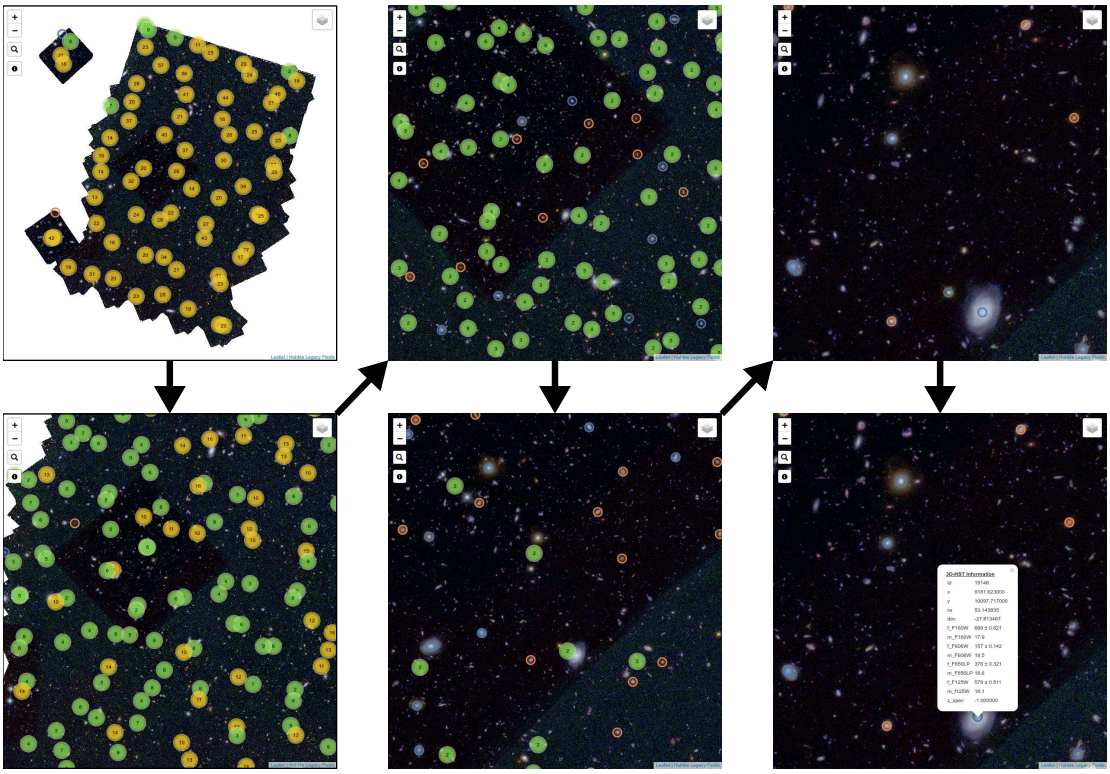

Figure 4.4: Viewing the catalog clustering levels as a function of zoom level. Starting from the upper-left panel and following the arrows until the upper right panel are progressively increasing zoomed-in views of the data released by Hausen and Robertson [104]. Two catalogs are displayed simultaneously, indicated in blue and orange. The final bottom right panel shows the same image as the upper right panel, but with the popup for one of the catalogs.

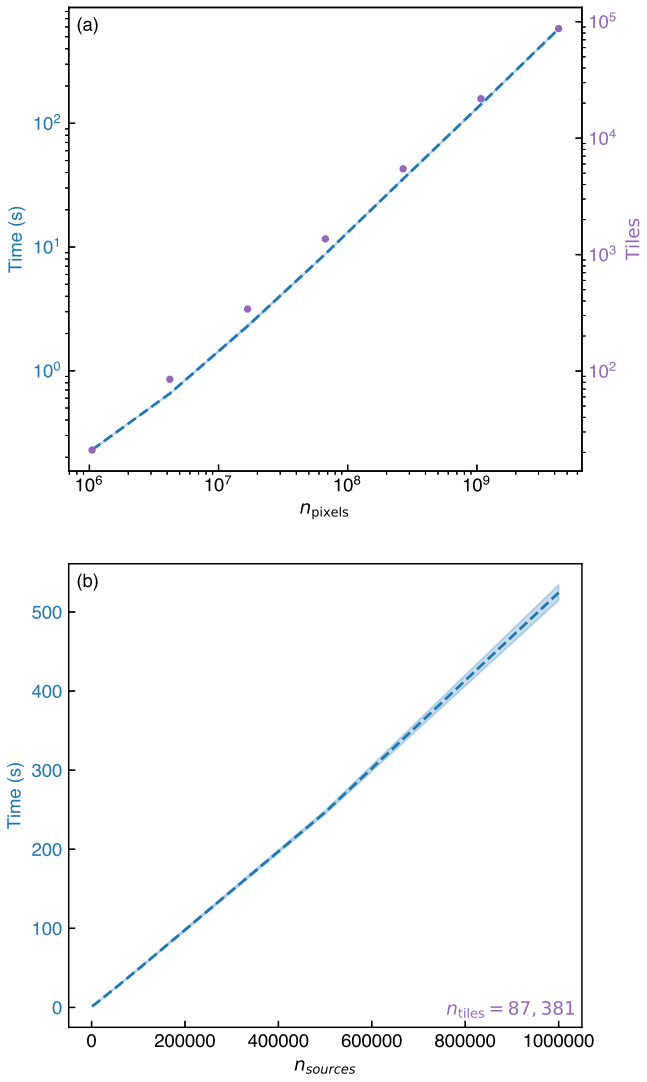

Figure 4.5: FitsMap performance parsing and tiling image and catalog data. (a) The time, measured in seconds, to tile and generate a FitsMap consisting of a single image. The image size is indicated along the x-axis, where the image size is $\sqrt{n_{\text{pixels}}} \times \sqrt{n_{\text{pixels}}}$. The primary y-axis (blue) indicates the total amount of time to tile the image and generate the output website directory. Every image is generated five times with the mean plotted as a blue dashed line, and a single standard deviation shaded as light blue around the mean line. The secondary y-axis (purple) indicates the tot number of tiles generated in the output tiled image directory. (b) The time, measured in seconds, to parse a tile catalog file. Each catalog was scaled to fit an image size of $65,536 \times 65,536$, resulting in approximately $87,381$ tiles. Each catalog was parsed and tiled fives times with the mean plotted as the blue dotted line, and a single standard deviation shaded around the mean in light blue.

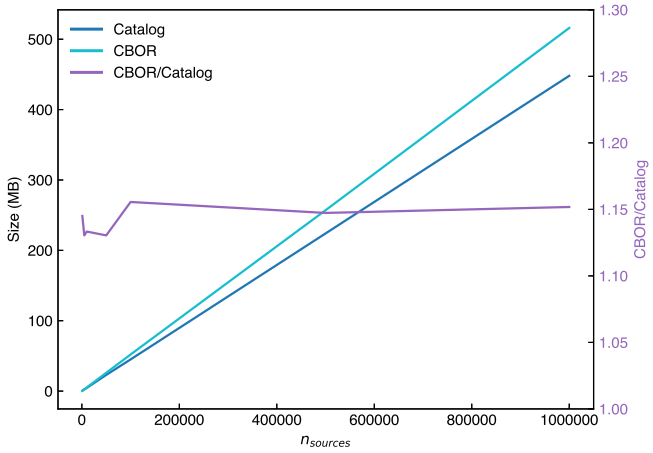

Figure 4.6: Storage comparison of catalog data and its CBOR-encoding used by *FitsMap*. The primary axis (black) shows the total storage use of the catalog data and the CBOR-encoded version (see Section 4.2.1 for more information on CBOR encoding). The secondary axis (purple) shows the ratio of the storage requirements of the CBOR encoded version of the catalog data to the original catalog data.

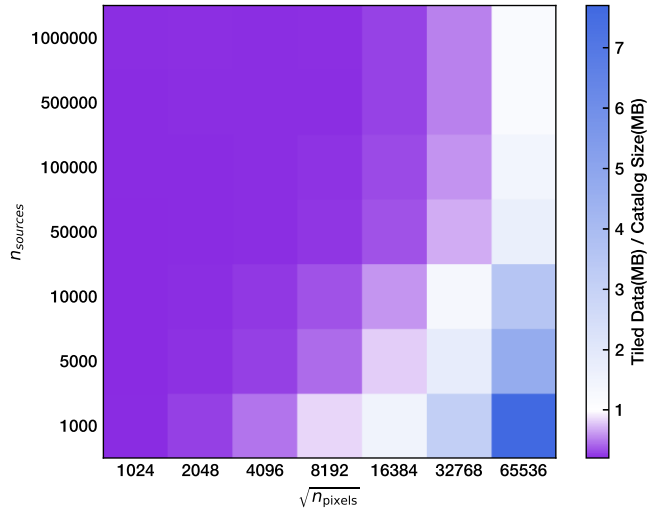

Figure 4.7: The ratio of the total storage requirements of the tiled catalog data to the original catalog size. The x-axis indicates the image size, and the y-axis indicates the catalog size. The color in the image indicates the value of the ratio, where purple indicates a value less than one, white indicates a value near or equal to one, and blue indicates a value greater than one.

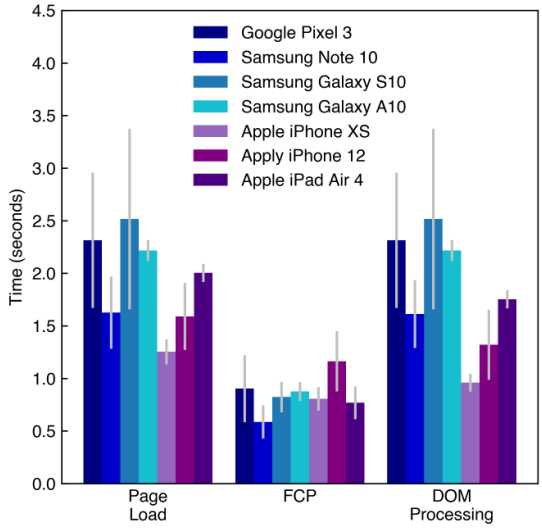

Figure 4.8: *FitsMap* page speed metrics for mobile browsers. *Page Load* indicates the amount of time required for the page to load fully after sending the request to the server. *FCP* (First Contentful Paint) indicates the amount of time from page loading to seeing something rendered on the page. *DOM Processing* indicates the amount of time after the browser starts processing the page until the content is loaded. Each test was performed three times with the mean and a standard deviation plotted.

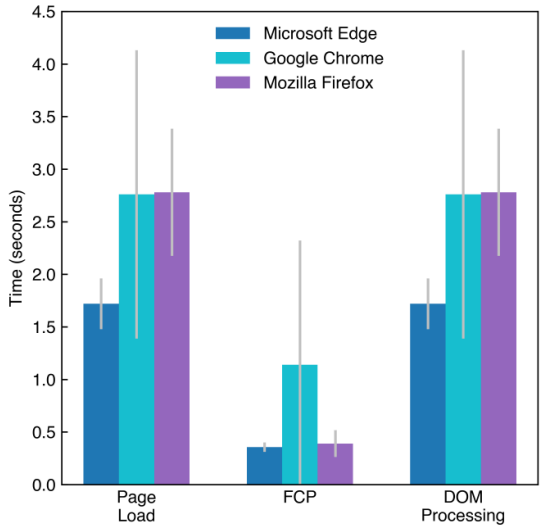

Figure 4.9: *FitsMap* page speed metrics for desktop browsers. *Page Load* indicates the amount of time it takes for the page to load fully. *FCP* (First Contentful Paint) indicates the amount of time from page loading to seeing something rendered on the page. *DOM Processing* indicates the amount of time after the browser starts processing the page until the content is loaded. Each test was performed three times with the mean and a standard deviation plotted.

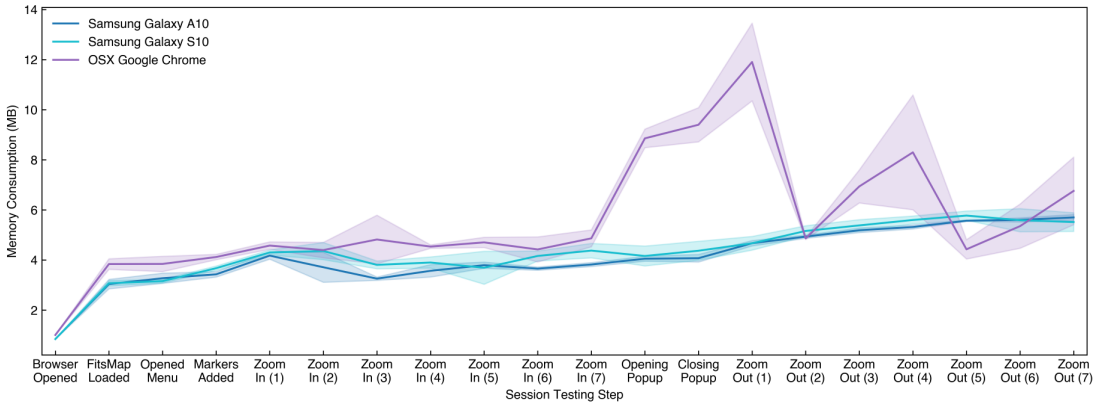

Figure 4.10: *Map Viewer Session Memory Consumption* during a typical interaction on a test *FitsMap*. The website was generated according to the procedure described in Section 4.3.2. The steps taken during the test are indicated in order along the x-axis. The memory consumption is indicated on the y-axis in megabytes. The tests were run using the Selenium browser automation tool and run on the BrowserStack test framework. Each test was run three times with the mean and the standard deviation reported above. Memory consumption reporting is reported for Chrome browsers which support reporting memory consumption via JavaScript.

Chapter 5

Revealing the Galaxy-Halo Connection Through Machine Learning

5.1 Introduction

Numerical simulation enables theoretical models of galaxy formation to include detailed physical models for baryonic processes. Simulations can capture the physics of cooling, supernova feedback, radiative feedback and ionization, and the role of dynamics simultaneously while tracking the growth of cosmological structure formation [e.g., 229, 209, 65]. The simulated galaxy populations that result from these models reproduce observed stellar mass sequences such as the main sequence of star-forming galaxies [37, 195] or the red sequence of quiescent galaxies [81]. The quest for realism in modeling these observed trends has also added substantial complexity, such that understanding which physical properties of a galaxy most influence its stellar mass and star formation rate can prove chal-

lenging. Many theoretical frameworks to describe these relations have been developed [e.g., 280], including halo occupation distribution models [e.g., 126], subhalo abundance matching [264, 55], and semi-analytic models [for a review, see 243]. The complex physics encoded by these models and simulations can be difficult to interpret, and the relative contribution of baryonic feedback, dark matter halo formation, and environment in setting galaxy properties remains challenging to disentangle.

This complexity extends to cosmological models of galaxy formation in the reionization epoch. To capture the distribution of sizes of ionized regions with converged simulations [117] and the largest observed features, such as dark gaps [302], the volume of reionization simulations should extend to a least several hundred megaparsecs. Modeling such large volumes in a single simulation while maintaining the spatial resolution needed to include the complex physics of the current state-of-the-art projects, such as Cosmic Reionization on Computers [CROC 92], THESAN [131], or Cosmic Dawn [CoDa, 198, 199], remains computationally infeasible. Instead, we desire an intermediate approach where large volumes are simulated and the physics of galaxy formation are implemented with a approximate model that recovers the mean trends for galaxy baryonic properties predicted by more detailed calculations. With this goal in mind, a model for reionization sources that encapsulates the results of projects like CROC in a simple module is the first necessary step for deploying lower resolution simulations with much larger ($L \sim 500$cMpc) simulation volumes. If the stellar mass and star formation rates of ionizing sources can be predicted from their dark matter halo properties and environment, then we can account for the ionizing photons produced by these sources in large-box simulations of the reionization process without resolving the baryonic physics in detail.

This chapter employs a machine learning method called Explainable Boosting Machines [EBMs 166] to infer how stellar mass $M_\star$ and star formation rate *SFR* depend on the physical parameters $\boldsymbol{\theta}$ of a host galaxy. In this chapter, we use the galaxy populations from the CROC simulations to provide our training and test data that populate samples in the multidimensional parameter space of $M_\star$-*SFR*-$\boldsymbol{\theta}$. For the additional parameters $\boldsymbol{\theta}$, we use a wide range of physical characteristics measured for galaxies in CROC including the virial mass $M_{\rm vir}$, redshift z, environmental properties averaged on a length scale R, and the maximum peak circular velocity $v_{\rm peak}$. We can then use this approximate machine learning-based EBM model for galaxy formation as a basis for future development to incorporate the CROC galaxy population as sources in lower resolution, large-volume reionization simulations.

EBMs represent a form of Generalized Additive Models [101, GAMs] where the dependencies of a *target* quantity, such as $M_\star$ or *SFR*, on each physical parameter θ_i are encapsulated by *feature functions* of one parameter (e.g., $f^i(\theta_i)$) or *interaction functions* of two parameters (e.g., $f^{ij}(\theta_i, \theta_j)$). An EBM model is trained to fit these functions from a provided multidimensional dataset. The predicted value of the target quantity given the parameters (e.g., $\gamma(M_\star|\boldsymbol{\theta})$) is then a sum of the functions f^i and f^{ij}. EBM models are often described as *interpretable* because the magnitudes of the functions f^i and f^{ij} directly indicate the relative importance of $\boldsymbol{\theta}$ in determining the target quantity. If a given parameter θ_i is unimportant for determining the target quantity, the EBM will find $f^i \to 0$. A formal defintion of the EBM is provided in Section 5.2.1.

Previous works have applied machine learning models to infer connections between simulated galaxy properties. Lovell et al. [167] use a tree-based learning method called Extremely Randomized Trees to map baryon information to dark

matter halos in the EAGLE simulations. Xu et al. [292] train a Random Forest to predict the number of central and satellite galaxies in dark matter halos in the Millennium simulation. Machado Poletti Valle et al. [170] used an XGBoost model to predict gas shapes in dark matter halos in the IllustrisTNG simulations. Bluck et al. [28] used Random Forest classifiers to study quenching mechanisms in observations, semianalytical models, and cosmological simulations. Piotrowska et al. [210] also used Random Forest classifiers to examine how supermassive black hole feedback quenches central galaxies in the EAGLE, Illustris, and IllustrisTNG simulations. Our approach complements these prior works by studying the detailed connection between halo and environmental properties, star formation rate, and stellar mass in a model that can be directly implemented in future large-volume cosmological simulations with limited spatial resolution.

The chapter is organized as follows. In Section 5.2 we review the EBM methodology, define our training dataset and procedure, and introduce the evaluation metrics used to assess the performance of the model. In Section 5.3 we present the average contribution of each parameter to the target quantities, the best-fit feature and interaction functions, and the performance of the model in determining the distributions of stellar mass and star formation rate as a function of halo virial mass. We then explore in Section 5.4 methods for constructing *composite* EBM models to recover the stellar mass (Section 5.6) and star formation rate (Section 5.5) of simulated galaxies that only use instantaneous halo virial properties and environmental measures (i.e., excluding $v_{\rm peak}$). We discuss our results in Section 5.7, and summarize them and conclude in Section 5.8.

5.2 Methods

To infer the connection between $M_\star$, *SFR*, and other physical properties of simulated galaxies, we apply EBM models to the CROC simulated galaxy catalogs. In Section 5.2.1, we define the EBM model. We select our model parameters and describe the simulated galaxy catalog used to train the model in Section 5.2.2. The training procedure is outlined in Section 5.2.3.

5.2.1 Explainable Boosting Machines

Explainable Boosting Machine [166, EBM] models provide a fitted representation of the relationship between the target quantities y and the parameters $\boldsymbol{\theta}$. EBMs are an extension of Generalized Additive Models [101, GAMs], which represent target quantities y as the sum of learned univariate functions $f^i(\theta_i)$ that depend on only one parameter θ_i. EBMs extend GAMs by including both univariate functions $f^i(\theta_i)$ and bivariate functions $f^{ij}(\theta_i, \theta_j)$ that represent dependencies on pairs of features (θ_i, θ_j) beyond the dependence of the target quantity on either feature independently. Both EBMs and GAMs are forms of regression where the feature functions f^i and f^{ij} can be quite general.

The EBM aims to encode the average dependence of a target quantity y on the parameters $\boldsymbol{\theta}$. Mathematically, an EBM can therefore be represented as

$$\gamma(y|\boldsymbol{\theta}) = \beta_y + \sum_{i=0}^{n_\mathrm{p}-1} f_y^i(\theta_i) + \sum_{i=0, i\neq j}^{n_\mathrm{p}-1} \sum_{j=0}^{n_\mathrm{p}-1} f_y^{ij}(\theta_i, \theta_j) \tag{5.1}$$

where $\gamma(y|\boldsymbol{\theta})$ is the predicted value of the target quantity y given n_p parameters $\boldsymbol{\theta} \in \mathbb{R}^n$ from the dataset. We will refer to learned parameter β_y as the *baseline value* of the target quanity y. Though f_y^i and f_y^{ij} can be any interpretable function (e.g., linear regression, splines, etc.), Lou et al. [165] found that gradient boosted

trees [83] work best in practice. Using gradient boosted trees, the functions f_y^i and f_y^{ij} will be piece-wise one- and two-dimensional functions, respectively. By expressing the dependence of y on $\boldsymbol{\theta}$ directly through the functions f_y^i and f_y^{ij}, EBMs are interpretable and decomposeable. Further, after training is complete the learned tree-based functions f_y^i and f_y^{ij} can be formulated as look-up tables for performant inference.

5.2.2 Simulated Galaxy Catalog Training Set

To engineer an EBM that describes the connection between simulated galaxy properties, their host dark matter halos, and features of the extrinsic environment, we turn to established observations and theoretical modeling to inform our choices for constructing a training dataset.

The stellar–mass—halo–mass (SHMR) has been directly constrained out to redshifts $z \lesssim 0.05$ and galaxy masses $M_{\rm vir} > 10^{12} M_\odot$ using galaxy kinematics [e.g. 188, 158], X-ray observations [e.g 160, 147] and gravitational lensing [e.g. 172, 273]. These constraints can be extended to higher redshifts ($z < 10$) and lower masses ($M_{\rm vir} < 10^{10}$) by including halo–galaxy connection modeling [e.g. 194, 62, 223, 19, 91]. Such models consistently infer that the average stellar mass of galaxies increases with halo mass.

At fixed redshift and halo mass, average galaxy masses of central galaxies differ from satellite galaxies. Halos grow through hierarchical merging, in which small halos merge to form larger halos. As subhalos merge into larger halos, tidal heating and stripping reduce the mass of the more extended dark matter halo, while the satellite galaxy mass remains largely unaffected. For this reason, galaxy mass often correlates better with halo properties at the time of accretion than the current halo mass [e.g. 55, 265, 192, 213]. In particular, SHAM models find

that using the halo peak circular velocity, $v_{\rm peak}$, to assign galaxy masss and/or luminosity best reproduces observed galaxy clustering [e.g. 213, 106, 155].

Star formation rates correlate tightly with galaxy masses, and increase with redshift at fixed stellar mass [e.g. 195, 247, 35]. While these trends hold on average, there is a distinct bimodal distribution in the star formation rates of galaxies, corresponding to star-forming and quiescent populations [e.g. 16]. The observed fraction of quiescent galaxies increases as the Universe evolves [e.g. 256], with the interpretation that some mechanism turns off star formation in galaxies. Many quenching mechanisms have been proposed, including secular/mass quenching [e.g. 134, 59] and environmental quenching [e.g. 66, 260]. Which of these processes dominate may vary with redshift [129].

Overdense environments may cause environmental quenching, by providing close pairs that can suppress gas accretion ("strangulation"), removing gas through ram-pressure stripping, or disrupting by interactions with other galaxies ("harassment"). Environment thereby influences star formation rates, and low-mass satellite galaxies are typically the most prone to environmental quenching [e.g. 67].

Given these established trends, galaxy mass and star formation rate may depend on redshift, halo mass, peak circular velocity, and environmental properties. We will therefore select corresponding parameters from the CROC simulated galaxy catalogs to provide our dataset for training the EBM models. Details of the CROC simulations can be found in [92]. At a range of redshifts z during the simulation, the computational grid and particle properties are written to disk. These simulation snapshots are post-processed to identify virialized galaxies, as described in Zhu et al. [300], and the properties of the simulated galaxies are recorded in catalogs. Merger trees are used to identify the properties of simulated

galaxies across redshift.

For our target quantities y, in this chapter we will model stellar mass $M_\star$ $[h^{-1}M_\odot]$ and star formation rate SFR $[M_\odot\ \mathrm{yr}^{-1}]$. The parameters $\boldsymbol{\theta}$ selected from the simulated catalog include both intrinsic properties of galaxies and extrinsic properties set by the large scale environment. For intrinsic properties we include the galaxy virial mass $M_{\rm vir}$ $[h^{-1}M_\odot]$, the redshift z at which the simulated galaxy properties were measured, and the maximum peak circular velocity $v_{\rm peak}$ $[\mathrm{km\ s}^{-1}]$ measured over the formation history of each galaxy. The extrinsic properties used are defined by a length scale R measured relative to each simulated galaxy. We follow convention and substitute R with a numerical value that indicates a number of comoving Mpc (e.g., σ_8 is the rms density fluctuations measured in spheres of radius of $R = 8$Mpc). We compute an environmental density $\rho_1 \equiv 1 + \Delta_1$, where Δ_1 is the dimensionless matter overdensity measured within 1 Mpc. We include an environmental gas temperature T_1 [K] averaged on 1 Mpc scales. From each simulated galaxy we also find the virial mass $M_{\rm max,0.1}$ of the most massive neighboring halo within 100 kpc. We then define the mass ratio $\Upsilon_{0.1} \equiv 1 + M_{\rm max,0.1}/M_{\rm vir}$

The simulated galaxy catalogs include roughly 8,426,327 objects covering a wide range of halo masses, stellar masses, star formation rates, redshifts, and other extrinsic properties. From the catalog of simulated galaxies, objects with a $SFR < 0.001\ M_\odot\ \mathrm{yr}^{-1}$ were excluded owing to resolution effects artificially limiting their star formation rates. After this culling, the catalog contained 5,950,357 objects that formed our dataset. At this stage, we constructed the training and test datasets from our catalog using the parameter vector $\boldsymbol{\theta} = [M_{\rm vir}, z, v_{\rm peak}, \rho_1, T_1, \Upsilon_{0.1}]$ to model the target quantities $\boldsymbol{y} = [M_\star, SFR]$. We use k-fold cross-validation [102] with $k = 5$, such that the test/training split is 20%/80%

for each k-folding.

5.2.3 Training Procedure

The calculations presented in this chapter leverage the InterpretML [196] implementation of EBMs, using the hyperparameters in Table 5.1. These InterpretML hyperparameters control the number of bins in the piece-wise f_y^i and f_y^{ij} functions ($\mathcal{Q}_{\rm max}$, $\mathcal{Q}_{\rm max,2D}$), the distribution of bins across the fitted domain ($\mathcal{B}$), and the learning rate of the optimization scheme ($\mathcal{R}_l$). The Nori et al. [196] implementation trains an EBM in two phases. First, the univariate functions are optimized using a gradient boosting approach applied round-robin on each parameter, as detailed in Lou et al. [165]. After the univariate functions have converged, the interaction terms are computed and the bivariate functions are optimized according the GA2M/FAST algorithms detailed in Lou et al. [166]. During training we use k-fold cross-validation, and merge the training and test datasets for the final performance evaluation of the model.

We evaluate the EBM performance using the mean absolute error (MAE), a variance metric r^2, and the total outlier fraction ζ_k. These statistics provide measures of how well the EBM reproduces the mean trends in the target quantities y as a function of the features θ, the width of the distribution about the mean trends in the training data, and the tails of that distribution.

We calculate the MAE of the model applied to the simulated galaxy sample as

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=0}^{N-1} |y_i - \hat{y}_i|, \tag{5.2}$$

where N is the number of objects, y_i is the true value of the target quantity for object i, and $\hat{y}_i$ is the predicted value from the model for object i.

We compute the $r^2 \in [0, 1]$ variance metric as

$$r^2 = 1 - \frac{\sum_{i=0}^{N-1}(y_i - \hat{y}_i)^2}{\sum_{i=0}^{N-1}(y_i - \overline{y})^2}, \tag{5.3}$$

which provides a measure of how well the model captures the variance in the data relative to the mean $\overline{y}$, with $r^2 = 1$ reflecting a perfect reproduction of the distribution of y in the training dataset. Note that the feature and interaction functions f_y^i and f_y^{ij} have a finite range, and thus not all values y_i can be represented by Equation 5.1 even when the input parameters $\boldsymbol{\theta}$ vary about the mean trends with halo mass or environment. Hence, even for high quality EBM models $r^2 < 1$ and we expect outliers. The ζ_k metric represents the fraction of the total dataset that lies outside the range of predicted values, $\{\hat{y}\}$, as a function of one of the features θ_k. We define

$$\zeta_k = \frac{1}{N}\sum_{i=0}^{N-1} g_{k,i}(y_i, \theta_{k,i}) \tag{5.4}$$

where the index i runs over the total number of samples N and $g_{k,i}(y_i, \theta_{k,i})$ is a function that returns 1 if the true target quantity for object i lies outside the predicted range, i.e., $y_i \notin \{\hat{y}\}$. In practice, we compute the outlier fraction for feature $k = \log_{10} M_{\rm vir}$, and use 2D histograms of $(y_i,\, \theta_{k,i})$ and $(\hat{y}_i,\, \theta_{k,i})$ to calculate $g_{k,i}$.

In Table 5.2 we present the evaluation metrics for our EBM model fully trained on the simulated galaxy catalog. For the EBM model for star formation rate ($y = SFR$), we find a MAE $\sim 0.14 \log_{10} M_\odot$ yr^{-1}, a variance metric $r^2 \sim 0.9$, and an outlier fraction of $< 3\%$. For the EBM model for stellar mass ($y = M_\star$), we report a MAE $\sim 0.19 \log_{10} M_\odot$ yr^{-1}, a variance metric $r^2 \sim 0.88$, and an outlier fraction of $< 1\%$. The good performance of the EBM models in these metrics reflects the abililty of the EBMs to capture both the mean trends and full

EBM Training Hyperparameters	
Hyperparameter	Value
Binning $\mathcal{B}$	"uniform"
Maximum Bins, Univariate $\mathcal{Q}_{\max}$	256
Maximum Bins, Bivariate $\mathcal{Q}_{\max,2D}$	32×32
Learning Rate $\mathcal{R}_l$	0.01

Table 5.1: Hyperparameters used to train the InterpretML [196] implementation of the EBM. All other model hyperparameters were set to the default values for InterpretML version 0.2.7.

EBM Training Results		
Metrics	$\gamma(SFR\|\boldsymbol{\theta})$	$\gamma(M_\star\|\boldsymbol{\theta})$
r^2	0.898 ± 0.0003	0.882 ± 0.0001
ζ	0.029 ± 0.004	0.008 ± 0.0010
	$\log_{10} SFR\ [M_\odot yr^{-1}]$	$\log_{10} M_\star\ [M_\odot]$
MAE	0.144 ± 0.0001	0.189 ± 0.0001

Table 5.2: Training results for the EBM using k-fold cross validation. See Section 5.2.3 for more information on the training process. Reported are values for the variance metric r^2, the outlier fraction ζ, and the mean absolute error (MAE). Uncertainties are computed from the variation among the k-fold trials.

distributions of the target quantities $y = [M_\star, SFR]$ in the training set given the parameters $\boldsymbol{\theta} = [M_{\rm vir}, z, v_{\rm peak}, \rho_1, T_1, \Upsilon_{0.1}]$. We describe the detailed model results in Section 5.3.

5.3 Results

After training the EBM model to reproduce the dependence of the target quantities $M_\star$ and SFR on the parameters $\boldsymbol{\theta}$, the relationships between the target quantities and the parameters can be analyzed. Below, we provide several analyses that quantify how the target quantities relate to the parameters and illustrate the performance of the EBM for our astrophysical applications.

5.3.1 Average Contribution

A key advantage of using EBM models over "black box" models (e.g., neural networks) is their clear interpretability (see Section 5.2.1). The contribution of each parameter θ_i to the model of the target quantity y is provided by the output functions f_y^i and f_y^{ij}.

Since these functions are vectors or two-dimensional matrices with a number of elements equal to the number of bins n_b in the piece-wise function (see Table 5.1), a summary scalar quantity for each feature function is helpful for comparing their relative importance. We can define the *average contribution* $\bar{f}_y^i$ that provides the average absolute value of f_y^i or f_y^{ij}, with the average computed over the number of bins n_b and weighted by the number of samples in each bin. Mathematically, we can write

$$\bar{f}_y^i = \frac{\sum_{j=0}^{n_b-1} |f(\theta_{i,j})| N_j}{\sum_{j=0}^{n_b-1} N_j} \tag{5.5}$$

where f is the feature function being averaged (f_y^i or f_y^{ij} from Equation 5.1), $\theta_{i,j}$ is value of the parameter θ_i in the jth bin, and N_j is the number of samples in bin j. Intuitively, the average contribution $\bar{f}_y^i$ summarizes the importance of each parameter θ_i for determining the target quantities when averaged over the samples in the final, merged dataset.

The average contributions of each feature (f_y^i) or combination of features (f_y^{ij}) are computed from the EBM. In each case, we rank order the features by decreasing average contribution and focus on the seven features or feature combinations with the largest average contribution. In each case the most important feature has an average contribution more than an order of magnitude larger than the seventh-ranked feature.

EBM Model Targeting Star Formation Rate *SFR*

Figure 5.1 shows the average contribution of the top seven features for the EBM model targeting star formation rate $\log_{10} SFR$. In decreasing order, the seven most important features in determining *SFR* are maximum peak circular velocity $v_{\rm peak}$, virial mass $M_{\rm vir}$, environmental density ρ_1, redshift z, environmental temperature T_1, mass ratio of nearby halos $\Upsilon_{0.1}$, and the interaction between $M_{\rm vir}$ and $\Upsilon_{0.1}$. The numerical values for the average contributions are provided in Table 5.3. The baseline value of *SFR* is $\beta_{\log_{10} SFR} = -2.1151$ $[\log_{10} M_\odot\ {\rm yr}^{-1}]$, typical of halos with $\log_{10} M_{\rm vir} \sim 9$. The average contribution of $v_{\rm peak}$ and $M_{\rm vir}$ are quite similar, providing $\Delta \log_{10} SFR > 0.2$ on average, but their interaction term is small with $\bar{f}(\log_{10} v_{\rm peak}, \log_{10} M_{\rm vir}) \ll 0.01$. Therefore peak circular velocity and virial mass provide important contributions to determining the star formation rate, and the univariate dependence of the *SFR* on these properties accounts for most of their contribution. At the few-percent level, environmental density, redshift, environmental gas temperature, and the presence of nearby massive halos also contribute.

The feature functions f_y^i for each feature are plotted in Figure 5.2. The functions indicate that there are positive correlations between the star formation rate $\log_{10} SFR$ and either the peak circular velocity $v_{\rm peak}$, virial mass $M_{\rm vir}$, or environmental density ρ_1. The star formation rate increases with increasing environmental temperature $\log_{10} T_1$, but near $T_1 \approx 10^4$K the univariate function shows an enhancement just as hydrogen becomes mostly neutral and a deficit near the temperature at which it becomes ionized. Star formation rate increases with decreasing redshift over the range $z \sim 5 - 15$, becoming more efficient after reionization.

The interaction functions f_y^{ij} learned by the EBM $\gamma(SFR|\boldsymbol{\theta})$ targeting the star

Average Contributions for the $\gamma(M_\star\|\boldsymbol{\theta})$ EBM	
Feature	Value $[\log_{10} M_\odot \text{ yr}^{-1}]$
$\beta_{\log_{10} M_\star}$	-2.1151
$\bar{f}(\log_{10} v_{\text{peak}})$	0.2380
$\bar{f}(\log_{10} M_{\text{vir}})$	0.2224
$\bar{f}(\log_{10} \rho_1)$	0.0475
$\bar{f}(z)$	0.0343
$\bar{f}(\log_{10} T_1)$	0.0252
$\bar{f}(\log_{10} \Upsilon_{0.1})$	0.0202
$\bar{f}(\log_{10} M_{\text{vir}}, \log_{10} \Upsilon_{0.1})$	0.0052

Table 5.3: Summary of the EBM model trained to predict *SFR*. The first entry, $\beta_{\log_{10} SFR}$, is the baseline value learned model (see Section 5.2.1). The next seven entries are the average contributions of the most important feature functions listed in descending order (see Equation 5.5).

formation rate *SFR* are plotted as "heat maps" in Figure 5.3. Most interaction functions do not contribute significantly to the star formation rate, and change the star formation rate by $\Delta \log_{10} SFR \lesssim 0.05$. However, halos with low neighboring halo mass ratios $\Upsilon_{0.1}$ and large peak circular velocity v_{peak} have their star formation rate enhanced by $\Delta \log_{10} SFR \approx 0.15$. Rephrased, locally dominant halos with large peak circular velocity show enhanced star formation. Such enhancements likely owe to recent merger activity.

While Equation 5.1 represents a complex, multidimensional manifold that provides the *SFR* as a function of the parameters $\boldsymbol{\theta}$, the distributions of simulated and predicted *SFR* as a function of a single parameter provide a graphical summary of the EBM model performance. Figure 5.4 shows the simulated and predicted *SFR* as a function of virial mass $\log_{10} M_{\text{vir}}$, and we will refer to this figure as the *model summary*. Shown in this model summary are the distributions of *SFR* in the CROC simulated galaxy catalogs with virial mass and the *SFR* predicted by the EBM model $\gamma(SFR|\boldsymbol{\theta})$ using the parameters $\boldsymbol{\theta}$ measured for each simulated galaxy. The EBM model captures roughly 97% of the simulated distribution of

SFR with virial mass. The EBM model is highly predictive of the simulated connection between *SFR* and the intrinsic and extrinsic properties $\boldsymbol{\theta}$.

EBM Model Targeting Stellar Mass $M_\star$

An EBM model $\gamma(M_\star|\boldsymbol{\theta})$ targeting stellar mass $M_\star$ using the properties $\boldsymbol{\theta}$ can be constructed through simple retraining. Using the simulated galaxy catalogs from CROC, we retrain the EBM to model $M_\star$ against $\boldsymbol{\theta}$. For reference, the baseline value of $M_\star$ is $\beta_{\log_{10} M_\star} = 5.9629$ $[\log_{10} M_\odot \text{ yr}^{-1}]$ (see Table 5.4), typical of halos with $\log_{10} M_{\text{vir}}/M_\odot \sim 9$.

Figure 5.5 shows the model summary for the EBM model $\gamma(M_\star|\boldsymbol{\theta})$. The EBM model provides an excellent representation of the distribution of stellar masses for the CROC simulated galaxy catalog. As the lower right panel of Figure 5.5 indicates, the $\gamma(M_\star|\boldsymbol{\theta})$ model results in few outliers for the CROC simulated galaxies and has an outlier fraction of $\lesssim 1\%$. Given the galaxy properties $\boldsymbol{\theta} = [M_{\text{vir}}, z, v_{\text{peak}}, \rho_1, T_1, \Upsilon_{0.1}]$, the distribution of stellar masses for CROC simulated galaxies can be recovered to 99% accuracy.

Figure 5.7 shows the average contribution of the seven most important features and interactions in the EBM model $\gamma(M_\star|\boldsymbol{\theta})$. In order of decreasing importance, these features include peak circular velocity, virial mass, redshift, environmental density, environmental temperature, the mass ratio of nearby halos, and the interaction between redshift and peak circular velocity. Peak circular velocity is about 50% more important than virial mass, which in turn is roughly a factor of two more important than redshift. The other features and interactions contribute to stellar mass at the $\lesssim 0.1$ dex level. For reference the numerical values for the average contributions are provided in Table 5.4.

The univariate functions determined by the EBM targeting stellar mass $M_\star$ are

Average Contributions for the $\gamma(M_\star\|\boldsymbol{\theta})$ EBM	
Feature	Value $[\log_{10} M_\odot]$
$\beta_{\log_{10} M_\star}$	5.9629
$\bar{f}(\log_{10} v_{\rm peak})$	0.3284
$\bar{f}(\log_{10} M_{\rm vir})$	0.2123
$\bar{f}(z)$	0.1238
$\bar{f}(\log_{10} \rho_1)$	0.0722
$\bar{f}(\log_{10} T_1)$	0.0545
$\bar{f}(\log_{10} \Upsilon_{0.1})$	0.0359
$\bar{f}(z, \log_{10} v_{\rm peak})$	0.0135

Table 5.4: Summary of the EBM model $\gamma(M_\star|\boldsymbol{\theta})$ trained to predict $M_\star$ as a function of the full parameter set $\boldsymbol{\theta}$. The first entry, $\beta_{\log_{10} M_\star}$, is the learned baseline value of the model (see Section 5.2.1). The next seven entries are the feature functions with the highest average contribution in descending order. Average contribution is calculated using the average of the absolute value of the feature functions weighted by the number of samples in each bin (see Equation 5.5).

shown in Figure 5.6. Stellar mass increases with increasing peak circular velocity, virial mass, environmental density, and neighboring halo mass ratio. Stellar mass increases with decreasing redshift. As with star formation rate, the stellar mass increases with increasing environmental temperature T_1, with a sharp enhancement near the temperature where hydrogen becomes neutral and a sharp deficit near where hydrogen ionizes.

The bivariate interaction functions f_y^{ij} (see Equation 5.1) learned by the EBM when targeting stellar mass $M_\star$ are plotted as heat maps in Figure 5.8. On average most interaction functions do not contribute significantly to galaxy stellar mass, but there are regions of parameter space where the interaction functions are important. For instance, halos with low environmental temperatures and high environmental densities have suppressed stellar mass. Large virial mass halos with small neighboring halo mass ratios $\log_{10} \Upsilon_{0.1}$, indicating halos that dominate their local environment, have stellar mass enhanced by ≈ 0.3 dex. This effect

exceeds the maximum univariate contribution of $\log_{10} \Upsilon_{0.1}$ alone. The deficit of stellar mass at environmental temperatures where hydrogen is becoming ionized is increased at high redshifts.

5.4 Composite EBMs for Restricted Parameter Sets

The EBM models $\gamma(\mathit{SFR}|\boldsymbol{\theta})$ and $\gamma(M_\star|\boldsymbol{\theta})$ presented in Section 5.3.1 are constructed using the parameter set $\boldsymbol{\theta} = [M_{\rm vir}, z, v_{\rm peak}, \rho_1, T_1, \Upsilon_{0.1}]$. Our results show that the full distribution of *SFR* and stellar mass in the simulated CROC galaxy catalogs can be recovered accurately with only $\approx 1-3\%$ outliers. These EBM models can therefore be applied to cosmological simulations using the parameters $\boldsymbol{\theta}$ measured from simulated galaxy catalogs to recover the distribution of *SFR* and stellar mass computed by CROC.

The parameters $\boldsymbol{\theta}$ include the peak circular velocity $v_{\rm peak}$, which requires both time-dependent tracking of formation histories for individual galaxies and high spatial resolution to capture the peak of the rotation curve for each object. As a result, as expressed above the models $\gamma(\mathit{SFR}|\boldsymbol{\theta})$ and $\gamma(M_\star|\boldsymbol{\theta})$ cannot be applied directly to cosmological simulations with low spatial resolution or without merger trees to capture formation histories.

Instead of fitting EBM models using the full parameter set $\boldsymbol{\theta}$, consider the construction of an EBM model using the restricted parameter set $\boldsymbol{\theta'} = [M_{\rm vir}, z, \rho_1, T_1, \Upsilon_{0.1}]$ that does not include $v_{\rm peak}$. The parameters $\boldsymbol{\theta'}$ can all be measured directly in cosmological simulations with sufficient resolution to capture individual galaxy-mass halos without the need to track merger trees. The EBM models $\gamma(\mathit{SFR}|\boldsymbol{\theta'})$ and $\gamma(M_\star|\boldsymbol{\theta'})$ using the restricted parameter set $\boldsymbol{\theta'}$ perform substantially less well

than the models $\gamma(SFR|\boldsymbol{\theta})$ and $\gamma(M_\star|\boldsymbol{\theta})$ trained on the full parameter set $\boldsymbol{\theta}$ that includes $v_{\rm peak}$. With the restricted parameter set $\boldsymbol{\theta'}$, the EBM model shows 7.6% outliers when targeting *SFR* and 2.8% when targeting $M_\star$. Comparing with the outlier fractions reported in Table 5.2 for the full parameter set including $v_{\rm peak}$, the EBM model trained on the restricted dataset has degraded its performance by a factor of $\sim 2-3$.

To ameliorate the poorer performance of the EBM models trained on restricted parameter sets, we use a *Composite EBM* (CEBM) model. Given a target quantity y and a parameter set $\boldsymbol{\theta'}$, we fit a *base* EBM $\gamma(y|\boldsymbol{\theta'})$ in the same manner as fitting the EBMs $\gamma(SFR|\boldsymbol{\theta})$ or $\gamma(M_\star|\boldsymbol{\theta})$. We construct a dataset from the galaxies whose y values lie outside the predictions from $\gamma(y|\boldsymbol{\theta'})$, and then fit an *outlier* EBM $\delta(y|\boldsymbol{\theta'})$ to these discrepant samples. We then weight the base and outlier EBMs to construct the CEBM model $\Gamma(M_\star|\boldsymbol{\theta'})$ using a *classifier* EBM $\phi_y(\boldsymbol{\theta'})$. Instead of fitting the change in star formation rate or stellar mass at a given sample in θ', the classifier EBM fits the log odds that a given sample in θ' is an outlier. We then define $\phi_y(\boldsymbol{\theta'})$ to be the sigmoid of these log odds, such that $\phi_y(\boldsymbol{\theta'}) \in [0,1]$. The CEBM can then be written as

$$\Gamma(M_\star|\boldsymbol{\theta'}) = [1-\phi_y(\boldsymbol{\theta'})]\gamma(y|\boldsymbol{\theta'}) + \phi_y(\boldsymbol{\theta'})\delta(y|\boldsymbol{\theta'}). \tag{5.6}$$

5.4.1 CEBM Feature and Interaction Functions

The feature functions of a single EBM are univariate and indicate directly how the expectation value of the targeted quantity depends on each parameter $\theta_i \in \boldsymbol{\theta}$. With a CEBM comprised of a weighted sum of two base EBMs, we define the analog of the feature function to be the weighted sum of the base EBM feature

functions. We can write that

$$\tilde{f}_y^i = \frac{1}{N} \sum_{j=0}^{N} \|\phi(\boldsymbol{\theta}_j) \odot \boldsymbol{f}_y^i(\boldsymbol{\theta}_j)\|_1, \tag{5.7}$$

where $\odot$ is the Hadamard or element-wise product operation and the sum is over the number of samples N. The quantity $\boldsymbol{f}_y^i$ is the vector of the individual EBM feature functions f_y^i. While the base EBM feature functions are individually univariate, by weighting the sum of these feature functions with the classifier EBM the resulting feature function analog in Equation 5.7 is *not* univariate.

The interaction functions $\tilde{f}_y^{ij}$ are defined as in Equation 5.7 but with the vector of the individual EBM interaction functions $\boldsymbol{f}_y^{ij}$ subsituted for $\boldsymbol{f}_y^i$. While the interaction functions for a single EBM are bivariate, the CEBM interaction functions are *not* bivariate.

5.4.2 CEBM Average Contribution

The average contribution of each feature in a CEBM can be defined in a manner analogous to the average contribution computed for a single EBM (Equation 5.5). The CEBM average contribution can be written as

$$\bar{f}_y^i = \frac{\sum_{j=0}^{n_b-1} \tilde{f}(\theta_{i,j}) N_j}{\sum_{j=0}^{n_b-1} N_j} \tag{5.8}$$

where $\tilde{f}$ is either the CEBM feature function $\tilde{f}_y^i$ or the CEBM interaction function $\tilde{f}_y^{ij}$. Equation 5.8 characterizes how important the parameter θ_i is for modeling the target quantity y.

5.4.3 Visualizing CEBM Feature and Interaction Functions

The feature and interaction functions $\tilde{f}_y^i$ and $\tilde{f}_y^{ij}$ are not univariate or bivariate by design, which allows them to model the outlier distribution about the base EBM model $\gamma(y|\boldsymbol{\theta}')$. To visualize the feature and interaction functions for CEBM models in a manner similar to the univariate feature and bivariate interaction functions for single EBM, we can average the values of $\boldsymbol{f}_y^i$ and $\boldsymbol{f}_y^{ij}$. For the feature function averaged over N samples, consider n_b bins along the θ_i direction, with central values $\theta_{i,b}$ and bin widths $\Delta\theta_{i,b}$. The bin-averaged CEBM feature and interaction functions are then

$$f_y^{i,b} = \frac{1}{N}\sum_{j=0}^{N-1} \alpha(\theta_{i,b}, \Delta\theta_{i,b}, \theta_{j,i})\phi(\boldsymbol{\theta}_j) \odot \boldsymbol{f}(\boldsymbol{\theta}_j) \tag{5.9}$$

where $\theta_{j,i}$ is the ith parameter of the jth sample $\boldsymbol{\theta}_j$ and the function $\alpha(\theta_{i,b}, \Delta\theta_{i,b}, \theta_{j,i}) =$ 1 if $\theta_{i,b} - \Delta\theta_{i,b}/2 \leq \theta_{j,i} \leq \theta_{i,b} + \Delta\theta_{i,b}/2$ and $\alpha = 0$ otherwise. The quantity $\boldsymbol{f}$ is either the vector of EBM feature functions $\boldsymbol{f}_y^i$ or the EBM interaction functions $\boldsymbol{f}_y^{ij}$. Equation 5.9 calculates the mean of the $\boldsymbol{f}$ values in each of the n_b bins, and can be modified to calculate its standard deviation.

5.5 Composite EBM Model for Star Formation Rate

The CEBM model $\Gamma(SFR|\boldsymbol{\theta}')$ for the star formation rate consists of a base EBM $\gamma(SFR|\boldsymbol{\theta}')$, a residual EBM $\delta(SFR|\boldsymbol{\theta}')$ that attempts to capture the outlying values of SFR not recovered by $\gamma(SFR|\boldsymbol{\theta}')$, and the classifier EBM $\phi_{SFR}(\boldsymbol{\theta}')$. For each of these individual EBMs that form the CEBM model, we plot the average

contribution, feature functions, and interaction functions.

Figure 5.9 shows the average contribution, feature functions, and interaction functions for the EBM model $\gamma(SFR|\boldsymbol{\theta}')$ that forms the base of the CEBM model. The differences between $\gamma(SFR|\boldsymbol{\theta})$ and $\gamma(SFR|\boldsymbol{\theta}')$ reflect the additional information provided by the maximum peak circular velocity $v_{\rm peak}$. Without access to $v_{\rm peak}$, the base EBM $\gamma(SFR|\boldsymbol{\theta}')$ upweights $\bar{f}(M_{\rm vir})$ such that its importance roughly equals the combined importance of $M_{\rm vir}$ and $v_{\rm peak}$ in determining $\gamma(SFR|\boldsymbol{\theta})$. The average contribution of ρ_1, T_1, z, $\Upsilon_{0.1}$, and $(M_{\rm vir}, \Upsilon_{0.1})$ are similar between the models. The additional interaction term in the top seven average contributions is (z, ρ_1), with a percent-level contribution to *SFR* relative to $M_{\rm vir}$. The feature functions for $\gamma(SFR|\boldsymbol{\theta}')$ have shapes similar to the feature functions for $\gamma(SFR|\boldsymbol{\theta})$, but their minimum and maximum contributions to *SFR* are adjusted to account for the missing $v_{\rm peak}$ contribution. The feature function $\bar{f}(z)$ is noisier overall. For the interaction functions, the largest contributors now involve $M_{\rm vir}$ rather than the missing parameter $v_{\rm peak}$, and the set of available functions is substantially different than with $\gamma(SFR|\boldsymbol{\theta})$.

Figure 5.10 shows the average contribution, feature functions, and interaction functions for the outlier EBM $\delta(SFR|\boldsymbol{\theta}')$ fit to the deviant samples not captured by the base EBM $\gamma(SFR|\boldsymbol{\theta}')$. The outlier EBM receives the highest contribution from virial mass, with an average contribution more than an order of magnitude larger than the next most important feature ρ_1. The redshift z and environmental temperature T_1 have comparable importance to ρ_1. The remaining features provide only percent-level contributions relative to $M_{\rm vir}$.

Figure 5.11 shows the average contribution, feature functions, and interaction functions for the classifier EBM $\delta(SFR|\boldsymbol{\theta}')$ that interpolates between the base and outlier EBMs when calculating the CEBM model. For the classifier EBM,

the most important features are ρ_1, $M_{\rm vir}$, and $\Upsilon_{0.1}$. Redshift z has middling importance, following by T_1, $[\rho_1, \Upsilon_{0.1}]$, and $[M_{\rm vir}, \rho_1]$. The feature functions show strong dependencies on ρ_1, $M_{\rm vir}$, $\Upsilon_{0.1}$, z, and T_1. The largest interaction functions involve the environmental temperature T_1, redshift z, and virial mass $M_{\rm vir}$.

By weighting the base and outlier EBM models with the classifier EBM, we construct the CEBM for star formation rate as $\Gamma(SFR|\boldsymbol{\theta}') \equiv [1-\phi_{SFR}(\boldsymbol{\theta}')]\gamma(SFR|\boldsymbol{\theta}')+\phi_{SFR}(\boldsymbol{\theta}')\delta(SFR|\boldsymbol{\theta}')$. Figures 5.12 show the average contribution, feature functions, and interaction functions for the *SFR* CEBM. The most important feature is $M_{\rm vir}$, which dominates by a factor of $\sim 4-10$ over environmental density ρ_1, environmental temperature T_1, $\Upsilon_{0.1}$, and redshift z. The interaction terms are roughly percent-level effects relative to $M_{\rm vir}$. The feature functions show a strongly increasing *SFR* with $M_{\rm vir}$, and enhanced *SFR* with environmental density ρ_1. The temperature dependence shows the feature at $\log_{10} T_1 \approx 4$ seen with the EBM model $\gamma(SFR|\boldsymbol{\theta})$. The interaction functions provide only important contributions over very limited areas of parameter space, with the most important adjustments occuring at low redshift and large virial mass, or for large temperatures and virial masses. For reference, the model summary Figure 5.13 illustrates the overall performance of the model.

Table 5.5 lists the evaluation metrics for the training of CEBM models targeting *SFR* and stellar mass without using $v_{\rm peak}$. The outlier fraction has improved to $\approx 5\%$ for CEBM model $\Gamma(SFR|\boldsymbol{\theta}')$ and to $\lesssim 2\%$ for $\Gamma(M_\star|\boldsymbol{\theta}')$. The average parameter contributions and baseline value $\beta_{\log_{10} SFR}$ from $\Gamma(SFR|\boldsymbol{\theta}')$ are provided in Table 5.6 and for the CEBM targeting stellar mass in Table 5.7.

Figure 5.13 shows the model summary for the CEBM targeting *SFR*, and Figure 5.14 shows the model summary for the CEBM targeting stellar mass. As both models demonstrate, the CEBM model accurately recovers the distribution

Composite EBM Training Results		
Metrics	$\gamma(SFR\|\boldsymbol{\theta})$	$\gamma(M_\star\|\boldsymbol{\theta})$
r^2	0.868 ± 0.0002	0.830 ± 0.0003
ζ	0.052 ± 0.0053	0.018 ± 0.0031
	$\log_{10} SFR\ [M_\odot yr^{-1}]$	$\log_{10} M_\star\ [M_\odot]$
MAE	0.165 ± 0.0001	0.233 ± 0.0002

Table 5.5: Training results for CEBM models for *SFR* and $M_\star$ using k-fold cross validation. See Section 5.2.3 for more information on the training process. Reported are values for the variance metric r^2, the outlier fraction ζ, and the mean absolute error (MAE). Uncertainties are computed from the variation among the k-fold trials.

of star formation rate and stellar mass in the CROC simulated galaxy sample. Between the models, the outlier fraction is only $\approx 2-5\%$ despite using the restricted set of parameters $\boldsymbol{\theta'}$ that does not include $v_{\rm peak}$ or any time-dependent tracking of individual systems.

5.6 Composite EBM Model for Stellar Mass

The CEBM model $\Gamma(M_\star|\boldsymbol{\theta'})$ for stellar mass is comprised of a base EBM $\gamma(M_\star|\boldsymbol{\theta'})$, an outlier EBM that attempts to model the $M_\star$ of samples not recovered by $\gamma(M_\star|\boldsymbol{\theta'})$, and the classifier EBM function $\phi_{M_\star}(\boldsymbol{\theta'})$ that interpolates between them. The average contribution, feature functions, and interaction functions from these component EBM models are presented below.

Figures 5.15 shows the average contribution, feature functions, and interaction functions for the base EBM model $\gamma(M_\star|\boldsymbol{\theta'})$. By removing $v_{\rm peak}$ from the dataset used to train the EBM, the base EBM model for the $M_\star$ CEBM replaces the dependence on $v_{\rm peak}$ with an additional dependence on $M_{\rm vir}$. The relative ordering and importance of redshift z, environmental density ρ_1, environmental tempera-

Overview of CEBM $\Gamma(SFR|\boldsymbol{\theta}')$

Feature	Value [$\log_{10} M_\odot\ \mathrm{yr}^{-1}$]
$\beta_{\log_{10} SFR}$	-1.7466
$\tilde{f}(\log_{10} M_{\rm vir})$	0.4327
$\tilde{f}(\log_{10} \rho_1)$	0.0625
$\tilde{f}(\log_{10} T_1)$	0.0327
$\tilde{f}(\log_{10} \Upsilon_{0.1})$	0.0215
$\tilde{f}(z)$	0.0190
$\tilde{f}(z, \log_{10} \rho_1)$	0.0077
$\tilde{f}(\log_{10} M_{\rm vir}, \log_{10} \Upsilon_{0.1})$	0.0056

Table 5.6: Average contribution to the CEBM model $\Gamma(SFR|\boldsymbol{\theta}')$ trained to predict *SFR* from the parameter set $\boldsymbol{\theta}'$. The first entry, $\beta_{\log_{10} SFR}$, is the learned baseline of the model. The next seven entries are the feature functions with the highest average contribution listed in descending order. The average contribution is calculated using the average of the absolute value of the base EBM function values weighted by the number of samples in each bin and the output of the classification EBM for each sample (see Section 5.4.2 for more details).

ture T_1, and $\Upsilon_{0.1}$ are approximately maintained. For the feature functions, the results shown for $\gamma(M_\star|\boldsymbol{\theta}')$ in Panel b) of Figure 5.15 can be compared with the results for $\gamma(M_\star|\boldsymbol{\theta})$ shown in Figure 5.7. As reflected by average contributions, the amplitude of the feature function $\bar{f}(M_{\rm vir})$ increases to account for the removal of $v_{\rm peak}$. The feature functions for z, ρ_1, T_1, and $\Upsilon_{0.1}$ are modified and remain similar in shape to those computed for the EBM $\gamma(M_\star|\boldsymbol{\theta})$. The interaction functions shared between $\gamma(M_\star|\boldsymbol{\theta}')$ and $\gamma(M_\star|\boldsymbol{\theta})$ are similar. There is an increase in $M_\star$ contribution for large $[M_{\rm vir}, T_1]$ and a decrease in the amplitude of $[M_{\rm vir}, \Upsilon_{0.1}]$.

Figure 5.16 shows the average contribution, feature functions, and interaction functions for the outlier EBM model $\delta(M_\star|\boldsymbol{\theta}')$. The average contribution is dominated by $M_{\rm vir}$, with the contributions from all other single parameters lower by a factor of ≈ 10 with the order of importance maintained relative to $\gamma(M_\star|\boldsymbol{\theta}')$. For the feature functions, the redshift dependence changes dramatically and now increases with increasing redshift. The feature function for environmental density

$\tilde{f}(\rho_1)$ becomes much weaker over a wide range of ρ_1, but increases dramatically at high ρ_1. Relative to the $\gamma(M_\star|\boldsymbol{\theta}')$ feature functions, the feature function $\tilde{f}(\Upsilon_{0.1})$ for $\delta(M_\star|\boldsymbol{\theta}')$ is weak and noisy. The interaction functions show increased contributions at large $[z, \rho_1]$, and for low T_1 and large ρ_1.

Figure 5.17 shows the average contribution, feature functions, and interaction functions for the classifier EBM $\phi_{M_\star}(\boldsymbol{\theta}')$. For each of these properties, we note that in determining $\phi_{M_\star}(\boldsymbol{\theta}')$ a sigmoid function σ is applied to the sum of β, f_y^i and f_y^{ij} that model the log odds that a galaxy is an outlier in stellar mass. In determining $M_\star$, the features with the largest average contribution are environmental density ρ_1, redshift z, $\Upsilon_{0.1}$, and virial mass $M_{\rm vir}$. The interaction terms with the largest contribution are (z, ρ_1) and (ρ_1, T_1). Clearly, environmental density plays an important role in determining whether a given simulated galaxy is an outlier relative to the base EBM $\gamma(M_\star|\boldsymbol{\theta}')$. The feature functions show that galaxies with large environmental densities ρ_1, at low redshift z, or with a large neighboring galaxy (expressed by $\Upsilon_{0.1}$) have an enhanced probability of being outliers relative to $\gamma(M_\star|\boldsymbol{\theta}')$. Galaxies at both high and low $M_{\rm vir}$ or large environmental temperature T_1 are also more likely to be outliers.

We construct the stellar mass CEBM with the sum $\Gamma(M_\star|\boldsymbol{\theta}') \equiv [1-\phi_{M_\star}(\boldsymbol{\theta}')]\gamma(M_\star|\boldsymbol{\theta}')+\phi_{M_\star}(\boldsymbol{\theta}')\delta(M_\star|\boldsymbol{\theta}')$. Figure 5.18 shows the average contribution, feature functions, and interaction functions for $\Gamma(M_\star|\boldsymbol{\theta}')$. The feature with largest average contribution is $M_{\rm vir}$, with redshift z, environmental density ρ_1, environmental temperature T_1, and the mass ratio of nearby galaxies $\Upsilon_{0.1}$ having an lower average contribution by a factor of $\sim 5-10$. Relative to $M_{\rm vir}$, the interactions $[z, \rho_1]$ and $[M_{\rm vir}, \rho_1]$ contribute at level of a few percent. The $M_\star$ CEBM feature function $\tilde{f}(M_{\rm vir})$ has increased in amplitude relative to the $M_\star$ EBM feature function $f(M_{\rm vir})$, subsuming some of the dependence on the missing $v_{\rm peak}$ feature. The remaining feature

Average Contributions for the CEBM $\Gamma(M_\star\|\boldsymbol{\theta}')$	
Feature	Value $[\log_{10} M_\odot]$
$\beta_{\log_{10} M_\star}$	6.6995
$\tilde{f}(\log_{10} M_{\rm vir})$	0.5008
$\tilde{f}(z)$	0.0961
$\tilde{f}(\log_{10} \rho_1)$	0.0902
$\tilde{f}(\log_{10} T_1)$	0.0576
$\tilde{f}(\log_{10} \Upsilon_{0.1})$	0.0336
$\tilde{f}(z, \log_{10} \rho_1)$	0.0172
$\tilde{f}(\log_{10} M_{\rm vir}, \log_{10} \rho_1)$	0.0108

Table 5.7: Summary of the CEBM model $\Gamma(M_\star|\boldsymbol{\theta}')$ trained to predict $M_\star$ using the restricted parameter set $\boldsymbol{\theta}'$. The first entry, $\beta_{\log_{10} M_\star}$, is the learned baseline of the model. The next seven entries are the learned functions with the highest average contribution in descending order. The average contribution is computed via Equation 5.8 (see Section 5.4.2 for more details).

functions for $\Gamma(M_\star|\boldsymbol{\theta}')$ are similar in shape and amplitude to those for $\gamma(M_\star|\boldsymbol{\theta})$, although the contribution at large T_1 and ρ_1 are increased and the dependence on redshift z is decreased. The interaction functions are similar between $\Gamma(M_\star|\boldsymbol{\theta}')$ and $\gamma(M_\star|\boldsymbol{\theta}')$, although there is a larger enhancement of $M_\star$ for large $[M_{\rm vir}, T_1]$ and a smaller enhancement for large $M_{\rm vir}$ and small $\Upsilon_{0.1}$ for the CEBM $\Gamma(M_\star|\boldsymbol{\theta}')$. For reference, the model summary Figure 5.14 illustrates the overall performance of the model.

5.7 Discussion

Explainable Boosting Machine (EBM) models provide a method to statistically infer relationships present in high-dimensional data. Given their statistical nature, EBM models remain ignorant of the physics that generate the connection between star formation rate, stellar mass, and the properties of dark matter halos that host galaxies. Nonetheless, given the results of detailed physical modeling in the form

of simulated galaxy catalogs from cosmological simulations, the EBM correctly identifies halo mass and maximum peak circular velocity as the most important halo properties for determining *SFR* and $M_\star$ (e.g., Figure 5.1). The EBM correctly infers that *SFR* and $M_\star$ increase with increasing halo mass or $v_{\rm peak}$, and the EBM univariate feature functions correctly identify the gas temperature at which star formation efficiency changes. To the extent that the physical connection between galaxy and halo properties are recorded in statistical relationships, the EBM models effectively recover some fraction of those relations.

EBM models also provide a means to implement a "sub-grid" prescription for galaxy formation based on the properties of halos and their environments. The EBM models $\gamma(SFR|\boldsymbol{\theta})$ and $\gamma(M_\star|\boldsymbol{\theta})$ capture better than 97% of the *SFR* and $M_\star$ distributions measured for simulated galaxies in the CROC simulations. The stellar masses and star formation rates of galaxies in CROC could be accurately recovered by using only the halo and environmental parameters in $\boldsymbol{\theta} = [M_{\rm vir}, z, v_{\rm peak}, \rho_1, T_1, \Upsilon_{0.1}]$.

Using the CEBM model trained on the restricted parameter set $\boldsymbol{\theta'} = [M_{\rm vir}, z, \rho_1, T_1, \Upsilon_{0.1}]$, $\approx 95-98\%$ of the distribution of *SFR* and $M_\star$ of the CROC galaxies is recovered. One advantage of this parameter set is that the spatial resolution in the simulations required to compute them is less demanding than for $v_{\rm peak}$. A simulation with coarser resolution than CROC, such that the details of the star formation and feedback processes cannot be resolved, may still leverage the CEBM models $\Gamma(SFR|\boldsymbol{\theta'})$ and $\Gamma(M_\star|\boldsymbol{\theta'})$ to model the star formation rate and stellar masses in dark matter halos. Further, the quantities $\boldsymbol{\theta'}$ used to train the CEBM models are measured at distinct redshifts such that no merger trees are required to recover accurately the CROC *SFR* and $M_\star$ distributions from halo and environmental properties. We note that for both the EBM and CEBM models the outlier frac-

tions not well captured by the model are roughtly percent-level or less in the *SFR* or $M_\star$ distributions, and we expect that corresponding inaccuracies induced in, e.g., the ionizing photon budget or topology of reionization will be minimal.

By editing the dataset and retraining, the impact of environment on the performance of the EBM models can be estimated. Relative to $\gamma(SFR|\boldsymbol{\theta})$ and $\gamma(M_\star|\boldsymbol{\theta})$ that use the full dataset $\boldsymbol{\theta}$ including all environmental parameters, EBM models trained only on maximum peak circular velocity $v_{\rm peak}$, halo virial mass $M_{\rm vir}$, and redshift z have an outlier fraction increased by only $\sim 1\%$ when modeling $M_\star$ and $\sim 10\%$ when modeling *SFR*. Further, removing $v_{\rm peak}$ and training only on $[M_{\rm vir}, z]$ substantially degrades the model performance, and the outlier fractions increase to $\sim 20\%$ when modeling $M_\star$ and $\sim 40\%$ when modeling *SFR*. The importance of including $v_{\rm peak}$ in the training dataset is much larger than the importance of accounting for the environmental measures selected in this analysis.

The EBM models enable an approximate translation of the galaxy formation model from one simulation to another. Provided the parameter sets $\boldsymbol{\theta}$ or $\boldsymbol{\theta'}$ can be measured in both simulations, an EBM can recover the connection between *SFR*, stellar masses, halo properties, and environment from the training simulation and then be used to instill those relations in a different simulation. Since the $\boldsymbol{\theta'}$ parameter set does not require very high spatial resolution to capture, the net results for *SFR* and stellar mass from a high resolution simulation accurately tracking detailed baryonic physics can be translated into a simulation with resolution insufficient to capture those physics directly. In future work, we plan to transfer the CROC baryonic galaxy formation model into *Cholla* cosmological simulations [e.g., 274, 275] via the EBM models presented here. Such a transferred model could be used to build models of feedback from galaxy formation on resolved scales that incorporate the regulatory effects of feedback on small-scale

star formation.

Lastly, the ability of the EBM models to recover the *SFR* and $M_\star$ distributions using only halo and environmental properties allows for the rapid replacement of galaxy formation models based on EBMs. Models can be trained on the simulated galaxy catalogs from a variety of expensive, high-resolution training simulations including a wide range of physics. These EBM models can then be used interchangeably as effective galaxy formation models in the target simulations, and can also be modified posteriori to allow a broad parameter search or correct the inaccuracies of the training simulation. Such an approach could reduce the sensitivity of conclusions about, e.g., the reionization process on the detailed *SFR* and $M_\star$ distributions as multiple EBM models for these properties could be trained and implemented in the target simulations.

5.8 Summary

A complex interplay of physical processes gives rise to the distribution of star formation rates (*SFR*s) and stellar masses $M_\star$ of galaxies over cosmic time. Cosmological simulations provide powerful methods for modeling these physical processes, but the connection between *SFR*, $M_\star$, and other galaxy properties can be obfuscated by complexity. Leveraging machine learning techniques, we use a variation of the Generalized Additive Model [101] called Explainable Boosting Machines [EBM 196] to infer the dependence of *SFR* and $M_\star$ in the Cosmic Reionization on Computers (CROC) simulations [92] on dark matter halo properties including virial mass $M_{\rm vir}$, peak maximum circular velocity $v_{\rm peak}$, redshift, environmental density, environmental gas temperature, and the mass of neighboring halos. Our findings include:

- *SFR* and $M_\star$ primarily depend on $M_{\rm vir}$ and $v_{\rm peak}$, followed by redshift, environmental density, and environmental gas temperature.

- When including $M_{\rm vir}$ and $v_{\rm peak}$ in the parameter set used to train the EBM, the model recovers better than 97% of the distribution of $M_\star$ or *SFR* with virial mass $M_{\rm vir}$ in the CROC simulations.

- If the model fit excludes $v_{\rm peak}$, the fraction of outliers in the CROC data relative to the predicted model distribution increases to 7.6% for *SFR* and 2.8% for $M_\star$.

- To ameliorate the degradation of the model performance when excluding $v_{\rm peak}$, we define a composite EBM model comprised of a weighted sum of the base EBM model fit to main trend of *SFR* and $M_\star$ with the halo properties and a second EBM model to fit the outliers not represented in the base EBM. The weighting coefficients are themselves determined by an EBM model fit.

- The composite EBM model improves the performance to $\approx 95-98\%$ accuracy in the distribution of *SFR* or $M_\star$ with virial mass, even when excluding $v_{\rm peak}$ measurements from the training dataset.

The EBM models quantify the relative importance of halo properties like virial mass and maximum peak circular velocity for determining the stellar mass and star formation rate of the galaxy it hosts. Through these models, the physics of baryonic galaxy formation can be connected to the properties of dark matter halos and enable galaxy formation to be implemented as a "sub-grid" prescription in dark matter-only simulations or hydrodynamical simulations that do not resolve the small scale details of star formation and feedback.

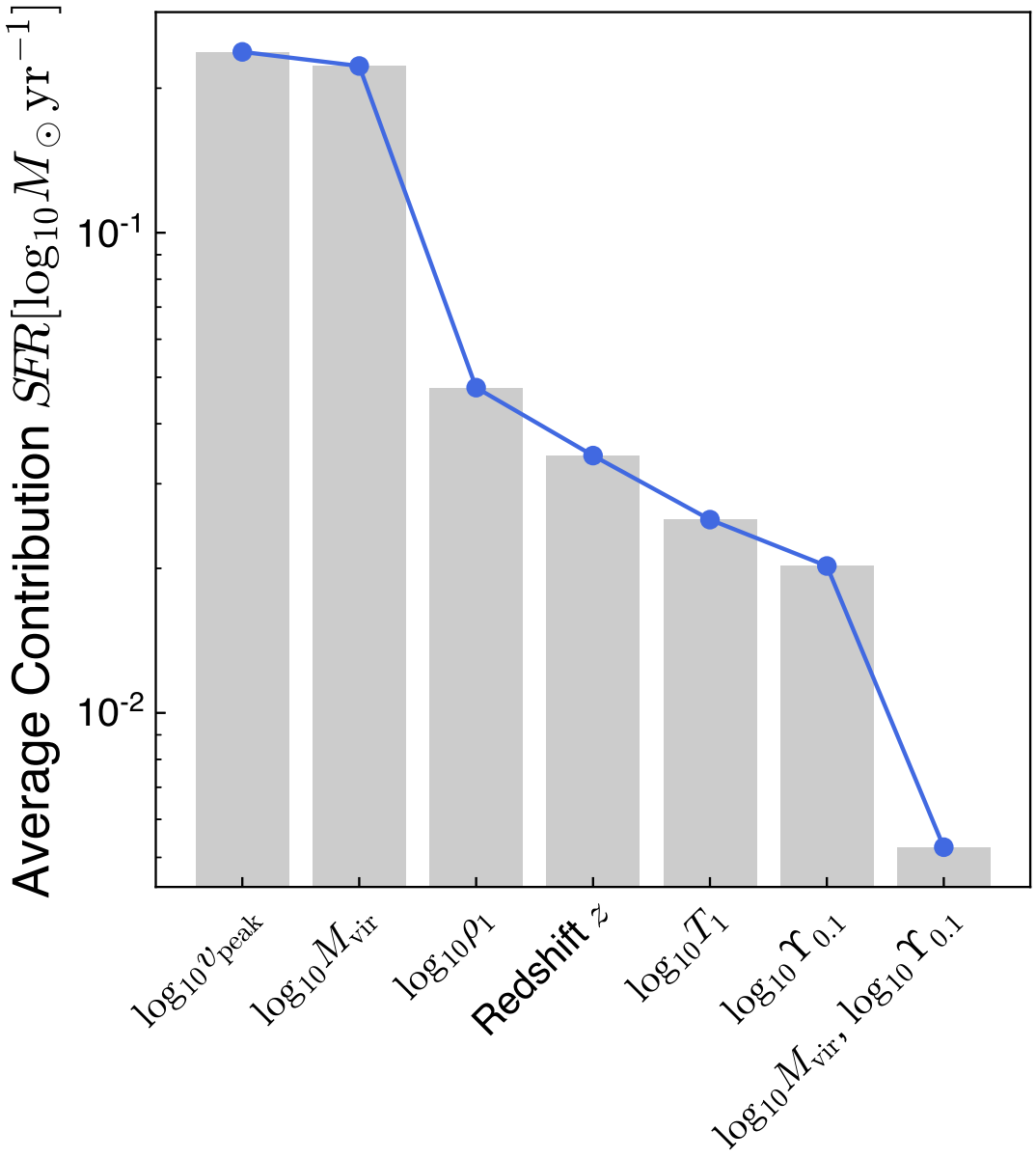

Figure 5.1: Top seven features with the highest average contribution in the EBM $\gamma(SFR|\boldsymbol{\theta})$ targeting the star formation rate SFR. In order of decreasing importance, these features include peak circular velocity v_{peak}, virial mass M_{vir}, environmental density ρ_1, redshift z, environmental temperature T_1, the mass ratio of nearby halos $\Upsilon_{0.1}$, and the interaction between virial mass M_{vir} and $\Upsilon_{0.1}$. Average contribution is calculated using the average of the absolute value of the feature functions weighted by the number of samples in each bin (see Equation 5.5).

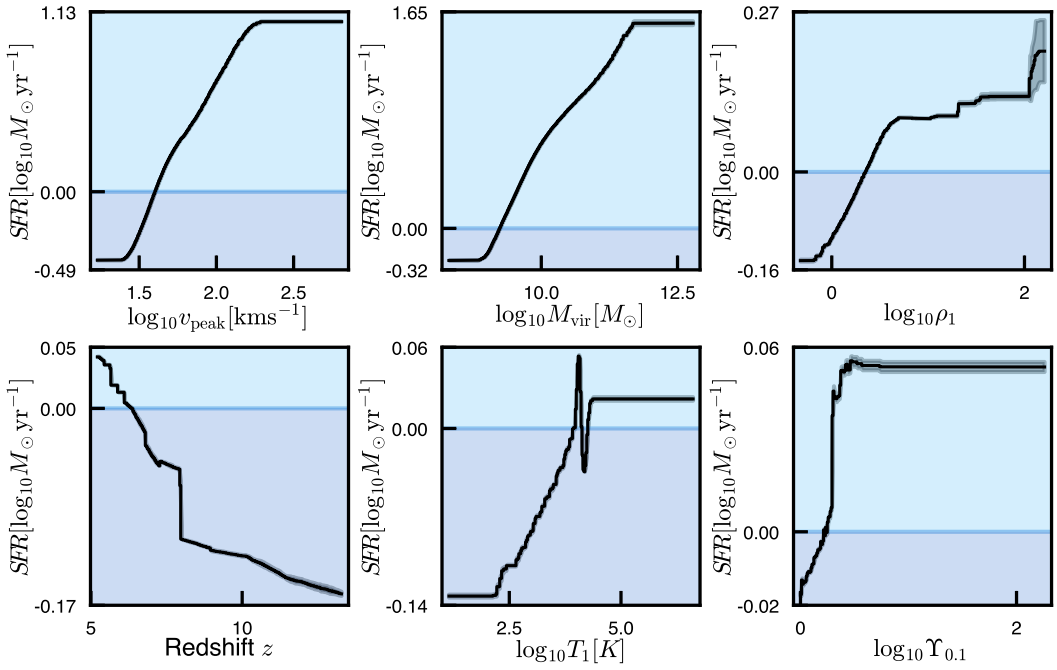

Figure 5.2: Learned univariate feature functions f_y^i for the EBM $\gamma(SFR|\boldsymbol{\theta})$ trained to predict the star formation rate *SFR*. Shown (left to right) are the feature functions for peak circular velocity v_{peak}, virial mass M_{vir}, environmental density ρ_1, redshift z, environmental temperature T_1, and nearby halo mass ratio $\Upsilon_{0.1}$. Light blue areas indicate regions where $f_y^i > 0$ and dark blue areas indicate regions where $f_y^i < 0$. The shaded areas show the variation in f_y^i between the k-fold iterations.

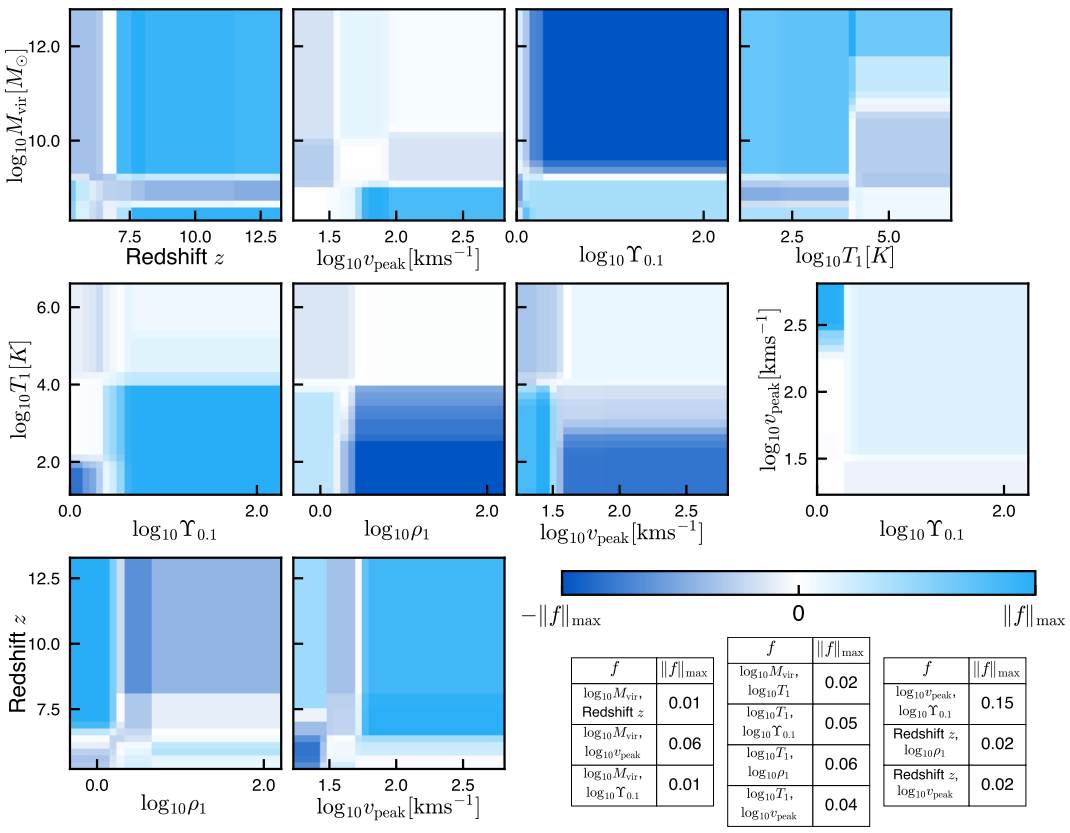

Figure 5.3: Most important learned interaction functions f_y^{ij} for the EBM model $\gamma(SFR|\boldsymbol{\theta})$ targeting the star formation rate SFR, as a function of their parameter pairs. Each panel shows the contribution of the bivariate interaction terms, normalized such that the color map ranges between plus or minus the maximum of the norm of each function $||f||_{\rm max}$. Light blue areas indicate regions of joint parameter space where the feature interactions contribute positively to the star formation rate, while dark blue areas indicate regions with negative contributions. The table lists $||f||_{\rm max}$ for the interaction functions, each with units $\log_{10} M_\odot\ {\rm yr}^{-1}$. In absolute terms, the largest interaction occurs for halos with large peak circular velocity $v_{\rm peak}$ and no large neighboring halos ($\Upsilon_{0.1} \approx 0$). The other interaction functions are relatively weak, and contribute changes to $\log SFR \lesssim 0.05$.

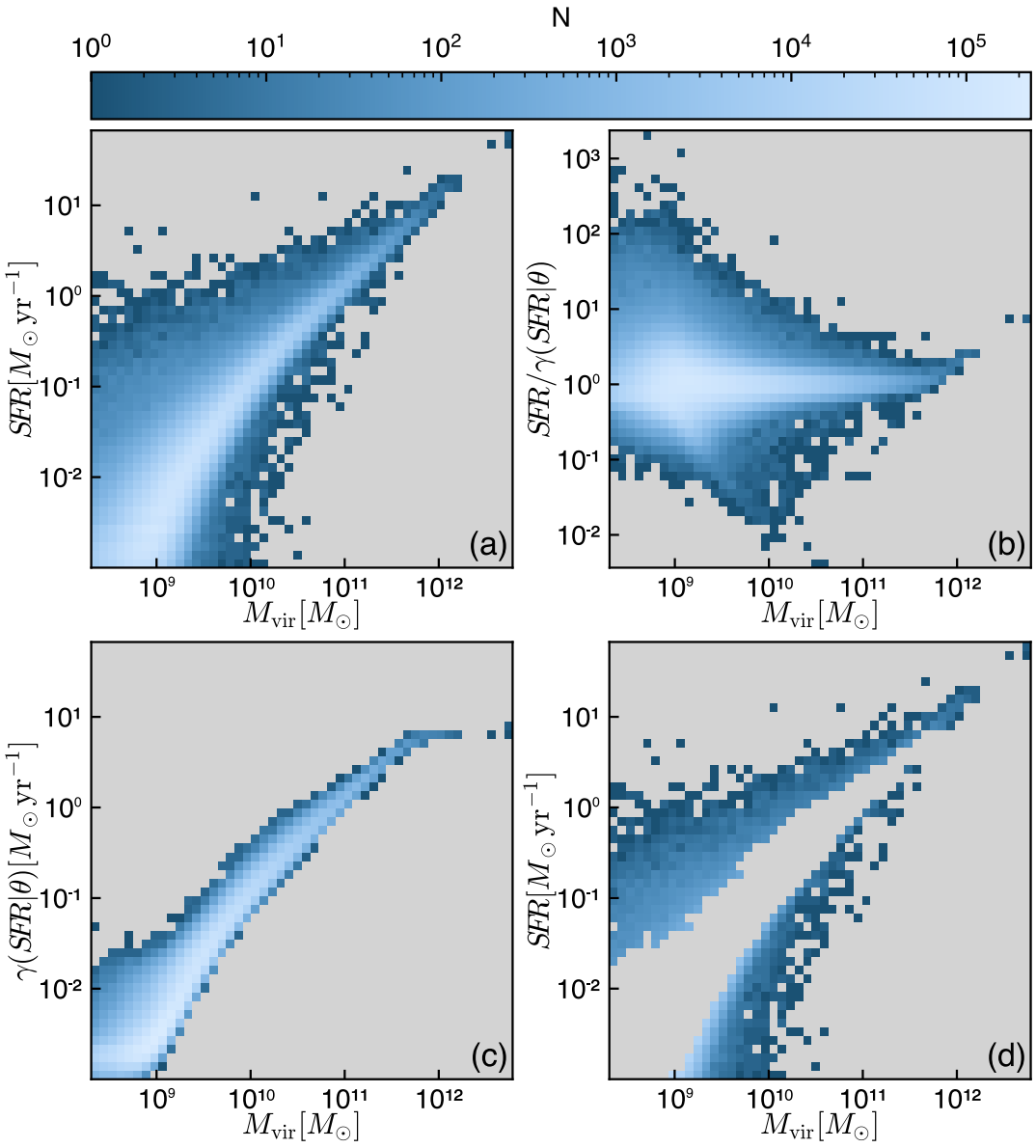

Figure 5.4: Summary of the EBM model $\gamma(SFR|\boldsymbol{\theta})$ targeting star formation rate (SFR) as a function of virial mass. The upper left panel shows the two-dimensional distribution of SFR with $M_{\rm vir}$ for galaxies in the CROC simulations, with the color scale showing the number of simulated galaxies at each $[SFR, M_{\rm vir}]$ location. The lower left panel shows the EBM model results for the distribution of SFR with $M_{\rm vir}$, where the SFR is computed from the EBM using the parameters $\boldsymbol{\theta} = [M_{\rm vir}, v_{\rm peak}, z, \rho_1, T_1, \Upsilon_{0.1}]$. The upper right panel shows the residuals between the simulated CROC galaxy SFRs and the EBM model results. The lower right panel shows the simulated CROC galaxy SFRs that lie outside the EBM model predictions. These outliers represent $\lesssim 3\%$ of simulated CROC galaxies.

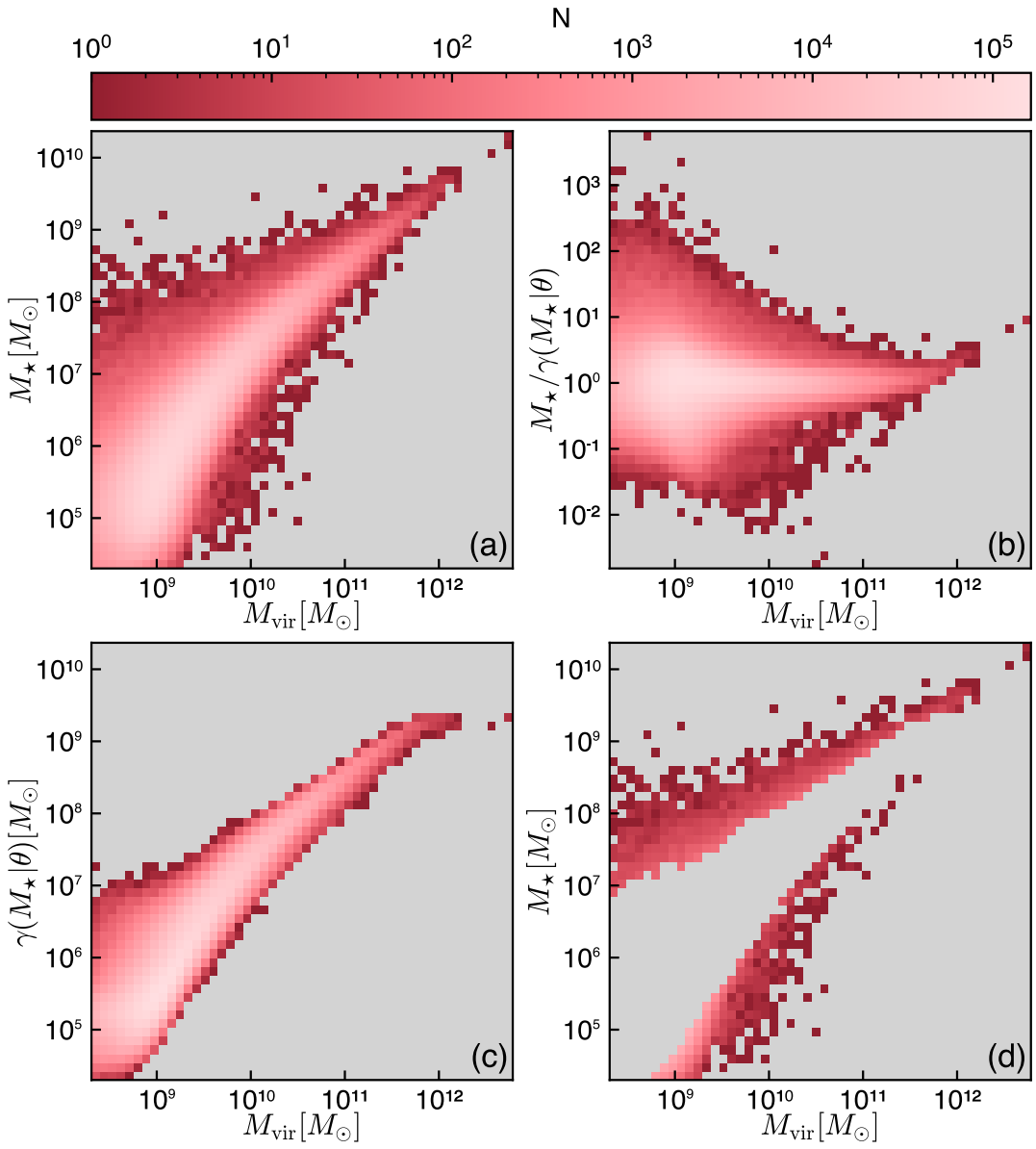

Figure 5.5: Summary of the EBM model $\gamma(M_\star|\boldsymbol{\theta})$ targeting stellar mass $M_\star$ as a function of virial mass. The upper left panel shows the distribution of $M_\star$ with virial mass $M_{\rm vir}$ in the CROC simulated galaxy catalogs, with the coloration indicating the number of galaxies at each $[M_\star, M_{\rm vir}]$ location. The lower left panel shows the EBM model prediction of the stellar mass distribution with virial mass given in the input parameters $\boldsymbol{\theta} = [M_{\rm vir}, z, v_{\rm peak}, \rho_1, T_1, \Upsilon_{0.1}]$. The upper right panel shows the residuals between the simulated and predicted $M_\star$ vs. $M_{\rm vir}$ distribution, and the lower right panel shows the outliers in the simulated distribution not captured by the EBM model $\gamma(M_\star|\boldsymbol{\theta})$. The fraction of outliers is $\lesssim 1\%$.

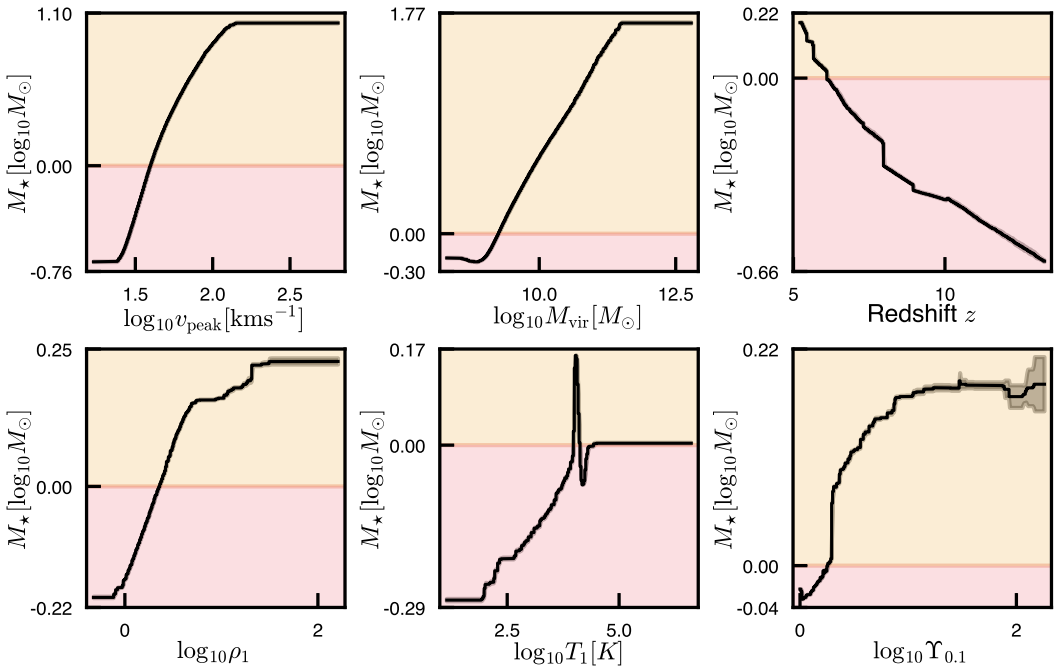

Figure 5.6: Learned univariate feature functions, f_y^i in Equation 5.1, for the EBM $\gamma(M_\star|\boldsymbol{\theta})$ trained to predict $M_\star$. Areas highlighted in orange indicate portions of the function that contribute positively to the predicted $M_\star$ and areas in red contribute negatively. Stellar mass increases with peak circular velocity and virial mass, increases with decreasing redshift, and increases with environmental density. Temperature correlates positively with stellar mass, with a strong feature near $T_1 \approx 10^4$ K where hydrogen ionizes. Stellar mass also increases with the mass ratio of neighboring halos.

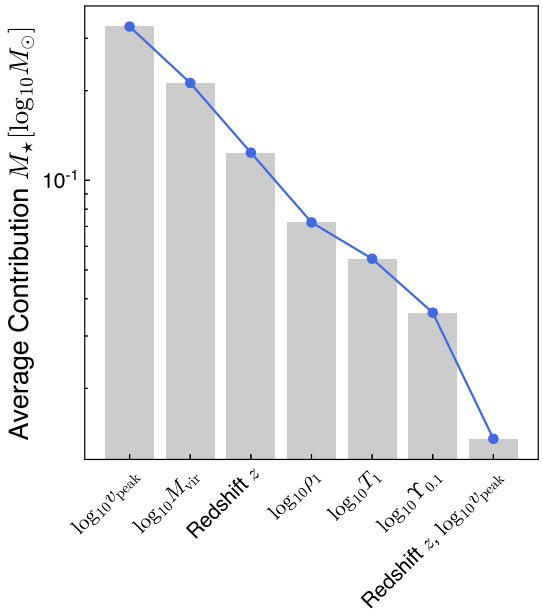

Figure 5.7: Features with the highest average contribution for the EBM $\gamma(M_\star|\boldsymbol{\theta})$ trained to predict $M_\star$. Average contribution is calculated using the average of the absolute value of the learned functions weighted by the number of samples in each bin (see Equation 5.5). The features with the largest contributon are $v_{\rm peak}$ and $M_{\rm vir}$, followed by redshift z, environmental density ρ_1, environmental temperature T_1, and mass ratio of nearby halos $\Upsilon_{0.1}$. The interaction with the largest average contribution involves [z,$v_{\rm peak}$].

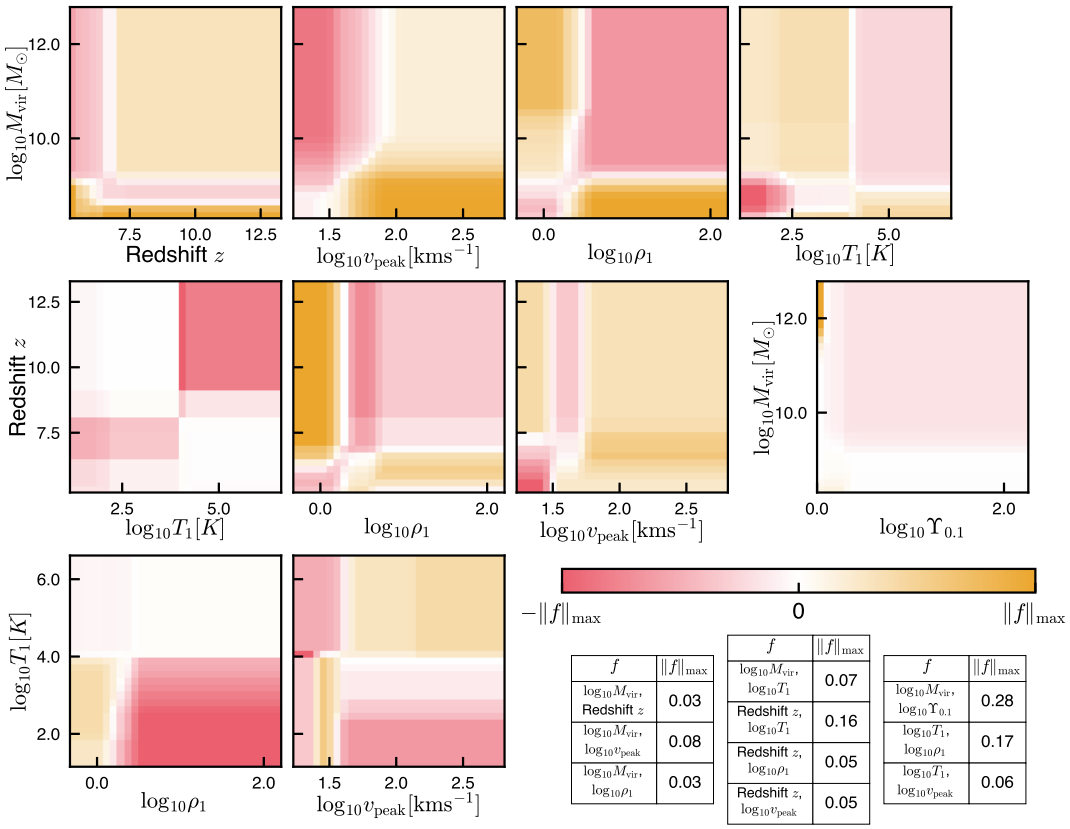

Figure 5.8: Learned bivariate interaction functions f_y^{ij} for the EBM $\gamma(M_\star|\boldsymbol{\theta})$ trained to predict $M_\star$. Areas highlighted in orange indicate portions of the functions that contribute positively to the predicted $M_\star$ while areas in red contribute negatively. Halos with large environmental temperatures T_1 at high redshift z show enhanced stellar mass. The stellar masses of halos with low environmental temperature $T_1 < 10^4$K correlate with environmental density, increasing with increasing ρ_1. Massive halos with no comparable large neighboring halos ($\Upsilon_{0.1} \approx 0$) also show enhanced stellar mass.

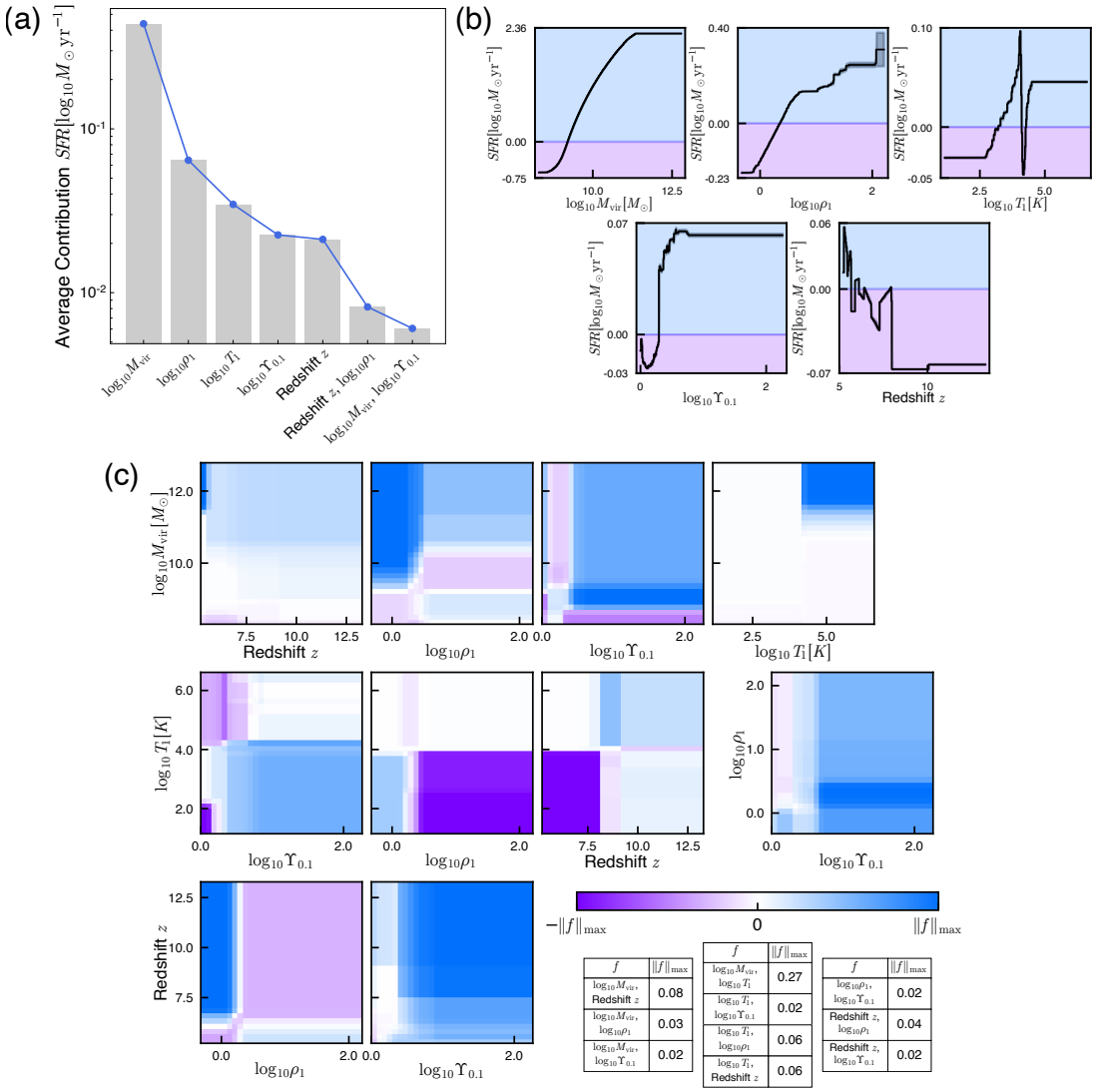

Figure 5.9: Details for the base EBM model $\gamma(SFR|\boldsymbol{\theta}')$ component of the CEBM $\Gamma(SFR|\boldsymbol{\theta}')$ trained to predict *SFR*. Panel a) displays the average contribution of features. Panel b) shows the feature functions contributing to the base EBM model. Panel c) presents the interaction functions for the base EBM $\gamma(SFR|\boldsymbol{\theta}')$. Each panel shows the contribution of the bivariate interaction terms, normalized such that the color map ranges between plus or minus the maximum of the norm of each function $||f||_{\max}$. Purple indicates negative contributions and blue indicates positive contributions. The table lists $||f||_{\max}$ for the interaction functions, each with units $\log_{10} M_\odot\ \mathrm{yr}^{-1}$.

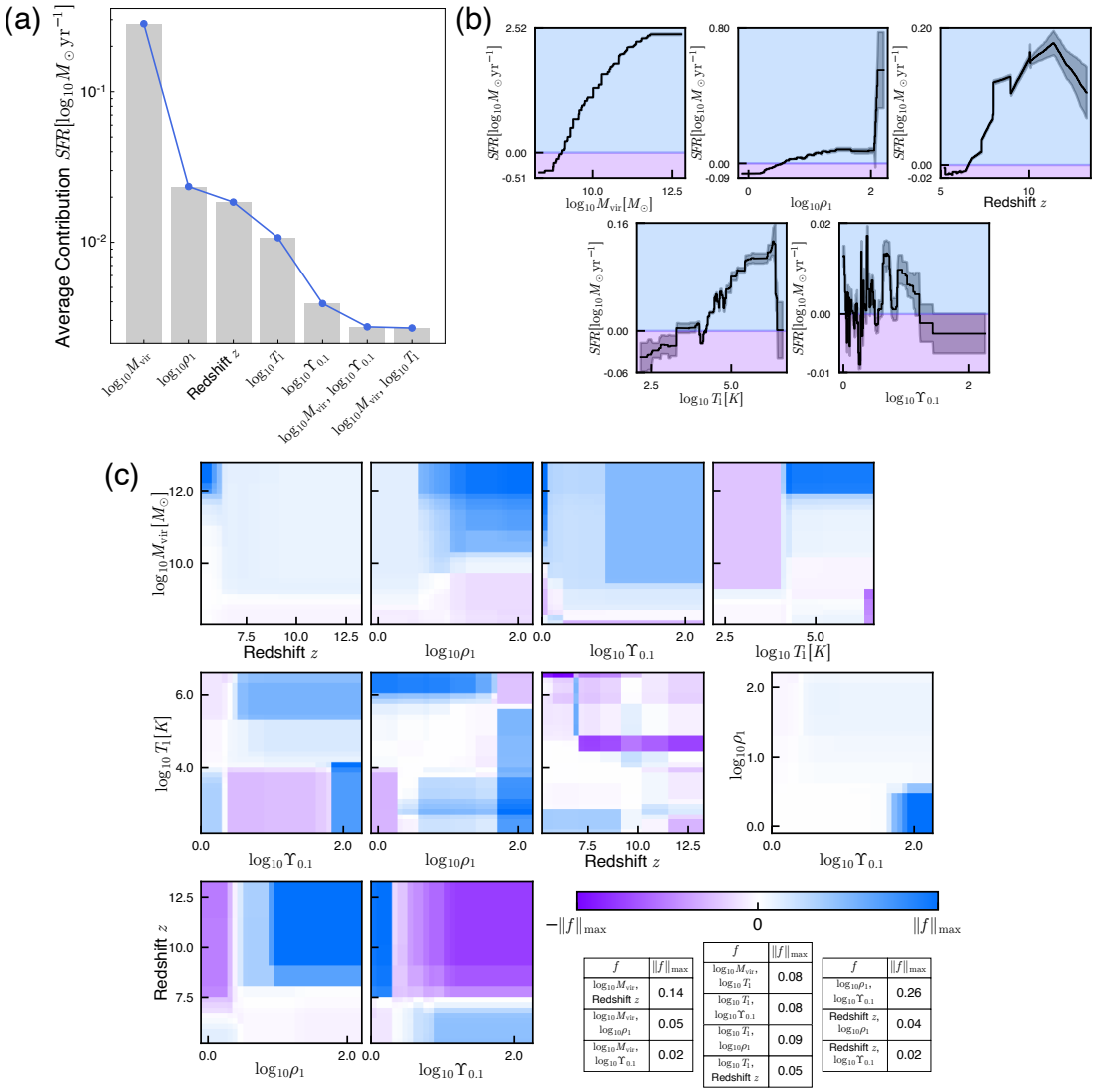

Figure 5.10: Details for the outlier EBM model $\delta(SFR|\boldsymbol{\theta}')$ component of the CEBM $\Gamma(SFR|\boldsymbol{\theta}')$ trained to predict *SFR*. Panel a) displays the average contribution of features. Panel b) shows the feature functions for the outlier EBM $\delta(SFR|\boldsymbol{\theta}')$. Panel c) presents the interaction functions for the outlier EBM $\delta(SFR|\boldsymbol{\theta}')$. Each panel shows the contribution of the bivariate interaction terms, normalized such that the color map ranges between plus or minus the maximum of the norm of each function $||f||_{\max}$. Purple indicates negative contributions and blue indicates positive contributions. The table lists $||f||_{\max}$ for the interaction functions, each with units $\log_{10} M_\odot \ \mathrm{yr}^{-1}$.

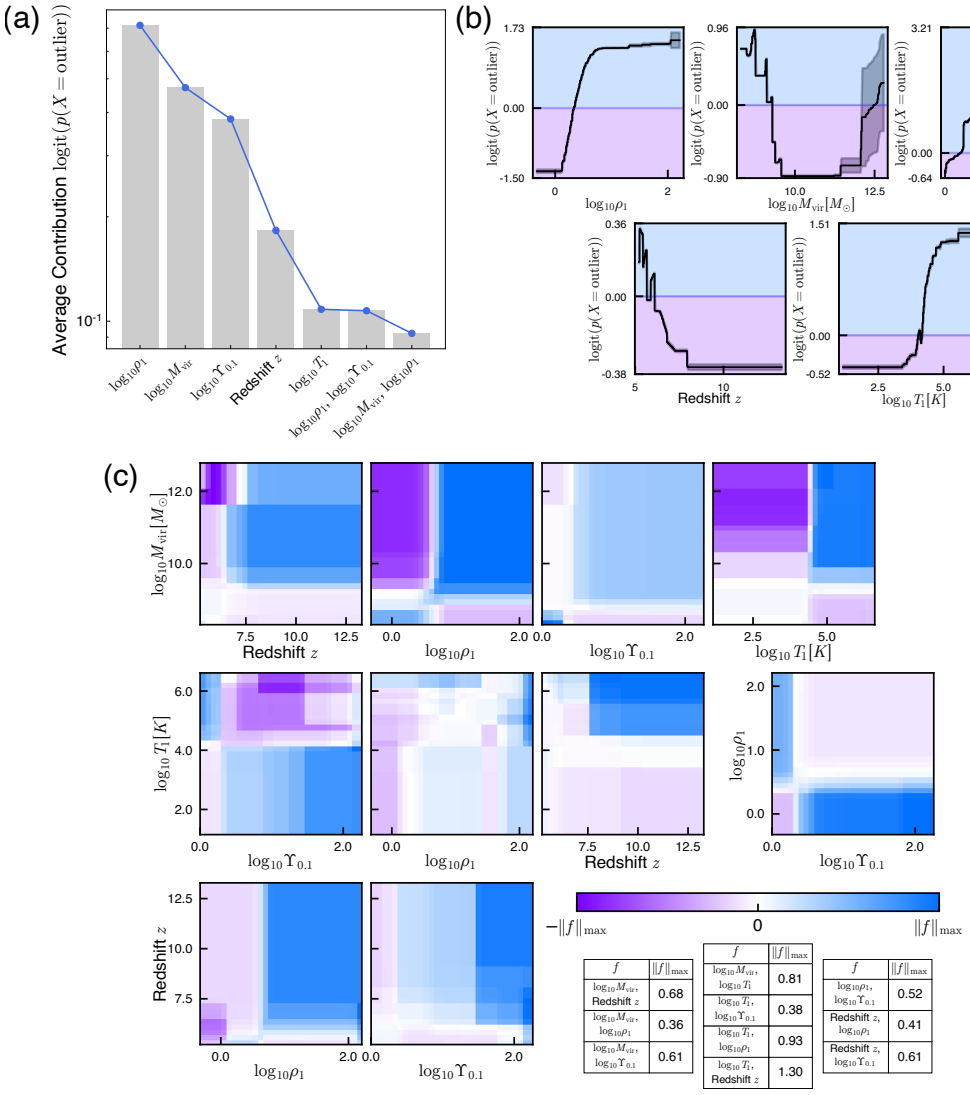

Figure 5.11: Details for the classification EBM model $\phi_{SFR}(\boldsymbol{\theta}')$ that interpolates between the base EBM $\gamma(SFR|\boldsymbol{\theta}')$ and the outlier EBM $\delta(SFR|\boldsymbol{\theta}')$ for creating the CEBM $\Gamma(SFR|\boldsymbol{\theta}')$. Panel a) displays the average contribution of features to the classification EBM model $\phi_{SFR}(\boldsymbol{\theta}')$. Panel b) shows the feature functions contributing to the classifier EBM $\phi_{SFR}(\boldsymbol{\theta}')$. Panel c) presents the interaction functions for the classifier EBM $\phi_{SFR}(\boldsymbol{\theta}')$. Each panel shows the contributions of the interaction terms, normalized such that the color map ranges between plus or minus the maximum of the norm of each function $||f||_{\max}$. Purple indicates negative log odds and blue indicates positive log odds that a given galaxy is an outlier in SFR. The table lists $||f||_{\max}$ for the interaction functions, listed as the corresponding change in log odds.

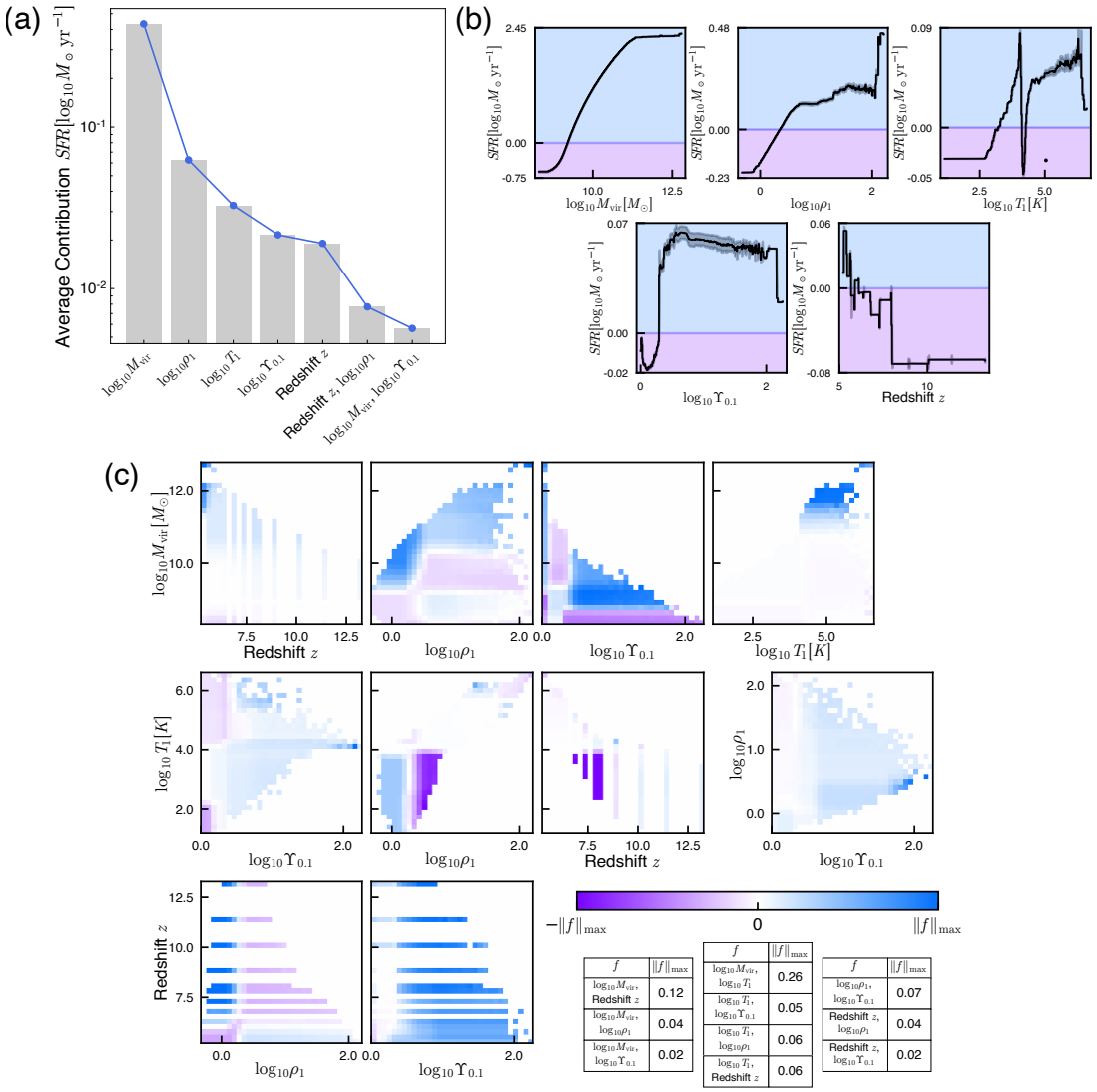

Figure 5.12: Details for the CEBM model $\Gamma(SFR|\boldsymbol{\theta}')$ trained to predict *SFR*. Panel a) displays the average contribution of features to the CEBM. Virial mass $M_{\rm vir}$ provides the largest average contribution to the star formation rate. Panel b) shows the feature functions contributing to the CEBM $\Gamma(SFR|\boldsymbol{\theta}')$. Panel c) presents the interaction functions for the CEBM $\Gamma(SFR|\boldsymbol{\theta}')$. Each panel shows the contribution of the interaction terms, normalized such that the color map ranges between plus or minus the maximum of the norm of each function $||f||_{\rm max}$. Purple indicates negative contributions and blue indicates positive contributions. The table lists $||f||_{\rm max}$ for the interaction functions, each with units $\log_{10} M_\odot\ \mathrm{yr}^{-1}$.

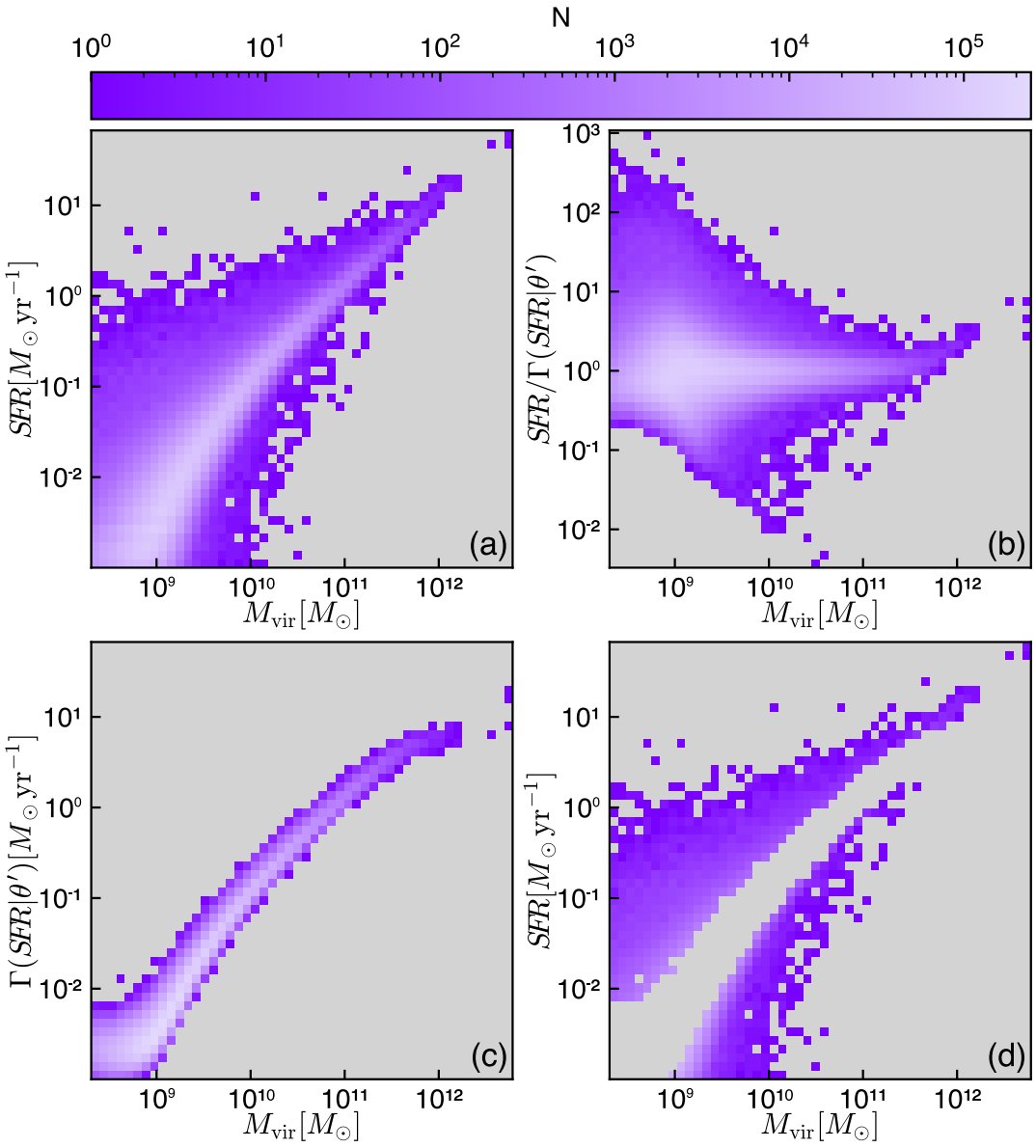

Figure 5.13: Summary of the CEBM model $\Gamma(SFR|\boldsymbol{\theta'})$ targeting star formation rate (SFR) as a function of virial mass. The upper left panel shows the two-dimensional distribution of SFR with $M_{\rm vir}$ for galaxies in the CROC simulations, with the color scale showing the number of simulated galaxies at each $[SFR, M_{\rm vir}]$ location. The lower left panel shows the CEBM model results for the distribution of SFR with $M_{\rm vir}$, where the SFR is computed from the CEBM using the parameters $\boldsymbol{\theta'} = [M_{\rm vir}, z, \rho_1, T_1, \Upsilon_{0.1}]$. The upper right panel shows the residuals between the simulated CROC galaxy SFRs and the CEBM model results. The lower right panel shows the simulated CROC galaxy SFRs that lie outside the CEBM model predictions. These outliers represent $\approx 5\%$ of simulated CROC galaxies.

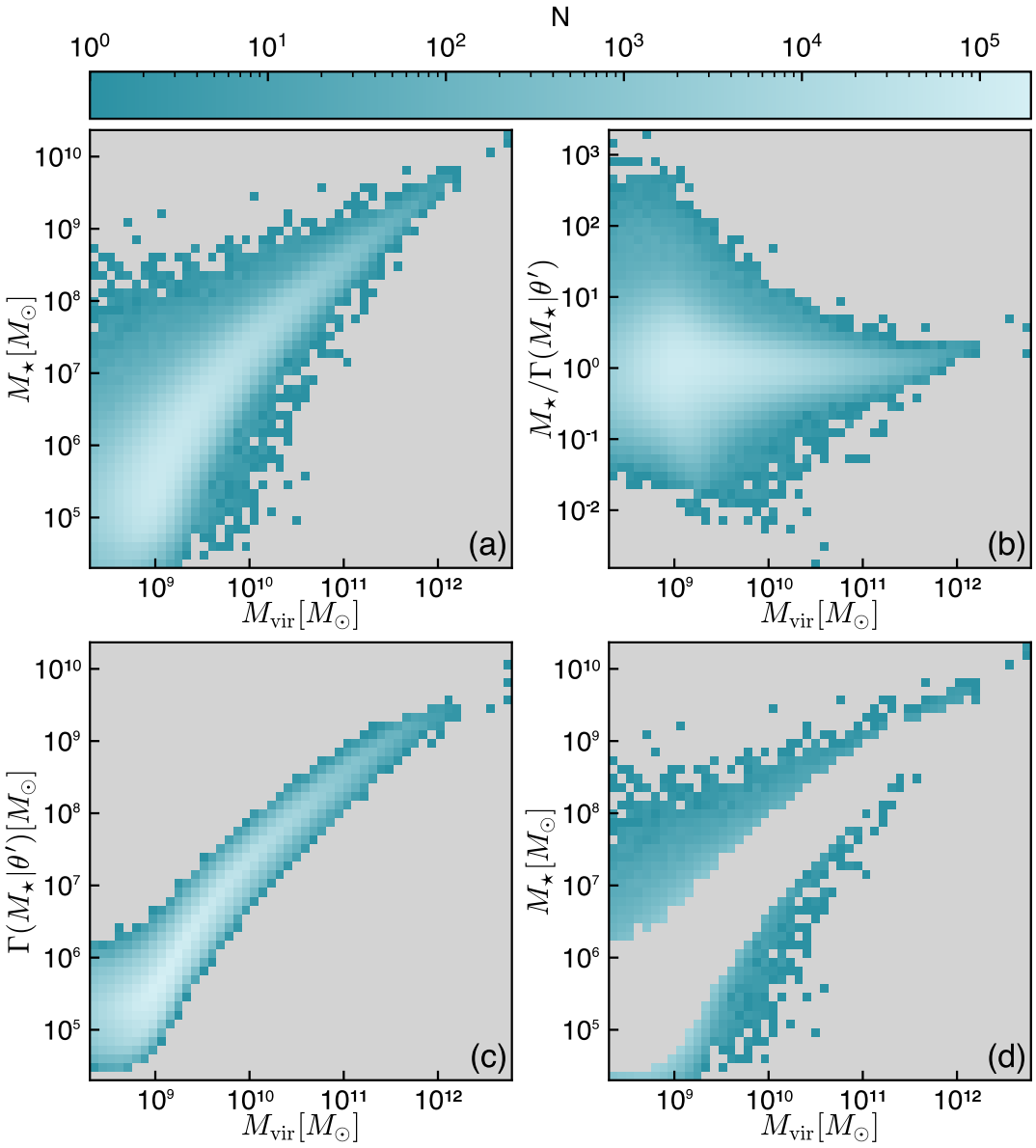

Figure 5.14: Summary of the CEBM model $\Gamma(M_\star|\boldsymbol{\theta'})$ targeting stellar mass $M_\star$ as a function of virial mass. The upper left panel shows the distribution of $M_\star$ with virial mass $M_{\rm vir}$ in the CROC simulated galaxy catalogs, with the coloration indicating the number of galaxies at each $[M_\star, M_{\rm vir}]$ location. The lower left panel shows the CEBM model prediction of the stellar mass distribution with virial mass given in the input parameters $\boldsymbol{\theta'} = [M_{\rm vir}, z, \rho_1, T_1, \Upsilon_{0.1}]$. The upper right panel shows the residuals between the simulated and predicted $M_\star$ vs. $M_{\rm vir}$ distribution, and the lower right panel shows the outliers in the simulated distribution not captured by the CEBM model $\Gamma(M_\star|\boldsymbol{\theta'})$. The fraction of outliers is $\lesssim 2\%$.

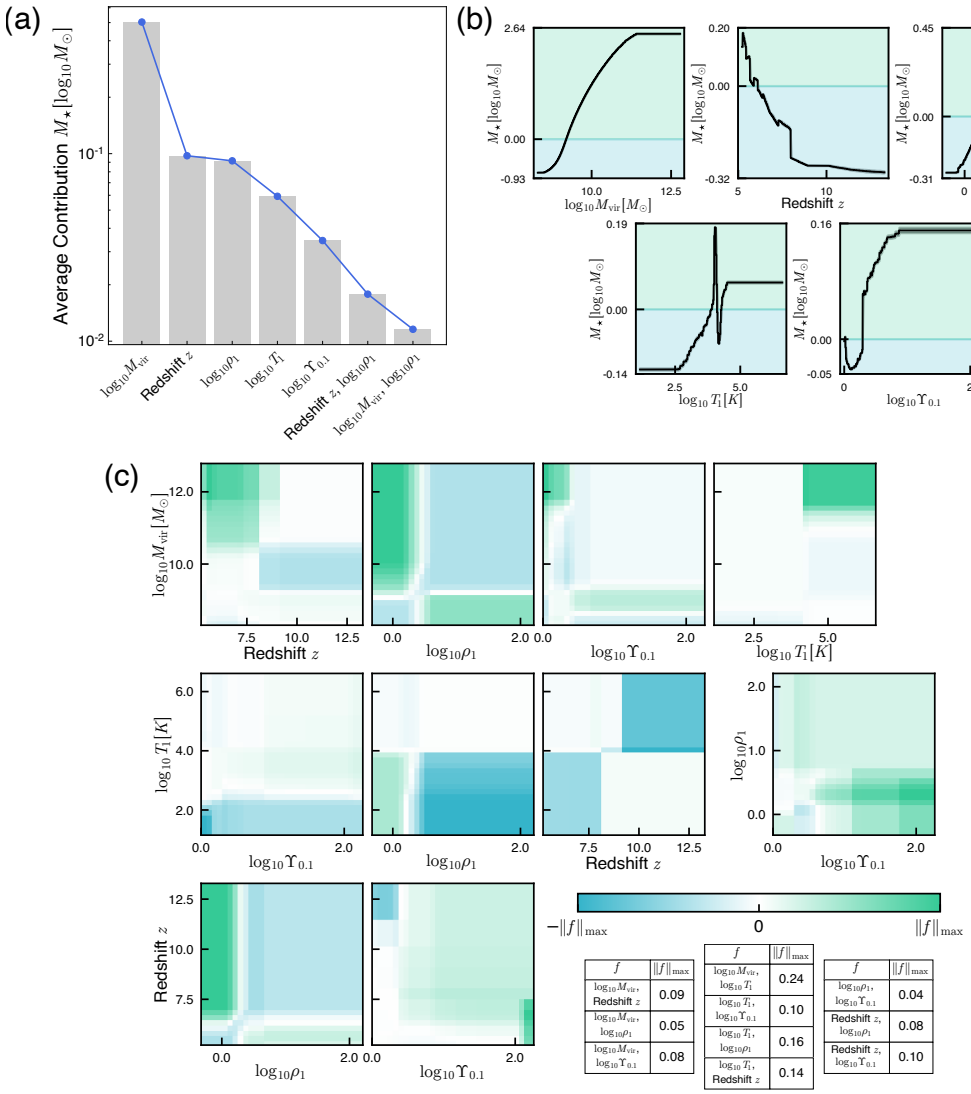

Figure 5.15: Details for the base EBM model $\gamma(M_\star|\boldsymbol{\theta}')$ component of the CEBM $\Gamma(M_\star|\boldsymbol{\theta}')$ trained to predict stellar mass $M_\star$. Panel a) displays the average contribution of features to the base EBM model $\gamma(M_\star|\boldsymbol{\theta}')$. Panel b) shows the feature functions contributing to the base EBM model $\gamma(M_\star|\boldsymbol{\theta}')$. Panel c) presents the interaction functions for the base EBM $\gamma(M_\star|\boldsymbol{\theta}')$. Each panel shows the contribution of the bivariate interaction terms, normalized such that the color map ranges between plus or minus the maximum of the norm of each function $||f||_{\rm max}$. Teal indicates negative contributions and green indicates positive contributions. The table lists $||f||_{\rm max}$ for the interaction functions, each with units $\log_{10} M_\odot$.

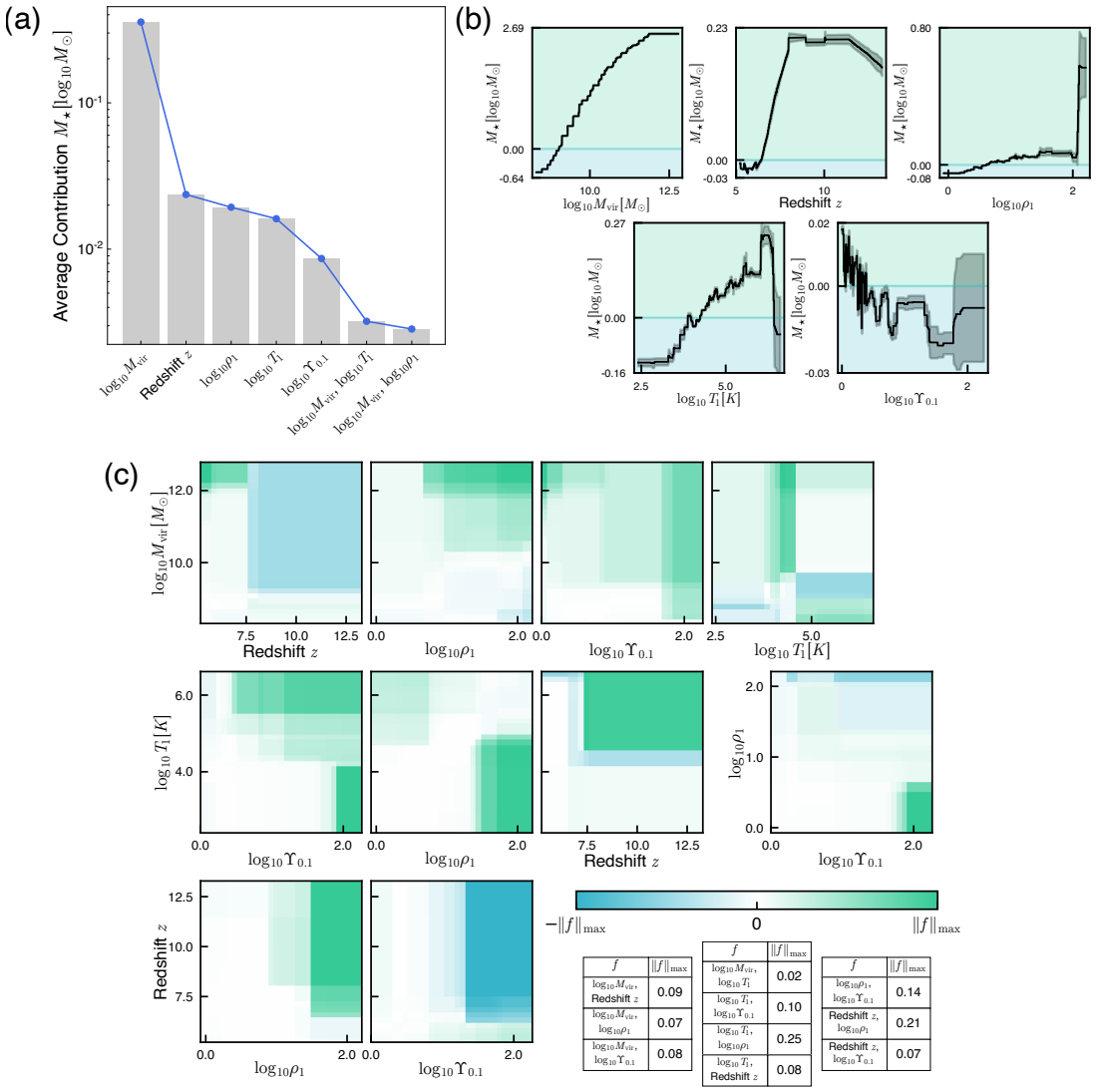

Figure 5.16: Details for the outlier EBM model $\delta(M_\star|\boldsymbol{\theta}')$ component of the CEBM $\Gamma(M_\star|\boldsymbol{\theta}')$ trained to predict $M_\star$. Panel a) displays the average contribution of features to the outlier EBM model $\delta(M_\star|\boldsymbol{\theta}')$. Panel b) shows the feature functions for the outlier EBM $\delta(M_\star|\boldsymbol{\theta}')$. qPanel c) presents the interaction functions for the outlier EBM $\delta(M_\star|\boldsymbol{\theta}')$. Each panel shows the contribution of the interaction terms, normalized such that the color map ranges between plus or minus the maximum of the norm of each function $||f||_{\rm max}$. Teal indicates negative contributions and green indicates positive contributions. The table lists $||f||_{\rm max}$ for the interaction functions, each with units $\log_{10} M_\odot$.

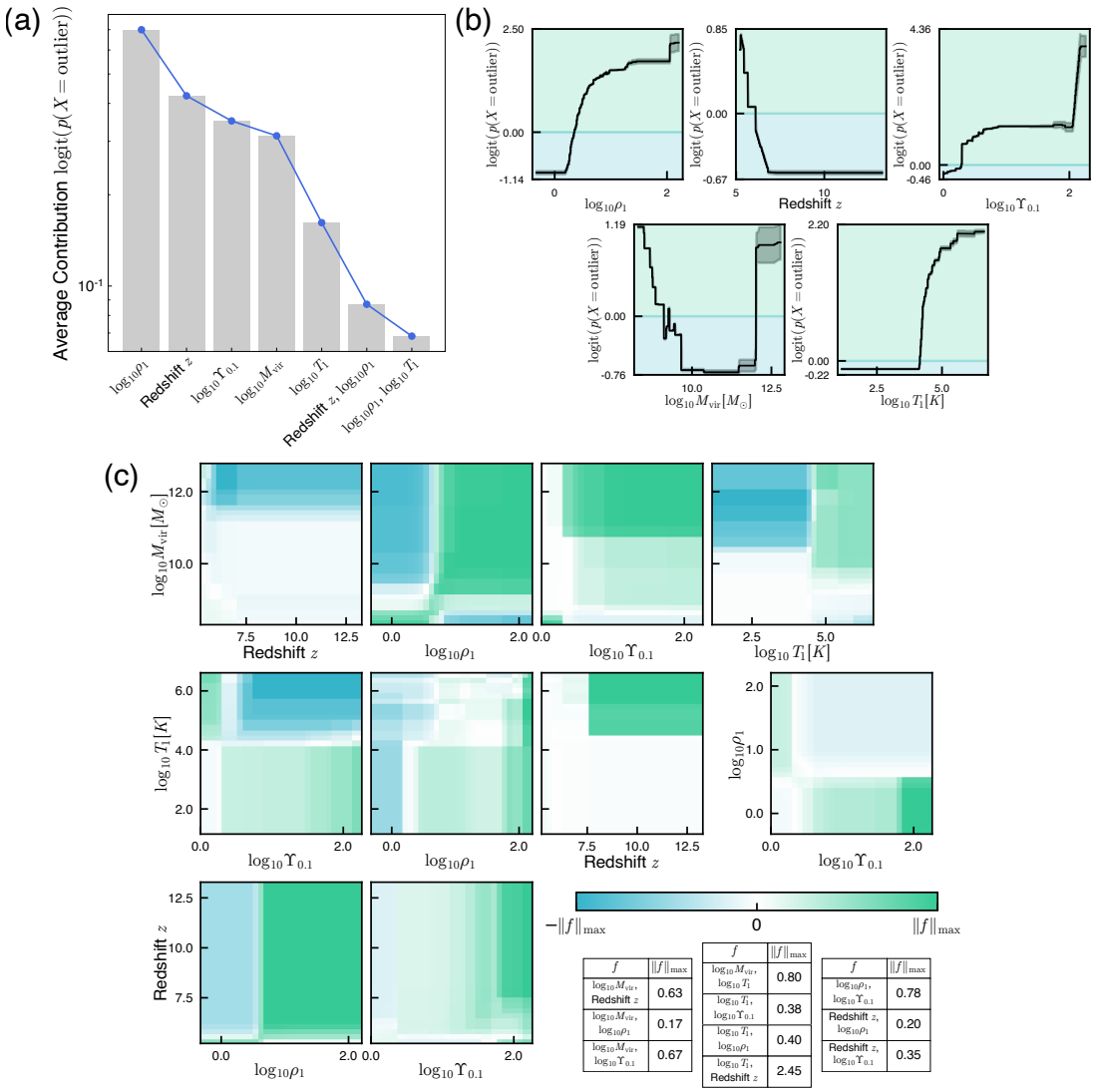

Figure 5.17: Details for the classification EBM model $\phi_{M_\star}(\boldsymbol{\theta}')$ that interpolates between the base EBM $\gamma(M_\star|\boldsymbol{\theta}')$ and the outlier EBM $\delta(M_\star|\boldsymbol{\theta}')$ for creating the CEBM $\Gamma(M_\star|\boldsymbol{\theta}')$. Panel a) displays the average contribution of features to the classification EBM model $\phi_{M_\star}(\boldsymbol{\theta}')$. Panel b) shows the feature functions contributing to the classifier EBM $\phi_{M_\star}(\boldsymbol{\theta}')$. These feature functions represent the change in log odds that a given galaxy will be an outlier in $M_\star$. Panel c) presents the interaction functions for the classifier EBM $\phi_{M_\star}(\boldsymbol{\theta}')$. Each panel shows the contributions of the interaction terms, normalized such that the color map ranges between plus or minus the maximum of the norm of each function $||f||_{\rm max}$. Teal indicates negative log odds and green indicates positive log odds that a given galaxy is an outlier in stellar mass. The table lists $||f||_{\rm max}$ for the interaction functions, listed as the corresponding change in log odds.

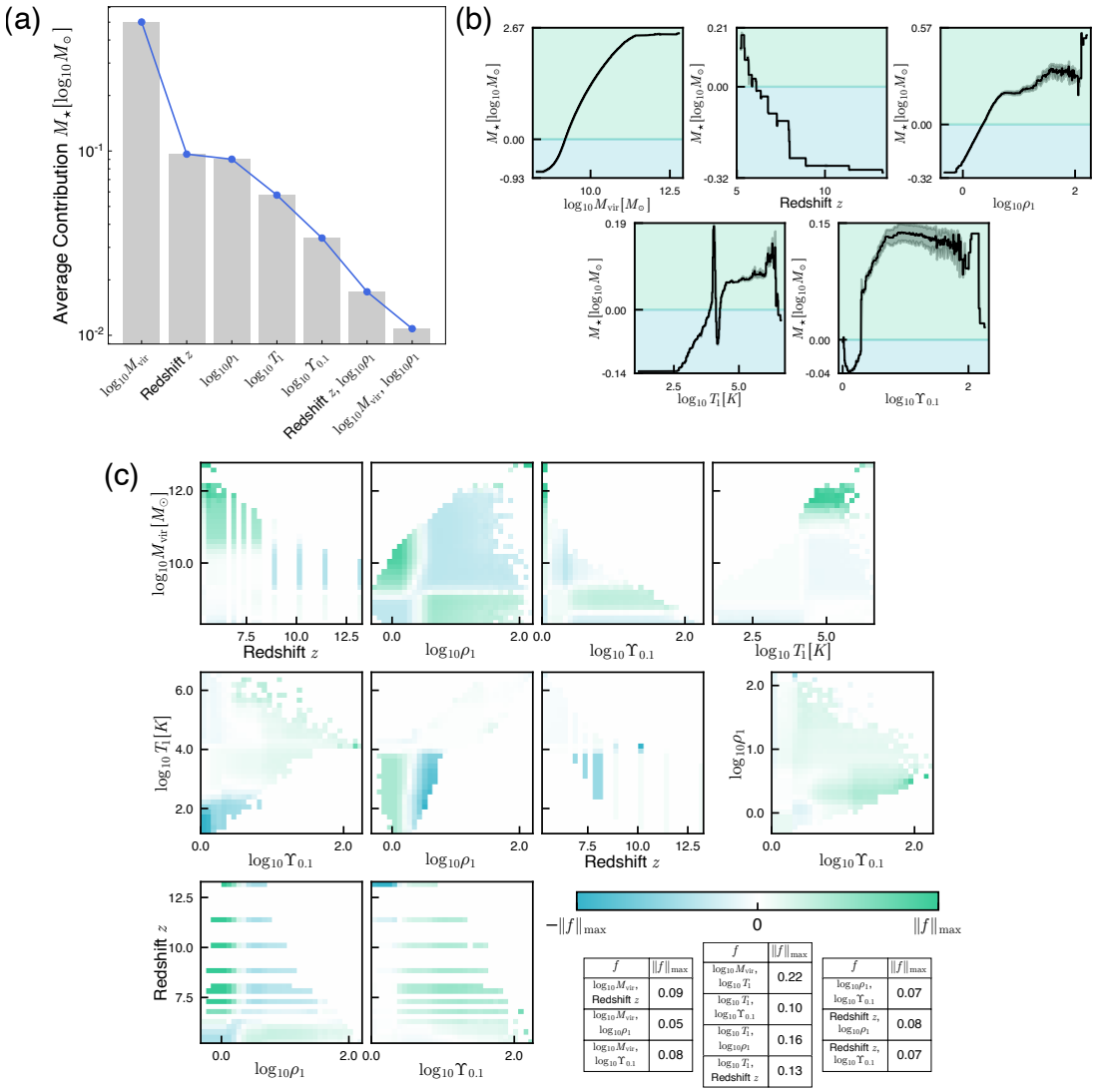

Figure 5.18: Details for the CEBM model $\Gamma(M_\star|\boldsymbol{\theta}')$ trained to predict stellar mass $M_\star$. Panel a) displays the average contribution of features to the CEBM model $\Gamma(M_\star|\boldsymbol{\theta}')$. Panel b) shows the feature functions contributing to the CEBM $\Gamma(M_\star|\boldsymbol{\theta}')$. Panel c) presents the interaction functions for the CEBM $\Gamma(M_\star|\boldsymbol{\theta}')$. Each panel shows the contribution of the interaction terms, normalized such that the color map ranges between plus or minus the maximum of the norm of each function $||f||_{\rm max}$. Teal indicates negative contributions and green indicates positive contributions. The table lists $||f||_{\rm max}$ for the interaction functions, each with units $\log_{10} M_\odot$.

Chapter 6

Conclusion

The large volume of richly complex data in astronomy is beyond the means of human labor alone. The development of tools and techniques to address the challenge of analyzing this data will be essential to furthering our understanding of the universe. As demonstrated in this work, the field of computer science is advanced in working on these challenges as well.

Chapter 2 introduced the Morpheus deep learning framework, a new approach to morphological classification that detects and classifies sources at the pixel level. Pixel-level morphological classifications will become increasingly crucial as the number of sources observed increases and their proximity decreases. Morpheus has been applied to the entire CANDELS survey encompassing an area of over two billion pixels. Further, the framework can quickly scale any pixel-level classification or regression scheme to large astronomical images.

Deblending more than two sources in an image was previously unapproached using deep learning owing to the varying dimensions of the deblended representation of sources. Chapter 3 proposed a new computer vision problem setting called Partial-Attribution Instance Segmentation and implemented a deep learning method to deblend a variable number of sources in an image. Further, the method

presented naturally integrates with the Morpheus framework to scale to large image data. Detecting and deblending sources at large scales will be paramount as one of the first steps in analyzing image data.

Chapter 4 presented FitsMap, an end-to-end tool for displaying astronomical image and catalog data using a web browser. FitsMap preclusters and tiles catalog data to reduce the computational burden on the server and the client at runtime. Preclustering and tiling the catalog data allows large-scale image and catalog data to be rendered on limited hardware like mobile devices. FitsMap is thoroughly benchmarked and is currently the only tool that can process image and catalog data and generate the HTML, CSS, and JavaScript files for the website. Visualizing and easily sharing image and catalog data will make collaboration and public outreach in astronomy much easier.

In Chapter 5, the relationship between stellar mass, star formation rate, and the physical properties of simulated galaxies are modeled using an Explainable Boosting Machine (EBM). The interpretable nature of EBMs allows scientists to more rigorously interrogate the model than other more opaque models like neural networks. Further, an ensemble method called a Composite Explainable Boosting Machine is proposed that can model outlier trends in data while maintaining a comparable level of interpretability. The trained EBMs will model stellar mass and star formation rate in simulated galaxies where that information is not available.

Astronomical data analysis is in an ideal position for computer scientists to work on. Large volumes of heterogeneous numerical data are being generated with vast amounts on the horizon. Working with astronomers, computer scientists not only stand to advance computer science but also perhaps learn how we all got here in the first place.

Appendix A

Deep Learning & Morpheus Code/Data Products

A.1 Deep Learning

The Morpheus(HR2020) deep learning framework incorporates a variety of technologies developed for machine learning applications. The following descriptions of deep learning techniques complement the overview of HR2020 provided in Section 2.2, and are useful for understanding optional configurations of the model.

A.1.1 Artificial Neuron

The basic unit of the HR2020 neural network is the artificial neuron (AN), which transforms an input vector $\mathbf{x}$ to a single output $AN(\mathbf{x})$. The AN is designed to mimic the activation of a neuron, producing a nonlinear response to an input stimulus value when it exceeds a rough threshold.

The first stage of an AN consists of a function

$$z(\mathbf{x}) = \sum_{i=1}^{n} w_i x_i + b \tag{A.1}$$

that adds the dot product of the n-element vector $\mathbf{x}$ with a vector of weights $\mathbf{w}$ to a bias b. The values of the $\mathbf{w}$ elements and b are parameters of the model that are set during optimization. The function $z(\mathbf{x})$ is equivalent to a linear transformation on input data $\mathbf{x}$.

In the second stage, a nonlinear function a is applied to the output of $z(\mathbf{x})$. It is written as

$$AN(\mathbf{x}) \equiv a(z(\mathbf{x})), \tag{A.2}$$

where $a(z)$ is called the activation function. The HR2020 framework allows the user to specify the activation function, including the sigmoid

$$\text{sigmoid}(z) = \frac{1}{1 + e^{-z}}, \tag{A.3}$$

the hyperbolic tangent

$$\tanh(z) = \frac{e^{z} - e^{-z}}{e^{z} + e^{-z}}, \tag{A.4}$$

and the rectified linear unit

$$\text{relu}(z) = \max(0, z). \tag{A.5}$$

These functions share a thresholding behavior, such that the function activates a nonlinear behavior at a characteristic value of z, but the domain of these functions differ. For the morphological classification problem presented in this chapter, the rectified linear unit (Equation A.5) was used as the activation function.

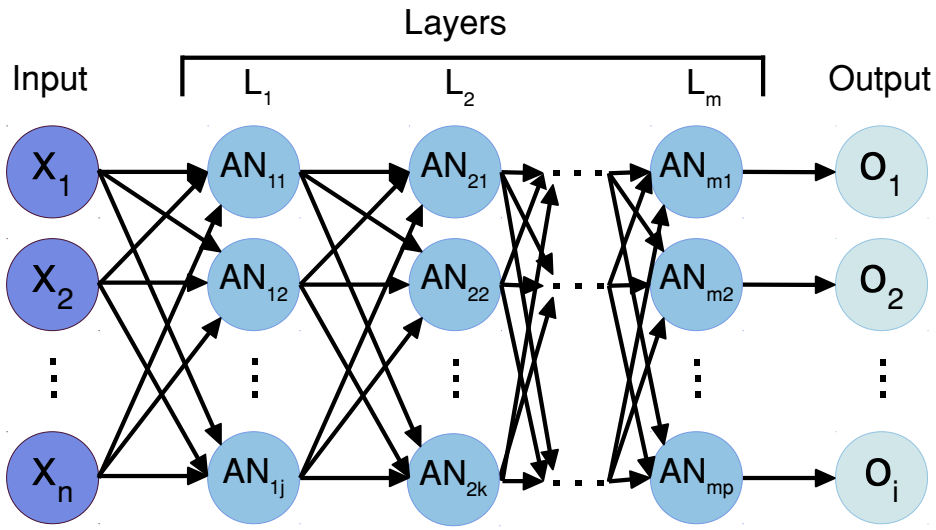

Figure A.1: Schematic of a simple neural network. Given an input vector **x**, the neural network applies a series of reductions and nonlinear transformations through a collection of layers **L** to produce an output **o**. Each layer L consists of a set of artificial neurons **AN** that perform a linear rescaling of their input data, followed by a nonlinear transformation via the application of an activation function (see Equation A.2). The activation function may vary across layers.

A.1.2 Neural Networks

Increasingly complex computational structures can be constructed from ANs. Single ANs are combined into layers, which are collections of distinct ANs that process the same input vector x. A collection of layers forms a neural network (NN), with the layers ordered such that the outputs from one layer provide the inputs to the neurons in the subsequent layer. Figure A.1 shows a schematic of a NN and how the initial input vector x is processed by multiple layers. As shown, these layers are commonly called fully-connected since each neuron in a given layer receives the outputs z from all neurons in the previous layer.

A.1.3 Convolutional Neural Networks

The HR2020 framework operates on image data with a convolutional neural network (CNN). A CNN includes at least one layer of ANs whose z function uses a discrete cross-correlation (convolution) in place of the dot product in Equation A.1. For a convolutional artificial neuron (CAN), it is written as

$$z(\mathbf{X}) = (\mathbf{X} * \mathbf{W}) + b\mathbf{J}, \tag{A.6}$$

where $\mathbf{X} * \mathbf{W}$ represents the convolution of an input image $\mathbf{X}$ and a kernel $\mathbf{W}$. The elements of the kernel $\mathbf{W}$ are parameters of the model, and $\mathbf{W}$ may differ in dimensions from $\mathbf{X}$. In HR2020, the dimensions of $\mathbf{W}$ are set to be 3×3 throughout. The bias b is a scalar as before, and $\mathbf{J}$ represents a matrix of 1s with the same dimensions as the result of the convolution. In HR2020, the convolution is zero-padded to maintain the dimensions of the input data.

The activation function of the neuron is computed element-wise after the convolution and bias have been applied to the input. It is written as

$$CAN(\mathbf{X}) \equiv a(z(\mathbf{X})). \tag{A.7}$$

The output from a CAN is referred to as a feature map.

As with fully-connected layers, convolutional layers consist of a group of CANs that process the same input data $\mathbf{X}$. Convolutional layers can also be arranged sequentially such that the output from one convolutional layer serves as input to the next. HR2020' neural network architecture, being U-Net based, is comprised of CANs (see Figure A.2 for a schematic). In typical convolutional neural network topologies, CANs are used to extract features from input images. The resulting feature maps are eventually flattened into a single vector and processed by a fully

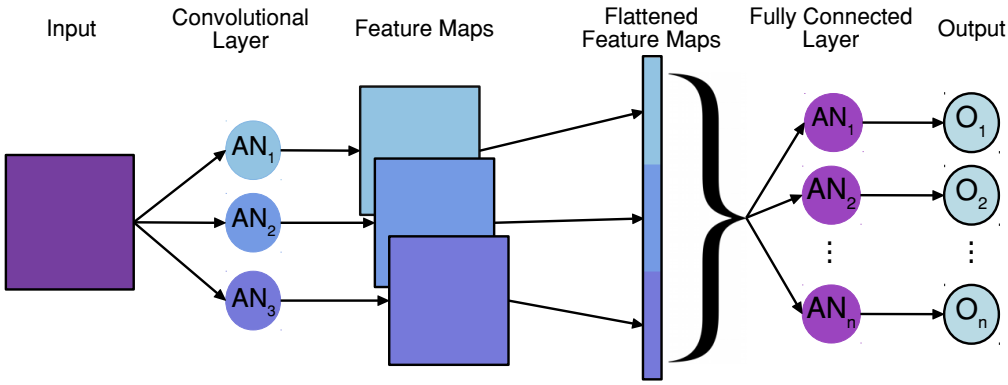

Figure A.2: Schematic of a convolutional neural network (CNN). Shown is a simplified CNN consisting of a convolutional layer feeding a fully connected layer. Each artificial neuron (AN) in the convolutional layer outputs a feature map as described by Equation A.7. Each output feature map is flattened and concatenated into a single vector. This vector is processed by each AN in the fully connected layer (see Equation A.2). The curly brace represents connections from all elements of the vector input.

connected layer to produce the output classification values.

A.1.4 Other Functions in Neural Networks

The primary computational elements of HR2020 are a convolutional neural network (Section A.1.3) and a fully connected layer (Section A.1.2). In detail, other layers are used to reformat or summarize the data, renormalize it, or combine data from different stages in the network.

Pooling

Pooling layers (Figure A.3) are composed of functions that summarize their input data to reduce its size while preserving some information. These layers perform a moving average (average pooling) or maximum (max pooling) over a window of data elements, repeating these reductions as the window scans through

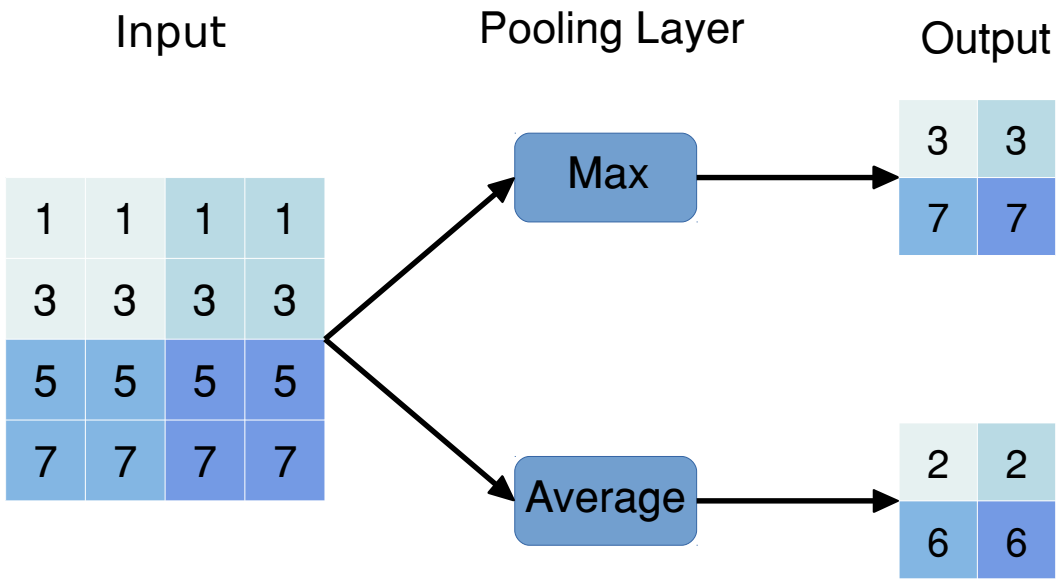

Figure A.3: Comparison of max and average pooling layers. Pooling layers perform reductions on subsets of feature maps, providing a local average or maximum of data elements in a window (2×2 in this schematic). Shown are cells of an input feature map (left), color-coded within a window to match the corresponding regions of the output feature map (right). The pooling layers perform a simple reduction with these windows, taking either a maximum (upper branch) or average (lower branch).

the input image with a stride equal to the window size. In the morphological classification tasks described in Chapter 2, HR2020 uses 2×2 windows and max pooling.

Up-sampling

Up-sampling layers expand the size of feature maps by a specified factor through an interpolation between input data elements. The up-sampling layers operate in the image dimensions of the feature map and typically employ bicubic and bilinear interpolation. In the morphological classification application explored in Chapter 2, HR2020 used 2×2 up-sampling and bicubic interpolation.

Concatenation

Concatenation layers combine multiple feature maps by appending them without changing their contents. For instance, the concatenation of red, green, and blue (RGB) channels into a three-color image would append three $N \times M$ images into an RGB image with dimensions $N \times M \times 3$. This operation is used in HR2020 to combine together data from the contraction phase with the output from bicubic interpolations in the expansion phase (see Figure 2.2).

Batch Normalization

A common pre-processing step for neural network architectures is to normalize the input data x using, e.g., the operation

$$\hat{x} = (x - \mu)/\sqrt{\sigma^2} \tag{A.8}$$

where $\hat{x}$ is the normalized data, and μ and σ are parameters of the model. Ioffe and Szegedy [121] extended this normalization step to apply to the inputs of layers within the network, such that activations (AN) and feature maps (CAN) are normalized over each batch. A batch consists of a subset of the training examples used during the training process. Simple normalization operations like Equation A.8 can reduce the range of values represented in the data provided to a layer, which can inhibit learning. Ioffe and Szegedy [121] addressed this issue by providing an alternative normalization operation that introduces additional parameters to be learned during training. The input data elements x_i are first rescaled as

$$\hat{x_i} = \frac{x_i - \mu_{\mathbf{x}}}{\sqrt{\sigma_{\mathbf{x}}^2 + \epsilon}}. \tag{A.9}$$

Here, x_i is a single element from the data output by a single AN or CAN over a batch, $\mu_{\mathbf{x}}$ is their mean, and $\sigma^2_{\mathbf{x}}$ is their variance. The parameter ϵ is learned during optimization. The new normalization $BN_{\hat{x}_i}$ is then taken to be a linear transformation

$$BN_{\hat{x}_i} = \gamma_x \hat{x}_i + \beta_x. \tag{A.10}$$

The parameters γ_x and β_x are also learned during optimization. Ioffe and Szegedy [121] demonstrated that batch normalization, in the form of Equation A.10, can increase overall accuracy and decrease training time, and this approach is addopted in the HR2020 framework.

A.1.5 U-Net Architecture

The HR2020 framework uses a U-Net architecture, first introduced by Ronneberger et al. [224]. The U-Net architecture was originally designed for segmentation of medical imagery, but has enjoyed success in other fields. The U-Net takes as input a set of images and outputs a classification image of pixel-level probability distributions. The architecture begins with a contraction phase composed of a series of convolutional and pooling layers, followed by an expansion phase composed of a series of convolutional and up-sampling layers. Each of the outputs from the down-sampling layers is concatenated with the output of an up-sampling layer when the height and width dimensions of the feature maps match. These concatenations help preserve the locality of learned features in the output of the NN.

A.2 code release

The code for HR2020 has been release via GitHub (https://github.com/morpheus-project/morpheus). HR2020 is also available as a python package installable via pip (https://pypi.org/project/morpheus-astro/) and as Docker images available via Docker Hub (https://hub.docker.com/r/morpheusastro/morpheus). HR2020 includes both a Python API and a command-line interface, the documentation of which can be found online at https://morpheus-astro.readthedocs.io/en/latest/.

A.3 code tutorial

An online tutorial demonstrating the HR2020 Python API in the form of a Jupyter notebook can be found at https://github.com/morpheus-project/morpheus/blob/master/examples/example_array.ipynb. The tutorial walks through the classification of an example image. Additionally, the tutorial explores other features of HR2020, including generating segmentation maps and morphological catalogs.

A.4 Data Release

The data release associated with this chapter consists of multiple data products. For each field in the CANDELS survey, I provide the following data products: pixel-level morphological classifications, segmentation maps, and value-added catalogs(see also Section 2.8) for the 3D-HST catalogs. Tables A.1-A.5 provide the URLs for each of the data products. Each of the fields has two types of segmentation maps, a segmentation map informed by the 3D-HST survey and a segmentation map informed only by the background values provided by HR2020

(see Algorithm 1). The classifications for the EGS and UDS fields may vary as a result of using the $F814W$ band in place of the $F850LP$ due to availability.

An interactive online visualization of the HST images, HR2020 classification images, and 3D-HST sources is available at https://morpheus-project.github.io/morpheus/.

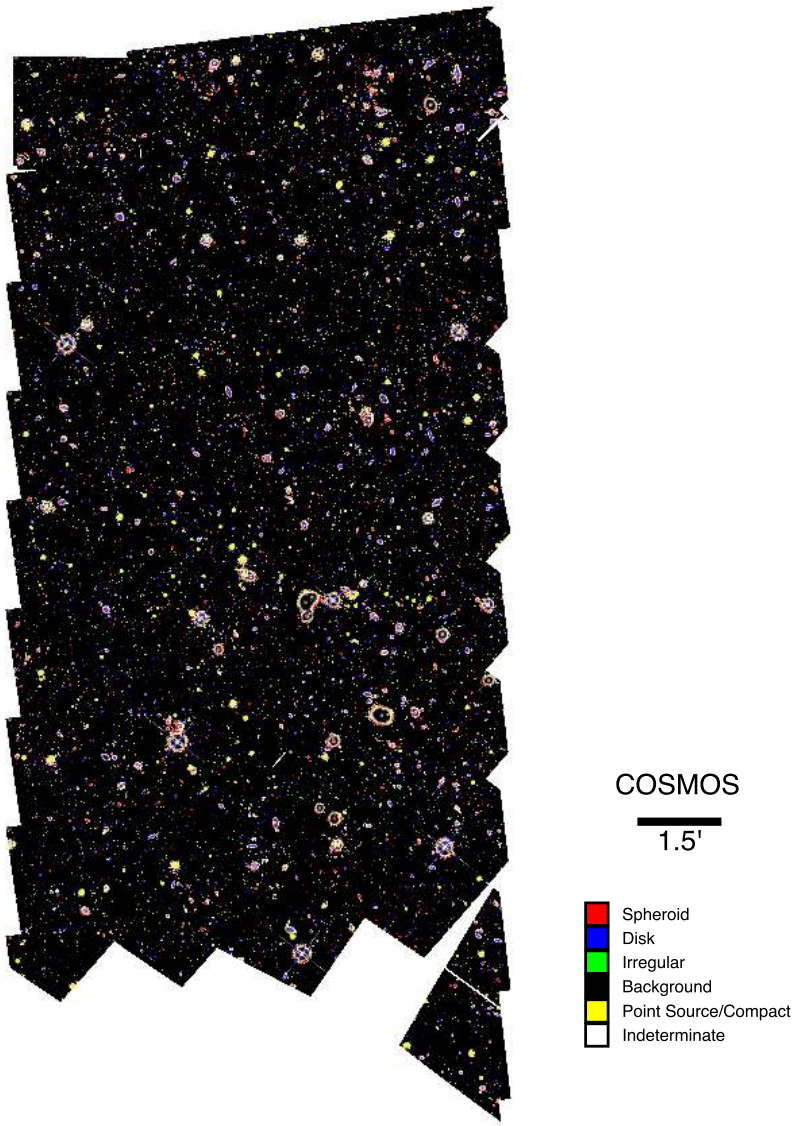

Figure A.4: Color composite of the HR2020 morphological classifications for the COSMOS field from the CANDELS survey [95, 144].

Figure A.5: Color composite of the HR2020 morphological classifications for the EGS field from the CANDELS survey [95, 144].

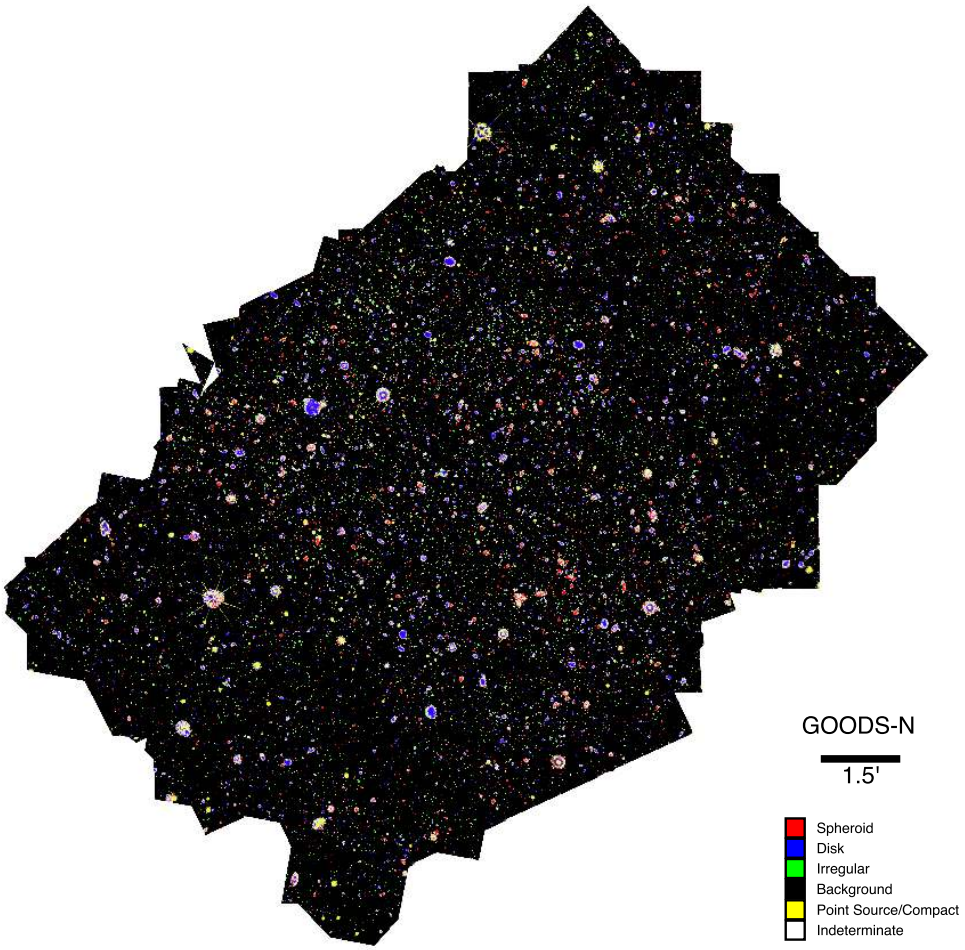

Figure A.6: Color composite of the HR2020 morphological classifications for the GOODS North field from the CANDELS survey [95, 144].

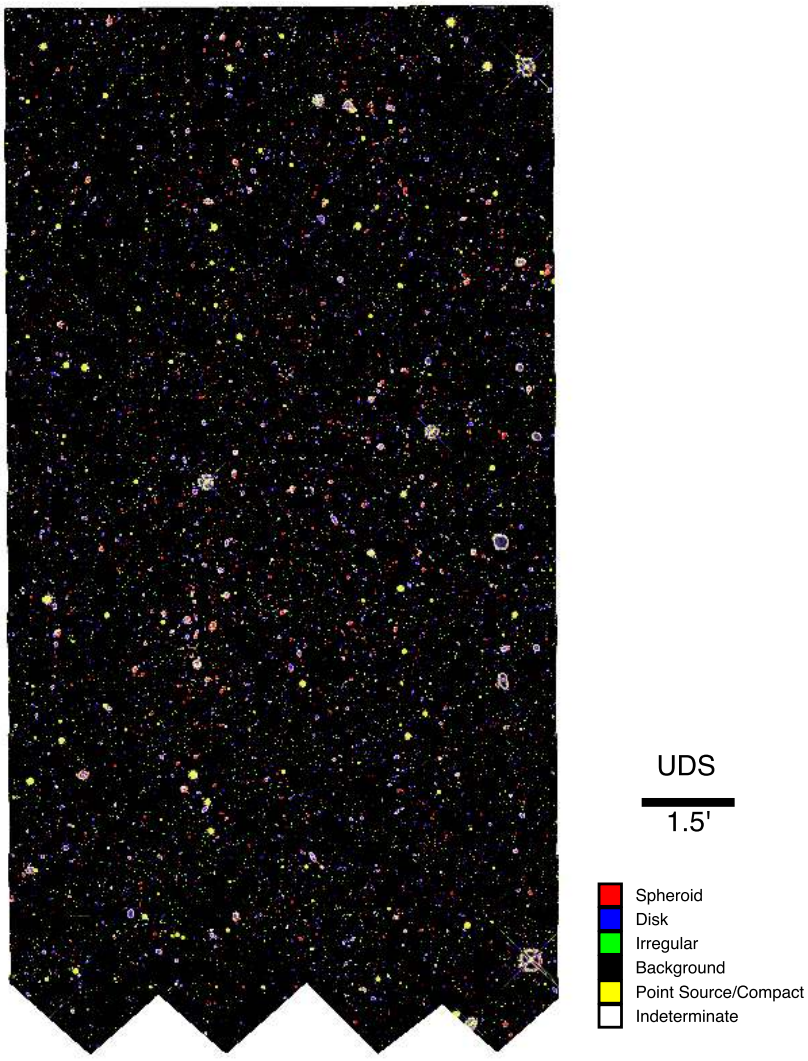

Figure A.7: Color composite of the HR2020 morphological classifications for the UDS field from the CANDELS survey [95, 144].

HR2020 Data Products for the COSMOS Field

File Name	URL
Pixel-level Morphological Classifications	
morpheus_COSMOS_spheroid.v1.0.fits	u/spheroid.html
morpheus_COSMOS_disk.v1.0.fits	u/disk.html
morpheus_COSMOS_irregular.v1.0.fits	u/irregular.html
morpheus_COSMOS_ps_compact.v1.0.fits	u/ps_compact.html
morpheus_COSMOS_background.v1.0.fits	u/background.html
morpheus_COSMOS_mask.v1.0.fits	u/mask.html
Segmentation Maps	
morpheus_COSMOS_segmap.v1.0.fits	u/segmap.html
morpheus_COSMOS_3dhst-segmap.v1.0.fits	u/3dhst-segmap.html
3D-HST Value Added Catalog	
morpheus_COSOMS_3dhst_catalog.v1.0.csv	u/value-added-catalog.html
morpheus_COSOMS_3dhst_catalog.v1.0.txt	u/value-added-catalog-mrt.html
All Files	
morpheus_COSMOS_all.v1.0.tar.gz	u/all.html

u = morpheus-project.github.io/morpheus/data-release/cosmos

Table A.1: Data release files generated by Morpheus and associated URLs for the COSMOS CANDELS field. The data release files for each field are organized into three groups: *pixel-level morphological classifications*, *segmentation maps*, and *3D-HST value-added catalogs*. The *pixel-level morphological classification* files are named according to the following scheme `morpheus_COSMOS_[morphology].v1.0.fits`, where `[morphology]` can be one of the morphological classes (*spheroid*, *disk*, *irregular*, *ps_compact*, *background*) or *mask*, a binary image mask indicating which pixels in the image were classified by HR2020. The *segmentation map* files are named according to the following scheme `morpheus_COSMOS_[segmap_type].v1.0.fits` , where `[segmap_type]` can be *3dhst-segmap* (indicating the 3D-HST informed segmap) or *segmap* (indicating a segmap based only on background class/flux values). Finally, the 3D-HST value-added catalog files are named according to the following scheme `morpheus_COSMOS_3dhst-catalog.v1.0.[file_type]`, where `[file_type]` can be *csv* for a comma-separated-value version of the value-added catalog and *txt* for the machine-readable table version described in Table 2.5. Additionally, a link to an archive containing all of the files associated with the COSMOS field is available in an additional section called *All Files*. See Appendix A.4 for details.

HR2020 Data Products for the EGS Field

File Name	URL
Pixel-level Morphological Classifications	
morpheus_EGS_spheroid.v1.0.fits	u/spheroid.html
morpheus_EGS_disk.v1.0.fits	u/disk.html
morpheus_EGS_irregular.v1.0.fits	u/irregular.html
morpheus_EGS_ps_compact.v1.0.fits	u/ps_compact.html
morpheus_EGS_background.v1.0.fits	u/background.html
morpheus_EGS_mask.v1.0.fits	u/mask.html
Segmentation Maps	
morpheus_EGS_segmap.v1.0.fits	u/segmap.html
morpheus_EGS_3dhst-segmap.v1.0.fits	u/3dhst-segmap.html
3D-HST Value Added Catalogs	
morpheus_EGS_3dhst_catalog.v1.0.csv	u/value-added-catalog.html
morpheus_EGS_3dhst_catalog.v1.0.txt	u/value-added-catalog-mrt.html
All Files	
morpheus_EGS_all.v1.0.tar.gz	u/all.html

u = morpheus-project.github.io/morpheus/data-release/egs

Table A.2: Data release files generated by Morpheus and associated URLs for the EGS CANDELS field. The data release files for each field are organized into three groups: *pixel-level morphological classifications*, *segmentation maps*, and *3D-HST value-added catalogs*. The *pixel-level morphological classification* files are named according to the following scheme `morpheus_EGS_[morphology].v1.0.fits`, where `[morphology]` can be one of the morphological classes (*spheroid*, *disk*, *irregular*, *ps_compact*, *background*) or *mask*, a binary image mask indicating which pixels in the image were classified by HR2020. The *segmentation map* files are named according to the following scheme `morpheus_EGS_[segmap_type].v1.0.fits` , where `[segmap_type]` can be *3dhst-segmap* (indicating the 3D-HST informed segmap) or *segmap* (indicating a segmap based only on background class/flux values). Finally, the 3D-HST value-added catalog files are named according to the following scheme `morpheus_EGS_3dhst-catalog-.v1.0.[file_type]`, where `[file_type]` can be *csv* for a comma-separated-value version of the value-added catalog and *txt* for the machine-readable table version described in Table 2.5. Additionally, a link to an archive containing all of the files associated with the EGS field is available in an additional section called *All Files*. See Appendix A.4 for details.

HR2020 Data Products for the GOODS North Field

File Name	URL
Pixel-level Morphological Classifications	
morpheus_GOODS-N_spheroid.v1.0.fits	u/spheroid.html
morpheus_GOODS-N_disk.v1.0.fits	u/disk.html
morpheus_GOODS-N_irregular.v1.0.fits	u/irregular.html
morpheus_GOODS-N_ps_compact.v1.0.fits	u/ps_compact.html
morpheus_GOODS-N_background.v1.0.fits	u/background.html
morpheus_GOODS-N_mask.v1.0.fits	u/mask.html
Segmentation Maps	
morpheus_GOODS-N_segmap.v1.0.fits	u/segmap.html
morpheus_GOODS-N_3dhst-segmap.v1.0.fits	u/3dhst-segmap.html
3D-HST Value Added Catalogs	
morpheus_GOODS-N_3dhst_catalog.v1.0.csv	u/value-added-catalog.html
morpheus_GOODS-N_3dhst_catalog.v1.0.txt	u/value-added-catalog-mrt.html
All Files	
morpheus_GOODS-N_all.v1.0.tar.gz	u/all.html

u = morpheus-project.github.io/morpheus/data-release/goods-n

Table A.3: Data release files generated by Morpheus and associated URLs for the GOODS North CANDELS field. The data release files for each field are organized into three groups: *pixel-level morphological classifications*, *segmentation maps*, and *3D-HST value-added catalogs*. The *pixel-level morphological classification* files are named according to the following scheme `morpheus_GOODS-N_[morphology].v1.0.fits`, where `[morphology]` can be one of the morphological classes (*spheroid*, *disk*, *irregular*, *ps_compact*, *background*) or *mask*, a binary image mask indicating which pixels in the image were classified by HR2020. The *segmentation map* files are named according to the following scheme `morpheus_GOODS-N_[segmap_type].v1.0.fits` , where `[segmap_type]` can be *3dhst-segmap* (indicating the 3D-HST informed segmap) or *segmap* (indicating a segmap based only on background class/flux values). Finally, the 3D-HST value-added catalog files are named according to the following scheme `morpheus_GOODS-N_3dhst-catalog.v1.0.[file_type]`, where `[file_type]` can be *csv* for a comma-separated-value version of the value-added catalog and *txt* for the machine-readable table version described in Table 2.5. Additionally, a link to an archive containing all of the files associated with the GOODS North field is available in an additional section called *All Files*. See Appendix A.4 for details.

HR2020 Data Products for the GOODS South Field

File Name	URL
Pixel-level Morphological Classifications	
morpheus_GOODS-S_spheroid.v1.0.fits	u/spheroid.html
morpheus_GOODS-S_disk.v1.0.fits	u/disk.html
morpheus_GOODS-S_irregular.v1.0.fits	u/irregular.html
morpheus_GOODS-S_ps_compact.v1.0.fits	u/ps_compact.html
morpheus_GOODS-S_background.v1.0.fits	u/background.html
morpheus_GOODS-S_mask.v1.0.fits	u/mask.html
morpheus_GOODS-S_spheroid.v1.0.fits	u/spheroid.html
Segmentation Maps	
morpheus_GOODS-S_segmap.v1.0.fits	u/segmap.html
morpheus_GOODS-S_3dhst_segmap.v1.0.fits	u/3dhst-segmap.html
3D-HST Value Added Catalogs	
morpheus_GOODS-S_3dhst_catalog.v1.0.csv	u/value-added-catalog.html
morpheus_GOODS-S_3dhst_catalog.v1.0.txt	u/value-added-catalog-mrt.html
All Files	
morpheus_GOODS-S_all.v1.0.tar.gz	u/all.html

u = morpheus-project.github.io/morpheus/data-release/goods-s

Table A.4: Data release files generated by Morpheus and associated URLs for the GOODS South CANDELS field. The data release files for each field are organized into three groups: *pixel-level morphological classifications*, *segmentation maps*, and *3D-HST value-added catalogs*. The *pixel-level morphological classification* files are named according to the following scheme `morpheus_GOODS-S_[morphology].v1.0.fits`, where `[morphology]` can be one of the morphological classes (*spheroid*, *disk*, *irregular*, *ps_compact*, *background*) or *mask*, a binary image mask indicating which pixels in the image were classified by HR2020. The *segmentation map* files are named according to the following scheme `morpheus_GOODS-S_[segmap_type].v1.0.fits` , where `[segmap_type]` can be *3dhst-segmap* (indicating the 3D-HST informed segmap) or *segmap* (indicating a segmap based only on background class/flux values). Finally, the 3D-HST value-added catalog files are named according to the following scheme `morpheus_GOODS-S_3dhst-catalog.v1.0.[file_type]`, where `[file_type]` can be *csv* for a comma-separated-value version of the value-added catalog and *txt* for the machine-readable table version described in Table 2.5. Additionally, a link to an archive containing all of the files associated with the GOODS South field is available in an additional section called *All Files*. See Appendix A.4 for details.

HR2020 Data Products for the UDS Field

File Name	URL
Pixel-level Morphological Classifications	
morpheus_UDS_spheroid.v1.0.fits	u/spheroid.html
morpheus_UDS_disk.v1.0.fits	u/disk.html
morpheus_UDS_irregular.v1.0.fits	u/irregular.html
morpheus_UDS_ps_compact.v1.0.fits	u/ps_compact.html
morpheus_UDS_background.v1.0.fits	u/background.html
morpheus_UDS_mask.v1.0.fits	u/mask.html
Segmentation Maps	
morpheus_UDS_segmap.v1.0.fits	u/segmap.html
morpheus_UDS_3dhst-segmap.v1.0.fits	u/3dhst-segmap.html
3D-HST Value Added Catalogs	
morpheus_UDS_3dhst_catalog.v1.0.csv	u/value-added-catalog.html
morpheus_UDS_3dhst_catalog.v1.0.txt	u/value-added-catalog-mrt.html
All Files	
morpheus_UDS_all.v1.0.tar.gz	u/all.html

u = morpheus-project.github.io/morpheus/data-release/uds

Table A.5: Data release files generated by Morpheus and associated URLs for the UDS CANDELS field. The data release files for each field are organized into three groups: *pixel-level morphological classifications*, *segmentation maps*, and *3D-HST value-added catalogs*. The *pixel-level morphological classification* files are named according to the following scheme `morpheus_UDS_[morphology].v1.0.fits`, where `[morphology]` can be one of the morphological classes (*spheroid*, *disk*, *irregular*, *ps_compact*, *background*) or *mask*, a binary image mask indicating which pixels in the image were classified by HR2020. The *segmentation map* files are named according to the following scheme `morpheus_UDS_[segmap_type].v1.0.fits` , where `[segmap_type]` can be *3dhst-segmap* (indicating the 3D-HST informed segmap) or *segmap* (indicating a segmap based only on background class/flux values). Finally, the 3D-HST value-added catalog files are named according to the following scheme `morpheus_UDS_3dhst-catalog.v1.0.[file_type]`, where `[file_type]` can be *csv* for a comma-separated-value version of the value-added catalog and *txt* for the machine-readable table version described in Table 2.5. Additionally, a link to an archive containing all of the files associated with the UDS field is available in an additional section called *All Files*. See Appendix A.4 for details.

Bibliography

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mane, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viegas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems. *arXiv e-prints*, art. arXiv:1603.04467, Mar 2016.

[2] R. G. Abraham, F. Valdes, H. K. C. Yee, and S. van den Bergh. The morphologies of distant galaxies. 1: an automated classification system. *ApJ*, 432:75–90, September 1994. doi: 10.1086/174550.

[3] R. G. Abraham, N. R. Tanvir, B. X. Santiago, R. S. Ellis, K. Glazebrook, and S. van den Bergh. Galaxy morphology to I=25 mag in the Hubble Deep Field. *MNRAS*, 279:L47–L52, April 1996. doi: 10.1093/mnras/279.3.L47.

[4] Roberto G. Abraham and Sidney van den Bergh. The morphological evolution of galaxies. *Science*, 293(5533):1273–1278, 2001. ISSN 0036-8075. doi: 10.1126/science.1060855. URL `http://science.sciencemag.org/content/293/5533/1273`.

[5] Sheelu Abraham, A. K. Aniyan, Ajit K. Kembhavi, N. S. Philip, and Kaustubh Vaghmare. Detection of bars in galaxies using a deep convolutional neural network. *MNRAS*, 477(1):894–903, June 2018. doi: 10.1093/mnras/sty627.

[6] Adriano Agnello, Brandon C. Kelly, Tommaso Treu, and Philip J. Marshall. Data mining for gravitationally lensed quasars. *MNRAS*, 448(2):1446–1462, Apr 2015. doi: 10.1093/mnras/stv037.

[7] Hiroaki Aihara, Nobuo Arimoto, Robert Armstrong, Stéphane Arnouts, Neta A. Bahcall, Steven Bickerton, James Bosch, Kevin Bundy, Peter L. Capak, and James H. H. Chan. The Hyper Suprime-Cam SSP Survey: Overview and survey design. *PASJ*, 70:S4, Jan 2018. doi: 10.1093/pasj/psx066.

[8] Hiroaki Aihara, Robert Armstrong, Steven Bickerton, James Bosch, Jean Coupon, Hisanori Furusawa, Yusuke Hayashi, Hiroyuki Ikeda, Yukiko Kamata, and Hiroshi Karoji. First data release of the Hyper Suprime-Cam Subaru Strategic Program. *PASJ*, 70:S8, Jan 2018. doi: 10.1093/pasj/psx081.

[9] Rachel Akeson, Lee Armus, Etienne Bachelet, Vanessa Bailey, Lisa Bartusek, Andrea Bellini, Dominic Benford, David Bennett, Aparna Bhattacharya, Ralph Bohlin, Martha Boyer, Valerio Bozza, Geoffrey Bryden, Sebastiano Calchi Novati, Kenneth Carpenter, Stefano Casertano, Ami Choi, David Content, Pratika Dayal, Alan Dressler, Olivier Doré, S. Michael Fall, Xiaohui Fan, Xiao Fang, Alexei Filippenko, Steven Finkelstein, Ryan Foley, Steven Furlanetto, Jason Kalirai, B. Scott Gaudi, Karoline Gilbert, Julien Girard, Kevin Grady, Jenny Greene, Puragra Guhathakurta, Chen Heinrich, Shoubaneh Hemmati, David Hendel, Calen Henderson, Thomas Henning, Christopher Hirata, Shirley Ho, Eric Huff, Anne Hutter, Rolf Jansen, Saurabh Jha, Samson Johnson, David Jones, Jeremy Kasdin, Patrick Kelly, Robert Kirshner, Anton Koekemoer, Jeffrey Kruk, Nikole Lewis, Bruce Macintosh, Piero Madau, Sangeeta Malhotra, Kaisey Mandel, Elena Massara, Daniel Masters, Julie McEnery, Kristen McQuinn, Peter Melchior, Mark Melton, Bertrand Mennesson, Molly Peeples, Matthew Penny, Saul Perlmutter, Alice Pisani, Andrés Plazas, Radek Poleski, Marc Postman, Clément Ranc, Bernard Rauscher, Armin Rest, Aki Roberge, Brant Robertson, Steven Rodney, James Rhoads, Jason Rhodes, Jr. Ryan, Russell, Kailash Sahu, David Sand, Dan Scolnic, Anil Seth, Yossi Shvartzvald, Karelle Siellez, Arfon Smith, David Spergel, Keivan Stassun, Rachel Street, Louis-Gregory Strolger, Alexander Szalay, John Trauger, M. A. Troxel, Margaret Turnbull, Roeland van der Marel, Anja von der Linden, Yun Wang, David Weinberg, Benjamin Williams, Rogier Windhorst, Edward Wollack, Hao-Yi Wu, Jennifer Yee, and Neil Zimmerman. The Wide Field Infrared Survey Telescope: 100 Hubbles for the 2020s. *arXiv e-prints*, February 2019.

[10] R. J. Allen, G. G. Kacprzak, K. Glazebrook, I. Labbé, K.-V. H. Tran, L. R. Spitler, M. Cowley, T. Nanayakkara, C. Papovich, R. Quadri, C. M. S. Straatman, V. Tilvi, and P. van Dokkum. The Size Evolution of Star-forming Galaxies since z ˜ 7 Using ZFOURGE. *ApJL*, 834:L11, January 2017. doi: 10.3847/2041-8213/834/2/L11.

[11] A. K. Aniyan and K. Thorat. Classifying Radio Galaxies with the Convolutional Neural Network. *ApJS*, 230(2):20, June 2017. doi: 10.3847/1538-4365/aa7333.

[12] M. Araya, M. Osorio, M. Díaz, C. Ponce, M. Villanueva, C. Valenzuela, and M. Solar. JOVIAL: Notebook-based astronomical data analysis in the cloud. *Astronomy and Computing*, 25:110–117, October 2018. doi: 10.1016/j.ascom.2018.09.001.

[13] M. Argudo-Fernández, S. Duarte Puertas, J. E. Ruiz, J. Sabater, S. Verley, and G. Bergond. LSSGalPy: Interactive Visualization of the Large-scale Environment Around Galaxies. *PASP*, 129(975):058005, May 2017. doi: 10.1088/1538-3873/aa5785.

[14] Astropy Collaboration, T. P. Robitaille, E. J. Tollerud, P. Greenfield, M. Droettboom, E. Bray, T. Aldcroft, M. Davis, A. Ginsburg, A. M. Price-Whelan, W. E. Kerzendorf, A. Conley, N. Crighton, K. Barbary, D. Muna, H. Ferguson, F. Grollier, M. M. Parikh, P. H. Nair, H. M. Unther, C. Deil, J. Woillez, S. Conseil, R. Kramer, J. E. H. Turner, L. Singer, R. Fox, B. A. Weaver, V. Zabalza, Z. I. Edwards, K. Azalee Bostroem, D. J. Burke, A. R. Casey, S. M. Crawford, N. Dencheva, J. Ely, T. Jenness, K. Labrie, P. L. Lim, F. Pierfederici, A. Pontzen, A. Ptak, B. Refsdal, M. Servillat, and O. Streicher. Astropy: A community Python package for astronomy. *A&A*, 558:A33, October 2013. doi: 10.1051/0004-6361/201322068.

[15] Astropy Collaboration, A. M. Price-Whelan, B. M. Sipőcz, H. M. Günther, P. L. Lim, S. M. Crawford, S. Conseil, D. L. Shupe, M. W. Craig, N. Dencheva, A. Ginsburg, J. T. Vand erPlas, L. D. Bradley, D. Pérez-Suárez, M. de Val-Borro, T. L. Aldcroft, K. L. Cruz, T. P. Robitaille, E. J. Tollerud, C. Ardelean, T. Babej, Y. P. Bach, M. Bachetti, A. V. Bakanov, S. P. Bamford, G. Barentsen, P. Barmby, A. Baumbach, K. L. Berry, F. Biscani, M. Boquien, K. A. Bostroem, L. G. Bouma, G. B. Brammer, E. M. Bray, H. Breytenbach, H. Buddelmeijer, D. J. Burke, G. Calderone, J. L. Cano Rodríguez, M. Cara, J. V. M. Cardoso, S. Cheedella, Y. Copin, L. Corrales, D. Crichton, D. D'Avella, C. Deil, É. Depagne, J. P. Dietrich, A. Donath, M. Droettboom, N. Earl, T. Erben, S. Fabbro, L. A. Ferreira, T. Finethy, R. T. Fox, L. H. Garrison, S. L. J. Gibbons, D. A. Goldstein, R. Gommers, J. P. Greco, P. Greenfield, A. M. Groener, F. Grollier, A. Hagen, P. Hirst, D. Homeier, A. J. Horton, G. Hosseinzadeh, L. Hu, J. S. Hunkeler, Ž. Ivezić, A. Jain, T. Jenness, G. Kanarek, S. Kendrew, N. S. Kern, W. E. Kerzendorf, A. Khvalko, J. King, D. Kirkby, A. M. Kulkarni, A. Kumar, A. Lee, D. Lenz, S. P. Littlefair, Z. Ma, D. M. Macleod, M. Mastropietro, C. McCully, S. Montagnac, B. M. Morris, M. Mueller, S. J. Mumford, D. Muna, N. A. Murphy, S. Nelson, G. H. Nguyen, J. P.

Ninan, M. Nöthe, S. Ogaz, S. Oh, J. K. Parejko, N. Parley, S. Pascual, R. Patil, A. A. Patil, A. L. Plunkett, J. X. Prochaska, T. Rastogi, V. Reddy Janga, J. Sabater, P. Sakurikar, M. Seifert, L. E. Sherbert, H. Sherwood-Taylor, A. Y. Shih, J. Sick, M. T. Silbiger, S. Singanamalla, L. P. Singer, P. H. Sladen, K. A. Sooley, S. Sornarajah, O. Streicher, P. Teuben, S. W. Thomas, G. R. Tremblay, J. E. H. Turner, V. Terrón, M. H. van Kerkwijk, A. de la Vega, L. L. Watkins, B. A. Weaver, J. B. Whitmore, J. Woillez, V. Zabalza, and Astropy Contributors. The Astropy Project: Building an Open-science Project and Status of the v2.0 Core Package. *AJ*, 156(3):123, September 2018. doi: 10.3847/1538-3881/aabc4f.

[16] Michael L. Balogh, Ivan K. Baldry, Robert Nichol, Chris Miller, Richard Bower, and Karl Glazebrook. The Bimodal Galaxy Color Distribution: Dependence on Luminosity and Environment. *ApJL*, 615(2):L101–L104, November 2004. doi: 10.1086/426079.

[17] Melanie R. Beck, Claudia Scarlata, Lucy F. Fortson, Chris J. Lintott, B. D. Simmons, Melanie A. Galloway, Kyle W. Willett, Hugh Dickinson, Karen L. Masters, and Philip J. Marshall. Integrating human and machine intelligence in galaxy morphology classification tasks. *MNRAS*, 476(4):5516–5534, Jun 2018. doi: 10.1093/mnras/sty503.

[18] Burger Becker, Mattia Vaccari, Matthew Prescott, and Trienko Grobler. CNN architecture comparison for radio galaxy classification. *MNRAS*, 503 (2):1828–1846, May 2021. doi: 10.1093/mnras/stab325.

[19] Peter Behroozi, Risa H. Wechsler, Andrew P. Hearin, and Charlie Conroy. UNIVERSEMACHINE: The correlation between galaxy growth and dark matter halo assembly from z = 0-10. *MNRAS*, 488(3):3143–3194, September 2019. doi: 10.1093/mnras/stz1182.

[20] E. F. Bell, A. van der Wel, C. Papovich, D. Kocevski, J. Lotz, D. H. McIntosh, J. Kartaltepe, S. M. Faber, H. Ferguson, A. Koekemoer, N. Grogin, S. Wuyts, E. Cheung, C. J. Conselice, A. Dekel, J. S. Dunlop, M. Giavalisco, J. Herrington, D. C. Koo, E. J. McGrath, D. de Mello, H.-W. Rix, A. R. Robaina, and C. C. Williams. What Turns Galaxies Off? The Different Morphologies of Star-forming and Quiescent Galaxies since z ˜ 2 from CANDELS. *ApJ*, 753:167, July 2012. doi: 10.1088/0004-637X/753/2/167.

[21] R. Bender, D. Burstein, and S. M. Faber. Dynamically hot galaxies. I - Structural properties. *ApJ*, 399:462–477, November 1992. doi: 10.1086/171940.

[22] E. Bertin and S. Arnouts. SExtractor: Software for source extraction. *A&AS*, 117:393–404, June 1996. doi: 10.1051/aas:1996164.

[23] E. Bertin, R. Pillay, and C. Marmo. Web-based visualization of very large scientific astronomy imagery. *Astronomy and Computing*, 10:43–53, April 2015. doi: 10.1016/j.ascom.2014.12.006.

[24] Emmanuel Bertin, Chiara Marmo, and Hervé Bouy. VisiOmatic 2: a Web Client for Remote Visualization With Real-time Mixing of Multispectral Data. In Marco Molinaro, Keith Shortridge, and Fabio Pasian, editors, *Astronomical Data Analysis Software and Systems XXVI*, volume 521 of *Astronomical Society of the Pacific Conference Series*, page 651, October 2019.

[25] R. Bezanson, P. G. van Dokkum, J. van de Sande, M. Franx, J. Leja, and M. Kriek. Tight Correlations between Massive Galaxy Structural Properties and Dynamics: The Mass Fundamental Plane was in Place by z ˜ 2. *ApJL*, 779:L21, December 2013. doi: 10.1088/2041-8205/779/2/L21.

[26] J. Binney. On the rotation of elliptical galaxies. *MNRAS*, 183:501–514, May 1978. doi: 10.1093/mnras/183.3.501.

[27] J. Binney and S. Tremaine. *Galactic dynamics*. Princeton, NJ: Princeton University Press, 1987.

[28] Asa F. L. Bluck, Roberto Maiolino, Simcha Brownson, Christopher J. Conselice, Sara L. Ellison, Joanna M. Piotrowska, and Mallory D. Thorp. The quenching of galaxies, bulges, and disks since cosmic noon. A machine learning approach for identifying causality in astronomical data. *A&A*, 659:A160, March 2022. doi: 10.1051/0004-6361/202142643.

[29] T. Boch and P. Fernique. Aladin Lite: Embed your Sky in the Browser. In N. Manset and P. Forshay, editors, *Astronomical Data Analysis Software and Systems XXIII*, volume 485 of *Astronomical Society of the Pacific Conference Series*, page 277, May 2014.

[30] C. R. Bom, A. Cortesi, G. Lucatelli, L. O. Dias, P. Schubert, G. B. Oliveira Schwarz, N. M. Cardoso, E. V. R. Lima, C. Mendes de Oliveira, L. Sodre, A. V. Smith Castelli, F. Ferrari, G. Damke, R. Overzier, A. Kanaan, T. Ribeiro, and W. Schoenell. Deep Learning assessment of galaxy morphology in S-PLUS Data Release 1. *MNRAS*, 507(2):1937–1955, October 2021. doi: 10.1093/mnras/stab1981.

[31] L. Bonavera, S. L. Suarez Gomez, J. González-Nuevo, M. M. Cueli, J. D. Santos, M. L. Sanchez, R. Muñiz, and F. J. de Cos. Point source detection with fully convolutional networks. Performance in realistic microwave sky simulations. *A&A*, 648:A50, April 2021. doi: 10.1051/0004-6361/201937171.

[32] F. Bonnarel, Ph. Paillou, F. Ochsenbein, M. Creze, and D. Egret. Aladin: Towards an Interactive Atlas of the Digitized Sky. In D. R. Crabtree, R. J. Hanisch, and J. Barnes, editors, *Astronomical Data Analysis Software and Systems III*, volume 61 of *Astronomical Society of the Pacific Conference Series*, page 215, January 1994.

[33] C. Bormann and P. Hoffman. Concise binary object representation (cbor). Rfc, RFC Editor, December 2020. URL `https://www.rfc-editor.org/rfc/rfc8949.html`.

[34] Alexandre Boucaud, Marc Huertas-Company, Caroline Heneka, Emille E. O. Ishida, Nima Sedaghat, Rafael S. de Souza, Ben Moews, Hervé Dole, Marco Castellano, Emiliano Merlin, Valerio Roscani, Andrea Tramacere, Madhura Killedar, Arlindo M. M. Trindade, and Collaboration COIN. Photometry of high-redshift blended galaxies using deep learning. *MNRAS*, 491(2):2481–2495, January 2020. doi: 10.1093/mnras/stz3056.

[35] R. J. Bouwens, G. D. Illingworth, P. A. Oesch, M. Franx, I. Labbé, M. Trenti, P. van Dokkum, C. M. Carollo, V. González, R. Smit, and D. Magee. UV-continuum Slopes at z ∼4-7 from the HUDF09+ERS+CANDELS Observations: Discovery of a Well-defined UV Color-Magnitude Relationship for z >= 4 Star-forming Galaxies. *ApJ*, 754(2):83, August 2012. doi: 10.1088/0004-637X/754/2/83.

[36] Micah Bowles, Anna M. M. Scaife, Fiona Porter, Hongming Tang, and David J. Bastien. Attention-gating for improved radio galaxy classification. *MNRAS*, 501(3):4579–4595, March 2021. doi: 10.1093/mnras/staa3946.

[37] J. Brinchmann, S. Charlot, S. D. M. White, C. Tremonti, G. Kauffmann, T. Heckman, and J. Brinkmann. The physical properties of star-forming galaxies in the low-redshift Universe. *MNRAS*, 351(4):1151–1179, July 2004. doi: 10.1111/j.1365-2966.2004.07881.x.

[38] V. A. Bruce, J. S. Dunlop, M. Cirasuolo, R. J. McLure, T. A. Targett, E. F. Bell, D. J. Croton, A. Dekel, S. M. Faber, H. C. Ferguson, N. A. Grogin, D. D. Kocevski, A. M. Koekemoer, D. C. Koo, K. Lai, J. M. Lotz, E. J. McGrath, J. A. Newman, and A. van der Wel. The morphologies of massive galaxies at $1 < z < 3$ in the CANDELS-UDS field: compact bulges, and the rise and fall of massive discs. *MNRAS*, 427:1666–1701, December 2012. doi: 10.1111/j.1365-2966.2012.22087.x.

[39] V. A. Bruce, J. S. Dunlop, A. Mortlock, D. D. Kocevski, E. J. McGrath, and D. J. Rosario. The bulge-disc decomposition of AGN host galaxies. *MNRAS*, 458:2391–2404, May 2016. doi: 10.1093/mnras/stw467.

[40] Colin J. Burke, Patrick D. Aleo, Yu-Ching Chen, Xin Liu, John R. Peterson, Glenn H. Sembroski, and Joshua Yao-Yu Lin. Deblending and classifying astronomical sources with Mask R-CNN deep learning. *MNRAS*, 490(3): 3952–3965, December 2019. doi: 10.1093/mnras/stz2845.

[41] L. Cabayol, I. Sevilla-Noarbe, E. Fernández, J. Carretero, M. Eriksen, S. Serrano, A. Alarcón, A. Amara, R. Casas, F. J. Castander, J. de Vicente, M. Folger, J. García-Bellido, E. Gaztanaga, H. Hoekstra, R. Miquel, C. Padilla, E. Sánchez, L. Stothert, P. Tallada, and L. Tortorelli. The PAU survey: star-galaxy classification with multi narrow-band data. *MNRAS*, 483(1):529–539, February 2019. doi: 10.1093/mnras/sty3129.

[42] Guillermo Cabrera-Vives, Ignacio Reyes, Francisco Förster, Pablo A. Estévez, and Juan-Carlos Maureira. Deep-HiTS: Rotation Invariant Convolutional Neural Network for Transient Detection. *ApJ*, 836(1):97, February 2017. doi: 10.3847/1538-4357/836/1/97.

[43] João Caldeira, Joshua Job, Steven H. Adachi, Brian Nord, and Gabriel N. Perdue. Restricted Boltzmann Machines for galaxy morphology classification with a quantum annealer. *arXiv e-prints*, art. arXiv:1911.06259, November 2019.

[44] Fernando Caro, Marc Huertas-Company, and Guillermo Cabrera. Morphology and Interaction of Galaxies using Deep Learning. In Massimo Brescia, S. G. Djorgovski, Eric D. Feigelson, Giuseppe Longo, and Stefano Cavuoti, editors, *Astroinformatics*, volume 325, pages 205–208, June 2017. doi: 10.1017/S1743921317000205.

[45] J. M. Casas, J. González-Nuevo, L. Bonavera, D. Herranz, S. L. Suarez Gomez, M. M. Cueli, D. Crespo, J. D. Santos, M. L. Sánchez, F. Sánchez-Lasheras, and F. J. de Cos. Multi-frequency point source detection with fully convolutional networks: Performance in realistic microwave sky simulations. *A&A*, 658:A110, February 2022. doi: 10.1051/0004-6361/202141874.

[46] M. K. Cavanagh and K. Bekki. Bars formed in galaxy merging and their classification with deep learning. *A&A*, 641:A77, September 2020. doi: 10.1051/0004-6361/202037963.

[47] Mitchell K. Cavanagh, Kenji Bekki, and Brent A. Groves. Morphological classification of galaxies with deep learning: comparing 3-way and 4-way CNNs. *MNRAS*, 506(1):659–676, September 2021. doi: 10.1093/mnras/stab1552.

[48] Matthew C. Chan and John P. Stott. Deep-CEE I: fishing for galaxy clusters with deep neural nets. *MNRAS*, 490(4):5770–5787, December 2019. doi: 10.1093/mnras/stz2936.

[49] Bowen Cheng, Maxwell D. Collins, Yukun Zhu, Ting Liu, Thomas S. Huang, Hartwig Adam, and Liang-Chieh Chen. Panoptic-DeepLab: A Simple, Strong, and Fast Baseline for Bottom-Up Panoptic Segmentation. *arXiv e-prints*, art. arXiv:1911.10194, November 2019.

[50] Ting-Yun Cheng, Christopher J. Conselice, Alfonso Aragón-Salamanca, Nan Li, Asa F. L. Bluck, Will G. Hartley, James Annis, David Brooks, Peter Doel, Juan García-Bellido, David J. James, Kyler Kuehn, Nikolay Kuropatkin, Mathew Smith, Flavia Sobreira, and Gregory Tarle. Optimizing automatic morphological classification of galaxies with machine learning and deep learning using Dark Energy Survey imaging. *MNRAS*, 493(3): 4209–4228, April 2020. doi: 10.1093/mnras/staa501.

[51] Ting-Yun Cheng, Christopher J. Conselice, Alfonso Aragón-Salamanca, M. Aguena, S. Allam, F. Andrade-Oliveira, J. Annis, A. F. L. Bluck, D. Brooks, D. L. Burke, M. Carrasco Kind, J. Carretero, A. Choi, M. Costanzi, L. N. da Costa, M. E. S. Pereira, J. De Vicente, H. T. Diehl, A. Drlica-Wagner, K. Eckert, S. Everett, A. E. Evrard, I. Ferrero, P. Fosalba, J. Frieman, J. García-Bellido, D. W. Gerdes, T. Giannantonio, D. Gruen, R. A. Gruendl, J. Gschwend, G. Gutierrez, S. R. Hinton, D. L. Hollowood, K. Honscheid, D. J. James, E. Krause, K. Kuehn, N. Kuropatkin, O. Lahav, M. A. G. Maia, M. March, F. Menanteau, R. Miquel, R. Morgan, F. Paz-Chinchón, A. Pieres, A. A. Plazas Malagón, A. Roodman, E. Sanchez, V. Scarpine, S. Serrano, I. Sevilla-Noarbe, M. Smith, M. Soares-Santos, E. Suchyta, M. E. C. Swanson, G. Tarle, D. Thomas, and C. To. Galaxy morphological classification catalogue of the Dark Energy Survey Year 3 data with convolutional neural networks. *MNRAS*, 507(3):4425–4444, November 2021. doi: 10.1093/mnras/stab2142.

[52] Igor V. Chilingarian, Ivan Yu. Zolotukhin, Ivan Yu. Katkov, Anne-Laure Melchior, Evgeniy V. Rubtsov, and Kirill A. Grishin. RCSED—A Value-added Reference Catalog of Spectral Energy Distributions of 800,299 Galaxies in 11 Ultraviolet, Optical, and Near-infrared Bands: Morphologies, Colors, Ionized Gas, and Stellar Population Properties. *ApJS*, 228(2):14, February 2017. doi: 10.3847/1538-4365/228/2/14.

[53] D. Cireşan, U. Meier, and J. Schmidhuber. Multi-column Deep Neural Networks for Image Classification. *arXiv e-prints*, February 2012.

[54] Liam Connor and Joeri van Leeuwen. Applying Deep Learning to Fast Radio Burst Classification. *AJ*, 156(6):256, December 2018. doi: 10.3847/1538-3881/aae649.

[55] Charlie Conroy, Risa H. Wechsler, and Andrey V. Kravtsov. Modeling Luminosity-dependent Galaxy Clustering through Cosmic Time. *ApJ*, 647 (1):201–214, Aug 2006. doi: 10.1086/503602.

[56] C. J. Conselice. The Relationship between Stellar Light Distributions of Galaxies and Their Formation Histories. *ApJS*, 147:1–28, July 2003. doi: 10.1086/375001.

[57] C. J. Conselice, M. A. Bershady, and A. Jangren. The Asymmetry of Galaxies: Physical Morphology for Nearby and High-Redshift Galaxies. *ApJ*, 529: 886–910, February 2000. doi: 10.1086/308300.

[58] C. J. Conselice, J. A. Blackburne, and C. Papovich. The Luminosity, Stellar Mass, and Number Density Evolution of Field Galaxies of Known Morphology from z = 0.5 to 3. *ApJ*, 620:564–583, February 2005. doi: 10.1086/426102.

[59] E. Contini, Q. Gu, X. Ge, J. Rhee, S. K. Yi, and X. Kang. The Roles of Mass and Environment in the Quenching of Galaxies. II. *ApJ*, 889(2):156, February 2020. doi: 10.3847/1538-4357/ab6730.

[60] M. C. Cooper, R. L. Griffith, J. A. Newman, A. L. Coil, M. Davis, A. A. Dutton, S. M. Faber, P. Guhathakurta, D. C. Koo, J. M. Lotz, B. J. Weiner, C. N. A. Willmer, and R. Yan. The DEEP3 Galaxy Redshift Survey: the impact of environment on the size evolution of massive early-type galaxies at intermediate redshift. *MNRAS*, 419:3018–3027, February 2012. doi: 10.1111/j.1365-2966.2011.19938.x.

[61] Michel Couprie and Gilles Bertrand. Topological gray-scale watershed transformation. In Robert A. Melter, Angela Y. Wu, and Longin J. Latecki, editors, *Vision Geometry VI*, volume 3168 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, pages 136–146, October 1997. doi: 10.1117/12.292778.

[62] Darren J. Croton, Adam R. H. Stevens, Chiara Tonini, Thibault Garel, Maksym Bernyk, Antonio Bibiano, Luke Hodkinson, Simon J. Mutch, Gregory B. Poole, and Genevieve M. Shattow. Semi-Analytic Galaxy Evolution (SAGE): Model Calibration and Basic Results. *ApJS*, 222(2):22, February 2016. doi: 10.3847/0067-0049/222/2/22.

[63] J.-M. Dai and J. Tong. Galaxy Morphology Classification with Deep Convolutional Neural Networks. *ArXiv e-prints*, July 2018.

[64] Dark Energy Survey Collaboration, T. Abbott, F. B. Abdalla, J. Aleksić, S. Allam, A. Amara, D. Bacon, E. Balbinot, M. Banerji, and K. Bechtol.

The Dark Energy Survey: more than dark energy - an overview. *MNRAS*, 460(2):1270–1299, Aug 2016. doi: 10.1093/mnras/stw641.

[65] Romeel Davé, Daniel Anglés-Alcázar, Desika Narayanan, Qi Li, Mika H. Rafieferantsoa, and Sarah Appleby. SIMBA: Cosmological simulations with black hole growth and feedback. *MNRAS*, 486(2):2827–2849, June 2019. doi: 10.1093/mnras/stz937.

[66] L. J. M. Davies, A. S. G. Robotham, S. P. Driver, M. Alpaslan, I. K. Baldry, J. Bland-Hawthorn, S. Brough, M. J. I. Brown, M. E. Cluver, B. W. Holwerda, A. M. Hopkins, M. A. Lara-López, S. Mahajan, A. J. Moffett, M. S. Owers, and S. Phillipps. Galaxy And Mass Assembly (GAMA): growing up in a bad neighbourhood - how do low-mass galaxies become passive? *MNRAS*, 455(4):4013–4029, February 2016. doi: 10.1093/mnras/stv2573.

[67] L. J. M. Davies, A. S. G. Robotham, C. del P. Lagos, S. P. Driver, A. R. H. Stevens, Y. M. Bahé, M. Alpaslan, M. N. Bremer, M. J. I. Brown, S. Brough, J. Bland-Hawthorn, L. Cortese, P. Elahi, M. W. Grootes, B. W. Holwerda, A. D. Ludlow, S. McGee, M. Owers, and S. Phillipps. Galaxy and Mass Assembly (GAMA): environmental quenching of centrals and satellites in groups. *MNRAS*, 483(4):5444–5458, March 2019. doi: 10.1093/mnras/sty3393.

[68] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

[69] S. Derriere. Building the ASTRODEEP Data Portal. In N. P. F. Lorente, K. Shortridge, and R. Wayth, editors, *Astronomical Data Analysis Software and Systems XXV*, volume 512 of *Astronomical Society of the Pacific Conference Series*, page 117, December 2017.

[70] Sander Dieleman, Kyle W. Willett, and Joni Dambre. Rotation-invariant convolutional neural networks for galaxy morphology prediction. *MNRAS*, 450:1441–1459, June 2015. doi: 10.1093/mnras/stv632.

[71] P. Dimauro, M. Huertas-Company, E. Daddi, P. G. Pérez-González, M. Bernardi, G. Barro, F. Buitrago, F. Caro, A. Cattaneo, H. Dominguez-Sánchez, S. M. Faber, B. Häußler, D. D. Kocevski, A. M. Koekemoer, D. C. Koo, C. T. Lee, S. Mei, B. Margalef-Bentabol, J. Primack, A. Rodriguez-Puebla, M. Salvato, F. Shankar, and D. Tuccillo. A catalog of polychromatic bulge-disc decompositions of ˜ 17.600 galaxies in CANDELS. *MNRAS*, 478: 5410–5426, August 2018. doi: 10.1093/mnras/sty1379.

[72] S. Djorgovski and M. Davis. Fundamental properties of elliptical galaxies. *ApJ*, 313:59–68, February 1987. doi: 10.1086/164948.

[73] H. Domínguez Sánchez, M. Huertas-Company, M. Bernardi, D. Tuccillo, and J. L. Fischer. Improving galaxy morphologies for SDSS with Deep Learning. *MNRAS*, 476(3):3661–3676, February 2018. doi: 10.1093/mnras/sty338.

[74] H. Domínguez Sánchez, M. Huertas-Company, M. Bernardi, S. Kaviraj, J. L. Fischer, T. M. C. Abbott, F. B. Abdalla, J. Annis, S. Avila, D. Brooks, E. Buckley-Geer, A. Carnero Rosell, M. Carrasco Kind, J. Carretero, C. E. Cunha, C. B. D'Andrea, L. N. da Costa, C. Davis, J. De Vicente, P. Doel, A. E. Evrard, P. Fosalba, J. Frieman, J. García-Bellido, E. Gaztanaga, D. W. Gerdes, D. Gruen, R. A. Gruendl, J. Gschwend, G. Gutierrez, W. G. Hartley, D. L. Hollowood, K. Honscheid, B. Hoyle, D. J. James, K. Kuehn, N. Kuropatkin, O. Lahav, M. A. G. Maia, M. March, P. Melchior, F. Menanteau, R. Miquel, B. Nord, A. A. Plazas, E. Sanchez, V. Scarpine, R. Schindler, M. Schubnell, M. Smith, R. C. Smith, M. Soares-Santos, F. Sobreira, E. Suchyta, M. E. C. Swanson, G. Tarle, D. Thomas, A. R. Walker, and J. Zuntz. Transfer learning for galaxy morphology from one survey to another. *MNRAS*, 484:93–100, March 2019. doi: 10.1093/mnras/sty3497.

[75] Nicole E. Drakos, Bruno Villasenor, Brant E. Robertson, Ryan Hausen, Mark E. Dickinson, Henry C. Ferguson, Steven R. Furlanetto, Jenny E. Greene, Piero Madau, Alice E. Shapley, Daniel P. Stark, and Risa H. Wechsler. Deep Realistic Extragalactic Model (DREaM) Galaxy Catalogs: Predictions for a Roman Ultra-Deep Field. *arXiv e-prints*, art. arXiv:2110.10703, October 2021.

[76] A. Dressler. Galaxy morphology in rich clusters - Implications for the formation and evolution of galaxies. *ApJ*, 236:351–365, March 1980. doi: 10.1086/157753.

[77] A. Dressler, D. Lynden-Bell, D. Burstein, R. L. Davies, S. M. Faber, R. Terlevich, and G. Wegner. Spectroscopy and photometry of elliptical galaxies. I - A new distance estimator. *ApJ*, 313:42–58, February 1987. doi: 10.1086/164947.

[78] A. Dressler, A. Oemler, Jr., W. J. Couch, I. Smail, R. S. Ellis, A. Barger, H. Butcher, B. M. Poggianti, and R. M. Sharples. Evolution since z = 0.5 of the Morphology-Density Relation for Clusters of Galaxies. *ApJ*, 490: 577–591, December 1997. doi: 10.1086/304890.

[79] Dmitry A. Duev, Ashish Mahabal, Quanzhi Ye, Kushal Tirumala, Justin Belicki, Richard Dekany, Sara Frederick, Matthew J. Graham, Russ R. Laher, Frank J. Masci, Thomas A. Prince, Reed Riddle, Philippe Rosnet, and Maayane T. Soumagnac. DeepStreaks: identifying fast-moving objects in the Zwicky Transient Facility data with deep learning. *MNRAS*, 486(3): 4158–4165, July 2019. doi: 10.1093/mnras/stz1096.

[80] D. M. Elmegreen, B. G. Elmegreen, D. S. Rubin, and M. A. Schaffer. Galaxy Morphologies in the Hubble Ultra Deep Field: Dominance of Linear Structures at the Detection Limit. *ApJ*, 631:85–100, September 2005. doi: 10.1086/432502.

[81] S. M. Faber, C. N. A. Willmer, C. Wolf, D. C. Koo, B. J. Weiner, J. A. Newman, M. Im, A. L. Coil, C. Conroy, M. C. Cooper, M. Davis, D. P. Finkbeiner, B. F. Gerke, K. Gebhardt, E. J. Groth, P. Guhathakurta, J. Harker, N. Kaiser, S. Kassin, M. Kleinheinrich, N. P. Konidaris, R. G. Kron, L. Lin, G. Luppino, D. S. Madgwick, K. Meisenheimer, K. G. Noeske, A. C. Phillips, V. L. Sarajedini, R. P. Schiavon, L. Simard, A. S. Szalay, N. P. Vogt, and R. Yan. Galaxy Luminosity Functions to z$\sim$1 from DEEP2 and COMBO-17: Implications for Red Galaxy Formation. *ApJ*, 665(1): 265–294, August 2007. doi: 10.1086/519294.

[82] M. Franx, P. G. van Dokkum, N. M. Förster Schreiber, S. Wuyts, I. Labbé, and S. Toft. Structure and Star Formation in Galaxies out to z = 3: Evidence for Surface Density Dependent Evolution and Upsizing. *ApJ*, 688:770–788, December 2008. doi: 10.1086/592431.

[83] Jerome H. Friedman. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29(5):1189 – 1232, 2001. doi: 10.1214/aos/1013203451. URL `https://doi.org/10.1214/aos/1013203451`.

[84] T. J. Galvin, M. T. Huynh, R. P. Norris, X. R. Wang, E. Hopkins, K. Polsterer, N. O. Ralph, A. N. O'Brien, and G. H. Heald. Cataloguing the radio-sky with unsupervised machine learning: a new approach for the SKA era. *MNRAS*, 497(3):2730–2758, September 2020. doi: 10.1093/mnras/staa1890.

[85] J. P. Gardner, J. C. Mather, M. Clampin, R. Doyon, M. A. Greenhouse, H. B. Hammel, J. B. Hutchings, P. Jakobsen, S. J. Lilly, K. S. Long, J. I. Lunine, M. J. McCaughrean, M. Mountain, J. Nella, G. H. Rieke, M. J. Rieke, H.-W. Rix, E. P. Smith, G. Sonneborn, M. Stiavelli, H. S. Stockman, R. A. Windhorst, and G. S. Wright. The James Webb Space Telescope. *SSRv*, 123:485–606, April 2006. doi: 10.1007/s11214-006-8315-7.

[86] Sarvesh Gharat and Yogesh Dandawate. Galaxy classification: a deep learning approach for classifying Sloan Digital Sky Survey images. *MNRAS*, 511 (4):5120–5124, April 2022. doi: 10.1093/mnras/stac457.

[87] C. Gheller, F. Vazza, and A. Bonafede. Deep learning based detection of cosmological diffuse radio sources. *MNRAS*, 480(3):3749–3761, November 2018. doi: 10.1093/mnras/sty2102.

[88] Aritra Ghosh, C. Megan Urry, Zhengdong Wang, Kevin Schawinski, Dennis Turp, and Meredith C. Powell. Galaxy Morphology Network: A Convolutional Neural Network Used to Study Morphology and Quenching in ∼100,000 SDSS and ∼20,000 CANDELS Galaxies. *ApJ*, 895(2):112, June 2020. doi: 10.3847/1538-4357/ab8a47.

[89] Fabian Gieseke, Steven Bloemen, Cas van den Bogaard, Tom Heskes, Jonas Kindler, Richard A. Scalzo, Valério A. R. M. Ribeiro, Jan van Roestel, Paul J. Groot, Fang Yuan, Anais Möller, and Brad E. Tucker. Convolutional neural networks for transient candidate vetting in large-scale surveys. *MNRAS*, 472(3):3101–3114, December 2017. doi: 10.1093/mnras/stx2161.

[90] F. Giordano, E. Racero, H. Norman, R. Vallés, B. Merín, D. Baines, M. López-Caniego, B. L. Martí, P. de Teodoro, J. Salgado, M. H. Sarmiento, R. Gutiérrez-Sánchez, R. Prieto, A. Lorca, S. Alberola, I. Valtchanov, G. de Marchi, R. Álvarez, and C. Arviset. ESASky: A science-driven discovery portal for space-based astronomy missions. *Astronomy and Computing*, 24: 97, July 2018. doi: 10.1016/j.ascom.2018.05.002.

[91] G. Girelli, L. Pozzetti, M. Bolzonella, C. Giocoli, F. Marulli, and M. Baldi. The stellar-to-halo mass relation over the past 12 Gyr. I. Standard ΛCDM model. *A&A*, 634:A135, February 2020. doi: 10.1051/0004-6361/201936329.

[92] Nickolay Y. Gnedin. Cosmic Reionization on Computers. I. Design and Calibration of Simulations. *ApJ*, 793(1):29, September 2014. doi: 10.1088/0004-637X/793/1/29.

[93] Hunter Goddard and Lior Shamir. A Catalog of Broad Morphology of Pan-STARRS Galaxies Based on Deep Learning. *ApJS*, 251(2):28, December 2020. doi: 10.3847/1538-4365/abc0ed.

[94] R. E. González, R. P. Muñoz, and C. A. Hernández. Galaxy detection and identification using deep learning and data augmentation. *ArXiv e-prints*, September 2018.

[95] N. A. Grogin, D. D. Kocevski, S. M. Faber, H. C. Ferguson, A. M. Koekemoer, A. G. Riess, V. Acquaviva, D. M. Alexander, O. Almaini, M. L. N. Ashby, M. Barden, E. F. Bell, F. Bournaud, T. M. Brown, K. I. Caputi, S. Casertano, P. Cassata, M. Castellano, P. Challis, R.-R. Chary, E. Cheung, M. Cirasuolo, C. J. Conselice, A. Roshan Cooray, D. J. Croton, E. Daddi, T. Dahlen, R. Davé, D. F. de Mello, A. Dekel, M. Dickinson, T. Dolch, J. L. Donley, J. S. Dunlop, A. A. Dutton, D. Elbaz, G. G. Fazio, A. V. Filippenko, S. L. Finkelstein, A. Fontana, J. P. Gardner, P. M. Garnavich, E. Gawiser, M. Giavalisco, A. Grazian, Y. Guo, N. P. Hathi, B. Häussler, P. F. Hopkins, J.-S. Huang, K.-H. Huang, S. W. Jha, J. S. Kartaltepe,

R. P. Kirshner, D. C. Koo, K. Lai, K.-S. Lee, W. Li, J. M. Lotz, R. A. Lucas, P. Madau, P. J. McCarthy, E. J. McGrath, D. H. McIntosh, R. J. McLure, B. Mobasher, L. A. Moustakas, M. Mozena, K. Nandra, J. A. Newman, S.-M. Niemi, K. G. Noeske, C. J. Papovich, L. Pentericci, A. Pope, J. R. Primack, A. Rajan, S. Ravindranath, N. A. Reddy, A. Renzini, H.-W. Rix, A. R. Robaina, S. A. Rodney, D. J. Rosario, P. Rosati, S. Salimbeni, C. Scarlata, B. Siana, L. Simard, J. Smidt, R. S. Somerville, H. Spinrad, A. N. Straughn, L.-G. Strolger, O. Telford, H. I. Teplitz, J. R. Trump, A. van der Wel, C. Villforth, R. H. Wechsler, B. J. Weiner, T. Wiklind, V. Wild, G. Wilson, S. Wuyts, H.-J. Yan, and M. S. Yun. CANDELS: The Cosmic Assembly Near-infrared Deep Extragalactic Legacy Survey. *ApJS*, 197:35, December 2011. doi: 10.1088/0067-0049/197/2/35.

[96] Yicheng Guo, Henry C. Ferguson, Mauro Giavalisco, Guillermo Barro, S. P. Willner, Matthew L. N. Ashby, Tomas Dahlen, Jennifer L. Donley, Sandra M. Faber, Adriano Fontana, Audrey Galametz, Andrea Grazian, Kuang-Han Huang, Dale D. Kocevski, Anton M. Koekemoer, David C. Koo, Elizabeth J. McGrath, Michael Peth, Mara Salvato, Stijn Wuyts, Marco Castellano, Asantha R. Cooray, Mark E. Dickinson, James S. Dunlop, G. G. Fazio, Jonathan P. Gardner, Eric Gawiser, Norman A. Grogin, Nimish P. Hathi, Li-Ting Hsu, Kyoung-Soo Lee, Ray A. Lucas, Bahram Mobasher, Kirpal Nand ra, Jeffery A. Newman, and Arjen van der Wel. CANDELS Multi-wavelength Catalogs: Source Detection and Photometry in the GOODS-South Field. *ApJS*, 207(2):24, August 2013. doi: 10.1088/0067-0049/207/2/24.

[97] R. Gupta, P. K. Srijith, and S. Desai. Galaxy morphology classification using neural ordinary differential equations. *Astronomy and Computing*, 38: 100543, January 2022. doi: 10.1016/j.ascom.2021.100543.

[98] R. H. R. Hahnloser, R. Sarpeshkar, M. A. Mahowald, R. J. Douglas, and H. S. Seung. Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit. *Nature*, 405:947–951, June 2000. doi: 10.1038/35016072.

[99] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585 (7825):357–362, September 2020. doi: 10.1038/s41586-020-2649-2. URL `https://doi.org/10.1038/s41586-020-2649-2`.

[100] A. H. Hassan, C. J. Fluke, D. G. Barnes, and V. A. Kilborn. Tera-scale astronomical data analysis and visualization. *mnras*, 429(3):2442–2455, March 2013. doi: 10.1093/mnras/sts513.

[101] Trevor Hastie and Robert Tibshirani. Generalized additive models. *Statistical Science*, 1(3):297–310, 1986. ISSN 08834237. URL `http://www.jstor.org/stable/2245459`.

[102] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc., New York, NY, USA, 2001.

[103] Ryan Hausen and Brant Robertson. FitsMap: Interactive astronomical image and catalog data visualizer. Astrophysics Source Code Library, record ascl:2201.004, January 2022.

[104] Ryan Hausen and Brant E. Robertson. Morpheus: A Deep Learning Framework for the Pixel-level Analysis of Astronomical Image Data. *ApJS*, 248 (1):20, May 2020. doi: 10.3847/1538-4365/ab8868.

[105] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. *arXiv e-prints*, art. arXiv:1703.06870, March 2017.

[106] Andrew P. Hearin, Andrew R. Zentner, Andreas A. Berlind, and Jeffrey A. Newman. SHAM beyond clustering: new tests of galaxy-halo abundance matching with galaxy groups. *MNRAS*, 433(1):659–680, July 2013. doi: 10.1093/mnras/stt755.

[107] Shoubaneh Hemmati, Peter Capak, Daniel Masters, Iary Davidzon, Olivier Dorè, Jeffrey Kruk, Bahram Mobasher, Jason Rhodes, Daniel Scolnic, and Daniel Stern. Photometric Redshift Calibration Requirements for WFIRST Weak-lensing Cosmology: Predictions from CANDELS. *ApJ*, 877(2):117, Jun 2019. doi: 10.3847/1538-4357/ab1be5.

[108] Yashar D. Hezaveh, Laurence Perreault Levasseur, and Philip J. Marshall. Fast automated analysis of strong gravitational lenses with convolutional neural networks. *Nature*, 548(7669):555–557, Aug 2017. doi: 10.1038/nature23463.

[109] Alex Hocking, James E. Geach, Yi Sun, and Neil Davey. An automatic taxonomy of galaxy morphology using unsupervised machine learning. *MNRAS*, 473(1):1108–1129, Jan 2018. doi: 10.1093/mnras/stx2351.

[110] Zafiirah Hosenie, Steven Bloemen, Paul Groot, Robert Lyon, Bart Scheers, Benjamin Stappers, Fiorenzo Stoppa, Paul Vreeswijk, Simon De Wet,

Marc Klein Wolt, Elmar Körding, Vanessa McBride, Rudolf Le Poole, Kerry Paterson, Daniëlle L. A. Pieterse, and Patrick Woudt. MeerCRAB: MeerLICHT classification of real and bogus transients using deep learning. *Experimental Astronomy*, June 2021. doi: 10.1007/s10686-021-09757-1.

[111] Ping Hu, Federico Perazzi, Fabian Caba Heilbron, Oliver Wang, Zhe Lin, Kate Saenko, and Stan Sclaroff. Real-time Semantic Segmentation with Fast Attention. *arXiv e-prints*, art. arXiv:2007.03815, July 2020.

[112] E. P. Hubble. Extragalactic nebulae. *ApJ*, 64, December 1926. doi: 10.1086/143018.

[113] M. Huertas-Company, R. Gravet, G. Cabrera-Vives, P. G. Pérez-González, J. S. Kartaltepe, G. Barro, M. Bernardi, S. Mei, F. Shankar, P. Dimauro, E. F. Bell, D. Kocevski, D. C. Koo, S. M. Faber, and D. H. Mcintosh. A Catalog of Visual-like Morphologies in the 5 CANDELS Fields Using Deep Learning. *The Astrophysical Journal Supplement Series*, 221:8, November 2015. doi: 10.1088/0067-0049/221/1/8.

[114] M. Huertas-Company, M. Bernardi, P. G. Pérez-González, M. L. N. Ashby, G. Barro, C. Conselice, E. Daddi, A. Dekel, P. Dimauro, S. M. Faber, N. A. Grogin, J. S. Kartaltepe, D. D. Kocevski, A. M. Koekemoer, D. C. Koo, S. Mei, and F. Shankar. Mass assembly and morphological transformations since z ˜ 3 from CANDELS. *MNRAS*, 462:4495–4516, November 2016. doi: 10.1093/mnras/stw1866.

[115] M. Huertas-Company, J. R. Primack, A. Dekel, D. C. Koo, S. Lapiner, D. Ceverino, R. C. Simons, G. F. Snyder, M. Bernardi, and Z. Chen. Deep Learning Identifies High-z Galaxies in a Central Blue Nugget Phase in a Characteristic Mass Range. *ApJ*, 858(2):114, May 2018. doi: 10.3847/1538-4357/aabfed.

[116] Marc Huertas-Company, Vicente Rodriguez-Gomez, Dylan Nelson, Annalisa Pillepich, Connor Bottrell, Mariangela Bernardi, Helena Domínguez-Sánchez, Shy Genel, Ruediger Pakmor, Gregory F. Snyder, and Mark Vogelsberger. The Hubble Sequence at z ˜0 in the IllustrisTNG simulation with deep learning. *MNRAS*, 489(2):1859–1879, Oct 2019. doi: 10.1093/mnras/stz2191.

[117] Ilian T. Iliev, Garrelt Mellema, Kyungjin Ahn, Paul R. Shapiro, Yi Mao, and Ue-Li Pen. Simulating cosmic reionization: how large a volume is large enough? *MNRAS*, 439(1):725–743, March 2014. doi: 10.1093/mnras/stt2497.

[118] G. Illingworth, D. Magee, R. Bouwens, P. Oesch, I. Labbe, P. van Dokkum, K. Whitaker, B. Holden, M. Franx, and V. Gonzalez. The Hubble Legacy Fields (HLF-GOODS-S) v1.5 Data Products: Combining 2442 Orbits of GOODS-S/CDF-S Region ACS and WFC3/IR Images. *arXiv e-prints*, June 2016.

[119] International Organization for Standardization. Iso/iec 15444-1:2019 information technology — jpeg 2000 image coding system — part 1: Core coding system. Technical report, International Organization for Standardization, 2019.

[120] International Organization for Standardization. Iso/iec 15948:2004 information technology — computer graphics and image processing — portable network graphics (png): Functional specification. Technical report, International Organization for Standardization, 2021.

[121] Sergey Ioffe and Christian Szegedy. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *ArXiv e-prints*, art. arXiv:1502.03167, February 2015.

[122] Ž. Ivezić, S. M. Kahn, J. A. Tyson, B. Abel, E. Acosta, R. Allsman, D. Alonso, Y. AlSayyad, S. F. Anderson, J. Andrew, and et al. LSST: from Science Drivers to Reference Design and Anticipated Data Products. *arXiv e-prints*, May 2008.

[123] Željko Ivezić, Steven M. Kahn, J. Anthony Tyson, Bob Abel, Emily Acosta, Robyn Allsman, David Alonso, Yusra AlSayyad, Scott F. Anderson, and John Andrew. LSST: From Science Drivers to Reference Design and Anticipated Data Products. *ApJ*, 873(2):111, Mar 2019. doi: 10.3847/1538-4357/ab042c.

[124] Peng Jia, Yifei Zhao, Gang Xue, and Dongmei Cai. Optical Transient Object Classification in Wide-field Small Aperture Telescopes with a Neural Network. *AJ*, 157(6):250, June 2019. doi: 10.3847/1538-3881/ab1e52.

[125] D. Jiang, F. S. Liu, X. Zheng, H. M. Yesuf, D. C. Koo, S. M. Faber, Y. Guo, A. M. Koekemoer, W. Wang, J. J. Fang, G. Barro, M. Jia, W. Tong, L. Liu, X. Meng, D. Kocevski, E. J. McGrath, and N. P. Hathi. The Isophotal Structure of Star-forming Galaxies at $0.5 < z < 1.8$ in CANDELS: Implications for the Evolution of Galaxy Structure. *ApJ*, 854:70, February 2018. doi: 10.3847/1538-4357/aaa5ad.

[126] Y. P. Jing, H. J. Mo, and G. Börner. Spatial Correlation Function and Pairwise Velocity Dispersion of Galaxies: Cold Dark Matter Models versus

the Las Campanas Survey. *ApJ*, 494(1):1–12, February 1998. doi: 10.1086/305209.

[127] William Joye. SAOImageDS9/SAOImageDS9 v8.0.1, January 2019.

[128] Stephanie Juneau, Knut Olsen, Robert Nikutta, Alice Jacques, and Stephen Bailey. Jupyter-Enabled Astrophysical Analysis Using Data-Proximate Computing Platforms. *Computing in Science and Engineering*, 23(2):15–25, March 2021. doi: 10.1109/MCSE.2021.3057097.

[129] Boris S. Kalita, Emanuele Daddi, Chiara D'Eugenio, Francesco Valentino, R. Michael Rich, Carlos Gómez-Guijarro, Rosemary T. Coogan, Ivan Delvecchio, David Elbaz, James D. Neill, Annagrazia Puglisi, and Veronica Strazzullo. An Ancient Massive Quiescent Galaxy Found in a Gas-rich z 3 Group. *ApJL*, 917(2):L17, August 2021. doi: 10.3847/2041-8213/ac16dc.

[130] Shreyas Kalvankar, Hrushikesh Pandit, and Pranav Parwate. Galaxy Morphology Classification using EfficientNet Architectures. *arXiv e-prints*, art. arXiv:2008.13611, August 2020.

[131] R. Kannan, E. Garaldi, A. Smith, R. Pakmor, V. Springel, M. Vogelsberger, and L. Hernquist. Introducing the THESAN project: radiation-magneto-hydrodynamic simulations of the Epoch of Reionization. *arXiv e-prints*, art. arXiv:2110.00584, October 2021.

[132] J. S. Kartaltepe, M. Mozena, D. Kocevski, D. H. McIntosh, J. Lotz, E. F. Bell, S. Faber, H. Ferguson, D. Koo, R. Bassett, M. Bernyk, K. Blancato, F. Bournaud, P. Cassata, M. Castellano, E. Cheung, C. J. Conselice, D. Croton, T. Dahlen, D. F. de Mello, L. DeGroot, J. Donley, J. Guedes, N. Grogin, N. Hathi, M. Hilton, B. Hollon, A. Koekemoer, N. Liu, R. A. Lucas, M. Martig, E. McGrath, C. McPartland, B. Mobasher, A. Morlock, E. O'Leary, M. Peth, J. Pforr, A. Pillepich, D. Rosario, E. Soto, A. Straughn, O. Telford, B. Sunnquist, J. Trump, B. Weiner, S. Wuyts, H. Inami, S. Kassin, C. Lani, G. B. Poole, and Z. Rizer. CANDELS Visual Classifications: Scheme, Data Release, and First Results. *ApJS*, 221:11, November 2015. doi: 10.1088/0067-0049/221/1/11.

[133] Reza Katebi, Yadi Zhou, Ryan Chornock, and Razvan Bunescu. Galaxy morphology prediction using Capsule Networks. *MNRAS*, 486(2):1539–1547, June 2019. doi: 10.1093/mnras/stz915.

[134] Guinevere Kauffmann, Simon D. M. White, Timothy M. Heckman, Brice Ménard, Jarle Brinchmann, Stéphane Charlot, Christy Tremonti, and Jon Brinkmann. The environmental dependence of the relations between stellar

mass, structure, star formation and nuclear activity in galaxies. *MNRAS*, 353(3):713–731, September 2004. doi: 10.1111/j.1365-2966.2004.08117.x.

[135] L. Kawinwanichakij, C. Papovich, R. F. Quadri, K. Glazebrook, G. G. Kacprzak, R. J. Allen, E. F. Bell, D. J. Croton, A. Dekel, H. C. Ferguson, B. Forrest, N. A. Grogin, Y. Guo, D. D. Kocevski, A. M. Koekemoer, I. Labbé, R. A. Lucas, T. Nanayakkara, L. R. Spitler, C. M. S. Straatman, K.-V. H. Tran, A. Tomczak, and P. van Dokkum. Effect of Local Environment and Stellar Mass on Galaxy Quenching and Morphology at $0.5 < z < 2.0$. *ApJ*, 847:134, October 2017. doi: 10.3847/1538-4357/aa8b75.

[136] B. C. Kelly and T. A. McKay. Morphological Classification of Galaxies by Shapelet Decomposition in the Sloan Digital Sky Survey. *AJ*, 127:625–645, February 2004. doi: 10.1086/380934.

[137] Brandon C. Kelly and Timothy A. McKay. Morphological classification of galaxies by shapelet decomposition in the sloan digital sky survey. ii. multiwavelength classification. *The Astronomical Journal*, 129(3):1287, 2005. URL `http://stacks.iop.org/1538-3881/129/i=3/a=1287`.

[138] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-Task Learning Using Uncertainty to Weigh Losses for Scene Geometry and Semantics. *arXiv e-prints*, art. arXiv:1705.07115, May 2017.

[139] Edward J. Kim and Robert J. Brunner. Star-galaxy classification using deep convolutional neural networks. *MNRAS*, 464(4):4463–4475, February 2017. doi: 10.1093/mnras/stw2672.

[140] D. P. Kingma and J. Ba. Adam: A Method for Stochastic Optimization. *ArXiv e-prints*, December 2014.

[141] Vladislav Klochkov, Ivan Katkov, Igor Chilingarian, Kirill Grishin, Anastasia Kasparova, Vladimir Goradzhanov, Victoria Toptun, Evgenii Rubtsov, and Sviatoslav Borisov. RCSEDv2: Open-source web tools for visualization of imaging and spectral data. *arXiv e-prints*, art. arXiv:2112.04867, December 2021.

[142] Thomas Kluyver, Benjamin Ragan-Kelley, Fernando Pérez, Brian Granger, Matthias Bussonnier, Jonathan Frederic, Kyle Kelley, Jessica Hamrick, Jason Grout, Sylvain Corlay, Paul Ivanov, Damián Avila, Safia Abdalla, and Carol Willing. Jupyter notebooks – a publishing format for reproducible computational workflows. In F. Loizides and B. Schmidt, editors, *Positioning and Power in Academic Publishing: Players, Agents and Agendas*, pages 87 – 90. IOS Press, 2016.

[143] D. D. Kocevski, S. M. Faber, M. Mozena, A. M. Koekemoer, K. Nandra, C. Rangel, E. S. Laird, M. Brusa, S. Wuyts, J. R. Trump, D. C. Koo, R. S. Somerville, E. F. Bell, J. M. Lotz, D. M. Alexander, F. Bournaud, C. J. Conselice, T. Dahlen, A. Dekel, J. L. Donley, J. S. Dunlop, A. Finoguenov, A. Georgakakis, M. Giavalisco, Y. Guo, N. A. Grogin, N. P. Hathi, S. Juneau, J. S. Kartaltepe, R. A. Lucas, E. J. McGrath, D. H. McIntosh, B. Mobasher, A. R. Robaina, D. Rosario, A. N. Straughn, A. van der Wel, and C. Villforth. CANDELS: Constraining the AGN-Merger Connection with Host Morphologies at z ˜ 2. *ApJ*, 744:148, January 2012. doi: 10.1088/0004-637X/744/2/148.

[144] A. M. Koekemoer, S. M. Faber, H. C. Ferguson, N. A. Grogin, D. D. Kocevski, D. C. Koo, K. Lai, J. M. Lotz, R. A. Lucas, E. J. McGrath, S. Ogaz, A. Rajan, A. G. Riess, S. A. Rodney, L. Strolger, S. Casertano, M. Castellano, T. Dahlen, M. Dickinson, T. Dolch, A. Fontana, M. Giavalisco, A. Grazian, Y. Guo, N. P. Hathi, K.-H. Huang, A. van der Wel, H.-J. Yan, V. Acquaviva, D. M. Alexander, O. Almaini, M. L. N. Ashby, M. Barden, E. F. Bell, F. Bournaud, T. M. Brown, K. I. Caputi, P. Cassata, P. J. Challis, R.-R. Chary, E. Cheung, M. Cirasuolo, C. J. Conselice, A. Roshan Cooray, D. J. Croton, E. Daddi, R. Davé, D. F. de Mello, L. de Ravel, A. Dekel, J. L. Donley, J. S. Dunlop, A. A. Dutton, D. Elbaz, G. G. Fazio, A. V. Filippenko, S. L. Finkelstein, C. Frazer, J. P. Gardner, P. M. Garnavich, E. Gawiser, R. Gruetzbauch, W. G. Hartley, B. Häussler, J. Herrington, P. F. Hopkins, J.-S. Huang, S. W. Jha, A. Johnson, J. S. Kartaltepe, A. A. Khostovan, R. P. Kirshner, C. Lani, K.-S. Lee, W. Li, P. Madau, P. J. McCarthy, D. H. McIntosh, R. J. McLure, C. McPartland, B. Mobasher, H. Moreira, A. Mortlock, L. A. Moustakas, M. Mozena, K. Nandra, J. A. Newman, J. L. Nielsen, S. Niemi, K. G. Noeske, C. J. Papovich, L. Pentericci, A. Pope, J. R. Primack, S. Ravindranath, N. A. Reddy, A. Renzini, H.-W. Rix, A. R. Robaina, D. J. Rosario, P. Rosati, S. Salimbeni, C. Scarlata, B. Siana, L. Simard, J. Smidt, D. Snyder, R. S. Somerville, H. Spinrad, A. N. Straughn, O. Telford, H. I. Teplitz, J. R. Trump, C. Vargas, C. Villforth, C. R. Wagner, P. Wandro, R. H. Wechsler, B. J. Weiner, T. Wiklind, V. Wild, G. Wilson, S. Wuyts, and M. S. Yun. CANDELS: The Cosmic Assembly Near-infrared Deep Extragalactic Legacy Survey—The Hubble Space Telescope Observations, Imaging Data Products, and Mosaics. *ApJS*, 197:36, December 2011. doi: 10.1088/0067-0049/197/2/36.

[145] J. Kormendy. Brightness distributions in compact and normal galaxies. II - Structure parameters of the spheroidal component. *ApJ*, 218:333–346, December 1977. doi: 10.1086/155687.

[146] M. Kornilov and K. Malanchev. Fips: An OpenGL based FITS viewer.

Astronomy and Computing, 26:61, January 2019. doi: 10.1016/j.ascom.2019.01.001.

[147] A. V. Kravtsov, A. A. Vikhlinin, and A. V. Meshcheryakov. Stellar Mass—Halo Mass Relation and Star Formation Efficiency in High-Mass Halos. *Astronomy Letters*, 44(1):8–34, January 2018. doi: 10.1134/S1063773717120015.

[148] John E. Krist, Richard N. Hook, and Felix Stoehr. 20 years of Hubble Space Telescope optical modeling using Tiny Tim. In Mark A. Kahan, editor, *Optical Modeling and Performance Predictions V*, volume 8127 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, page 81270J, October 2011. doi: 10.1117/12.892762.

[149] John E. Krist, Richard N. Hook, and Felix Stoehr. 20 years of Hubble Space Telescope optical modeling using Tiny Tim. In *Optical Modeling and Performance Predictions V*, volume 8127 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, page 81270J, October 2011. doi: 10.1117/12.892762.

[150] Evan Kuminski and Lior Shamir. A Computer-generated Visual Morphology Catalog of ∼3,000,000 SDSS Galaxies. *ApJS*, 223(2):20, April 2016. doi: 10.3847/0067-0049/223/2/20.

[151] R. Laureijs, J. Amiaux, S. Arduini, J. L. Augueres, J. Brinchmann, R. Cole, M. Cropper, C. Dabin, L. Duvet, and A. Ealet. Euclid Definition Study Report. *arXiv e-prints*, art. arXiv:1110.3193, Oct 2011.

[152] Y. Lecun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521:436–444, May 2015. doi: 10.1038/nature14539.

[153] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, and Wenzhe Shi. Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network. *arXiv e-prints*, art. arXiv:1609.04802, September 2016.

[154] B. Lee, M. Giavalisco, C. C. Williams, Y. Guo, J. Lotz, A. Van der Wel, H. C. Ferguson, S. M. Faber, A. Koekemoer, N. Grogin, D. Kocevski, C. J. Conselice, S. Wuyts, A. Dekel, J. Kartaltepe, and E. F. Bell. CANDELS: The Correlation between Galaxy Morphology and Star Formation Activity at z ˜ 2. *ApJ*, 774:47, September 2013. doi: 10.1088/0004-637X/774/1/47.

[155] Benjamin V. Lehmann, Yao-Yuan Mao, Matthew R. Becker, Samuel W. Skillman, and Risa H. Wechsler. The Concentration Dependence of the

Galaxy-Halo Connection: Modeling Assembly Bias with Abundance Matching. *ApJ*, 834(1):37, January 2017. doi: 10.3847/1538-4357/834/1/37.

[156] V. A. Lepingwell, A. J. Bird, and S. R. Gunn. New approaches for faint source detection in hard X-ray surveys. *MNRAS*, 510(3):4031–4039, March 2022. doi: 10.1093/mnras/stab3770.

[157] Eleanor Leung, Kenji Bekki, and Lyndon While. Automated Simulations of Galaxy Morphology Evolution using Deep Learning and Particle Swarm Optimisation. *arXiv e-prints*, art. arXiv:1904.02906, April 2019.

[158] Cheng Li, Y. P. Jing, Shude Mao, Jiaxin Han, Qiuying Peng, Xiaohu Yang, H. J. Mo, and Frank van den Bosch. Internal Kinematics of Groups of Galaxies in the Sloan Digital Sky Survey Data Release 7. *ApJ*, 758(1):50, October 2012. doi: 10.1088/0004-637X/758/1/50.

[159] Ke Li and Jitendra Malik. Amodal Instance Segmentation. *arXiv e-prints*, art. arXiv:1604.08202, April 2016.

[160] Yen-Ting Lin, Joseph J. Mohr, and S. Adam Stanford. K-Band Properties of Galaxy Clusters and Groups: Luminosity Function, Radial Distribution, and Halo Occupation Number. *ApJ*, 610(2):745–761, August 2004. doi: 10.1086/421714.

[161] C. J. Lintott, K. Schawinski, A. Slosar, K. Land, S. Bamford, D. Thomas, M. J. Raddick, R. C. Nichol, A. Szalay, D. Andreescu, P. Murray, and J. Vandenberg. Galaxy Zoo: morphologies derived from visual inspection of galaxies from the Sloan Digital Sky Survey. *MNRAS*, 389:1179–1189, September 2008. doi: 10.1111/j.1365-2966.2008.13689.x.

[162] E. K. Lofthouse, S. Kaviraj, C. J. Conselice, A. Mortlock, and W. Hartley. Major mergers are not significant drivers of star formation or morphological transformation around the epoch of peak cosmic star formation. *MNRAS*, 465:2895–2900, March 2017. doi: 10.1093/mnras/stw2895.

[163] J. M. Lotz, J. Primack, and P. Madau. A New Nonparametric Approach to Galaxy Morphological Classification. *AJ*, 128:163–182, July 2004. doi: 10.1086/421849.

[164] J. M. Lotz, M. Davis, S. M. Faber, P. Guhathakurta, S. Gwyn, J. Huang, D. C. Koo, E. Le Floc'h, L. Lin, J. Newman, K. Noeske, C. Papovich, C. N. A. Willmer, A. Coil, C. J. Conselice, M. Cooper, A. M. Hopkins, A. Metevier, J. Primack, G. Rieke, and B. J. Weiner. The Evolution of Galaxy Mergers and Morphology at $z < 1.2$ in the Extended Groth Strip. *ApJ*, 672:177–197, January 2008. doi: 10.1086/523659.

[165] Yin Lou, Rich Caruana, and Johannes Gehrke. Intelligible models for classification and regression. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '12, page 150–158, New York, NY, USA, 2012. Association for Computing Machinery. ISBN 9781450314626. doi: 10.1145/2339530.2339556. URL https://doi.org/10.1145/2339530.2339556.

[166] Yin Lou, Rich Caruana, Johannes Gehrke, and Giles Hooker. Accurate intelligible models with pairwise interactions. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '13, page 623–631, New York, NY, USA, 2013. Association for Computing Machinery. ISBN 9781450321747. doi: 10.1145/2487575.2487579. URL https://doi.org/10.1145/2487575.2487579.

[167] Christopher C. Lovell, Stephen M. Wilkins, Peter A. Thomas, Matthieu Schaller, Carlton M. Baugh, Giulio Fabbian, and Yannick Bahé. A machine learning approach to mapping baryons onto dark matter halos using the EAGLE and C-EAGLE simulations. *arXiv e-prints*, art. arXiv:2106.04980, June 2021.

[168] LSST Science Collaboration, P. A. Abell, J. Allison, S. F. Anderson, J. R. Andrew, J. R. P. Angel, L. Armus, D. Arnett, S. J. Asztalos, T. S. Axelrod, and et al. LSST Science Book, Version 2.0. *arXiv e-prints*, December 2009.

[169] Vesna Lukic, Francesco de Gasperin, and Marcus Brüggen. ConvoSource: Radio-Astronomical Source-Finding with Convolutional Neural Networks. *Galaxies*, 8(1):3, December 2019. doi: 10.3390/galaxies8010003.

[170] Luis Fernando Machado Poletti Valle, Camille Avestruz, David J. Barnes, Arya Farahi, Erwin T. Lau, and Daisuke Nagai. SHAPing the gas: understanding gas shapes in dark matter haloes with interpretable machine learning. *MNRAS*, 507(1):1468–1484, October 2021. doi: 10.1093/mnras/stab2252.

[171] K. Makhlouf, D. Turpin, D. Corre, S. Karpov, D. A. Kann, and A. Klotz. O'TRAIN: a robust and flexible Real/Bogus classifier for the study of the optical transient sky. *arXiv e-prints*, art. arXiv:2112.10280, December 2021.

[172] Rachel Mandelbaum, Argyro Tasitsiomi, Uroš Seljak, Andrey V. Kravtsov, and Risa H. Wechsler. Galaxy-galaxy lensing: dissipationless simulations versus the halo model. *MNRAS*, 362(4):1451–1462, October 2005. doi: 10.1111/j.1365-2966.2005.09417.x.

[173] B. Margalef-Bentabol, C. J. Conselice, A. Mortlock, W. Hartley, K. Duncan, H. C. Ferguson, A. Dekel, and J. R. Primack. The formation of bulges, discs

and two-component galaxies in the CANDELS Survey at z < 3. *MNRAS*, 461:2728–2746, September 2016. doi: 10.1093/mnras/stw1451.

[174] M. Masias, J. Freixenet, X. Lladó, and M. Peracaula. A review of source detection approaches in astronomical images. *MNRAS*, 422(2):1674–1689, May 2012. doi: 10.1111/j.1365-2966.2012.20742.x.

[175] Viera Maslej-Krešňáková, Khadija El Bouchefry, and Peter Butka. Morphological classification of compact and extended radio galaxies using convolutional neural networks and data augmentation techniques. *MNRAS*, 505(1): 1464–1475, July 2021. doi: 10.1093/mnras/stab1400.

[176] Daniel Masters, Peter Capak, Daniel Stern, Olivier Ilbert, Mara Salvato, Samuel Schmidt, Giuseppe Longo, Jason Rhodes, Stephane Paltani, and Bahram Mobasher. Mapping the Galaxy Color-Redshift Relation: Optimal Photometric Redshift Calibration Strategies for Cosmology Surveys. *ApJ*, 813(1):53, Nov 2015. doi: 10.1088/0004-637X/813/1/53.

[177] John C. McCallum. Disk drive prices 1955+, 2022. URL `https://jcmit.net/diskprice.htm`.

[178] B. F. Meade and C. J. Fluke. Evaluating virtual hosted desktops for graphics-intensive astronomy. *Astronomy and Computing*, 23:124, April 2018. doi: 10.1016/j.ascom.2018.04.002.

[179] P. Melchior, F. Moolekamp, M. Jerdee, R. Armstrong, A. L. Sun, J. Bosch, and R. Lupton. SCARLET: Source separation in multi-band images by Constrained Matrix Factorization. *Astronomy and Computing*, 24:129, July 2018. doi: 10.1016/j.ascom.2018.07.001.

[180] T. B. Miller, P. van Dokkum, L. Mowla, and A. van der Wel. A New View of the Size—Mass Distribution of Galaxies: Using r_{20} and r_{80} Instead of r_{50}. *ApJL*, 872:L14, February 2019. doi: 10.3847/2041-8213/ab0380.

[181] F. Milletari, N. Navab, and S.-A. Ahmadi. GV-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation. *ArXiv e-prints*, June 2016.

[182] Devina Mohan, Anna M. M. Scaife, Fiona Porter, Mike Walmsley, and Micah Bowles. Quantifying uncertainty in deep learning approaches to radio galaxy classification. *MNRAS*, 511(3):3722–3740, April 2022. doi: 10.1093/mnras/stac223.

[183] P. Mohan, C. Hawkins, R. Klapaukh, and M. Johnston-Hollitt. Three Tools to Aid Visualisation of FITS Files for Astronomy. In N. P. F. Lorente,

K. Shortridge, and R. Wayth, editors, *Astronomical Data Analysis Software and Systems XXV*, volume 512 of *Astronomical Society of the Pacific Conference Series*, page 465, December 2017.

[184] A. Moitinho, A. Krone-Martins, H. Savietto, M. Barros, C. Barata, A. J. Falcão, T. Fernandes, J. Alves, A. F. Silva, M. Gomes, J. Bakker, A. G. A. Brown, J. González-Núñez, G. Gracia-Abril, R. Gutiérrez-Sánchez, J. Hernández, S. Jordan, X. Luri, B. Merin, F. Mignard, A. Mora, V. Navarro, W. O'Mullane, T. Sagristà Sellés, J. Salgado, J. C. Segovia, E. Utrilla, F. Arenou, J. H. J. de Bruijne, F. Jansen, M. McCaughrean, K. S. O'Flaherty, M. B. Taylor, and A. Vallenari. Gaia Data Release 1. The archive visualisation service. *A&A*, 605:A52, September 2017. doi: 10.1051/0004-6361/201731059.

[185] I. G. Momcheva, G. B. Brammer, P. G. van Dokkum, R. E. Skelton, K. E. Whitaker, E. J. Nelson, M. Fumagalli, M. V. Maseda, J. Leja, M. Franx, H.-W. Rix, R. Bezanson, E. Da Cunha, C. Dickey, N. M. Förster Schreiber, G. Illingworth, M. Kriek, I. Labbé, J. Ulf Lange, B. F. Lundgren, D. Magee, D. Marchesini, P. Oesch, C. Pacifici, S. G. Patel, S. Price, T. Tal, D. A. Wake, A. van der Wel, and S. Wuyts. The 3D-HST Survey: Hubble Space Telescope WFC3/G141 Grism Spectra, Redshifts, and Emission Line Measurements for ˜ 100,000 Galaxies. *ApJS*, 225:27, August 2016. doi: 10.3847/0067-0049/225/2/27.

[186] Y. L. Mong, K. Ackley, D. K. Galloway, T. Killestein, J. Lyman, D. Steeghs, V. Dhillon, P. T. O'Brien, G. Ramsay, S. Poshyachinda, R. Kotak, L. Nuttall, E. Pallé, D. Pollacco, E. Thrane, M. J. Dyer, K. Ulaczyk, R. Cutter, J. McCormac, P. Chote, A. J. Levan, T. Marsh, E. Stanway, B. Gompertz, K. Wiersema, A. Chrimes, A. Obradovic, J. Mullaney, E. Daw, S. Littlefair, J. Maund, L. Makrygianni, U. Burhanudin, R. L. C. Starling, R. A. J. Eyles-Ferris, S. Tooke, C. Duffy, S. Aukkaravittayapun, U. Sawangwit, S. Awiphan, D. Mkrtichian, P. Irawati, S. Mattila, T. Heikkilä, R. Breton, M. Kennedy, D. Mata Sánchez, and E. Rol. Machine learning for transient recognition in difference imaging with minimum sampling effort. *MNRAS*, 499(4):6009–6017, December 2020. doi: 10.1093/mnras/staa3096.

[187] F. Moolekamp and E. Mamajek. Toyz: A framework for scientific analysis of large datasets and astronomical images. *Astronomy and Computing*, 13: 50–57, November 2015. doi: 10.1016/j.ascom.2015.10.001.

[188] Surhud More, Frank C. van den Bosch, Marcello Cacciato, H. J. Mo, Xiaohu Yang, and Ran Li. Satellite kinematics - II. The halo mass-luminosity relation of central galaxies in SDSS. *MNRAS*, 392(2):801–816, January 2009. doi: 10.1111/j.1365-2966.2008.14095.x.

[189] T. Morishita, T. Ichikawa, and M. Kajisawa. The Evolution of Galaxy Size and Morphology at z ˜ 0.5-3.0 in the GOODS-N Region with Hubble Space Telescope/WFC3 Data. *ApJ*, 785:18, April 2014. doi: 10.1088/0004-637X/785/1/18.

[190] Warren R. Morningstar, Yashar D. Hezaveh, Laurence Perreault Levasseur, Roger D. Blandford, Philip J. Marshall, Patrick Putzky, and Risa H. Wechsler. Analyzing interferometric observations of strong gravitational lenses with recurrent and convolutional neural networks. *arXiv e-prints*, art. arXiv:1808.00011, Jul 2018.

[191] Warren R. Morningstar, Laurence Perreault Levasseur, Yashar D. Hezaveh, Roger Blandford, Phil Marshall, Patrick Putzky, Thomas D. Rueter, Risa Wechsler, and Max Welling. Data-Driven Reconstruction of Gravitationally Lensed Galaxies using Recurrent Inference Machines. *arXiv e-prints*, art. arXiv:1901.01359, Jan 2019.

[192] Benjamin P. Moster, Rachel S. Somerville, Christian Maulbetsch, Frank C. van den Bosch, Andrea V. Macciò, Thorsten Naab, and Ludwig Oser. Constraints on the Relationship between Stellar Mass and Halo Mass at Low and High Redshift. *ApJ*, 710(2):903–923, Feb 2010. doi: 10.1088/0004-637X/710/2/903.

[193] Rafaël I. J. Mostert, Kenneth J. Duncan, Huub J. A. Röttgering, Kai L. Polsterer, Philip N. Best, Marisa Brienza, Marcus Brüggen, Martin J. Hardcastle, Nika Jurlin, Beatriz Mingo, Raffaella Morganti, Tim Shimwell, Dan Smith, and Wendy L. Williams. Unveiling the rarest morphologies of the LOFAR Two-metre Sky Survey radio source population with self-organised maps. *A&A*, 645:A89, January 2021. doi: 10.1051/0004-6361/202038500.

[194] D. Nelson, A. Pillepich, S. Genel, M. Vogelsberger, V. Springel, P. Torrey, V. Rodriguez-Gomez, D. Sijacki, G. F. Snyder, B. Griffen, F. Marinacci, L. Blecha, L. Sales, D. Xu, and L. Hernquist. The illustris simulation: Public data release. *Astronomy and Computing*, 13:12–37, November 2015. doi: 10.1016/j.ascom.2015.09.003.

[195] K. G. Noeske, B. J. Weiner, S. M. Faber, C. Papovich, D. C. Koo, R. S. Somerville, K. Bundy, C. J. Conselice, J. A. Newman, D. Schiminovich, E. Le Floc'h, A. L. Coil, G. H. Rieke, J. M. Lotz, J. R. Primack, P. Barmby, M. C. Cooper, M. Davis, R. S. Ellis, G. G. Fazio, P. Guhathakurta, J. Huang, S. A. Kassin, D. C. Martin, A. C. Phillips, R. M. Rich, T. A. Small, C. N. A. Willmer, and G. Wilson. Star Formation in AEGIS Field Galaxies since z=1.1: The Dominance of Gradually Declining Star Formation, and the Main Sequence of Star-forming Galaxies. *ApJL*, 660(1):L43–L46, May 2007. doi: 10.1086/517926.

[196] Harsha Nori, Samuel Jenkins, Paul Koch, and Rich Caruana. Interpretml: A unified framework for machine learning interpretability. *arXiv preprint arXiv:1909.09223*, 2019.

[197] A. A. Novikov, D. Lenis, D. Major, J. Hladυka, M. Wimmer, and K. Bühler. Fully Convolutional Architectures for Multi-Class Segmentation in Chest Radiographs. *ArXiv e-prints*, January 2017.

[198] Pierre Ocvirk, Nicolas Gillet, Paul R. Shapiro, Dominique Aubert, Ilian T. Iliev, Romain Teyssier, Gustavo Yepes, Jun-Hwan Choi, David Sullivan, Alexander Knebe, Stefan Gottlöber, Anson D'Aloisio, Hyunbae Park, Yehuda Hoffman, and Timothy Stranex. Cosmic Dawn (CoDa): the First Radiation-Hydrodynamics Simulation of Reionization and Galaxy Formation in the Local Universe. *MNRAS*, 463(2):1462–1485, December 2016. doi: 10.1093/mnras/stw2036.

[199] Pierre Ocvirk, Dominique Aubert, Jenny G. Sorce, Paul R. Shapiro, Nicolas Deparis, Taha Dawoodbhoy, Joseph Lewis, Romain Teyssier, Gustavo Yepes, Stefan Gottlöber, Kyungjin Ahn, Ilian T. Iliev, and Yehuda Hoffman. Cosmic Dawn II (CoDa II): a new radiation-hydrodynamics simulation of the self-consistent coupling of galaxy formation and reionization. *MNRAS*, 496(4):4087–4107, August 2020. doi: 10.1093/mnras/staa1266.

[200] J. B. Oke and J. E. Gunn. Secondary standard stars for absolute spectrophotometry. *ApJ*, 266:713–717, March 1983. doi: 10.1086/160817.

[201] Ricky O'Steen, Nicholas Earl, P. L. Lim, Jesse Averbukh, Ivo Busko, Erik Tollerud, Kyle Conroy, and Thomas Robitaille. spacetelescope/jdaviz: c2.2.0, December 2021. URL `https://doi.org/10.5281/zenodo.5802276`.

[202] Boris Panes, Christopher Eckner, Luc Hendriks, Sacha Caron, Klaas Dijkstra, Guðlaugur Jóhannesson, Roberto Ruiz de Austri, and Gabrijela Zaharijas. Identification of point sources in gamma rays using U-shaped convolutional neural networks and a data challenge. *A&A*, 656:A62, November 2021. doi: 10.1051/0004-6361/202141193.

[203] J. Pasquet-Itam and J. Pasquet. Deep learning approach for classifying, detecting and predicting photometric redshifts of quasars in the Sloan Digital Sky Survey stripe 82. *A&A*, 611:A97, April 2018. doi: 10.1051/0004-6361/201731106.

[204] S. G. Patel, M. Fumagalli, M. Franx, P. G. van Dokkum, A. van der Wel, J. Leja, I. Labbé, G. Brammer, R. E. Skelton, I. Momcheva, K. E. Whitaker, B. Lundgren, A. Muzzin, R. F. Quadri, E. J. Nelson, D. A. Wake, and H.-W. Rix. The Structural Evolution of Milky-Way-like Star-forming Galaxies

since z ˜ 1.3. *ApJ*, 778:115, December 2013. doi: 10.1088/0004-637X/778/2/115.

[205] C. Y. Peng, L. C. Ho, C. D. Impey, and H.-W. Rix. Detailed Decomposition of Galaxy Images. II. Beyond Axisymmetric Models. *AJ*, 139:2097–2129, June 2010. doi: 10.1088/0004-6256/139/6/2097.

[206] M. Pérez-Carrasco, G. Cabrera-Vives, M. Martinez-Marin, P. Cerulo, R. Demarco, P. Protopapas, J. Godoy, and M. Huertas-Company. Multiband Galaxy Morphologies for CLASH: A Convolutional Neural Network Transferred from CANDELS. *PASP*, 131(1004):108002, October 2019. doi: 10.1088/1538-3873/aaeeb4.

[207] Simon Perkins, Jacques Questiaux, Stephen Finniss, Robin Tyler, Sarah Blyth, and Michelle M. Kuttel. Scalable desktop visualisation of very large radio astronomy data cubes. *New Astronomy*, 30:1–7, July 2014. doi: 10.1016/j.newast.2013.12.007.

[208] M. A. Peth, J. M. Lotz, P. E. Freeman, C. McPartland, S. A. Mortazavi, G. F. Snyder, G. Barro, N. A. Grogin, Y. Guo, S. Hemmati, J. S. Kartaltepe, D. D. Kocevski, A. M. Koekemoer, D. H. McIntosh, H. Nayyeri, C. Papovich, J. R. Primack, and R. C. Simons. Beyond spheroids and discs: classifications of CANDELS galaxy structure at $1.4 < z < 2$ via principal component analysis. *MNRAS*, 458:963–987, May 2016. doi: 10.1093/mnras/stw252.

[209] Annalisa Pillepich, Volker Springel, Dylan Nelson, Shy Genel, Jill Naiman, Rüdiger Pakmor, Lars Hernquist, Paul Torrey, Mark Vogelsberger, Rainer Weinberger, and Federico Marinacci. Simulating galaxy formation with the IllustrisTNG model. *MNRAS*, 473(3):4077–4106, January 2018. doi: 10.1093/mnras/stx2656.

[210] Joanna M. Piotrowska, Asa F. L. Bluck, Roberto Maiolino, and Yingjie Peng. On the quenching of star formation in observed and simulated central galaxies: evidence for the role of integrated AGN feedback. *MNRAS*, 512 (1):1052–1090, May 2022. doi: 10.1093/mnras/stab3673.

[211] M. C. Powell, C. M. Urry, C. N. Cardamone, B. D. Simmons, K. Schawinski, S. Young, and M. Kawakatsu. Morphology and the Color-mass Diagram as Clues to Galaxy Evolution at z& approx1. *ApJ*, 835:22, January 2017. doi: 10.3847/1538-4357/835/1/22.

[212] L. Y. Pratt. Discriminability-based transfer between neural networks. In S. J. Hanson, J. D. Cowan, and C. L. Giles, editors, *Advances in Neural Information Processing Systems 5*, pages 204–211. Morgan-Kaufmann, 1993. URL `http://papers.nips.cc/paper/`

641-discriminability-based-transfer-between-neural-networks.pdf.

[213] Rachel M. Reddick, Risa H. Wechsler, Jeremy L. Tinker, and Peter S. Behroozi. The Connection between Galaxies and Dark Matter Structures in the Local Universe. *ApJ*, 771(1):30, Jul 2013. doi: 10.1088/0004-637X/771/1/30.

[214] David M. Reiman and Brett E. Göhre. Deblending galaxy superpositions with branched generative adversarial networks. *MNRAS*, 485(2):2617–2627, May 2019. doi: 10.1093/mnras/stz575.

[215] Moonzarin Reza. Galaxy morphology classification using automated machine learning. *Astronomy and Computing*, 37:100492, October 2021. doi: 10.1016/j.ascom.2021.100492.

[216] S. Rezaei, J. P. McKean, M. Biehl, and A. Javadpour. DECORAS: detection and characterization of radio-astronomical sources using deep learning. *MNRAS*, 510(4):5891–5907, March 2022. doi: 10.1093/mnras/stab3519.

[217] Jason Rhodes, Robert C. Nichol, Éric Aubourg, Rachel Bean, Dominique Boutigny, Malcolm N. Bremer, Peter Capak, Vincenzo Cardone, Benoît Carry, and Christopher J. Conselice. Scientific Synergy between LSST and Euclid. *ApJS*, 233(2):21, Dec 2017. doi: 10.3847/1538-4365/aa96b0.

[218] Dezső Ribli, László Dobos, and István Csabai. Galaxy shape measurement with convolutional neural networks. *MNRAS*, 489(4):4847–4859, November 2019. doi: 10.1093/mnras/stz2374.

[219] M. S. Roberts and M. P. Haynes. Physical Parameters along the Hubble Sequence. *ARA&A*, 32:115–152, 1994. doi: 10.1146/annurev.aa.32.090194.000555.

[220] B. E. Robertson, M. Banerji, M. C. Cooper, R. Davies, S. P. Driver, A. M. N. Ferguson, H. C. Ferguson, E. Gawiser, S. Kaviraj, J. H. Knapen, C. Lintott, J. Lotz, J. A. Newman, D. J. Norman, N. Padilla, S. J. Schmidt, G. P. Smith, J. A. Tyson, A. Verma, I. Zehavi, L. Armus, C. Avestruz, L. F. Barrientos, R. A. A. Bowler, M. N. Bremer, C. J. Conselice, J. Davies, R. Demarco, M. E. Dickinson, G. Galaz, A. Grazian, B. W. Holwerda, M. J. Jarvis, V. Kasliwal, I. Lacerna, J. Loveday, P. Marshall, E. Merlin, N. R. Napolitano, T. H. Puzia, A. Robotham, S. Salim, M. Sereno, G. F. Snyder, J. P. Stott, P. B. Tissera, N. Werner, P. Yoachim, K. D. Borne, and Members of the LSST Galaxies Science Collaboration. Large Synoptic Survey Telescope Galaxies Science Roadmap. *arXiv e-prints*, August 2017.

[221] Brant E. Robertson. Galaxy Formation and Reionization: Key Unknowns and Expected Breakthroughs by the James Webb Space Telescope. *arXiv e-prints*, art. arXiv:2110.13160, October 2021.

[222] Brant E. Robertson, Manda Banerji, Sarah Brough, Roger L. Davies, Henry C. Ferguson, Ryan Hausen, Sugata Kaviraj, Jeffrey A. Newman, Samuel J. Schmidt, J. Anthony Tyson, and Risa H. Wechsler. Galaxy formation and evolution science in the era of the large synoptic survey telescope. *Nature Reviews Physics*, 2019. doi: 10.1038/s42254-019-0067-x. URL `https://doi.org/10.1038/s42254-019-0067-x`.

[223] Aldo Rodríguez-Puebla, Joel R. Primack, Vladimir Avila-Reese, and S. M. Faber. Constraining the galaxy-halo connection over the last 13.3 Gyr: star formation histories, galaxy mergers and structural properties. *MNRAS*, 470 (1):651–687, September 2017. doi: 10.1093/mnras/stx1172.

[224] O. Ronneberger, P. Fischer, and T. Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. *ArXiv e-prints*, May 2015.

[225] Philip Rosenfield, Jonathan Fay, Ronald K. Gilchrist, Chenzhou Cui, A. David Weigel, Thomas Robitaille, Oderah Justin Otor, and Alyssa Goodman. AAS WorldWide Telescope: A Seamless, Cross-platform Data Visualization Engine for Astronomy Research, Education, and Democratizing Data. *ApJS*, 236(1):22, May 2018. doi: 10.3847/1538-4365/aab776.

[226] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *Nature*, 323:533–536, October 1986. doi: 10.1038/323533a0.

[227] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. doi: 10.1007/s11263-015-0816-y.

[228] Ashwin Samudre, Lijo T. George, Mahak Bansal, and Yogesh Wadadekar. Data-efficient classification of radio galaxies. *MNRAS*, 509(2):2269–2280, January 2022. doi: 10.1093/mnras/stab3144.

[229] Joop Schaye, Robert A. Crain, Richard G. Bower, Michelle Furlong, Matthieu Schaller, Tom Theuns, Claudio Dalla Vecchia, Carlos S. Frenk, I. G. McCarthy, John C. Helly, Adrian Jenkins, Y. M. Rosas-Guevara, Simon D. M. White, Maarten Baes, C. M. Booth, Peter Camps, Julio F. Navarro, Yan Qu, Alireza Rahmati, Till Sawala, Peter A. Thomas, and

James Trayford. The EAGLE project: simulating the evolution and assembly of galaxies and their environments. *MNRAS*, 446(1):521–554, January 2015. doi: 10.1093/mnras/stu2058.

[230] Evan E. Schneider and Brant E. Robertson. CHOLLA: A New Massively Parallel Hydrodynamics Code for Astrophysical Simulation. *ApJS*, 217(2): 24, April 2015. doi: 10.1088/0067-0049/217/2/24.

[231] Nima Sedaghat and Ashish Mahabal. Effective image differencing with convolutional neural networks for real-time transient hunting. *MNRAS*, 476(4): 5365–5376, June 2018. doi: 10.1093/mnras/sty613.

[232] J. L. Sersic. *Atlas de Galaxias Australes*. Cordoba, Argentina: Observatorio Astronomico, 1968.

[233] S. Shen, H. J. Mo, S. D. M. White, M. R. Blanton, G. Kauffmann, W. Voges, J. Brinkmann, and I. Csabai. The size distribution of galaxies in the Sloan Digital Sky Survey. *MNRAS*, 343:978–994, August 2003. doi: 10.1046/j.1365-8711.2003.06740.x.

[234] K. Sheth, D. M. Elmegreen, B. G. Elmegreen, P. Capak, R. G. Abraham, E. Athanassoula, R. S. Ellis, B. Mobasher, M. Salvato, E. Schinnerer, N. Z. Scoville, L. Spalsbury, L. Strubbe, M. Carollo, M. Rich, and A. A. West. Evolution of the Bar Fraction in COSMOS: Quantifying the Assembly of the Hubble Sequence. *apj*, 675:1141–1155, March 2008. doi: 10.1086/524980.

[235] K. Sheth, M. Regan, J. L. Hinz, A. Gil de Paz, K. Menéndez-Delmestre, J.-C. Muñoz-Mateos, M. Seibert, T. Kim, E. Laurikainen, H. Salo, D. A. Gadotti, J. Laine, T. Mizusawa, L. Armus, E. Athanassoula, A. Bosma, R. J. Buta, P. Capak, T. H. Jarrett, D. M. Elmegreen, B. G. Elmegreen, J. H. Knapen, J. Koda, G. Helou, L. C. Ho, B. F. Madore, K. L. Masters, B. Mobasher, P. Ogle, C. Y. Peng, E. Schinnerer, J. A. Surace, D. Zaritsky, S. Comerón, B. de Swardt, S. E. Meidt, M. Kasliwal, and M. Aravena. The Spitzer Survey of Stellar Structure in Galaxies (S4G). *pasp*, 122:1397, December 2010. doi: 10.1086/657638.

[236] T. Shibuya, M. Ouchi, and Y. Harikane. Morphologies of ˜ 190,000 Galaxies at z = 0-10 Revealed with HST Legacy Data. I. Size Evolution. *ApJS*, 219: 15, August 2015. doi: 10.1088/0067-0049/219/2/15.

[237] B. D. Simmons, T. Melvin, C. Lintott, K. L. Masters, K. W. Willett, W. C. Keel, R. J. Smethurst, E. Cheung, R. C. Nichol, K. Schawinski, M. Rutkowski, J. S. Kartaltepe, E. F. Bell, K. R. V. Casteels, C. J. Conselice, O. Almaini, H. C. Ferguson, L. Fortson, W. Hartley, D. Kocevski, A. M. Koekemoer, D. H. McIntosh, A. Mortlock, J. A. Newman,

J. Ownsworth, S. Bamford, T. Dahlen, S. M. Faber, S. L. Finkelstein, A. Fontana, A. Galametz, N. A. Grogin, R. Grützbauch, Y. Guo, B. Häußler, K. J. Jek, S. Kaviraj, R. A. Lucas, M. Peth, M. Salvato, T. Wiklind, and S. Wuyts. Galaxy Zoo: CANDELS barred discs and bar fractions. *MNRAS*, 445:3466–3474, December 2014. doi: 10.1093/mnras/stu1817.

[238] B. D. Simmons, C. Lintott, K. W. Willett, K. L. Masters, J. S. Kartaltepe, B. Häußler, S. Kaviraj, C. Krawczyk, S. J. Kruk, D. H. McIntosh, R. J. Smethurst, R. C. Nichol, C. Scarlata, K. Schawinski, C. J. Conselice, O. Almaini, H. C. Ferguson, L. Fortson, W. Hartley, D. Kocevski, A. M. Koekemoer, A. Mortlock, J. A. Newman, S. P. Bamford, N. A. Grogin, R. A. Lucas, N. P. Hathi, E. McGrath, M. Peth, J. Pforr, Z. Rizer, S. Wuyts, G. Barro, E. F. Bell, M. Castellano, T. Dahlen, A. Dekel, J. Ownsworth, S. M. Faber, S. L. Finkelstein, A. Fontana, A. Galametz, R. Grützbauch, D. Koo, J. Lotz, B. Mobasher, M. Mozena, M. Salvato, and T. Wiklind. Galaxy Zoo: quantitative visual morphological classifications for 48 000 galaxies from CANDELS. *MNRAS*, 464:4420–4447, February 2017. doi: 10.1093/mnras/stw2587.

[239] R. E. Skelton, K. E. Whitaker, I. G. Momcheva, G. B. Brammer, P. G. van Dokkum, I. Labbé, M. Franx, A. van der Wel, R. Bezanson, E. Da Cunha, M. Fumagalli, N. Förster Schreiber, M. Kriek, J. Leja, B. F. Lundgren, D. Magee, D. Marchesini, M. V. Maseda, E. J. Nelson, P. Oesch, C. Pacifici, S. G. Patel, S. Price, H.-W. Rix, T. Tal, D. A. Wake, and S. Wuyts. 3D-HST WFC3-selected Photometric Catalogs in the Five CANDELS/3D-HST Fields: Photometry, Photometric Redshifts, and Stellar Masses. *ApJS*, 214: 24, October 2014. doi: 10.1088/0067-0049/214/2/24.

[240] I. Smail, A. Dressler, W. J. Couch, R. S. Ellis, A. Oemler, Jr., H. Butcher, and R. M. Sharples. A Catalog of Morphological Types in 10 Distant Rich Clusters of Galaxies. *ApJS*, 110:213–225, June 1997. doi: 10.1086/312997.

[241] Michael J. Smith and James E. Geach. Generative deep fields: arbitrarily sized, random synthetic astronomical images through deep learning. *arXiv e-prints*, art. arXiv:1904.10286, Apr 2019.

[242] Irwin Sobel and Gary Feldman. A 3x3 isotropic gradient operator for image processing",. *presented at the Stanford Artificial Intelligence Project*, 1968.

[243] Rachel S. Somerville and Romeel Davé. Physical Models of Galaxy Formation in a Cosmological Framework. *ARA&A*, 53:51–113, Aug 2015. doi: 10.1146/annurev-astro-082812-140951.

[244] D. Spergel, N. Gehrels, J. Breckinridge, M. Donahue, A. Dressler, B. S. Gaudi, T. Greene, O. Guyon, C. Hirata, J. Kalirai, N. J. Kasdin, W. Moos,

S. Perlmutter, M. Postman, B. Rauscher, J. Rhodes, Y. Wang, D. Weinberg, J. Centrella, W. Traub, C. Baltay, J. Colbert, D. Bennett, A. Kiessling, B. Macintosh, J. Merten, M. Mortonson, M. Penny, E. Rozo, D. Savransky, K. Stapelfeldt, Y. Zu, C. Baker, E. Cheng, D. Content, J. Dooley, M. Foote, R. Goullioud, K. Grady, C. Jackson, J. Kruk, M. Levine, M. Melton, C. Peddie, J. Ruffa, and S. Shaklan. Wide-Field InfraRed Survey Telescope-Astrophysics Focused Telescope Assets WFIRST-AFTA Final Report. *arXiv e-prints*, art. arXiv:1305.5422, May 2013.

[245] D. Spergel, N. Gehrels, C. Baltay, D. Bennett, J. Breckinridge, M. Donahue, A. Dressler, B. S. Gaudi, T. Greene, O. Guyon, C. Hirata, J. Kalirai, N. J. Kasdin, B. Macintosh, W. Moos, S. Perlmutter, M. Postman, B. Rauscher, J. Rhodes, Y. Wang, D. Weinberg, D. Benford, M. Hudson, W. S. Jeong, Y. Mellier, W. Traub, T. Yamada, P. Capak, J. Colbert, D. Masters, M. Penny, D. Savransky, D. Stern, N. Zimmerman, R. Barry, L. Bartusek, K. Carpenter, E. Cheng, D. Content, F. Dekens, R. Demers, K. Grady, C. Jackson, G. Kuan, J. Kruk, M. Melton, B. Nemati, B. Parvin, I. Poberezhskiy, C. Peddie, J. Ruffa, J. K. Wallace, A. Whipple, E. Wollack, and F. Zhao. Wide-Field InfrarRed Survey Telescope-Astrophysics Focused Telescope Assets WFIRST-AFTA 2015 Report. *arXiv e-prints*, art. arXiv:1503.03757, March 2015.

[246] Ashley Spindler, James E. Geach, and Michael J. Smith. AstroVaDEr: astronomical variational deep embedder for unsupervised morphological classification of galaxies and synthetic image generation. *MNRAS*, 502(1):985–1007, March 2021. doi: 10.1093/mnras/staa3670.

[247] Daniel P. Stark, Richard S. Ellis, Andrew Bunker, Kevin Bundy, Tom Targett, Andrew Benson, and Mark Lacy. The Evolutionary History of Lyman Break Galaxies Between Redshift 4 and 6: Observing Successive Generations of Massive Galaxies in Formation. *ApJ*, 697(2):1493–1511, June 2009. doi: 10.1088/0004-637X/697/2/1493.

[248] C. H Sudre, W. Li, T. Vercauteren, S. Ourselin, and M. J. Cardoso. Generalised Dice overlap as a deep learning loss function for highly unbalanced segmentations. *ArXiv e-prints*, July 2017.

[249] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[250] H. Tang, A. M. M. Scaife, O. I. Wong, and S. S. Shabala. Radio Galaxy Zoo: giant radio galaxy classification using multidomain deep learning. *MNRAS*, 510(3):4504–4524, March 2022. doi: 10.1093/mnras/stab3553.

[251] F. Tarsitano, W. G. Hartley, A. Amara, A. Bluck, C. Bruderer, M. Carollo, C. Conselice, P. Melchior, B. Moraes, A. Refregier, I. Sevilla-Noarbe, J. Woo, T. M. C. Abbott, S. Allam, J. Annis, S. Avila, M. Banerji, E. Bertin, D. Brooks, D. L. Burke, A. Carnero Rosell, M. Carrasco Kind, J. Carretero, C. E. Cunha, C. B. D'Andrea, L. N. da Costa, C. Davis, J. De Vicente, S. Desai, P. Doel, J. Estrada, J. Frieman, J. García-Bellido, D. Gruen, R. A. Gruendl, G. Gutierrez, D. Hollowood, K. Honscheid, D. J. James, T. Jeltema, E. Krause, K. Kuehn, N. Kuropatkin, O. Lahav, M. A. G. Maia, F. Menanteau, R. Miquel, A. A. Plazas, A. K. Romer, A. Roodman, E. Sanchez, B. Santiago, R. Schindler, M. Smith, R. C. Smith, M. Soares-Santos, F. Sobreira, E. Suchyta, M. E. C. Swanson, G. Tarle, D. Thomas, V. Vikram, and A. R. Walker. A catalogue of structural and morphological measurements for DES Y1. *MNRAS*, 481:2018–2040, December 2018. doi: 10.1093/mnras/sty1970.

[252] F. Tarsitano, C. Bruderer, K. Schawinski, and W. G. Hartley. Image feature extraction and galaxy classification: a novel and efficient approach with automated machine learning. *MNRAS*, 511(3):3330–3338, April 2022. doi: 10.1093/mnras/stac233.

[253] Duncan Tilley, Christopher W. Cleghorn, Kshitij Thorat, and Roger Deane. Point Proposal Network: Accelerating Point Source Detection Through Deep Learning. *arXiv e-prints*, art. arXiv:2008.02093, August 2020.

[254] S. Toft, P. van Dokkum, M. Franx, I. Labbe, N. M. Förster Schreiber, S. Wuyts, T. Webb, G. Rudnick, A. Zirm, M. Kriek, P. van der Werf, J. P. Blakeslee, G. Illingworth, H.-W. Rix, C. Papovich, and A. Moorwood. Hubble Space Telescope and Spitzer Imaging of Red and Blue Galaxies at z ˜ 2.5: A Correlation between Size and Star Formation Activity from Compact Quiescent Galaxies to Extended Star-forming Galaxies. *ApJ*, 671:285–302, December 2007. doi: 10.1086/521810.

[255] Emma Tolley, Damien Korber, Aymeric Galan, Austin Peel, Mark T. Sargent, Jean-Paul Kneib, Frederic Courbin, and Jean-Luc Starck. Lightweight HI source finding for next generation radio surveys. *arXiv e-prints*, art. arXiv:2204.09288, April 2022.

[256] Adam R. Tomczak, Ryan F. Quadri, Kim-Vy H. Tran, Ivo Labbé, Caroline M. S. Straatman, Casey Papovich, Karl Glazebrook, Rebecca Allen, Gabriel B. Brammer, Glenn G. Kacprzak, Lalitwadee Kawinwanichakij, Daniel D. Kelson, Patrick J. McCarthy, Nicola Mehrtens, Andrew J. Monson, S. Eric Persson, Lee R. Spitler, Vithal Tilvi, and Pieter van Dokkum. Galaxy Stellar Mass Functions from ZFOURGE/CANDELS: An Excess of

Low-mass Galaxies since z = 2 and the Rapid Buildup of Quiescent Galaxies. *ApJ*, 783(2):85, Mar 2014. doi: 10.1088/0004-637X/783/2/85.

[257] S. Tremaine, K. Gebhardt, R. Bender, G. Bower, A. Dressler, S. M. Faber, A. V. Filippenko, R. Green, C. Grillmair, L. C. Ho, J. Kormendy, T. R. Lauer, J. Magorrian, J. Pinkney, and D. Richstone. The Slope of the Black Hole Mass versus Velocity Dispersion Correlation. *ApJ*, 574:740–753, August 2002. doi: 10.1086/341002.

[258] I. Trujillo, G. Rudnick, H.-W. Rix, I. Labbé, M. Franx, E. Daddi, P. G. van Dokkum, N. M. Förster Schreiber, K. Kuijken, A. Moorwood, H. Röttgering, A. van der Wel, P. van der Werf, and L. van Starkenburg. The Luminosity-Size and Mass-Size Relations of Galaxies out to z ˜ 3. *ApJ*, 604:521–533, April 2004. doi: 10.1086/382060.

[259] I. Trujillo, N. M. Förster Schreiber, G. Rudnick, M. Barden, M. Franx, H.-W. Rix, J. A. R. Caldwell, D. H. McIntosh, S. Toft, B. Häussler, A. Zirm, P. G. van Dokkum, I. Labbé, A. Moorwood, H. Röttgering, A. van der Wel, P. van der Werf, and L. van Starkenburg. The Size Evolution of Galaxies since z˜3: Combining SDSS, GEMS, and FIRES. *ApJ*, 650:18–41, October 2006. doi: 10.1086/506464.

[260] James Trussler, Roberto Maiolino, Claudia Maraston, Yingjie Peng, Daniel Thomas, Daniel Goddard, and Jianhui Lian. Both starvation and outflows drive galaxy quenching. *MNRAS*, 491(4):5406–5434, February 2020. doi: 10.1093/mnras/stz3286.

[261] D. Tuccillo, M. Huertas-Company, E. Decencière, and S. Velasco-Forero. Deep learning for studies of galaxy morphology. In Massimo Brescia, S. G. Djorgovski, Eric D. Feigelson, Giuseppe Longo, and Stefano Cavuoti, editors, *Astroinformatics*, volume 325, pages 191–196, June 2017. doi: 10.1017/S1743921317000552.

[262] D. Tuccillo, M. Huertas-Company, E. Decencière, S. Velasco-Forero, H. Domínguez Sánchez, and P. Dimauro. Deep learning for galaxy surface brightness profile fitting. *MNRAS*, 475:894–909, March 2018. doi: 10.1093/mnras/stx3186.

[263] A. Vafaei Sadr, Etienne E. Vos, Bruce A. Bassett, Zafiirah Hosenie, N. Oozeer, and Michelle Lochner. DEEPSOURCE: point source detection using deep learning. *MNRAS*, 484(2):2793–2806, April 2019. doi: 10.1093/mnras/stz131.

[264] A. Vale and J. P. Ostriker. Linking halo mass to galaxy luminosity. *MNRAS*, 353(1):189–200, September 2004. doi: 10.1111/j.1365-2966.2004.08059.x.

[265] A. Vale and J. P. Ostriker. The non-parametric model for linking galaxy luminosity with halo/subhalo mass. *MNRAS*, 371(3):1173–1187, Sep 2006. doi: 10.1111/j.1365-2966.2006.10605.x.

[266] Stéfan van der Walt, Johannes L. Schönberger, Juan Nunez-Iglesias, François Boulogne, Joshua D. Warner, Neil Yager, Emmanuelle Gouillart, Tony Yu, and the scikit-image contributors. scikit-image: image processing in Python. *PeerJ*, 2:e453, 6 2014. ISSN 2167-8359. doi: 10.7717/peerj.453. URL `https://doi.org/10.7717/peerj.453`.

[267] A. van der Wel, E. F. Bell, B. Häussler, E. J. McGrath, Y.-Y. Chang, Y. Guo, D. H. McIntosh, H.-W. Rix, M. Barden, E. Cheung, S. M. Faber, H. C. Ferguson, A. Galametz, N. A. Grogin, W. Hartley, J. S. Kartaltepe, D. D. Kocevski, A. M. Koekemoer, J. Lotz, M. Mozena, M. A. Peth, and C. Y. Peng. Structural Parameters of Galaxies in CANDELS. *ApJS*, 203: 24, December 2012. doi: 10.1088/0067-0049/203/2/24.

[268] A. van der Wel, M. Franx, P. G. van Dokkum, R. E. Skelton, I. G. Momcheva, K. E. Whitaker, G. B. Brammer, E. F. Bell, H.-W. Rix, S. Wuyts, H. C. Ferguson, B. P. Holden, G. Barro, A. M. Koekemoer, Y.-Y. Chang, E. J. McGrath, B. Häussler, A. Dekel, P. Behroozi, M. Fumagalli, J. Leja, B. F. Lundgren, M. V. Maseda, E. J. Nelson, D. A. Wake, S. G. Patel, I. Labbé, S. M. Faber, N. A. Grogin, and D. D. Kocevski. 3D-HST+CANDELS: The Evolution of the Galaxy Size-Mass Distribution since z = 3. *ApJ*, 788:28, June 2014. doi: 10.1088/0004-637X/788/1/28.

[269] P. G. van Dokkum, K. E. Whitaker, G. Brammer, M. Franx, M. Kriek, I. Labbé, D. Marchesini, R. Quadri, R. Bezanson, G. D. Illingworth, A. Muzzin, G. Rudnick, T. Tal, and D. Wake. The Growth of Massive Galaxies Since z = 2. *ApJ*, 709:1018–1041, February 2010. doi: 10.1088/0004-637X/709/2/1018.

[270] Guido Van Rossum and Fred L Drake Jr. *Python tutorial*. Centrum voor Wiskunde en Informatica Amsterdam, The Netherlands, 1995.

[271] G. Vaucouleurs. Classification and Morphology of External Galaxies. *Handbuch der Physik*, 53:275–310, 1959. doi: 10.1007/978-3-642-45932-0_7.

[272] J. Vega-Ferrero, H. Domínguez Sánchez, M. Bernardi, M. Huertas-Company, R. Morgan, B. Margalef, M. Aguena, S. Allam, J. Annis, S. Avila, D. Bacon, E. Bertin, D. Brooks, A. Carnero Rosell, M. Carrasco Kind, J. Carretero, A. Choi, C. Conselice, M. Costanzi, L. N. da Costa, M. E. S. Pereira, J. De Vicente, S. Desai, I. Ferrero, P. Fosalba, J. Frieman, J. García-Bellido, D. Gruen, R. A. Gruendl, J. Gschwend, G. Gutierrez, W. G. Hartley, S. R. Hinton, D. L. Hollowood, K. Honscheid, B. Hoyle, M. Jarvis, A. G.

Kim, K. Kuehn, N. Kuropatkin, M. Lima, M. A. G. Maia, F. Menanteau, R. Miquel, R. L. C. Ogando, A. Palmese, F. Paz-Chinchón, A. A. Plazas, A. K. Romer, E. Sanchez, V. Scarpine, M. Schubnell, S. Serrano, I. Sevilla-Noarbe, M. Smith, E. Suchyta, M. E. C. Swanson, G. Tarle, F. Tarsitano, C. To, D. L. Tucker, T. N. Varga, and R. D. Wilkinson. Pushing automated morphological classifications to their limits with the Dark Energy Survey. *MNRAS*, 506(2):1927–1943, September 2021. doi: 10.1093/mnras/stab594.

[273] Malin Velander, Edo van Uitert, Henk Hoekstra, Jean Coupon, Thomas Erben, Catherine Heymans, Hendrik Hildebrandt, Thomas D. Kitching, Yannick Mellier, Lance Miller, Ludovic Van Waerbeke, Christopher Bonnett, Liping Fu, Stefania Giodini, Michael J. Hudson, Konrad Kuijken, Barnaby Rowe, Tim Schrabback, and Elisabetta Semboloni. CFHTLenS: the relation between galaxy dark matter haloes and baryons from weak gravitational lensing. *MNRAS*, 437(3):2111–2136, January 2014. doi: 10.1093/mnras/stt2013.

[274] Bruno Villasenor, Brant Robertson, Piero Madau, and Evan Schneider. Effects of Photoionization and Photoheating on Lyα Forest Properties from Cholla Cosmological Simulations. *ApJ*, 912(2):138, May 2021. doi: 10.3847/1538-4357/abed5a.

[275] Bruno Villasenor, Brant Robertson, Piero Madau, and Evan Schneider. Inferring the Thermal History of the Intergalactic Medium from the Properties of the Hydrogen and Helium Lyman-alpha Forest. *arXiv e-prints*, art. arXiv:2111.00019, October 2021.

[276] Mark Vogelsberger, Federico Marinacci, Paul Torrey, and Ewald Puchwein. Cosmological simulations of galaxy formation. *Nature Reviews Physics*, 2 (1):42–66, January 2020. doi: 10.1038/s42254-019-0127-2.

[277] Mike Walmsley, Lewis Smith, Chris Lintott, Yarin Gal, Steven Bamford, Hugh Dickinson, Lucy Fortson, Sandor Kruk, Karen Masters, Claudia Scarlata, Brooke Simmons, Rebecca Smethurst, and Darryl Wright. Galaxy Zoo: probabilistic morphology through Bayesian CNNs and active learning. *MNRAS*, 491(2):1554–1574, January 2020. doi: 10.1093/mnras/stz2816.

[278] Philip Walton. First contentful paint (fcp), 2022. URL `https://web.dev/fcp/`.

[279] Pascal Wassong and Sébastien Derriere. The ASTRODEEP Frontier Fields Portal. In Marco Molinaro, Keith Shortridge, and Fabio Pasian, editors, *Astronomical Data Analysis Software and Systems XXVI*, volume 521 of *Astronomical Society of the Pacific Conference Series*, page 268, October 2019.

[280] Risa H. Wechsler and Jeremy L. Tinker. The Connection Between Galaxies and Their Dark Matter Halos. *ARA&A*, 56:435–487, Sep 2018. doi: 10.1146/annurev-astro-081817-051756.

[281] A. K. Weigel, K. Schawinski, N. Caplar, A. Carpineti, R. E. Hart, S. Kaviraj, W. C. Keel, S. J. Kruk, C. J. Lintott, R. C. Nichol, B. D. Simmons, and R. J. Smethurst. Galaxy Zoo: Major Galaxy Mergers Are Not a Significant Quenching Pathway. *ApJ*, 845:145, August 2017. doi: 10.3847/1538-4357/aa8097.

[282] D. C. Wells, E. W. Greisen, and R. H. Harten. FITS - a Flexible Image Transport System. *A&AS*, 44:363, June 1981.

[283] K. E. Whitaker, M. Franx, R. Bezanson, G. B. Brammer, P. G. van Dokkum, M. T. Kriek, I. Labbé, J. Leja, I. G. Momcheva, E. J. Nelson, J. R. Rigby, H.-W. Rix, R. E. Skelton, A. van der Wel, and S. Wuyts. Galaxy Structure as a Driver of the Star Formation Sequence Slope and Scatter. *ApJL*, 811: L12, September 2015. doi: 10.1088/2041-8205/811/1/L12.

[284] K. W. Willett, K. Schawinski, B. D. Simmons, K. L. Masters, R. A. Skibba, S. Kaviraj, T. Melvin, O. I. Wong, R. C. Nichol, E. Cheung, C. J. Lintott, and L. Fortson. Galaxy Zoo: the dependence of the star formation-stellar mass relation on spiral disc morphology. *MNRAS*, 449:820–827, May 2015. doi: 10.1093/mnras/stv307.

[285] Kyle W. Willett, Chris J. Lintott, Steven P. Bamford, Karen L. Masters, Brooke D. Simmons, Kevin R. V. Casteels, Edward M. Edmondson, Lucy F. Fortson, Sugata Kaviraj, William C. Keel, Thomas Melvin, Robert C. Nichol, M. Jordan Raddick, Kevin Schawinski, Robert J. Simpson, Ramin A. Skibba, Arfon M. Smith, and Daniel Thomas. Galaxy Zoo 2: detailed morphological classifications for 304 122 galaxies from the Sloan Digital Sky Survey. *MNRAS*, 435:2835–2860, November 2013. doi: 10.1093/mnras/stt1458.

[286] Kyle W. Willett, Melanie A. Galloway, Steven P. Bamford, Chris J. Lintott, Karen L. Masters, Claudia Scarlata, B. D. Simmons, Melanie Beck, Carolin N. Cardamone, Edmond Cheung, Edward M. Edmondson, Lucy F. Fortson, Roger L. Griffith, Boris Häußler, Anna Han, Ross Hart, Thomas Melvin, Michael Parrish, Kevin Schawinski, R. J. Smethurst, and Arfon M. Smith. Galaxy Zoo: morphological classifications for 120 000 galaxies in HST legacy imaging. *MNRAS*, 464:4176–4203, February 2017. doi: 10.1093/mnras/stw2568.

[287] C. C. Williams, M. Giavalisco, R. Bezanson, N. Cappelluti, P. Cassata, T. Liu, B. Lee, E. Tundo, and E. Vanzella. Morphology Dependence of Stellar Age in Quenched Galaxies at Redshift ˜1.2: Massive Compact Galax-

ies Are Older than More Extended Ones. *ApJ*, 838:94, April 2017. doi: 10.3847/1538-4357/aa662f.

[288] C. C. Williams, E. Curtis-Lake, K. N. Hainline, J. Chevallard, B. E. Robertson, S. Charlot, R. Endsley, D. P. Stark, C. N. A. Willmer, S. Alberts, R. Amorin, S. Arribas, S. Baum, A. Bunker, S. Carniani, S. Crandall, E. Egami, D. J. Eisenstein, P. Ferruit, B. Husemann, M. V. Maseda, R. Maiolino, T. D. Rawle, M. Rieke, R. Smit, S. Tacchella, and C. J. Willott. The JWST Extragalactic Mock Catalog: Modeling Galaxy Populations from the UV through the Near-IR over 13 Billion Years of Cosmic History. *ApJS*, 236:33, June 2018. doi: 10.3847/1538-4365/aabcbb.

[289] Chen Wu, Oiwei Ivy Wong, Lawrence Rudnick, Stanislav S. Shabala, Matthew J. Alger, Julie K. Banfield, Cheng Soon Ong, Sarah V. White, Avery F. Garon, Ray P. Norris, Heinz Andernach, Jean Tate, Vesna Lukic, Hongming Tang, Kevin Schawinski, and Foivos I. Diakogiannis. Radio Galaxy Zoo: CLARAN - a deep learning classifier for radio morphologies. *MNRAS*, 482(1):1211–1230, January 2019. doi: 10.1093/mnras/sty2646.

[290] S. Wuyts, N. M. Förster Schreiber, A. van der Wel, B. Magnelli, Y. Guo, R. Genzel, D. Lutz, H. Aussel, G. Barro, S. Berta, A. Cava, J. Graciá-Carpio, N. P. Hathi, K.-H. Huang, D. D. Kocevski, A. M. Koekemoer, K.-S. Lee, E. Le Floc'h, E. J. McGrath, R. Nordon, P. Popesso, F. Pozzi, L. Riguccini, G. Rodighiero, A. Saintonge, and L. Tacconi. Galaxy Structure and Mode of Star Formation in the SFR-Mass Plane from z ˜ 2.5 to z ˜ 0.1. *ApJ*, 742: 96, December 2011. doi: 10.1088/0004-637X/742/2/96.

[291] Hui Xie, Zhe Pan, Leixin Zhou, Fahim A Zaman, Danny Chen, Jost B Jonas, Yaxing Wang, and Xiaodong Wu. Globally Optimal Segmentation of Mutually Interacting Surfaces using Deep Learning. *arXiv e-prints*, art. arXiv:2007.01259, July 2020.

[292] Xiaoju Xu, Saurabh Kumar, Idit Zehavi, and Sergio Contreras. Predicting halo occupation and galaxy assembly bias with machine learning. *MNRAS*, 507(4):4879–4899, November 2021. doi: 10.1093/mnras/stab2464.

[293] M. Yano, M. Kriek, A. van der Wel, and K. E. Whitaker. The Relation between Galaxy Structure and Spectral Type: Implications for the Buildup of the Quiescent Galaxy Population at $0.5 < z < 2.0$. *ApJL*, 817:L21, February 2016. doi: 10.3847/2041-8205/817/2/L21.

[294] Joshua Yao-Yu Lin, Song-Mao Liao, Hung-Jin Huang, Wei-Ting Kuo, and Olivia Hsuan-Min Ou. Galaxy Morphological Classification with Efficient Vision Transformer. *arXiv e-prints*, art. arXiv:2110.01024, October 2021.

[295] Lei Yu, Bin Liu, Yan Zhu, Ru-Rong Chen, Hong-Wei Xi, Cheng-Jin Jin, and Bo Peng. A deep learning framework for Square Kilometre Array Science Data Challenge 1. *MNRAS*, 511(3):4305–4315, April 2022. doi: 10.1093/mnras/stac281.

[296] W. Yu, M. Carrasco Kind, and R. J. Brunner. Vizic: A Jupyter-based interactive visualization tool for astronomical catalogs. *Astronomy and Computing*, 20:128–139, July 2017. doi: 10.1016/j.ascom.2017.06.004.

[297] H. Zhang, J. R. Primack, S. M. Faber, D. C. Koo, A. Dekel, Z. Chen, D. Ceverino, Y.-Y. Chang, J. J. Fang, Y. Guo, L. Lin, and A. v. d. Wel. The evolution of galaxy shapes in CANDELS: from prolate to discy. *MNRAS*, 484:5170–5191, April 2019. doi: 10.1093/mnras/stz339.

[298] Xia Zhang, Foivos I. Diakogiannis, Richard Dodson, and Andreas Wicenec. Radio Transient Detection with Closure Products and Machine Learning. *arXiv e-prints*, art. arXiv:2204.01958, April 2022.

[299] Yanning Zhou, Hao Chen, Huangjing Lin, and Pheng-Ann Heng. Deep Semi-supervised Knowledge Distillation for Overlapping Cervical Cell Instance Segmentation. *arXiv e-prints*, art. arXiv:2007.10787, July 2020.

[300] Hanjue Zhu, Camille Avestruz, and Nickolay Y. Gnedin. Cosmic Reionization On Computers: The Galaxy-Halo Connection between $5 \leq z \leq 10$. *ApJ*, 899(2):137, August 2020. doi: 10.3847/1538-4357/aba26d.

[301] Xiao-Pan Zhu, Jia-Ming Dai, Chun-Jiang Bian, Yu Chen, Shi Chen, and Chen Hu. Galaxy morphology classification with deep convolutional neural networks. *Ap&SS*, 364(4):55, April 2019. doi: 10.1007/s10509-019-3540-1.

[302] Yongda Zhu, George D. Becker, Sarah E. I. Bosman, Laura C. Keating, Holly M. Christenson, Eduardo Bañados, Fuyan Bian, Frederick B. Davies, Valentina D'Odorico, Anna-Christina Eilers, Xiaohui Fan, Martin G. Haehnelt, Girish Kulkarni, Andrea Pallottini, Yuxiang Qin, Feige Wang, and Jinyi Yang. Chasing the Tail of Cosmic Reionization with Dark Gap Statistics in the Lyα Forest over $5 < z < 6$. *ApJ*, 923(2):223, December 2021. doi: 10.3847/1538-4357/ac26c2.

[303] A. W. Zirm, A. van der Wel, M. Franx, I. Labbé, I. Trujillo, P. van Dokkum, S. Toft, E. Daddi, G. Rudnick, H.-W. Rix, H. J. A. Röttgering, and P. van der Werf. NICMOS Imaging of DRGs in the HDF-S: A Relation between Star Formation and Size at z ˜ 2.5. *ApJ*, 656:66–72, February 2007. doi: 10.1086/510713.