Predicting Localized Primordial Star Formation With Deep Convolutional Neural Networks

AZTON I. WELLS¹ AND MICHAEL L. NORMAN¹,²

¹Center for Astrophysics and Space Sciences
University of California, San Diego, La Jolla, CA, 92093
²San Diego Supercomputer Center
University of California, San Diego, La Jolla, CA, 92093

(Dated: November 4, 2020)

ABSTRACT

We present a series of deep convolutional neural networks used to predict the location of primordial star forming regions in hydrodynamic cosmological simulations. The predictor consists of two deep neural networks: one predicts subvolumes of size $10^3$ comoving kpc$^3$ which host star forming regions, another predicts which voxels within the subvolume may be forming stars. We find that the combined deep neural network module predicts primordial star forming regions with accuracy $\geq 99.8\%$, and star formation can be localized within the region to $\lesssim 5^3$ voxels ($\sim 1.60$ comoving kpc$^3$) in simulations where the total volume is $17.58$ comoving Mpc$^3$. We further verify the module by applying it to spatially under-resolved hydrodynamical simulations without primordial star formation to locate star-forming regions. When applied to a simulation with the same mass resolution as the training data simulation but of lower spatial resolution, the model reproduces a reasonable number of star-forming regions as compared to the highly resolved full physics simulations that explicitly models primordial star formation and feedback. When applied to simulations with coarser mass resolution, we find that the method is capable of finding star-forming regions at later redshifts, but cannot completely overcome the delayed structure formation that is a consequence of lower mass and force resolution. This model predicts first generation star formation sites without the use of halo finding, allowing prediction of primordial stars in spatially under-resolved simulations that cannot accurately resolve primordial star forming halos.

Keywords: Deep Learning, Convolutional Neural Networks, Population III, First Stars

1. INTRODUCTION

Despite the rise of petascale computing, astrophysical simulations continue to push the limits of the most advanced supercomputers. While dark matter (DM) or hydrodynamic gas-only simulations can now simulate massive volumes (e.g., Vogelsberger et al. 2014), the inclusion of more complete physical models, such as resolved star formation and feedback (SFF) processes, severely limits the volume of feasible simulations (Hopkins et al. 2018; Smith et al. 2015). When attempting precision modelling of galaxy formation with associated SFF, one must usually choose whether to simulate the primordial star (Pop III ) formation era, or to adopt a simplification scheme to compensate for the lack thereof. Starting at $z \simeq 30$, Pop III stars begin to form from pristine (H, He, H$_2$) gas in mini-halos with virial mass $M_{\text{vir}} \gtrsim 10^{5.5-6.5}$ M$_\odot$ (Bromm 2013). After formation, Pop III stars may directly collapse to black holes (BHs), or if formed in the right mass range, live out a main sequence followed by a supernova (SN) of some type (Woosley & Heger 2015). The SNe considered in this work fall into three categories determined by the stellar mass of their progenitor: Type-II supernovae (SNe), hypernovae (HNe), and pair-instability supernovae (PISNe). Depending on the mass of the star and mass of its host halo, the SN may completely disrupt the halo, ejecting the majority of gas and metal (Whalen et al. 2008b), effectively preventing any further star formation until the gas has recycled back into the halo and cooled (Tumlinson et al. 2017).

aiwells@ucsd.edu
On the other end of the feedback spectrum, Pop III stars that will collapse to a BH will still emit ionizing radiation for their main sequence lifetime before collapse. Radiative feedback has been shown to shut down continuing star formation (Whalen et al. 2008a; Wise et al. 2012; Hopkins et al. 2019), limiting the conversion of gas to stars in star-forming regions. Unfortunately, having the extremely high resolution ($\lesssim 20$ pc cm$^{-1}$, $M_{DM} \lesssim 10^4 M_\odot$) required to precisely model Pop III SFF means that simulations of large (i.e., statistically relevant to the observable universe) volumes have not been able to approach modern redshifts (Wise et al. 2012; Xu et al. 2016; Smith et al. 2015; Hopkins et al. 2019). To avoid the computational expense of precisely modelling the Pop III era, some practitioners adopt a metallicity floor (Hopkins et al. 2018), while others altogether disregard the effect of Pop III pre-enrichment on star formation (Vogelsberger et al. 2014). Neither of these simplifications account for the fact that enrichment by Pop III stars is (a) non-uniformly distributed in space, (b) rare, and (c) necessary for enriched star formation. In fact, recent work (Jeon et al. 2017; Hicks et al. 2020) has shown that extremely low metallicity stars may form in halos that have been enriched by an external Pop III SN event. This represents a sequence of star formation that is impossible to model using the above simplifications.

While artificial neural networks (ANN), and particularly deep convolutional neural networks (CNN) have been used for image recognition for nearly a decade since AlexNet (Krizhevskiy et al. 2012), they have also begun to foray into varied scientific applications, e.g., Aniyan & Thorat (2017); Jin et al. (2018); Mohan et al. (2019); Zhang et al. (2019); Hajiabadi et al. (2020). Inspired by the monumental advancements in the field, this work will begin to lay the groundwork for a novel surrogate model for pre-enrichment using trained CNN to identify primordial star forming regions without resort to halo finding and predict their feedback influence, relaxing the severe resolution requirements somewhat. The rest of this paper outlines the development of the first phase of this feedback algorithm, to identify star forming regions, as follows: In Section 2 we outline the simulations and methods used to generate training data; Section 3 presents the network architectures used and discusses the parameter exploration that led to our design; finally, we test the CNN designs and present the results in Section 4, and finish with a discussion in Section 5. The python code used to generate this project is available at https://github.com/axton/Pop3Net.

2. SIMULATIONS AND DATA

2.1. PHX256 Simulations

Training neural networks depends on copious amounts of data. To that end, we used the astrophysical adaptive mesh refinement (AMR) simulation code Enzo (Bryan et al. 2014; Brummel-Smith et al. 2019) to produce two simulations (PHX256-1 and PHX256-2) from which to draw train/validation/test data. Table 1 summarizes the simulations used within this work, including simulations used for testing and generalization studies. The PHX256-1,2 simulations are modeled after the Renaissance Simulations (Xu et al. 2016) in terms of included physics, parameters, and resolution. All simulations in this work were run on the TACC-Frontera supercomputer, with the PHX256-1,2 consuming ~ 500 k cpu-hours total. Both simulations have identical Planck 2014 cosmological parameters: $\Omega_L = 0.6889, \Omega_m = 0.3111, \Omega_b = 0.048975, \sigma_8 = 0.811, n = 0.965$ (Ade et al. 2014), but use different random seeds in the initial conditions, generated using MUSIC (Hahn & Abel 2011). Whereas the Renaissance Simulations used a zoom-in simulation setup with a hierarchy of static nested grids, we simulate a periodic box of size comparable to the finest nested grid in the former, at identical mass and spatial resolution. The simulation volume for both PHX256 simulations is $2.61^3$ (Mpc)$^3$ in extent, with $256^3$ root grid cells and dark matter (DM) particles and 9 levels AMR; the cell width at the deepest AMR level is 19 pc. With the given cosmology, the DM particles have mass $2.38 \times 10^4 M_\odot (M_{dm})$, and the baryon mass ($M_b$) in a root grid cell of average density is $1.17 \times 10^3 M_\odot$.

Refinement of the AMR grid occurs based on $M_b$ and $M_{dm}$, the baryonic and dark matter mass of a cell, respectively, with the minimum mass for refinement being 3 times the initial values above ($M_{min}$). At each iteration, any grid cell on level $l$ with $M_{cell} \geq M_{min} \times 2^{-0.4l}$ will be refined to the next level; the grid refinement is super-lagrangian, so that the AMR levels will have $M_{cell} < M_{root grid}$. In addition to the mass criteria, any cells with Pop III star particles are refined to the level such that the supernova radius parameter (10 pc) is resolved by $\geq 4$ cell widths. With resolved halos are defined as those with $\geq 100$ DM particles, the minimum mass of a resolved halo in the PHX256 simulations is $2.3 \times 10^6 M_\odot$. We include radiation hydrodynamics using the Moray ray-tracing solver (Wise & Abel 2011), with Pop III and enriched star clusters as radiating sources. A uniform, $z$-dependent Lyman-Werner background is included to account for H$_2$ dissociating.

1 Comoving units have -cm appended to the base unit throughout this paper; the base unit is assumed to be proper.
radiation sources originating from outside the simulation volume. Nonequilibrium primordial gas chemistry for the 9 species H, H$^+$, H$^-$, H$_2$, H$_2^+$, e$^-$, He, He$^+$, He$^{++}$ is computed, and radiative heating and cooling of the gas includes both primordial and metal-line cooling contributions as in Smith et al. 2008.

2.2. Star Formation

Since this work aims to predict star formation, we will briefly review the relevant algorithms here. For a more detailed review of star formation in Enzo, please refer to the Enzo documentation\textsuperscript{2}. The PHX256 simulations both include Pop III single star and Pop II star cluster formation. At each grid timestep, the finest grid cell at each location is evaluated for star formation. The Pop III formation criteria and their parameter values which are checked are:

- Number density $n \geq 100$.
- H$_2$ density: $\rho_{H2}/\rho_b \geq 10^{-3}$.
- Metallicity$^3$: $Z \leq Z_c$ with $Z_c = -5.5$ for Pop III formation. Pop II formation requires $Z \geq Z_c$.
- The freefall time should be less than the cooling time: $t_{ff} < t_{cool}$.
- Converging gas flow: $\nabla \cdot v_{gas} < 0$.

If these criteria are met, a Pop II ($Z > Z_c$) or Pop III ($Z < Z_c$) star particle is formed from a sphere containing twice the mass of the star centered on the star forming grid cell. In the Pop III case, the particle represents a single star with mass taken from the modified Salpeter IMF of the form

$$f(\log M)dM = M^{-1.3}\exp\left[-\left(\frac{M_{\text{char}}}{M}\right)^{1.6}\right]dM,$$

with $M_{\text{char}} = 20$ M$_\odot$. Pop II particles are formed if $Z > Z_c$ (the H$_2$ requirement is ignored). In this case, a single particle represents a radiating stellar cluster with $M_{\text{min}} = 1000$ M$_\odot$, assuming an unmodified Salpeter IMF. Although the grid cell that was identified for Pop II star creation must have $Z > Z_{crit}$, the gas surrounding likely has lower or higher metallicity: when mass-averaged into the star particle, the resulting particle may have $Z \neq Z_{cell}$.

2.3. Data reduction and preparation

Both simulations output the simulation state every 200 kyr, from $z = 30$ to the final redshift noted in Table 1. Between $z = 30$ to $z = 10$, there will be 1806 individual data outputs that are a snapshot of the entire simulation domain. This project serves as a proof of concept, so uses the simulations at their most progressed state ($z_{\text{final}}$ in Table 1). Future work will incorporate all available data, and progress simulations to $z \leq 10$ to broaden the training data set as much as possible. The simulation outputs occupy $> 75$ TB in their unprocessed state at the current redshifts. Although copious, this raw data is not acceptable for input to a neural network, so the both simulations are post-processed to generate training, validation, and testing data, as well as to reduce the size of the data to a more manageable footprint.

To generate training data, we use pairs of snapshots from a single simulation, $\{D, D_{-1}\}$, where $D$ is the current output and $D_{-1}$ is the output immediately prior. Each $D$ output is checked for new star particles since $D_{-1}$. If found, we generate a uniform grid with volume $10^3$ kpc$^3$ centered on the new star particle in $D$ using \texttt{YT} (Turk et al. 2011). Each hydrodynamic and color field in the region in $D_{-1}$ is saved to this cube. To label the star location, we flag cells of a $3^3$ cube centered on the star-forming grid cell as star-forming voxels. For sample augmentation, we additionally generate $n_{\text{shifted}}$ volumes that are centered randomly, but still contain the target star particle. After all stars are accounted

\begin{table}[h]
\centering
\begin{tabular}{|c|c|c|c|c|c|}
\hline
Name & $dx_{\text{root}}$ [kpc cm] & $dx_{\text{min}}$ [kpc cm] & $L_{\text{max}}$ & $M_{\text{DM}} [M_\odot]$ & $z_{\text{final}}$ \\
\hline
PHX256-1 & 10.183 & 19.89 & 9 & $2.384 \times 10^4$ & 13.57 \\
PHX256-2 & 10.183 & 19.89 & 9 & $2.384 \times 10^4$ & 13.86 \\
P3N-128 & 20.366 & 156.8 & 7 & $1.910 \times 10^5$ & 10.0 \\
PHX256-HYD & 10.183 & 156.8 & 6 & $2.384 \times 10^4$ & 18.05 \\
\hline
\end{tabular}
\caption{Source simulations for train/validation/test data.}
\end{table}

\textsuperscript{2} https://enzo.readthedocs.io/en/latest
\textsuperscript{3} $Z$ denotes log metallicity relative to solar. With metal mass $M_z$, $Z = \log \frac{M_z}{M_\odot} - \log \frac{M_{\odot}}{M_\odot}$
for, we generate random samples of regions in $D_{-1}$ with the restriction that any candidate volume must have a volume average density $\langle \rho \rangle$ greater than the mean $\langle \rho \rangle$: $\langle \rho \rangle / \rho \geq 1.0$. There are $n_{\text{star}}(1 + n_{\text{shifted}}) + n_{\text{star}} \times 50$ samples generated for each snapshot of the simulation, where we set $n_{\text{shifted}} = 5$.

For PHX256-1, samples are separated into training and validation, which is randomly determined for each volume during initial data reduction while ensuring that volumes with the same star particle belong to the same split. This proof of concept uses 81,512 (9380 with stars) training samples with 6,439 (412 with stars) validation samples. There is a lower ratio of star-containing volumes in validation because we take care to remove volumes that also contain a star in the training dataset. In principle, it is possible (and happens within the PHX256 simulations) to form several Pop III stars in close proximity within a short time period, i.e., a co-evolving Pop III cluster. If a star particle is assigned to training, but its neighbor is assigned to validation, we could see neighboring volumes and star-forming regions in the training and validation data. To guarantee a pristine test dataset, we perform data reduction as above on PHX256-2, generating 27,232 (3,782 with stars) samples for the testing dataset.

In the final step of preparation, each sample is scaled by the standard deviation and mean of all training data, i.e., for density, $\rho_{\text{scaled}} = (\rho - \langle \rho \rangle) / \langle \sigma(\rho) \rangle$ for $\langle \rho \rangle$, the average voxel density in training and $\langle \sigma(\rho) \rangle$, the averaged standard deviation of voxel density in training data.

In order to maximize the network training data, we implicitly assume that Pop III formation is independent of $z$. In other words, we train our networks on star formation events across a range of redshifts without treating data at different redshifts as fundamentally different. The justification for this is the set of criteria that must be met to form a Pop III star in a fully resolved simulation (bulleted list above) depend only on local conditions which are decoupled from global conditions. The one exception is the globally evolving UV Lyman-Werner background, which affects the $H_2$ fraction of the gas, which in turn enables the gas to cool and condense into stars. However, since our model includes the $H_2$ fraction, this global influence is taken into account. In support of our assertion that redshift is not an important parameter of the problem is the analysis of Xu et al. (2016), who showed that the statistical properties of high-z dwarf galaxies in the Renaissance Simulations were insensitive to redshift, but rather principally dependent on the halo virial mass. There, galaxies form from gas pre-enriched by Pop III stars whose formation process is modeled directly.

### 3. NETWORK DESIGN

In a grid-based simulation code like Enzo, the state of the simulation is stored on a fixed grid, where quantities like baryon density are stored as cell-centered quantities. Every quantity that is advanced by the simulation is tracked on the grid, or derived from quantities that are tracked there. Therefore, when inspecting a hydrodynamical simulation, it can be viewed as a 3-dimensional volumetric image, where each hydrodynamic quantity is analogous to the RGB color channels of a typical image. Of course, different hydrodynamic fields may carry nearly independent information, so the analogy to color channels is only surface deep; nonetheless, this logic leads us to classify volumes of a simulation as if they are volumetric images with hydrodynamic information as channels. Such an approach lets us take advantage of the numerous developments in the field of computer vision for this problem.

We use a two-stage approach to predict localized Pop III star formation. The first stage ($S1$) is a classifier used to
to quickly decide if a potential region is capable of forming stars in any of its volume. If the classifier identifies a star-forming region, then a more complicated voxel segmentation network ($S_2$) is used to identify which voxels within the volume are forming stars. The module composed of $S_1$ and $S_2$ (StarFind) must agree on the star-forming state of the volume in order to classify the volume as star-forming. All of the architectures presented here are implemented in Pytorch (Paszke et al. 2019) and trained using 4 K80 GPUs on the SDSC-Comet supercomputer.

### 3.1. Stage 1: Classification Network

We tested various network architectures for classifying regions as star-forming (SFR) or non-star-forming (nSFR). Initial tests included 3D adapted versions of 16-layer ResNet (He et al. 2015), 16-layer DenseNet (Huang et al. 2016), and GoogLeNet as described in Szegedy et al. (2014), however it quickly became apparent that these models were too complex for the task at hand. To reduce the model complexity, this work uses small classifiers based on ideas from those seminal papers, with vastly reduced network depth. In addition to changing the depth of the network, this work also implements several changes to the above architecture designs:

- Number of input channels represents input hydrodynamic fields and is a hyperparameter.
- The network width is another hyperparameter, tunable using the number of channels at the first layer, $N_0$.
- Each convolution, regardless of base architecture, is followed by dropout, batch normalization, and ReLU activation, with exceptions in Inception blocks (IBs) (Figure 1).

Figure 1 describes the basic convolution and IBs used throughout this work. By default, the convolution block has a 3D $K^3$ convolutional filter, dropout with probability 0.2, batch normalization, and ReLU activation. The default stride and padding are set such that the input dimension is unchanged. Note that dropout zeros the output of an entire channel in the output of the layer (Tompson et al. 2014). The IBs presented in Figure 1 are used in the classifier architecture. In principle, all channel numbers in the IB are tunable hyperparameters, however this parameter space was not explored in this work.

In an effort to create the smallest effective network architecture, we implemented the small classifier networks as shown in Figure 2. These architectures are inspired by GoogLeNet, DenseNet, and ResNet, but are severely truncated. After initial convolutions common to each architecture, each small classifier network is composed of only two key blocks, as seen in Figure 2. The key blocks may be inception modules (SINet), standard convolutions with residual skips (SRNet), or densely connected convolutions (SDNet) (see Figure 3). The rest of this work will focus exclusively on the use of small SINet, SDNet, and SRNet, as there was no appreciable gain in accuracy to using the full architectures from ResNet,
DenseNet, or GoogLeNet, while inference and training time was significantly slower.

Before final classification, we remove dependence on input dimension by reducing to a \( \{N_{\text{channel}}, 2, 2\} \) volume via average pooling before the fully-connected layers. All network variations use Adam optimization (Kingma & Ba 2014) and cross entropy loss given by

\[
L(\hat{y}, y, c) = \left\{ w(c) \left[ -y_c \hat{y}_c + \log \left( \sum_j \exp(\hat{y}_j) \right) \right] \right\}, \tag{2}
\]

\[
L = \frac{\sum_c L(\hat{y}, y, c)}{\sum_c w(c)} \tag{3}
\]

where \( w(c) \) represents weights given to the \( c^{th} \) class (here, classes = \{0, 1\} = \{nSFR, SFR\}), the network output \( \hat{y}_j \) and true label \( y_j \).

A uniformly sampled dataset of the simulations would have \( \ll 0.001\% \) of the volume classified as SFR. We deal with this extreme bias via multiple methods: a) sampling is restricted to volumes with \( \langle \rho \rangle/\bar{\rho} > 1.0 \), so the region has some marginal overdensity on average, b) instead of sampling every possible \( 10^3 \) kpc/cm\(^3\) region, nSFR are taken at random positions throughout the volume, in a fixed ratio to extracted SFR regions, as described in section 2.3, c) we use weights in the loss function with \( w = [1, 4] \) for \{nSFR, SFR\}. Several combinations of weights were testing in hyperparameter tuning: \( w = [1, < 3] \) results in predictions collapsing to nSFR, while \( w = [1, > 6] \) generates an unacceptable number of false positives. We additionally employ \( L_2 \) regularization to guard against overfitting with \( L_2 \) parameter \( \lambda = 10^{-4} \). We use a plateau method to reduce the learning rate: the rate is reduced by half if the loss on the validation dataset has not reduced for 10 epochs until a minimum learning rate of \( 10^{-8} \). We also employ a checkpointing method, saving the model each time a new record low loss is achieved on validation data. The final model used in the following sections is the check-point using these best-loss weights.

We choose which hydrodynamic fields to train on based on our knowledge of star formation criteria in Enzo. Each volume has 5 channels as input: baryon density \( \rho_b \), \( \text{H}_2 \) density \( \rho_{\text{H}_2} \), gas velocity divergence \( \nabla \cdot \nu \), total metallicity \( Z_{\text{sum}} \), and total (kinetic + thermal) energy \( E_T \). Our selection of fields can also be based on physical intuition: Pop III star formation only in regions with high \( \rho_B, \rho_{\text{H}_2} \), and very low \( Z_{\text{sum}} \). \( E_T \) serves as a dual probe: strong radiation fields will increase temperature, thereby increasing thermal energy, while fast-moving gas would increase the kinetic energy. Star formation is not expected in either fast moving or hot gas, so having a high value of \( E_T \) should disqualify the area for Pop III star formation. The effect of \( Z_{\text{sum}} \) is essentially binary: \( Z_{\text{sum}} \leq Z_{\text{crit}} \) should enable Pop III star formation, but any other case should immediately disqualify the region. The \( \nabla \cdot \nu \) field should immediately disqualify star formation if the gas is not converging.

Although a model with fewer fields may appear effective, losing any of these probes into the hydrodynamical state would be expected to not be as robust across all redshifts. For example, if \( \rho_{\text{H}_2} \) is ignored, the model may start to fail after the LW background becomes strong enough to dissociate \( \text{H}_2 \) in relatively dense gas. If \( E_T \) were removed, the model would have to learn to infer the energy of the gas from \( \nabla \cdot \nu \), while losing all probes into

### Table 2. Tested classifier architectures for S1.

| Model   | Key Block | \( N_0 \) | \( N_{\text{ip}} \) [M] | \( N_b \) | \( E_{BL} \) |
|---------|-----------|----------|-----------------|--------|---------|
| SINet   | IB        | 16       | 0.41            | 480    | 160     |
| SDNet   | Dense     | 32       | 14.67           | 460    | 156     |
| SRNet   | Residual  | 32       | 21.29           | 460    | 193     |

Note—All models were trained on input data with dimension dim = 64\(^3\). \( N_{\text{ip}} \) is the number of trainable parameters in the model, and \( N_b \) is the batch size. The Key Block describes the architecture used in place of KB1 and KB2 in Figure 2. The epoch of the checkpoint with the best validation loss, which is used for evaluation in this work, is recorded in \( E_{BL} \).
the temperature, except that \( \rho H_2 \) and \( \rho \) would likely be lower at higher temperature.

3.2. Stage 2: Segmentation Network

After classification, we use deep neural network architectures designed for pixel segmentation, which we have adapted to 3 dimensional voxel segmentation, to identify star-forming voxels (SFVs) in stage 2 (S2). Two architectures were tested here, U-Net (Ronneberger et al. 2015) and a variation of U-Net inspired by Zhang et al. (2020) that uses IBs instead of standard convolutions (IUNet). In early testing, U-net was plagued by false positives, and was abandoned in favor of the IUNet architecture presented in Figure 4. As outlined in section 2.3, each star in a SFR is labelled by a 3\(^3\) cube centered on the particles host cell; a star particle is represented by 27 cells in a 64\(^3\) box, so the data is extremely biased. To deal with this bias, we use class weights in the cross entropy loss function, with \( w = [1, 9] \) for classes {SFV, non-SFV(nSFV)}. Several other weight combinations were tested: with \( w = [1, > 12] \), the predictions suffered from large swaths of SFV predictions, while \( w = [1, < 5] \), the output predictions collapse to nSFV for nearly all voxels and regions. As with the classifier networks, we use the Adam optimizer with \( L_2 \) regularization using \( \lambda = 10^{-5} \). We use an initial learning rate of \( 5 \times 10^{-3} \) with the same learning rate plateau adaptation method as S1.

The input is the 64\(^3\) region as prepared in section 3.1 with the same input fields as S1. The encoding branch reduces dimensionality via max pooling while increasing the number of channels via concatenation and convolution operations until the lower bottleneck, where the process is reversed in the decoding branch. The encode and decode layers are connected by concatenating the encode output to the decode input at a given layer. Finally, the output is reduced to two classes, representing nSFV and SFV, for each voxel in the region.

The skips in IUNet allow more efficient backpropagation and allow information to flow directly from the encoding branch to decoding branch, potentially reducing the size of the training set required to attain a robust and generalizable model (Ronneberger et al. 2015). The skip connections may also have a detrimental effect here though, as star formation is highly correlated to peaks in the density field, so the entire network may be skipped with predictions being made directly from the input fields. As seen in Zhang et al. (2019), to reduce the direct communication from the input fields to the final prediction, we removed the top most skip connection to force processing at deeper layers of the network.

3.2. Stage 2: Segmentation Network

After classification, we use deep neural network architectures designed for pixel segmentation, which we have adapted to 3 dimensional voxel segmentation, to identify star-forming voxels (SFVs) in stage 2 (S2). Two architectures were tested here, U-Net (Ronneberger et al. 2015) and a variation of U-Net inspired by Zhang et al. (2020) that uses IBs instead of standard convolutions (IUNet). In early testing, U-net was plagued by false positives, and was abandoned in favor of the IUNet architecture presented in Figure 4. As outlined in section 2.3, each star in a SFR is labelled by a 3\(^3\) cube centered on the particles host cell; a star particle is represented by 27 cells in a 64\(^3\) box, so the data is extremely biased. To deal with this bias, we use class weights in the cross entropy loss function, with \( w = [1, 9] \) for classes {SFV, non-SFV(nSFV)}. Several other weight combinations were tested: with \( w = [1, > 12] \), the predictions suffered from large swaths of SFV predictions, while \( w = [1, < 5] \), the output predictions collapse to nSFV for nearly all voxels and regions. As with the classifier networks, we use the Adam optimizer with \( L_2 \) regularization using \( \lambda = 10^{-5} \). We use an initial learning rate of \( 5 \times 10^{-3} \) with the same learning rate plateau adaptation method as S1.

The input is the 64\(^3\) region as prepared in section 3.1 with the same input fields as S1. The encoding branch reduces dimensionality via max pooling while increasing the number of channels via concatenation and convolution operations until the lower bottleneck, where the process is reversed in the decoding branch. The encode and decode layers are connected by concatenating the encode output to the decode input at a given layer. Finally, the output is reduced to two classes, representing nSFV and SFV, for each voxel in the region.

The skips in IUNet allow more efficient backpropagation and allow information to flow directly from the encoding branch to decoding branch, potentially reducing the size of the training set required to attain a robust and generalizable model (Ronneberger et al. 2015). The skip connections may also have a detrimental effect here though, as star formation is highly correlated to peaks in the density field, so the entire network may be skipped with predictions being made directly from the input fields. As seen in Zhang et al. (2019), to reduce the direct communication from the input fields to the final prediction, we removed the top most skip connection to force processing at deeper layers of the network.

### Table 3. Results of S1 on testing data from PHX256-2.

| Model   | Accuracy     | \( P \)     | \( R \)     | \( F_1 \)      |
|---------|--------------|-------------|-------------|---------------|
| SINet   | 27201/27231 (0.9989) | 0.9932     | 0.9998     | 0.9960        |
| SDNet   | 27201/27231 (0.9989) | 0.9927     | 0.9999     | 0.9961        |
| SRNet   | 27198/27231 (0.9988) | 0.9921     | 0.9999     | 0.9958        |
| IUNet   | 27200/27231 (0.9989) | 0.9932     | 0.9998     | 0.9960        |

Note—We also present S2 using IUNet as if it were used to classify SFR. Every classifier tested is able to achieve high accuracy \( \geq 99.8\% \), reinforced by very high precision \( P \), recall \( R \), and \( F_1 \) score (metrics as defined in Section 4.1).

The IB architecture has several different sizes of convolutional filters \((K = 1^3, 3^3, 5^3)\), which can give sensitivity to different scales of features in the data, e.g., 100 pc scale infalling gas toward a density peak 1-10 pc in radius. IBs have the design to be sensitive to both of those scales of features that will be important in predicting star formation.

Although S2 minimizes cross-entropy loss for training, we also calculate the intersection over union (IoU) metric judge the quality of the S2 voxel-wise prediction. If \( P_t \) is the set of positive predicted voxels, and \( P_T \) is the set of ground truth (GT) positive voxels, the IoU is given by

\[
IoU = 1 - \frac{P_t \cap P_T}{P_t \cup P_T}.
\]

With this representation, \( IoU = 0 \) is a perfect prediction with no false negatives or false positives.

4. RESULTS

4.1. S1 Volume Classifier

We use three measures to quantify the success of an S1 model: (a) Raw accuracy as simply \( A = N_{correct}/N_{total} \), (b) precision, \( P = P_T/(P_T + P_F) \), with true positives \( P_T \) and false positives \( P_F \), and (c) recall, \( R = N_T/(N_T + N_F) \) with true negatives \( N_T \) and false negatives \( N_F \). We will also extend the \( P \) and \( R \) metrics with the \( F_1 \) score, given by \( F_1 = P_T/(P_T + 0.5(P_T + N_F)) = 2 \times P \times R/(P + R) \) to provide another measure of classification ability. With these definitions, perfect result would have \( P = R = F_1 = A = 1 \). Figure 5 presents the \( P \) and \( R \) measures along with the loss for training and validation of all tested S1 architectures. SDNet and SRNet, despite having similar memory requirements as SINet in training, have significantly more trainable parameters. Their increased complexity appears to affect how quickly they converge to a trained state, as they seem to have similar accuracy to SINet at much earlier epochs. This effect is particularly noticeable in the recall. On testing data in Table 3, we find that all three S1 models perform extremely well, with all three having \( A \geq 0.9988 \).
All three additionally have very similar precision and recall, as measured on their volume-wise classification. In \( F_1 \) score, SDNet performs best (\( F_1 = 0.9962 \)), followed closely by SINet (\( F_1 = 0.9952 \)) and SRNet (\( F_1 = 0.9951 \)). We do not measure the inference rate of S1 as it does not produce a desirable prediction in this work; it needs to be coupled with S2 to localize star formation to a precise region.

### 4.2. S2 Voxel Segmentation

Loss, \( (\text{IoU}) \), and volumetric accuracy \( (\text{Acc}) \) for training and validation for S2 are presented in Figure 6. S2 is a capable volume classification model, with \( A > 0.994 \) in both training and validation. After training, S2 was also applied independently to the test dataset (Table 3), where S2 classified regions with \( A = 0.9989 \) with \( P = 0.9932 \), \( R = 0.9998 \), and \( F_1 = 0.9960 \), indicating that S2 is as capable as the S1 models at volumetric classification for SFRs. Despite this high accuracy in classifying regions, S2 struggles to match individual star-forming voxels, with \( (\text{IoU}) = 0.776 \). Indeed, if the voxel-wise accuracy is quantified by averaged precision and recall, we find \( (P) = 0.2589 \) and \( (R) = 0.9999 \) yielding \( (F_1) = 0.3996 \). These combined measures show a significant propensity to false positive SFV. This is reinforced by simple statistics; there are an average of 36.34 (\( \sim 3.3^3 \) voxels) true SFV in each SFR, with \( \sigma = 14.78 \) while predictions average 89.86 (\( \sim 4.4^3 \)) SFV per SFR with \( \sigma = 45.9 \).

The predicted region still covers the GT voxels, as indicated by the high value of the voxel-wise recall. Aside from recall, this behaviour can be quantified in a spatial sense: if we define a star-forming center as the average location of SFV, the euclidean distance from the prediction to GT center is another measure of accuracy. Here, we measure the euclidean distance in voxel widths, and find that the average distance between SFV center and GT center is 1.90 voxel-widths. Further, the standard deviation of the distribution of distances is 2.368 voxel-widths, while the largest observed distance is 23.72 voxel-widths. In examining the distribution of distances, we find that 98.1% of SFV centers are identified within 10 voxels of the GT central point, and that 0.796% of centers are more than 15 voxel-widths from the ground-

\footnote{\( \sigma \) here represents the standard deviation of the distribution}
Figure 5. Training (a) and validation (b) results for the training of small classifier architectures. We further quantify the accuracy with the precision (P) and recall (R) measures, as defined in section 4.1. All three models show continually decreasing loss (L) in training, although this metric is fairly constant in validation. Despite the bias in our samples, the errors in S1 are dominated by false positives indicated by the relatively lower P across all architectures. N/1000 represents one output every 10 iterations within each epoch—validation has fewer data points because there are fewer validation iterations in each epoch.

Table 4. Results of different configurations of StarFind modules on testing data from PHX256-2.

| S1   | A(volumetric) | (IoU) | P   | R   | F1  | Rproc |
|------|---------------|-------|-----|-----|-----|-------|
| SINet | 27195/27231 (0.9986) | 0.8191 | 0.2590 | 0.9999(8) | 0.3997 | 465.75 |
| SDNet | 27199/27231 (0.9988) | 0.8196 | 0.2584 | 0.9999(8) | 0.3991 | 484.32 |
| SRNet | 27188/27231 (0.9984) | 0.8195 | 0.2592 | 0.9999(8) | 0.3999 | 444.04 |

Note—Each row represents the StarFind module using a different architecture for S1. The accuracy (A) refers to volumetric classification accuracy, while the averaged IoU (⟨IoU⟩), P, R, and F1 pertain to voxel-wise predictions within the regions. Rproc is the rate of inference with batch size 1 processing on a single Nvidia RTX 2080 GPU. When using S2 as the sole classifier has Rproc = 58.65; using S1 as a filter increases Rproc by up to 8×.

Predicting Primordial Stars

The single largest shortcoming of using S2 for all processing is in its inherently slower computation time. Processing every volume through S2 can only be done at a rate Rproc of 54.67 volumes/second, when only timing the inference of the model. Given that > 90% of volumes contain no stars, a quick, simple model that can discard obvious nSFR would greatly expedite the inference of the final model. For this reason, we chain together S1 and S2 to form the StarFind module.

4.3. StarFind Module

To process a sample through the StarFind module, we filter the sample volumes using S1, only passing those where S1 predicts SFR on to S2. The two models are designed to be independent to test the efficacy of different combinations of architectures. The results of this test are tabulated in Table 4. S2 has high region classification accuracy overshadowed by slow processing; implementing an S1 model to filter regions increases the rate of processing samples (Rproc), by up to 8×, with no significant change in volumetric classification accuracy (|A_{S2} − A_{StarFind}| < 0.02%). Also of note is that the actual architecture of S1 is largely irrelevant given the models developed here; SINet, SDNet, and SRNet all perform very similarly. However, the best performing model uses SDNet: the combination of SDNet + IUNet has (marginally) the highest accuracy on testing data, with (P) and (R) leading to the best (F1) score, as well as the fastest Rproc. Unfortunately, the inclusion of S1 was unable to significantly improve the voxel-wise predictions (e.g., ⟨IoU⟩) of S2 by prefiltering nSFR—this implies that there are very few false positive regions being identified by S2 that are not also classified as SFR by S1.

4.4. Generalizability

In all training, testing and validation data, the data was prepared identically, from simulations that have the required resolution and physical models for Pop III star formation. As a generalization test, StarFind was
Figure 6. Accuracy (top, $Acc$), averaged $IoU$ (middle, $⟨IoU⟩$), and cross entropy loss (lower, $L$) from training and validation of IUNet. $N$ represents intra-epoch recordings, not end of epoch data, and validation data has been expanded to align with training. These data reflect epochs of training up to the best recorded validation loss.

Table 5. Found volumes using the StarFind module as designed for simulations.

| Module   | Dataset    | $z$  | $N_f$ |
|----------|------------|------|-------|
| SINet+S2 | PHX256-2   | 21.18| 17    |
| SINet+S2 | PHX-HYD    | 21.18| 19    |
| SINet+S2 | P3N-128    | 19.0 | 1     |

Note—We annotate the redshift ($z$) of the output, as well as the number of volumes ($N_f$) found at that redshift. Note that P3N-128 only has SFV found at a later $z$: this is due to resolution effects of the actual simulation, not the StarFind module.

Figure 7. Comparing the density field of simulations with the same ICs. 7a shows the PHX256-2 simulation at $z = 21.19$ with Pop III star forming regions annotated. P3N-128 is shown at a similar redshift in 7b; the density peaks in P3N-128 are both less extreme (note the difference in color scaling) and more diffuse due to lower mass and force resolution.

AMR refinement levels so that $dx_{min} = 156$ pccm. P3N-128 has no Pop III star formation, and has mass resolution $\sim 1/8$ that of the PHX series. Ten DM particles in P3N-128 have more mass than a halo expected to form Pop III stars: this simulation will test how StarFind performs in simulations that cannot resolve these first star forming halos.

For this test, we select a data output, and iterate through all AMR grids in the simulation hierarchy. Any grids less than level 3 are automatically skipped, as they cannot qualify for star formation. Grids at a deep enough AMR level are tiled in $10^3$ kpc$^3$ volumes and each volume with $(\rho_B)/\bar{\rho} > 2$ is passed through the StarFind module. If $S1$ returns nSFV, the rest of the
Figure 8. SFRs identified in PHX256-HYD at progressing redshifts. The middle panel is at the same redshift as PHX256-2 presented in Figure 7a. There are more SFR present, e.g., at $z = 21.18$, than in PHX256-2, however we have verified that the missing SFRs are present in PHX256-2 by $z = 20.86$. StarFind is identifying SFR in PHX256-HYD at reasonable locations reflected in PHX256-2, and all of the regions identified by StarFind are mirrored by or precede star formation in PHX256-2.

Example projections of the predictions from our generalization tests are shown in Figure 9. Shown are fields that correspond more easily to physical intuition than the fields which generated the predictions; we show number density $n_b$ in place of baryon density, temperature, $T$, in place of energy, and $H_2$ neutral fraction ($\rho_{H_2}/\rho_b$) in place of neutral $H_2$ density. Figure 9a shows an ideal prediction; the predicted region is compact and well defined, and appears to agree with peaks in both $n_b$ and $\rho_{H_2}/\rho_b$, which are both to be expected from the star formation algorithms in Enzo. Figure 9b shows the prediction of two star forming regions in one volume. While examples like this are possible, they become more common as the simulation proceeds without stellar feedback and enriched star formation to begin disqualifying regions for Pop III star formation. The dual predictions still coincide with sensible values of the projected fields. Figure 9c shows the first prediction of SFR in P3N-128, in a neighboring region near 9a at $z = 19.0$. The relative

computation is skipped. If $S1$ returns SFR, then the volume is passed to $S2$, which will return a SFV classification for each of the $64^3$ voxels in the volume.

In the simulations without SFF, those regions that do collapse to be identified as SFR will continue to collapse: there is no star formation to provide a sink for the gas, nor feedback to disperse or photoionize it. If we identify all star forming regions at, e.g., $z = 21.18$, then this will represent all the star forming regions that have formed since the simulation start. Since the actual star formation rate (SF rate) will depend heavily on the stellar initial mass function (IMF), here we aim to match the number of star forming regions, with the assumption that a stellar IMF can then be applied to match the SF rate of higher-resolution simulations (i.e., PHX256).

By $z = 21.18$, PHX256-2 has formed 17 clusters\(^5\) of Pop III stars, with 226 individual star particles (see Figure 7a): this early star formation is strongly clustered, averaging $> 10$ star particles per 10 kpc cm cluster. With a baseline for the number of star-forming regions in hand from PHX256-2, we apply the module to PHX256-HYD. The results of applying StarFind are presented in Figure 8. At $z = 21.18$, StarFind identified 21 regions. If compared to Figure 7a, we immediately identify star regions in PHX256-HYD that are not yet present in PHX256-2. These regions however, have been verified to begin forming stars in PHX256-2 by $z = 20.86$. StarFind is identifying SFR in PHX256-HYD at reasonable locations reflected in PHX256-2, and all of the regions identified by StarFind are mirrored by or precede star formation in PHX256-2.

\(^5\) Here, a cluster is defined as any group of stars all within 10 kpc cm of the central star.
Figure 9. Test result examples from generalizability test. Fields presented are the projection of $\log_{10}$ of baryon number density ($n_b$), temperature ($T$), and $H_2$ neutral fraction ($\rho_{H_2}/\rho$); the “Predicted” panel shows voxels identified as SFV by S2 with the number of positive voxels annotated. (a) shows a nearly ideal result with tightly clustered cells predicting a star forming region at a peak of density and $H_2$ fraction. (b) shows the interesting possibility of finding more than one star forming region per volume. (c) shows identification of the first identified star forming volume (8 kpccm from that in 9a) in an under-resolved simulation. Since the dynamics of structure formation are less resolved, the star forming region is not found until a much later redshift.

Figure 10. Redshift evolution of identified SFR in P3N-128. Halos are identified by red circles (scaled by virial radius), while SFR identified at different redshifts are annotated in various colors. Most SFR are identified before their host halo is identified—by up to $\sim 37$ Myr for the first SFR predicted at $z = 19.0$ whose host halo is identified at $z = 16.77$. Lateness of this prediction is an artifact of structure formation dynamics in under-resolved simulations; as seen in Figure 7b, at $z = 21$, the density field in P3N-128 is amorphous, showing no sharp, distinct features as seen in the higher resolution simulations. We will expand on the predictions in P3N-128 in Section 5.

In applying the StarFind module to P3N-128, the module identifies no star forming regions at $z = 21.18$. To analyze how star formation as identified by StarFind may proceed, we iterate through all available outputs of P3N-128 to find both the first SFR, and subsequent SFR. The results of this test are presented in Figure 10. The first SFR is identified at $z = 19$, with more found at each subsequent output. Figure 10 does not include every SFR—those that are very close together are only plotted once to aid readability: at $z = 16.77$, there are 17 distinct SFR identified in P3N-128. This raw number of SFR and their locations in the projection agrees well with those in PHX256-2 at $z = 21.19$, as seen in Figure 7a.

5. DISCUSSION

We have designed a classification-segmentation algorithm using deep ANN that is capable of predicting Pop III star formation sites in Enzo hydrodynamical simulations. We have found that notable image recognition architectures (adapted to 3D) are effective at this task, as are common pixel segmentation architectures. In fact, common image classification models are far too complex for this task, as there was no appreciable gain in accuracy from using state-of-the-art GoogLeNet, ResNet, or DenseNet over the small classifier architec-
tures. The StarFind module developed here can identify $10^3$ kpc m$^3$ star forming volumes with > 99.7% accuracy, and the prediction method has been applied with reasonable results to hydrodynamic simulations that have no star formation routines enabled.

We have also noted that while IUNet is a very good volumetric classifier, it fails to have high accuracy in voxel-wise predictions. It over predicts the size of localized regions in SFR, which is reinforced by the high $\langle IoU \rangle$ and simple statistical counting of SFV. This may be an unavoidable issue that must be addressed in post-processing; most applications for image segmentation have clear features, i.e., sharp boundaries between organs in organ segmentation, or the outline of a car in a street image. Here, star-forming regions have no clear delineation from their neighboring regions, instead being separated by a gradual increase in density and related variables. Although $P = 1$ may be out of reach, $S2$ may be improved through testing different loss functions or a combination of loss functions (Asgari Taghanaki et al. 2018; Hajiabadi et al. 2020). Improving the $\langle IoU \rangle$ and voxel-wise $\langle P \rangle$ and $\langle R \rangle$ will be a priority in developing $S2$ into a production model.

Despite this, the star forming regions appear well defined (e.g., Figure 9). $S2$ predicts the region of formation reliably, but predicts a larger region than exists in the ground truth. The predicted region still covers the ground truth SFV, as indicated by the high value of the voxel-wise recall.

5.1. Further discussions of generalizability

The StarFind module has one major goal: to identify Pop III star-forming regions in simulations without the necessary resolution for star formation. The lack of resolution could present itself in a spatial sense, as in PHX256-HYD, or in a mass sense, as in P3N-128. Figure 7 shows a comparison between PHX256-2 and P3N-128. PHX256-2 is shown at $z = 21.18$; P3N-128 is shown at varying redshift, annotated in the legend. Note that halo formation begins at $z \sim 17.17$, and low-mass, Pop III forming halos ($M_{\text{vir}} \sim 10^{6.5} M_\odot$) are never formed in P3N-128 due to its low mass resolution.

Despite this lack of resolution, as presented in Figure 7, PHX256-2 (purple) is shown at $z = 21.18$; P3N-128 is shown at varying redshift, annotated in the legend. Note that halo formation begins at $z \sim 17.17$, and low-mass, Pop III forming halos ($M_{\text{vir}} \sim 10^{6.5} M_\odot$) are never formed in P3N-128 due to its low mass resolution.

Despite this lack of resolution, as presented in Figure 10, StarFind predicts stars that will be in halos that form 10-30 Myr after the SFR identification. It is widely accepted that the mass (particle mass or gas mass in a grid cell) is a determining factor in the effectiveness of stellar feedback (Hopkins et al. 2019). To determine the potential difference of feedback applied at $z = 19$ or $z = 16.77$, we analyzed the mass distribution of the SFR ($M_{\text{SFR}}$) as compared to the later halo ($M_{\text{halo}}$). We found that $M_{\text{SFR}} < 0.5 M_{\text{halo}}$, when using the virial radius ($R_{\text{vir}}$) of the halo at $z = 16.77$ to define a volume at both redshifts, but even more importantly, the gas is less compact in the central regions e.g., at $R = R_{\text{vir}}/7$, $M_{\text{SFR}} < 0.2 M_{\text{halo}}$. When we extend this analysis to all SFR identified by $z = 17.59$, comparing each SFR to the later halo it is nearest to, we find that the SFR has overdensity $\langle \rho \rangle / \bar{\rho}_{\text{SFR}} \lesssim 0.5 \langle \rho \rangle / \bar{\rho}_{\text{halo}}$. If halos are too massive when they start forming Pop III stars, as the particle mass is simply too high and those halos are unresolved.
III stars, the feedback will be confined and unable to pollute the local environment (Whalen et al. 2008), so applying feedback at the less dense states identified by StarFind may reduce the effect of the massive halos found in P3N-128. Without a developed feedback mechanism, the StarFind module presents an improved method to identify Pop III SFR compared to methods that rely on halo finding and using halo virial mass relationships to determine the halo occupation of Pop III stars.

One of the primary concerns when applying a deep learning model to a production application is to understand how the model fails. If the failure modes are predictable, then they can be accounted for in the deployment of the model, but if they are seemingly random, the model may not even be usable. Here, we are most concerned with failures to classify in a volumetric sense; identifying “nearby” star formation is acceptable, but misclassifying an entire region is a much more egregious failure. To analyse these critical failures of the StarFind module, we passed every training/test/validation sample through the model, and plotted a) the SFV labels if S1 was incorrect, or b) the SFV predictions and labels if S2 incorrectly classified the volume. 65% of the false positives in S1 were cases where the star voxels were within 3 voxel-widths of the border of the volume. All false positive cases of S2 were on the edge of the volume, and 75% of the false negatives had star forming labels on the border. The star-forming border is a major avenue of failure for the module, but also an easy one to remedy. In a production pipeline, two obvious possibilities could alleviate these failures: a) ignore the SFR if the flagged voxels are on the edge, as the routine will be called again in < 1 Myr, at which point the SFV will have likely moved to a more central location, or b) recenter the volume on the suspicious SFV and re-run the module to receive a more reliable result. The results in this work used the second option to reduce false positives and negatives in analysis of the PHX256-HYD and P3N-128 simulations. Future work will attempt to robustly quantify these failures, and generate a method to identify them while in a production scenario.

This method will ultimately be used in a novel sub-grid feedback method, where SFR identified using this algorithm will then be evolved to a post-star state using another series of deep neural networks. The goal throughout this work was to design the model that could be used in a production capacity (further training notwithstanding); to that end, here we estimate the runtime implications of using this module to perform Pop III SFF.

Evolving PHX256-2 from $z=21.19$ to $z=17.53$ consumed 7K cpu-hours. Most of this time is spent evolving hot supernova remnants, and their associated hydrodynamic and stellar radiation fields in the deep AMR levels of the volume. PHX256-HYD evolves the same region in 14 cpu-hours by excluding Pop III star formation and restricting the maximum AMR level. Applying the StarFind module to locate star-forming regions once (i.e., one panel in Figure 8) consumes ~4 CPU-hours, processing 5.2 volumes/sec on average. To localize Pop III star forming regions every Myr would require performing this evaluation ~ 30 times, consuming a total of 120 CPU-hours. Most of the StarFind processing time is spent iterating through grids and disqualifying regions before they even enter S1: only $\lesssim 5\%$ of regions are classified by S1, with 3% continuing to S2, where only 2% finally qualify as SFR (0.003% of candidate volumes), so we estimate that adding a feedback routine after S2 would not significantly increase the processing time of the algorithm as a whole. This method, when finalized with a feedback algorithm, may produce up to 50× speedup as compared to explicitly star-forming simulations, with more speedup likely from optimizing the final method.

This rate makes this method feasible to use inline with an Enzo simulation using inline python with YT, particularly since the tests run here do not include optimization of any kind. We could expect further speedup by incorporating the method into Enzo’s source C/C++ code and/or optimizing StarFind using MPI-parallelism on CPU architecture (e.g., Mathuriya et al. 2018). Future work will focus on evolving Pop III remnants in regions identified using this method, and incorporating the framework into an Enzo simulation, along with further training of the models presented here.

This research was supported by National Science Foundation grant CDS&E grant AST-1615848 to M.L.N. The simulations were performed using Enzo on the Frontera supercomputer operated by the Texas Advanced Computing Center (TACC) with LRAC allocation AST-20007. Data analysis and model development, training, and testing were performed on the Comet supercomputer operated for XSEDE by the San Diego Supercomputer Center under XRAC allocation TG-AST200019.

REFERENCES

Ade, P. A. R., Aghanim, N., Armitage-Caplan, C., et al. 2014, Astronomy & Astrophysics, 571, A16.
http://dx.doi.org/10.1051/0004-6361/201321591

Aniyan, A. K., & Thorat, K. 2017, The Astrophysical Journal Supplement Series, 230, 20.
https://doi.org/10.3847%2F1538-4365%2Faa7333
