Probabilistic cosmic web classification using fast-generated training data

Brandon Buncher1⋆ and Matias Carrasco Kind2,3
1Department of Physics, University of Illinois, Champaign, IL 61820 USA
2Department of Astronomy, University of Illinois, Urbana, IL 61801 USA
3National Center for Supercomputing Applications, Urbana, IL 61801 USA

11 December 2019

ABSTRACT
We present a novel method of robust probabilistic cosmic web particle classification in three dimensions using a supervised machine learning algorithm. Training data was generated using a simplified $\Lambda$CDM toy model with pre-determined algorithms for generating halos, filaments, and voids. While this model lacks physical detail, it can be generated substantially more quickly than an N-body simulation without loss in classification accuracy. For each particle in this dataset, measurements were taken of the local density field and directionality. These measurements were used to train a random forest algorithm with, which was used to assign class probabilities to each particle in a $\Lambda$CDM, dark matter-only N-body simulation with $256^3$ particles, as well as on another toy model data set. By comparing the trends in the ROC curves and other statistical metrics of predictions made on each of the datasets using different feature sets, we demonstrate that the combination of measurements of the local density field magnitude and directionality enables accurate and consistent classification of halo, filament, and void particles in varied environments. We also show that this combination of training features ensures that the construction of our toy model does not affect classification. The use of a fully supervised algorithm allows greater control over the information deemed important for classification, preventing issues arising from hyperparameters and mode collapse in deep learning models. Due to the speed of training data generation, our method is highly scalable, making it particularly suited for classifying large datasets, including observed data.

Key words: cosmology: large-scale structure of the Universe – dark matter – galaxies: fundamental parameters – halos – methods: data analysis – statistical

1 INTRODUCTION
Large-scale structure (LSS) describes the largest scale inhomogeneities in the universe. LSS is comprised of clusters, filaments, and voids. Clusters are small, compact groups of tens to tens of thousands of particles with radii on the scale of Mpc or tens of Mpc. Typically, galaxies form around dark matter halos, large dark matter overdensities with masses on the scale of $10^{11} - 10^{15} M_\odot$ (White & Rees 1978). Galaxy filaments are long, strand-like visible/dark matter overdensities that connect clusters, with lengths between 50 and 80 Mpc $h^{-1}$ (Bharadwaj et al. 2004). While filament properties are not well understood, recent data presented by Pereyra et al. (2019) has provided constraints on a variety of properties, including the density profile and total mass. Voids are visible/dark matter underdensities that fill the space between filaments, with a typical density of around $2 \times 10^{-2} \text{ Mpc}^{-3} h^3$ (Hamaus et al. 2014). Voids are roughly triaxial ellipsoidal, with volumes on the order of $10^4 - 10^5 \text{ Mpc}^{-3} h^3$ (Foster & Nelson 2009).

The halo (filament) mass fraction refers to the ratio of the mass of halo (filament) particles relative to the total mass of particles; for the purposes of this paper. The equivalent metric for void particles, the fraction of particles in underdense regions, will hereafter be referred to as the void mass fraction. The relative mass fractions of clusters, filaments, and voids have been studied extensively; however, there exists substantial disagreements on their values depending on the methodology used. Using $\Lambda$CDM N-body simulations cluster mass fraction has been estimated to be between 9 - 41% (Forero-Romero et al. 2009; Shandarin et al. 2012); the filament mass fraction has been found to be between 18 - 50% (Shandarin et al. 2012; Cautun et al. 2014); and the void mass fraction has been estimated to be between 13 - 27% (Hoffman et al. 2012a; Aragon-Calvo et al. 2010a). In this work, all particles have the same mass, so the mass fraction is equivalent to the fraction of particles that are a member of a given class.

For the purposes of this work, we model the underlying dark matter distribution for halos, filaments, and voids, which provides an excellent representation of galaxies; hence, particles here refer to dark matter particles.
The formation and evolution of galaxies is controlled by a variety of local properties, including the local density of dark and visible matter (Mo et al. 2010). The beginning of galaxy formation is primarily guided by the mass and density profile of the proto-galaxy’s dark matter halo (Green & van den Bosch 2019). The collapse of an overdense region of dark and baryonic matter leaves a dark matter halo, a triaxial ellipsoidal dark matter overdensity. The baryonic matter that remains in the gravitational well of the dark matter halo cools, it begins to collapse into star-forming regions. While star formation depends weakly on the local density (Mo et al. 2010), the dynamics of the halo’s gravitational collapse, which dictates the proto-galaxy’s size and density profile, is governed primarily by this property. The local density also governs the abundance of dark matter halos and, hence, interactions between local galaxies, such as tidal stripping of matter (Mo et al. 2010; Green & van den Bosch 2019). As a result, it is crucial to understand the local dark matter density to understand how galaxies form and evolve.

A galaxy’s LSS class provides substantial information about the local density. Thus, it is important to create an efficient, reliable method for determining an individual galaxy’s morphological LSS class to understand problems such as these. However, a universal, deterministic algorithm is too complex to construct explicitly (Mo et al. 2010). Various classification algorithms, some of which are non-deterministic and/or non-universal, have been created; several of these are summarized in Table 1 (Libeskind et al. 2018), which we discuss in greater detail below.

Classifiers that do not use machine learning (ML) typically exploit physical or geometric properties of the structures they attempt to classify. These may be further divided into those that classify individual particles and those that determine the location and extent of individual structures. To classify particles, cluster finding algorithms frequently utilize connectedness among particles (Davis et al. 1985; Alpaslan et al. 2013) and/or on local geometric information such as density (Kitaura & Angulo 2012). Filament finding algorithms, however, must include information on the local and global density field, as well as some additional information that differentiates them from halos. Filament finding methods are typically geometrical (Tempel et al. 2016; Aragon-Calvo et al. 2004) or topological in nature (Aragon-Calvo et al. 2010b; Sousbie 2011), though some graph-based methods exist (Alpaslan et al. 2013). Geometric algorithms (Cautun et al. 2012; Kitaura & Angulo 2012) have been used to classify both halos and filaments. However, of the methods we discuss here, only Alpaslan et al. (2013) and Falck et al. (2012) assign classes directly to individual particles, and none assign class probabilities. This substantially hinders the effectiveness of classification: due to difficulties in predicting fundamental properties of LSS analytically, the extent of these structures is highly dependent on arbitrary parameters. For example, many algorithms differentiate between structures using arbitrary density/scale cutoffs, either implicitly or explicitly (Libeskind et al. 2018; Tsizh et al. 2019). Differences between these cut-off values lead to substantial inconsistencies between the predictions of these algorithms, such as structural mass/volume fractions and the halo mass function (HMF); a discussion of these differences can be found at Cautun et al. (2012).

Some machine learning-based methods use deep convolutional neural networks; an example of this method may be found in (Aragon-Calvo 2018). Alternative techniques utilize supervised learning from a variety of time snapshots over the evolution of an N-body simulation (Lucie-Smith et al. 2018; Hui et al. 2018). While ML-based methods are generally more efficient than statistical classifiers, they do have substantial drawbacks. Due to a lack of understanding of the internal classification methodology of deep learning algorithms, these algorithms provide little information on the hallmark features of structural classes. This also may lead to classification being performed using features that do not represent the actual structures. In addition, deep learning methods are generally highly sensitive to initial hyperparameters, further inhibiting our understanding of why a particular class was chosen for a particular region and potentially introducing arbitrarily-selected biases. This sensitivity may inhibit the generation of a widely-applicable algorithm with reproducible results, as small changes in the test data set may require alteration of these hyperparameters. Methods to avoid these biases have been implemented in other scenarios, such as in the fast generation of cosmic web images (Rodriguez et al. 2018); however, these methods require knowledge of the expected output, for which there is little consensus in the context of LSS classification. While supervised techniques allow greater understanding of the features utilized to determine a particle or region’s class, known algorithms require information from a variety of time snapshots, increasing computational expense. These methods require training data extracted from multiple N-body simulation snapshots, which are computationally expensive. In addition, these methods are primarily designed to understand the time-evolution of halo mass distributions, which is not the goal of this project. While it may be possible to adapt these methods to classify halo particles, it would still require multiple time snapshots.

A recent topological cosmic web classifier was presented by Tsizh et al. (2019), in which the authors classified LSS particles by treating the cosmic web as a complex network. Halos were found using a friends-of-friends (FOF) algorithm and used as nodes when constructing the network, and various metrics based on particle position and velocity were used in classification. Unfortunately, classification was not successful, as demonstrated by an average confusion matrix score of 70%. A major contribution to the poor performance of this method stems from the difficulty in classifying halo particles found in large voids as this method, along with many of the topological models described in Table 1 perform classification using a relatively small range of length scales. For example, in Tsizh et al. (2019), the linking lengths used when constructing the network ranged from 1.6 - 2.4 Mpc h\(^{-1}\), which fails to cover the radii of even medium-sized voids (Foster & Nelson 2009). As is noted in (Libeskind et al. 2018), local density is highly scale-dependent, indicating that density magnitude alone is ineffective when classifying halo particles (Tsizh et al. 2019; Libeskind et al. 2018). As such, a robust cosmic web classifier must take into account information beyond the local density, and must also ensure that a strict density magnitude cutoff is implicitly used when distinguishing between structures.

To simplify these classification routines, in this paper, we present an efficient ML-based classification routine that does not fit any of the categories summarized in Table 1. Our algorithm requires substantially less information than others; in particular, we demonstrate that training using only information derived from particle positions in a single toy model generates enough information to classify a particles in a substantially larger N-body simulation. We generate training data using a toy model constructed from pre-determined structural creation algorithms which are distributed pseudo-randomly throughout a particle field. After performing measurements of the local density magnitude and density field directionality for each particle (each of which retains a “true” class inherited from its creation algorithm), we train a random forest ML algorithm to classify particles in an N-body simu-
Probabilistic trained cosmic web classification

We aim to classify individual LSS particles using a random forest (Breiman 2001) ML algorithm trained using a fast generated data. We developed a toy model that simulated a particle field comprised of halos, filaments, and voids. Measurements of the local, global, and isotropic densities and direction fields were taken for each particle and used to train the ML algorithm. The trained algorithm was used to make predictions of LSS class values for each particle in an N-body simulation we ran, hereafter referred to as “SIM”.

SIM is a ΛCDM model simulation consisting of \(N_{\text{tot}}\) collisionless dark matter particles with particle number density \(n_{\text{tot}}\), each of which has a mass of \(M_p\). The parameters for this simulation were taken from the WMAP+BAO+H\(_0\) results found in (Komatsu et al. 2011). The cosmological parameters used were \(\Omega_{\text{m},0} = 0.272, \Omega_{\Lambda,0} = 0.728\), and \(h = 0.704\), where the Hubble parameter \(H_0 = 100\ h\ \text{km s}^{-1}\ \text{Mpc}^{-1}\). Initial conditions were generated using second-order Lagrangian perturbation theory (2PT) instead of the standard Zeldovic approximation (see (Crocce et al. 2006) and (Scoccimarro 1998) for an explanation of this code). The primordial linear power spectrum was generated using CAMB. For this cosmology, the power spectrum was normalized using \(\sigma_8 = 0.810\) and spectral index \(n_s = 0.967\). As the simulation included only dark matter particles, we evolved them using the parallel tree N-body/smoothed particle hydrodynamics (SPH) code GADGET-2 (Springel 2005). Only the tree code was used for this simulation. The simulation started at redshift \(z = 50\) (corresponding with scale factor \(a = 0.0196\)) and evolved until the scale factor \(a\) reached 1.

For the purposes of this work, we used a snapshot of SIM at \(z = 0\).

2.1 Toy Model Simulation Generation

The method for generating a toy model dataset consisted of creation algorithms for halos, filaments, and voids. Note that the toy model only reproduces general structural features: rather than simulating the time evolution of matter due to gravity from the beginning of the universe, each structure is produced without regard to a physical creation process. While this toy model lacks the physical processes seen in N-body simulations, the generation process is substantially faster and more computationally efficient.

In the toy model, each particle’s mass is defined as \(M_p\), and the density of the universe \(\rho = M_p\ Mpc^{-3}\ h^3\), corresponding to a particle number density \(n_p = 1\ Mpc^{-3}\ h^3\).

A diagram of the toy model creation process can be found in Figure 1.

2.1.1 Halo Generation

The toy model simulation algorithm begins with halo generation. Halo masses were sampled from a halo mass function (Warren et al. 2006) with minimum (maximum) halo sizes \(N_{\text{min}}\) and \(N_{\text{max}}\) particles; this HMF model was chosen due to matching well with simulation data. It is expected that the halo radius would correspond with \(R_{200}\), the radius such that the halo’s density \(\rho_{200} = 200\); however, this caused the ML algorithm to determine a particle’s halo membership solely based on the local density. As filament densities are less understood than halo densities, filament classification would be highly dependent on the filament density range, which was created by hand. Thus, to vary halo densities, we found a probability density function (defined in Eqns. (1), (2)) for the

2 METHODS

Unless otherwise stated, all parameters found in this section are listed in Table 2.
Table 1. An overview of the methods compared in Libeskind et al. (2018); “all” indicates that the algorithm classifies particles as members of halos, filaments, voids, or sheets/walls.

| Method                                      | Web types | Input Type | Type                        | Main References                        |
|---------------------------------------------|-----------|------------|-----------------------------|----------------------------------------|
| Adapted Minimal Spanning Tree (MST)         | filaments | halos      | Graph & Percolation          | Alpaslan et al. (2013)                 |
| Bissous                                     | filaments | halos      | Stochastic                  | Tempel et al. (2016)                   |
| FINE                                        | filaments | halos      | Stochastic                  | González & Padilla (2010)              |
| Tidal Shear Tensor (T-web)                  | all       | particles  | Hessian                     | Forero-Romero et al. (2009)            |
| Velocity Shear Tensor (V-web)               | all       | particles  | Hessian                     | Hoffman et al. (2012b)                 |
| CLASSIC                                     | all       | particles  | Hessian                     | Kitaura & Angulo (2012)                |
| NEXUS+                                      | all       | particles  | Scale-Space, Hessian        | Cautun et al. (2012)                   |
| Multiscale Morphology Filter-2 (MMF-2)      | all except halos | particles  | Scale-Space, Hessian        | Aragon-Calvo (2004)                    |
| This work                                   | all except sheets/walls | particles  | Non-deterministic geometrical | Ramachandra & Shandarin (2015)         |

radius as a function of mass. It was assumed that, for halos of a constant density $\rho_H = 200\rho$, $R_{200} \sim M_{200}^{1/3}$. (Hansen et al. 2005) demonstrate through analysis of observation and simulation data that $M_H \sim N_{200}^{\alpha}$, where $\alpha$ is close to unity, and thus that $R_{200} \sim N_{200}^{1/3}$. We performed a similar fit on SIM using the halo mass $M$ and radii $R(M)$ calculated by a friend-of-friend cluster finding algorithm in Ester et al. (1996) and Turk et al. (2011), determining that

$$\langle R(M) \rangle_H = R_0 \left( \frac{M}{M_{\text{gal}}} \right)^{\alpha},$$

(1)

where $\langle R(M) \rangle_H$ is the expected halo radius for a given mass $M$, which corresponds well with these prior results.

Based on empirical calculations, we assumed that the probability density function for the radii $R(M)$ for halos of a given mass $M$ followed a log-normal distribution; this assumption was based on observation of the radius histogram for a given mass. For each $M = M_0$ with at least 100 particles, we fit $R(M = M_0)$ to a log-normal histogram where the mean $\mu = (R(M = M_0))$. We then empirically found that the standard deviation $\sigma(M)$ also exhibited a power-law dependence on the mass:

$$\sigma(M) = \sigma_0 \left( \frac{M}{M_{\text{gal}}} \right)^{\beta}$$

(2)

To create a halo, the halo mass was sampled from the HMF described by Warren et al. using algorithms from Turk et al. (2011) and Murray et al. (2013), and the halo radius was sampled from the corresponding log-normal distribution. Once a halo’s mass and radius were determined, particles were generated by sampling their radial distance from the halo’s center from a truncated spherical normal spatial distribution with standard deviation $\sigma_R(R_H) = \frac{R_H}{2}$. Halo masses were sampled from the HMF until the halo mass fraction reached the desired value, i.e. $\frac{M}{M_{\text{tot}}} \geq \delta_h$. Halo centers...


Table 2. A glossary of acronyms, measurement parameters, and numerical values used throughout this paper

| Acronym   | Definition                                                                 |
|-----------|-----------------------------------------------------------------------------|
| TSIM      | Toy model simulation for predictions                                         |
| SIM       | N-body simulation for predictions                                            |
| VOR       | Measurements of Voronoi cell volumes (density magnitude)                     |
| CMD       | Measurements of the distance between a particle and the center of mass of   |
|           | particles within a radius $R_{\text{CME}}$ (density magnitude)             |
| MI        | Measurements of the moment of inertia of particles within a radius $R_{\text{CME}}$ (density magnitude) |
| ENC       | Measurements of the number of particles within a radius $R_{\text{CME}}$ (density magnitude) |
| PCA       | Measurements of the difference between the maximum and minimum explained variance ratio from a PCA decomposition of particles within a radius $R_{\text{PCA}}$ (density field directionality) |

**General**

- **Particle mass in toy model and SIM**: $M_p = 7.55 \times 10^{10} \, M_\odot$
- **Number of particles in SIM**: $N_{\text{SIM}} = 13245$
- **R0**: 0.12 Mpc $h^{-1}$
- **$\alpha$**: 0.38
- **$\sigma_0$**: 0.12 Mpc $h^{-1}$
- **$\beta$**: 0.16

**Section 2.1.1 (Halo Generation)**

- **Minimum number of particles in toy model halos**: $N_{\text{min}} = 8$
- **Maximum number of particles in toy model halos**: $N_{\text{max}} = 13245$
- **$R_0$**: 0.12 Mpc $h^{-1}$
- **$\sigma_0$**: 0.12 Mpc $h^{-1}$
- **$\beta$**: 0.16

**Section 2.1.2 (Filament Generation)**

- **Minimum filament radius**: $R_F = 0.3$ Mpc $h^{-1}$
- **Maximum filament radius**: $R_F = 0.6$ Mpc $h^{-1}$
- **$B_\text{min}$**: 24.07 Mpc$^{-1}$ $h^4$
- **$B_\text{max}$**: 42.59 Mpc$^{-1}$ $h^4$
- **$n_0$**: 2.85 Mpc$^{-1}$ $h$
- **$n_F = n_F \pm \delta n_F$**: 3.75 ± 0.25 Mpc$^{-1}$ $h$

were pseudo-randomly placed throughout the particle field of side length $L_{\text{Toy}}$, then populated via the process described above.

2.1.2 Filament Generation

Filaments were constructed by first creating a spine, then populating the surrounding volume with particles. The spine was created by selecting two halo centers as endpoints, then creating a Bezier curve between them of degree 2 (Hermes 2017). The Bezier nodes were perturbed from the axis connecting the endpoints by $\Delta r_r$, were $0 < \Delta r_r < L_s$, where $L_s$ is the distance between the two endpoints. Particles were populated within a cylinder of radius $R_F \text{min} \leq R_F \leq R_F \text{max}$ around the spine. The radial number density was calculated by sampling from a uniform distribution with maximum and minimum values

\[
\begin{align*}
    n_F &= B_{\text{min}}(R_F - R_{\text{min}})^3 + n_0 \quad (3) \\
    n_F &= B_{\text{max}}(R_F - R_{\text{max}})^3 + n_0 \\
\end{align*}
\]

for $R_F \text{min} \leq R_F \leq R_F \text{max}$.

This corresponded with fixing the filament radial number density $n_F (R_F \text{max}) = n_0$ and allowing the density for the filaments with the minimum radius to vary as $n_F (R_F \text{min}) = n_F \pm \delta n_F$ max. These values were chosen to visually match filaments seen in SIM; we approximated the minimum and maximum filament radii and number densities using several prominent filaments in SIM, then applied an
arbitrarily-chosen bridging function (Eqn. 3) to ensure that smaller radii correlated with higher densities. As we will demonstrate that these selections do not strongly affect our predictions, this process may be easily replicated for another target dataset.

To create a filament, the two halos to use as endpoints were selected pseudorandomly (excluding halo pairs that already have a filament generated between one another), selecting a density, generating the spine, and using the density to calculate the number of particles inside. Particles were placed pseudorandomly along the spine, then perturbed orthogonally from the spine using a truncated normal distribution with standard deviation $\sigma_F(R_F) = \frac{3}{4}R_F$. This value was chosen so that the density at the edge of the filament was roughly $\frac{1}{4}$ the density of the center, ensuring that the filament boundary corresponded closely with the edge of the particle overdensity.

Filaments were created until the filament number density $n_F$ exceeded the desired density $n_{\text{tot}}$. To ensure that the filament number density was close to the desired density, filaments were iteratively destroyed and recreated until

$$\left| n_F - n_{\text{tot}} \delta M_F \right| \leq 0.05$$

(4)

Once this condition was satisfied, the filament mass fraction’s deviation from $\delta M_F$ was deemed small enough to begin background generation.

### 2.1.3 Background generation

Background (void) particles were sampled from a uniform distribution so that the number of void particles $N_{\text{tot}} = N_{\text{int}} - N_{\text{halo} - \text{tot}} - N_{\text{filament}} \approx N_{\text{int}} \delta M_V$. Note that, due to the fact that a truncated normal distribution was used to populate both halos and filaments, a sharp cutoff exists at the boundary of each halo and filament. This was a done to simplify the simulation and provide more control over its parameters; we will show that it did not affect our results.

#### 2.1.4 LSS Labels

Each particle inherited an LSS class label (halo, filament, or void) from its creation algorithm; however, to prevent contamination of measurement results from these segments, particles were relabelled according to a hierarchy. Particles within the boundaries of a halo were relabelled as halo particles; any remaining particles within the boundaries of a filament were relabelled as filament particles; and the rest remained void particles.

### 2.2 Measurements

Next, measurements of the local, global, and isotropic density and direction field were taken to use as training data. We used five separate measurements of the density magnitude and one measurement of directionality. While each of them may measure similar properties, each carries different information, so a combination can improve classification accuracy and robustness. Throughout the remainder of this paper, we discuss which measurements proved most effective. All measurements were normalized such that all values lay between 0 and 1, and are described below.

#### 2.2.1 Voronoi Cell Volume (VOR)

A Voronoi diagram is a method of partitioning of some multidimensional space. For each particle, there is a corresponding Voronoi cell, a region bounded by a convex polytope representing the set of all points that are closer to that point (using a Euclidean distance metric) than to any other point. We created a 3D Voronoi diagram and recorded the Voronoi cell’s volume for each particle (Virtanen et al. 2019; Bradford Barber et al. 1996; Gillies et al. 2007). As a Voronoi cell’s volume is closely related to the number of nearby particles, we expect that the volume of a particle’s corresponding Voronoi cell will act as an effective measure of local density; in particular, we expect it to effectively classify halo particles.

#### 2.2.2 Center of Mass Distance (CMD) and Moment of Inertia (MI)

Using a KD tree, the coordinates for particles within a radius $R_{\text{CMD}}$ of each particle were found. We found the center of mass for particles in this region, then used the distance between the center of mass and the particle of interest as a training feature. For small $R_{\text{CMD}}$, this algorithm measures the local density, while for large $R_{\text{CMD}}$, this measures the global density.

Using the same set of particles, the moment of inertia was calculated. The radius is expected to correspond similarly with regard to the local and global densities.

Both of these algorithms are expected to primarily influence halo particle predictions.

#### 2.2.3 Number of Particles Enclosed (ENC)

Using a KD tree, the number of particles within a radius $R_{\text{ENC}}$ of each particle were calculated. For small $R_{\text{ENC}}$, this algorithm measures the local density, while for large $R_{\text{ENC}}$, this measures the global density. This calculation will primarily affect halo particle predictions, especially for those near the centers of the halos.

#### 2.2.4 Distance to the k-Nearest Neighbor (KNN)

A ball tree was used to find the distance to the $k$-nearest neighbors for each particle, where the $k$-values used can be found in Table 2. For small $k$, this algorithm measures the local density; while for large $k$, this algorithm measures the global density. We expect this algorithm to primarily influence halo classification, primarily for small halos. As the smallest halos consist of 8 particles (typically, it is assumed that the minimum size for halos is 3 - 20 particles), it is crucial to include $k \approx 8$ measurements in the measurements to differentiate between halos and filaments/voids.

All previously listed measurements take into account properties of all particles within a fixed radius. As a result, they may fail to account for the spatial extent of the structure a particle is a member of. On the other hand, KNN measures only measure the properties of the environment of the closest particles. By training with very small $k$-values, we can obtain information about not only the density near a particular particle, but also of the natural length scale of the structure that particle is a member of, as the $k^{th}$ nearest neighbor for small $k$-values will likely contain only particles that are a member of that structure. As a result, we expect KNN measurements to provide information that cannot be obtained with the other density magnitude measurements.
2.2.5 Principal Component Analysis of Local Particles (PCA)

Principal component analysis (PCA) provides a method for determining the principal component axes, an uncorrelated orthogonal basis set such that the first component takes on the highest possible variance. Using the explained variances, this provided a method for determining the directionality of the data for use in differentiating between filaments and halos.

Prior to performing PCA analysis, the particle field was resampled to ensure an adequate number of particles were contained within each sphere surrounding a given particle. First, a Gaussian filter with standard deviation $\sigma_{\text{PCA}}$ was applied to the particles within a given sphere to smooth the density distribution. After binning the coordinates within a given sphere, the density distribution was resampled and particles placed such that the total number density increased by a factor of $\delta n_{\text{PCA}}$. An additional uniform background was added with density $1.0 ~\text{Mpc}^{-3} ~h^{-1}$ to prevent the effects of background particles from being washed out.

After resampling, a PCA decomposition (Pedregosa et al. 2011; Tipping & Bishop 1998) was performed on all particles within a radius $R_{\text{PCA}}$, and the explained variance ratio for each axis was found. The variance of particles within this region may be described by a covariance matrix $C$, with total variance $\Sigma_{ij} C_{ij} = \sigma^2_C$. After performing the PCA decomposition, $C$ undergoes the transformation $C \rightarrow C'$ such that $C'$ is diagonal and $\text{Tr}(C') = \sigma^2_C$. The explained variance of principal component axis $i$ is $C'_{ii}$, and the explained variance ratio is $\frac{C'_{ii}}{\sigma^2_C}$. After PCA decomposition is performed, a data set that may initially be correlated is transformed to a data set that exhibits no cross correlation. The explained variance ratio describes the proportion of the total variance $\sigma^2_C$ that may be attributed to the variance of particles with respect to a given axis. Intuitively, as the principal component axes correspond to the principal axes when calculating the moment of inertia, this provides information about the spread of particles about an axis such that the mass distribution around that axis is uniform.

After calculating the explained variance ratio for each axis for each particle, the value of $V$ was found, where

$$V = -\ln (\Delta r^2),$$

$$\Delta r^2 = \sigma^2_{\text{max}} - \sigma^2_{\text{min}}.$$  \hspace{1cm} (5)

$\Delta r^2$ describes the difference between the minimum and maximum explained variance ratio. The natural logarithm of this difference was taken to accentuate the differences between the filaments and halos so that, by using $V$ as a training metric, fewer data points would be required to perform accurate classification.

For a filament, it would be expected that the variance about the spin axis would be much smaller than the variance around the other axes due to the density field being preferentially aligned along this axis, producing a small value for $V_F$. In contrast, as the halo density field tends to exhibit very little directionality preference, it would be expected that the explained variances should vary little between the different axes, meaning that $V_H$ would be large. Though it may be expected that the explained variance ratios for void particles would exhibit similar properties to those of halos, and hence have a large $V$; however, due to the low density of voids, nearby structures would heavily influence the directionality values of void particles. As a result, it is expected that void particles should exhibit a small value for $V_V$, though a larger spread that of filaments. While this may lead to difficulty in differentiating between filaments and voids, the combination of this measurement with a density magnitude measurement should enable their differentiation due to filaments exhibiting a much higher density than voids.

2.3 Training and Prediction

A random forest algorithm (Breiman 2001) is a supervised learning method constructed from several decision trees. Each tree classifies a given particle using a randomized subset of features, and the class assigned to that particle is the class selected by the largest number of trees. We chose this algorithm because it improves the robustness of the classifier when using a small number of features and allows the assignment of class probabilities; its effectiveness has been demonstrated in (Carrasco Kind & Brunner 2013). By subsampling the set of features and performing classification using a large number of independent trees, the statistical robustness of the classifier is improved. As we aim to simplify the classifier as much as possible by minimizing the number of features required for classification, a random forest algorithm ensures that our classification is affected less by statistical fluctuations resulting from a small sample size. In addition, we may assign each particle a class probability based on the number of independent trees that assigned that particle a given class.

A random forest algorithm (Pedregosa et al. 2011; Breiman 2001) was trained using the measurements from one simulation. While it is typical to use multiple datasets to train a classifier, we found that additional training datasets did not influence the results, so only one was used. A random forest algorithm was chosen due to the improvements in classification accuracy it provides when training using a data set substantially smaller than the prediction data set. Using 200 trees, class prediction probabilities were generated for each particle in $\text{SIM}$. Each particle was then assigned the class associated with the highest probability; in cases where multiple classes were assigned the maximum probability, halos were prioritized over filaments, which were prioritized over voids. An FOF clustering algorithm (Davis et al. 1985) was applied to halo particles to differentiate halos from one another, allowing the creation of an HMF from the data and remove noise. This clusterer was not applied to filament or void particles due to their low density or elongated structure; nevertheless, this information is not important for the purposes of this work.

3 PREDICTION

It is expected that the primary way to differentiate between structures would be through local density magnitude calculations; however, to account for the arbitrary densities used for filaments, density field directionality measurements were included to differentiate between filaments and halos/voids. Of the methods used, only $\text{PCA}$ provided directionality measurements.

3.1 Toy-to-$\text{SIM}$

First, predictions were made on $\text{SIM}$ using a 660,000-particle toy model. A 6 Mpc thick slice of the toy model is shown in Figure 2.

3.1.1 Measurement Histograms

To provide an initial guess as to which density magnitude methods would provide the most information, histograms were created using
measurements on the training dataset. After performing all measurements on the training data set, each measurement was normalized so that all values fell between 0 and 1, ensuring that all measurement methods would be treated equally when training. Measurements that provide the most information for use in classification should show little overlap between the measurements on each structure and exhibit large peaks distinct from one another. Examples of some measurement histograms are displayed in Figure 7.

Examining Figure 7, it appears that CMD calculations will provide little information when classifying particles because there is little difference between the location and magnitude of peaks; this issue is also present in MI and ENC. VOR also shows large peaks for small volumes; however, the distribution is much more spread out, indicating that it may be useful for classification. KNN and PCA both exhibit strong differentiation between the peaks; in particular, KNN provides substantial information for differentiating between all classes, while PCA separates filaments and voids from halos.

Figure 7 provides clues as to which measurements will be most effective for classification. The histograms for 7a CMD and 7b VOR both exhibit a large spike at the origin, indicating that, for each class, the distance from nearly every particle to the center of mass was very small, save for several very large outliers.

However, the measurements for 7c KNN exhibited much more differentiation between each of the different structure classes. As expected based on local density, halos exhibited the lowest distance to the 8th-nearest neighbor, followed by filaments, and finally voids. The larger spread on filaments and voids reflects the higher likelihood of contamination by nearby structures due to their low density. As the classes are strongly differentiated from one another, we expect KNN to be an effective proxy for local density.

While not shown, larger k-values led to less differentiation between the classes due to each class exhibiting a greater spread.

The measurement histograms for 7d PCA show strong peaks for filaments and voids, yet a multi-peaked halo distribution with large spread. The strong peak for filaments, especially compared to the poorly-defined halo distribution, indicates that these calculations are effectively measuring the density field directionality, as the roughly spherical halos are not expected to exhibit substantial directionality. The strong peak for voids is likely due to contamination by nearby structures: as voids have very low density, any particles from adjacent halos or filaments would lead to a strong directionality. These results bode well for the use of PCA in tandem with KNN to create a robust classifier, as PCA provides a natural way to differentiate between halos and filaments independent of the local density magnitude. While it may seem that these calculations may cause difficulty when distinguishing between filaments and voids, the strong differentiation between these structures from KNN calculations is expected to prevent this issue.

### 3.1.2 Class Predictions

Next, predictions were made on SIM (described in Section 2). In order to achieve probabilistic classification, each particle was classified using 200 trees, each of which independently assigned that particle a class. The classification probability for a given class was the proportion of estimators that assigned a particular particle that class. The probability difference plots in Figures 4 and 5 provide a visualization of these probabilities. Each halo and filament particle was plotted, and the color corresponded to the difference in probability of halo vs. filament classification (normalized by removing the void probability from the calculation such that the total

---

**Figure 2.** (a) A 6 Mpc thick slice of the 3D training dataset used throughout this work. This figure includes void particles (blue). (b) The same training data set, but with void particles removed. Removing the void particles provides a better view of the filaments. Both of these figures include the largest halo.
Figure 3. Measurement histograms from the true values in the toy model for (a) CMD, (b) VOR, (c) KNN, and (d) PCA. (a) and (b) show features with less distinguishable distributions, while (c) and (d) are more distinguishable.

probability was 1). A probability difference of 1 indicates that every estimator classified that particle as a member of a halo, while a probability difference of -1 indicates that every estimator classified that particle as a member of a filament. A particle colored black indicates that the estimator was unable to precisely differentiate between the particle’s class, indicating that there is ambiguity as to whether it is a halo or filament member. It would be expected that, near the high-density center of halos, the probability difference would be close to 1, while near the edges, especially where the halo connected to a filament, the probability difference would be closer to 0. Note that void particles are not displayed in these plots as void probability assignments were generally close to unity.

All particles in SIM were classified by each metric set as detailed below. Figures 4 and 5 show the halo and filament class probabilities assigned to particles within a 7 Mpc thick slice of this 3D simulation. While the predictions made by each feature set shown are visually realistic, we discuss their differences below. Figures 4a and 4b show the class probabilities assigned by a classifier trained using only VOR and CMD, respectively. Relative to the predictions by CMD, VOR overpredicted the number of halo and filament particles, indicating that it was not sensitive to the low density void regions. In addition, the class probabilities assigned were generally lower, indicating that VOR alone did not provide enough information to distinguish between the classes easily. On
the training dataset seen in Figure 2. To determine whether the toy level of detail; a plot of TSIM predictions on simulations with substantially demonstrate the robustness of our methodology as due to its similar pre-

B. Buncher

P the high-probability halo regions (clumps). These regions had a density magnitude between that of the halos predicted by the classifier with \( k \leq 40 \) were generally larger and had a higher class probability. In addition, both classifiers, particularly the classifier with \( k \leq 40 \), predicted the existence of elongated halos, demonstrating difficulty in distinguishing between dense filaments and low-density halos.

The inclusion of PCA calculations (Figure 5c and 5d) helped eliminate these issues by providing information emphasizing the directionality component of filaments. Including larger \( k \) values for KNN calculations led to many filament particles being classified as halo particles, likely due to the fact that measurements using large \( k \) would often include information from a variety of structure classes, blurring their distinction. However, even though the classifier in Figure 5d used \( k \)-values much larger than those in 5c, the class probabilities are very similar, especially when compared to Figures 5a and 5b. By including information about the local density field directionality, class predictions were less affected by contamination from distant structures.

Figure 6 show the HMFs for the predicted halos. For all classifiers, it is clear that classification exhibits the greatest error for small halos; elsewhere, the predicted HMF corresponded very closely with (Warren et al. 2006). The inaccuracy for small halos had the same source as for FOF calculations (Springel et al. 2005).

Additional evidence for the effectiveness of KNN and PCA measurements may be seen in Figure 7. Here, it may be seen that, unlike CMD and VOR, KNN and PCA calculations exhibit substantial differentiation between the different structures. In particular, when comparing these measurements to Figure 7, the distributions of the KNN and PCA calculations appear quite similar, though with somewhat less differentiation due to the fact that they were used on different datasets. This demonstrates that these measurements provided similar information about the training dataset as they did about SIM, which improves the robustness of the classifier. The stronger peak for halos in PCA calculations is likely due to structure of filaments as they connect with halos. As the density of filaments increases near halos, the directionality of halo particles is likely to be substantially higher than in the toy model. Nevertheless, there is still substantial differentiation between the structures.

3.2 Toy-to-SIM

The true class values for SIM are not known, and there is not a known verifiable method that can predict these true classes. Thus, direct verification of the predictions made by any classifier is not possible; however, by comparing statistics between our class predictions of particles in SIM to that of a toy model TSIM, we demonstrate the robustness of our methodology as due to its similar predictions on simulations with substantially different structure and level of detail; a plot of TSIM appears nearly identical to that of the training dataset seen in Figure 2. To determine whether the toy model provides enough information, we used the same toy model shown in Figure 2 to train a classifier to make predictions on TSIM, an realization of the toy model that uses the same parameters as the training dataset.

Next, predictions were made on TSIM; these predictions may be seen in Figures 8 and 9. The properties of these predictions were
Probabilistic trained cosmic web classification

To toy-to-SIM Predictions for $KNN + PCA$

Figure 5. Prediction results for SIM. The halo-filament predictions, colored based on relative prediction likelihood, are shown in (a) $(KNN, k \leq 8)$, (b) $(KNN, k \leq 40)$, (c) $(KNN + PCA, k \leq 8)$, and (d) $(KNN + PCA, k = 40)$. The displayed particles are from the same 7 Mpc thick slice of the 3D N-body simulation SIM centered on the halo with the greatest mass as in Figure 4.

largely the same as those of the predictions made on SIM seen in Figure 4 and 5: VOR and CMD (8a and 8b, respectively) generally overpredicted halo abundance, with VOR predictions having lower probability assignments than CMD; $KNN$-trained predictors with only small $k$ (Figure 9a) assigned lower probabilities to all particles and predicted smaller halos than a classifier with large $k$ (Figure 9b); and the addition of $PCA$ calculations (Figure 9c and 9d) helped remove the differences in predicted classes associated with different ranges in $k$-values.

One notable exception, however, lies in the predictions made by the classifier trained with only CMD (8b). Unlike in the predictions made by this classifier on SIM (Figure 8b), predictions on TSIM (Figure 8) did not feature clumps of ambiguous particles near particularly dense halo-filament boundaries. This is likely due to the fact that the training data used an identical generation procedure to TSIM. This highlights the robustness of $KNN$ (and $PCA$) calculations. The predictions made by $CMD$ calculations are heavily tied to the generation algorithm, indicating that it is not suited for training using a simplified dataset such as the toy model we developed. On the other hand, the predictions made by a classifier trained with $KNN$ calculations, especially when paired with $PCA$ calculations, are less affected by the exact structure of the toy model, enabling their use even when training was performed using a simplified toy model. This indicates that $KNN$ calculations establish a natural length scale for halos and filaments together when performing classification, and the addition of $PCA$ calculations help distinguish between halos and filaments by establishing distinct length scales for these structures individually. Note that $PCA$ calculations are most effective for structures with a length scale that is not much larger than $R_{PCA}$. Additional discussion of these properties may be found in Section 4.4.

In the toy model, the border between halos and voids manifested itself as a sharp density cutoff that did not exist in SIM; this property was a conscious decision not only to simplify the toy
model generation process, but also to prevent overfitting the halo-filament and halo-void boundaries. As the classes of particles on these boundaries are ambiguous due to a lack of strong directionality and a density gradient, any algorithm that would assign classes to particles in these regions in the toy model would be arbitrary. As a result of this decision, however, halo-filament and halo-void borders in the toy model are very clear, while this is not the case in SIM. Density magnitude calculations in the toy model had a different distribution near these boundaries than in SIM: as filament densities lay between those of halos and voids in the toy model, there would be some region on halo-void boundaries where the density corresponded well with filament densities in the toy model. The inclusion of PCA measurements helped fill this information gap: as halos are typically found within filaments, particles on halo-void boundaries would exhibit some density field directionality due to their proximity to filaments. This directionality would be independent of density magnitude, meaning that directionality measurements of halo-void boundary particles was similar in both the toy model and SIM, helping with their classification. As a result, though the toy model exhibited simpler, less realistic halo-filament and halo-void boundaries than in SIM in order to improve computation time, the combination of appropriate measurement techniques and the robustness of ML ensured realistic prediction properties.

Figure 10 shows the HMFs for Toy-to-TSIM predictions. All feature combinations produced very accurate HMFs; as a result, we focus on the low- and high-mass regions, which feature the largest visual deviations between classifiers. For all figures, it is clear that classification generally exhibits the highest amount of error at the extreme ends of the mass range. As in the predictions on SIM, the inaccuracy for small halos had the same source as for FOF calculations (Springel et al. 2005). However, note that the high-mass deviation for predictions on TSIM is an order of magnitude larger than the high-mass error in SIM (see Figure 6). This indicates that, though there were few large halos in both the training data and TSIM, the source of this error may be due to the lack of large halos in TSIM rather than insufficient training for large halos. This shows the high scalability of our algorithm: though training was performed using very few large halos, classification accuracy on SIM in this region was extremely high due to the robustness of the classifier.

4 ANALYSIS

By correlating predictions made on SIM with those made on TSIM, we can demonstrate that the toy model is effective as a training data set: if the predictions made on SIM are statistically and/or visually similar to those on TSIM, we have demonstrated that the methodology is robust enough to make similar predictions on two datasets with markedly different properties, allowing its use to be expanded to observed data. The goal of this section is not only to demonstrate the validity of our predictions, but also to establish the importance of utilizing measurements of both local density magnitude and directionality to ensure that our predictions are not strongly influenced by the somewhat arbitrary parameters used to generate structures, particularly filaments, in the toy model.

4.1 Feature Importances

First, we aimed to decrease the number of measurements required to minimize computation time and understand the role of each metric. We utilized the classifier’s feature importances, which describes how relevant each feature was when performing class assignment; this was determined by the frequency a metric was used to choose a branch as the classifier descended a tree. Figure 11 shows the feature importances for a variety of different metrics. From here, we see that KNN is weighted the most heavily, indicating that it may provide valuable information for the classifier. This expectation correlates with the measurement histogram seen in Figure 7c, where it may be seen that the distance to the eighth-nearest neighbor separates each of the classes distinctly from one another.

Figure 11 shows the feature importances for several different feature sets. In general, small radii/k were deemed most important. From 11a, the feature importances for all features, density magnitude calculations were weighted more heavily than PCA, and of the density magnitude calculations, KNN was weighted most heavily,
reflecting the lack of differentiation between structures in the measurement histograms for the other density magnitude calculations. In both Figures 11a and 11b, the feature importances for KNN, the most important measurements were those with $k \leq 8$, reaching a peak at $k = 8$. This is likely due to the fact that the smallest halos had 8 particles, and as small halos dominated the halo mass function, they will be utilized most by the training algorithm to determine a feature’s importance. These phenomena were also reflected in 11c $KNN + PCA$.

4.2 ROC AUC

The receiver operating characteristic (ROC) curve is a way of demonstrating a classifier’s ability to make predictions. It consists of a plot of the classifier’s true positive rate as a function of its false positive rate. A classifier that cannot discriminate between classes would have a 50% probability of assigning the correct class to a given data point, and would have equal true positive and false positive rates; hence, its ROC curve would appear as a line with unit slope. On the other hand, an effective classifier would have a much larger true positive rate than false positive rate. The area under the curve (AUC) is a measure for a classifier’s effectiveness: a classi-
The particles seen are from a 7 Mpc thick slice of the 3D N-body simulation colored based on the relative likelihood of assignment to a particular class. Class predictions made using \(\text{VOR}\) and \(\text{CMD}\) measurements for \(k \leq 8, 40\). Notably, the addition of \(\text{PCA}\) calculations improved classifier performance for all LSS classes, particularly for filaments. As seen in the \(\text{SIM}\) predictions made by a classifier trained by \(\text{KNN} + \text{PCA}\) (see Figure 5), the addition of \(\text{PCA}\) calculations diminished the dependence of halo classification on the values of \(k\) used. Figure 12c demonstrates that this stabilization applies to all classes, especially filaments. The benefits of this stabilization are immense: using large \(k\)-values allows the classifier to take into account the global environment when performing classification, improving classification of large halos, but lessens its sensitivity to properties of the local environment. Including \(\text{PCA}\) calculations enables the global environment to be used in training without contaminating information about small-scale properties. The combination of small-scale and large-scale information in training enables classification based on the density field contrast, which is particularly useful for classifying halos isolated in large void volumes (a major issue discussed in Tsizh et al. (2019), 2019 and Libeskind et al., 2018).

4.3 HMF Comparison: Mean Absolute Proportion

We define the mean absolute proportion (MAP) as

\[
\text{MAP} = \frac{\mu(|n_{\text{pred}} - n_{\text{Warren}}|)}{\mu(|n_{\text{Warren}}|)}, \tag{6}
\]

where \(n_{\text{pred}}\) and \(n_{\text{Warren}}\) were the predicted and Warren et al. (2006) HMF, respectively. In Figure 13, we show the MAP for each of the metrics for both \(\text{SIM}\) and \(\text{SIM}\). The metrics are ordered based on the MAP value for \(\text{SIM}\). From this plot, it can be seen that, as before, classifiers trained using only \(\text{VOR}\) or \(\text{PCA}\) performed substantially worse than all other feature combinations when classifying \(\text{SIM}\) particles. In addition, though not previously discussed, a classifier trained using \(\text{VOR} + \text{PCA}\) performed poorly when classifying \(\text{SIM}\) particles, emphasizing the importance of a robust density magnitude metric. For \(\text{SIM}\), these three also performed very poorly; however, unlike in \(\text{SIM}\) classification, classifiers trained using \(\text{CMD}, M_1\), and ENC also exhibited a large MAP. This further supports the conclusion that these methods are not effective due to their strong dependence on the training model generation however, these values did not exist in \(\text{SIM}\). As a result, we chose predictions by all features to use as a fiducial comparison dataset. This provides the most generality, as it effectively allows the comparison of each prediction set to all others simultaneously. As we found that most metric sets produced similar results, and that the AUC for \(\text{SIM}\) predictions using all features was the largest for all classes, we believe that predictions made by all features will provide a sufficient approximation to the true class values to use as a fiducial comparison dataset.

In general, the ROC curves in Figure 12 demonstrate that filament classification was the most difficult. In addition, classification on \(\text{SIM}\) was more accurate than on \(\text{SIM}\), as evidenced by the shape of the curves and the AUC values.

Figure 12a and 12b show the ROC curves for classifiers trained using only \(\text{VOR}, \text{CMD},\) and \(\text{PCA}\). From this, it can be seen that classifiers trained on \(\text{VOR}\) or \(\text{PCA}\) alone were not effective when classifying particles in \(\text{SIM}\) and on \(\text{SIM}\). This further demonstrates that \(\text{VOR}\) does not suffice as a proxy for local density magnitude. The poor performance of \(\text{PCA}\) alone may be attributed to the lack of information provided to the classifier about local density magnitude.

Figure 12c and 12d show the ROC curves for classifiers trained using \(\text{KNN} (\text{PCA})\) measurements for \(k \leq 8, 40\). Notably, the addition of \(\text{PCA}\) calculations improved classifier performance for all LSS classes, particularly for filaments. As seen in the \(\text{SIM}\) predictions made by a classifier trained by \(\text{KNN} + \text{PCA}\) (see Figure 5), the addition of \(\text{PCA}\) calculations diminished the dependence of halo classification on the values of \(k\) used. Figure 12c demonstrates that this stabilization applies to all classes, especially filaments. The benefits of this stabilization are immense: using large \(k\)-values allows the classifier to take into account the global environment when performing classification, improving classification of large halos, but lessens its sensitivity to properties of the local environment. Including \(\text{PCA}\) calculations enables the global environment to be used in training without contaminating information about small-scale properties. The combination of small-scale and large-scale information in training enables classification based on the density field contrast, which is particularly useful for classifying halos isolated in large void volumes (a major issue discussed in Tsizh et al. (2019), 2019 and Libeskind et al., 2018).
Probabilistic trained cosmic web classification

Toyn-TOISM Predictions for KNN (+ PCA)

Figure 9. Prediction results for TOISM. The halo-filament predictions, colored based on relative prediction likelihood, are shown in (a) (KNN, k \leq 8), (b) (KNN, k \leq 40), (c) (KNN + PCA, k \leq 8), and (d) (KNN + PCA, k = 40). The particles seen are from the same 7 Mpc thick slice of the 3D N-body simulation TSIM as 8.

algorithm. In addition, for SIM, there is a noticeable decrease in performance for KNN and KNN + PCA for large k; however, this deviation was curbed by the addition of PCA calculations, indicating that the contamination from distant structures affected classification less. Notably, KNN + PCA for k = 5 was noticeably less accurate than all other KNN and KNN + PCA classifiers. As the smallest halos contained a minimum of 8 particles, this classifier’s poor performance likely arises from the classifier taking into account the density only at very small distance scales. This lack of information of the density magnitude likely resulted in poor classification for the same reasons that classification using only PCA calculations failed.

The analysis performed in Sections 3.1.1 and 4.1 both indicated that KNN calculations would likely be the most important, and the results from Sections 4.2 and 4.3 supported this by demonstrating that KNN calculations were robust and generated the most accurate results for both SIM and TSIM. As a result, in the following section, we will use KNN alone as the predictor for local density magnitude. In addition, Sections 3.1.1, 4.2, and 4.3 suggest that the inclusion of PCA calculations may also be of benefit, particularly for filament classification. In the next section, we present additional measurements to emphasize the importance of PCA calculations when creating a robust classifier.

4.4 Robustness of PCA Calculations

As discussed previously, PCA calculations provide substantial benefit when paired with KNN calculations; however, as noted in Section 4.3, these benefits are most apparent for structures with length scales no larger than the maximum radius used in PCA calculations (r = 2.0 Mpc). Halo vs. filament probability plots for a region where this is true can be seen in Figure 14.
HMFs for predictions made on TSIM using (a) VOR and CMD and (b) KNN (+ PCA) for $k \leq 8, 40$.

Figure 10. Halo mass functions for predictions made on TSIM using (a) VOR and CMD and (b) KNN (+ PCA) for $k \leq 8, 40$.

In addition, due to the ambiguous class of particles on halo-filament boundaries, we expect the particles in these regions would have a probability difference near zero (these particles are colored black in Figure 14). However, as the density contrast between halos and voids is very large, we expect the class assignments on halo-void boundaries to have higher probabilities. In Figure 14a, there are an abundance of ambiguous particles on halo-filament and halo-void boundaries, as well as in the interior of halos; this is most visible in the halo in the upper-left of Region II. In contrast, Figure 14b predominantly lacks ambiguous border particles. The addition of PCA calculations in 14c and 14d stabilize these border regions, removing most ambiguous particles inside halo interiors and on halo-void boundaries, and clarify the halo-filament boundary with a thin layer of ambiguous particles, consistently making particle classes more physically realistic.

Figure 14a and 14b show the classification results for classifiers trained with KNN, $k \leq 8, 40$, respectively. As discussed previously, introducing PCA calculations (Figure 14c and 14d) curbed the classifier’s dependence on the maximum value of $k$ and improved its ability to distinguish between halos and filaments. Regions I and II in Figure 14 demonstrate this clearly: in 14a and 14b, the classes assigned to particles varied greatly, with the classifier with $k \leq 40$ substantially overpredicting halo particles. This produced halos that are elongated along one axis, a hallmark trait of filaments. However, PCA calculations prevented this issue, as exemplified by the similarity between these regions in 14c and 14d. The reclassification of these elongated halos as filaments shows that PCA calculations are an effective proxy for density field directionality.

Now that we have established the robustness of our classifier, we aim to show that the exact construction of our toy model (particularly that of filaments, which was based on visual appearance) did not substantially bias our results.

4.4.1 Structural Mass Fractions

Next, we aim to demonstrate the robustness provided by the addition of PCA calculations using trends in the normalized mass fractions. Here, we assumed the null hypothesis: the predictions made by a given classifier are independent of the set of features used, so the predictions made by a classifier using a particular feature set is taken to be a single trial in a set of predictions made by a single classifier. Under this assumption, we may normalize the predictions made by a classifier using a particular feature set by removing differences resulting from their global properties. The normalized mass fractions $P_n$ for a set of features $f$ are defined by...
Probabilistic trained cosmic web classification

Feature Importances for Toy-to-SIM

Figure 11. The feature importances for the Toy-to-SIM calculations using (a) all metrics, (b) KNN, and (c) KNN + PCA.

\[ \bar{P}_f = \frac{P_f - \bar{P}}{\sigma} \]

where, for a given class (halo, filament, or void) \( P_f \) is the actual mass fraction for a feature set \( f \), \( \bar{P} \) is the average of these values over all \( f \), and \( \sigma \) is the standard deviation of these values over all \( f \). Note that the feature sets used in this analysis consisted of KNN (+ PCA) for a variety of values for \( k \).

A plot of the normalized mass fractions can be seen in Figure 16, where several trends may be seen. For both TSIM and SIM, as \( k \) increased, the normalized mass fraction for halos increased, while that of filaments decreased, regardless of whether or not PCA calculations were used. In addition, for both SIM and TSIM, PCA calculations generally affected the values of the normalized mass fractions for halos and filaments minimally. Without PCA calculations, the TSIM normalized mass fraction for both halos and filaments varied widely for small values of \( k \); for filaments in particular, this large difference was seen for \( k < 30 \). In addition, for \( k < 10 \), the TSIM normalized mass fractions for halos and filaments did not exhibit a particular trend with increasing \( k \).

However, the addition of PCA calculations substantially improved the classifier’s robustness by making the normalized mass fraction values and trends more consistent between SIM and TSIM for halos and filaments, particularly for \( k < 10 \). This is likely because halos are only found as nodes on filaments in the toy model, so for very small halos, using only density calculations blurred the lines between halos and filaments through contamination of filament point measurements by nearby halos. The incorporation of PCA calculations helped differentiate filaments from halos, improving the robustness. This claim is further supported by the very large filament normalized mass fraction for \( k = 5 \); as the smallest halos in the toy model had 8 particles, measurements for \( k < 5 \) would not be able to include all particles in a halo. As a result, many halos were classified as filaments due to the fact that directionality effects would dominate the local density magnitude for halos, causing overrepresentation of filaments.

Voids showed no recognizable trend for TSIM, regardless of
whether or not PCA calculations were included. For SIM predictions, the normalized mass fraction showed a general downward trend, though as before, the normalized mass fraction without PCA calculations for \( k \leq 10 \) showed an inconsistent trend.

5 CONCLUSIONS

We have presented a novel method for cosmic web classification, demonstrating that supervised machine learning using a simplified toy model as training data provides a potential avenue for robust and efficient cosmic web classification. The use of a random forest algorithm in particular provides a method for achieving probabilistic classification. In addition, we found that the use of density field directionality measurements in tandem with local density
**Figure 13.** A plot of the HMF MAP; the metric sets are ordered such that the TSIM MAP decreased from left to right.

**Halo vs. Filament Probability Scatterplots for KNN**

(a) **KNN**, \( k \leq 8 \) Halo vs. Filament Predictions on SIM

(b) **KNN**, \( k \leq 40 \) Halo vs. Filament Predictions on SIM

(c) **KNN + PCA**, \( k \leq 8 \) Halo vs. Filament Predictions on SIM

(d) **KNN + PCA**, \( k \leq 40 \) Halo vs. Filament Predictions on SIM

**Figure 14.** Halo vs. filament probability scatterplots for **KNN** ((a) and (b)) and **KNN + PCA** ((c) and (d)) for \( k \leq 8, 40 \). Particles are from a slice of width 2.0 Mpc \( h^{-1} \). In this region, most halos had radii that were not substantially larger than the maximum radius used for **PCA** calculations (\( r = 2.0 \) Mpc).
magnitude measurements are crucial for distinguishing between halos, filaments, and voids. This combination ensures that the classes are assigned consistently in a variety of different environments by training the classifier to favor local density field contrast rather than its magnitude. In particular, we have provided a method that can classify isolated halos inside large voids, an outstanding problem discussed in (Tsizh et al. 2019; Libeskind et al. 2018). Through calculating and comparing statistical data about our classifications, we found a method to verify our calculations, demonstrating that our algorithm is robust and is not biased by our training data creation algorithm.

The key advantage of our method is the speed and efficiency of toy model generation. While N-body simulations require substantial computational expense and lack true class values, a new toy model can be generated much more efficiently, and this model provides enough information to accurately classify substantially more complicated N-body simulations and potentially observed data. This makes the method especially suitable for large datasets: using a single training dataset, we were able to assign class values to an N-body simulation substantially larger than the toy model. Due to the speed of generation, our method is extremely scalable, as generating additional training datasets would allow us to assign class values to very large N-body simulations at no cost.

In addition, the use of a toy model is particularly suited for cosmic web classification of observed data. Observed datasets contain masked regions and areas with non-uniform depth. The use of a toy model helps account for these issues: for each field, an individual toy model can be generated that matches the density, mask, and depth of that field. By classifying each field using its corresponding training dataset, class assignments would be generated consistently for each field. The use of periodic boundary conditions or padding (as we used here) could avoid issues associated with masked regions.

The ability to assign probabilistic classes to individual galaxies opens doors to a variety of novel data analysis techniques. Observables such as density and physical composition are known to be linked to LSS class membership, so correlations with class probability would enable novel methods for understanding these relationships. For example, spectral analysis may be used to understand chemical composition of galaxies in different environments. Correlating R/G-band magnitudes with cluster-filament probability differences could help establish not only the differences in the chemical compositions of clusters and filaments, but also how that composition changes as the cluster-filament boundary is crossed.

As PCA calculations clarify halo-filament boundaries, the application of a directionality metric can be used to differentiate halos and filaments in general, as well as study the fundamental properties of LSS. Capturing snapshots of an N-body simulation over large time scales and tracking filament halos as they cross a halo-filament boundary could provide a deeper understanding of the matter inflow from filaments to halos, separating its role in halo formation and collapse from other processes.

While our algorithm is a robust classifier for halos, filaments, and voids, our feature selection is not ideal; in particular, we found
Probabilistic trained cosmic web classification

that PCA calculations did not perform well for filaments and halos with radii substantially larger than the radius used in the PCA decomposition calculations. As increasing this radius leads to substantial cross-contamination, future work should focus on identifying and implementing a directionality metric that can more effectively capture properties of large filaments and halos.

In addition, we chose not to differentiate between sheets/walls and filaments in our classifier to the complexity of creating a simple algorithm for this purpose. Future work could be devoted to expanding this algorithm to allow sheet/wall classification.

Though untested, we expect our classifier to be just as effective in cosmology models other than ΛCDM. The length scales for LSS are much larger than the scales at which deviations from ΛCDM are detectable. As our training data only reproduces properties of LSS at these large scales, we expect classification to be independent of the cosmology of the target data set, so the same training data sets may be used to classify fields with equivalent geometric parameters (e.g. average density) but different cosmologies.

6 SOFTWARES USED

The Python packages Bezier [Hermes 2017], DBSCAN [Ester et al. 1996], HMF [Murray et al. 2013], matplotlib [Hunter 2007], numpyp [Oliphant 2006], scikit-learn [Pedregosa et al. 2011], SciPy [Virtanen et al. 2019], Shapely [Gillies et al. 2007], and yt [Turk et al. 2011] were used extensively in this work.

7 ACKNOWLEDGEMENTS

This material is based upon work supported by the National Science Foundation Graduate Research Fellowship Program under Grant No. DGE — 1746047.

The second author’s work has been supported by grants NSF AST 07-15036 and NSF AST 08-13543.

This research is part of the Blue Waters sustained-petascale computing project, which is supported by the National Science Foundation (awards OCI-0725070 and ACI-1238993) and the state of Illinois. Blue Waters is a joint effort of the University of Illinois at Urbana-Champaign and its National Center for Supercomputing Applications.

REFERENCES

Alpaslan M., et al., 2013, Monthly Notices of the Royal Astronomical Society, 438, 177
Aragon-Calvo M. A., 2014, Mon. Not. Roy. Astron. Soc., 440, 46
Aragon-Calvo M. A., 2018, preprint
Aragon-Calvo M. A., Jones B. J. T., van de Weygaert R., van der Hulst M. J., 2004, Astron. Astrophys., 474, 315
Aragon-Calvo M. A., van de Weygaert R., Jones B. J. T., 2010a, Monthly Notices of the Royal Astronomical Society, 408, 2163
Aragon-Calvo M. A., van de Weygaert R., Jones B. J. T., 2010b, Monthly Notices of the Royal Astronomical Society, 408, 2163
Bardeen J. M., Bond J. R., Kaiser N., Szalay A. S., 1986, ApJ, 304, 15
Bharadwaj S., Bhavsar S. P., Sheth J. V., 2004, ApJ, 606, 25
Bradford Barber C., Dobkin D. P., Huhdanpaa H., 1996, ACM TRANSACTIONS ON MATHEMATICAL SOFTWARE, 22, 469
Breiman L., 2001, Machine Learning, 45, 5
Carrasco Kind M., Brunner R. J., 2013, Mon. Not. Roy. Astron. Soc., 432, 1483
Cautun M., van de Weygaert R., Jones B. J. T., 2012, MNRAS, 429, 1286
Cautun M., van de Weygaert R., Jones B. J. T., Frenk C. S., 2014, Mon. Not. Roy. Astron. Soc., 441, 2923
Codis S., Pogosyan D., Pichon C., 2018, Mon. Not. Roy. Astron. Soc., 479, 973
Crocce M., Pueblas S., Scoccimarro R., 2006, MNRAS, 373, 369
Davis M., Efstathiou G., Frenk C. S., White S. D. M., 1985, ApJ, 292, 371
Ester M., Kriegel H., Sander J., Xu X., 1996, in Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, AAAI Press, pp 226-231
Falcó B., Neyrinck M. C., 2015, Monthly Notices of the Royal Astronomical Society, 450, 3239
Falcón B. L., Neyrinck M. C., Szalay A. S., 2012, ApJ, 754, 126
Fawcett T., 2006, Pattern Recogn. Lett., 27, 861
Forero-Romero J. E., Hoffman Y., Gottlöber S., Klypin A.,Yepes G., 2009, Mon. Not. Roy. Astron. Soc., 396, 1815
Foster C., Nelson L. A., 2009, ApJ, 699, 1252
Gillies S., et al., 2017, Shapely: manipulation and analysis of geometric objects
González R. E., Padilla N. D., 2010, MNRAS, 407, 1449
Green S. B., van den Bosch F. C., 2019, Mon. Not. Roy. Astron. Soc., 490, 2091
Hamaus N., Sutter P. M., Wandelt B. D., 2014, Phys. Rev. Lett., 112, 251302
Hansen S. M., McKay T. A., Wechsler R. H., Annis J., Sheldon E. S., Kimball A., 2005, Astrophys. J., 633, 122
Hermes D., 2017, The Journal of Open Source Software, 2, 267
Hoffman Y., Metuki O., Yepes G., Gottlöber S., Forero-Romero J. E., Libeskind N. I., Knebe A., 2012a, MNRAS, 425, 2049
Hoffman Y., Metuki O., Yepes G., Gottlöber S., Forero-Romero J. E., Libeskind N. I., Knebe A., 2012b, MNRAS, 425, 2049
Hui J., Aragon M., Cui X., Flegal J. M., 2018, MNRAS, 475, 4494
Hunter J. D., 2007, Computing in Science & Engineering, 9, 90
Katama McClish D., 1989, Medical Decision Making, 9, 190
Kitaura F., Angulo R. E., 2012, Monthly Notices of the Royal Astronomical Society, 425, 2443
Komatsu E., et al., 2011, ApJ, 192, 18
Kraljic K., et al., 2019, preprint
Libeskind N. I., et al., 2018, Mon. Not. Roy. Astron. Soc., 473, 1195
Lucie-Smith L., Pears H. V., Pontzen A., Lochner M., 2018, Mon. Not. Roy. Astron. Soc., 479, 3405
Mo H., van den Bosch F. C., White S., 2010, Galaxy Formation and Evolution
Murray S., Power C., Robotham A., 2013, preprint
Oliphant T., 2006, NumPy: A guide to NumPy, USA: Trelgol Publishing
Pereyra L. A., Sgró M. A., Merchán M. E., Stasyszyn F. A., Paz D. J., 2019, arXiv e-prints, p. arXiv:1911.06768
Ramachandra N. S., Shandarin S. F., 2015, Monthly Notices of the Royal Astronomical Society, 452, 1643
Rodriguez A. C., Kacprzak T., Lucchi A., Amara A., Sgier R., Fluri J., Hofmann T., Réfrégier A., 2018, Comput. Astrophys. Cosmol., 5, 4
Scoccimarro R., 1998, MNRAS, 299, 1097
Shandarin S., Habib S., Heitmann K., 2012, Phys. Rev. D, 85, 083005
Sousbie T., 2011, Monthly Notices of the Royal Astronomical Society, 414, 350
Springel V., 2005, MNRAS, 364, 1105
Springel V., et al., 2005, Nature, 435, 629
Tempel E., Stoica R. S., Kipper R., Saar E., 2016, Astron. Comput., 16, 17
Tipping M. E., Bishop C. M., 1998, Mixtures of Probabilistic Principal Component Analysers
Tsizh M., Novosyadlyj B., Holovatch Y., Libeskind N. I., 2019, preprint
Turk M. J., Smith B. D., Oishi J. S., Skory S., Skillman S. W., Abel T., Norman M. L., 2011, The Astrophysical Journal Supplement Series, 192, 9
Virtanen P., et al., 2019, arXiv e-prints, p. arXiv:1907.10121
Warren M. S., Abazajian K., Holz D. E., Teodoro L., 2006, ApJ, 646, 881
White S. D. M., Rees M. J., 1978, MNRAS, 183, 341
This paper has been typeset from a \TeX\/\LaTeX file prepared by the author.