Effective Automated Stellar Substructure Detection using the Supervised Neural Clustering Algorithm

Michael Huang

Under the direction of

Prof. Lina Necib
Professor of Astrophysics
Kavli Institute, MIT

Xiaowei Ou
Kavli Institute, MIT

Tri Nguyen
Kavli Institute, MIT

Research Science Institute
August 1, 2022

Abstract

Dwarf galaxies and globular clusters accreted by the Milky Way become tidally stripped stellar substructures. Identifying these stellar substructures is of tremendous interest for galactic archaeology and can facilitate dark matter detection experiments. To make progress towards effective automatic identification of stellar substructures, we propose the Supervised Neural Clustering (SNC) algorithm, which leverages an Edge-based Graph Convolutional Neural Network to learn to generate a co-association matrix of stars from the `FIRE` galactic simulation dataset. Then, another graph neural network is utilized to produce cluster assignments from the co-association matrix in an unsupervised fashion. We discuss the problem of evaluating clustering quality for the stellar substructure identification task and perform extensive evaluation and ablation study on our SNC algorithm. Our SNC algorithm is shown to outperform existing clustering algorithms for stellar substructure identification dramatically.

Summary

Galaxies grow in size by consuming other smaller dwarf galaxies. 95% of the stars in the outer halo of the Milky Way originate from a smaller dwarf galaxy that is consumed by the Milky Way. Identifying these dwarf galaxy remnants can help us understand the formation history of the Milky Way as well as aid dark matter detection experiments. Stars that originate from the same dwarf galaxy generally share similar values for conserved physical parameters. Using this property, one can identify remnants of dwarf galaxies by looking for clusters of stars with similar physical parameter values. We develop a machine learning based algorithm, called the Supervised Neural Clustering (SNC) algorithm to automatically identify clusters of stars that likely originate from the same dwarf galaxy. Our algorithm achieves state-of-the-art accuracy when tested on a computer simulated galaxy.

1 Introduction

Galaxies grow in a hierarchical manner through a series of merger events where a galaxy gravitationally attracts and accretes smaller dwarf galaxies [1, 10, 21]. The accreted dwarf galaxies form tidally disrupted stellar structures that constitute 95% of the Galactic halo of the Milky Way [37]. Identifying stellar substructures in the Milky Way is of tremendous interest because they encode the assembly history of the Milky Way [26]. Moreover, studying accreted dwarf galaxies can help us deduce the kinematics profile of dark matter particles traveling with these dwarf galaxies remnants [26, 41]. Because dark matter detection experiments are contingent on the velocity profiles of dark matter particles, the designs and calibrations of such experiments can be aided by the identification and study of accreted stellar substructures [41, 48].

The primary methods of stellar substructure detection are chemical-based [23, 34, 38, 49] and kinematics-based [15, 37, 43]. Stars from the same progenitor share similar chemical composition, making easily measurable chemical abundances such as metallicity and α-abundance useful identifiers of stellar substructures [22, 23]. A second method for stellar substructure identification leverages stars' Integrals of Motion (IoM) (*e.g.* energy, angular momentum, and action). IoM values can be computed from stars' 6D kinematics (3D position and 3D velocity) measurements. They are adiabatic invariants under the assumption of an axisymmetric, time-independent gravitational potential and low dynamical friction [6]; they are observed to be approximately conserved in the Milky Way [6]. Therefore, stellar substructures can be effectively detected by identifying dense clusters of stars in the chemical and integral of motion space.

Many chemical abundances and integrals of motions parameters can be considered for stellar substructure identification. The high dimensionality of the parameter space makes it difficult to manually identify clusters of data points in the parameter space. Therefore,

researchers have devoted much attention to applying clustering algorithms to automatically identify accreted stellar substructures [8, 9, 14, 20, 25, 27, 30, 35, 42]. However, utilizing clustering algorithms for accreted stellar substructure identification is an extraordinarily challenging task. Accreted stellar substructures are generally phase-mixed in the position space, velocity space, and even the integrals of motions spaces due to tidal disruption, dynamical friction, or the sheer amount of time that has passed since they are accreted into the Milky Way [9]. Accreted substructures are also not highly separable in the chemical abundance spaces. Brauer *et al.* recently surveyed the effectiveness of different clustering algorithms for stellar substructure identification, and found no effective clustering algorithm that can consistently recover stellar substructures [9].

We try to remedy this by proposing a Supervised Neural Clustering (SNC) algorithm for stellar substructure identification. Our SNC algorithm is able to effectively distinguish highly phase-mixed accreted substructures by learning to perform substructure clustering on the `FIRE` [28, 51] computer simulated galaxy.

In Section 2.1, we describe the `FIRE` simulation dataset we used to train our model. In Section 3.1, we provide the relevant background on existing clustering algorithms. In Section 3.2, we introduce the mathematical definitions and notations used in our SNC algorithm. In Section 4, we propose the Supervised Neural Clustering (SNC) algorithm for stellar substructure clustering. In Section 5, we discuss the challenges of evaluating stellar substructure clustering accuracy and propose a novel probabilistic based metric for to evaluate clustering algorithms' ability to recover true clusters. In Section 6, we experiment the SNC algorithm on computer simulated galaxies and compare its performance against existing clustering algorithms.

2 Galactic Simulation Data

2.1 FIRE Simulation Data

The main dataset used in this work is the *Latte* suite of FIRE-2 cosmological hydrodynamic simulations of Milky Way-like galaxies [28, 50, 51]; specifically, we focus on the m12i and m12f simulated galaxies. The dataset contains the 6D kinematics and chemical abundances of over $2 \cdot 10^5$ accreted stars in m12i and m12f. Each star is labeled with its progenitor dwarf galaxy by Ostdiek *et al.* [40]. These ground-truth cluster assignment labels allow us to consider the problem of stellar substructure clustering in a supervised setting and also enable us to evaluate the clustering quality of an algorithm by comparing the cluster assignment generated by the algorithm against the actual cluster label of each star. In this work, we use m12i to train our Supervised Neural Clustering (SNC) algorithm, and evaluate the algorithm on m12f.

2.2 Stellar Parameter Space

We describe the set of kinematic, orbital and chemical stellar parameters we select for stellar substructure clustering. Existing works on using cluster analysis to identify stellar substructures have been performed exclusively on either the integrals of motion variables [8, 20, 25, 27, 35], the positional variables [30], or the chemical abundances variables [7]. This is sensible because the three variable spaces are not physically well connected, and cannot be trivially combined for cluster analysis. Since our clustering approach leverages a neural network that is capable of learning complex transforms, we can consider all three variable spaces. Performing cluster analysis on a high dimensional parameter space is advantageous because the stellar substructures have high degrees of overlap in low dimensional parameter spaces, which makes them challenging to distinguish. The high overlap of stellar substructures in 2D space is illustrated by Figure 1.

Variable	Description
E	the specific orbital energy, defined as the total orbital energy divided by the mass of the star.
$\vec{L} = (L_x, L_y, L_z)$	the specific angular momentum of stars about the galactic center, defined as the angular momentum divided by the mass of the star.
$\vec{J} = (J_z, J_r, J_\phi)$	the orbital actions [24], with each coordinate representing a star's degree of motion along that coordinate. Note that since $J_\phi \approx L_z$, we don't include J_ϕ in practice.
ecc	the orbital eccentricity.
z	the z location of the star in the galactocentric coordinate.
$[\frac{\text{Fe}}{\text{H}}]$	the abundance of iron in the star, which approximately measures the metallicity of the star.
$[\frac{\text{Mg}}{\text{Fe}}]$	the abundance of magnesium in the star, which is approximately measures its α-abundance value.

Table 1: All the stellar parameters selected for cluster analysis in this study. E, $\vec{L}$, and $\vec{J}$ belong in the integral of motion space. ecc and z are stars' positional/orbital parameters. $[\frac{\text{Fe}}{\text{H}}]$ and $[\frac{\text{Mg}}{\text{Fe}}]$ constitute the chemical abundances space considered in this study. All stellar parameters chosen are considered as useful for stellar substructure identification in the literature [37].

3 Background and Preliminaries

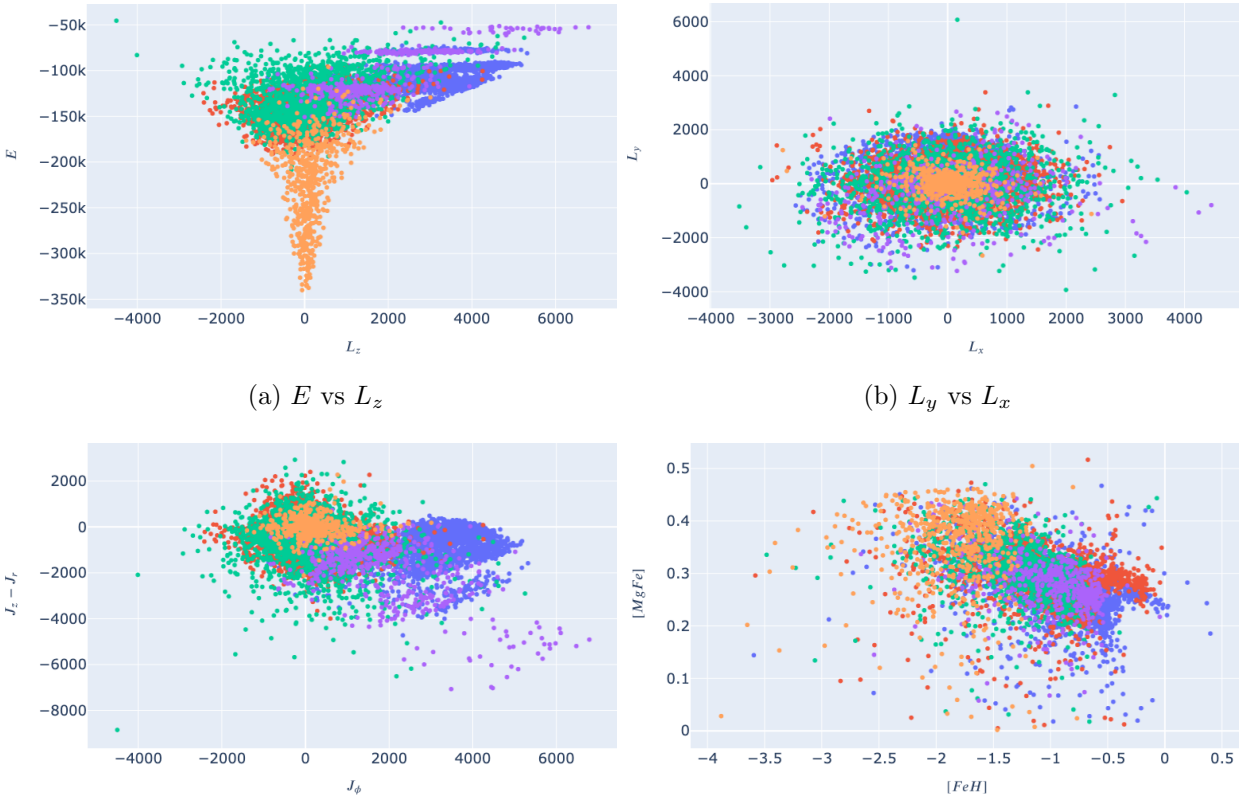
Before introducing our Supervised Neural Clustering (SNC) algorithm, we first provide background material on existing clustering algorithms as well as related works. Then, we define the mathematical notations used throughout this paper.

3.1 Clustering Algorithms Background

A clustering algorithm takes as input a set of data points and assigns each data point to a cluster. The concept of a cluster is defined vaguely as a set of data points sharing some similar characteristics, which may vary depending on the application's context.

3.1.1 Unsupervised Clustering vs Supervised Clustering

Unsupervised clustering algorithms assign data points to clusters based on some manually designed distance metric between data points. The metric or criteria an algorithm adopts for cluster assignment reflect the algorithm's definition of a cluster. Unsupervised



(a) E vs L_z

(b) L_y vs L_x

(c) $J_z - J_r$ vs J_ϕ

(d) $[MgFe]$ vs $[FeH]$

Figure 1: Scatter plot of the 5 largest accreted stellar substructures in `m12f`'s galactic halo in various parameter spaces. Each color represents a stellar substructure [40]. The stellar substructures are highly phase-mixed in all parameter spaces.

cluster analysis is a well-established field, with hundreds of algorithms proposed [52]. Many of these algorithms have clustering metrics and criteria designed for one particular application, limiting their generalizability [52]. Some general unsupervised clustering algorithms include K-Means [17], DBSCAN [16], HDBSCAN [12], Gaussian Mixture Model [45], and Spectral Clustering [46].

Supervised clustering algorithms learn the distance metric for assigning data points to clusters from a set of training data, thus voiding the need to design different metrics/criteria for specific use cases. Supervised clustering is an emerging field of research. Over the years, multiple supervised clustering algorithms have been developed as extensions of existing un-

supervised clustering. Examples include the supervised spectral clustering algorithm [2], supervised correlation clustering algorithm [18], and supervised k-means algorithm [19]. Hsu et al. [29] adopts a two step approach towards supervised clustering. In the first step, a trained Convolutional Neural Network (CNN) [39] is used to compute the pairwise probability of two data points belonging in the same cluster. Then, a second CNN is applied to generate the cluster assignments. Hsu et al. [29]'s work is, however, limited to semantic image clustering.

3.1.2 Hard Clustering vs Soft Clustering

Hard clustering algorithms assign each data point to one particular cluster. Most classic clustering algorithms, such as K-Means [17], DBSCAN [16], HDBSCAN [12], and Spectral Clustering [46] fall under the category of hard clustering algorithms.

Soft clustering algorithms do not assign each data point to a particular cluster. Instead, they compute the probability of a data point belonging to each cluster, which is advantageous in scenarios where a probabilistic clustering result is desired or when the true clusters overlap to such a degree that it does not make sense to assign a data point in the overlapping region to one cluster over another. Because stellar substructure clusters overlap to a high degree, as demonstrated by Figure 1, we argue that soft clustering is more suitable for the task of detecting stellar substructures.

Soft clustering algorithms include Fuzzy C-Means [4], Gaussian Mixture Model [45], and a line of recent works that use Graph Neural Networks to generate cluster assignments by optimizing for an unsupervised cost function [5, 44, 47].

Our proposed SNC algorithm follows closely along the line of work of using GNNs to generate soft cluster assignments but differ from prior work in that our SNC algorithm is supervised. The proposed SNC algorithm adopts a 2-step approach towards supervised clustering similar to the work of Hsu et al. [29]. Different from Hsu et al., the SNC algorithm

solves the general supervised clustering problem (as opposed to being focused on semantic image clustering), and leverage a GNN based architecture instead of the CNN based architecture used by Hsu et al. [29].

3.2 Mathematical Preliminaries

To describe the cluster assignment generated by a clustering algorithm, we use a an $N \times K$ cluster assignment matrix T, following the framework adopted by [5, 44, 47]. N denotes the number of data points in the dataset while K denotes the total number of clusters generated by the clustering algorithm. We use T_{ik} to denote the i^{th} row and k^{th} column of matrix T. For hard clustering algorithms, $T_{ik} = 1$ if the i^{th} data point is assigned to the k^{th} cluster and $T_{ik} = 0$ otherwise. For soft clustering algorithms, $T_{ik} =$ the probability that the i^{th} data point belongs in the k^{th} cluster. In all cases, each row of T sums to 1.

We use C to denote a length N vector that encodes the label of each data point's true cluster. $C_i =$ label of the i^{th} data point's cluster.

We use S to represent the $N \times N$ co-association matrix as defined by Bulo et al. [11]. S_{ij} represents the probability of data points i and j belonging in the same cluster.

Let X denote the feature matrix of all N data points. X has N rows and 10 columns (each corresponding to one of the 10 scalar-valued stellar parameters we have chosen in Section 2.2). Let X_{il} denote the i^{th} star's l^{th} feature.

In our proposed SNC algorithm, we model the dataset of N data points as a graph G, with each data point represented as a vertex. The proposed SNC algorithm is general purpose and does not place a constraint on how graph G is constructed from the data points. Each vertex of the graph is associated with a feature vector with dimension d. We use an $N \times d$ matrix V to denote all N vertices' features. In the context of a Graph Convolutional Neural (GCN) Network, we use V^t to denote the vertex features in the t^{th} layer of the GCN. Each

edge is associated with a d' dimension feature vector. We use a 3D tensor E to denote all edges' features. E_{ijl} denotes the l^{th} feature of edge (i, j). In the context of a GCN network, we use E^t to represent the edge features in the t^{th} layer of the GCN.

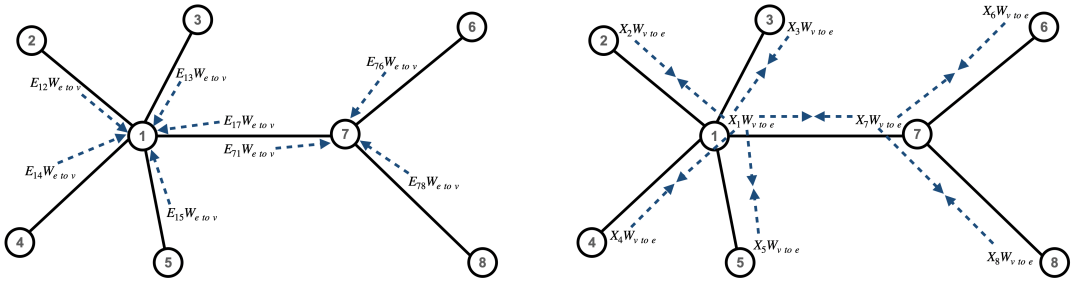
We use A to denote the adjacency matrix of graph G, where $A_{ij} = 1$ if vertices i, j are connected and $A_{ij} = 0$ otherwise. Let D be a $N \times N$ diagonal matrix where $D_{ii} = \deg(i)$, the degree of vertex i. The normalized adjacency matrix is defined as $\hat{A} = D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$ [33].

4 Supervised Neural Clustering

In this section, we describe our proposed Supervised Neural Clustering (SNC) algorithm for solving the general supervised clustering task. In the first step of the algorithm, we generate the co-association matrix S of all the data points using an Edge-based Graph Convolutional Neural Network (EGCN) architecture [31]. The EGCN is trained on the train dataset and then applied to the test dataset without extra training. In the second step, after computing a co-association matrix of data points in the test dataset, SNC leverages another graph neural network to generate a cluster assignment from that co-association matrix. Note that the second step of SNC is unsupervised while the first step is supervised.

4.1 Edge-based Graph Convolutional Neural Network

Traditional Graph Convolutional Neural Networks (GCNs) operate directly on the vertex features [33], which are the features of the input data points. This is non-ideal for the purpose of supervised clustering because the GCN tends to learn an absolute mapping from the input data points' parameter space to some feature space in which data points with the same true labels coalesce together. These absolute mapping can lead to extreme overfitting because the model could learn to draw specific decision boundaries in the input parameter space and map two data points to coalescing regions based on whether they fall on the same



(a) Vertex Convolution (b) Edge Convolution

Figure 2: Illustration of the Vertex Convolution and Edge Convolution operations used in our SNC architecture.

side of some non-general decision boundary in the input parameter space. To avoid the non-generality of absolute mapping, we choose an Edge-based Graph Convolutional Neural (EGCN) architecture [31] that focuses on the features of the edges. Our EGCN model utilizes the following update rules:

$$X_i^{t+1} = \text{ReLU}\left(X_i^t W_{v\text{ to }v}^{t+1} + \frac{1}{\deg(i)}\sum_j E_{ij}^t W_{e\text{ to }v}^{t+1}\right)$$

$$E_{ij}^{t+1} = \text{ReLU}\left(E_{ij}^t W_{e\text{ to }e}^{t+1} + [X_i, X_j] W_{v\text{ to }e}^{t+1}\right)$$

The first update rule represent a vertex convolution and the second update rule denotes an edge convolution; they can be visualized as shown in Figure 2. The $W_{v\text{ to }v}^t$, $W_{v\text{ to }e}^t$, $W_{e\text{ to }v}^t$, $W_{e\text{ to }e}^t$ matrices are learned weight matrices. $[..., ...]$ indicates a concatenation operation. Compared to Kearnes et al. [31]'s architecture, we adopt a more restrictive update rule for our EGCN — we use an addition instead of an additional linear map to combine the edge and node features. The more restrictive architecture helps reduce overfitting for our SNC model. We also make use of row-wise normalization [33] to ensure the vertex and edge features share a similar scale across layers of the EGCN.

4.2 Generating Co-association Matrix with EGCN

To produce the co-association matrix, we use an EGCN with two layers and a feature dimension of 32 for both vertex features and edge features. The final edge classification is produced from the E^2 features via $F(i,j) = \sigma\left(E^2_{ij} \cdot W_{\text{classify}}\right)$. The final output of the network is the co-association matrix S with $S_{ij} = F(i,j)$.

We initialize the input vertex features of the EGCN with $V^0 = \mathbf{0}$ and input edge features with $E^0_{ijl} = |X_{il} - X_{jl}|$.

The EGCN-based model is trained end-to-end with the standard binary crossentropy classification loss, which is defined as

$$L = -\sum_{C_i = C_j} \log(S_{ij}) - \sum_{C_i \neq C_j} \log(1 - S_{ij}).$$

4.3 Cluster Generation from Co-association Matrix

As the second step of the SNC algorithm, we take as input the co-association matrix S generated by the EGCN-based model described Section 4.1 and 4.2, and produce the cluster assignment matrix T based on the co-association matrix. Note that this step is performed in a completely unsupervised setting. The algorithm we propose for this task extend the work pioneered by Bulo et al. [11]. Bulo et al. noted that it follows from the definition of the soft cluster assignment matrix T that $\sum_{0 \leq k < K} T_{ik} T_{jk} = P(C_i = C_j \mid T)$ [11]. From there, Bulo et al. [11] proposed to find the optimal T by minimizing the L2 distance between $P(C_i = C_j \mid T)$ and S_{ij}. The cost function they proposed is

$$L = \sum_{\text{all } (i,j)} \left(S_{ij} - \sum_{0 \leq k < K} T_{ik} T_{jk} \right)^2.$$

Directly minimizing L proves to be ineffective for scenarios with large disparities in cluster

sizes, which is the case for stellar substructure clusters. In these scenarios, the generated T aggressively assigns most data points to a small number of clusters, leaving the rest of the clusters "dead." We propose a novel regularization constraint $L_{\text{regularize}}$ to mitigate dead clusters.

$$L_{\text{regularize}} = -\sum_{0\leq k<K} \log\left(1 - \exp\left(\sum_{0\leq i<N} \log\left(1 - T_{ik}^2\right)\right)\right). \tag{1}$$

Thus, the final loss function is $L_{\text{total}} = L + \alpha L_{\text{regularize}}$, where α is a manually chosen parameter. See Appendix A for a probabilistic justification for this regularization loss function.

Due to the additional regularization, the mathematical approach Bulo et al. proposed for finding the optimal T that minimizes L cannot be applied [11]. Instead of optimizing for T directly, we propose to use a generative graph neural network to generate T, and optimize for the parameters of the GNN instead. This is a promising solution because it follows the approach introduced by Shchur and Günnemann [44] to minimize a similar cost function for unsupervised clustering. Shchur and Günnemann [44] found that optimizing the parameters of a generative GNN produces better clustering results than directly optimizing for T.

We adopt a single-layer, 64-channel conventional GCN [33] followed by a K-class classification head as our GNN-based model to generate T. The input to the GCN-based classifier is simply $V^0 = X$. Mathematically, the output $T = \text{softmax}\left(\text{GCN}(X)W_{\text{classify}}\right)$. The model is trained on the test dataset in an unsupervised manner using the cost function L_{total}. The cluster assignment generated by this GCN is the final output of the SNC algorithm.

5 Cluster Evaluation for Overlapping Clusters

Before presenting the experimental results of our SNC algorithm, we discuss the challenges involved in evaluating the quality of cluster assignment in the setting of highly over-

lapping clusters and more specifically stellar substructure detection. To evaluate the quality of a cluster assignment, we focus on the cluster assignment's ability to recover real clusters, because we are interested in the algorithm's ability to recover and identify stellar substructures. For this reason, we use **Intersection over Union (IoU) Precision and Recall**, which is a direct generalization of the Recovery Rate metric used by Brauer et al. [9]. For IoU Precision and Recall, a true cluster A is considered to be correctly recovered by a generated cluster B if $\frac{|A \cap B|}{|A \cup B|} \geq$ threshold, where threshold ≥ 0.5. The precision and recall follow the standard definition as $\frac{\#\text{correctly recovered clusters}}{\#\text{generated clusters}}$ and $\frac{\#\text{correctly recovered clusters}}{\#\text{true clusters}}$.

However, simple cluster recovery metrics such as the IoU metric cannot effectively deal with highly overlapping clusters, which is the case for stellar substructure clusters, as illustrated by Figure 1. To demonstrate the point, consider the toy dataset in Figure 3, where true cluster A_2 overlaps with A_1. Even if the clustering algorithm returns the correct cluster detection results, as represented by ellipses B_1 and B_2, the IoU metric would consider (A_2, B_2) as a case of failed cluster recovery because $|A_1 \cap B_2| > |A_2 \cap B_2|$, which means $\frac{|A_2 \cap B_2|}{|B_2|} < 0.5$, and thus $\frac{|A_2 \cap B_2|}{|A_2 \cup B_2|} < 0.5$.

To address the challenge of evaluating cluster recovery rate in a situation with highly overlapping clusters, we propose a novel precision and recall metric for soft clustering algorithms. We name them, the **Soft Precision and Recall** metrics. Instead of counting the overlap of data points between a true cluster and a generated cluster, we consider the "matchedness" of a true cluster and a generated cluster in a probabilistic manner. We define the matchedness of a true cluster A and generated cluster B to be the log-likelihood of generating A from B by sampling data points based on their probability of belonging to cluster B. Recall that T_{iB} represents the probability of the i^{th} data point belonging to cluster B. The log-likelihood M_{AB} can be computed by

$$M_{AB} = \sum_{i \in A} \log(T_{iB}) + \sum_{i \notin A} \log(1 - T_{iB}) \tag{2}$$

Then, A is considered to be correctly recovered by B if $M_{AB} = \max_{B'}(M_{AB'})$ and $M_{AB} = \max_{A'}(M_{A'B})$. In other words, A is the true cluster that is most likely to be sampled from B among all true clusters; B is the generated cluster from which A is most likely to be sampled, among all generated clusters from which A could be sampled.

The Soft Precision and Recall metrics can also be used to evaluate hard clustering algorithms, in which case we can consider $M_{AB} \propto |A \cap B| - |A \cup B|$. See Appendix B for a proof of this result.

Finally, we also use the **Adjusted Mutual Information (AMI)** as an auxiliary metric because AMI performs well in the context of overlapping clusters [36].

6 Experiment and Discussion

6.1 Training Details

We train our Supervised Neural Clustering algorithm on simulated galaxy `m12i` and evaluate its performance on `m12f`. To construct the graph G used in the SNC algorithm, we simply connect every single data point to every other data point to form a complete graph. Note that structural information of the dataset (*i.e.* the distance between data points) is preserved through the edge features fed into the EGCN. Unfortunately, this method of graph construction imposes a huge limit on the computational efficiency of the SNC algorithm. Under this graph construction method, the SNC algorithm can only process datasets of size up to ~ 1000 or ~ 10000. Since the clustering quality is not expected to depend on the size of the dataset, the SNC algorithm is trained and tested by sampling 1000 data points from `m12i` and `m12f`. We first train the EGCN edge classifier on `m12i` for 1500 epochs using

the Adam optimizer [32] with a learning rate of 0.01 and weight decay of 10^{-5}. The trained EGCN is then used to produce a co-association matrix on `m12f`. We train the unsupervised cluster assignment generator on the co-association until the loss converge. We consider the K (maximum number of clusters) and α (regularization coefficient) parameters used in the unsupervised cluster assignment generator as hyperparameters of our SNC clustering algorithm.

6.2 Baselines

We compare SNC against Gaussian Mixture Model (GMM) [13] and HDBSCAN [12]. Note that HDBSCAN had been identified by [9] as the most effective clustering algorithm for stellar substructure identification. GMM only accept one hyperparameter — the maximal number of expected clusters. HDBSCAN has two significant hyperparameters, `min_cluster_size` and `min_samples`. Both baseline algorithms are run on 1000 data points samples from `m12f` for fairness of comparison.

6.3 Hyperparameter Selection

Note that in actual application settings, the hyperparameters of clustering algorithms are tuned on the Milky Way data against a catalog of already discovered stellar substructures and are also tuned based on researchers' qualitative assessments of the algorithm's clustering quality. For this reason, we allow the hyperparameters of all clustering algorithms to be tuned on the test dataset `m12f` for fair comparison similar to the approach taken by Brauer et al. [9]. To fairly compare our algorithms, we select hyperparameters that achieve strong performance while maintaining a relative balance between the precision and recall values. For HDBSCAN, we use `min_cluster_size` $= 2$ and `min_samples` $= 1$. For GMM, we choose $K = 30$. For SNC, we find that setting $K = 30$, $\alpha = 10^{-5}$ produces balanced results.

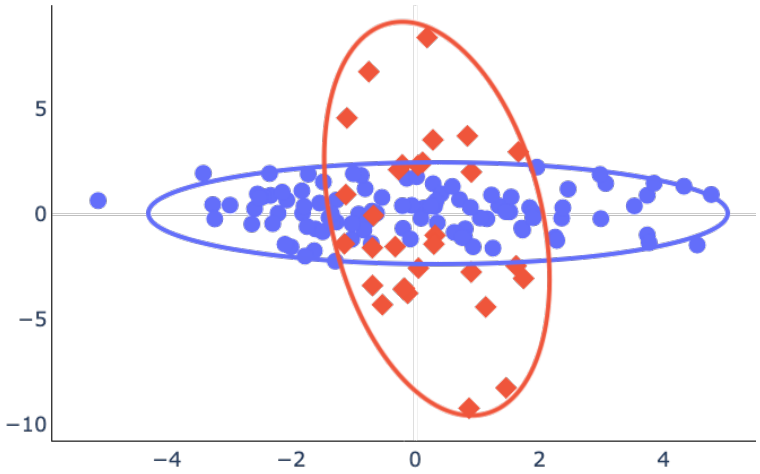


Figure 3: Hypothetical clustering result on a toy dataset. Blue dots are data points in real cluster A_1. Red dots are data points in real cluster A_2. Blue ellipse corresponds to the region covered by the generated cluster B_1 and the red ellipse correspond to the generated cluster B_2.

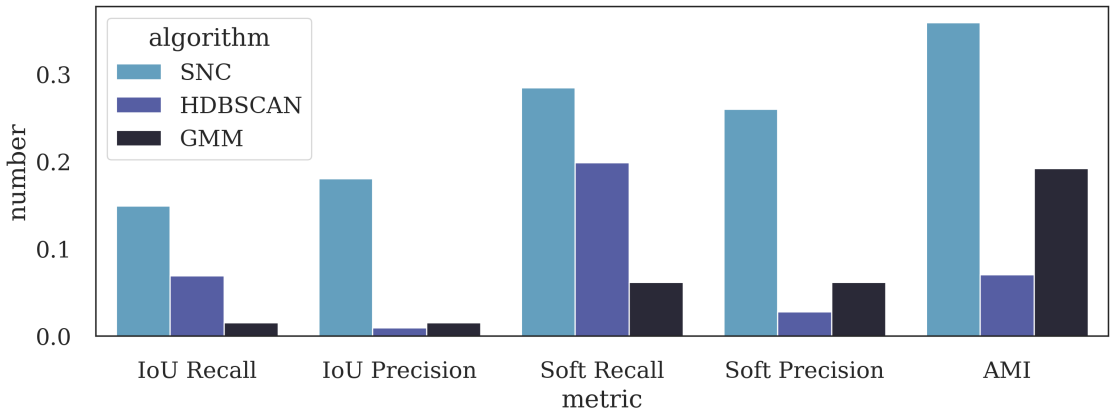


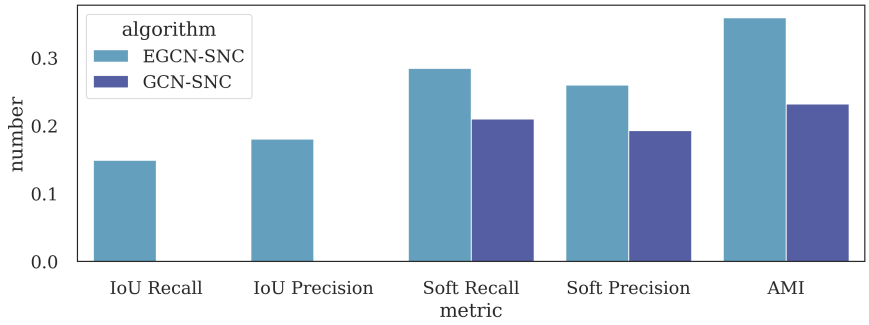
Figure 4: Comparison of performances of different stellar substructure clustering algorithms. Our proposed SNC algorithm outperforms all other algorithms by large margins and across all metrics.

6.4 Performance Comparison

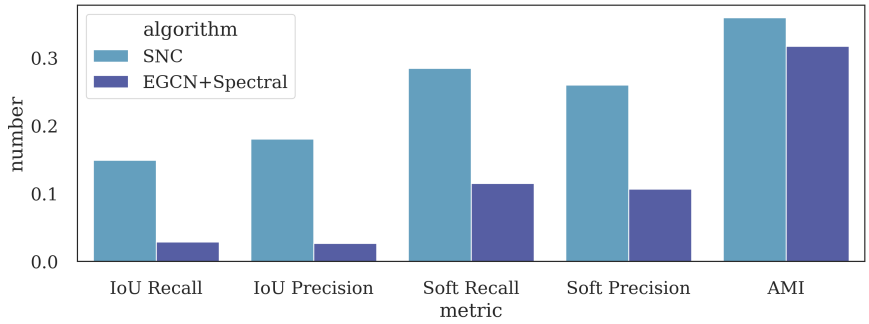
Figure 4 demonstrates effectiveness of our proposed SNC algorithm. SNC outperforms HDBSCAN and GMM on all metrics dramatically. Notably, the SNC algorithm achieves over 2x the IoU recall rate of HDBSCAN while attaining 17x higher IoU precision compared to

HDBSCAN.

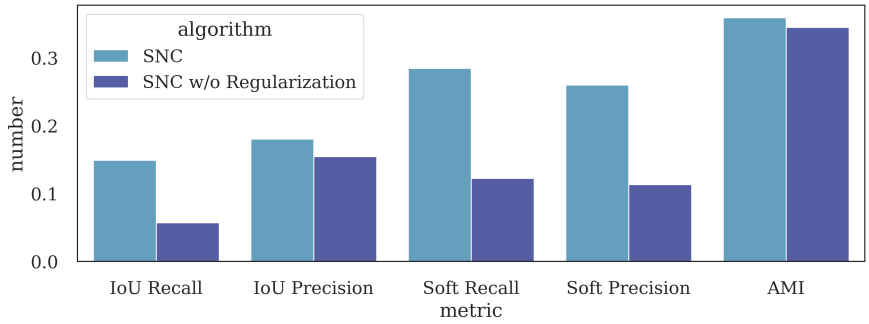
6.5 Ablation Study



(a) Effectiveness of EGCN



(b) Effectiveness of GNN-based Cluster Assignment Generator



(c) Effectiveness of our proposed regularization in the GNN-based Cluster Assignment Generator

Figure 5: Performance comparison between SNC and its ablated versions.

Effectiveness of EGCN

We replace the EGCN with a 2-layer 32-channel standard GCN [33] with the same classification head. Instead of feeding edge features, $E_{ijl} = |X_{il} - X_{jl}|$ into the ECGN, we feed X_{il} directly into each node of the GCN. For the reasons outlined in Section 4.1, we observe the GCN model is significantly less generalizable, as demonstrated by its low performance in Figure 5a.

Effectiveness of GNN-based Cluster Assignment Generator

The EGCN computes the co-association matrix, which can be directly fed into a Spectral Clustering algorithm to generate clusters with optimal normalized cuts [46]. It is, therefore, natural to ask whether the complicated GNN-based Cluster Assignment Generator is necessary to attain high clustering performance. To test this hypothesis, we apply the Spectral Clustering algorithm to the outputs of the EGCN and compare the results against those produced by the SNC. Figure 5b demonstrates that the EGCN + Spectral Clustering combination falls short of the performance of SNC over all metrics, thereby verifying the effectiveness and importance of the GNN-based Cluster Assignment Generator module.

Effectiveness of our proposed regularization

Figure 5c compares the performance of SNC with regularization and without regularization. The unregularized SNC algorithm generates a high number of dead clusters (clusters with no data points assigned to them), costing a significant drop in recall rates. SNC clearly achieves significantly more balanced and accurate clustering results.

7 Limitations and Future Work

While the SNC algorithm achieves astounding improvements in clustering quality, the SNC algorithm is computationally expensive because we construct the graph used by the SNC algorithm by directly connecting every pair of data points. Therefore, a potential direction

of future work is to explore the effectiveness of SNC on a smaller, incomplete graph. For example, a kd-tree [3] based K-Nearest Neighbor search could be used to quickly generate a graph by connecting each data point with its K-Nearest Neighbors.

8 Conclusion

To make progress towards effective automated stellar substructure identification, we considered the problem of stellar substructure clustering as a supervised clustering problem. We proposed the Supervised Neural Clustering (SNC) algorithm for stellar substructure identification. Then, we undertake an extensive discussion of the challenges of evaluating clustering quality on a dataset with overlapping clusters. We propose a probabilistic evaluation metric to address those challenges. Finally, we perform extensive experimentation and ablation study on the SNC algorithm, demonstrating the effectiveness of our algorithm against existing clustering algorithms on the task of stellar substructure identification.

9 Practical Takeaways

The central result of this research is a machine learning based computer software that can automatically analyze data of stars in the Milky Way and identify clusters of stars that could correspond to prospective previously-unknown dwarf galaxies or globular clusters accreted by the Milky Way. The identification of these stellar substructures can be used to determine the merger history of the Milky Way and aid the design of dark matter detection experiments.

The SNC algorithm's application is not limited to stellar substructure identification. It is a general-purpose supervised clustering algorithm. To the best of the author's knowledge, the SNC algorithm has the following novelties:

1. It is the first graph neural network based algorithm used for supervised clustering.

2. It is the first Edge-based Graph Convolutional Neural Network (EGCN) architecture proposed for the general clustering task.

3. It pioneers a two-step approach towards supervised clustering where in the first step predicts a co-association matrix and the second step generates clustering based on to co-association matrix.

Beyond our SNC algorithm, we proposed a statistically interpretable metric for soft clustering that can deal with situations with overlapping clusters.

10 Acknowledgments

I would like to thank Prof. Lina Necib for her illuminating insights and mentorship, Xiaowei Ou, Tri Nguyen, Cian Roche, and Nora Shipp for their untiring support and guidance. I would also like to thank Dr. John Rickert and Peter Gaydarov for their mathematical instructions. I would like to acknowledge insightful conversations with Caleb Oh, Catherine Xue, and Ali Yang. Finally, I would like to acknowledge RSI, CEE, and MIT for supporting this project and making it possible. I would like to thank Mr. Jay Ding, Professor Anping Hou, Ms. Liyuan Wu, and Mr. Songrong Hou for their generous sponsorship.

References

[1] Mario G Abadi, Julio F Navarro, and Matthias Steinmetz. Stars beyond galaxies: the origin of extended luminous haloes around galaxies. *Monthly Notices of the Royal Astronomical Society*, 365(3):747–758, 2006.

[2] Francis Bach and Michael Jordan. Learning spectral clustering. *Advances in neural information processing systems*, 16, 2003.

[3] Jon Louis Bentley. Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18(9):509–517, 1975.

[4] James C Bezdek, Robert Ehrlich, and William Full. Fcm: The fuzzy c-means clustering algorithm. *Computers & geosciences*, 10(2-3):191–203, 1984.

[5] Filippo Maria Bianchi, Daniele Grattarola, and Cesare Alippi. Spectral clustering with graph neural networks for graph pooling. In *International Conference on Machine Learning*, pages 874–883. PMLR, 2020.

[6] James Binney and Scott Tremaine. *Galactic dynamics*, volume 13. Princeton university press, 2011.

[7] Joss Bland-Hawthorn, Torgny Karlsson, Sanjib Sharma, Mark Krumholz, and Joe Silk. The chemical signatures of the first star clusters in the universe. *The Astrophysical Journal*, 721(1):582, 2010.

[8] Nicholas W Borsato, Sarah L Martell, and Jeffrey D Simpson. Identifying stellar streams in gaia dr2 with data mining techniques. *Monthly Notices of the Royal Astronomical Society*, 492(1):1370–1384, 2020.

[9] Kaley Brauer, Hillary D Andales, Alexander P Ji, Anna Frebel, Mohammad K Mardini, Facundo A Gomez, and Brian W O'Shea. Possibilities and limitations of kinematically identifying stars from accreted ultra-faint dwarf galaxies. *arXiv preprint arXiv:2206.07057*, 2022.

[10] James S Bullock and Kathryn V Johnston. Tracing galaxy formation with stellar halos. i. methods. *The Astrophysical Journal*, 635(2):931, 2005.

[11] Samuel Rota Bulo, André Lourenço, Ana Fred, and Marcello Pelillo. Pairwise probabilistic clustering using evidence accumulation. In *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*, pages 395–404. Springer, 2010.

[12] Ricardo JGB Campello, Davoud Moulavi, and Jörg Sander. Density-based clustering based on hierarchical density estimates. In *Pacific-Asia conference on knowledge discovery and data mining*, pages 160–172. Springer, 2013.

[13] Richard O Duda, Peter E Hart, and David G Stork. *Pattern classification and scene analysis*, volume 3. Wiley New York, 1973.

[14] George Efstathiou, Carlos S Frenk, Simon DM White, and Marc Davis. Gravitational clustering from scale-free initial conditions. *Monthly Notices of the Royal Astronomical Society*, 235:715–748, 1988.

[15] OJ Eggen, D Lynden-Bell, and AR Sandage. Evidence from the motions of old stars that the galaxy collapsed. *The Astrophysical Journal*, 136:748, 1962.

[16] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd*, volume 96, pages 226–231, 1996.

[17] Brian S. Everitt, Sabine Landau, and Morven Leese. *Cluster Analysis*. Wiley Publishing, 4th edition, 2009. ISBN 0340761199.

[18] Thomas Finley and Thorsten Joachims. Supervised clustering with support vector machines. In *Proceedings of the 22nd international conference on Machine learning*, pages 217–224, 2005.

[19] Thomas Finley and Thorsten Joachims. Supervised k-means clustering. Technical report, 2008.

[20] Paola Re Fiorentin, Alessandro Spagna, Mario G Lattanzi, and Michele Cignoni. Icarus: A flat and fast prograde stellar stream in the milky way disk. *The Astrophysical Journal Letters*, 907(1):L16, 2021.

[21] Andreea S Font, Kathryn V Johnston, James S Bullock, and Brant E Robertson. Phase-space distributions of chemical abundances in milky way-type galaxy halos. *The Astrophysical Journal*, 646(2):886, 2006.

[22] Anna Frebel and John E. Norris. Near-Field Cosmology with Extremely Metal-Poor Stars. 53:631–688, August 2015. doi: 10.1146/annurev-astro-082214-122423.

[23] Ken Freeman and Joss Bland-Hawthorn. The new galaxy: signatures of its formation. *arXiv preprint astro-ph/0208106*, 2002.

[24] Herbert Goldstein, Charles Poole, and John Safko. Classical mechanics, 2002.

[25] Dmitrii Gudin, Derek Shank, Timothy C Beers, Zhen Yuan, Guilherme Limberg, Ian U Roederer, Vinicius Placco, Erika M Holmbeck, Sarah Dietz, Kaitlin C Rasmussen, et al. The r-process alliance: Chemodynamically tagged groups of halo r-process-enhanced stars reveal a shared chemical-evolution history. *The Astrophysical Journal*, 908(1):79, 2021.

[26] Amina Helmi. Streams, substructures and the early history of the milky way. *arXiv preprint arXiv:2002.04340*, 2020.

[27] Amina Helmi and P Tim de Zeeuw. Mapping the substructure in the galactic halo with the next generation of astrometric satellites. *Monthly Notices of the Royal Astronomical Society*, 319(3):657–665, 2000.

[28] Philip F Hopkins, Andrew Wetzel, Dušan Kereš, Claude-André Faucher-Giguère, Eliot Quataert, Michael Boylan-Kolchin, Norman Murray, Christopher C Hayward, Shea Garrison-Kimmel, Cameron Hummels, et al. Fire-2 simulations: physics versus numerics in galaxy formation. *Monthly Notices of the Royal Astronomical Society*, 480(1): 800–863, 2018.

[29] Yen-Chang Hsu, Zhaoyang Lv, and Zsolt Kira. Learning to cluster in order to transfer across domains and tasks. *arXiv preprint arXiv:1711.10125*, 2017.

[30] Emily L Hunt and Sabine Reffert. Improving the open cluster census-i. comparison of clustering algorithms applied to gaia dr2 data. *Astronomy & Astrophysics*, 646:A104, 2021.

[31] Steven Kearnes, Kevin McCloskey, Marc Berndl, Vijay Pande, and Patrick Riley. Molecular graph convolutions: moving beyond fingerprints. *Journal of computer-aided molecular design*, 30(8):595–608, 2016.

[32] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[33] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks, 2016. URL `https://arxiv.org/abs/1609.02907`.

[34] Duane M Lee, Kathryn V Johnston, Bodhisattva Sen, and Will Jessop. Reconstructing the accretion history of the galactic stellar halo from chemical abundance ratio distributions. *The Astrophysical Journal*, 802(1):48, 2015.

[35] S Sofie Lövdal, Tomás Ruiz-Lara, Helmer H Koppelman, Tadafumi Matsuno, Emma Dodd, and Amina Helmi. Substructure in the stellar halo near the sun. i. data-driven clustering in integrals of motion space. *arXiv preprint arXiv:2201.02404*, 2022.

[36] Aaron F. McDaid, Derek Greene, and Neil Hurley. Normalized mutual information to evaluate overlapping community finding algorithms, 2011. URL `https://arxiv.org/abs/1110.2515`.

[37] Rohan P Naidu, Charlie Conroy, Ana Bonaca, Benjamin D Johnson, Yuan-Sen Ting, Nelson Caldwell, Dennis Zaritsky, and Phillip A Cargile. Evidence from the h3 survey that the stellar halo is entirely comprised of substructure. *The Astrophysical Journal*, 901(1):48, 2020.

[38] Heidi Jo Newberg, Jeffrey L Carlin, et al. Tidal streams in the local group and beyond. *Astrophysics and Space Science Library*, 420, 2016.

[39] Keiron O'Shea and Ryan Nash. An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458*, 2015.

[40] B. Ostdiek, L. Necib, T. Cohen, M. Freytsis, M. Lisanti, S. Garrison-Kimmmel, A. Wetzel, R. E. Sanderson, and P. F. Hopkins. Cataloging accreted stars within gaia dr 2 using deep learning. *Astronomy Astrophysics*, 636:A75, apr 2020. doi: 10.1051/0004-6361/201936866. URL `https://doi.org/10.1051%2F0004-6361%2F201936866`.

[41] Ciaran AJ O'Hare, N Wyn Evans, Christopher McCabe, GyuChul Myeong, and Vasily Belokurov. Velocity substructure from gaia and direct searches for dark matter. *Physical Review D*, 101(2):023006, 2020.

[42] Tomás Ruiz-Lara, Tadafumi Matsuno, S Sofie Lövdal, Amina Helmi, Emma Dodd, and Helmer H Koppelman. Substructure in the stellar halo near the sun. ii. characterisation of independent structures. *arXiv preprint arXiv:2201.02405*, 2022.

[43] Leonard Searle and Robert Zinn. Compositions of halo clusters and the formation of the galactic halo. *The Astrophysical Journal*, 225:357–379, 1978.

[44] Oleksandr Shchur and Stephan Günnemann. Overlapping community detection with graph neural networks. *arXiv preprint arXiv:1909.12201*, 2019.

[45] Noam Shental, Aharon Bar-Hillel, Tomer Hertz, and Daphna Weinshall. Computing gaussian mixture models with em using equivalence constraints. *Advances in neural information processing systems*, 16, 2003.

[46] Jianbo Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000. doi: 10.1109/34.868688.

[47] Anton Tsitsulin, John Palowitch, Bryan Perozzi, and Emmanuel Müller. Graph clustering with graph neural networks. *arXiv preprint arXiv:2006.16904*, 2020.

[48] Teresa Marrodan Undagoitia and Ludwig Rauch. Dark matter direct-detection experiments. *Journal of Physics G: Nuclear and Particle Physics*, 43(1):013001, 2015.

[49] Kim A Venn, Mike Irwin, Matthew D Shetrone, Christopher A Tout, Vanessa Hill, and Eline Tolstoy. Stellar chemical signatures and hierarchical galaxy formation. *The Astronomical Journal*, 128(3):1177, 2004.

[50] Andrew Wetzel, Christopher C Hayward, Robyn E Sanderson, Xiangcheng Ma, Daniel Angles-Alcazar, Robert Feldmann, TK Chan, Kareem El-Badry, Coral Wheeler, Shea Garrison-Kimmel, et al. Public data release of the fire-2 cosmological zoom-in simulations of galaxy formation. *arXiv preprint arXiv:2202.06969*, 2022.

[51] Andrew R Wetzel, Philip F Hopkins, Ji-hoon Kim, Claude-André Faucher-Giguère, Dušan Kereš, and Eliot Quataert. Reconciling dwarf galaxies with λcdm cosmology: simulating a realistic population of satellites around a milky way–mass galaxy. *The Astrophysical Journal Letters*, 827(2):L23, 2016.

[52] Dongkuan Xu and Yingjie Tian. A comprehensive survey of clustering algorithms. *Annals of Data Science*, 2(2):165–193, 2015.

A Justification for the proposed regularization loss

Intuitively, to prevent "dead clusters" is to ensure every cluster consistent gets at least one data point assigned to it across multiple samplings of hard cluster assignments from T. To simplify, consider sampling a hard cluster assignment from T twice in a row. We formally define a cluster k to be dead if does not receive at least one data point assigned to it in both of the two samplings. Then, cluster k is dead with probability

$$\begin{aligned} P(k \text{ is dead}) &= \prod_{0 \le i < N} \left(1 - T_{ik}^2\right) \\ &= \exp\left(\sum_{0 \le i < N} \log\left(1 - T_{ik}^2\right)\right) \end{aligned}$$

We propose to penalize the negative log-likelihood that none of the K clusters is dead. The negative log-likelihood can be computed as

$$\begin{aligned} L_{\text{regularize}} &= -\log\left(\prod_{0 \le k < K} \left(1 - P(k \text{ is dead})\right)\right) \\ &= -\sum_{0 \le k < K} \log\left(1 - \exp\left(\sum_{0 \le i < N} \log\left(1 - T_{ik}^2\right)\right)\right), \end{aligned}$$

which is equivalent to Equation 1 and is used as our regularization loss.

B Extending the Soft Precision and Recall metrics to evaluate hard cluster assignments

The Soft Precision and Recall metrics we proposed can be extended to evaluate hard cluster assignment via the proportionality $M_{AB} \propto |A \cap B| - |A \cup B|$. We provide a brief justification for this result here.

Notice that M_{AB} is actually undefined for a hard cluster assignment matrix T where $T_{iB} = 0$ if $i \notin B$ and $T_{iB} = 1$ if $i \in B$. However, we can define a limiting version of T where $T_{iB} = \lambda$ if $i \notin B$ and $T_{iB} = 1 - (K-1)\lambda$. In this case, Equation 2 can be reduced to

$$\begin{aligned} M_{AB} = &|A \setminus B| \cdot \log(\lambda) + |A \cap B| \log(1 - (K-1)\lambda) \\ &+ |B \setminus A| \cdot \log\left(1 - (1 - (K-1)\lambda)\right) + |U \setminus (A \cup B)| \cdot \log\left(1 - \lambda\right), \end{aligned}$$

where $\setminus$ denotes the set subtraction operation and U denotes the universal set. Once we take the limit

$$\begin{aligned} \lim_{\lambda \to 0} M_{AB} &= |A \setminus B| \cdot \log(\lambda) + |B \setminus A| \cdot \log(\lambda) \\ &= (|A \cup B| - |A \cap B|) \cdot \log(\lambda), \end{aligned}$$

which means $M_{AB} \propto |A \cap B| - |A \cup B|$ in the limiting case where $\lambda \to 0$.