

MASTERARBEIT | MASTER'S THESIS

Titel | Title

DeNSe: A Deep Semi-Analytic Clustering Approach for
Identifying Nearby Stellar Populations in Gaia DR3

verfasst von | submitted by
Nora Wagner BSc MSc

angestrebter akademischer Grad | in partial fulfilment of the requirements for the degree of
Master of Science (MSc)

Wien | Vienna, 2025

Studienkennzahl lt. Studienblatt | Degree
programme code as it appears on the
student record sheet:

UA 066 910

Studienrichtung lt. Studienblatt | Degree
programme as it appears on the student
record sheet:

Masterstudium Computational Science

Betreut von | Supervisor:

Univ.-Prof. Joao Alves PhD

Mitbetreut von | Co-Supervisor:

DI Dr.techn. Sebastian Ratzenböck BSc

Acknowledgements

First and foremost, I would like to express my deepest gratitude to my co-supervisor, Sebastian Ratzenböck, for his continuous support and invaluable guidance throughout this thesis. Our many discussions and brainstorming sessions were crucial in shaping the direction of this work. His patience, openness to new ideas, and willingness to dedicate his time to help me navigate challenges have been immensely appreciated.

I am also sincerely thankful to João Alves for supervising me during this thesis and for welcoming me into his research group. Being part of his team has been an inspiring experience that allowed me to gain first-hand insight into astronomical research within a dynamic, creative, and supportive environment. His vision and enthusiasm for the field have been truly motivating.

Furthermore, I would like to extend my heartfelt gratitude to my parents, whose unwavering belief in me has been the foundation of my academic and personal journey. Their constant encouragement, patience, and unconditional support have provided me with strength and inspiration at every stage of this work.

Finally, I also would like to thank my colleagues in the research group for their stimulating discussions and shared moments that made this experience enjoyable.

Abstract

Stars form in clustered environments, sharing similar initial positions and velocities. As these systems evolve, many dissolve into the Galactic field, erasing most direct evidence of their common origin. Recovering such coeval groups is essential for reconstructing recent star formation history. The *Gaia* mission has enabled the discovery of thousands of new clusters by applying density-based algorithms to the five-parameter astrometry, but in the immediate Solar neighborhood strong projection effects blur tangential-velocity overdensities. Clustering in six-dimensional Galactic Cartesian phase space would avoid this problem, yet the transformation requires radial velocities, which are only available for a minority of *Gaia* sources.

This thesis introduces DENSE (Deep Semi-analytic Embedding of the Nearby Stellar Environment), a semi-analytic neural network model that learns an approximate 6D Galactic Cartesian representation from *Gaia*-like astrometry with missing radial velocities. A neural encoder maps the 5D observables to a latent space interpreted as $(XYZUVW)$, while an analytic decoder performs the exact coordinate transformations back to astrometric quantities and radial velocity. The model is trained to reconstruct the observables and preserve relative velocities, effectively inferring the missing line-of-sight components. DENSE is evaluated on two synthetic *Gaia*-like data sets with several clusters embedded in a uniform background. For a simpler configuration, it recovers positions and velocities with sub-parsec and few km s^{-1} accuracy, and latent-space HDBSCAN clustering retrieves all input groups with a normalized mutual information (NMI) of 1.0, matching clustering in the true 6D space and outperforming 5D baselines. For a more complex sample with more dispersed, overlapping structures, DENSE still reaches NMI values of ~ 0.6 – 0.68 , roughly doubling the 5D score. These results show that semi-analytic neural methods such as DENSE can recover informative phase-space embeddings and significantly improve the detection of nearby stellar associations, even when radial velocities are largely unavailable.

Kurzfassung

Sterne entstehen in Gruppen, die zu Beginn ähnliche Positionen und Geschwindigkeiten aufweisen. Im Laufe der Zeit lösen sich viele dieser Systeme auf und vermischen sich mit der Feldpopulation der Milchstraße, sodass nur noch indirekte Spuren ihres gemeinsamen Ursprungs erhalten bleiben. Die Identifikation solcher gleichaltriger Sternassoziationen ist entscheidend, um die jüngere Sternentstehungsgeschichte in der Umgebung der Sonne zu rekonstruieren. Mithilfe dichtebasierter Clustering-Algorithmen, die auf die präzisen fünfdimensionalen astrometrischen Daten der *Gaia*-Mission angewendet werden, konnten bereits Tausende neuer Gruppen entdeckt werden. Im unmittelbaren solaren Umfeld führen jedoch starke Projektionseffekte dazu, dass Überdichten in der Tangentialgeschwindigkeit weitgehend verloren gehen. Eine Clusteranalyse im sechsdimensionalen galaktischen kartesischen Phasenraum könnte dieses Problem umgehen, erfordert allerdings Radialgeschwindigkeiten, die nur für einen Bruchteil der *Gaia*-Sterne vorliegen.

In dieser Arbeit wird DENSE (Deep Semi-analytic Embedding of the Nearby Stellar Environment) vorgestellt, ein semi-analytisches neuronales Netz, das aus *Gaia*-ähnlicher Astrometrie mit fehlenden Radialgeschwindigkeiten eine annähernde Darstellung im 6D-Phasenraum ($XYZUVW$) lernt. Ein neuronaler Encoder bildet die 5D-Observablen auf einen latenten Raum ab, der als galaktischer kartesischer Phasenraum interpretiert wird, während ein analytischer Decoder die exakten Koordinatentransformationen zurück zu Astrometrie und Radialgeschwindigkeit ausführt. Das Modell wird so trainiert, dass es die Observablen rekonstruiert und relative Geschwindigkeiten zwischen Sternen bewahrt und damit die fehlende Radialgeschwindigkeit effektiv erschließt.

DENSE wird an zwei synthetischen, *Gaia*-ähnlichen Datensätzen mit mehreren in einen gleichförmigen Hintergrund eingebetteten Gruppen getestet. Für ein einfacheres Szenario rekonstruiert das Modell Positionen mit Unterparsec-Genauigkeit und Geschwindigkeiten mit Fehlern von nur wenigen km s^{-1} . Ein anschließendes HDBSCAN-Clustering im latenten Raum findet alle ursprünglichen Cluster mit einem NMI (Normalized Mutual Information) von 1.0 und erreicht damit dieselbe Qualität wie ein Clustering im wahren 6D-Phasenraum, während eine direkte Anwendung von HDBSCAN in den 5D-Observablen deutlich schlechter abschneidet. Für einen komplexeren Datensatz mit ausgedehnten, teilweise überlappenden Strukturen erzielt DENSE NMI-Werte von ~ 0.6 – 0.68 und verdoppelt damit ungefähr die Resultate eines 5D-basierten Clusterings. Insgesamt zeigen die Ergebnisse, dass semi-analytische neuronale Ansätze wie DENSE informative Phasenraum-Einbettungen liefern und die Detektion naher stellarer Assoziationen deutlich verbessern können, selbst wenn Radialgeschwindigkeiten nur lückenhaft vorliegen.

Contents

Acknowledgements	i
Abstract	iii
Kurzfassung	v
List of Tables	ix
List of Figures	xi
1. Introduction	1
2. Background	5
2.1. Young Nearby Stellar Associations	5
2.2. Data	7
2.2.1. <i>Gaia</i> Data	7
2.2.2. Coordinate Systems	8
2.2.3. From <i>Gaia</i> Astrometry to full 6D Phase Space	9
2.3. Generated Data Samples	10
2.3.1. Simple Data Set	11
2.3.2. Challenging Data Set	11
2.4. Clustering	11
2.4.1. <i>K</i> -Means Clustering	14
2.4.2. DBSCAN	14
2.4.3. HDBSCAN	15
2.4.4. Limitations due to Projection Effects	16
2.5. Overcoming Projection Effects	16
3. Related Work	21
3.1. Clustering in Semi-Metric Spaces	21
3.1.1. Softmax Algorithm	21
3.1.2. Softmax Clustering of Optimal-Velocity Distances	23
3.1.3. Caveats	24
3.2. Deep Learning	24
3.2.1. Neural Networks	25
3.2.2. Autoencoder	26
3.2.3. Deep Clustering	27

4. DeNSE Algorithm	31
4.1. Problem Formulation and Data Representation	31
4.2. Batch Construction in Mixed Spatial–Kinematic Space	33
4.3. Semi-Analytic Autoencoder Architecture	37
4.4. Training Objective	41
4.5. Optimization and Implementation Details	44
4.6. Clustering in the DENSE Latent Space	45
5. Results	47
5.1. Baseline Clustering	47
5.1.1. HDBSCAN on 5D <i>Gaia</i> -like Observables	47
5.1.2. HDBSCAN on 6D Galactic Cartesian Phase Space	48
5.2. Performance of DENSE	49
5.2.1. Hyperparameter Grid Search	49
5.3. Experiments	51
5.3.1. Additional Loss Terms	51
5.3.2. Additional Input Information	52
5.4. Ablation Study	55
5.5. Best-Performing Configurations	56
5.5.1. Simple Data Set	56
5.5.2. Challenging Data Set	61
5.6. Running Time Analysis	71
6. Discussion	73
6.1. Main Findings	73
6.2. Comparison to Semi-Metric Clustering	74
6.3. Limitations and Caveats	75
6.4. Future Work	75
7. Conclusion	77
Bibliography	79
A. Appendix	85
A.1. Ad Fig. 2.4	85
A.2. Learning Curves of Individual Loss Terms	85

List of Tables

5.1. Best HDBSCAN performance on the synthetic samples in 5D observable astrometric space and in the true 6D Galactic phase space. Shown are the hyperparameters <code>min_samples</code> (m_{pts}) and <code>min_cluster_size</code> (m_{clSize}), the number of detected clusters N_{Cl} , and the normalized mutual information (NMI), computed ignoring background stars.	49
5.2. Representative DENSE configurations for the simple synthetic sample. Shown are batch size M , repeats per epoch K , loss weights, HDBSCAN hyperparameters, mean position and velocity errors $\bar{\epsilon}_{\text{xyz}}$ and $\bar{\epsilon}_{\text{uvw}}$, the number of detected clusters N_{Cl} , and NMI (ignoring background stars). .	50
5.3. Representative DENSE configurations for the challenging synthetic sample. Columns are as in Table 5.2.	51
5.4. Effect of additional XYZ and UVW loss terms on the simple sample. . .	52
5.5. Effect of additional XYZ and UVW loss terms on the challenging sample. .	52
5.6. DENSE performance on the simple sample when augmenting the encoder input with Cartesian positions (XYZ).	53
5.7. Same as Table 5.6, but for the challenging sample.	53
5.8. DENSE performance on the simple sample when augmenting the encoder input with the trial UVW velocity used for batch construction.	54
5.9. Same as Table 5.8, but for the challenging sample.	54
5.10. Ablation study for the simple sample, starting from a configuration with $M = 130$ and $K = 5$. Each row shows a different combination of active loss terms.	56
5.11. Ablation study for the challenging sample, based on the run with additional trial UVW input and hyperparameters $M = 250$ and $K = 5$	56
5.12. Composition of the three clusters detected by DENSE in the simple sample, for the chosen best-performing configuration.	60
5.13. Composition of the five clusters detected by DENSE in the challenging sample, for the chosen best-performing configuration.	66

List of Figures

2.1.	2D projection of the star cluster distribution in the 250-pc box closest to the Sun. The clusters are colored by age and sized by the number of star members. This figure includes the most recent cluster census (Hunt and Reffert 2023), the young local associations from Gagné et al. (2018), and the clustering solution for Sco-Cen from Ratzenböck et al. (2023). The Sun is located at the center, indicated by the yellow circle. Link to interactive 3D figure.	6
2.2.	Top panel: Simple synthetic data sample in Galactic Cartesian phase space ($XYZUVW$). The Sun is located at the origin. The axes are set equal to show that the velocity space is more compact. Bottom panel: Same sample in observational equatorial space, showing sky positions (α, δ) and tangential velocities.	12
2.3.	Top panel: Challenging synthetic data sample in Galactic Cartesian phase space ($XYZUVW$). The Sun is located at the origin. The axes are set equal to show that the velocity space is more compact. Bottom panel: Same sample in observational equatorial space, where the clusters appear diffuse and heavily blended in projected position and tangential velocity.	13
2.4.	Proper-motion dispersion as a function of distance for stars within 1 kpc from <i>Gaia</i> DR3. The black point and dashed lines indicate the adopted distance cutoff identified with <i>KneeLocator</i> (Arvai 2020). Below this distance projection effects significantly reduce the effectiveness of density-based clustering algorithms such as HDBSCAN (see Appendix A.1 for details).	17
4.1.	Batch construction for the simple data sample for different trial UVW values. Each row corresponds to one trial velocity and shows the stars in X - Y (left) and U - V (right) space, colored by batch ID.	35
4.2.	Same as Figure 4.1, but for the challenging data sample with five clusters.	36
4.3.	Recovery fraction (left) and entropy (right) as a function of the offset $ \Delta\vec{v} $ between the trial and true cluster velocities for the simple (top row) and challenging (bottom row) data samples.	38
4.4.	True cluster centroids in the U - V plane (circles), shifted centroids for a representative offset $ \Delta\vec{v} = 10 \text{ km s}^{-1}$ (squares), and the corresponding shift directions (arrows) for the respective samples. These shifts are used to generate the curves in Figure 4.3.	39

List of Figures

4.5. Distribution of stars across batches for $ \Delta\vec{v} = 0 \text{ km s}^{-1}$, i.e., when the trial velocity is set equal to the true mean velocity of the corresponding cluster. Each panel shows, for the background or a given cluster, the number of stars assigned to each batch ID.	40
4.6. Schematic overview of the baseline DENSE architecture. A neural encoder maps the observables to a 6D latent space, and an analytic decoder implements the coordinate transformation back to the observed astrometry. During training, DENSE minimizes a weighted sum of $\mathcal{L}_{\text{mat}}$, $\mathcal{L}_{\text{ast}}$, $\mathcal{L}_{\text{rv}}$, and $\mathcal{L}_{\text{ws}}$ on each mini-batch. After training, HDBSCAN is run in the latent space, and the final clustering C^* is selected as the one with the highest NMI over all considered epochs and hyperparameter settings. The visual layout is inspired by the architecture diagram in Beer et al. (2024).	42
5.1. Training and validation loss versus epoch for the selected DENSE configuration on the simple sample.	57
5.2. Latent-space embedding of the simple sample in predicted Cartesian positions XYZ (top) and velocities UVW (bottom). Points are colored by their ground-truth cluster labels.	58
5.3. True versus predicted Cartesian coordinates for the simple sample in positions XYZ (top) and velocities UVW (bottom).	59
5.4. Axis-wise errors in the latent-space positions (left) and velocities (right) for the simple sample. The mean absolute errors in XYZ are (0.5, 0.4, 0.5) pc and in UVW (2.0, 2.0, 1.7) km s^{-1}	60
5.5. Best HDBSCAN clustering result in the DENSE latent space for the simple sample. Points are colored by their ground-truth cluster labels.	60
5.6. NMI of HDBSCAN clusterings in the DENSE latent space for the simple sample.	61
5.7. True Galactic Cartesian coordinates for the simple sample, colored by detected HDBSCAN clusters. Top row: positions XYZ , bottom row: velocities UVW . Left column: clusters obtained by running HDBSCAN directly in the true 6D phase space ($XYZUVW$). Right column: clusters obtained by running HDBSCAN in the DENSE latent space.	62
5.8. Training and validation loss versus epoch for the chosen DENSE configuration on the challenging sample. Both curves indicate convergence of the training after roughly 150 epochs. This epoch coincides with the point at which the NMI curve in Figure 5.13 levels off.	63
5.9. Latent-space embedding of the challenging sample in predicted XYZ (top) and UVW (bottom). Points are colored by their ground-truth cluster labels.	64
5.10. True versus predicted Cartesian coordinates for the challenging sample in positions XYZ (top) and velocities UVW (bottom).	65
5.11. Axis-wise errors in the latent-space positions (left) and velocities (right) for the challenging sample. The mean absolute errors in XYZ are (3.5, 2.6, 2.1) pc and in UVW (4.5, 3.9, 3.2) km s^{-1}	66

5.12. Best HDBSCAN clustering result in the DENSE latent space for the challenging sample. Points are colored by their ground-truth cluster labels.	67
5.13. Same as Figure 5.6, but for the challenging sample. At each epoch, multiple HDBSCAN hyperparameter combinations are evaluated, producing a broad spread of NMI values. The upper envelope and typical (median) NMI increase during the first ~ 150 epochs and then remain in the range $\sim 0.5-0.65$, with no further systematic trend, consistent with the flattening of the loss curves in Figure 5.8.	68
5.14. True Galactic Cartesian coordinates for the challenging sample, colored by detected HDBSCAN clusters. Top row: positions XYZ , bottom row: velocities UVW . Left column: clusters obtained by running HDBSCAN directly in the true 6D phase space ($XYZUVW$). Right column: clusters obtained by running HDBSCAN in the DENSE latent space.	69
5.15. True Galactic Cartesian coordinates for the challenging sample, colored by detected HDBSCAN clusters. Top row: positions XYZ , bottom row: velocities UVW . Left column: clusters obtained by running HDBSCAN directly in the true 6D phase space ($XYZUVW$). Right column: stars plotted at their true coordinates, colored by the clusters obtained from HDBSCAN in the hybrid space ($X_{\text{true}}, Y_{\text{true}}, Z_{\text{true}}, 5U_{\text{pred}}, 5V_{\text{pred}}, 5W_{\text{pred}}$).	70
A.1. Training and validation loss for runs using only the matrix loss $\mathcal{L}_{\text{mat}}$ on the simple sample.	86
A.2. Training and validation loss for runs using only the astrometric reconstruction loss $\mathcal{L}_{\text{ast}}$ on the simple sample.	86
A.3. Training and validation loss for runs using only the MSE radial velocity loss $\mathcal{L}_{\text{rv}}$ on the simple sample.	87
A.4. Training and validation loss for runs using only the Wasserstein distance loss $\mathcal{L}_{\text{ws}}$ on the simple sample.	87
A.5. Training and validation loss for runs using only the XYZ position loss $\mathcal{L}_{\text{xyz}}$ on the simple sample.	88
A.6. Training and validation loss for runs using only the UVW velocity loss $\mathcal{L}_{\text{uvw}}$ on the simple sample.	88

1. Introduction

Stars do not form in isolation. They originate in groups, commonly called star clusters or associations, that develop from the same clouds of gas and dust. Early in their lifetimes, the shared positions and motions of these stars still reveal their common origins. Some clusters remain gravitationally bound for hundreds of millions of years, while others disperse more quickly, leaving behind only subtle traces of their shared past. Identifying both bound and unbound clusters is essential for understanding the early phases of star and planet formation and for reconstructing the recent star formation history of our Galactic neighborhood (e.g., [Ambartsumian 1947](#); [Blaauw 1964](#)).

Star clusters have been known since the beginning of astronomy, as some of them, such as the Pleiades and the Hyades, are visible even to the naked eye. During the past decades, the HIPPARCOS ([Perryman et al. 1997](#)) and the *Gaia* mission of the European Space Agency ([Gaia Collaboration et al. 2016](#); [Gaia Collaboration et al. 2022](#)) have revolutionized our view of the Milky Way. The *Gaia* satellite measures the positions and motions of more than a billion stars with remarkable precision, allowing astronomers to map our Galaxy in three dimensions and to search for groups of stars that originated together.

Classical clustering methods such as DBSCAN ([Ester et al. 1996](#)) and HDBSCAN ([Campello et al. 2013](#)) have been widely used to detect star clusters from *Gaia* data within our Galactic neighborhood. They identify clusters as overdensities of stars that lie close together in space and move in similar directions. Applied to high-quality *Gaia* astrometry, they have proven to be an effective tool, substantially revising the pre-*Gaia* cluster census (e.g. [Kharchenko et al. 2013](#)) and leading to the discovery of thousands of previously unrecognized groups ([Cantat-Gaudin et al. 2018](#); [Castro-Ginard et al. 2018](#); [Hunt and Reffert 2021](#); [Hunt and Reffert 2023](#)).

Despite the high astrometric precision for nearby stars, clustering becomes difficult at close distances because projection effects dominate the observable quantities. Nearby stars span large areas of the sky, and geometric projection effects distort the way their motions appear in the sky. Two stars moving with nearly identical three-dimensional velocities can have markedly different proper motions depending on their positions relative to the Sun. In contrast, unrelated stars may display similar projected motions. These distortions make clustering in the observed astrometric space unreliable. As a result, many of the closest young stellar groups blend into the general field population, leading to incomplete or inconsistent catalogs of associations in the Solar neighborhood ([Zuckerman and Song 2004](#); [Torres et al. 2008](#); [Gagné et al. 2018](#); [Gagné 2024](#)).

1. Introduction

In principle, these challenges could be mitigated by applying clustering algorithms not to the observed coordinates provided by *Gaia* but to Galactic Cartesian coordinates, where the true spatial and kinematic structure of stars becomes directly accessible (e.g. [Wilkinson et al. 2005](#)). In this coordinate system, projection effects are eliminated, and nearby associations appear as compact groups relative to the general field population. However, this transformation requires radial velocities, which measure the component of stellar motion along the line of sight. The majority of stars observed by *Gaia* lack measured radial velocities, preventing the reconstruction of their full three-dimensional velocity vectors ([Naik and Widmark 2023](#)). Without complete kinematic information, clustering in physically meaningful space is impossible for most stars.

This thesis introduces DENSE, a Deep Semi-analytic Embedding of the Nearby Stellar Environment, a data-driven approach designed to overcome the limitations imposed by missing radial velocities in *Gaia* data. DENSE combines the underlying coordinate transformations from observed to Cartesian space with a neural network that learns to infer the missing kinematic dimensions, producing a latent representation that approximates the true Galactic Cartesian phase space of stars, even when their line-of-sight velocities are unavailable. In this learned latent space, projection effects are mitigated, allowing standard clustering algorithms, such as HDBSCAN, to identify coherent stellar groups more effectively. By bridging the gap between incomplete astrometric measurements and fully reconstructed six-dimensional kinematics, DENSE aims to improve the completeness and reliability of the current census of nearby stellar associations within the Solar neighborhood.

The main contributions of this work are:

- A semi-analytic deep-learning framework for recovering latent Galactic Cartesian coordinates without radial velocities. The DENSE model combines analytic coordinate transformations with a neural architecture that learns to approximate full six-dimensional phase-space information from incomplete astrometric data, allowing clustering in a physically meaningful space.
- A method for constructing dynamic, physically motivated training batches. The training pipeline introduces batch construction based on clustering and connectivity in mixed spatial-kinematic feature spaces, ensuring that each epoch exposes the model to different plausible kinematic realizations consistent with the observed astrometry and improves the stability of the learned latent representation.
- A synthetic benchmark data set designed to emulate *Gaia* astrometry with missing radial velocities. This data set includes multiple Gaussian clusters and background populations transformed into ICRS coordinates, providing controlled ground truth to evaluate the accuracy of the model reconstruction and its ability to recover the three-dimensional structure.

- A detailed evaluation of the learned latent space. This work analyzes the accuracy of the recovered six-dimensional coordinates across multiple passes per star, quantifying how well the model reduces projection distortions and reconstructs coherent groups in the Solar neighborhood.
- A demonstration of improved cluster recoverability relative to clustering directly in the observed astrometric space. Standard clustering tools, when applied to the learned latent representation, recover spatially extended and kinematically coherent groups that remain difficult to detect in projected coordinates, pointing toward a more complete census of nearby stellar associations.

The remainder of this thesis is organized as follows: Chapter 2 introduces the background of the *Gaia* mission, projection effects, and the data sets used in this work. Chapter 3 reviews related studies and clustering approaches in astronomy and machine learning. Chapter 4 presents the DENSE algorithm and its methodological components. Chapter 5 evaluates the method on synthetic data and examines its ability to recover the underlying stellar structure. Chapter 6 discusses the main findings and highlights the directions for future work.

2. Background

This chapter provides the astronomical, observational, and methodological background needed to understand the challenges of identifying stellar associations in the Solar neighborhood and to motivate the design of the DENSE algorithm. It introduces the *Gaia* mission and its data products, outlines the physical and computational difficulties of clustering in astrometric space, and describes the synthetic data sets used for model development and evaluation.

2.1. Young Nearby Stellar Associations

Within approximately 100 parsecs¹ of the Sun, roughly a dozen young, low-density stellar associations have been identified. These populations are typically younger than 100 Myr, span spatial extents of up to 20 pc, and comprise hundreds of candidate members. Due to their proximity, these associations serve as prime benchmarks for studying the processes of star and planet formation, yet their census remains incomplete and, in many cases, uncertain (Gagné et al. 2018; Gagné 2024).

Figure 2.1 shows the currently known young stellar groups within the local 100-pc region. Interestingly, no star clusters have yet been identified in the northern hemisphere of this local sphere, a feature that remains poorly understood. In the Solar neighborhood, projection effects and missing radial velocity measurements complicate the identification of nearby clusters by limiting the interpretability of velocity-space information. Existing clustering methods struggle to address these challenges, leading to incomplete or inconsistent catalogs of young stellar associations. This incomplete census hinders efforts to answer key questions about the dynamics, evolution, and star formation history of the nearest stellar and substellar populations.

Early efforts to identify young stellar populations in the Solar neighborhood relied on a combination of astrometric catalogs and X-ray observations from the ROSAT all-sky survey (Voges et al. 1999). Young stars exhibit enhanced magnetic activity, which leads to elevated X-ray emission, making X-ray detection a strong but biased indicator of youth (Zuckerman and Song 2004; Torres et al. 2008). These searches succeeded in uncovering the first nearby co-moving groups, yet they remained sensitive primarily to the youngest and most compact associations (Gagné 2024).

¹A parsec (pc) is a unit of distance commonly used in astronomy, equivalent to about 3.26 light-years.

2. Background

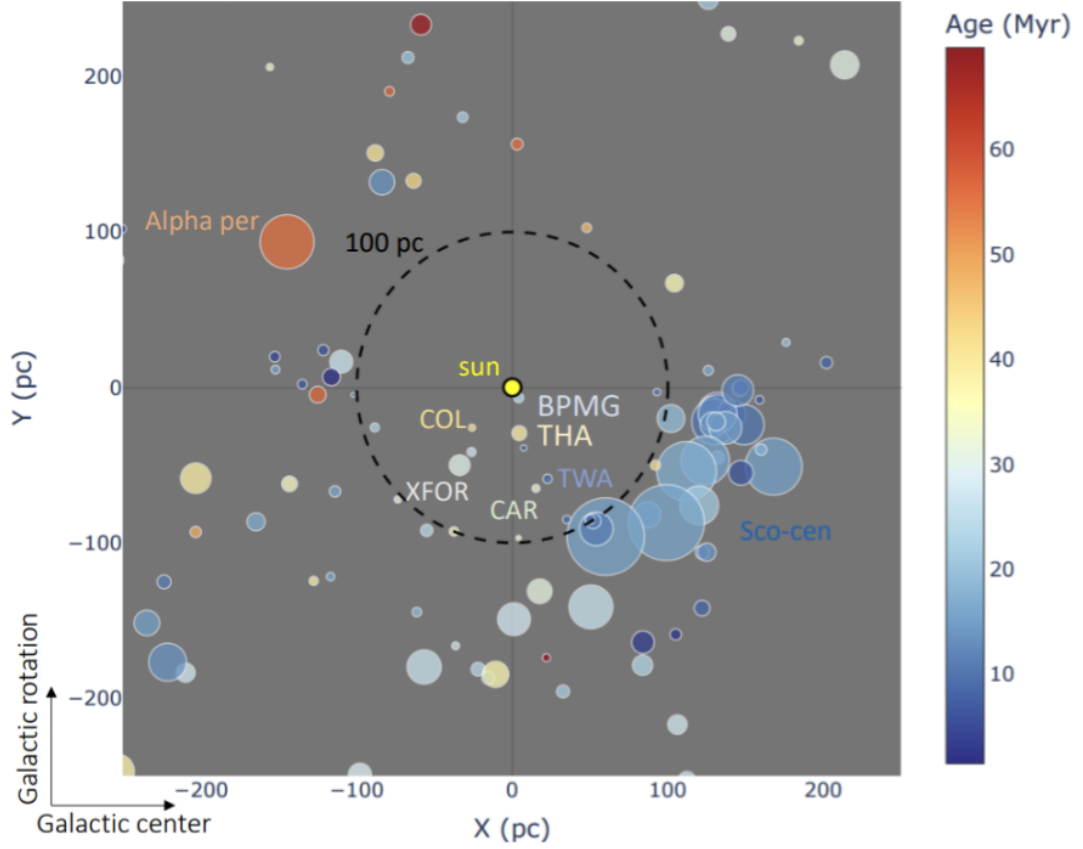


Figure 2.1.: 2D projection of the star cluster distribution in the 250-pc box closest to the Sun. The clusters are colored by age and sized by the number of star members. This figure includes the most recent cluster census (Hunt and Reffert 2023), the young local associations from Gagné et al. (2018), and the clustering solution for Sco-Cen from Ratzenböck et al. (2023). The Sun is located at the center, indicated by the yellow circle. [Link to interactive 3D figure.](#)

To build a more complete census, subsequent analyses shifted toward kinematic selection. Methods such as the convergent point technique (de Bruijne 1999), the “good box” selection (Zuckerman and Song 2004), and goodness-of-fit membership metrics (Kraus et al. 2014) use characteristic motion patterns to identify likely members of known groups. The convergent-point method searches for stars whose proper motions intersect at a common point on the sky, implying a shared three-dimensional velocity (de Bruijne 1999; Montes et al. 2001). The “good box” approach defines empirical bounding regions in Cartesian velocity space, selecting stars whose three-dimensional velocities fall within the known distribution of a group (Zuckerman and Song 2004). Goodness-of-fit models

compare observed proper motions and photometric properties to those expected for a given association, incorporating the characteristic overluminosity of young pre-main-sequence stars (Kraus et al. 2014).

More recently, probabilistic and Bayesian frameworks have improved the robustness of membership assessments (e.g., Malo et al. 2013; Gagné et al. 2014; Riedel et al. 2017; Gagné et al. 2018a; Gagné et al. 2018b; Gagné and Faherty 2018). These models describe each association as a multivariate Gaussian component in six-dimensional phase space, enabling the computation of membership probabilities even in the absence of radial velocities (Riedel et al. 2017; Gagné et al. 2018). Such approaches have substantially expanded the known populations of nearby groups and reduced contamination rates within individual associations.

Despite their success, all of these techniques assume that the associations already exist. They require prior knowledge of the position and velocity distributions of each group. As a result, these methods are not suitable for discovering previously unknown stellar associations (Gagné 2024). They can refine membership lists, but they cannot independently reveal new structures in phase space.

The clustering algorithm developed in this work is designed to address this gap. It is tailored specifically for high-precision *Gaia* astrometry and aims to identify previously unrecognized stellar associations in the Solar neighborhood by directly addressing and mitigating spherical projection artifacts.

2.2. Data

This section describes the observational data, coordinate conventions, and synthetic data sets used throughout this work. The goal is to provide a clear link between *Gaia* astrometric measurements and the full six-dimensional phase-space coordinates recovered by the DENSE algorithm. The last part of this section introduces the generated data sets used to test and validate the algorithm in regimes where traditional clustering approaches are known to fail.

2.2.1. *Gaia* Data

Launched in 2013 by the European Space Agency (ESA), the *Gaia* mission is the most precise astrometric survey ever conducted. Operating from the second Sun–Earth Lagrange point, *Gaia* continuously scans the sky to measure the positions, distances, and motions of more than a billion stars in the Milky Way (Gaia Collaboration et al. 2016).

The most recent release, *Gaia* DR3 (Gaia Collaboration et al. 2022), provides five-parameter astrometry, including positions (ra, dec), parallaxes, and proper motions for nearly two billion sources. DR3 also includes radial velocities for tens of millions of stars,

2. Background

a substantial improvement over previous releases, yet still only a small fraction of the full catalog. Within 100 pc, only about 20% of the stars in the catalog have full 6D measurements.

Despite this limitation, the excellent precision of the *Gaia* 5-parameter astrometry has revolutionized our view of the Solar neighborhood. For example, it has enabled the discovery of new clusters (Cantat-Gaudin et al. 2018; Castro-Ginard et al. 2018; Kounkel and Covey 2019; Hunt and Reffert 2023) and the identification of kinematic substructure and streams (Meingast and Alves 2019; Kamdar et al. 2019; Meingast et al. 2019; Meingast et al. 2021; Kamdar et al. 2021; Ratzenböck et al. 2025), the mapping of their 3D morphology (Meingast et al. 2021; Tarricq et al. 2022), and the characterization of large-scale dynamical structures such as the Radcliffe Wave (Alves et al. 2020) and the Split (Lallement et al. 2019), among many other results.

For methods seeking to uncover previously unknown stellar associations, *Gaia* offers an exceptional but incomplete six-dimensional view. To exploit this view, we must translate native angular observables from *Gaia* into Cartesian phase space, which motivates the coordinate conventions introduced next.

2.2.2. Coordinate Systems

Equatorial (ICRS) Coordinates

Astrometric measurements in the *Gaia* catalog are reported in the International Celestial Reference System (ICRS), a quasi-inertial, barycentric reference frame defined by the positions of distant extragalactic sources. The axes of the spherical coordinate system are closely aligned with the mean celestial equator and the mean celestial equinox at the J2000 epoch.

The position of a star is described by right ascension (α) and declination (δ), while its motion across the sky is encoded in the proper motions ($\mu_{\alpha*}, \mu_{\delta}$), where $\mu_{\alpha*} = \mu_{\alpha} \cos \delta$. Distances are inferred from the parallax ϖ , and radial velocities (when available) measure the line-of-sight component of space motion. In the *Gaia* catalog, α and δ are given in degrees, $\mu_{\alpha*}$ and μ_{δ} in mas yr^{-1} , ϖ in mas, and radial velocities in km s^{-1} .²

ICRS coordinates reflect what *Gaia* observes directly: angular positions, their time derivatives, and an annual parallax shift. They are therefore the natural starting point for converting *Gaia* measurements into 3D positions and velocities.

²Here mas denotes milliarcseconds, i.e. $1 \text{ mas} = 10^{-3} \text{ arcseconds}$, and mas yr^{-1} an angular change of one milliarcsecond per year.

Galactic Coordinates

The Galactic coordinate system reorients the sky relative to the Milky Way disk. Latitude (b) measures the angular distance from the Galactic plane, longitude (l) points toward the Galactic center, and proper motions (μ_l, μ_b) describe projected motion along these directions.

This system offers a more natural coordinate system, but is still fundamentally angular and projected. For the purposes of this work, it is therefore advantageous to adopt the associated rectilinear frame, in which stars have explicit 3D positions and velocities, allowing for more reliable clustering.

Galactic Cartesian Coordinates

The associated rectilinear frame is the Galactic Cartesian system, a right-handed Cartesian coordinate system centered on the Sun. Positions (XYZ) and velocities (UVW) are defined such that X and U point toward the Galactic center, Y and V follow the direction of Galactic rotation, and Z and W point toward the north Galactic pole.

This framework is the natural space for clustering algorithms seeking physically coherent structures and provides the target representation for the DENSE algorithm.

2.2.3. From *Gaia* Astrometry to full 6D Phase Space

Recovering the full 6D phase space ($XYZUVW$) from *Gaia* observables requires a non-linear transformation from angles, parallaxes, proper motions, and radial velocities to Cartesian coordinates. Conceptually, the angular coordinates and parallax specify a star's position in space and, when combined with proper motions and radial velocity, determine its full 3D velocity.

For nearby stars ($\lesssim 200$ pc) and small parallax uncertainties considered in this work, distances are estimated from the parallax ϖ as

$$d [\text{pc}] \approx \frac{1000}{\varpi [\text{mas}]},$$

which is a good approximation in this regime³ (e.g. [Bailer-Jones 2015](#)).

Positions in the ICRS Cartesian frame follow from the standard spherical-to-Cartesian

³For stars in the Solar neighborhood, *Gaia* parallaxes typically have uncertainties of the order of a few tens to a few hundred μas . At the small distances (i.e. large parallaxes) considered here, this corresponds to a high parallax signal-to-noise ratio for most stars, so that the simple inverse-parallax estimator provides an accurate distance.

2. Background

conversion:

$$\begin{aligned}x &= d \cos \delta \cos \alpha, \\y &= d \cos \delta \sin \alpha, \\z &= d \sin \delta,\end{aligned}$$

and are then rotated into the Galactic Cartesian frame via the fixed ICRS-to-Galactic rotation matrix $R_{\text{gal} \leftarrow \text{icrs}}$:

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = R_{\text{gal} \leftarrow \text{icrs}} \begin{pmatrix} x \\ y \\ z \end{pmatrix}.$$

Equivalently, one may regard this as computing the Galactic longitude and latitude (l, b) from (α, δ) using the same fixed rotation and then applying

$$X = d \cos b \cos l, \quad Y = d \cos b \sin l, \quad Z = d \sin b.$$

Proper motions describe angular motion on the sky rather than physical speeds. The tangential velocity components in the directions of increasing right ascension and declination can be written as

$$v_\alpha = 4.74047 \frac{\mu_{\alpha*}}{\varpi}, \quad v_\delta = 4.74047 \frac{\mu_\delta}{\varpi},$$

where $\mu_{\alpha*}$ and μ_δ are in mas yr^{-1} , ϖ is in mas , and the numerical factor $4.74047 \text{ km s}^{-1}$ corresponds to 1 AU yr^{-1} expressed in km s^{-1} , ensuring that the resulting velocities are in km s^{-1} .

Let $\hat{\mathbf{r}}$, $\hat{\boldsymbol{\alpha}}$, and $\hat{\boldsymbol{\delta}}$ denote the orthonormal basis vectors pointing along the line of sight, increasing right ascension, and increasing declination at (α, δ) . The space velocity in the ICRS Cartesian frame can then be expressed as

$$\mathbf{v}_{\text{icrs}} = v_{\text{rad}} \hat{\mathbf{r}} + v_\alpha \hat{\boldsymbol{\alpha}} + v_\delta \hat{\boldsymbol{\delta}},$$

where v_{rad} denotes the radial velocity. Applying the same fixed rotation as for the positions yields the Galactic Cartesian velocity components:

$$\begin{pmatrix} U \\ V \\ W \end{pmatrix} = R_{\text{gal} \leftarrow \text{icrs}} \mathbf{v}_{\text{icrs}}.$$

In this way, the *Gaia* observables $(\alpha, \delta, \varpi, \mu_{\alpha*}, \mu_\delta, v_{\text{rad}})$ are mapped to full Galactic Cartesian phase-space coordinates $(XYZUVW)$, which serve as the target representation for the clustering analysis.

2.3. Generated Data Samples

To evaluate the performance of the clustering method developed in this thesis, I constructed two synthetic *Gaia*-like data sets that differ in their clustering complexity. In both data sets, only 10% of the stars are assigned radial velocities, replicating the incompleteness present in real *Gaia* observations.

2.3.1. Simple Data Set

The first data set consists of three compact Gaussian clusters with 100 stars each, embedded in a uniform background of 1000 stars. It provides a controlled environment in which the algorithm is expected to recover the underlying structure almost perfectly. Each cluster is compact in the full 6D Cartesian phase space ($XYZUVW$), with small dispersions in both position and velocity, while the background population fills a rectangular volume around the clusters.

Figure 2.2 depicts the three generated clusters in 6D position–velocity space in the top panel and their projected positions and motions in the bottom panel. Although the clusters are compact in Galactic Cartesian space, in tangential-velocity space they appear intertwined, which already hinders efficient clustering with classical methods that operate purely in observational space.

2.3.2. Challenging Data Set

The second data set is designed to better resemble the complexity of the Solar neighborhood. It contains five clusters with broader intrinsic dispersions and non-trivial spatial covariances. Each cluster contains 250 stars, while the 2500 background stars are distributed uniformly. This data set serves as a realistic stress test for the algorithm. The clusters are still coherent in phase space, but they are loose and partially overlapping, mimicking the structure of nearby associations.

Figure 2.3 shows this more realistic data sample in 6D Cartesian phase space in the top panel and the projected positions and motions in the bottom panel. The clusters are compact in Galactic Cartesian velocity space but more dispersed in position space, as is also expected for real clusters that have begun to shear and expand. In equatorial coordinates, the clusters are not dense at all. In projected position and velocity space, they appear intertwined and widely distributed. This sample is therefore a demanding test case for any method that attempts clustering directly in observational space.

Applying classical clustering techniques to these two generated data samples directly in the observed 5D astrometric space ($\alpha, \delta, \varpi, \mu_{\alpha*}, \mu_\delta$) performs poorly because projection effects blur the intrinsic phase-space structure and wash out the overdensities that define the clusters.

2.4. Clustering

Clustering is an unsupervised learning technique that partitions a data set into subsets, or clusters, based on a measure of similarity or distance between data points. Formally, given a set of N observations $\{x_i\}_{i=1}^N$ in a d -dimensional feature space, clustering aims to construct a mapping $f : \mathbb{R}^d \rightarrow \{1, \dots, K\}$ that assigns each observation to one of

2. Background

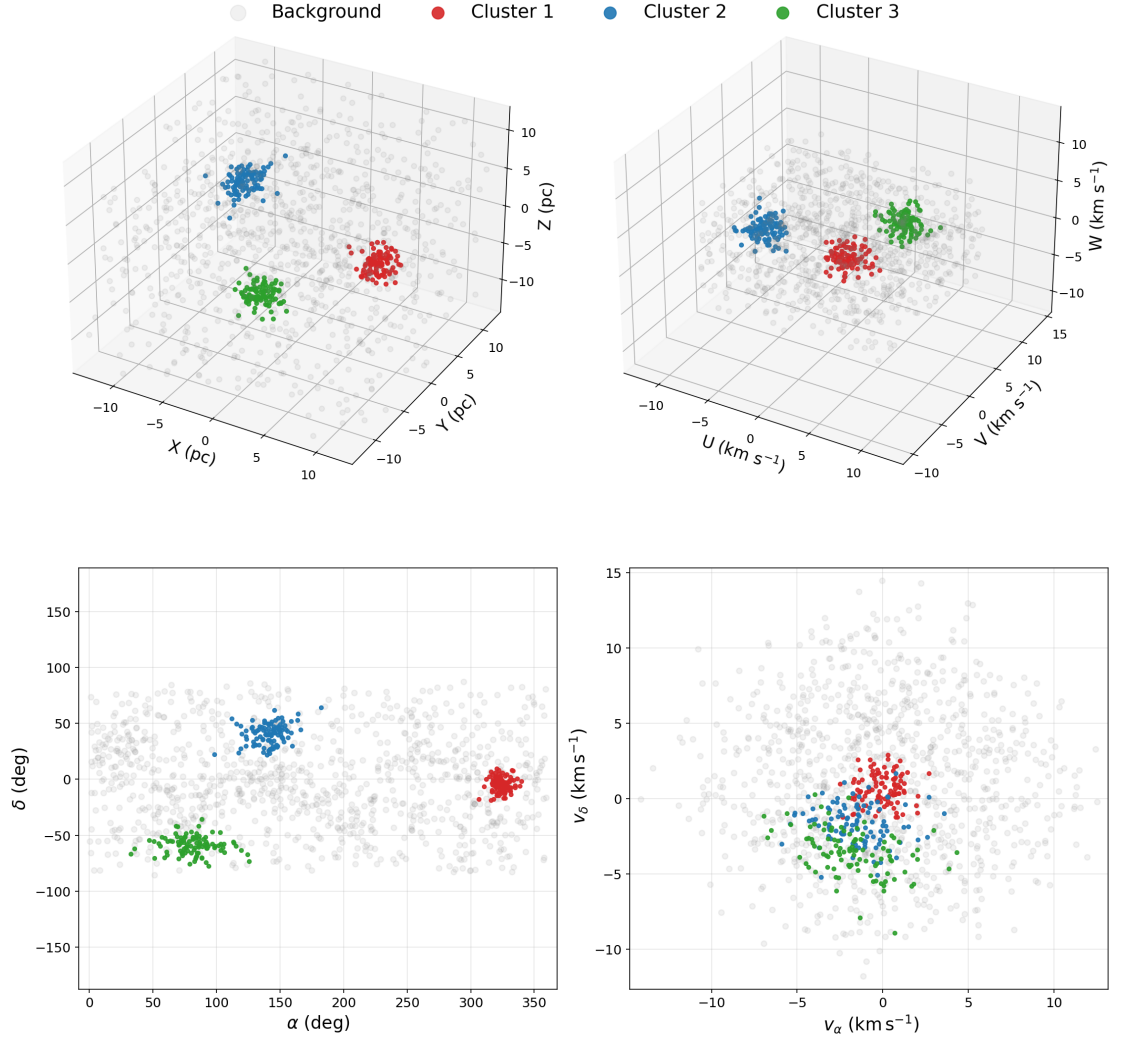


Figure 2.2.: Top panel: Simple synthetic data sample in Galactic Cartesian phase space ($XYZUVW$). The Sun is located at the origin. The axes are set equal to show that the velocity space is more compact. Bottom panel: Same sample in observational equatorial space, showing sky positions (α, δ) and tangential velocities.

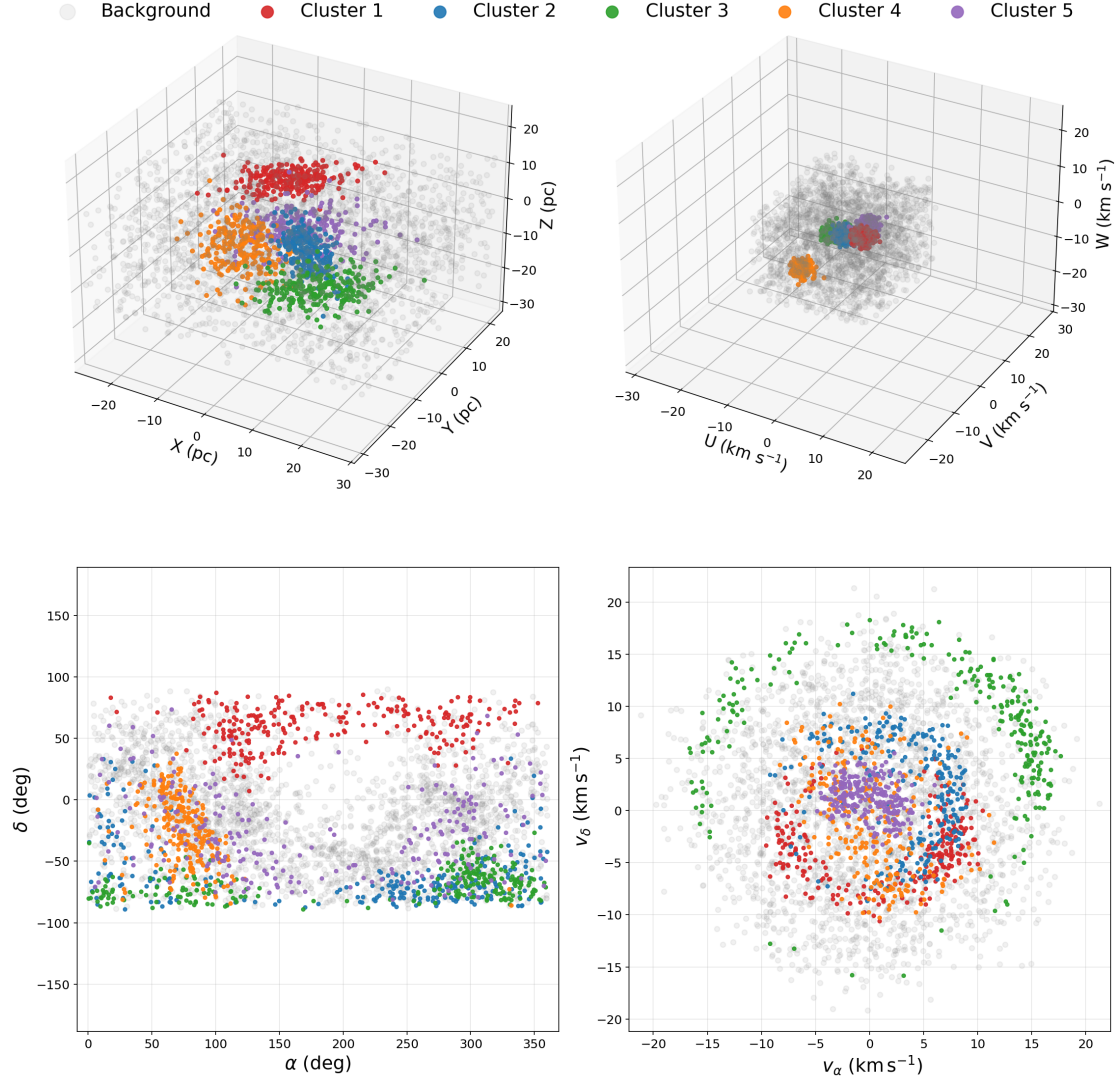


Figure 2.3.: Top panel: Challenging synthetic data sample in Galactic Cartesian phase space ($XYZUVW$). The Sun is located at the origin. The axes are set equal to show that the velocity space is more compact. Bottom panel: Same sample in observational equatorial space, where the clusters appear diffuse and heavily blended in projected position and tangential velocity.

2. Background

K clusters such that intra-cluster similarity is maximized and inter-cluster similarity is minimized. In contrast to supervised learning, clustering operates without predefined labels and instead exploits the intrinsic structure of the data (e.g., Jain et al. 1999; Xu and Wunsch 2005).

In the context of stellar dynamics, clustering methods are key tools for identifying co-moving stellar populations in large-scale surveys such as *Gaia*. As discussed in Section 2.2.1, its precise astrometry has enabled systematic searches for star clusters, which appear as overdensities in positional–kinematic phase space. Density-based clustering techniques are particularly well suited to this task. They can reveal groups of arbitrary shape, separate genuine clusters from noise, and do not require the number of clusters to be specified in advance. Consequently, density-based algorithms have become central tools for uncovering kinematically coherent stellar structures in *Gaia* data (e.g., Castro-Ginard et al. 2018; Cantat-Gaudin et al. 2018; Kounkel and Covey 2019; Hunt and Reffert 2021; Hunt and Reffert 2023; Ratzenböck et al. 2023; Ratzenböck et al. 2025).

In this section, I introduce the partitioning algorithm k -means and the density-based clustering algorithms DBSCAN and HDBSCAN, which are commonly used to identify stellar clusters, and discuss their limitations.

2.4.1. K -Means Clustering

K -means is a classical partitioning algorithm that divides the data into K clusters by minimizing the within-cluster sum of squared Euclidean distances (MacQueen 1967). Starting from an initial set of K centroids, the algorithm iterates between assigning each point to its nearest centroid and updating the centroids as the mean of their assigned points. Geometrically, this corresponds to a Voronoi tessellation of the input space into K convex regions. For non-overlapping, approximately equal-sized convex clusters, k -means is optimal with respect to this objective. In Galactic astronomy, k -means is mainly used as a building block in membership-cleaning methods such as UPMASK (Krone-Martins and Moitinho 2014) rather than for blind cluster searches. Because it assumes a fixed number of convex clusters of comparable size, k -means is generally less suitable than density-based methods for detecting star clusters with varying densities (Hunt and Reffert 2021).

2.4.2. DBSCAN

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is a clustering algorithm introduced by Ester et al. (1996) that groups data points according to regions of high local density. Unlike partitioning methods such as k -means, which assume convex cluster shapes and require the number of clusters as input, DBSCAN defines clusters as contiguous areas of high point density separated by regions of lower density. The algorithm relies on two parameters: ϵ , which defines the radius of a neighborhood around each point, and m_{pts} , the minimum number of points required within that neighborhood

to form a dense region. A point is considered a core point if it has at least m_{pts} neighbors within ϵ , while border points lie within the neighborhood of a core point but lack sufficient density themselves. Clusters are defined as non-empty, maximal sets of points that are density-connected, meaning that all points in the cluster are mutually reachable along paths that lie fully inside high-density regions and originate from at least one core point. Points not assigned to any cluster are treated as noise. DBSCAN is efficient for large databases, with an average run time complexity of $\mathcal{O}(n \log n)$. However, a key limitation of the method is its sensitivity to the choice of ϵ . Because only a single global density threshold can be used, DBSCAN cannot capture both compact and diffuse structures simultaneously.

2.4.3. HDBSCAN

HDBSCAN (The Hierarchical Density-Based Spatial Clustering of Applications with Noise; [Campello et al. 2013](#)) extends DBSCAN by eliminating the need for a single global density threshold. Instead of requiring a fixed ϵ value, it builds a hierarchy of clusters across all possible density levels, effectively exploring every possible DBSCAN solution in one run. The only parameter required is the minimum number of points per dense region, m_{pts} . The algorithm computes the core distance for each point, defined as the distance to its m_{pts} -nearest neighbor. From these values, it constructs a complete graph where each pair of points is connected by an edge whose weight is the mutual reachability distance. The mutual reachability distance is the maximum of their individual core distances and their pairwise distance. This measure replaces the global ϵ with a locally adaptive notion of distance, allowing comparison between regions of varying density and effectively pushing noise points to larger distances, so that they become more isolated from genuine clusters. From this graph, HDBSCAN builds a minimum spanning tree (MST) by progressively removing edges in decreasing order of mutual reachability distance. This tree reveals how clusters form, split, or disappear as the density threshold varies and represents all DBSCAN clusterings for every possible ϵ value.

To extract a meaningful and compact hierarchy, HDBSCAN introduces a minimum cluster size (m_{clSize} , typically equal to m_{pts}). Clusters smaller than this threshold collapse into noise, while true cluster splits occur when both resulting subclusters exceed this size. The stability of each cluster quantifies how long a cluster exists over the range of density thresholds before it breaks apart or vanishes and provides a criterion for selecting the most significant clusters. HDBSCAN obtains the final clustering by solving an optimization problem to maximize the total stability across all clusters. Its computational complexity scales as $\mathcal{O}(n^2)$. This hierarchical approach makes HDBSCAN less sensitive to parameter tuning and effective for data sets containing clusters of varying densities, which is especially useful for detecting stellar groups with different spatial concentrations.

2. Background

2.4.4. Limitations due to Projection Effects

Despite their strengths, density-based algorithms like DBSCAN and HDBSCAN face limitations when applied to nearby stars ($\lesssim 100$ pc). When clustering is performed in a 5D space that omits the full 3D velocity, either in the observed astrometric space $(\alpha, \delta, \varpi, \mu_{\alpha*}, \mu_{\delta})$ or in mixed spaces such as $(X, Y, Z, v_{\alpha}, v_{\delta})$, projection effects dominate. Stars sharing nearly identical true space velocities can exhibit large differences in proper motion across the sky, simply because the same velocity projects differently at different sky positions. This spread inflates the tangential-velocity dispersion and lowers the overdensity contrast needed for reliable clustering (Moranta et al. 2022).

Figure 2.4 illustrates this effect, showing how the dispersion of proper motions increases toward smaller distances for *Gaia* DR3 stars within 1 kpc (see Appendix A.1). The black marker and dashed lines indicate the distance cutoff (~ 112 pc), identified via knee detection, beyond which projection effects become sufficiently small for density-based clustering in five-dimensional spaces (e.g., $X, Y, Z, v_{\alpha}, v_{\delta}$) to be applied reliably. Below this distance, the projection geometry strongly shears the tangential velocities across the sky and violates the assumptions of density-based clustering, preventing robust identification of nearby associations.

2.5. Overcoming Projection Effects

The limitations discussed in the previous section suggest that clustering should be based directly on the true 3D velocities of stars, rather than on their projected tangential motions. Intuitively, if two stars are close to each other in space and their full velocity vectors are nearly identical, the probability that they belong to the same physical association is high. This perspective naturally leads to a distance-based formulation in which one defines a physically motivated pairwise distance between stars and then applies clustering algorithms to the resulting distance matrix.

Many clustering methods can be phrased in terms of pairwise dissimilarities. Given N stars and a distance function $d(i, j)$, one can construct a symmetric $N \times N$ matrix with entries $d(i, j)$ that encode how similar or dissimilar each pair of stars is. In a metric (Euclidean) space, the matrix of all pairwise distances already fixes the relative geometry of the data. Apart from shifting or rotating the entire configuration, everything is determined by the distances, so they contain all information relevant for clustering. Density-based and graph-based clustering algorithms may operate on this matrix or on the corresponding weighted graph instead of directly on coordinate vectors in a Euclidean feature space. In the present context, the key question becomes how to define a distance $d(i, j)$ that faithfully reflects dynamical similarity.

For stars with full 6D phase-space information, a natural choice is the norm of the velocity difference in Galactic Cartesian coordinates. Writing

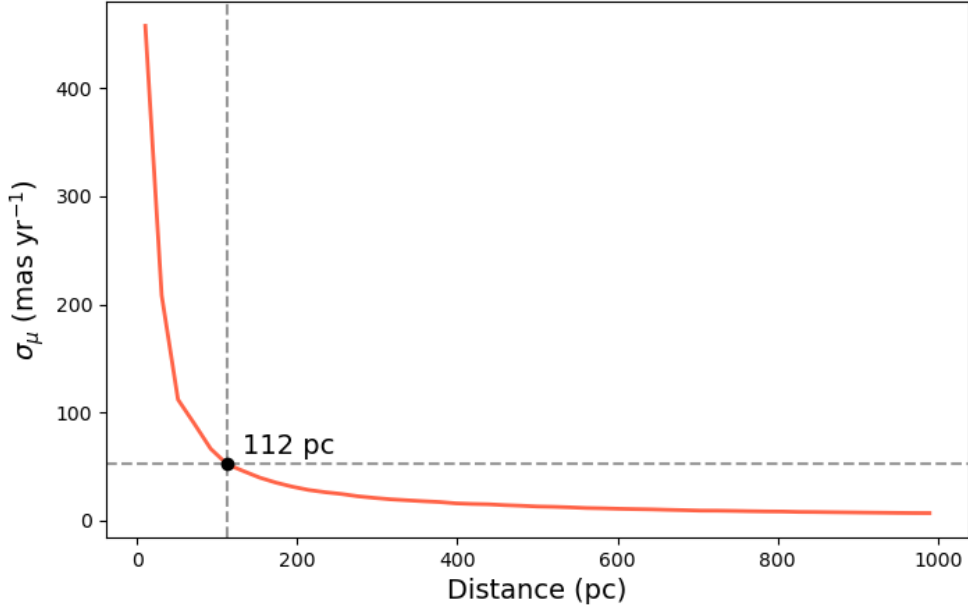


Figure 2.4.: Proper-motion dispersion as a function of distance for stars within 1 kpc from *Gaia* DR3. The black point and dashed lines indicate the adopted distance cutoff identified with `KneeLocator` (Arvai 2020). Below this distance projection effects significantly reduce the effectiveness of density-based clustering algorithms such as HDBSCAN (see Appendix A.1 for details).

$$\mathbf{v}_1 = (U_1, V_1, W_1)^T, \quad \mathbf{v}_2 = (U_2, V_2, W_2)^T,$$

the pairwise 3D velocity difference is

$$\|\Delta \mathbf{v}_{1,2}\| \equiv \|\mathbf{v}_1 - \mathbf{v}_2\| = \sqrt{(U_1 - U_2)^2 + (V_1 - V_2)^2 + (W_1 - W_2)^2}. \quad (2.1)$$

A clustering algorithm that operates on the matrix of all such $\|\Delta \mathbf{v}_{i,j}\|$ values would then group together stars that are both spatially close (through a separate position-based criterion) and kinematically similar, which is precisely the notion of a co-moving association. Moreover, this Euclidean norm is measured in km s^{-1} and can be directly compared to the expected internal velocity dispersions of coeval stellar populations, so it provides a physically meaningful scale for defining kinematic similarity.

In practice, however, most *Gaia* stars lack radial velocities, so their full (UVW) cannot be computed directly. What is available are sky positions and tangential velocities derived from proper motions. For a star at position $\mathbf{x} = (\alpha, \delta, \varpi)$, one can write its 3D velocity in Cartesian Galactic coordinates as a linear transformation of the proper motions and

2. Background

radial velocity,

$$\mathbf{v} = T(\mathbf{x}) \mathbf{q}, \quad (2.2)$$

with

$$\mathbf{q} = \begin{pmatrix} \mu_{\alpha*} \\ \mu_{\delta} \\ v_{\text{rad}} \end{pmatrix},$$

and $T(\mathbf{x})$ a 3×3 transformation matrix that depends on $\mathbf{x}$, the position on the sky, the distance, and the adopted ICRS-to-Galactic rotation. The first two components of $\mathbf{q}$ are measured by *Gaia*, while v_{rad} is unknown for the majority of stars.

Consider now two stars with positions $\mathbf{x}_1$ and $\mathbf{x}_2$ and measured proper motions $(\mu_{\alpha*,1}, \mu_{\delta,1})$ and $(\mu_{\alpha*,2}, \mu_{\delta,2})$. Their velocities can be written as

$$\mathbf{v}_1 = T(\mathbf{x}_1) \mathbf{q}_1, \quad \mathbf{v}_2 = T(\mathbf{x}_2) \mathbf{q}_2,$$

where the radial velocities $v_{\text{rad},1}$ and $v_{\text{rad},2}$ are the only unknown components of $\mathbf{q}_1$ and $\mathbf{q}_2$. The squared velocity difference between the two stars is then

$$\|\Delta \mathbf{v}\|^2 = \|T(\mathbf{x}_1) \mathbf{q}_1 - T(\mathbf{x}_2) \mathbf{q}_2\|^2, \quad (2.3)$$

which is a quadratic function of $v_{\text{rad},1}$ and $v_{\text{rad},2}$. This observation motivates the concept of *optimal radial velocities*, which solve for the smallest possible velocity difference between any two stars, given their positions and proper motions. For each pair of stars, one can choose $v_{\text{rad},1}$ and $v_{\text{rad},2}$ such that their 3D velocity difference is minimized. Formally, this corresponds to solving

$$\frac{\partial}{\partial v_{\text{rad},1}} \|\Delta \mathbf{v}\|^2 = 0, \quad \frac{\partial}{\partial v_{\text{rad},2}} \|\Delta \mathbf{v}\|^2 = 0,$$

which yields a linear 2×2 system for the optimal values of $(v_{\text{rad},1}, v_{\text{rad},2})$ that can be solved analytically. The resulting minimum,

$$d_{\text{opt}}(1, 2) \equiv \min_{v_{\text{rad},1}, v_{\text{rad},2}} \|\Delta \mathbf{v}\|, \quad (2.4)$$

can be interpreted as the smallest possible 3D velocity difference consistent with the observed proper motions and positions of the two stars.

Building on this idea, this thesis developed an optimal radial velocity function, which is called `pairwise_min_velocity_diff` and implements the above minimization, returning the pairwise minimal velocity differences $d_{\text{opt}}(i, j)$ for a given set of stars. In this thesis, I employ this method to construct distance matrices in which each entry encodes how similar the 3D velocities of two stars could be, given the constraints imposed by their *Gaia* astrometry. In regions where stars share genuinely similar space motions, $d_{\text{opt}}(i, j)$ becomes small, while for unrelated field stars, it is typically large.

2.5. Overcoming Projection Effects

This construction provides a physically motivated way to mitigate projection effects. Instead of clustering in distorted proper-motion space, one clusters in a space defined by minimal 3D velocity differences. However, it also introduces a subtle complication. The optimal radial velocities are determined *pairwise*. There is no guarantee that the set of distances $\{d_{\text{opt}}(i, j)\}$ arises from stars occupying a single common Euclidean velocity space with a consistent (UVW) for each star. As a consequence, the distance matrix generally does not define a proper metric space and can violate the triangle inequality, which for any three stars i, j, k requires $d(i, k) \leq d(i, j) + d(j, k)$. When these pairwise distances are fed directly into traditional clustering algorithms that implicitly assume an underlying Euclidean embedding, they do not produce stable or meaningful clusters. For example, in single-link hierarchical clustering, almost all data points tend to merge into a single extended cluster, because the pairwise optimization practically guarantees that each star has at least one other star with a very small optimal 3D velocity difference purely by chance. These chance neighbors are then linked into long chains across the data set.

These difficulties motivate the methods introduced in the next chapter. Broadly speaking, two complementary strategies have been proposed to exploit the optimal-velocity information while overcoming projection effects: (i) applying clustering schemes that operate directly on semi-metric similarity matrices and (ii) using deep-learning frameworks to learn an explicit 6D phase-space representation, with the optimal radial velocities providing part of the training signal.

3. Related Work

As described in the previous chapter, clustering nearby stars is fundamentally limited by projection effects and missing radial velocities. These limitations blur the intrinsic 3D phase-space structure of young stellar associations and prevent density-based algorithms from reliably separating clusters from the field. To overcome this, alternative strategies are needed that either work directly in semi-metric spaces implicitly defined through optimal-velocity computations or reconstruct an approximate 6D phase-space representation before clustering.

This chapter provides an overview of methods that address these challenges. The first part discusses approaches that operate directly on pairwise distance matrices, focusing on clustering algorithms designed for semi-metric spaces. In particular, it summarizes work carried out by my colleague Luiza, who applied the softmax clustering framework to the optimal-velocity distance matrix introduced in Section 2.5. The second part reviews deep learning and deep clustering approaches, which form the building blocks of the DENSE algorithm.

3.1. Clustering in Semi-Metric Spaces

One way to exploit the information contained in the optimal radial velocities is to treat the pairwise minimal velocity differences as a semi-metric and apply clustering algorithms that operate directly on the resulting distance matrix. These methods work with semi-metric similarity structures and can, in principle, accommodate the inconsistencies introduced by pairwise optimization.

In this section, I describe such an approach developed by my colleague Luiza during the course of this project. Her method utilizes the softmax clustering algorithm of Chang and Chang (2017) to cluster stars based on the optimal-velocity distance matrix, providing a direct baseline for comparison with the deep-learning framework introduced later in this chapter.

3.1.1. Softmax Algorithm

The softmax algorithm proposed by Chang and Chang (2017) is a probabilistic clustering method designed for data in semi-metric spaces, i.e., spaces where the distance function $d(x, y)$ is non-negative, symmetric, and vanishes on the diagonal, but does not necessarily satisfy the triangle inequality. This is precisely the situation encountered for the optimal-velocity distances introduced in Section 2.5, where pairwise radial velocities are adjusted

3. Related Work

independently for each star pair and the resulting distance matrix cannot, in general, be embedded in a single Euclidean space. Instead of operating directly on $d(x, y)$, the softmax framework first converts distances into a similarity measure, the *semi-cohesion*, and then clusters the data by grouping points that are strongly cohesive with each other.

Given a finite set $\Omega = \{x_1, \dots, x_n\}$ and a semi-metric $d(x, y)$, the associated semi-cohesion $\gamma(a, b)$ between two points $a, b \in \Omega$ is defined as

$$\gamma(a, b) = \frac{1}{n} \sum_{z_2 \in \Omega} d(z_2, b) + \frac{1}{n} \sum_{z_1 \in \Omega} d(a, z_1) - \frac{1}{n^2} \sum_{z_2 \in \Omega} \sum_{z_1 \in \Omega} d(z_2, z_1) - d(a, b). \quad (3.1)$$

This transformation turns dissimilarities into a notion of cohesion. Pairs that are far apart compared to the overall average distance have small or negative $\gamma(a, b)$, whereas pairs that are unusually close have large positive $\gamma(a, b)$. The values $\gamma(x_i, x_j)$ form a symmetric semi-cohesion matrix $\Gamma = (\gamma_{ij})$, which is the basic input to the softmax clustering procedure.

The algorithm maintains, for each point i and cluster label k , a probability $p_i(k)$ that expresses how strongly point i is associated with cluster k . These probabilities are initialized in a non-uniform way so that different clusters are represented from the start. At each iteration, the algorithm computes for every point and cluster an expected cohesion

$$z_i(k) = \sum_{j \neq i} \gamma_{ji} p_j(k),$$

which averages how strongly point i is connected to other points that currently belong to cluster k . If $z_i(k)$ is large and positive, point i fits well into cluster k , whereas negative values indicate that i is anti-correlated with that cluster.

Cluster memberships are then updated with a softmax rule. For an inverse temperature parameter $\theta > 0$, the unnormalized probabilities

$$\tilde{p}_i(k) = e^{\theta z_i(k)} p_i(k)$$

are computed and renormalized across clusters as

$$p_i(k) \leftarrow \frac{\tilde{p}_i(k)}{\sum_{\ell=1}^K \tilde{p}_i(\ell)}.$$

Clusters with higher expected cohesion receive more probability mass, while clusters with low or negative cohesion are suppressed. The parameter θ controls how sharply the algorithm favors the most cohesive cluster. For small θ the assignments remain diffuse, whereas for large θ the algorithm produces almost hard assignments, with each point putting most of its probability mass on a single cluster. In practice, θ is increased gradually during the iterations, so that the clustering becomes more confident over time

while still allowing the algorithm to escape poor initial configurations.

Under mild conditions, these updates increase a global cohesion objective at each step and converge to a local optimum. At convergence, most $p_i(k)$ are close to either 0 or 1, which yields an effective hard clustering, and the expected cohesions $z_i(k)$ provide an embedding of the data into a K -dimensional space that reflects the semi-metric structure. In the context of this thesis, the main advantage of the softmax algorithm is that it can operate directly on the semi-cohesion matrix derived from the optimal-velocity distances, without requiring a consistent Euclidean embedding or a strictly metric distance. In the following subsection, I summarize how this framework was implemented and applied to the challenging synthetic data set (see Section 2.3.2) by my colleague Luiza Krzepkowska.

3.1.2. Softmax Clustering of Optimal-Velocity Distances

Building on the optimal-velocity framework introduced in Section 2.5 and the softmax clustering algorithm described above, my colleague Luiza Krzepkowska implemented a complete clustering pipeline for the same challenging synthetic *Gaia*-like data set described in Section 2.3.2, working directly in semi-metric space. The data consist of five Gaussian clusters embedded in a large background population, with positions and velocities chosen to mimic stellar clusters in the Solar neighborhood. In this setup, radial velocities are treated as unknown and inferred pairwise by minimizing the 3D velocity difference between stars. The resulting minimal velocity differences define a semi-metric $d(i, j)$, from which the semi-cohesion matrix Γ is constructed according to Equation (3.1).

As the softmax algorithm scales poorly with sample size, Luiza adopted a divide-and-conquer strategy to make the method applicable to large samples and to reflect the spatial inhomogeneity of real *Gaia* data. The spatial domain was partitioned into grid cells, and the softmax clustering algorithm was run independently in each cell. This local application of softmax reduces the computational cost from a global $\mathcal{O}(N^2)$ dependence on the number of stars to roughly $\mathcal{O}(\sum_m n_m^2)$, where n_m is the number of stars in cell m , and limits the influence of distant, unrelated field stars on the local cohesion structure. To obtain multiple, overlapping clusterings for each star and to let stars interact with different neighbors beyond a single cell, she repeated this procedure for several shifted and rotated grid configurations. In different grid realizations, the same star shared a cell with different subsets of neighbors, so that physically related stars had multiple chances to be placed in the same softmax cluster at least once.

The multiple local solutions were then merged using consensus clustering techniques. For each pair of stars, a co-association matrix was formed that counts how often the two stars were assigned to the same cluster across all grid realizations. This co-association matrix was then clustered with standard consensus solvers (e.g., CSPA; Strehl and Ghosh 2003, NMF; Li et al. 2007, or MCLA; Strehl and Ghosh 2003) to obtain a final global clustering.

Applied to the challenging synthetic data set, this pipeline achieved a normalized mutual

3. Related Work

information (NMI) score of ≈ 0.9 . The final clusters closely traced the five input groups, and a substantial fraction of background stars was assigned to a background-dominated cluster. However, experiments showed that the choice of grid parameters (cell size, number of rotations and shifts, and the consensus solver) affected both the cluster purity and runtime. These findings demonstrate that clustering directly in a semi-metric space using softmax and consensus strategies can recover complex cluster structures from optimal-velocity distances under specific conditions.

3.1.3. Caveats

Despite its good performance on the synthetic tests, the softmax approach has important limitations for the present problem. A practical issue is scalability, since it relies on the full $N \times N$ distance (or semi-cohesion) matrix. Thus, both memory usage and runtime grow quadratically with the number of stars. The grid-based consensus strategy reduces this cost in practice, but at the expense of additional complexity and several tunable design choices (grid geometry, cell size, number of rotations and shifts, and consensus solver), all of which can influence the final clustering. Because clustering is unsupervised, these grid and consensus parameters cannot be tuned against a clear target and are not very intuitive. Therefore, finding a good configuration in practice is often tedious and requires extensive experimentation.

3.2. Deep Learning

The semi-metric approach described in the previous section treats the optimal-velocity distances as the primary objective for clustering. A complementary strategy is to use these distances, together with the observed astrometry, as constraints for learning a six-dimensional representation for all stars directly with deep neural networks.

Although clustering in 6D phase space (XYZ positions plus UVW velocities) would be ideal, it requires knowledge of stellar radial velocities, which are unavailable for a large fraction of nearby stars. Traditional density-based clustering algorithms such as DBSCAN or HDBSCAN can therefore only be applied to a subset of stars with measured radial velocities (as, for example, in [Moranta et al. 2022](#)), limiting the completeness of current cluster solutions. Deep neural networks offer a way to overcome this limitation. They can learn to infer missing quantities such as radial velocities, and thus the full 3D space velocities, from the available astrometric parameters (positions, parallaxes, and proper motions). Once such a latent 6D phase-space representation has been learned, clustering can be performed directly in this embedding and only requires distance matrices for mini-batches or local subsets of stars. This avoids the need to construct and store the full $N \times N$ distance matrix and therefore reduces the memory complexity of the overall approach. By mapping incomplete observational data into a learned latent representation that approximates the full 6D phase space, deep learning enables clustering analyses to

be extended to much larger stellar samples.

In this section, I briefly introduce the basic principles of neural networks and explain how autoencoders can be used to learn physically meaningful representations of stellar phase-space data.

3.2.1. Neural Networks

Neural networks are computational models inspired by the structure and functioning of biological neural systems (see, e.g., [Goodfellow et al. 2016](#) for a detailed introduction). They consist of interconnected layers of nodes, or neurons, that process information by transmitting and transforming signals between layers. The most common type is the feedforward neural network (FFN), also known as a multilayer perceptron (MLP), which learns a mapping between inputs and outputs:

$$\mathbf{y} = f(\mathbf{x}; \boldsymbol{\theta}),$$

where $\mathbf{x}$ represents the input, $\mathbf{y}$ the output, and $\boldsymbol{\theta}$ the set of learnable parameters (weights and biases).

A typical FFN consists of an input layer, one or more hidden layers, and an output layer, with the number of layers defining the network's depth. Each hidden layer applies a linear transformation followed by a non-linear activation function. The activations in layer l are computed as

$$\mathbf{h}^{(l)} = g\left(W^{(l)\top} \mathbf{h}^{(l-1)} + \mathbf{b}^{(l)}\right),$$

where $W^{(l)}$ and $\mathbf{b}^{(l)}$ are the weight matrix and bias vector of layer l , $\mathbf{h}^{(l-1)}$ is the output of the previous layer, and $g(\cdot)$ is an activation function such as the rectified linear unit (ReLU), hyperbolic tangent (tanh), or logistic sigmoid. Activation functions introduce non-linearity, which is essential for modeling complex relationships between inputs and outputs. Intuitively, they can be thought of as mechanisms that determine which information is passed on to subsequent layers, similar to how biological neurons selectively respond to certain inputs.

Training a neural network involves optimizing the parameters $\boldsymbol{\theta} = \{W, \mathbf{b}\}$ to minimize a loss function $\mathcal{L}(\boldsymbol{\theta})$ that quantifies the difference between the model predictions $\hat{\mathbf{y}}$ and the target values $\mathbf{y}$. The parameters are updated iteratively using gradient-based optimization methods, such as stochastic gradient descent (SGD) or adaptive variants like Adam ([Kingma and Ba 2017](#)). In these methods, the gradient of the loss with respect to each parameter, $\nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta})$, indicates the direction of steepest increase of the loss. The parameters are then adjusted in the opposite direction to reduce the loss:

$$\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t)} - \eta \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}^{(t)}),$$

3. Related Work

where η is the learning rate controlling the step size, and the superscript t denotes the training iteration (or epoch). In practice, the loss is computed over small subsets of the data, called minibatches, rather than the entire data set, which makes the optimization computationally efficient.

The gradients used in these updates are computed using the backpropagation algorithm, which applies the chain rule of calculus to propagate errors backward through the network. In each layer, backpropagation determines how the loss changes with respect to the weights and biases of the layer. These local derivatives are then accumulated to obtain the gradients for all network parameters (Goodfellow et al. 2016).

Although feedforward networks are commonly used for supervised learning tasks such as regression and classification, the same principles apply to unsupervised architectures such as autoencoders.

3.2.2. Autoencoder

An autoencoder is an unsupervised type of neural network designed to learn a hidden representation of the input data without requiring external labels (Makhzani and Frey 2013; Goodfellow et al. 2016). It consists of two components: an encoder, which maps the input $\mathbf{x} \in \mathbb{R}^d$ to a latent representation $\mathbf{z} \in \mathbb{R}^m$, and a decoder, which reconstructs the input from this latent code:

$$\mathbf{z} = f_{\theta}(\mathbf{x}), \quad \hat{\mathbf{x}} = g_{\phi}(\mathbf{z}),$$

where θ and ϕ denote the parameters of the encoder and decoder networks, respectively.

The network is trained by minimizing a reconstruction loss function, typically the mean squared error (MSE),

$$\mathcal{L}_{\text{rec}} = \|\mathbf{x} - \hat{\mathbf{x}}\|^2,$$

which encourages the decoded output $\hat{\mathbf{x}}$ to approximate the original input $\mathbf{x}$ as closely as possible (Kramer 1991; Makhzani and Frey 2013; Goodfellow et al. 2016).

The main interest lies not in the reconstruction $\hat{\mathbf{x}}$, but in the latent representation $\mathbf{z}$, which can reveal interesting structures hidden in the input data. When the latent dimension m is smaller than the input dimension d , the model is said to be undercomplete. The reduced latent space forces the network to encode only the most informative aspects of the input. Conversely, an overcomplete autoencoder ($m > d$) possesses greater representational capacity, which can be useful for capturing features not explicitly present in the data but implied by the underlying structure (Ranzato et al. 2007; Makhzani and Frey 2013; Goodfellow et al. 2016).

To prevent the network from only memorizing its input, additional constraints can be imposed on the latent space. Such regularization strategies, including sparsity constraints, smoothness penalties, and probabilistic priors, encourage the model to capture meaningful structures in the data rather than reproduce the input trivially. There are numerous specialized variants of autoencoders, for example, denoising autoencoders (DAEs; Vincent et al. 2008), which are trained to reconstruct clean inputs from corrupted versions, or variational autoencoders (VAEs; Kingma and Welling 2014), which introduce a probabilistic framework in the latent space to enable generative modeling (see, e.g., Goodfellow et al. 2016).

In the present work, the autoencoder is trained to learn a nonlinear mapping between observed 5D astrometric quantities $(\alpha, \delta, \varpi, \mu_{\alpha*}, \mu_{\delta})$ and a latent representation that approximates the full 6D Galactic phase space $(XYZUVW)$. Within this latent space, stars that are spatially or kinematically associated tend to form compact, overdense regions relative to the diffuse background population. The resulting representation thus captures the intrinsic structure of the stellar distribution and serves as a basis for subsequent clustering analyses, either performed separately in a two-step pipeline or integrated directly within deep clustering frameworks.

3.2.3. Deep Clustering

Deep clustering combines representation learning with unsupervised cluster discovery by integrating an autoencoder and a clustering objective into a single training framework. The key idea is to learn a latent space that both reconstructs the input data and reveals its intrinsic cluster structure (Xie et al. 2016; Guo et al. 2017; Yang et al. 2017).

In this section, I present an overview of commonly used centroid-based deep clustering methods and introduce SHADE, a novel density-based technique.

Centroid-Based Deep Clustering

Most existing deep clustering frameworks, such as Deep Embedded Clustering (DEC; Xie et al. 2016), Improved Deep Embedded Clustering (IDEC; Guo et al. 2017), and Deep Clustering Network (DCN; Yang et al. 2017), use centroid-based objectives inspired by k -means. These methods optimize a clustering loss jointly with the autoencoder reconstruction loss, iteratively refining cluster centroids in the latent space. The clustering objective minimizes the distance between each latent representation $\mathbf{z}_i$ and its assigned cluster center $\boldsymbol{\mu}_k$, thus enforcing compact, convex clusters:

$$\mathcal{L}_{\text{clust}} = \sum_i \|\mathbf{z}_i - \boldsymbol{\mu}_{q_i}\|^2,$$

where $\mathbf{z}_i$ denotes the latent representation of the i -th data point, and $\boldsymbol{\mu}_{q_i}$ is its assigned cluster centroid. This approach enforces compact, roughly spherical clusters and ensures

3. Related Work

that the latent space is optimized for separability.

Although this formulation works well for data sets where clusters are approximately spherical and well-separated, it imposes strong geometric constraints. Centroid-based methods favor uniform cluster densities and tend to distort density-based relationships by pulling points toward the nearest cluster center, thereby erasing the original topological structure of the data. As a result, they are unable to recover overlapping or elongated clusters. These approaches also require the number of clusters K to be specified in advance, a parameter that is often unknown in exploratory settings. When the intrinsic structure of the data is uncertain or evolving, this assumption constrains the method’s ability to reveal unexpected or subtle substructure.

Density-Based Deep Clustering with SHADE

The approach developed in this thesis draws on SHADE (Structure-preserving High-dimensional Analysis with Density-based Exploration; [Beer et al. 2024](#)), a recently proposed deep density-based clustering method designed to recover arbitrarily shaped structures in high-dimensional and noisy data. Density-based algorithms excel at identifying non-convex and intertwined clusters, yet extending these methods to high-dimensional spaces is challenging because of the empty-space problem, where distances become less informative as the dimensionality increases.

SHADE addresses these limitations by optimizing a deep autoencoder with two complementary objectives: a reconstruction loss $\mathcal{L}_{\text{rec}}$, which preserves the global geometry of the input space, and a density-connectivity loss $\mathcal{L}_{\text{d}}$, which preserves the density-based structure of the data. The density-connectivity term relies on the tree metric $d_{\text{dc}}(x_i, x_j)$, defined as the path length between points i and j on the minimum spanning tree (MST) constructed from mutual-reachability distances. This metric captures connectivity through high-density regions and is robust to non-convex geometries.

The density-connectivity loss encourages latent distances to reflect this density structure:

$$\mathcal{L}_{\text{d}} = \sum_{i,j} (d_{\text{dc}}(x_i, x_j) - \|\mathbf{z}_i - \mathbf{z}_j\|_2)^2,$$

where $\mathbf{z}_i$ and $\mathbf{z}_j$ denote the latent representations of points i and j , and $d_{\text{dc}}(x_i, x_j)$ is their density-connectivity distance on the MST. Minimizing $\mathcal{L}_{\text{d}}$ compresses points that are density-connected and separates points from different density components, enabling the latent space to retain the structure of elongated, intertwined, or multimodal clusters.

A major advantage of SHADE is that cluster discovery is fully automatic. Instead of requiring a predefined number of clusters, SHADE extracts a cluster hierarchy in the embedding space and identifies stable groups using a density-based stability criterion. This procedure isolates consistent cluster cores and naturally identifies noise points without the

need for manual thresholds or hyperparameters controlling the expected number of clusters.

Experiments reported in [Beer et al. \(2024\)](#) show that SHADE recovers complex non-convex structures more reliably than centroid-based deep clustering methods and remains robust across a wide range of noise levels. Although centroid-based models may be more appropriate when the underlying data are dominated by Gaussian-like clusters, SHADE excels when cluster boundaries are governed by density connectivity rather than convex geometry.

In the context of nearby stellar associations, the strengths of SHADE are highly attractive. Stellar groups in the Solar neighborhood can be elongated, partially overlapping, and are embedded in a noisy background, so a density-based latent representation is more appropriate than one based on convex clusters. However, SHADE in its original form operates on generic feature spaces and does not explicitly exploit the geometric structure of astrometric data or the semi-analytic information contained in the optimal-velocity distances.

These considerations motivate the development of the DENSE algorithm presented in the next chapter. DENSE adopts the idea of learning a density-aware latent space from SHADE but combines it with analytic coordinate transformations and the optimal-velocity framework. The goal is to obtain an explicit 6D Galactic Cartesian representation for each star that respects both the astrometric constraints and the pairwise velocity information. In contrast to SHADE, which incorporates a density-connectivity tree metric directly into the training loss and derives the clustering from this learned density structure, DENSE separates these stages. It uses the optimal-velocity distances as kinematic constraints to learn a physically interpretable 6D phase-space embedding and then applies HDBSCAN in the trained latent space as a post-processing step to identify stellar associations.

4. DeNSE Algorithm

As discussed in Chapters 2 and 3, projection effects and missing radial velocities make it difficult to identify nearby stellar associations. For stars within ~ 100 pc, the same 3D space velocity can project into very different proper motions across the sky, whereas stars with unrelated motions may appear kinematically similar in the projected coordinates. The DeNSE algorithm (Deep Semi-analytic Embedding of the Nearby Stellar Environment) is designed to address these issues by combining a deep neural network with analytic coordinate transformations. It uses optimal-velocity information as a constraint to learn an explicit 6D Galactic Cartesian representation for each star. In this learned latent space, stars that share similar 3D positions and velocities form compact overdensities that can be recovered with standard density-based clustering algorithms such as HDBSCAN (Section 2.4.3).

In this chapter, I describe the main components of the DeNSE algorithm, namely the construction of training batches in mixed spatial–kinematic space, the neural network architecture, the loss functions that tie the latent space to the observable phase space, and the clustering step applied to the learned representation.

4.1. Problem Formulation and Data Representation

The goal of DeNSE is to learn, for each star, a 6D latent vector

$$\hat{\mathbf{z}}_i = (\hat{X}_i, \hat{Y}_i, \hat{Z}_i, \hat{U}_i, \hat{V}_i, \hat{W}_i),$$

that can be interpreted as an approximation to its Galactic Cartesian phase-space coordinates, given only the observed *Gaia*-like astrometry and, where available, radial velocities. The input to the model consists of a set of N stars with five- or six-parameter astrometric measurements

$$\mathbf{x}_i = (\alpha_i, \delta_i, \varpi_i, \mu_{\alpha*,i}, \mu_{\delta,i}, v_{\text{rad},i}),$$

where $v_{\text{rad},i}$ is missing for the majority of stars. For the synthetic data sets introduced in Section 2.3 the ground-truth Galactic Cartesian coordinates

$$\mathbf{z}_i = (X_i, Y_i, Z_i, U_i, V_i, W_i)$$

are known and are used for evaluation and, in some experiments, as additional supervision.

4. DENSE Algorithm

From a machine-learning perspective, DENSE seeks a mapping

$$f_\theta : \mathbf{x}_i \mapsto \hat{\mathbf{z}}_i$$

implemented by a neural encoder with parameters θ , together with a fixed analytic decoder

$$g_{\text{ana}} : \hat{\mathbf{z}}_i \mapsto \hat{\mathbf{x}}_i$$

that converts Galactic Cartesian phase space back to ICRS astrometric quantities using the transformations described in Section 2.2.3. The function g_{ana} is implemented as a differentiable analytic mapping in PYTORCH, so that gradients can be backpropagated through the coordinate transformations during training. Together, f_θ and g_{ana} define a mapping $\mathbf{x}_i \mapsto g_{\text{ana}}(f_\theta(\mathbf{x}_i))$ from observables to latent phase space and back.

A key difficulty arises from the missing radial velocities. For most stars, the full 3D velocity vector cannot be reconstructed uniquely from $\mathbf{x}_i$. Instead, DENSE exploits pairwise information derived from the optimal-velocity framework of Section 2.5. For a pair of stars (i, j) , this framework defines radial velocities $v_{\text{rad},i}$ and $v_{\text{rad},j}$ that minimize the norm of their 3D velocity difference in Galactic Cartesian space. The value of this minimum yields the optimal-velocity distance

$$d_{\text{opt}}(i, j) = \min_{v_{\text{rad},i}, v_{\text{rad},j}} \|\mathbf{v}_i - \mathbf{v}_j\|,$$

and the collection of all such distances forms a symmetric semi-metric matrix over the full sample,

$$D_{\text{opt}} = (d_{\text{opt}}(i, j))_{1 \leq i, j \leq N},$$

which encodes how similar the underlying 3D velocities of two stars *could* be, given their observed astrometry. Small values of $d_{\text{opt}}(i, j)$ indicate that two stars can have nearly identical space motions for some choice of radial velocities, whereas large values imply that their velocities must differ substantially, regardless of the missing line-of-sight information.

In DENSE, D_{opt} provides additional constraints on the latent velocity representation. The velocity components

$$\hat{\mathbf{z}}_i^{\text{UVW}} = (\hat{U}_i, \hat{V}_i, \hat{W}_i)$$

are encouraged to be similar for pairs with small $d_{\text{opt}}(i, j)$ and dissimilar for pairs with large $d_{\text{opt}}(i, j)$. In addition, for the subset of stars with measured radial velocities, the line-of-sight velocity implied by $g_{\text{ana}}(\hat{\mathbf{z}}_i)$ is tied to the observed $v_{\text{rad},i}$. At the same time, the position components $(\hat{X}_i, \hat{Y}_i, \hat{Z}_i)$ are constrained by the requirement that the decoded astrometry $g_{\text{ana}}(\hat{\mathbf{z}}_i)$ reproduces the observed positions and proper motions. Training thus amounts to finding parameters θ such that the decoded astrometric quantities match the *Gaia*-like measurements, while the geometry of the latent velocity space is consistent with the optimal-velocity distances. The concrete loss terms used to enforce these constraints are described in Section 4.4.

4.2. Batch Construction in Mixed Spatial–Kinematic Space

Meaningful batching is particularly important for the present learning task. Ideally, each batch should contain stars that are close to one another in both position and velocity, so that their pairwise relations carry information about local structure. Where possible, members of the same cluster should appear in the same batch. This is especially relevant for *Gaia*-like samples, which are dominated by background stars. If cluster members almost never co-occur in the same batch, the network cannot reliably learn their mutual relationships. To address this, I developed a dynamic batching strategy, implemented in the custom iterable data set `KmeansSubset6D`, that combines trial 3D velocities, analytic radial-velocity inference, and constrained k -means clustering.

The basic idea is to process each star multiple times during training under different assumed 3D motions, while ensuring that, in any given realization, stars are grouped into spatially and kinematically coherent batches of fixed size M . To this end, a Gaussian distribution in Galactic velocity space (UVW) is constructed from the subset of stars with known full phase space. For each training epoch $e \in \{1, \dots, E\}$ and for each repeat $k \in \{1, \dots, K\}$, a trial velocity vector

$$(U_k, V_k, W_k)$$

is drawn from this distribution. Given this trial vector, `vr_solver` computes, for every star, the radial velocity that minimizes the squared difference between the star’s inferred Galactic velocity and the trial velocity. Together with the observed astrometry, this yields a complete 6D phase-space realization for each star in ICRS. These realizations are then transformed into Galactic Cartesian coordinates using the transformations described in Section [2.2.3](#).

Batch formation proceeds by clustering in these transformed ($XYZUVW$) coordinates with the constrained k -means algorithm `KMeansConstrained` (Levy-Kramer 2018), a variant of k -means that enforces user-specified bounds on cluster sizes. Before clustering, the velocity components are rescaled by a constant factor so that the typically smaller spread in (UVW) becomes comparable to that in (XYZ). This prevents the Euclidean distance used by k -means from being dominated by positions or velocities. The number of clusters (batches) is chosen so that each cluster (batch) contains exactly M stars, ensuring that the clustering induces a partition of the data into balanced, non-overlapping batches for the given repeat (e, k) . Within a single repeat, each star appears in exactly one batch. Across the K repeats in an epoch, every star is seen K times in different local neighborhoods.

This batching strategy combines repeated processing of each star under different trial 3D velocities with analytic inference of missing radial velocities for each trial and constrained clustering into size-controlled local groups. As a result, the autoencoder is trained on batches that represent locally coherent regions of phase space under multiple trial velocity

4. DENSE Algorithm

realizations, rather than on random mixtures of unrelated stars.

The behavior of this batching scheme can be illustrated on the simple and challenging synthetic data sets introduced in Section 2.3. For a given trial velocity (U_k, V_k, W_k) , the procedure described above produces a partition of the stars into $B = N/M$ batches, which can be visualized by coloring the stars according to their batch assignment in position-velocity space.

Figure 4.1 shows the resulting batch assignments for the simple data sample using six different trial velocities. In the first three rows, the trial velocity is set exactly to the true mean velocity of each cluster, followed by three trials with randomly drawn velocities from the Gaussian distribution in (UVW) fitted to the subset of stars with known UVW velocities. In each row, the left panel displays the stars in the X – Y plane, and the right panel in the U – V plane, with colors indicating batch IDs. The text below each row summarizes, for each true cluster C_i , which batch B_j contains the largest fraction of its members and what that completeness fraction is. When the trial velocity matches the true cluster mean, all stars of that cluster fall into a single batch, as intended. For random trial velocities, the behavior is more variable, but most stars from each cluster still end up in one dominant batch.

The same experiment for the more realistic data sample is shown in Figure 4.2. Here, the clusters are more extended and partially overlapping, making the assignment problem intrinsically harder than in the simple case. When the trial velocity matches the true mean motion of a cluster, its members are still typically grouped into a single dominant batch. For randomly drawn UVW trials, the exact batch boundaries vary from run to run, because each trial-velocity draw slightly perturbs the inferred 6D vectors. Nevertheless, the method consistently groups stars into locally coherent, cluster-dominated batches. In practice, the batch size M is treated as a training hyperparameter and should be chosen to be on the order of the expected cluster size so that associations can be represented within dominant batches whenever possible.

To quantify how sensitively the batching depends on the choice of trial velocity, I computed, for each cluster, how its members are distributed across batches as the trial velocity shifts away from the true mean. Starting from the true mean velocity of a given cluster, the trial velocity is displaced along a fixed direction in (UVW) by an offset $\Delta\vec{v}$ with magnitude $|\Delta\vec{v}|$ between 0 and 20 km s^{−1}. For each offset, the optimal radial velocities are recomputed, the stars are re-clustered with `KMeansConstrained`, and two summary statistics are evaluated:

- the *recovery fraction*, defined as the fraction of cluster members contained in the single batch that contains the largest number of stars from that cluster; and
- the *entropy* of the cluster’s batch distribution, defined as the Shannon entropy of the normalized counts of its members over all batches. The entropy is zero if all

4.2. Batch Construction in Mixed Spatial–Kinematic Space

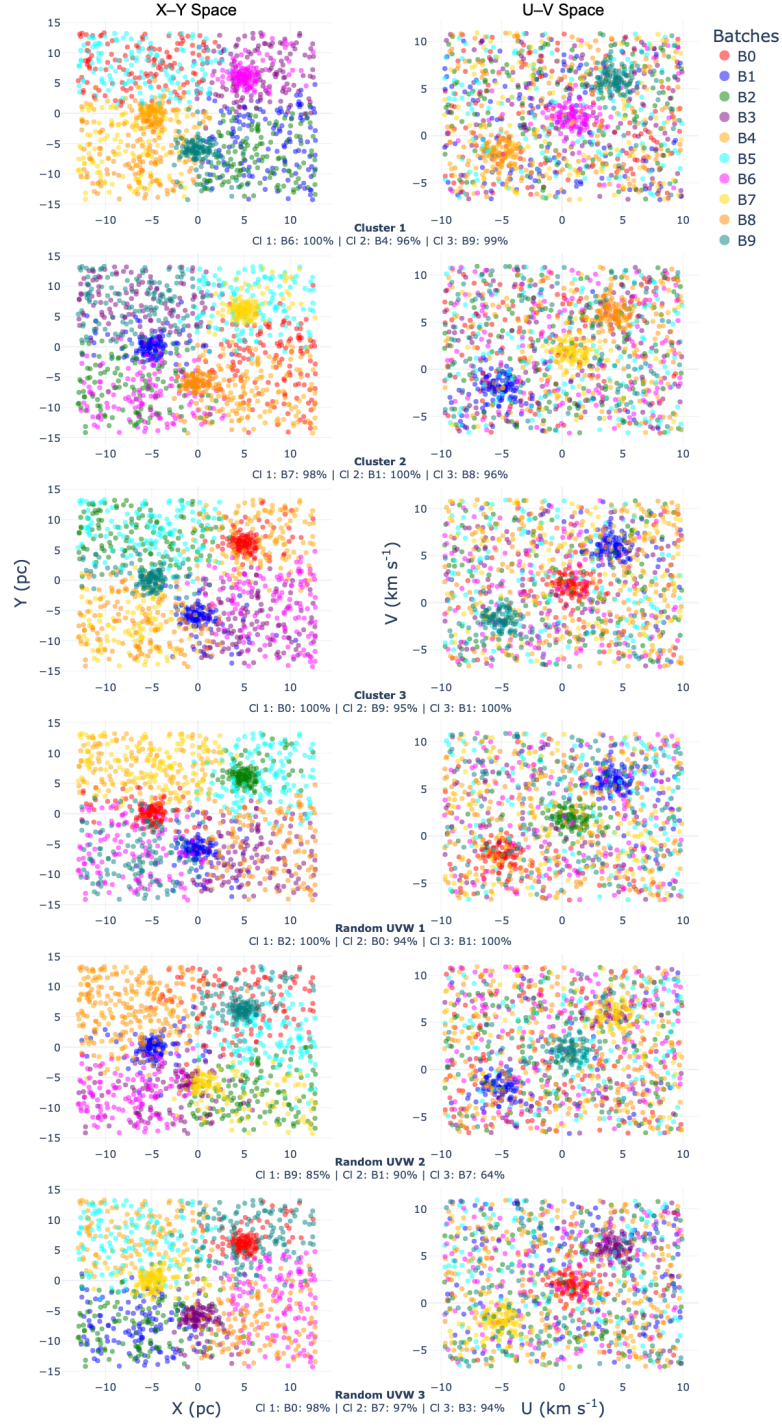


Figure 4.1.: Batch construction for the simple data sample for different trial UVW values. Each row corresponds to one trial velocity and shows the stars in X - Y (left) and U - V (right) space, colored by batch ID.

4. DENSE Algorithm

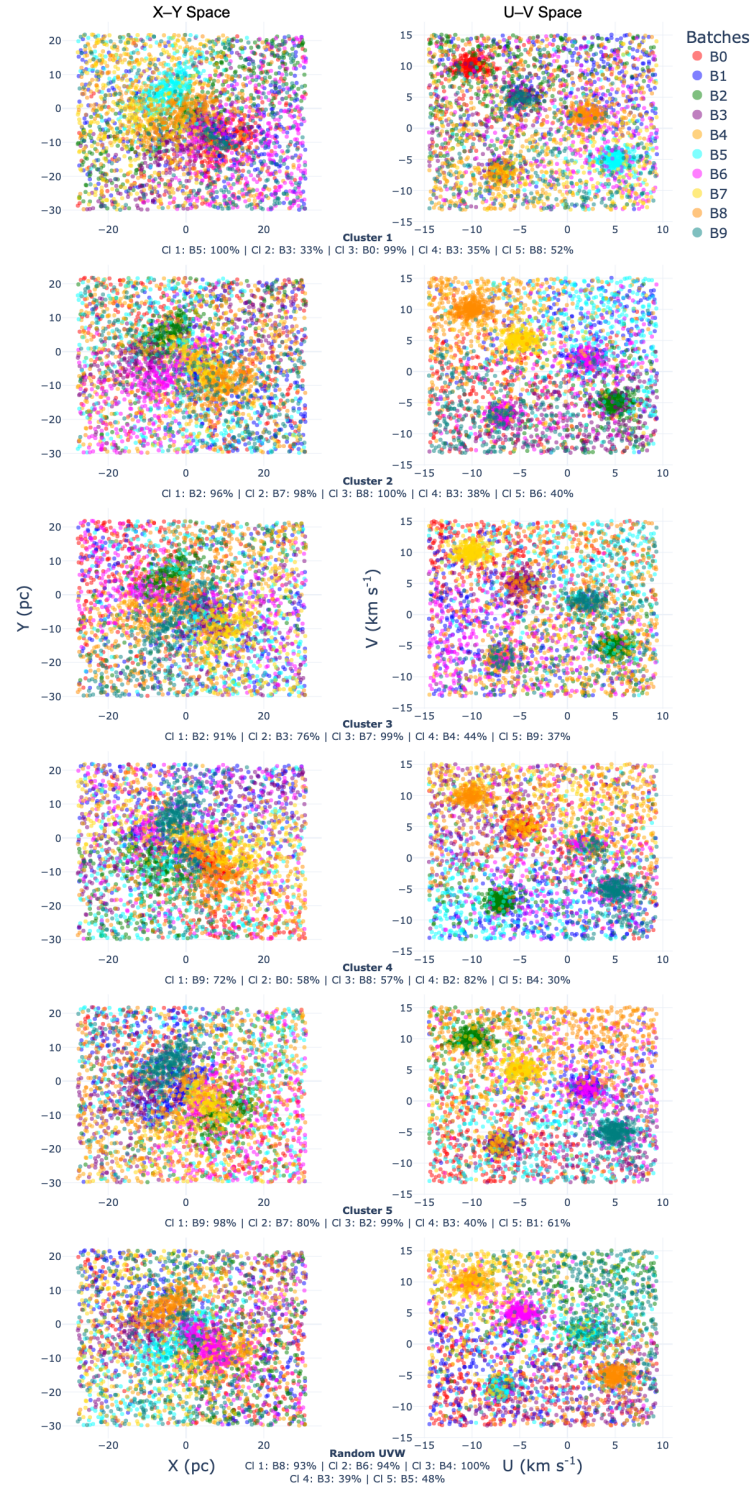


Figure 4.2.: Same as Figure 4.1, but for the challenging data sample with five clusters.

4.3. Semi-Analytic Autoencoder Architecture

members of the cluster lie in a single batch and increases toward its maximum as the members are spread more evenly over many different batches.

Figure 4.3 shows the recovery fraction (left panels) and the entropy (right panels) as a function of $|\Delta\vec{v}|$ for the simple (top row) and challenging (bottom row) samples. As expected, when the trial velocity matches the true mean ($|\Delta\vec{v}| \approx 0$), the recovery fraction is close to unity, and the entropy is low for the simple sample, indicating that almost all cluster members are assigned to a single batch. As $|\Delta\vec{v}|$ increases, the recovery fraction decreases and the entropy increases, reflecting that cluster members become increasingly spread over multiple batches.

For the challenging sample, two of the clusters never reach a recovery fraction of unity, even when $|\Delta\vec{v}| \approx 0$. Their maximum values are only ~ 0.6 – 0.7 . This reflects their extended and overlapping structure, which causes them to be split across neighboring batches even under an optimal trial velocity. Nevertheless, the same qualitative trend is present as in the simple case: larger offsets $|\Delta\vec{v}|$ systematically lower recovery fractions and increase entropies.

The choice of shift direction in velocity space is illustrated in Figure 4.4. The filled circles mark the true (U, V) centroids of the clusters, and the arrows indicate the direction along which the trial velocity is displaced. The squares show the shifted centroids for a representative offset of $|\Delta\vec{v}| = 10 \text{ km s}^{-1}$. These vectors correspond to the shifts used to generate the recovery-fraction and entropy curves in Figure 4.3. Empirically, repeating the experiment with different shift directions did not reveal any systematic dependence of the results on the chosen direction.

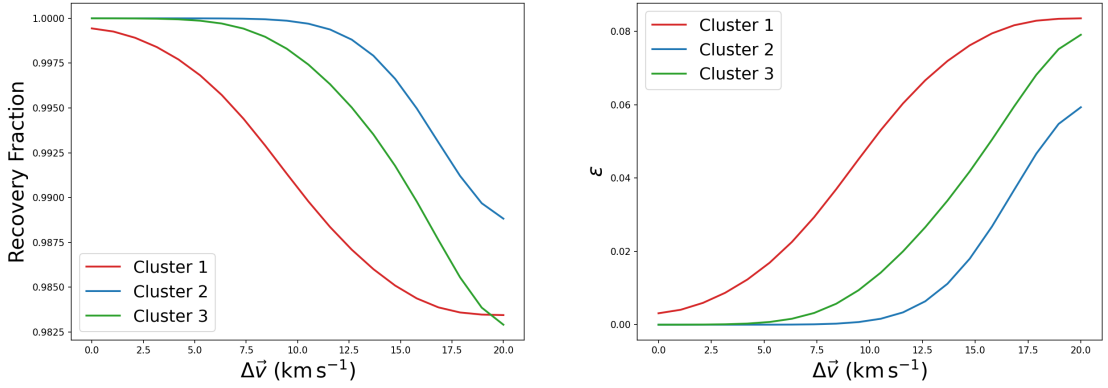
As a complementary view, Figure 4.5 shows, for the case $|\Delta\vec{v}| = 0 \text{ km s}^{-1}$, how the stars of each true component are distributed across batches. For each cluster C_i , the trial velocity is set equal to its true mean UVW , so the panels do not correspond to a single shared batching run but to separate runs optimized for each component. The resulting distributions are shown for the background and for each cluster, with the bars indicating the number of stars assigned to each batch ID. In both the simple and challenging samples, the background population is spread relatively uniformly over all batches, whereas each cluster exhibits a pronounced dominant batch (or a small set of dominant batches).

4.3. Semi-Analytic Autoencoder Architecture

DENSE is implemented as a semi-analytic autoencoder (see Fig. 4.6 for an overview of the baseline architecture). The encoder is a feedforward neural network that maps observed astrometry to a 6D latent vector interpreted as Galactic Cartesian phase space, while the decoder is replaced by the analytic coordinate transformations introduced in Section 2.2.3. These transformations are implemented in an autodifferentiable way, ensuring that gradients can be propagated through the full mapping during training. In this sense, only

4. DENSE Algorithm

(a) Simple data sample



(b) Challenging data sample

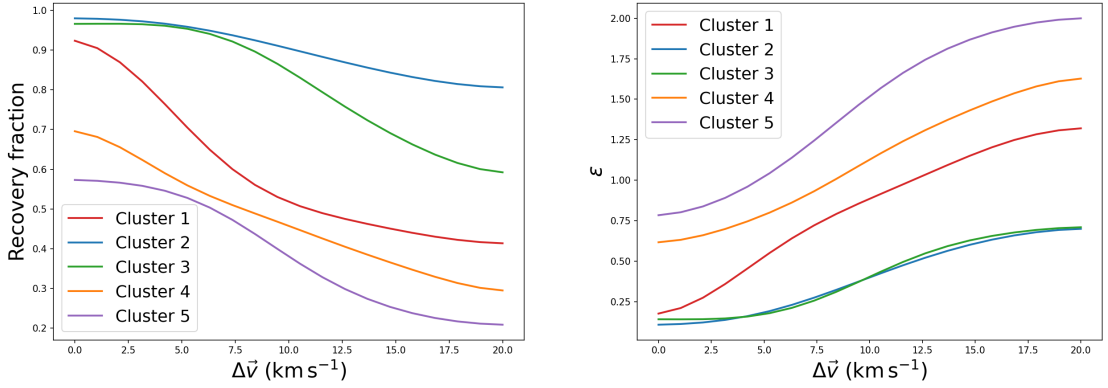


Figure 4.3.: Recovery fraction (left) and entropy (right) as a function of the offset $|\Delta\vec{v}|$ between the trial and true cluster velocities for the simple (top row) and challenging (bottom row) data samples.

the encoder is learned, whereas the mapping back to observable space is fully deterministic.

The astrometric measurements $\mathbf{x}_i$ are first standardized component-wise by

$$\mathbf{x}_i^{\text{std}} = \frac{\mathbf{x}_i - \boldsymbol{\mu}_{\text{icrs}}}{\boldsymbol{\sigma}_{\text{icrs}}},$$

where $\boldsymbol{\mu}_{\text{icrs}}$ and $\boldsymbol{\sigma}_{\text{icrs}}$ denote the mean and standard deviation of the six astrometric parameters in the training set. The radial-velocity entry of $\mathbf{x}_i^{\text{std}}$ is used only in the loss terms for stars with measured $v_{\text{rad},i}$. The encoder itself operates on the 5D subvector without radial velocity,

$$\mathbf{x}_i^{(5\text{D})\text{std}} = (\alpha_i, \delta_i, \varpi_i, \mu_{\alpha*,i}, \mu_{\delta,i})^{\text{std}}.$$

The encoder is a multilayer perceptron (MLP) with two hidden layers of widths 128 and

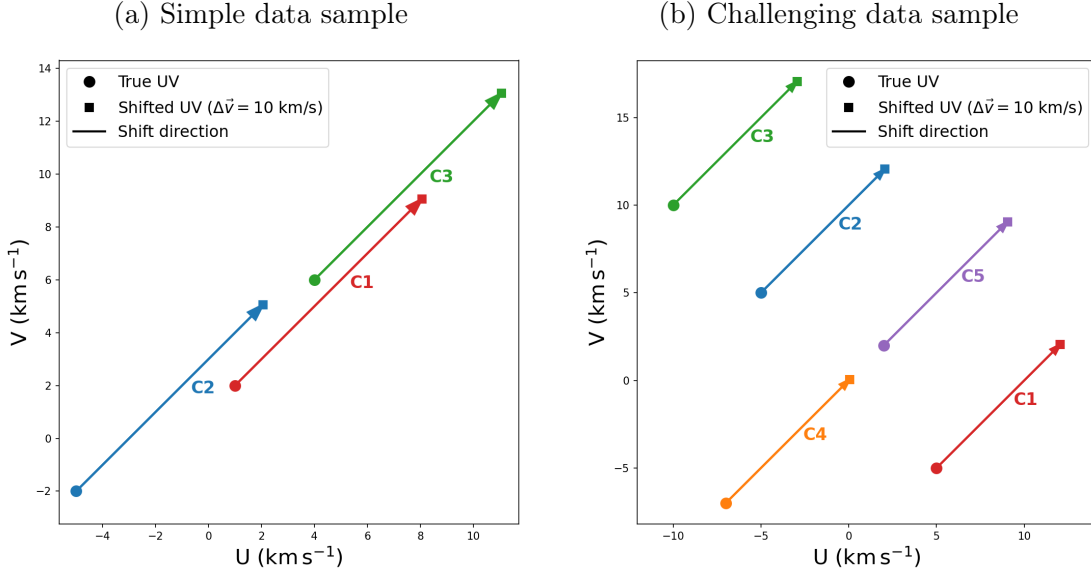


Figure 4.4.: True cluster centroids in the U - V plane (circles), shifted centroids for a representative offset $|\Delta\vec{v}| = 10$ km s⁻¹ (squares), and the corresponding shift directions (arrows) for the respective samples. These shifts are used to generate the curves in Figure 4.3.

64 that maps the standardized input to a standardized 6D latent vector,

$$\hat{\mathbf{z}}_i^{\text{std}} = f_{\theta}(\mathbf{x}_i^{(5\text{D})\text{std}}),$$

constructed by stacking linear layers with rectified linear unit (ReLU) activations,

$$\mathbf{h}^{(1)} = \text{ReLU}(W^{(1)}\mathbf{h}^{(0)} + \mathbf{b}^{(1)}),$$

$$\mathbf{h}^{(2)} = \text{ReLU}(W^{(2)}\mathbf{h}^{(1)} + \mathbf{b}^{(2)}),$$

$$\mathbf{h}^{(3)} = W^{(3)}\mathbf{h}^{(2)} + \mathbf{b}^{(3)},$$

with $\mathbf{h}^{(0)} \equiv \mathbf{x}_i^{(5\text{D})\text{std}}$ and $\mathbf{h}^{(3)} \equiv \hat{\mathbf{z}}_i^{\text{std}}$. The 6D vector $\hat{\mathbf{z}}_i^{\text{std}}$ resides in a standardized Galactic Cartesian space. To obtain a latent representation in physical units, it is de-standardized using the mean and standard deviation of the true Galactic Cartesian coordinates in the training set,

$$\hat{\mathbf{z}}_i = \hat{\mathbf{z}}_i^{\text{std}} \odot \boldsymbol{\sigma}_{\text{gal}} + \boldsymbol{\mu}_{\text{gal}},$$

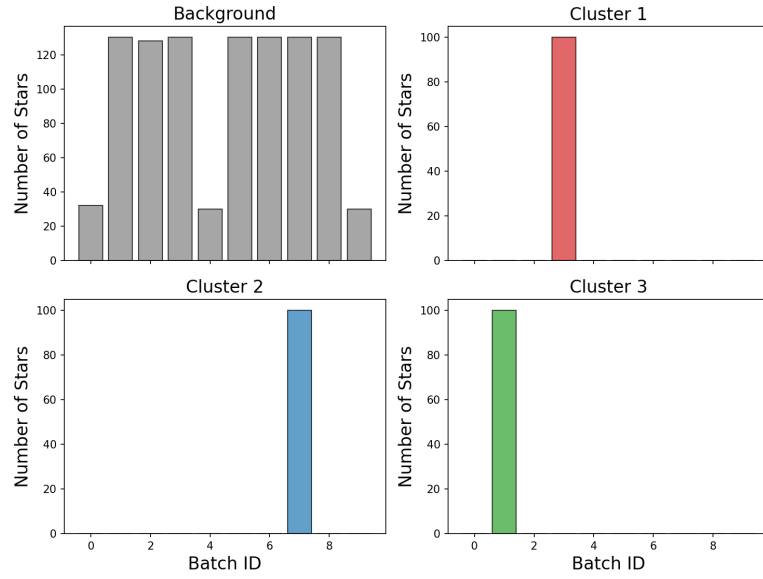
where $\odot$ denotes element-wise multiplication. The result

$$\hat{\mathbf{z}}_i = (\hat{X}_i, \hat{Y}_i, \hat{Z}_i, \hat{U}_i, \hat{V}_i, \hat{W}_i),$$

is the final latent representation used throughout the remainder of this chapter.

4. DENSE Algorithm

(a) Simple data sample



(b) Challenging data sample

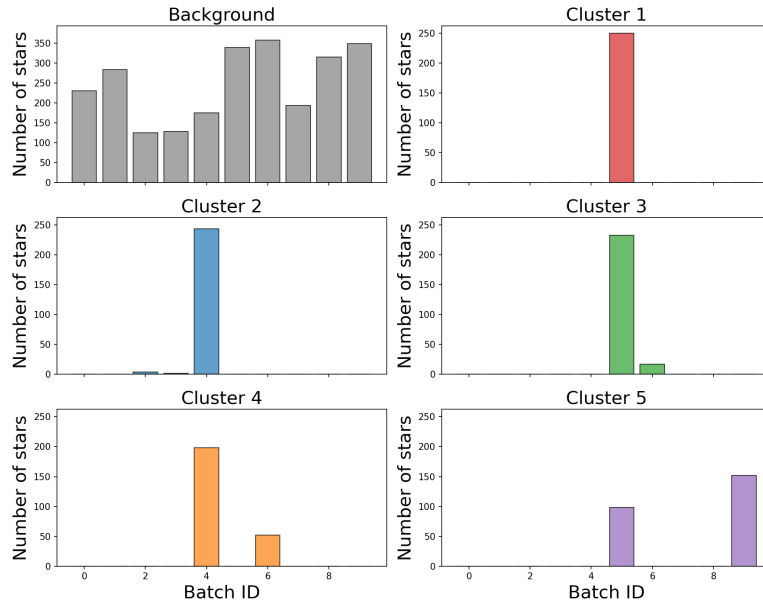


Figure 4.5.: Distribution of stars across batches for $|\Delta\vec{v}| = 0 \text{ km s}^{-1}$, i.e., when the trial velocity is set equal to the true mean velocity of the corresponding cluster. Each panel shows, for the background or a given cluster, the number of stars assigned to each batch ID.

Decoding from latent to observable space is purely analytic. For each $\hat{\mathbf{z}}_i$, the inverse of the ICRS-to-Cartesian transformation from Section 2.2.3 is applied to obtain the predicted parameters

$$\hat{\mathbf{x}}_i = (\hat{\alpha}_i, \hat{\delta}_i, \hat{\varpi}_i, \hat{\mu}_{\alpha*,i}, \hat{\mu}_{\delta,i}, \hat{v}_{\text{rad},i}).$$

These predictions are then standardized with the same $(\boldsymbol{\mu}_{\text{icrs}}, \boldsymbol{\sigma}_{\text{icrs}})$ as the inputs.

Together, the encoder and analytic decoder define a deterministic mapping from observed astrometry to Galactic Cartesian phase space and back. What remains is to specify how this mapping is constrained during training, so that the latent coordinates track the underlying Galactic phase-space structure of the data.

4.4. Training Objective

The training objective combines several terms that tie the latent representation $\hat{\mathbf{z}}_i$ to both the optimal-velocity constraints and the observable parameters. For a given batch of M stars, the model outputs

- latent phase-space vectors $\hat{\mathbf{z}}_i$,
- predicted observable parameters $\hat{\mathbf{x}}_i$, and
- the batch submatrix of optimal-velocity distances $d_{\text{opt}}(i, j)$, computed from the (de-standardized) ICRS quantities using the `pairwise_min_velocity_diff` routine.

The total loss is a weighted sum of four components:

$$\mathcal{L} = \beta_{\text{mat}} \mathcal{L}_{\text{mat}} + \beta_{\text{ast}} \mathcal{L}_{\text{ast}} + \beta_{\text{rv}} \mathcal{L}_{\text{rv}} + \beta_{\text{ws}} \mathcal{L}_{\text{ws}},$$

with the weights $(\beta_{\text{mat}}, \beta_{\text{ast}}, \beta_{\text{rv}}, \beta_{\text{ws}})$ chosen empirically (see below).

Pairwise matrix loss

The matrix loss $\mathcal{L}_{\text{mat}}$ encourages a consistent notion of local neighborhood scale between the optimal-velocity space and the latent velocity space. Let

$$\hat{\mathbf{z}}_i^{\text{UVW}} = (\hat{U}_i, \hat{V}_i, \hat{W}_i)$$

denote the velocity subvector of the latent representation $\hat{\mathbf{z}}_i$. For each batch, the Euclidean distances in latent velocity space,

$$d_{\text{lat}}(i, j) = \|\hat{\mathbf{z}}_i^{\text{UVW}} - \hat{\mathbf{z}}_j^{\text{UVW}}\|_2,$$

are computed for all pairs (i, j) , forming an $M \times M$ pairwise distance matrix

$$D_{\text{lat}} = (d_{\text{lat}}(i, j))_{1 \leq i, j \leq M}.$$

4. DENSE Algorithm

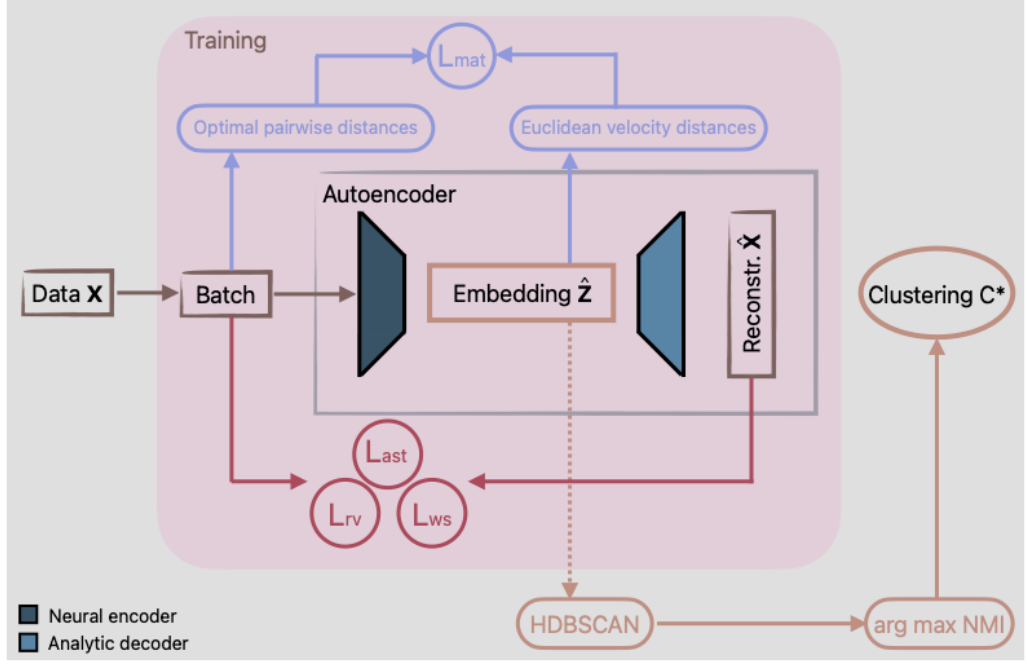


Figure 4.6.: Schematic overview of the baseline DENSE architecture. A neural encoder maps the observables to a 6D latent space, and an analytic decoder implements the coordinate transformation back to the observed astrometry. During training, DENSE minimizes a weighted sum of $\mathcal{L}_{\text{mat}}$, $\mathcal{L}_{\text{ast}}$, $\mathcal{L}_{\text{rv}}$, and $\mathcal{L}_{\text{ws}}$ on each mini-batch. After training, HDBSCAN is run in the latent space, and the final clustering $\mathbf{C}^*$ is selected as the one with the highest NMI over all considered epochs and hyperparameter settings. The visual layout is inspired by the architecture diagram in [Beer et al. \(2024\)](#).

This is compared to the corresponding $M \times M$ optimal-velocity distance matrix

$$D_{\text{opt}} = (d_{\text{opt}}(i, j))_{1 \leq i, j \leq M},$$

where $d_{\text{opt}}(i, j)$ is the optimal velocity difference between stars i and j .

For each star i , the k smallest distances in row i of D_{opt} and D_{lat} (with $k = 10$ in the experiments) are selected, and their mean values are computed. Let $\bar{d}_i^{\text{opt}}$ and $\bar{d}_i^{\text{lat}}$ denote these mean k -nearest-neighbor distances in optimal-velocity space and latent velocity space, respectively. The vectors

$$\bar{\mathbf{d}}^{\text{opt}} = (\bar{d}_1^{\text{opt}}, \dots, \bar{d}_M^{\text{opt}}), \quad \bar{\mathbf{d}}^{\text{lat}} = (\bar{d}_1^{\text{lat}}, \dots, \bar{d}_M^{\text{lat}})$$

summarize, for each star, the typical scale of its local neighborhood in the two spaces.

The matrix loss is defined as

$$\mathcal{L}_{\text{mat}} = 1 - \rho(\bar{\mathbf{d}}^{\text{opt}}, \bar{\mathbf{d}}^{\text{lat}}),$$

where ρ is the Pearson correlation coefficient, used here to encourage a linear relationship between local neighborhood scales in optimal-velocity space and in the latent embedding. Minimizing $\mathcal{L}_{\text{mat}}$ thus maximizes the correlation between neighborhood scales in optimal-velocity space and in the latent velocity representation, encouraging stars that are close according to d_{opt} to also be close in $\hat{\mathbf{z}}^{\text{UVW}}$, up to an overall affine rescaling.

For stars in a real cluster, the batch construction ensures that many of their neighbors in the same batch are other cluster members with similar 3D velocities, so their pairwise optimal-velocity distances d_{opt} are typically small. The matrix loss then pushes their latent velocities $\hat{\mathbf{z}}^{\text{UVW}}$ to be similarly close, so that the corresponding d_{lat} values are also small. Field stars, on the other hand, tend to have large d_{opt} to most other stars in the batch. To minimize $\mathcal{L}_{\text{mat}}$, they are encouraged to occupy regions of latent velocity space where their d_{lat} to other stars is likewise large, which helps to separate them from clustered populations.

Astrometric Reconstruction Loss

The astrometric loss $\mathcal{L}_{\text{ast}}$ measures how well the analytically decoded astrometry reproduces the observed *Gaia*-like quantities. It is defined as the mean squared error (MSE) between the standardized predicted and true five-parameter astrometry,

$$\mathcal{L}_{\text{ast}} = \frac{1}{M} \sum_{i=1}^M \left\| \hat{\mathbf{x}}_i^{(5\text{D})\text{std}} - \mathbf{x}_i^{(5\text{D})\text{std}} \right\|_2^2.$$

This term anchors the latent space in the observed positions and proper motions and prevents the encoder from drifting away from the astrometric constraints while satisfying the pairwise velocity structure.

Radial-Velocity Loss

For the subset of stars with measured radial velocities, the model is additionally trained to predict the correct line-of-sight velocities. Let $v_{\text{rad},i}^{\text{std}}$ and $\hat{v}_{\text{rad},i}^{\text{std}}$ denote the standardized true and predicted radial velocities of stars in the batch with finite measurements. The radial-velocity loss is given by

$$\mathcal{L}_{\text{rv}} = \frac{1}{N_{\text{rv}}} \sum_{i \in \mathcal{I}_{\text{rv}}} \left\| \hat{v}_{\text{rad},i}^{\text{std}} - v_{\text{rad},i}^{\text{std}} \right\|_2^2,$$

where $\mathcal{I}_{\text{rv}}$ is the index set of stars with known radial velocities in the batch and $N_{\text{rv}} = |\mathcal{I}_{\text{rv}}|$. This term provides direct supervision on the velocity component that is otherwise missing for most stars.

4. DENSE Algorithm

In addition to the pointwise MSE, a distributional constraint is imposed via a 1D Wasserstein distance between the predicted and true radial velocities in the batch,

$$\mathcal{L}_{\text{ws}} = W_2(\{\hat{v}_{\text{rad},i}^{\text{std}}\}_{i \in \mathcal{I}_{\text{rv}}}, \{v_{\text{rad},i}^{\text{std}}\}_{i \in \mathcal{I}_{\text{rv}}}),$$

where W_2 denotes the second-order Wasserstein (earth mover's) distance. This term encourages the overall distribution of predicted radial velocities to match the empirical distribution of the available measurements, which is a valid assumption if radial velocities are missing at random.

The model architecture (see Fig. 4.6) serves as the baseline architecture of DENSE. Variants with additional input channels or loss terms that directly constrain latent positions and velocities to the true (XYZ) and (UVW) values for stars with known 6D phase space are tested in Chapter 5.

4.5. Optimization and Implementation Details

The model is trained for 500 epochs using the Adam optimizer (Kingma and Ba 2017) with a learning rate of 10^{-3} . All trainable weights belong to the encoder, whereas the analytic transformation between Galactic Cartesian and ICRS coordinates is fixed, contains no free parameters, and is implemented as a native PYTORCH function so that gradients can be backpropagated through this function into the encoder. ReLU activations are used for all hidden layers because of their simplicity, and they empirically produce more stable training compared to other explored activation functions.

For all experiments, the data set is divided into 80 % training stars and 20 % validation stars, and this split is kept fixed throughout the training. In each epoch, K new `KmeansSubset6D` data sets are instantiated for training and validation splits, using the batching scheme described in Section 4.2. Consequently, every star appears in K different batches per epoch, each associated with a different trial velocity and cluster assignment, and thus provides K independent constraints on its latent representation. Training and validation losses are averaged over all batches in the respective split, and basic diagnostics are logged during training.

To analyze the learned latent representations, the 6D latent vectors $\hat{\mathbf{z}}_{i,k}^{(e)}$ are stored for each star i , for each of its K occurrences $k = 1, \dots, K$ in every epoch e . For downstream analysis, these K samples are averaged within each epoch,

$$\langle \hat{\mathbf{z}}_i^{(e)} \rangle = \frac{1}{K} \sum_{k=1}^K \hat{\mathbf{z}}_{i,k}^{(e)},$$

so that every star is represented by a single 6D latent vector per epoch. In the following analysis, these epoch-averaged latent vectors are used to quantify the accuracy of the recovered phase-space coordinates and to apply the clustering procedure to the learned representation.

4.6. Clustering in the DeNSE Latent Space

Once the DENSE model has been trained, each star i in each epoch e is represented by an epoch-averaged 6D latent vector $\langle \hat{\mathbf{z}}_i^{(e)} \rangle$ that approximates its Galactic Cartesian phase-space coordinates. The final step of the algorithm is to identify stellar associations as overdensities in this latent space.

For every epoch e , the set of latent vectors $\{\langle \hat{\mathbf{z}}_i^{(e)} \rangle\}_i$ (combining both training and validation stars) is passed to the HDBSCAN algorithm introduced in Section 2.4.3. HDBSCAN is evaluated on a small grid of values for the minimum cluster size and minimum samples, yielding a family of clustering solutions

$$C^{(e,\lambda)}, \quad \lambda \in \Lambda,$$

where Λ denotes the set of hyperparameter combinations.

In the experiments on synthetic data, each clustering $C^{(e,\lambda)}$ is compared to the known input groups in Galactic Cartesian space using the normalized mutual information (NMI). The final clustering C^* is defined as the solution that maximizes the NMI with respect to the ground truth,

$$C^* = \arg \max_{e \in \mathcal{E}, \lambda \in \Lambda} \text{NMI}(C^{\text{true}}, C^{(e,\lambda)}),$$

where C^{true} denotes the ground-truth cluster assignment. In practice, I restrict the maximization over epochs to the last part of training, $\mathcal{E} = \{450, \dots, 500\}$, to ensure that the model has approximately converged.

For real *Gaia* data, the ground truth C^{true} is not available, so this NMI-based selection cannot be applied. In that case, an alternative criterion would be required to select a preferred clustering solution, for example, by choosing the epoch and hyperparameters whose clustering best recovers the few known nearby stellar associations.

5. Results

This chapter evaluates the DENSE algorithm on the synthetic data sets introduced in Chapter 2. I first establish a baseline by applying the HDBSCAN clustering algorithm directly to the observable astrometric space and, for comparison, to the full 6D Galactic phase space that is available only for the simulations. This quantifies both the intrinsic difficulty of the clustering problem in observable space and an approximate upper bound on what density-based clustering can achieve in the ideal 6D case.

I then present the performance of DENSE in its basic configuration, including reconstruction errors, the geometry of the learned latent space, and the resulting clustering quality. Afterwards, I discuss several variations of the architecture and report an ablation study that probes the impact of individual loss terms on the learned representation. Finally, I summarize the best-performing configurations for both synthetic samples.

5.1. Baseline Clustering

Before analyzing the performance of DENSE, it is useful to quantify how well a standard density-based clustering method performs when applied directly to the observable astrometric quantities. This establishes a reference for the latent-space results and clarifies how much is gained by learning a new representation.

5.1.1. HDBSCAN on 5D *Gaia*-like Observables

For both the simple and challenging synthetic samples, I first applied HDBSCAN to the 5D astrometric space $(\alpha, \delta, \varpi, \mu_{\alpha*}, \mu_{\delta})$. To obtain a fair baseline, I explored a grid of `min_cluster_size` and `min_samples` values,

$$m_{\text{clSize}} \in \{10, 20, 30, 40, 50, 60, 70, 80, 90, 110, 120, 150, 200\},$$

$$m_{\text{pts}} \in \{10, 15, 20, 25, 30, 35, 40, 60\},$$

and evaluated each configuration using the normalized mutual information (NMI) between the resulting labels and the ground-truth clusters. Throughout this chapter, NMI is computed ignoring background stars, i.e., assuming perfect background removal so that only the signal clusters enter the score. HDBSCAN often recovers the main clusters but struggles with its noise treatment, either by assigning too many background stars to clusters or by splitting the background into several small noise-dominated groups (e.g., [Hunt and Reffert 2021](#); [Hunt and Reffert 2023](#)). The noise-removed NMI, therefore, provides a cleaner measure of how well the intrinsic cluster structure is recovered and

5. Results

how easy the clusters are to find, while imperfections in noise handling can be mitigated by additional post-processing or complementary algorithms.

The best configurations for each sample are summarized in Table 5.1. For the simple sample, HDBSCAN recovers three clusters with a maximum NMI of 0.82, and several neighboring hyperparameter choices yield nearly identical scores, indicating that this result is not overly sensitive to the exact choice of m_{clSize} and m_{pts} . Even in this relatively simple case, however, projection effects in astrometric space already complicate what is an easy clustering problem in the underlying 6D phase space.

For the challenging sample, no choice of hyperparameters leads to a satisfactory solution. The best configuration formally recovers five clusters but reaches only $\text{NMI} \approx 0.33$. Across most of the grid, HDBSCAN instead returns just two broad clusters with $\text{NMI} < 0.3$. These failures reflect the projection effects and large positional dispersion discussed in Chapter 2, which cause clusters that are compact in $(XYZUVW)$ to appear diffuse and heavily blended in observable space.

For completeness, I also tested an alternative 5D space in which the proper-motion components are replaced by tangential velocities (v_α, v_δ) derived from the astrometry. This yields similar clustering performance to the proper-motion baseline. In principle, the performance in such a 5D space could be improved further by rescaling the tangential-velocity axes when clusters are kinematically compact. In practice, however, the appropriate scaling depends on the unknown sizes and internal dispersions of real associations, so I keep the baseline metric unscaled and agnostic to such choices.

5.1.2. HDBSCAN on 6D Galactic Cartesian Phase Space

In an idealized setting where the full 6D Galactic phase space $(XYZUVW)$ is known, density-based clustering methods such as HDBSCAN are expected to perform much better. To quantify this, I repeated the same grid search over m_{clSize} and m_{pts} in the true 6D phase space for both synthetic samples. This experiment is only possible for simulations, but it provides an approximate upper bound on the clustering quality that DENSE could, in principle, approach.

The best results are again listed in Table 5.1. For the simple sample, HDBSCAN recovers all three clusters perfectly with $\text{NMI} = 1.0$ over a wide range of hyperparameters. For the challenging sample, the algorithm identifies all five clusters with a maximum NMI of 0.9. Although the recovery is not completely perfect, the performance is dramatically better than in 5D observable space.

These experiments highlight two key points. First, directly applying HDBSCAN to the observable astrometry does not yield a satisfactory clustering solution, even under optimized hyperparameters. Second, the same algorithm performs almost ideally when given access to the true 6D phase space.

Table 5.1.: Best HDBSCAN performance on the synthetic samples in 5D observable astrometric space and in the true 6D Galactic phase space. Shown are the hyperparameters `min_samples` (m_{pts}) and `min_cluster_size` (m_{clSize}), the number of detected clusters N_{Cl} , and the normalized mutual information (NMI), computed ignoring background stars.

Sample	5D observables				6D phase space			
	m_{pts}	m_{clSize}	N_{Cl}	NMI	m_{pts}	m_{clSize}	N_{Cl}	NMI
Simple	15	30	3	0.82	10	10	3	1.00
Challenging	20	10	5	0.33	10	10	5	0.90

5.2. Performance of DeNSE

This section presents the performance of the baseline DENSE architecture on the two synthetic data sets described in Section 2.3.

5.2.1. Hyperparameter Grid Search

To assess the robustness of DENSE and identify reasonable hyperparameter choices, I conducted a grid search over batch-construction parameters, loss weights, and HDBSCAN clustering settings.

For the batch size M and the number of repeats K per epoch, I used

$$M_{\text{simple}} \in \{65, 130, 260\}, \quad M_{\text{challenging}} \in \{150, 250, 375\}, \quad K \in \{5, 10, 15\}.$$

The loss weights were varied over

$$\beta_{\text{ast}} \in \{0.7, 1.0, 1.3, 5.0, 8.0\}, \quad \beta_{\text{mat}} \in \{0.1, 0.2, 0.4\},$$

$$\beta_{\text{rv}} \in \{0.1, 0.3, 0.5, 0.7\}, \quad \beta_{\text{ws}} \in \{0.1, 0.2, 0.4\},$$

and the HDBSCAN hyperparameters over

$$m_{\text{clSize}} \in \{10, 20, 30, 40, 50, 60\}, \quad m_{\text{pts}} \in \{10, 15, 20, 30, 35\},$$

from which I selected 15 representative $(m_{\text{clSize}}, m_{\text{pts}})$ pairs for evaluation. The explored ranges reflect values for which the model converged reliably and produced meaningful latent representations and clusterings in preliminary tests.

The configurations found for each data set are summarized in Tables 5.2 and 5.3. The average position and velocity errors, $\bar{\epsilon}_{\text{xyz}}$ and $\bar{\epsilon}_{\text{uvw}}$, denote the mean Euclidean distances between the true and predicted Cartesian phase-space coordinates, averaged over stars.

For the simple data set, the DENSE baseline architecture yields an NMI of 1.0 (ignoring background stars) for essentially all tested hyperparameter combinations. This indicates

5. Results

that the clustering performance is robust and not particularly sensitive to the exact choice of batch size, repeats, or loss weights. The mean position errors $\bar{\epsilon}_{\text{xyz}}$ span $\sim 0.5\text{--}2.5$ pc across the grid and can be reduced to $\bar{\epsilon}_{\text{xyz}} \simeq 0.5$ pc by increasing the weight β_{ast} of the astrometric loss. This suggests that β_{ast} should be chosen comparatively large when accurate position reconstruction is desired. The corresponding velocity errors $\bar{\epsilon}_{\text{uvw}}$ remain at the level of $1.5\text{--}2.6$ km s $^{-1}$.

For the challenging sample, the resulting NMI values are lower, typically around 0.6, but again fairly stable across the grid. The highest NMI of 0.68 is achieved for a configuration with $M = 375$ and $K = 5$, although this setup also exhibits some of the largest reconstruction errors in the latent 6D phase space. Over the full grid, the latent reconstruction quality varies from $\bar{\epsilon}_{\text{xyz}} \sim 1.9$ to 3.9 pc and $\bar{\epsilon}_{\text{uvw}}$ from ~ 3.2 to 4.3 km s $^{-1}$. This leaves room for targeted improvements of the architecture and loss function that focus on reducing $\bar{\epsilon}_{\text{xyz}}$ and $\bar{\epsilon}_{\text{uvw}}$.

The number of repeats K per epoch and the batch size M do not appear to have a strong influence on the overall performance. However, intermediate choices generally yield the most stable training curves and provide a better compromise between clustering quality and reconstruction accuracy. Because NMI is computed ignoring background stars, additional clusters that contain only background objects do not affect the score.

Table 5.2.: Representative DENSE configurations for the simple synthetic sample. Shown are batch size M , repeats per epoch K , loss weights, HDBSCAN hyperparameters, mean position and velocity errors $\bar{\epsilon}_{\text{xyz}}$ and $\bar{\epsilon}_{\text{uvw}}$, the number of detected clusters N_{Cl} , and NMI (ignoring background stars).

M	K	β_{ast}	β_{mat}	β_{rv}	β_{ws}	m_{pts}	m_{clSize}	$\bar{\epsilon}_{\text{xyz}}$ (pc)	$\bar{\epsilon}_{\text{uvw}}$ (km s $^{-1}$)	N_{Cl}	NMI
65	5	1.3	0.4	0.7	0.1	10	10	1.4	1.8	3	1.0
65	10	0.7	0.4	0.3	0.1	10	10	1.1	1.5	3	1.0
65	10	8.0	0.4	0.1	0.1	20	10	2.1	2.6	3	1.0
130	5	0.7	0.4	0.3	0.1	10	10	1.8	2.2	3	1.0
130	5	1.0	0.2	0.5	0.1	35	10	1.4	1.7	3	1.0
130	5	1.0	0.1	0.1	0.2	20	10	0.9	1.8	3	1.0
130	5	1.0	0.4	0.5	0.2	10	10	1.4	1.5	3	1.0
130	5	5.0	0.1	0.1	0.2	20	10	0.6	1.8	3	1.0
130	5	8.0	0.1	0.1	0.2	20	10	0.5	1.9	3	1.0
130	5	8.0	0.1	0.5	0.2	20	10	1.2	2.0	3	1.0
130	10	8.0	0.2	0.5	0.1	35	10	0.6	1.6	3	1.0
260	5	1.3	0.4	0.7	0.1	20	10	1.9	2.1	3	1.0
260	10	8.0	0.4	0.1	0.1	20	10	2.5	2.6	3	1.0
260	15	1.0	0.1	0.1	0.2	10	10	1.1	1.9	3	1.0

Table 5.3.: Representative DENSE configurations for the challenging synthetic sample. Columns are as in Table 5.2

M	K	β_{ast}	β_{mat}	β_{rv}	β_{ws}	m_{pts}	m_{clSize}	$\bar{\epsilon}_{\text{xyz}}$ (pc)	$\bar{\epsilon}_{\text{uvw}}$ (km s ⁻¹)	N_{Cl}	NMI
150	5	1.3	0.1	0.7	0.4	10	50	3.3	4.2	5	0.60
250	5	5.0	0.2	0.1	0.1	10	50	1.9	3.4	6	0.64
250	10	1.0	0.2	0.3	0.1	10	30	2.8	3.5	7	0.62
250	10	5.0	0.2	0.1	0.2	10	30	2.8	3.9	5	0.63
250	10	8.0	0.4	0.1	0.2	10	50	2.0	3.5	6	0.63
250	15	1.0	0.2	0.5	0.2	10	30	2.8	3.2	5	0.62
375	5	0.7	0.4	0.3	0.1	10	30	3.9	4.3	5	0.68
375	10	1.0	0.2	0.5	0.2	10	50	2.7	3.7	5	0.60
375	10	1.0	0.2	0.3	0.1	10	50	2.6	4.0	6	0.60

5.3. Experiments

In addition to the baseline setup, I explored several adaptations of the DENSE architecture and loss function to investigate potential improvements.

5.3.1. Additional Loss Terms

To test whether direct supervision in Cartesian phase space can improve the latent representation, I added mean-squared-error (MSE) losses in positions (XYZ) and velocities (UVW) on top of the baseline objective. The corresponding loss weights were varied over

$$\beta_{\text{xyz}}, \beta_{\text{uvw}} \in \{0.1, 0.3, 0.5, 0.8\}.$$

Additional XYZ Loss

In the first experiment, I augmented the baseline loss with an additional XYZ term that penalizes the MSE between the true Cartesian positions and the latent-space positions. The true XYZ coordinates can be computed by coordinate transformation from the observables without using radial velocities (see Section 2.2.3).

As summarized in Tables 5.4 and 5.5, this additional term improves the position reconstruction in the latent space for both the simple and the challenging samples. For the simple sample, $\bar{\epsilon}_{\text{xyz}}$ decreases relative to the baseline while the NMI remains essentially unchanged.

For the challenging sample, adding the XYZ loss also reduces $\bar{\epsilon}_{\text{xyz}}$, but the corresponding clustering solutions tend to exhibit more clusters than present in the ground truth even when the NMI is comparable to the baseline.

5. Results

Additional UVW Loss

Analogously, I also tested an additional UVW loss that penalizes the MSE between the true and reconstructed (UVW) for the subset of stars with measured radial velocities (10% of the sample). This term therefore constrains the latent velocity field only where full 6D information is available.

For the simple data set, a moderately weighted UVW loss can slightly improve both $\bar{\epsilon}_{xyz}$ and $\bar{\epsilon}_{uvw}$ compared to the baseline configuration, while preserving $NMI \approx 1.0$ (Table 5.4).

In contrast, for the challenging sample, the behavior is less favorable. Here, an additional UVW loss tends to worsen the position errors $\bar{\epsilon}_{xyz}$ and slightly degrades the clustering solution, while leaving $\bar{\epsilon}_{uvw}$ essentially unchanged relative to the baseline (Table 5.5).

Combining both XYZ and UVW losses yields a latent representation that balances position and velocity reconstruction in ($XYZUVW$), but does not produce an increase in NMI over the baseline. In addition, the corresponding runs show clearer signs of overfitting in their learning curves. Overall, these experiments indicate that direct supervision in Cartesian phase space is most useful as a tool to refine reconstruction accuracy, whereas the main improvement in cluster recovery already comes from the combination of astrometric, matrix-based, and radial-velocity terms in the baseline objective.

Table 5.4.: Effect of additional XYZ and UVW loss terms on the simple sample.

M	K	β_{ast}	β_{mat}	β_{rv}	β_{ws}	β_{xyz}	β_{uvw}	m_{pts}	m_{clSize}	$\bar{\epsilon}_{\text{xyz}}$ (pc)	$\bar{\epsilon}_{\text{uvw}}$ (km s ⁻¹)	N_{Cl}	NMI
130	5	1.0	0.2	0.5	0.1	0.5	/	35	10	0.8	1.6	3	1.0
130	5	1.0	0.2	0.5	0.1	/	0.8	20	10	1.0	1.4	3	1.0
130	5	1.0	0.2	0.5	0.1	0.5	0.8	20	10	0.9	1.4	3	0.99

Table 5.5.: Effect of additional XYZ and UVW loss terms on the challenging sample.

M	K	β_{ast}	β_{mat}	β_{rv}	β_{ws}	β_{xyz}	β_{uvw}	m_{pts}	m_{clSize}	$\bar{\epsilon}_{\text{xyz}}$ (pc)	$\bar{\epsilon}_{\text{uvw}}$ (km s ⁻¹)	N_{Cl}	NMI
250	5	1.0	0.2	0.3	0.1	0.3	/	20	20	1.4	3.6	8	0.62
250	5	1.0	0.2	0.3	0.1	/	0.8	10	30	3.6	3.5	7	0.59
250	5	1.0	0.2	0.3	0.1	0.3	0.8	10	50	1.6	3.5	4	0.62

5.3.2. Additional Input Information

As a complementary direction, I tested whether providing the encoder with additional input channels can help DENSE to learn a better representation.

Additional XYZ Input

In one experiment, I extended the encoder input from 5D astrometry to an 8D vector by appending the Cartesian positions XYZ , which can be computed without radial velocities. The encoder thus maps an 8D observable-space vector to the 6D latent phase space.

The corresponding results are listed in Tables 5.6 and 5.7. For the simple sample, including XYZ as input preserves the clustering performance ($NMI = 1.0$) while slightly improving the latent position accuracy. In the first setup of Table 5.6, the mean position error is reduced to $\bar{\epsilon}_{xyz} = 0.5$ pc with unchanged NMI, at the cost of somewhat larger velocity errors compared to the 5D-input run using the same hyperparameters (cf. Table 5.2). Introducing an explicit position loss term (second row) does not yield further gains and, in fact, slightly increases both the position and velocity errors.

For the challenging sample, the effect on NMI remains modest. The first configuration in Table 5.7 reaches $NMI \approx 0.62$, comparable to 5D-input runs, but does so by fragmenting the data into a larger number of clusters ($N_{Cl} = 14$). A second configuration with an explicit XYZ loss term attains a slightly higher $NMI \approx 0.67$ and recovers approximately the correct number of clusters ($N_{Cl} = 5$) with improved mean position and velocity errors. However, its learning curves exhibit clear signs of overfitting, so these improvements are likely not robust.

Table 5.6.: DENSE performance on the simple sample when augmenting the encoder input with Cartesian positions (XYZ).

M	K	β_{ast}	β_{mat}	β_{rv}	β_{ws}	β_{xyz}	m_{pts}	m_{clSize}	$\bar{\epsilon}_{xyz}$ (pc)	$\bar{\epsilon}_{uvw}$ (km s ⁻¹)	N_{Cl}	NMI
130	10	1.0	0.2	0.5	0.1	/	20	10	0.5	2.1	3	1.0
130	10	1.0	0.2	0.5	0.1	0.3	10	30	0.8	2.5	3	1.0

Table 5.7.: Same as Table 5.6, but for the challenging sample.

M	K	β_{ast}	β_{mat}	β_{rv}	β_{ws}	β_{xyz}	m_{pts}	m_{clSize}	$\bar{\epsilon}_{xyz}$ (pc)	$\bar{\epsilon}_{uvw}$ (km s ⁻¹)	N_{Cl}	NMI
375	5	1.0	0.2	0.5	0.1	/	10	10	3.8	4.2	14	0.62
250	5	1.0	0.2	0.5	0.4	0.3	10	30	1.3	3.5	5	0.67

Additional UVW Input

Finally, I tested an 8D-to-6D encoder in which the input is augmented by the trial UVW velocity used to construct the current batch. In this setup, the encoder receives both

5. Results

the astrometric observables and the trial UVW drawn from the Gaussian prior for batch construction. The idea is that this additional context might help the encoder recognize when a particular batching realization is suboptimal concerning the true underlying motions.

Tables 5.8 and 5.9 show the corresponding results. For the simple sample, adding UVW as input preserves the clustering performance, albeit with a slight decrease in the second configuration. The latent position and velocity errors remain within the range of the 5D-input baselines. For example, the configuration in Table 5.8 with a non-zero β_{xyz} achieves $\bar{\epsilon}_{xyz} \simeq 0.8$ pc and $\bar{\epsilon}_{uvw} \simeq 1.8$ km s⁻¹, which is comparable to the best 5D runs rather than representing a clear improvement.

For the challenging sample, the best configurations with UVW input achieve NMI values of around 0.60–0.67, which is similar to the baseline grid search in Section 5.2. The position and velocity errors also lie within the same general range, with $\bar{\epsilon}_{xyz} \sim 1.7$ –3.2 pc and $\bar{\epsilon}_{uvw} \sim 3.4$ –4.2 km s⁻¹. A mildly positive effect is that all UVW -input configurations recover the true number of clusters, N_{Cl} , which is five. Overall, however, the benefits of including the UVW trial in the input are marginal and do not significantly improve clustering quality or reconstruction accuracy compared to the 5D input baseline.

Table 5.8.: DENSE performance on the simple sample when augmenting the encoder input with the trial UVW velocity used for batch construction.

M	K	β_{ast}	β_{mat}	β_{rv}	β_{ws}	β_{xyz}	m_{pts}	m_{clSize}	$\bar{\epsilon}_{xyz}$ (pc)	$\bar{\epsilon}_{uvw}$ (km s ⁻¹)	N_{Cl}	NMI
130	5	1.0	0.2	0.5	0.2	/	20	10	1.8	2.7	3	1.0
130	10	1.0	0.2	0.5	0.2	/	35	10	1.2	2.5	3	0.99
130	5	1.0	0.2	0.5	0.2	0.5	10	30	0.8	1.8	3	1.0

Table 5.9.: Same as Table 5.8, but for the challenging sample.

M	K	β_{ast}	β_{mat}	β_{rv}	β_{ws}	β_{xyz}	β_{uvw}	m_{pts}	m_{clSize}	$\bar{\epsilon}_{xyz}$ (pc)	$\bar{\epsilon}_{uvw}$ (km s ⁻¹)	N_{Cl}	NMI
250	5	1.0	0.2	0.5	0.4	0.3	/	10	50	2.1	4.2	5	0.60
250	10	1.0	0.2	0.5	0.2	/	/	10	50	2.7	3.9	5	0.65
250	10	1.0	0.2	0.5	0.2	0.5	/	15	40	1.7	3.5	5	0.67
375	5	1.0	0.2	0.5	0.2	/	/	10	50	3.2	4.0	5	0.59
375	10	1.0	0.2	0.5	0.1	/	0.8	10	50	3.0	3.4	5	0.66

5.4. Ablation Study

To better understand the role of the individual loss components, I conducted a small ablation study. Starting from a well-performing configuration for each sample, I selectively removed loss terms and examined how this affected the latent reconstruction and clustering performance.

For the simple sample, I chose the sixth hyperparameter setup in Table 5.2, which is a configuration that achieves good NMI and accurate approximations of the 6D latent space. The performance changes when removing or isolating different losses are summarized in Table 5.10. The learning curves for runs using each loss term separately are shown in Appendix A.2.

Using only the matrix loss $\mathcal{L}_{\text{mat}}$ is insufficient to recover the true 6D structure. While the model learns an approximate location of the most densely clustered stars in velocity space, the overall distribution is distorted, and the NMI drops to 0.82. However, this loss alone already enables partial cluster separation that can be exploited by density-based algorithms. Using only the astrometric loss $\mathcal{L}_{\text{ast}}$ yields $\text{NMI} \approx 0.99$ and reasonably small reconstruction errors, indicating that the astrometric constraints alone already encode much of the positional and kinematic information for the simple sample.

Using only the radial-velocity loss $\mathcal{L}_{\text{rv}}$ or the Wasserstein loss $\mathcal{L}_{\text{ws}}$ provides some information on the velocity distribution, but not enough to achieve stable clustering on their own ($\text{NMI} \leq 0.68$). Using only the XYZ loss $\mathcal{L}_{\text{xyz}}$ yields extremely accurate positions ($\bar{\epsilon}_{\text{xyz}} \approx 0.1$ pc), and even if it leaves the velocities poorly constrained, it attains an NMI of 1.0. In contrast, using only the UVW loss $\mathcal{L}_{\text{uvw}}$ leads to comparatively small velocity errors but very poor positional accuracy ($\bar{\epsilon}_{\text{xyz}} \sim 8$ pc), while still reaching a relatively high NMI of ~ 0.94 .

When several losses are combined (upper rows of Table 5.10), the situation improves markedly: configurations that retain $\mathcal{L}_{\text{ast}}$ and $\mathcal{L}_{\text{mat}}$, optionally augmented by $\mathcal{L}_{\text{xyz}}$ and weak RV/WS terms, all reach $\text{NMI} \approx 1.0$ while keeping $\bar{\epsilon}_{\text{xyz}}$ and $\bar{\epsilon}_{\text{uvw}}$ at the level of ~ 1 pc and ~ 2 km s⁻¹, respectively.

For the challenging sample, I performed the same tests using the first setup of hyperparameters in Table 5.9 as reference (Table 5.11). Here, the impact of the ablation is stronger. Removing the radial velocity and the Wasserstein terms results in larger velocity errors and a decrease in NMI from 0.60 to 0.50, highlighting the importance of these components under more challenging conditions. Configurations retaining only a single loss term perform significantly worse ($\text{NMI} \lesssim 0.5$ and/or significantly increased reconstruction errors in most single-term runs), confirming that the challenging sample requires the full combination of constraints to approach the ideal 6D clustering performance.

5. Results

Table 5.10.: Ablation study for the simple sample, starting from a configuration with $M = 130$ and $K = 5$. Each row shows a different combination of active loss terms.

β_{ast}	β_{mat}	β_{rv}	β_{ws}	β_{xyz}	β_{uvw}	m_{pts}	m_{clSize}	$\bar{\epsilon}_{\text{xyz}}$ (pc)	$\bar{\epsilon}_{\text{uvw}}$ (km s ⁻¹)	N_{Cl}	NMI
1.0	0.1	0.1	0.2	/	/	20	10	0.9	1.8	3	1.0
1.0	0.1	/	/	0.5	/	10	10	0.6	2.8	3	1.0
/	0.1	0.1	0.2	0.5	/	10	10	0.5	5.1	3	1.0
/	1	/	/	/	/	10	10	6.1	3.8	3	0.82
1	/	/	/	/	/	10	50	1.3	3.4	3	0.99
/	/	1	/	/	/	10	10	6.2	4.2	2	0.68
/	/	/	1	/	/	10	10	7.0	4.2	4	0.49
/	/	/	/	1	/	10	10	0.1	4.1	3	1.0
/	/	/	/	/	1	10	10	7.7	1.3	3	0.94

Table 5.11.: Ablation study for the challenging sample, based on the run with additional trial UVW input and hyperparameters $M = 250$ and $K = 5$.

β_{ast}	β_{mat}	β_{rv}	β_{ws}	β_{xyz}	β_{uvw}	m_{pts}	m_{clSize}	$\bar{\epsilon}_{\text{xyz}}$ (pc)	$\bar{\epsilon}_{\text{uvw}}$ (km s ⁻¹)	N_{Cl}	NMI
1.0	0.2	0.5	0.4	0.3	/	10	50	2.1	4.2	5	0.60
1.0	0.2	/	/	0.3	/	40	20	2.0	5.0	5	0.50
/	0.2	0.5	0.4	0.3	/	10	10	1.2	6.4	5	0.46
/	1	/	/	/	/	10	10	9.9	6.9	6	0.42
1	/	/	/	/	/	10	10	2.4	4.4	6	0.50
/	/	1	/	/	/	10	10	14.8	7.3	6	0.26
/	/	/	1	/	/	10	10	12.2	6.9	5	0.32
/	/	/	/	1	/	20	10	0.5	6.3	3	0.34
/	/	/	/	/	1	10	10	11.1	3.8	5	0.48

5.5. Best-Performing Configurations

To illustrate the results, I selected a hyperparameter setup for each data sample that produced particularly good performance, even though many other configurations are very similar.

5.5.1. Simple Data Set

For the simple sample, almost all of the tested configurations achieved an NMI of 1.0, so selecting a preferred setup is mainly a trade-off between keeping the model as simple as possible and reducing reconstruction errors and overfitting. I adopt the ninth example from Table [5.2](#), which satisfies these criteria and attains $\bar{\epsilon}_{\text{xyz}} = 0.5$ pc and $\bar{\epsilon}_{\text{uvw}} = 1.9$ km s⁻¹.

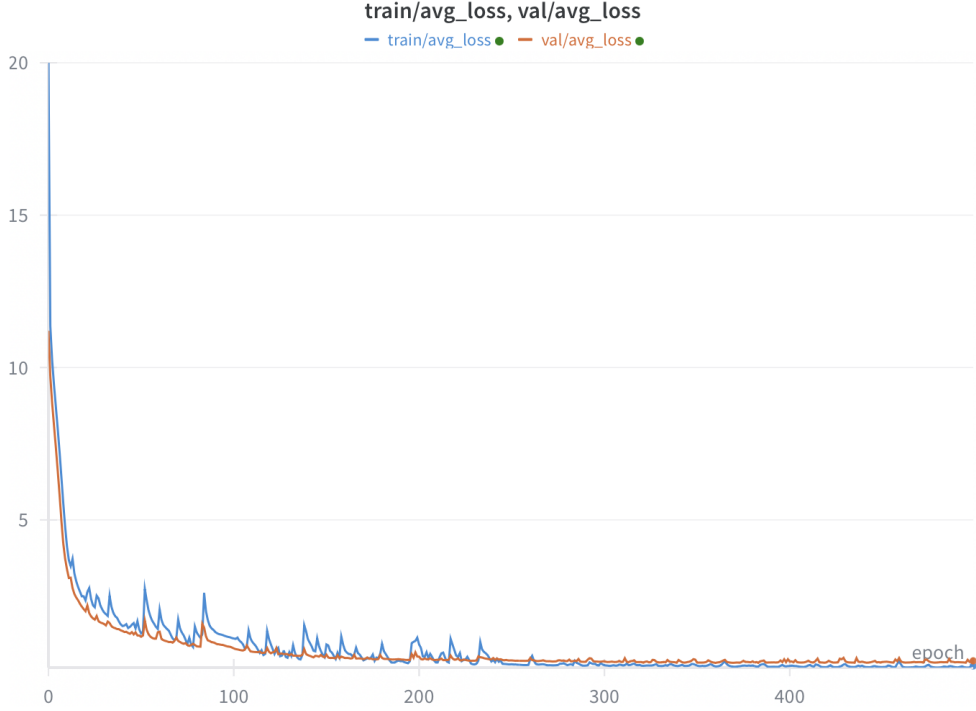


Figure 5.1.: Training and validation loss versus epoch for the selected DENSE configuration on the simple sample.

Figure 5.1 shows the evolution of the training and validation losses for the chosen configuration. The validation loss closely follows the training loss, and both curves flatten quickly, indicating stable convergence without signs of overfitting.

The learned latent-space distribution in XYZ and UVW is shown in Figure 5.2. The three clusters remain clearly separated and retain their original structure in position and velocity space. The embedding is slightly tighter in position than in velocity, reflecting the perfect astrometric constraints on the spatial distribution.

Figure 5.3 compares the true and predicted Cartesian coordinates in XYZ (top) and UVW (bottom), while Figure 5.4 shows the corresponding error histograms. The mean axis-wise position and velocity errors are (0.5, 0.4, 0.5) pc and (2.0, 2.0, 1.7) km s⁻¹, respectively. This demonstrates that the latent representation closely tracks the true 6D phase-space structure.

The best clustering solution obtained with HDBSCAN in the latent space is shown in Figure 5.5. All three clusters are clearly separated and contain only a modest number of contaminating background stars. The exact numbers for the detected clusters are given

5. Results

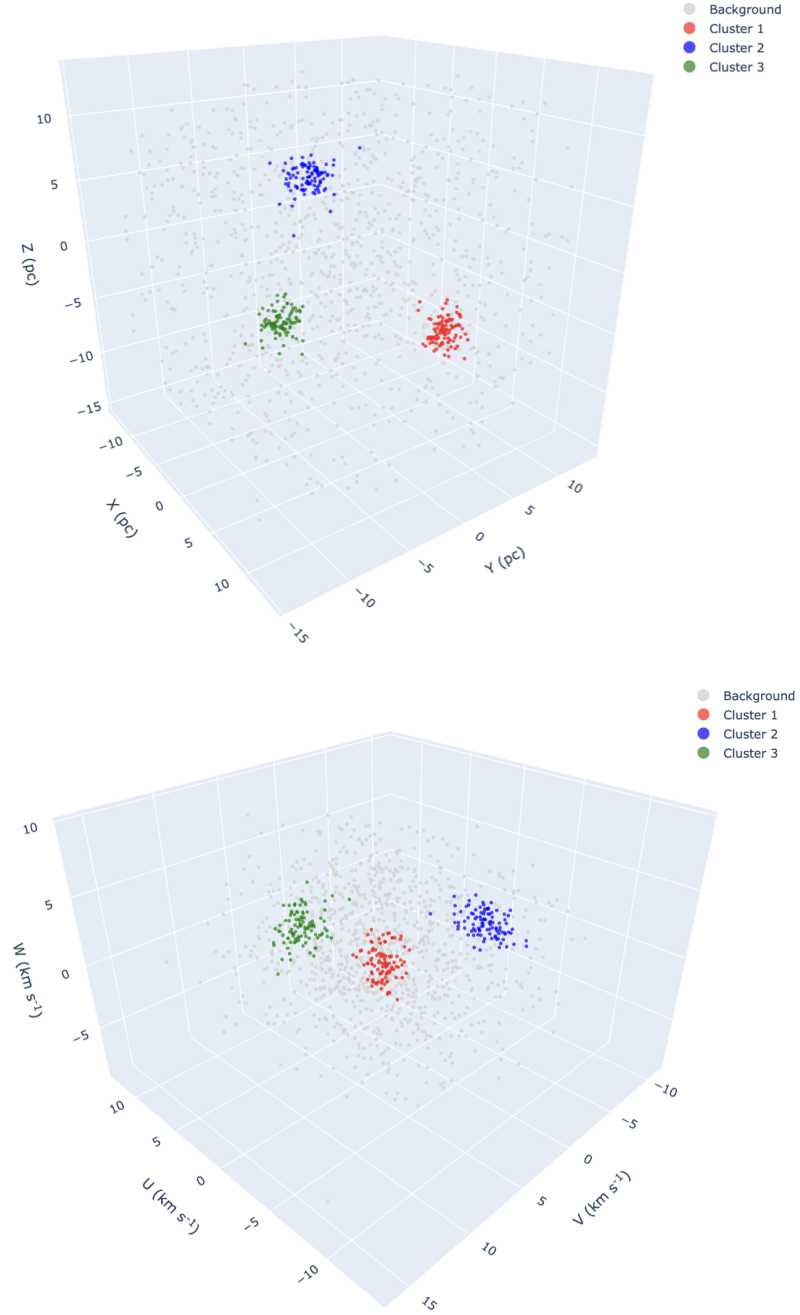


Figure 5.2.: Latent-space embedding of the simple sample in predicted Cartesian positions XYZ (top) and velocities UVW (bottom). Points are colored by their ground-truth cluster labels.

5.5. Best-Performing Configurations

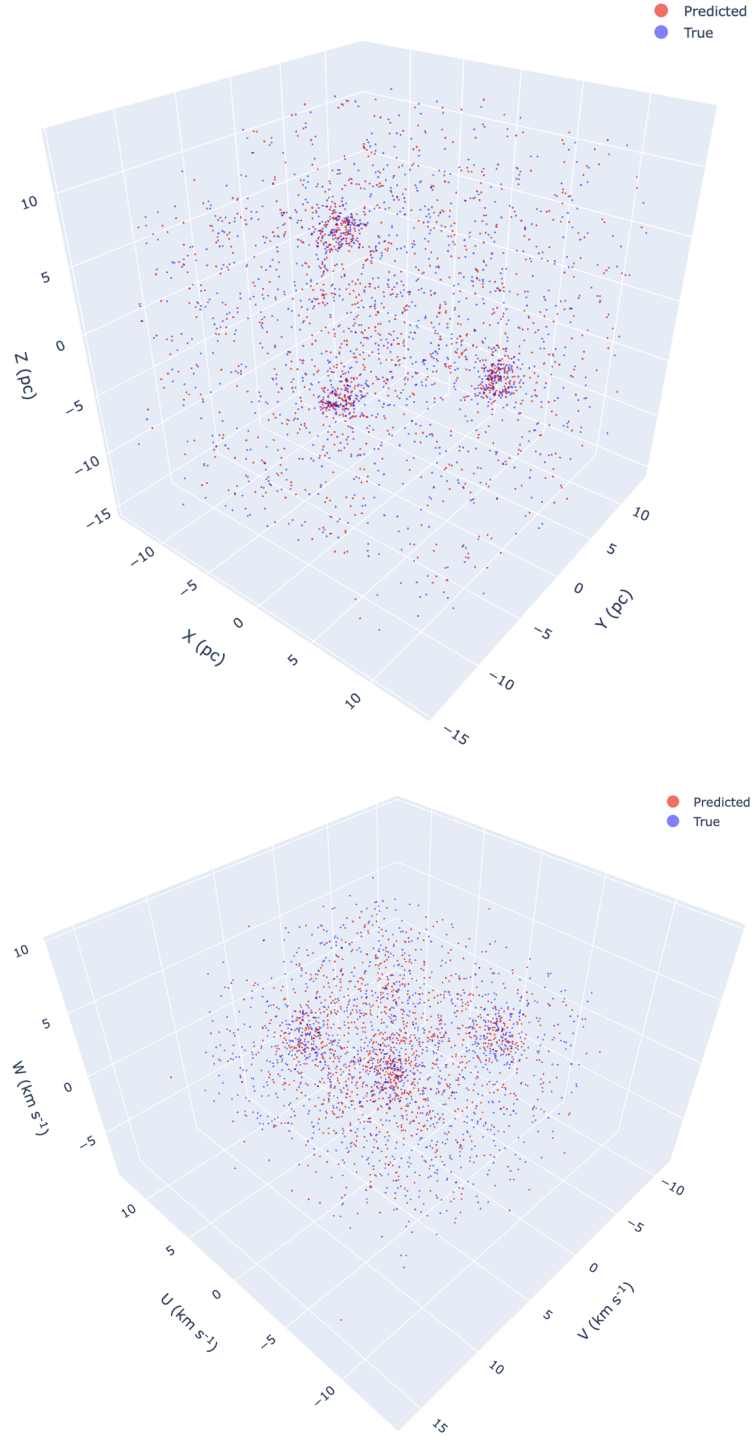


Figure 5.3.: True versus predicted Cartesian coordinates for the simple sample in positions XYZ (top) and velocities UVW (bottom).

5. Results

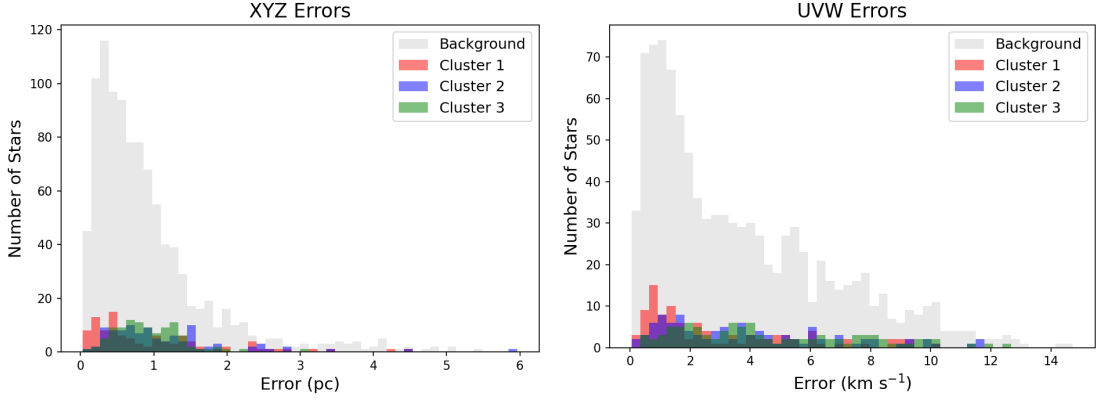


Figure 5.4.: Axis-wise errors in the latent-space positions (left) and velocities (right) for the simple sample. The mean absolute errors in XYZ are (0.5, 0.4, 0.5) pc and in UVW (2.0, 2.0, 1.7) km s^{-1} .

in Table 5.12. Each recovered cluster contains 100 true members and 16–47 contaminants, corresponding to contamination fractions of about 14–32%.

Table 5.12.: Composition of the three clusters detected by DENSE in the simple sample, for the chosen best-performing configuration.

	Cluster 1	Cluster 2	Cluster 3
Total stars	116	147	117
Background	16	47	17
True label	100	100	100

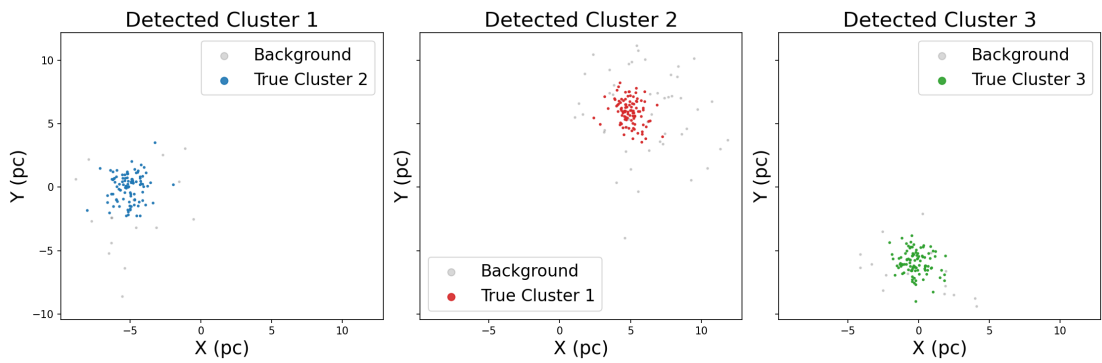


Figure 5.5.: Best HDBSCAN clustering result in the DENSE latent space for the simple sample. Points are colored by their ground-truth cluster labels.

The distribution of NMI values over epochs and HDBSCAN hyperparameters is summarized in Figure 5.6. The NMI is 1.0 for almost all epochs and parameter combinations, which confirms the robustness of the clustering result.

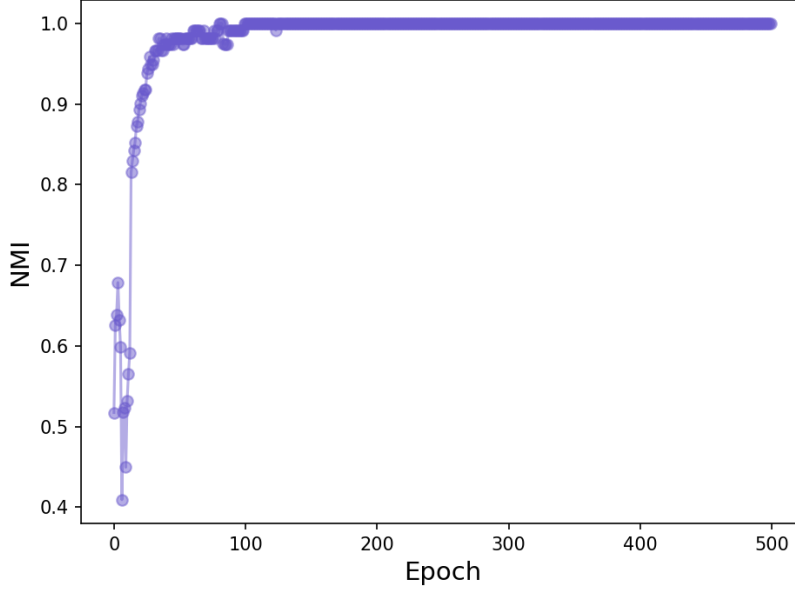


Figure 5.6.: NMI of HDBSCAN clusterings in the DENSE latent space for the simple sample.

A direct comparison between the ideal 6D clustering (see Section 5.1.2) and the DENSE-based solution is provided in Figure 5.7. The left column displays the clusters obtained by running HDBSCAN directly in the true 6D phase space and projecting the labels to XYZ (top) and UVW (bottom). The right column shows the same stars colored according to the clusters found in the DENSE latent space, again plotted in the true coordinates. In both position and velocity space, the recovered groups closely resemble the shapes and sizes of the ideal clusters. In each case, some nearby background stars are also assigned to the clusters. Contamination fractions are about 14–16% for the ideal 6D clustering and 14–32% for the DeNSE solution (cf. Table 5.12). In Figure 5.7, this appears as a thin population of field stars at the edges of the compact cluster cores in both columns. For the simple sample, clustering in the DENSE latent space therefore yields results that are very close to those obtained by clustering directly in the true 6D phase space.

5.5.2. Challenging Data Set

For the challenging data sample, all configurations deliver broadly similar results, but the most stable setups combine the baseline loss with an additional UVW input channel to the encoder (see Table 5.9). In these runs, the input to the network is augmented by the

5. Results

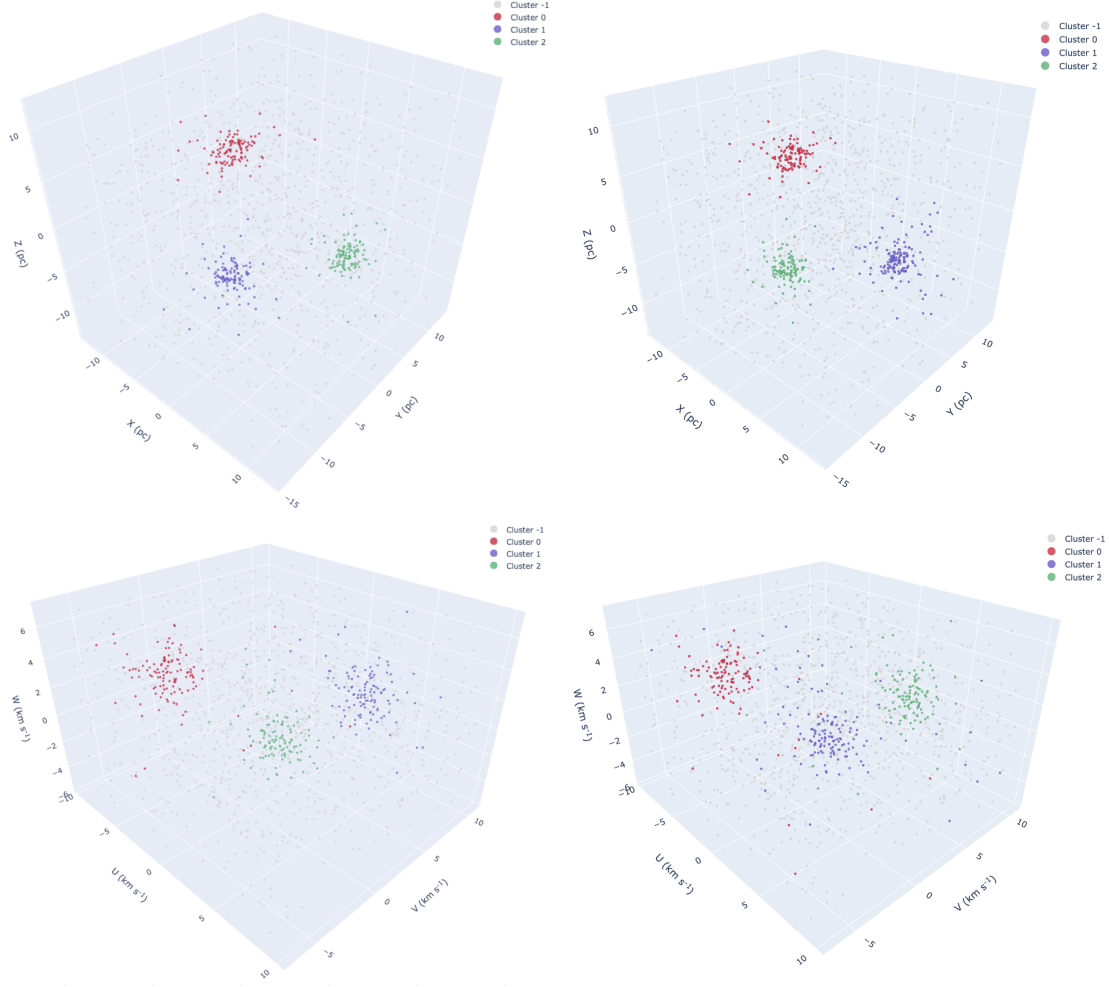


Figure 5.7.: True Galactic Cartesian coordinates for the simple sample, colored by detected HDBSCAN clusters. Top row: positions XYZ , bottom row: velocities UVW . Left column: clusters obtained by running HDBSCAN directly in the true 6D phase space ($XYZUVW$). Right column: clusters obtained by running HDBSCAN in the DENSE latent space.

trial (U, V, W) velocity used to construct the current batch. Thus, the encoder receives the astrometric observables and the trial velocity drawn from the Gaussian prior. This additional context helps the model to interpret each batch realization and to cope with suboptimal trial velocities. These configurations achieve NMI values of about 0.59–0.67 while keeping the latent position and velocity errors at the level of a few parsecs and a few km s^{-1} , respectively. For the illustrations in this section, I use the second configuration in Table 5.9, which reaches $\text{NMI} = 0.65$ and relies solely on the baseline losses.

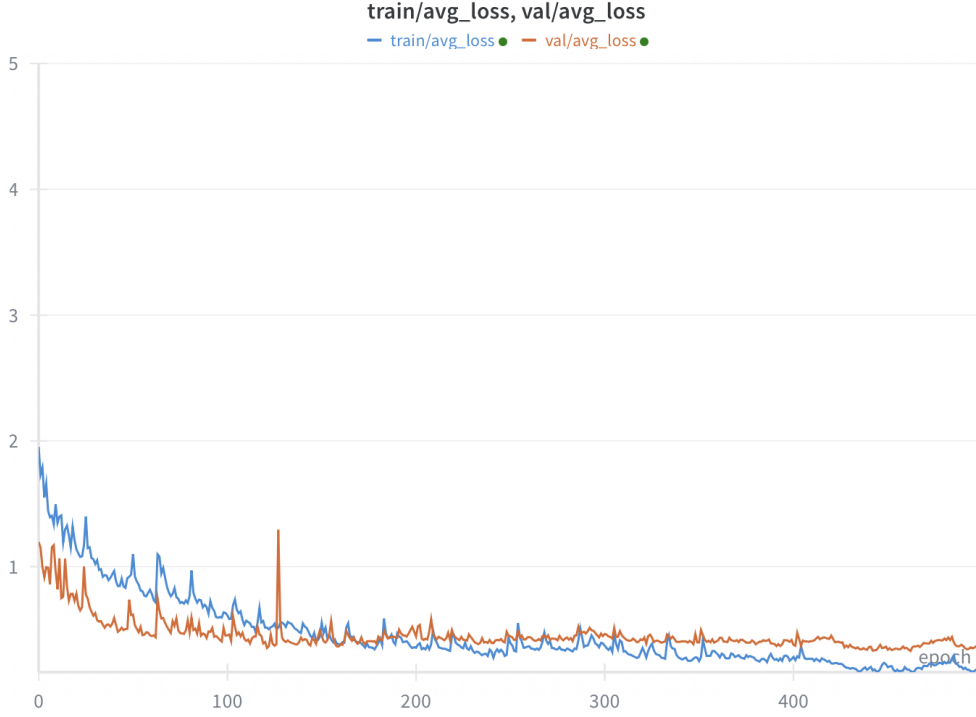


Figure 5.8.: Training and validation loss versus epoch for the chosen DENSE configuration on the challenging sample. Both curves indicate convergence of the training after roughly 150 epochs. This epoch coincides with the point at which the NMI curve in Figure 5.13 levels off.

Figure 5.8 shows the training and validation loss curves for this configuration. As in the simple sample, the validation loss tracks the training loss reasonably well, although the fluctuations are larger, reflecting the increased difficulty of the problem. After roughly 150 epochs, no further systematic improvement is visible. This coincides with the point at which the upper envelope and typical NMI values in Figure 5.13 stop increasing, indicating that the clustering performance has saturated, even though individual HDBSCAN parameter choices still yield a broad range of NMI values at all epochs.

The learned latent-space embedding is shown in Figure 5.9. The five clusters are clearly distinguishable, and the predicted positions and velocities of their member stars appear to be similar to the actual values. In contrast, the background stars are mapped into a more compact region than in the true phase-space distribution. Figure 5.10 shows that the reconstruction is not perfect, but the main phase-space structure of the clusters is still recovered. The error histograms in Figure 5.11 show that most stars are reconstructed to within a few parsecs in position and a few km s^{-1} in velocity, although the tails are broader than in the simple sample. The mean component-wise errors are (3.5, 2.6, 2.1) pc and (4.5, 3.9, 3.2) km s^{-1} .

5. Results

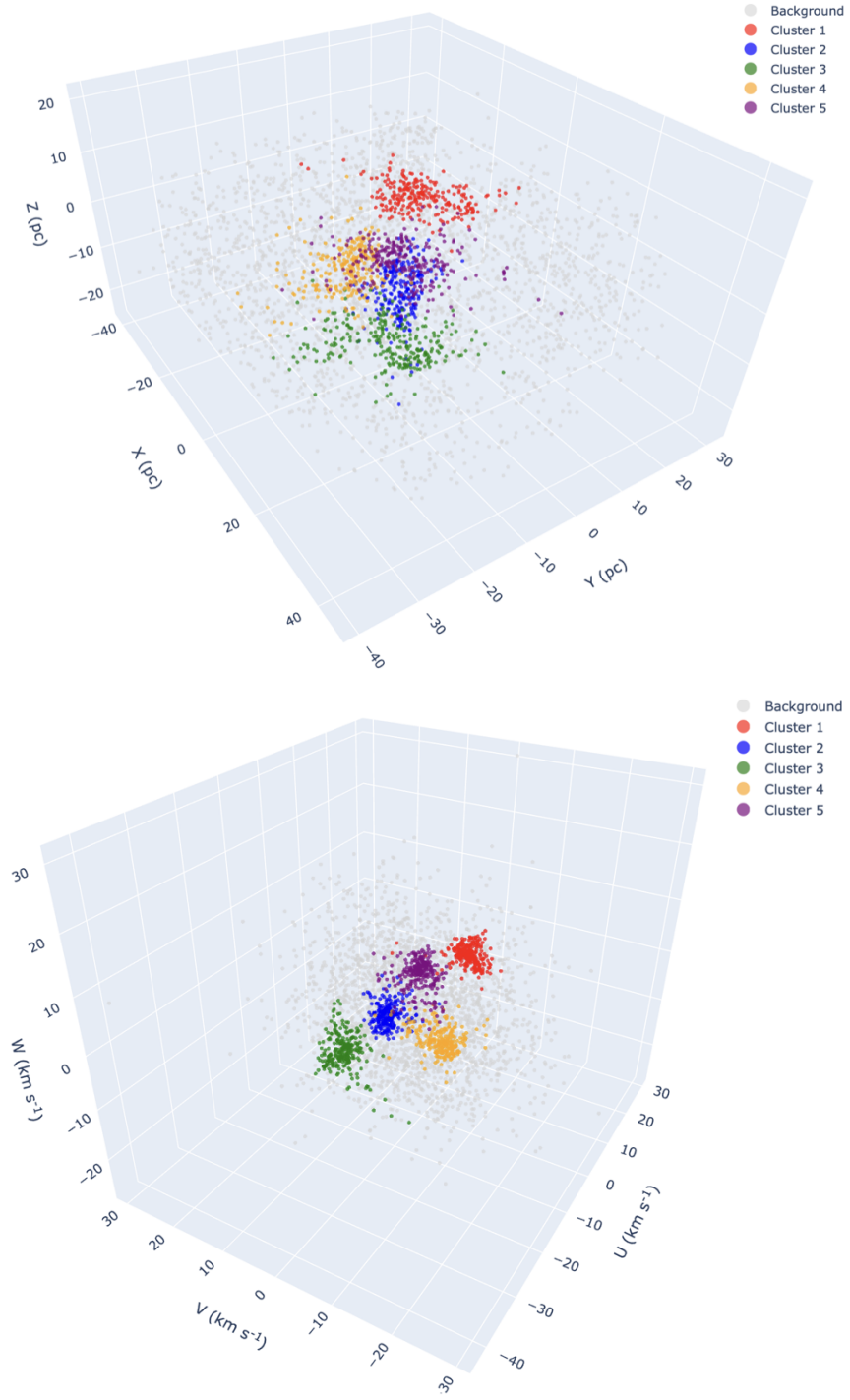


Figure 5.9.: Latent-space embedding of the challenging sample in predicted XYZ (top) and UVW (bottom). Points are colored by their ground-truth cluster labels.

5.5. Best-Performing Configurations

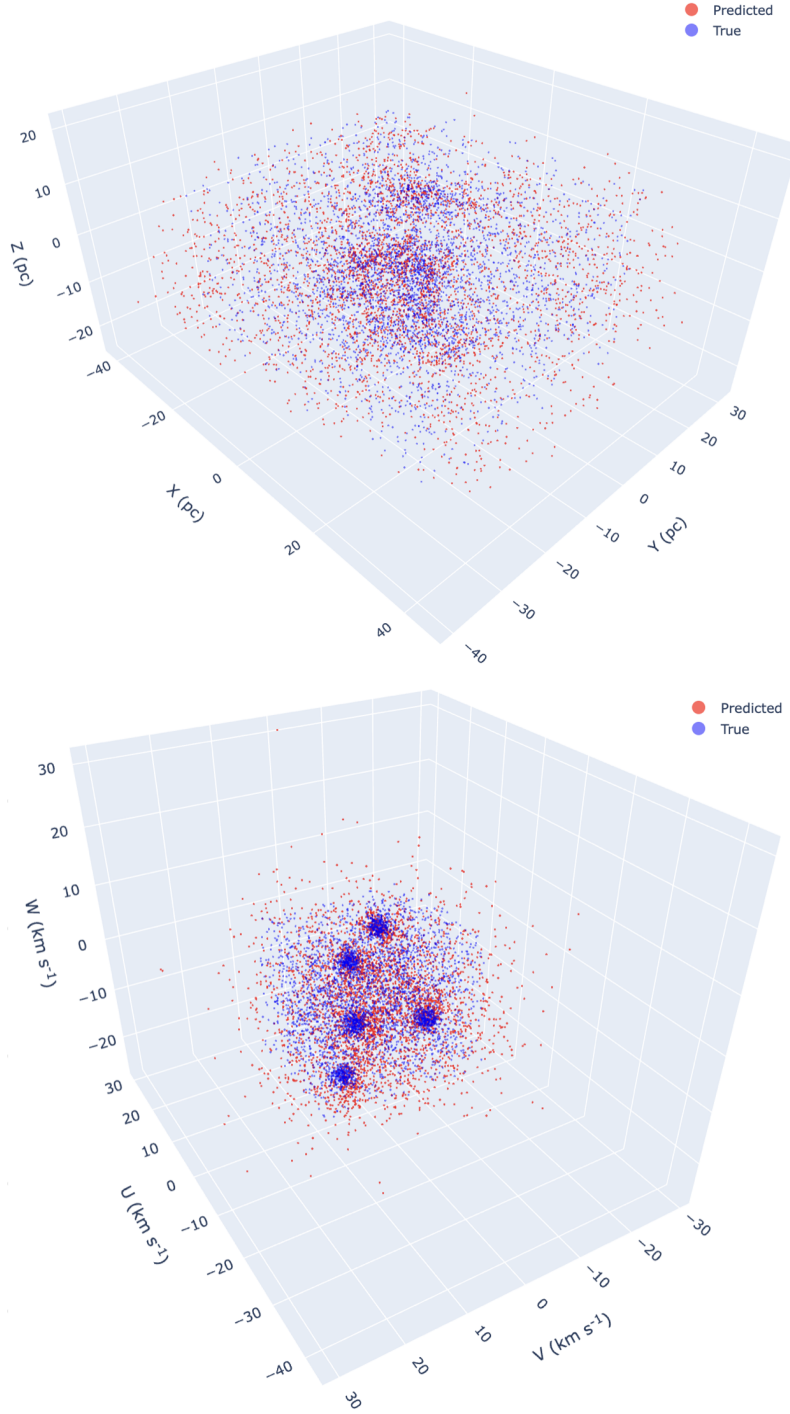


Figure 5.10.: True versus predicted Cartesian coordinates for the challenging sample in positions XYZ (top) and velocities UVW (bottom).

5. Results

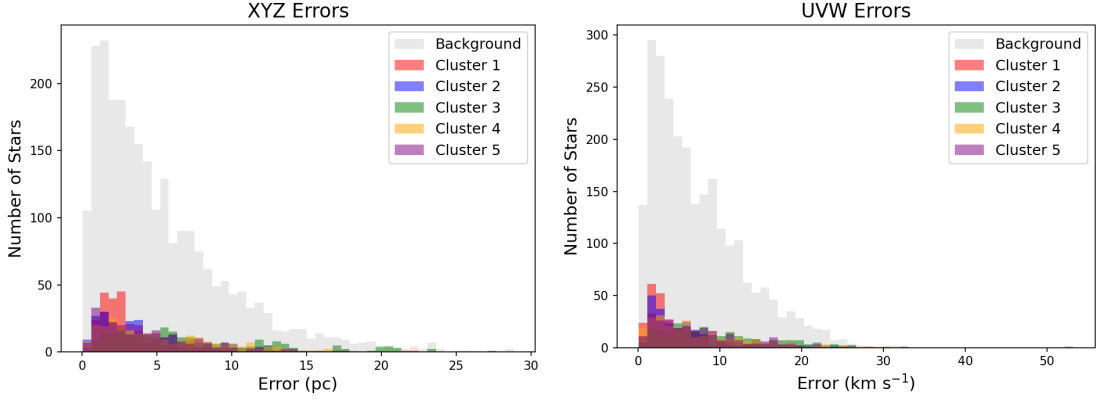


Figure 5.11.: Axis-wise errors in the latent-space positions (left) and velocities (right) for the challenging sample. The mean absolute errors in XYZ are (3.5, 2.6, 2.1) pc and in UVW (4.5, 3.9, 3.2) km s^{-1} .

Figure 5.12 displays the best clustering result obtained in the latent space. All five input clusters are recovered, but their memberships are incomplete. The composition of the recovered groups is listed in Table 5.13. Each cluster retains a substantial fraction of its true members ($\gtrsim 38\%$), yet a non-negligible number of true cluster stars are missed. Contamination from other clusters and from background stars is generally low, at the level of $\sim 1\text{--}8\%$ across the five groups.

Table 5.13.: Composition of the five clusters detected by DENSE in the challenging sample, for the chosen best-performing configuration.

	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5
Total stars	239	182	171	98	153
Background	7	2	3	2	2
True label	232	180	157	95	150
Other label	/	/	11	1	1

The evolution of NMI with epoch and HDBSCAN hyperparameters is summarized in Figure 5.13. The maximum NMI is significantly higher than for direct clustering in 5D observable space (Section 5.1), but still falls short of the ideal 6D upper bound, reflecting the intrinsic difficulty of the challenging sample.

Figure 5.14 compares the clustering solution in the true 6D phase space to that in the DENSE latent space for the challenging sample. The left column shows the detected clusters obtained by applying HDBSCAN directly to the true ($XYZUVW$) coordinates, while the right column shows the clusters found in the DENSE latent space, projected

5.5. Best-Performing Configurations

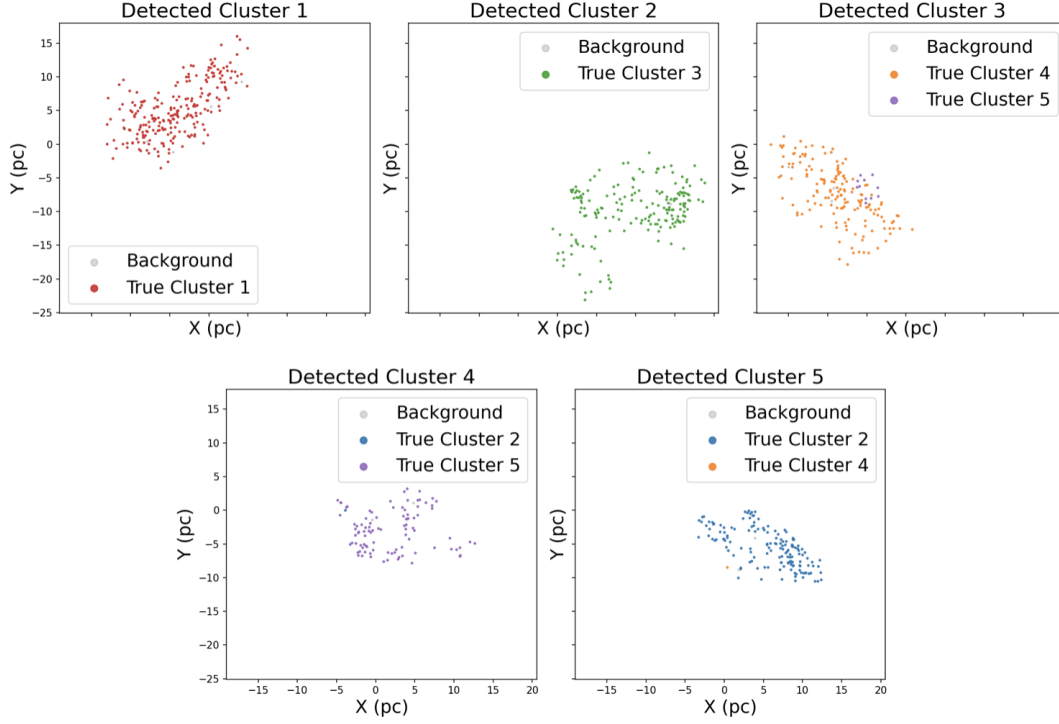


Figure 5.12.: Best HDBSCAN clustering result in the DENSE latent space for the challenging sample. Points are colored by their ground-truth cluster labels.

back to the true XYZ (top) and UVW (bottom) coordinates. The DENSE clusters agree well with the ideal 6D clusters in position and velocity space, but they do not include all true cluster members, so that a fraction of cluster stars remain unassigned.

Best-Case NMI

The clusters in the challenging sample are substantially more compact in velocity than in position, similar to many nearby associations in the Solar neighborhood. In such cases, clustering performance can often be improved by giving more weight to the velocity dimensions. Previous work has suggested scaling the velocity axes by factors of 4–6 relative to position (Kerr et al. 2021). Motivated by this, I adopt a representative velocity scaling factor of $s = 5$ for the challenging sample.

In a realistic application, one would not use reconstructed positions from the network but rather the best available astrometry. I therefore constructed a hybrid 6D space for the challenging sample using the true Cartesian positions and the DENSE-predicted velocities,

$$(X_{\text{true}}, Y_{\text{true}}, Z_{\text{true}}, s U_{\text{pred}}, s V_{\text{pred}}, s W_{\text{pred}}),$$

5. Results

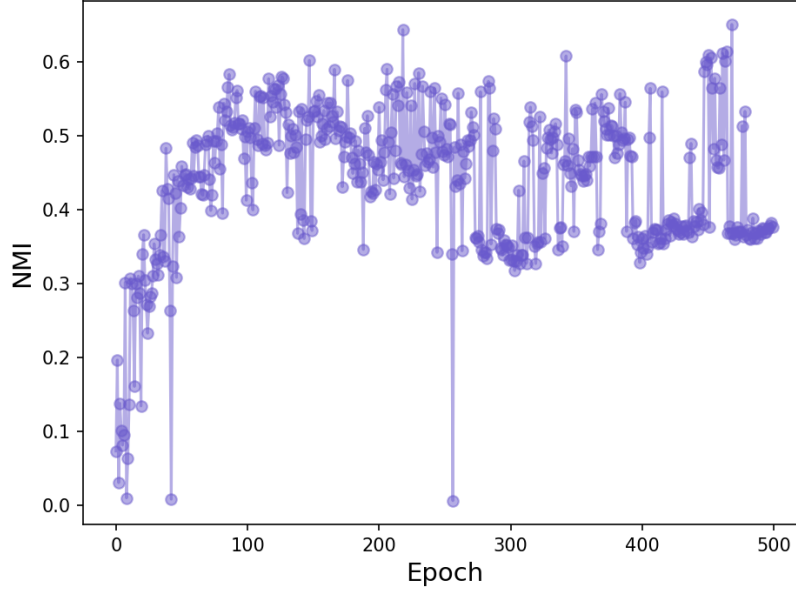


Figure 5.13.: Same as Figure 5.6, but for the challenging sample. At each epoch, multiple HDBSCAN hyperparameter combinations are evaluated, producing a broad spread of NMI values. The upper envelope and typical (median) NMI increase during the first ~ 150 epochs and then remain in the range $\sim 0.5 - 0.65$, with no further systematic trend, consistent with the flattening of the loss curves in Figure 5.8.

and applied HDBSCAN to this space. The true positions $(X_{\text{true}}, Y_{\text{true}}, Z_{\text{true}})$ can be obtained from the observables without radial velocities (Section 2.2.3), so this hybrid representation corresponds to a realistic use case in which *Gaia*-quality positions are combined with 3D velocities inferred by DENSE.

For the best-performing DENSE configuration of the challenging sample, rescaling the velocities by $s = 5$ improves the clustering performance relative to the unscaled hybrid case, achieving an NMI of approximately 0.74. For comparison, applying HDBSCAN directly in a scaled 5D observable space $(\alpha, \delta, \varpi, 5v_\alpha, 5v_\delta)$ without using DENSE yields a maximum NMI of ≈ 0.53 and fragments the true associations into several smaller groups. The DENSE-based best-case result therefore substantially improves on what is achievable in the scaled observable space alone, although it still falls short of a perfect NMI score. Figure 5.15 illustrates the resulting clusters in true (XYZ) and (UVW) for this representative scaling.

Visually, the best-case clusters in Figure 5.15 are very similar to those obtained by clustering directly in the true 6D phase space (Figure 5.14). The main difference is quantitative. Each of the five detected groups contains about 70–95% of the members of a single true

5.5. Best-Performing Configurations

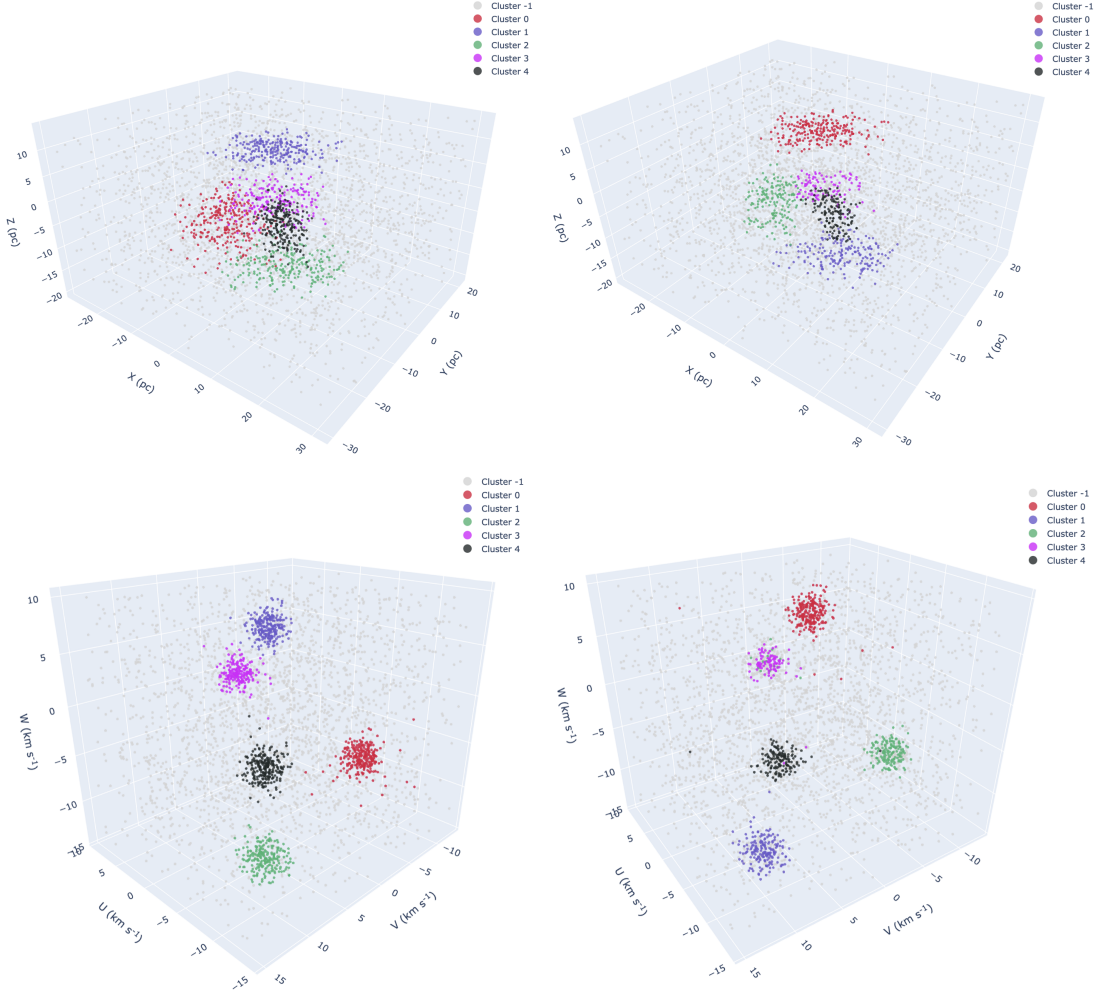


Figure 5.14.: True Galactic Cartesian coordinates for the challenging sample, colored by detected HDBSCAN clusters. Top row: positions XYZ , bottom row: velocities UVW . Left column: clusters obtained by running HDBSCAN directly in the true 6D phase space ($XYZUVW$). Right column: clusters obtained by running HDBSCAN in the DENSE latent space.

association, while contamination from background stars and other clusters remains at only the few-per-cent level.

The exact velocity scaling that would be appropriate for real *Gaia* data is not known a priori and will depend on the intrinsic sizes and internal dispersions of the associations. The experiment presented here should therefore be interpreted as a “best-case” upper limit on the clustering performance that DENSE could achieve when combined with sensibly up-weighted velocity information. Repeating the same experiment for several

5. Results

other well-performing DENSE configurations of the challenging sample yields maximum NMI values in the same range, $\text{NMI} \approx 0.7\text{--}0.8$, indicating that this best-case scenario is robust to reasonable changes in the network hyperparameters.

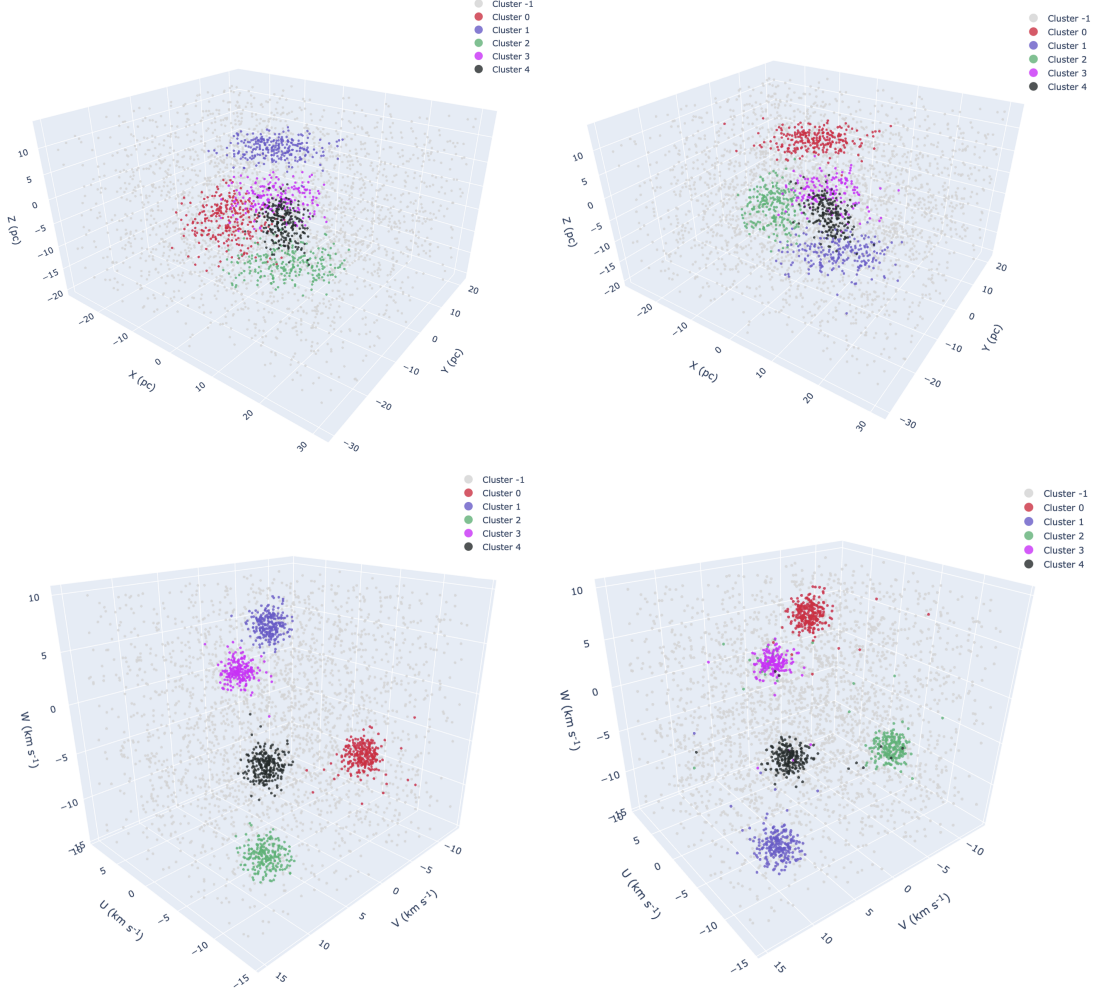


Figure 5.15.: True Galactic Cartesian coordinates for the challenging sample, colored by detected HDBSCAN clusters. Top row: positions XYZ , bottom row: velocities UVW . Left column: clusters obtained by running HDBSCAN directly in the true 6D phase space ($XYZUVW$). Right column: stars plotted at their true coordinates, colored by the clusters obtained from HDBSCAN in the hybrid space ($X_{\text{true}}, Y_{\text{true}}, Z_{\text{true}}, 5U_{\text{pred}}, 5V_{\text{pred}}, 5W_{\text{pred}}$).

5.6. Running Time Analysis

All computations were performed on a MacBook Pro (model Mac16,8, model number MX2H3D/A) with an Apple M4 Pro chip (12-core CPU with 8 performance and 4 efficiency cores, 16-core integrated GPU) and 24 GB Micron LPDDR5 memory. On this machine, training a single DENSE model for $N_{\text{ep}} = 500$ epochs with a representative configuration ($M = 130$, $K = 5$) takes approximately 23 minutes of wall-clock time, dominated by the forward and backward passes through the encoder and by the construction of the $\mathcal{O}(M^2)$ pairwise matrix operations in the loss. The HDBSCAN grid search over all latent representations is comparatively cheap, requiring only about 2 minutes to test all $(m_{\text{clSize}}, m_{\text{pts}})$ combinations from the grid over all epochs. In general, the total runtime is set mainly by the number of epochs, the number of repeats K , and the batch size M , with the cost of the pairwise matrix operations scaling approximately as $\mathcal{O}(M^2)$.

6. Discussion

In this chapter, I discuss the main results of this work, compare DENSE to semi-metric clustering approaches, and mention limitations and future directions, with a view toward applications to real *Gaia* data.

6.1. Main Findings

The results depicted in Section 5.5.1 show that on the simple synthetic sample with three compact Gaussian clusters embedded in a uniform background, DENSE performs essentially optimally. For the selected configuration, the semi-analytic autoencoder reconstructs the true 6D Cartesian phase space with mean position errors of the order $\bar{\epsilon}_{xyz} \simeq 0.5$ pc and mean velocity errors $\bar{\epsilon}_{uvw} \simeq 2$ km s⁻¹, and multiple hyperparameter setups achieve similar values. HDBSCAN applied to the latent representation recovers all three clusters with NMI = 1.0 when background stars are ignored. This matches the performance of HDBSCAN in the true 6D phase space and clearly improves on direct clustering in the 5D astrometric space, which reaches a maximum NMI $\simeq 0.82$ for a wide variety of tested hyperparameters.

The challenging synthetic sample represents a more realistic and demanding regime. It contains five clusters with broader intrinsic dispersions and non-trivial spatial covariances. In this case, HDBSCAN performs poorly in the observable space, reaching at best NMI $\simeq 0.33$, while applied in the true 6D phase space achieves NMI $\simeq 0.9$. DENSE closes a significant part of this gap. Across many hyperparameter choices, clustering with HDBSCAN in the latent representation yields NMI values between about 0.6 and 0.68. The mean reconstruction errors are a few parsecs in position and a few km s⁻¹ in velocity, which is sufficient to preserve the overall phase-space geometry of the clusters. A best-case experiment, in which the true Cartesian positions computed from the astrometry are combined with the DENSE-predicted velocities and the velocity axes are scaled by a factor $s = 5$, pushes the NMI further to ~ 0.7 – 0.8 . The best clustering solutions recover all five input groups, with each detected cluster containing a substantial fraction of the corresponding true members and only little contamination from background or other clusters.

These results show that DENSE can learn a 6D latent representation approximating the underlying true 6D phase space structure from incomplete *Gaia*-like astrometry, such that clustering in this latent space clearly outperforms direct clustering in observable coordinates and is robust against moderate changes in batch construction and loss weights.

6.2. Comparison to Semi-Metric Clustering

Section 3.1 introduced an alternative strategy that works directly with pairwise minimal 3D velocity differences. For each stellar pair, optimal radial velocities minimize the norm of the 3D velocity difference in Galactic Cartesian space. This defines a semi-metric distance matrix that encodes how similar the space motions of two stars could be given their astrometry, and clustering algorithms can then operate in this semi-metric space.

Within our group, this idea has been implemented with the softmax clustering algorithm of Chang and Chang (2017), combined with a grid and consensus scheme. The sky is divided into cells, softmax clustering is run independently in each cell, and consensus methods merge the partial solutions. On the challenging synthetic sample, this pipeline can achieve clustering quality comparable to HDBSCAN in the true 6D space, with NMI values close to 0.9, provided that the grid and consensus parameters are carefully tuned. However, many parameter choices perform substantially worse, yielding NMI values around 0.3 to 0.4.

On the challenging sample, DENSE reaches NMI values of about 0.6 to 0.68. With a hybrid setup in which the true Cartesian positions are combined with DENSE-predicted velocities and the velocity axes are scaled by a factor $s = 5$, this can be improved to best-case NMI values around 0.7–0.8. These scores are higher than those of direct HDBSCAN in observable space, but they are still somewhat lower than the best semi-metric softmax solutions and lower than HDBSCAN in the true 6D phase space. DENSE therefore does not recover all information that is available to an ideal 6D method. In return, it offers three important advantages.

- **Hyperparameter robustness** On the simple sample, almost all DENSE configurations achieve $\text{NMI} = 1.0$, and on the challenging sample, the NMI values remain in a narrow interval around 0.6. The semi-metric softmax pipeline shows a much broader range of outcomes and is very sensitive to grid and consensus parameters. For real *Gaia* data without ground truth, the robustness of DENSE is likely more useful than optimizing a single synthetic benchmark.
- **Scalability** DENSE only requires pairwise computations within batches and does not need to store the full distance matrix. This makes it tractable for very large samples, such as a volume-limited *Gaia* sample within 100 pc.
- **Explicit latent phase space** DENSE produces a 6D Cartesian embedding that approximates the true Galactic phase space and can be useful beyond clustering. It is, e.g., suitable for dynamical interpretations and comparisons with simulations.

In this sense, DENSE and semi-metric softmax clustering are complementary. Softmax can reach higher NMI values on some synthetic tests, while DENSE offers a more stable framework that appears better suited to exploratory studies of real *Gaia* data.

6.3. Limitations and Caveats

Several limitations of the present work need to be considered.

All quantitative results are based on synthetic data. The samples were designed to resemble the Solar neighborhood and include multiple clusters, a background population, and incomplete radial-velocity coverage. However, DENSE may behave differently on more irregular real data.

The NMI values reported in Chapter 5 are computed while ignoring background stars to make them comparable to results from semi-metric clustering approaches. This setting isolates the performance on the clusters.

The matrix loss $\mathcal{L}_{\text{mat}}$ uses a fixed number of nearest neighbors ($k = 10$) to define local neighborhood scales in optimal-velocity and latent-velocity space. This choice was kept fixed in all experiments, but is an additional hyperparameter, and different values of k could shift the emphasis between very local and more global structure during training.

Finally, the DENSE architecture is deliberately simple. The encoder is a fully connected multilayer perceptron, and the loss contains only a small number of handcrafted terms. The ablation study in Section 5.4 shows that all of these components contribute to the final performance but also indicates that the design is not unique. More advanced encoder architectures or alternative loss formulations, for example, losses based on density connectivity, might improve both reconstruction and clustering.

6.4. Future Work

DENSE opens up several natural directions for future work.

On the methodological side, further work can focus on reducing the difference between the latent representation and the true 6D phase space. Possible steps include deeper or residual encoder networks and more explicitly SHADE-inspired loss terms that preserve density connectivity via tree-based or mutual-reachability metrics.

On the clustering side, it will be important to test alternative density-based algorithms in the DENSE latent space. In particular, applying clustering algorithms specifically tuned to astronomical phase space data, such as SigMA (Ratzenböck et al. 2023), to the learned embedding may improve the recovery of diffuse or overlapping associations and could narrow the gap to the ideal 6D HDBSCAN performance on challenging data sets. The best-case experiment in Chapter 5, in which the true Cartesian positions are combined with the up-weighted DENSE-predicted velocities, already increases the NMI from ~ 0.65 to about 0.74 on the challenging sample. This suggests that a better use of

6. Discussion

positional information, axis weighting, and clustering strategies in latent or hybrid spaces is a promising way to further improve performance.

The main motivation of this work is the application to real *Gaia* data. A natural first target is a volume-limited DR3 sample within about 100 pc of the Sun. Known nearby associations, such as β Pictoris, Tucana–Horologium, and Carina, can serve as partial reference sets to assess cluster recovery in the DENSE latent space. After successful testing, DENSE should be applied to the entire local 100 pc sphere, contributing towards a homogeneous cluster census of the Solar neighborhood.

7. Conclusion

The thesis developed and evaluated a methodology for identifying stellar associations in the immediate Solar neighborhood, where strong projection effects and incomplete kinematic information hinder classical density-based clustering algorithms from recovering kinematically coherent groups. The *Gaia* mission provides precise five-parameter astrometry for a vast number of stars, but only a fraction has measured radial velocities. As a result, the full six-dimensional Galactic Cartesian phase space, where projection effects are mitigated, is directly accessible only for a minority of sources.

To address this problem, I introduced DENSE, a semi-analytic neural network model that combines fixed coordinate transformations with a learned encoder that infers missing kinematic dimensions. The encoder maps *Gaia*-like astrometry to a six-dimensional latent representation, designed to approximate underlying Galactic Cartesian coordinates ($XYZUVW$). The decoder is fully analytic and reconstructs observable quantities from this latent space. The training objective couples astrometric reconstruction, pairwise information derived from optimal-velocity differences, and radial-velocity constraints where available. After training, standard density-based clustering algorithms, such as HDBSCAN, are applied in the learned latent space.

On the synthetic data sets constructed for this work, DENSE learned a latent representation in which kinematically coherent structures are more easily separable than in the original observable space. Clustering in the latent space substantially improves upon direct clustering in astrometric coordinates and, for the simpler sample with compact and well-separated clusters, approaches the performance of clustering in the true six-dimensional phase space available in the simulations. The analysis of reconstruction errors, latent space morphology, and ablation experiments shows that the combination of astrometric, pairwise, and radial velocity terms is essential for preserving the underlying structure.

At the same time, the results highlight that incomplete kinematic information and strong projection effects remain significant challenges, especially for samples that are spatially extended and overlapping in position space. In these cases, the best clustering performance in latent space still falls short of the ideal six-dimensional upper bound. This underscores the intrinsic limitations imposed by missing radial velocities and by the blending of cluster signals in the observed tangential velocity space. DENSE should therefore be viewed not as a complete solution, but as a step toward more robust phase space reconstruction under realistic observational constraints.

7. Conclusion

Future work can explore alternative architectures and loss functions that better preserve density connectivity and apply different clustering methods to the learned latent space. The ultimate aim is to apply DENSE to *Gaia* data within about one hundred parsecs and to build a more complete and homogeneous census of young associations in the Solar neighborhood.

Bibliography

- Alves, João et al. (Feb. 2020). ‘A Galactic-scale gas wave in the solar neighbourhood’. In: *Nature* 578(7794), pp. 237–239. DOI: [10.1038/s41586-019-1874-z](https://doi.org/10.1038/s41586-019-1874-z). arXiv: [2001.08748](https://arxiv.org/abs/2001.08748) [astro-ph.GA].
- Ambartsumian, V. A. (Aug. 1947). ‘The Evolution of Stars and Astrophysics’. In: *Communications of the Byurakan Astrophysical Observatory* 69(1), pp. 127–139. DOI: [10.52526/25792776-22.69.1-127](https://doi.org/10.52526/25792776-22.69.1-127).
- Arvai, Kevin (2020). ‘kneed’. In: *Python*. <https://doi.org/10.5281/zenodo.6496267>.
- Bailer-Jones, Coryn A. L. (2015). ‘Estimating Distances from Parallaxes’. In: *Publications of the Astronomical Society of the Pacific* 127(956), pp. 994–1009. DOI: [10.1086/683116](https://doi.org/10.1086/683116).
- Beer, Anna et al. (2024). *SHADE: Deep Density-based Clustering*. arXiv: [2410.06265](https://arxiv.org/abs/2410.06265) [cs.LG]. URL: <https://arxiv.org/abs/2410.06265>.
- Blaauw, Adriaan (Jan. 1964). ‘The O Associations in the Solar Neighborhood’. In: *Annu. Rev. Astron. Astrophys.* 2, p. 213. DOI: [10.1146/annurev.aa.02.090164.001241](https://doi.org/10.1146/annurev.aa.02.090164.001241).
- Campello, Ricardo J. G. B., Davoud Moulavi and Joerg Sander (2013). ‘Density-Based Clustering Based on Hierarchical Density Estimates’. In: *Advances in Knowledge Discovery and Data Mining (PAKDD 2013)*. Vol. 7819. Lecture Notes in Computer Science. Springer: Berlin, Heidelberg, pp. 160–172. DOI: [10.1007/978-3-642-37456-2_14](https://doi.org/10.1007/978-3-642-37456-2_14).
- Cantat-Gaudin, T. et al. (Oct. 2018). ‘A Gaia DR2 view of the open cluster population in the Milky Way’. In: *Astron. Astrophys.* 618, A93, A93. DOI: [10.1051/0004-6361/201833476](https://doi.org/10.1051/0004-6361/201833476). arXiv: [1805.08726](https://arxiv.org/abs/1805.08726) [astro-ph.GA].
- Castro-Ginard, A. et al. (Oct. 2018). ‘A new method for unveiling open clusters in Gaia. New nearby open clusters confirmed by DR2’. In: *Astron. Astrophys.* 618, A59, A59. DOI: [10.1051/0004-6361/201833390](https://doi.org/10.1051/0004-6361/201833390). arXiv: [1805.03045](https://arxiv.org/abs/1805.03045) [astro-ph.GA].
- Chang, Chia-Tai and Cheng-Shang Chang (2017). ‘A Unified Framework for Sampling, Clustering and Embedding Data Points in Semi-Metric Spaces’. In: *CoRR* abs/1708.00316. arXiv: [1708.00316](https://arxiv.org/abs/1708.00316). URL: <http://arxiv.org/abs/1708.00316>.
- de Bruijne, Jos H. J. (June 1999). ‘A refurbished convergent-point method for finding moving groups in the HIPPARCOS Catalogue’. In: *Mon. Not. R. Astron. Soc.* 306(2), pp. 381–393. DOI: [10.1046/j.1365-8711.1999.02643.x](https://doi.org/10.1046/j.1365-8711.1999.02643.x). arXiv: [astro-ph/9903239](https://arxiv.org/abs/astro-ph/9903239) [astro-ph].
- Ester, Martin et al. (1996). ‘A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise’. In: *Proc. of 2nd International Conference on Knowledge Discovery and*, pp. 226–231.
- Gagné, Jonathan (June 2024). ‘A Quick Guide to Nearby Young Associations’. In: *Publ. Astron. Soc. Pac.* 136(6), 063001, p. 063001. DOI: [10.1088/1538-3873/ad4e6a](https://doi.org/10.1088/1538-3873/ad4e6a). arXiv: [2405.12860](https://arxiv.org/abs/2405.12860) [astro-ph.SR].

Bibliography

- Gagné, Jonathan and Jacqueline K. Faherty (Aug. 2018). ‘BANYAN. XIII. A First Look at Nearby Young Associations with Gaia Data Release 2’. In: *Astrophys. J.* 862(2), 138, p. 138. DOI: [10.3847/1538-4357/aaca2e](https://doi.org/10.3847/1538-4357/aaca2e), arXiv: [1805.11715](https://arxiv.org/abs/1805.11715) [astro-ph.SR].
- Gagné, Jonathan et al. (Mar. 2018a). ‘BANYAN. XI. The BANYAN Σ Multivariate Bayesian Algorithm to Identify Members of Young Associations with 150 pc’. In: *Astrophys. J.* 856(1), 23, p. 23. DOI: [10.3847/1538-4357/aaae09](https://doi.org/10.3847/1538-4357/aaae09), arXiv: [1801.09051](https://arxiv.org/abs/1801.09051) [astro-ph.SR].
- Gagné, Jonathan et al. (June 2018b). ‘BANYAN. XII. New Members of Nearby Young Associations from GAIA-Tycho Data’. In: *Astrophys. J.* 860(1), 43, p. 43. DOI: [10.3847/1538-4357/aac2b8](https://doi.org/10.3847/1538-4357/aac2b8), arXiv: [1804.03093](https://arxiv.org/abs/1804.03093) [astro-ph.SR].
- Gagné, Jonathan et al. (Feb. 2014). ‘BANYAN. II. VERY LOW MASS AND SUBSTELLAR CANDIDATE MEMBERS TO NEARBY, YOUNG KINEMATIC GROUPS WITH PREVIOUSLY KNOWN SIGNS OF YOUTH’. In: *The Astrophysical Journal* 783(2), p. 121. DOI: [10.1088/0004-637X/783/2/121](https://doi.org/10.1088/0004-637X/783/2/121), URL: <https://dx.doi.org/10.1088/0004-637X/783/2/121>.
- Gaia Collaboration et al. (Nov. 2016). ‘The Gaia mission’. In: *Astron. Astrophys.* 595, A1, A1. DOI: [10.1051/0004-6361/201629272](https://doi.org/10.1051/0004-6361/201629272), arXiv: [1609.04153](https://arxiv.org/abs/1609.04153) [astro-ph.IM].
- Gaia Collaboration et al. (July 2022). In: *arXiv e-prints*, arXiv:2208.00211, arXiv:2208.00211. DOI: [10.48550/arXiv.2208.00211](https://doi.org/10.48550/arXiv.2208.00211), arXiv: [2208.00211](https://arxiv.org/abs/2208.00211) [astro-ph.GA].
- Goodfellow, Ian, Yoshua Bengio and Aaron Courville (2016). *Deep Learning*. <http://www.deeplearningbook.org>. MIT Press.
- Guo, Xifeng et al. (2017). ‘Improved deep embedded clustering with local structure preservation’. In: *Proceedings of the 26th International Joint Conference on Artificial Intelligence*. IJCAI’17. AAAI Press: Melbourne, Australia, 1753–1759. ISBN: 9780999241103.
- Hunt, Emily L. and Sabine Reffert (Feb. 2021). ‘Improving the open cluster census. I. Comparison of clustering algorithms applied to Gaia DR2 data’. In: *Astron. Astrophys.* 646, A104, A104. DOI: [10.1051/0004-6361/202039341](https://doi.org/10.1051/0004-6361/202039341), arXiv: [2012.04267](https://arxiv.org/abs/2012.04267) [astro-ph.GA].
- Hunt, Emily L. and Sabine Reffert (May 2023). ‘Improving the open cluster census. II. An all-sky cluster catalogue with Gaia DR3’. In: *Astron. Astrophys.* 673, A114, A114. DOI: [10.1051/0004-6361/202346285](https://doi.org/10.1051/0004-6361/202346285), arXiv: [2303.13424](https://arxiv.org/abs/2303.13424) [astro-ph.GA].
- Jain, A. K., M. N. Murty and P. J. Flynn (Sept. 1999). ‘Data clustering: a review’. In: *ACM Comput. Surv.* 31(3), 264–323. ISSN: 0360-0300. DOI: [10.1145/331499.331504](https://doi.org/10.1145/331499.331504), URL: <https://doi.org/10.1145/331499.331504>.
- Kamdar, Harshil, Charlie Conroy and Yuan-Sen Ting (2021). *Stellar Streams in the Galactic Disk: Predicted Lifetimes and Their Utility in Measuring the Galactic Potential*. arXiv: [2106.02050](https://arxiv.org/abs/2106.02050) [astro-ph.GA], URL: <https://arxiv.org/abs/2106.02050>.
- Kamdar, Harshil et al. (Oct. 2019). ‘Stars that Move Together Were Born Together’. In: *The Astrophysical Journal Letters* 884(2), p. L42. ISSN: 2041-8213. DOI: [10.3847/2041-8213/ab4997](https://doi.org/10.3847/2041-8213/ab4997), URL: [http://dx.doi.org/10.3847/2041-8213/ab4997](https://dx.doi.org/10.3847/2041-8213/ab4997).
- Kerr, Ronan M. P. et al. (Aug. 2021). ‘Stars with Photometrically Young Gaia Luminosities Around the Solar System (SPYGLASS). I. Mapping Young Stellar Structures and Their

- Star Formation Histories'. In: *Astrophys. J.* 917(1), 23, p. 23. DOI: [10.3847/1538-4357/ac0251](https://doi.org/10.3847/1538-4357/ac0251). arXiv: [2105.09338](https://arxiv.org/abs/2105.09338) [astro-ph.GA].
- Kharchenko, N. V. et al. (Oct. 2013). 'Global survey of star clusters in the Milky Way. II. The catalogue of basic parameters'. In: *Astron. Astrophys.* 558, A53, A53. DOI: [10.1051/0004-6361/201322302](https://doi.org/10.1051/0004-6361/201322302). arXiv: [1308.5822](https://arxiv.org/abs/1308.5822) [astro-ph.GA].
- Kingma, Diederik P. and Jimmy Ba (2017). *Adam: A Method for Stochastic Optimization*. arXiv: [1412.6980](https://arxiv.org/abs/1412.6980) [cs.LG]. URL: <https://arxiv.org/abs/1412.6980>.
- Kingma, Diederik P. and Max Welling (2014). 'Auto-Encoding Variational Bayes'. In: *Proceedings of the 2nd International Conference on Learning Representations (ICLR 2014)*. arXiv:1312.6114.
- Kounkel, Marina and Kevin Covey (Sept. 2019). 'Untangling the Galaxy. I. Local Structure and Star Formation History of the Milky Way'. In: *Astron. J.* 158(3), 122, p. 122. DOI: [10.3847/1538-3881/ab339a](https://doi.org/10.3847/1538-3881/ab339a). arXiv: [1907.07709](https://arxiv.org/abs/1907.07709) [astro-ph.GA].
- Kramer, Mark A. (1991). 'Nonlinear principal component analysis using autoassociative neural networks'. In: *Aiche Journal* 37, pp. 233–243. URL: <https://api.semanticscholar.org/CorpusID:15907287>.
- Kraus, Adam L. et al. (May 2014). 'A STELLAR CENSUS OF THE TUCANA–HOROLOGIIUM MOVING GROUP'. In: *The Astronomical Journal* 147(6), p. 146. DOI: [10.1088/0004-6256/147/6/146](https://doi.org/10.1088/0004-6256/147/6/146). URL: <https://dx.doi.org/10.1088/0004-6256/147/6/146>.
- Krone-Martins, A. and A. Moitinho (2014). 'UPMASK: Unsupervised photometric membership assignment in stellar clusters'. In: *Astronomy & Astrophysics* 561, A57. DOI: [10.1051/0004-6361/201321143](https://doi.org/10.1051/0004-6361/201321143).
- Lallement, R. et al. (May 2019). 'Gaia-2MASS 3D maps of Galactic interstellar dust within 3 kpc'. In: *Astron. Astrophys.* 625, A135, A135. DOI: [10.1051/0004-6361/201834695](https://doi.org/10.1051/0004-6361/201834695). arXiv: [1902.04116](https://arxiv.org/abs/1902.04116) [astro-ph.GA].
- Levy-Kramer, Josh (Apr. 2018). *k-means-constrained*. URL: <https://github.com/joshlk/k-means-constrained>.
- Li, Tao, Chris Ding and Michael I. Jordan (2007). 'Solving Consensus and Semi-supervised Clustering Problems Using Nonnegative Matrix Factorization'. In: *Proceedings of the Seventh IEEE International Conference on Data Mining (ICDM 2007)*. IEEE, pp. 577–582.
- MacQueen, J. B. (1967). 'Some Methods for Classification and Analysis of MultiVariate Observations'. In: *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability*. Ed. by L. M. Le Cam and J. Neyman. Vol. 1. University of California Press, pp. 281–297.
- Makhzani, Alireza and Brendan J. Frey (2013). 'k-Sparse Autoencoders'. In: *CoRR* abs/1312.5663. URL: <https://api.semanticscholar.org/CorpusID:14850799>.
- Malo, Lison et al. (Dec. 2013). 'BAYESIAN ANALYSIS TO IDENTIFY NEW STAR CANDIDATES IN NEARBY YOUNG STELLAR KINEMATIC GROUPS*'. In: *The Astrophysical Journal* 762(2), p. 88. DOI: [10.1088/0004-637X/762/2/88](https://doi.org/10.1088/0004-637X/762/2/88). URL: <https://dx.doi.org/10.1088/0004-637X/762/2/88>.

Bibliography

- Meingast, Stefan and João Alves (Jan. 2019). ‘Extended stellar systems in the solar neighborhood. I. The tidal tails of the Hyades’. In: *Astron. Astrophys.* 621, L3, p. L3. DOI: [10.1051/0004-6361/201834622](https://doi.org/10.1051/0004-6361/201834622), arXiv: [1811.04931](https://arxiv.org/abs/1811.04931) [[astro-ph.GA](#)].
- Meingast, Stefan, João Alves and Verena Fürnkranz (Feb. 2019). ‘Extended stellar systems in the solar neighborhood . II. Discovery of a nearby 120° stellar stream in Gaia DR2’. In: *Astron. Astrophys.* 622, L13, p. L13. DOI: [10.1051/0004-6361/201834950](https://doi.org/10.1051/0004-6361/201834950), arXiv: [1901.06387](https://arxiv.org/abs/1901.06387) [[astro-ph.GA](#)].
- Meingast, Stefan, João Alves and Alena Rottensteiner (Jan. 2021). ‘Extended stellar systems in the solar neighborhood. V. Discovery of coronae of nearby star clusters’. In: *Astron. Astrophys.* 645, A84, A84. DOI: [10.1051/0004-6361/202038610](https://doi.org/10.1051/0004-6361/202038610), arXiv: [2010.06591](https://arxiv.org/abs/2010.06591) [[astro-ph.GA](#)].
- Montes, D. et al. (Nov. 2001). ‘Late-type members of young stellar kinematic groups - I. Single stars’. In: *Mon. Not. R. Astron. Soc.* 328(1), pp. 45–63. DOI: [10.1046/j.1365-8711.2001.04781.x](https://doi.org/10.1046/j.1365-8711.2001.04781.x), arXiv: [astro-ph/0106537](https://arxiv.org/abs/astro-ph/0106537) [[astro-ph](#)].
- Moranta, Leslie et al. (Nov. 2022). ‘New Coronae and Stellar Associations Revealed by a Clustering Analysis of the Solar Neighborhood’. In: *The Astrophysical Journal* 939(2), p. 94. ISSN: 1538-4357. DOI: [10.3847/1538-4357/ac8c25](https://doi.org/10.3847/1538-4357/ac8c25), URL: <http://dx.doi.org/10.3847/1538-4357/ac8c25>.
- Naik, Aneesh P and Axel Widmark (Dec. 2023). ‘The missing radial velocities of Gaia: a catalogue of Bayesian estimates for DR3’. In: *Monthly Notices of the Royal Astronomical Society* 527(4), 11559–11574. ISSN: 1365-2966. DOI: [10.1093/mnras/stad3822](https://doi.org/10.1093/mnras/stad3822), URL: <http://dx.doi.org/10.1093/mnras/stad3822>.
- Perryman, M. A. C. et al. (July 1997). ‘The HIPPARCOS Catalogue’. In: *Astron. Astrophys.* 323, pp. L49–L52.
- Ranzato, Marc’ Aurelio, Y-Lan Boureau and Yann LeCun (2007). ‘Sparse feature learning for deep belief networks’. In: *Proceedings of the 21st International Conference on Neural Information Processing Systems*. NIPS’07. Curran Associates Inc.: Red Hook, NY, USA, 1185–1192. ISBN: 9781605603520.
- Ratzenböck, Sebastian et al. (Sept. 2023a). ‘Significance mode analysis (SigMA) for hierarchical structures. An application to the Sco-Cen OB association’. In: *Astron. Astrophys.* 677, A59, A59. DOI: [10.1051/0004-6361/202243690](https://doi.org/10.1051/0004-6361/202243690), arXiv: [2211.14225](https://arxiv.org/abs/2211.14225) [[astro-ph.GA](#)].
- Ratzenböck, Sebastian et al. (Feb. 2023b). ‘The star formation history of the Sco-Cen association. Coherent star formation patterns in space and time’. In: *arXiv e-prints*, arXiv:2302.07853, arXiv:2302.07853. DOI: [10.48550/arXiv.2302.07853](https://doi.org/10.48550/arXiv.2302.07853), arXiv: [2302.07853](https://arxiv.org/abs/2302.07853) [[astro-ph.SR](#)].
- Ratzenböck, Sebastian et al. (Feb. 2025). ‘Toward the fabric of the Milky Way: I. The density of disk streams from a local 2503 pc³ volume’. In: *Astronomy & Astrophysics* 694, A307. ISSN: 1432-0746. DOI: [10.1051/0004-6361/202451866](https://doi.org/10.1051/0004-6361/202451866), URL: <http://dx.doi.org/10.1051/0004-6361/202451866>.
- Riedel, Adric R. et al. (Mar. 2017). ‘LACEwING: A New Moving Group Analysis Code’. In: *Astron. J.* 153(3), 95, p. 95. DOI: [10.3847/1538-3881/153/3/95](https://doi.org/10.3847/1538-3881/153/3/95), arXiv: [1702.02219](https://arxiv.org/abs/1702.02219) [[astro-ph.SR](#)].

- Strehl, Alexander and Joydeep Ghosh (2003). ‘Cluster ensembles – a knowledge reuse framework for combining multiple partitions’. In: *Journal of Machine Learning Research* 3, pp. 583–617.
- Tarricq, Y. et al. (Mar. 2022). ‘Structural parameters of 389 local open clusters’. In: *Astron. Astrophys.* 659, A59, A59. DOI: [10.1051/0004-6361/202142186](https://doi.org/10.1051/0004-6361/202142186), arXiv: [2111.05291](https://arxiv.org/abs/2111.05291) [astro-ph.GA].
- Torres, Carlos A. O. et al. (2008). *Young Nearby Loose Associations*. arXiv: [0808.3362](https://arxiv.org/abs/0808.3362) [astro-ph]. URL: <https://arxiv.org/abs/0808.3362>.
- Vincent, Pascal et al. (2008). ‘Extracting and Composing Robust Features with Denoising Autoencoders’. In: *Proceedings of the 25th International Conference on Machine Learning (ICML 2008)*. ACM, pp. 1096–1103. DOI: [10.1145/1390156.1390294](https://doi.org/10.1145/1390156.1390294).
- Voges, W. et al. (Sept. 1999). ‘The ROSAT all-sky survey bright source catalogue’. In: *Astron. Astrophys.* 349, pp. 389–405. DOI: [10.48550/arXiv.astro-ph/9909315](https://doi.org/10.48550/arXiv.astro-ph/9909315), arXiv: [astro-ph/9909315](https://arxiv.org/abs/astro-ph/9909315) [astro-ph].
- Wilkinson, M. I. et al. (June 2005). ‘Spectroscopic survey of the Galaxy with Gaia-II. The expected science yield from the Radial Velocity Spectrometer’. In: *Monthly Notices of the Royal Astronomical Society* 359(4), 1306–1335. ISSN: 1365-2966. DOI: [10.1111/j.1365-2966.2005.09012.x](https://doi.org/10.1111/j.1365-2966.2005.09012.x). URL: <http://dx.doi.org/10.1111/j.1365-2966.2005.09012.x>.
- Xie, Junyuan, Ross Girshick and Ali Farhadi (2016). *Unsupervised Deep Embedding for Clustering Analysis*. arXiv: [1511.06335](https://arxiv.org/abs/1511.06335) [cs.LG]. URL: <https://arxiv.org/abs/1511.06335>.
- Xu, Rui and D. Wunsch (2005). ‘Survey of clustering algorithms’. In: *IEEE Transactions on Neural Networks* 16(3), pp. 645–678. DOI: [10.1109/TNN.2005.845141](https://doi.org/10.1109/TNN.2005.845141).
- Yang, Bo et al. (Aug. 2017). ‘Towards K-means-friendly Spaces: Simultaneous Deep Learning and Clustering’. In: *Proceedings of the 34th International Conference on Machine Learning*. Ed. by Doina Precup and Yee Whye Teh. Vol. 70. Proceedings of Machine Learning Research. PMLR, pp. 3861–3870. URL: <https://proceedings.mlr.press/v70/yang17b.html>.
- Zuckerman, B. and Inseok Song (Sept. 2004). ‘Young Stars Near the Sun’. In: *Annu. Rev. Astron. Astrophys.* 42(1), pp. 685–721. DOI: [10.1146/annurev.astro.42.053102.134111](https://doi.org/10.1146/annurev.astro.42.053102.134111).

A. Appendix

A.1. Ad Fig. 2.4

To produce Fig. 2.4, I downloaded data from the [Gaia DR3 archive](#), selecting stars within 1 kpc that have high-quality astrometry using the following ADQL query:

```
1 SELECT source_id, ra, dec, parallax, parallax_error, pmra, pmra_error,
   pmdec, pmdec_error, ruwe
2 FROM gaiadr3.gaia_source
3 WHERE parallax_over_error > 10
4 AND ruwe < 1.4
5 AND parallax > 1
```

I computed the distances as $d = 1000/\varpi$ and the total proper motions as $\mu = \sqrt{\mu_{\alpha*}^2 + \mu_{\delta}^2}$. I binned the data by distance and calculated the standard deviation of μ in each bin to estimate the proper motion dispersion. Then I applied the [KneeLocator](#) algorithm ([Arvai 2020](#)) to $\sigma_{\mu}(d)$ to identify the characteristic distance cutoff.

A.2. Learning Curves of Individual Loss Terms

The training curves of individual loss terms discussed in the ablation study in Section 5.4 are depicted in Figures A.1–A.6. These plots show that the astrometric reconstruction loss ($\mathcal{L}_{\text{ast}}$) has the largest impact on the overall learning curve. Runs that include $\mathcal{L}_{\text{ast}}$ exhibit rapid and stable convergence, while runs without it tend to converge more slowly or to higher residual loss values.

In contrast, the losses based solely on radial velocities and latent UVW , $\mathcal{L}_{\text{rv}}$, and $\mathcal{L}_{\text{uvw}}$ show clear signs of overfitting when used in isolation. The training loss continues to decrease while the validation loss flattens or even increases (see Figures A.3 and A.6). Neither term alone provides a sufficiently rich or stable training objective. In the full configuration, these terms act as auxiliary constraints on top of the matrix and astrometric losses, and the combined loss exhibits no such overfitting.

The training curves for the sophisticated example from the ablation study are not shown here, but they follow the same qualitative trends.

A. Appendix

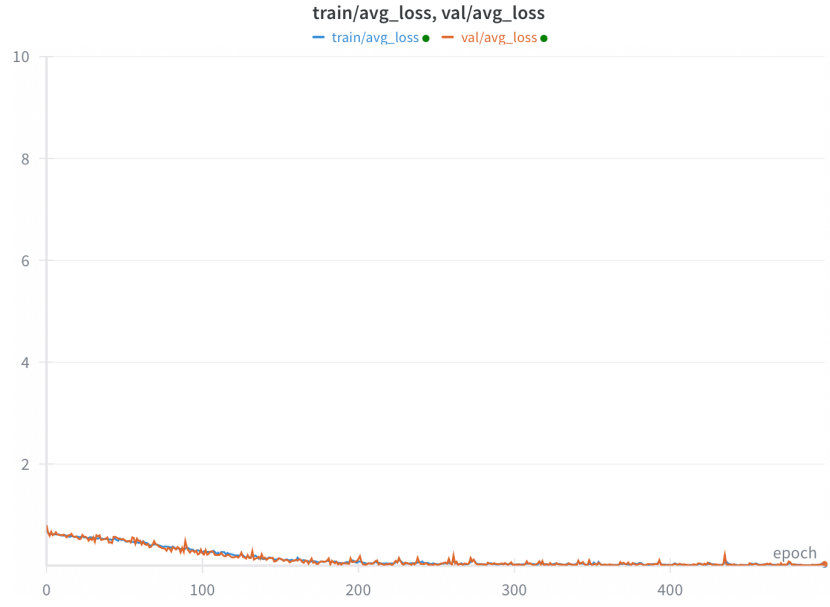


Figure A.1.: Training and validation loss for runs using only the matrix loss $\mathcal{L}_{\text{mat}}$ on the simple sample.

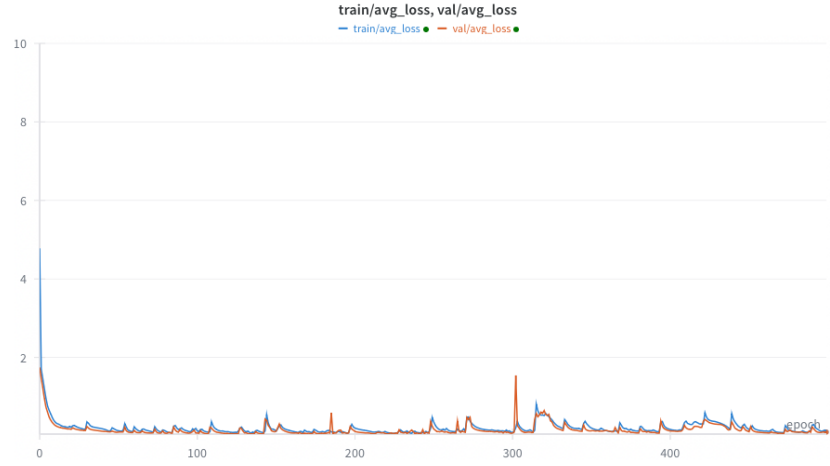


Figure A.2.: Training and validation loss for runs using only the astrometric reconstruction loss $\mathcal{L}_{\text{ast}}$ on the simple sample.

A.2. Learning Curves of Individual Loss Terms

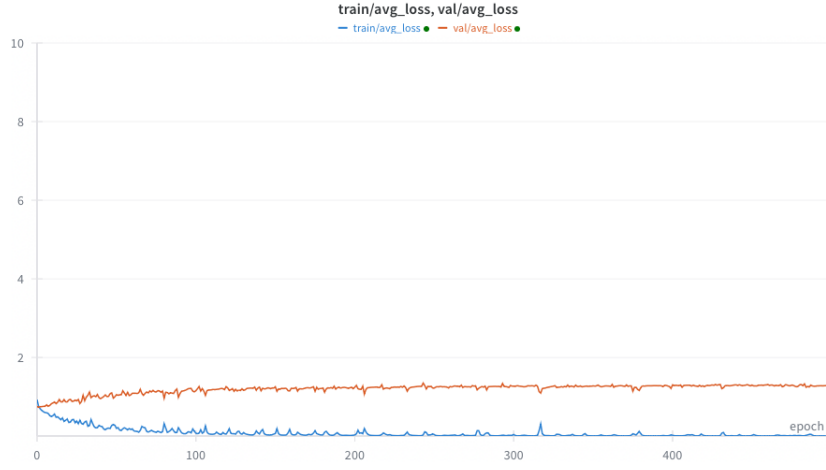


Figure A.3.: Training and validation loss for runs using only the MSE radial velocity loss $\mathcal{L}_{rv}$ on the simple sample.

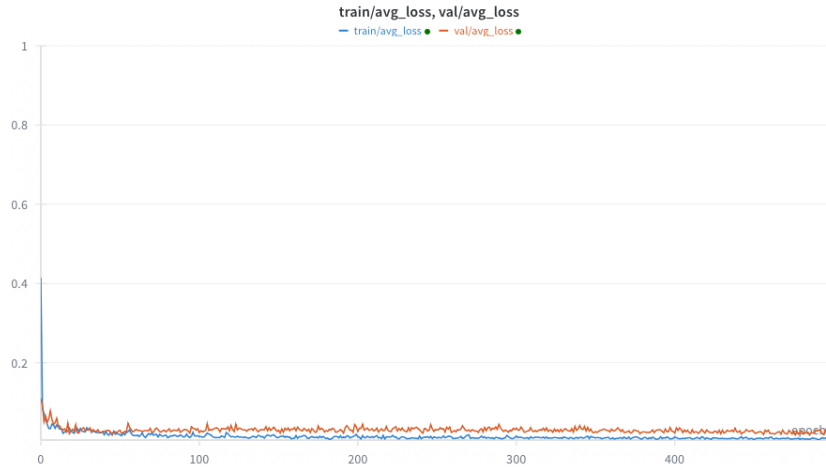


Figure A.4.: Training and validation loss for runs using only the Wasserstein distance loss $\mathcal{L}_{ws}$ on the simple sample.

A. Appendix

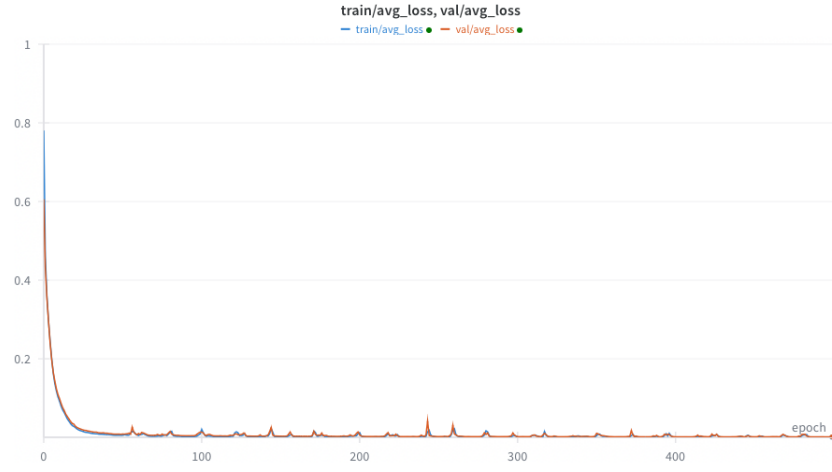


Figure A.5.: Training and validation loss for runs using only the XYZ position loss $\mathcal{L}_{xyz}$ on the simple sample.

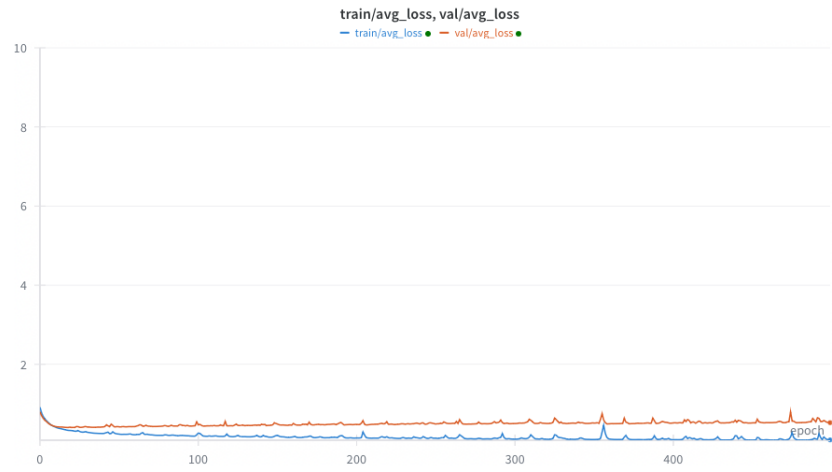


Figure A.6.: Training and validation loss for runs using only the UVW velocity loss $\mathcal{L}_{uvw}$ on the simple sample.