Kinematic and Structural Analysis of Young Stellar Associations: CFN and SCYA-89

Mayesha Rahaman

Introduction

This project investigates the structure and kinematics of two young stellar associations: Cepheus Far North (CFN) and SCYA-89. Using astrometric and photometric data, we perform coordinate transformations, velocity calculations, clustering analysis, and age estimation. Python scripts were used for data processing, figure generation, and optimization routines. This report summarizes the approach and findings in detail, organized step-by-step according to the assignment guidelines.

Main Project

1. RA/Dec Coordinates and Proper Motions

We begin by loading the CFN and SCYA-89 datasets into pandas DataFrames. For each association, we plot the right ascension vs declination, as well as the proper motions in RA and Dec (pmra vs pmdec) to visualize their sky distribution and motion. For CFN, we apply a 180-degree shift to the right ascension values to eliminate a discontinuity.

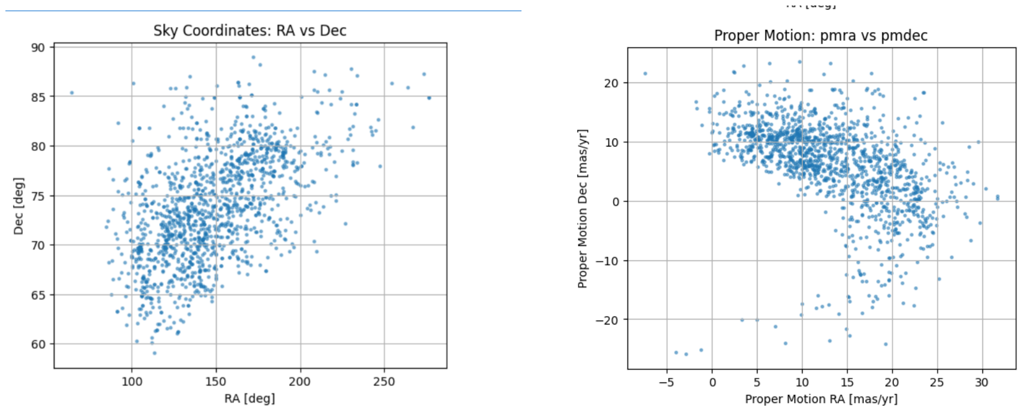

Figure 1: Left: Sky coordinates of CFN in RA vs Dec. Right: Proper motions in RA and Dec.

Figure 1 provides an overview of both the spatial and kinematic trends of CFN members. These visualizations reveal the spatial distribution and bulk motion of the association.

2. Galactic Coordinate Conversion

We convert RA/Dec coordinates to Galactic coordinates (ℓ, b) using the provided gal function which uses the Astropy `SkyCoord` transformation. For the CFN association, we verify the results against the SPYGLASS interactive visualization to ensure the axes values line up, then use the same code for SCYA-89.

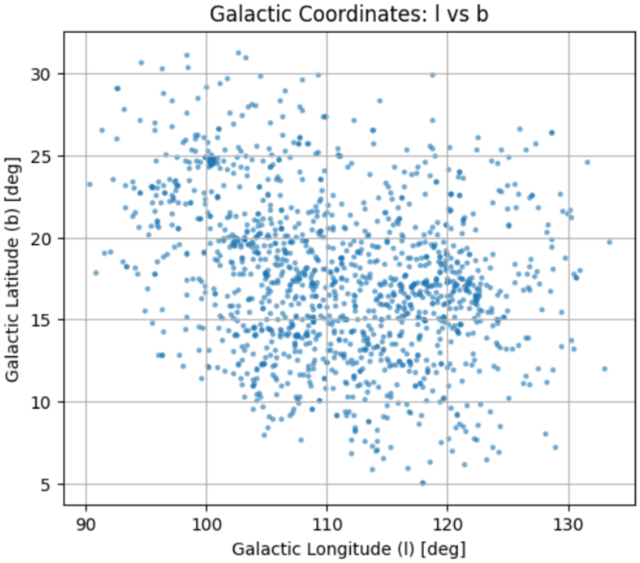

Figure 2: Galactic sky coordinates (ℓ, b)

Figure 2 shows the distribution of stars in Galactic coordinates.

3. Galactic Cartesian Coordinates

The galactic Cartesian coordinates (X, Y, Z) are computed using the provided `gal_xyz` function. This function takes Galactic longitude ℓ, latitude b (or optionally RA and Dec), and either parallax or distance to return Cartesian positions in parsecs. The conversion is based on the following equations:

$$x = d \cdot \cos b \cdot \cos \ell \tag{1}$$

$$y = d \cdot \cos b \cdot \sin \ell \tag{2}$$

$$z = d \cdot \sin b \tag{3}$$

where ℓ and b are in radians and d is the distance in parsecs. These coordinates align the x-axis away from the Galactic center (positive toward the anticenter), consistent with the definition of the U velocity component. As seen in Figure 3, we generate three 2D projections: X vs Y, X vs Z, and Y vs Z, and compare them to SPYGLASS-II Figure 6.

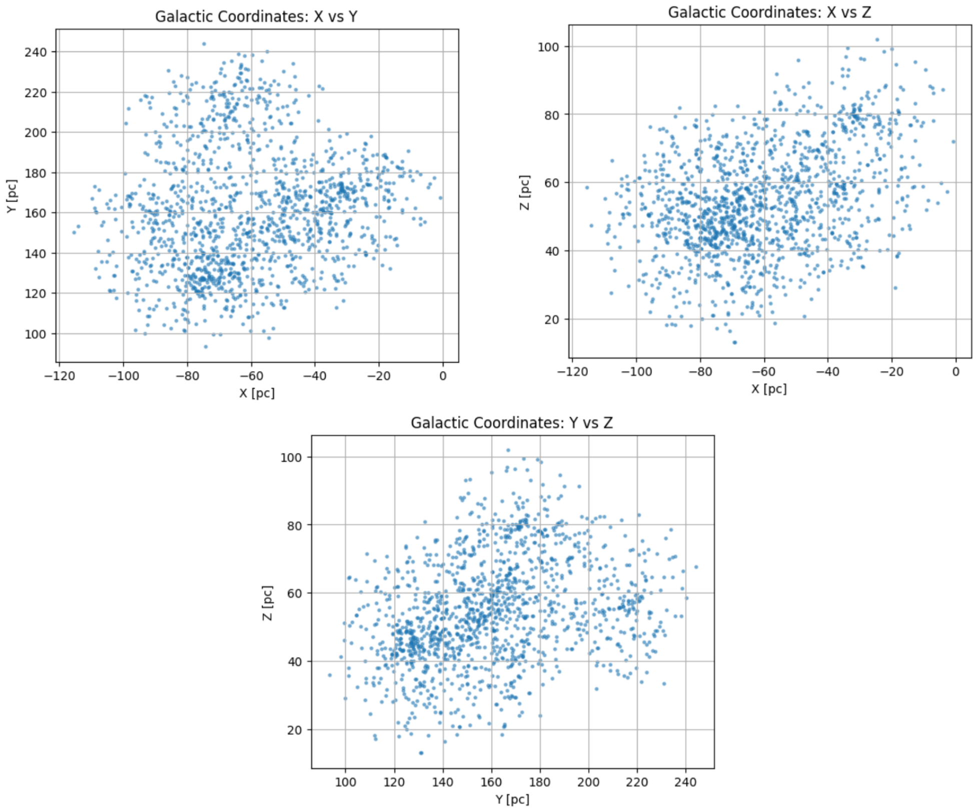

Figure 3: Cartesian projections in Galactic coordinates. Left: X vs Y. Center: X vs Z. Right: Y vs Z.

4. Transverse Velocities

To compute transverse velocities from proper motion, we first convert proper motions from RA/Dec to Galactic coordinates using the `pmgal` function, which uses `astropy`'s built-in transformation tools. We then apply the function `Vt`, which calculates transverse velocity components $v_{t\ell}$ and v_{tb} using:

$$v_{t\ell} = \frac{d \cdot pml}{206264807} \cdot \texttt{pctokm} \div \texttt{ytos}, \quad v_{tb} = \frac{d \cdot pmb}{206264807} \cdot \texttt{pctokm} \div \texttt{ytos} \tag{4}$$

where d is distance in parsecs, pml, pmb are proper motions in Galactic longitude and latitude (in mas/yr), `pctokm` is the number of kilometers in a parsec, and `ytos` converts years to seconds. These equations reflect the internal calculation and ensure proper unit handling. We visualize the resulting velocity field by plotting $v_{t\ell}$ vs v_{tb}.

5. UVW Space Velocities

Full 3D space velocities (U, V, W) are obtained using the `gal_uvw` function from pyutils, which implements the transformation described by Johnson & Soderblom (1987). This method converts proper motion, radial velocity, and distance into Cartesian velocities in the Galactic frame, where:

- U is positive toward the Galactic anticenter
- V is positive in the direction of Galactic rotation

- W is positive toward the North Galactic Pole

We apply this to our dataset by inputting RA, Dec, proper motions (pmra/pmdec), radial velocities, and distances. In Figure 4, we plot U vs V, U vs W, and V vs W to visualize the kinematic distribution. We also compute the average values of each velocity component.

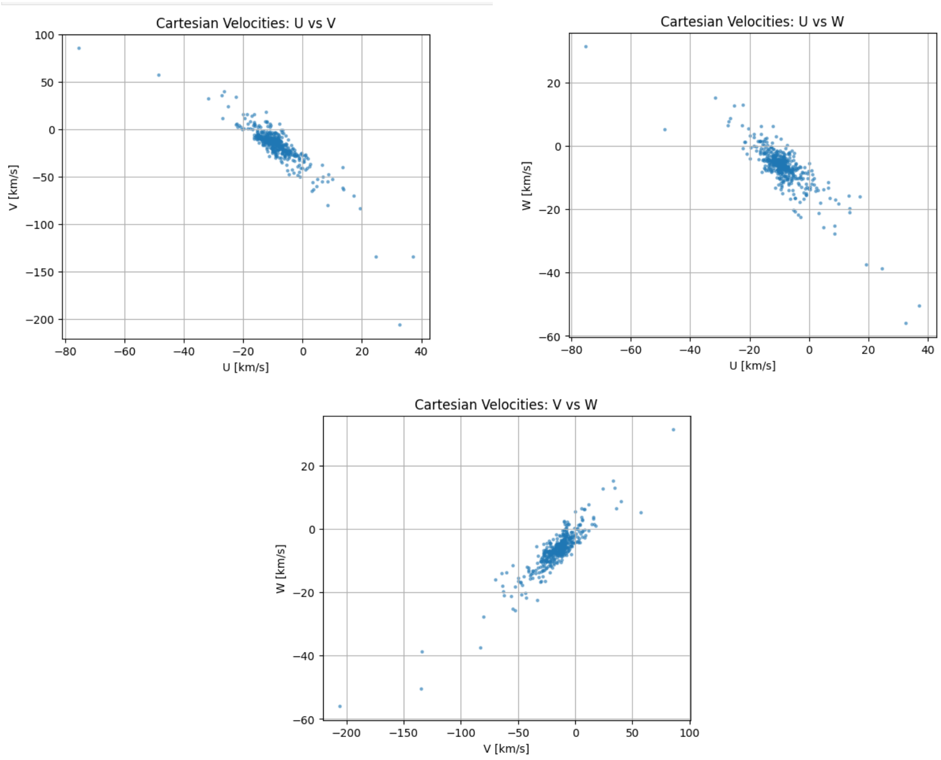

Figure 4: Cartesian Velocities. Left: U vs V. Center: U vs W. Right: V vs W.

6. Projected UVW Vectors

We compute the projected transverse velocity components corresponding to the mean bulk motion of the association. First, we create arrays filled with the average values of U, V, and W computed in the previous step. These constant vectors are then passed into the `propmot` function, which inverts the UVW transformation and returns the radial velocity and proper motions in RA and Dec corresponding to the mean space velocity.

Proper motions are then converted back into Galactic coordinates using `pmgal`, and the projected transverse velocity components $v_{t\ell}^{\mathrm{proj}}, v_{tb}^{\mathrm{proj}}$ are calculated using the `Vt` function as described in Section 4.

7. Transverse Velocity Anomalies

Subtracting the projected mean UVW transverse velocity vector from each star's transverse velocity yields the transverse velocity anomaly. This helps identify outliers and internal kinematic structure, which we compare to SPYGLASS-II Figure 7 for CFN.

Subtracting the projected mean UVW transverse velocity vector from each star's transverse velocity yields the transverse velocity anomaly. This highlights deviations from bulk motion. These residuals are then plotted in Figure 5 to assess internal kinematic structure. The results are compared to SPYGLASS-II Figure 7 for CFN.

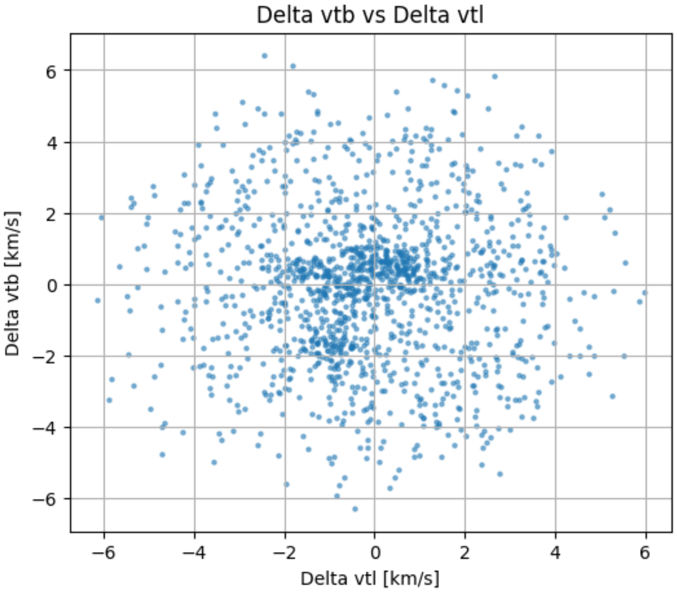

Figure 5: Transverse velocity anomalies. Each point represents the deviation of a star's transverse velocity from the projected mean UVW vector, plotted in the $(\Delta v_{t\ell}, \Delta v_{tb})$ plane.

8. Repeat for SCYA-89

Steps 1 through 7 are repeated for SCYA-89. To reduce contamination and isolate pre-main sequence stars, we apply three filtering conditions:

- $P_{\text{mem}} > 0.3$ to retain high-probability spatial members,
- $P_{\text{sp}} > 0.2$, where P_{sp} is the joint spatial-photometric probability derived from P_{mem} and P_{young} using a Bayesian prior $P_i = 50/11200$,
- Non-zero astrometric flags,
- Non-zero photometric flags.

This ensures we retain reliable astrometric and photometric data and exclude spurious measurements. The same visualizations are reproduced below for SCYA-89.

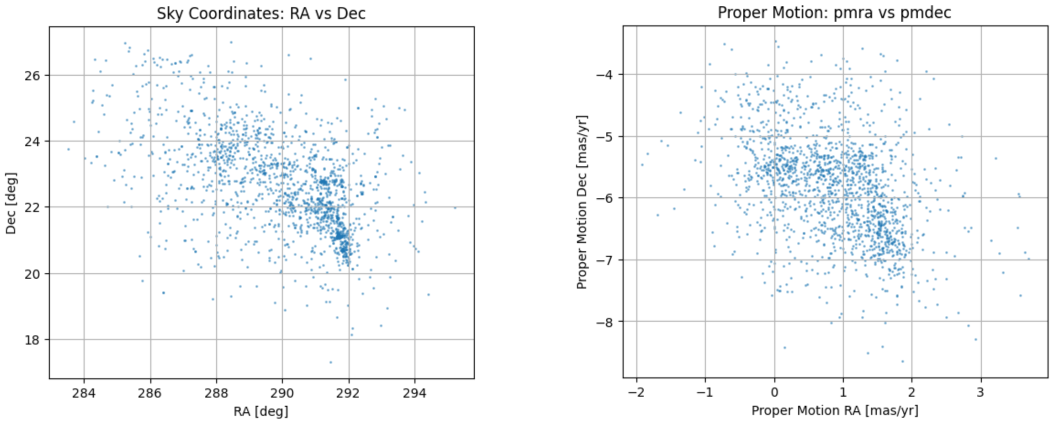

Figure 6: Left: Sky coordinates in RA vs Dec. Right: Proper motions in RA and Dec.

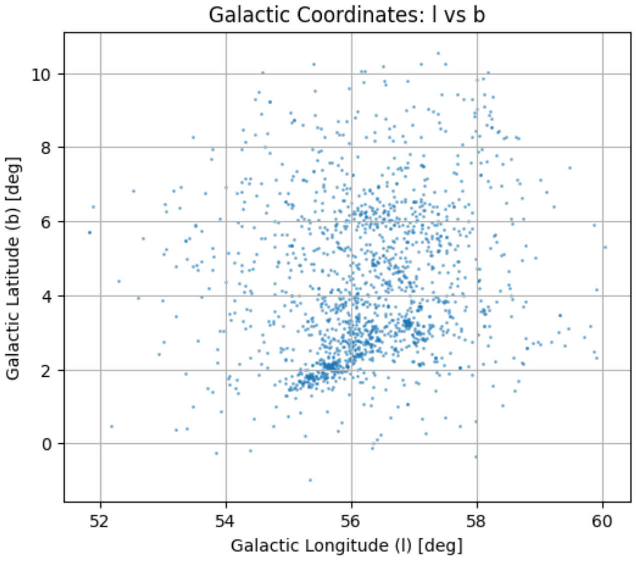

Figure 7: Galactic sky coordinates (ℓ, b)

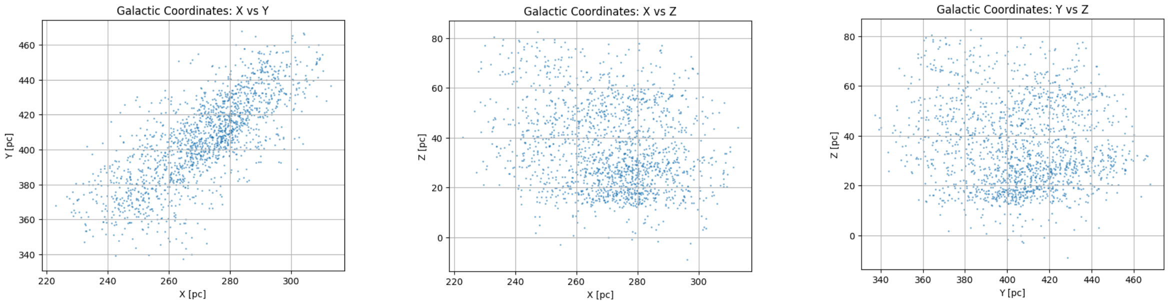

Figure 8: Cartesian projections in Galactic coordinates. Left: X vs Y. Center: X vs Z. Right: Y vs Z.

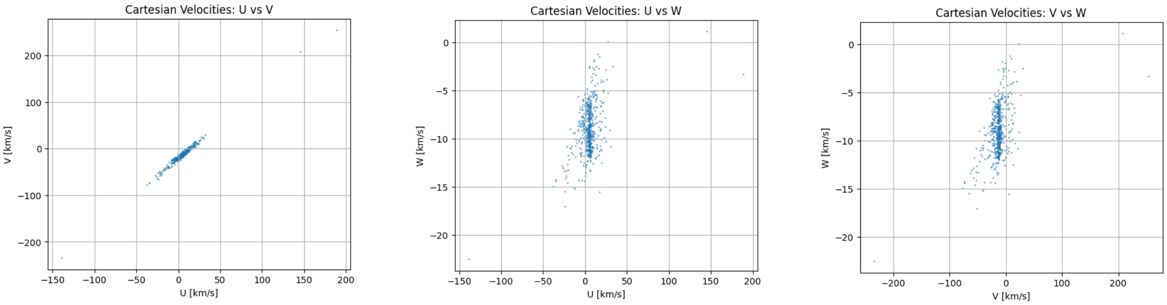

Figure 9: Cartesian Velocities. Left: U vs V. Center: U vs W. Right: V vs W.

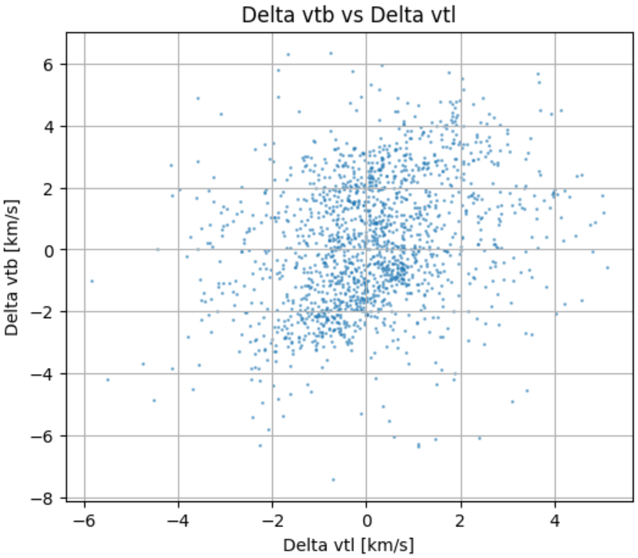

Figure 10: Transverse velocity anomalies. Each point represents the deviation of a star's transverse velocity from the projected mean UVW vector, plotted in the $(\Delta v_{t\ell}, \Delta v_{tb})$ plane.

9. Color-Magnitude Diagram (CMD)

To investigate the photometric properties of the SCYA-89 association, we construct a color-magnitude diagram (CMD) using Gaia DR3 photometry. We compute the absolute G-band magnitude M_G from the apparent G magnitude and distance using the standard distance modulus:

$$M_G = G - 5 \cdot \log_{10}\left(\frac{d}{10}\right) \tag{5}$$

where G is the apparent magnitude and d is the distance in parsecs. The color is computed as $G_{\mathrm{BP}} - G_{\mathrm{RP}}$.

We plot M_G vs $G_{\mathrm{BP}} - G_{\mathrm{RP}}$, inverting the y-axis so it is oriented towards increasing brightness.

The resulting CMD, in Figure 11, shows a distinct population of young, luminous stars consistent with pre-main sequence evolution. This diagram also forms the foundation for the isochrone-fitting routine in the bonus age estimation section.

As shown in Equation 5, absolute magnitude is derived directly from distance and apparent magnitude assuming negligible extinction.

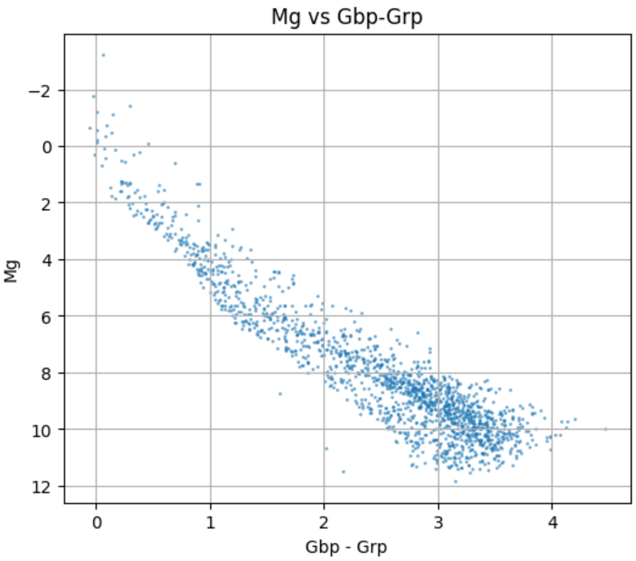

Figure 11: Color-magnitude diagram of SCYA-89 showing M_G vs $G_{\mathrm{BP}} - G_{\mathrm{RP}}$ for filtered members.

10. HDBSCAN Clustering

We install and run the HDBSCAN clustering algorithm on SCYA-89 using a 5D array composed of combinations such as (X, Y, Z, $v_{t\ell}$, v_{tb}) and others suggested by SPYGLASS-VI. We scale the velocity dimensions to match the spatial scales.

11. Coordinate Experiments

To evaluate different clustering strategies, we apply the HDBSCAN algorithm to SCYA-89 using four different coordinate combinations. All methods use a velocity scaling factor of 8 to bring kinematic and spatial dimensions to comparable scales. The clustering is performed using `HDBSCAN` with `min_cluster_size = 7` and `min_samples = 7`, with the leaf cluster selection method.

Method A: $(X, Y, Z, v_{t\ell}, v_{tb})$ We cluster in 3D Cartesian position coordinates combined with scaled transverse velocities:

- X, Y, Z from galactic Cartesian transformation
- $v_{t\ell}$ and v_{tb} scaled by the velocity factor

This method resembles the original SPYGLASS-II approach and is sensitive to both position and velocity.

Method B: $(\ell, b, d, v_{t\ell}, v_{tb})$ **with projected angular coordinates** Here, angular coordinates (ℓ, b) are projected to Cartesian distances using the average distance to avoid uncertainty amplification:

- $\ell \cdot d_{\mathrm{avg}}$, $b \cdot d_{\mathrm{avg}}$, and d
- Transverse velocities $v_{t\ell}$, v_{tb} (scaled)

This approach balances position information with minimized distance-based spread.

Method C: $(\ell, b, d, \mu_\ell, \mu_b)$ We cluster using proper motions instead of transverse velocities to decouple the distance dependency:

- Angular position and distance: ℓ, b, d
- Proper motions: μ_ℓ, μ_b

This method reduces the impact of distance uncertainties on the velocity dimensions.

Method D: $(\ell, b, d, \Delta v_{t\ell}, \Delta v_{tb})$ We use transverse velocity anomalies computed with an average distance to define velocity structure relative to bulk motion. This captures deviations from the mean projected motion, enhancing contrast between dynamically coherent groups.

12. Clustering Visualization

We visualize the clustering results in various coordinate projections as seen in Figure 12

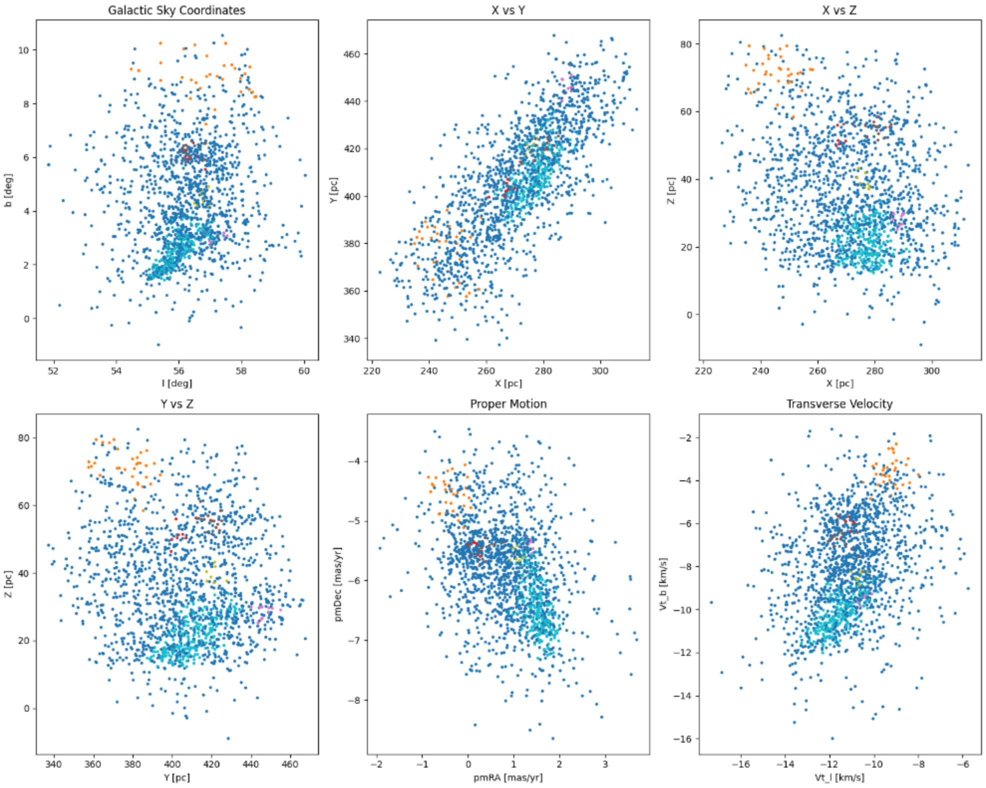

Figure 12: HDBSCAN clustering in XYZ space for SCYA-89. Colors indicate distinct clusters.