

# PowerBin: fast adaptive data binning with Centroidal Power Diagrams

Michele Cappellari  

*Sub-Department of Astrophysics, Department of Physics, University of Oxford, Denys Wilkinson Building, Keble Road, Oxford OX1 3RH, UK*

Accepted 2025 October 6. Received 2025 October 6; in original form 2025 September 8

## ABSTRACT

Adaptive binning is a crucial step in the analysis of large astronomical data sets, such as those from integral-field spectroscopy, to ensure a sufficient signal-to-noise ratio ( $S/\mathcal{N}$ ) for reliable model fitting. However, the widely used Voronoi-binning method and its variants suffer from two key limitations: they scale poorly with data size, often as  $\mathcal{O}(N^2)$ , creating a computational bottleneck for modern surveys, and they can produce undesirable non-convex or disconnected bins. I introduce POWERBIN, a new algorithm that overcomes these issues. I frame the binning problem within the theory of optimal transport, for which the solution is a Centroidal Power Diagram (CPD), guaranteeing convex bins. Instead of formal CPD solvers, which are unstable with real data, I develop a fast and robust heuristic based on a physical analogy of packed soap bubbles. This method reliably enforces capacity constraints even for non-additive measures like  $S/\mathcal{N}$  with correlated noise. I also present a new bin-accretion algorithm with  $\mathcal{O}(N \log N)$  complexity, removing the previous bottleneck. The combined POWERBIN algorithm scales as  $\mathcal{O}(N \log N)$ , making it about two orders of magnitude faster than previous methods on million-pixel data sets. I demonstrate its performance on a range of simulated and real data, showing it produces high-quality, convex tessellations with excellent  $S/\mathcal{N}$  uniformity. The public PYTHON implementation provides a fast, robust, and scalable tool for the analysis of modern astronomical data.

**Key words:** methods: data analysis – methods: numerical – techniques: image processing – techniques: imaging spectroscopy – software: data analysis – galaxies: kinematics and dynamics.

## 1 INTRODUCTION

Modern astronomical surveys, particularly those using integral-field spectroscopy (IFS), produce vast, spatially resolved data sets containing millions of spectra (e.g. M. Cappellari et al. 2011; S. F. Sánchez et al. 2012; J. J. Bryant et al. 2015; K. Bundy et al. 2015). A common task is to fit these data with complex physical models to extract quantities like stellar kinematics (e.g. K. B. Westfall et al. 2019), star formation histories, or chemical abundances (e.g. R. M. McDermid et al. 2015; N. Scott et al. 2017; S. Lu et al. 2023). However, the signal-to-noise ratio ( $S/\mathcal{N}$ ) of individual spatial pixels (spaxels) is often too low for reliable model fitting (M. Cappellari & Y. Copin 2003). Fitting a non-linear model to low- $S/\mathcal{N}$  data typically yields a highly non-Gaussian posterior probability distribution for the model parameters, making it difficult to derive meaningful best-fitting values and uncertainties from Bayesian methods (e.g. A. Gelman et al. 2014). Crucially, simply averaging the biased results from low- $S/\mathcal{N}$  fits does not recover the true parameters, because the posterior does not need to be symmetric around the true parameters.

The most effective solution is to combine the data from adjacent spaxels into larger bins to reach a sufficient  $S/\mathcal{N}$  before performing the model fit. This process, known as adaptive binning, is a crucial pre-processing step in the analysis of IFS data and other 2D data sets like X-ray images (e.g. J. S. Sanders et al. 2004; S. Diehl & T. S. Statler 2007).

### 1.1 The Voronoi-binning method

To address this need, M. Cappellari & Y. Copin (2003) introduced the Voronoi-binning algorithm, implemented for example in the VORBIN package.<sup>1</sup> This method has become a standard tool in astrophysics for partitioning 2D data to meet three key criteria for an optimal binning scheme: (i) the bins must form a complete, non-overlapping tessellation of the data; (ii) the bins should be as compact (round) as possible to preserve spatial resolution; and (iii) a user-defined scalar function of each bin should be as uniform as possible around a target value.

This optimized function is entirely general: while it is often chosen to be the bin's  $S/\mathcal{N}$ , the algorithm places no restriction on its form. It may represent, for example, the fractional error in a physical parameter derived from a spectral fit to the bin's data, or a composite metric such as a weighted combination of  $S/\mathcal{N}$  values measured in different photometric bands over the same bin area.

The algorithm achieved this through a two-stage process (M. Cappellari & Y. Copin 2003). First, a *bin-accretion stage* provides an initial tessellation satisfying the target  $S/\mathcal{N}$ . This greedy algorithm starts with the unbinned pixel with the highest  $S/\mathcal{N}$ , accretes its nearest neighbours until the target  $S/\mathcal{N}$  is reached, and repeats this process until all pixels are binned. Second, an iterative *regularization stage* improves the bin morphology using a Centroidal Voronoi Tessellation (CVT; Q. Du, V. Faber & M. Gunzburger 1999). In each iteration, a Voronoi tessellation is generated from the current

\* E-mail: [michele.cappellari@physics.ox.ac.uk](mailto:michele.cappellari@physics.ox.ac.uk)

<sup>1</sup>Python version 3.1 from <https://pypi.org/project/vorbin/>.

bin generators, and these generators are then updated to be the new centroids of their corresponding cells, making the bins more compact.

## 1.2 Limitations of existing methods

Despite its widespread success, the evolution of astronomical surveys has revealed some limitations of the original method and its subsequent extensions.:

(i) **Non-Convexity:** An important extension by S. Diehl & T. S. Statler (2006), which is included in VORBIN, introduced a *multiplicatively weighted* Voronoi tessellation (WVT) to improve  $S/N$  uniformity and bin shapes. While effective, the adopted tessellation sacrifices a key morphological property: the guarantee of convex bins of the original CVT method. Although not a common occurrence, this can result in undesirable non-convex or even disconnected bins, which can be problematic in certain situations.

(ii) **Computational Speed:** The original implementation was not designed for the massive and rapidly growing data sets of modern and future astronomical surveys. Instruments like MUSE (R. Bacon et al. 2010), employed in large surveys (e.g. M. Sarzi et al. 2018; D. A. Gadotti et al. 2019; E. Emsellem et al. 2022; A. Fraser-McKelvie et al. 2025), can already produce mosaics of millions of pixels – for example, the 3 and 9 million spaxel mosaics of M42 and NGC 253 by P. M. Weilbacher et al. (2015) and E. Congiu et al. (2025) respectively – and this trend is set to accelerate. Next-generation concepts like the Wide-field Spectroscopic Telescope (WST) (R. M. Bacon et al. 2024) are projected to produce data sets orders of magnitude larger still. Both the bin-accretion and the iterative regularization stages of VORBIN scale poorly with the number of input spaxels, creating a significant computational bottleneck. In particular, the multiplicatively weighted Voronoi (MWV) diagram has a fundamental time complexity of  $\mathcal{O}(n^2)$ , where  $n$  is the number of bins (F. Aurenhammer & H. Edelsbrunner 1984), making it impractical for large  $n$ , and the same is true for the bin-accretion algorithm.

The goal of this paper is to introduce POWERBIN, a fast and robust algorithm that addresses the limitations of previous adaptive-binning methods. I recast adaptive binning as a semidiscrete optimal-transport, or data-quantization, problem whose solutions are Centroidal Power Diagrams (CPDs; F. Aurenhammer 1987; F. Aurenhammer, F. Hoffmann & B. Aronov 1998; Q. Mérigot 2011; F. De Goes et al. 2012; B. Lévy 2015). Building on a simple geometric/physical insight, POWERBIN iteratively adjusts the power-diagram weights to enforce per-bin capacity targets while keeping cells convex and compact. The resulting scheme is computationally efficient, stable in the presence of realistic, non-additive capacity measures (for example, when pixel noise is correlated), and scales to the large data sets produced by modern astronomical surveys. A public PYTHON reference implementation accompanies this paper.

This paper is structured as follows. Section 2 provides practical examples illustrating the necessity of binning. Section 3 reviews the family of weighted Voronoi diagrams. Section 4 introduces the optimal transport framework and its connection to CPDs. Section 5 presents the core physical insight behind my new fast regularization algorithm. Section 6 describes the new fast bin-accretion algorithm. Section 7 demonstrates the performance of the new method on real and simulated data. Section 8 presents execution-time benchmarks. Finally, Section 9 summarizes my findings.

## 2 EXAMPLES ILLUSTRATING THE NEED FOR BINNING

While Section 1 described the general motivation for binning, the importance of this pre-processing step is best understood through practical examples. The core issue is that fitting complex, non-linear models to low- $S/N$  data can lead to results that are not just uncertain, but systematically biased. This section presents two common scenarios in spectral analysis that demonstrate this effect and motivate the need for an optimal binning strategy. The behaviour shown is generic and applies to many types of data analysis in other fields. An ideal binning scheme should partition the data according to three criteria (M. Cappellari & Y. Copin 2003): (1) a topological criterion, ensuring all data are used without overlap; (2) a morphological criterion, requiring bins to be as compact (round) as possible to preserve spatial resolution; and (3) a uniformity criterion, minimizing the  $S/N$  scatter around a target value. The following examples highlight why achieving a target  $S/N$  is paramount.

### 2.1 Extracting stellar kinematics of multiple components

This example examines the problem of separating the kinematics of multiple stellar populations along a single line of sight. The goal is to illustrate a generic characteristic of non-linear problems in general, and astrophysical spectrum fitting in particular: at low  $S/N$  the model posterior is not informative of the true solution. For this reason, the choice of spectra and kinematics is rather arbitrary and is not meant to represent a specific situation. I construct a synthetic spectrum made from two distinct stellar components and attempt to recover their velocities and dispersions at two representative per-pixel signal-to-noise levels, showing that low  $S/N$  yields multimodal, biased parameters while higher  $S/N$  permits reliable parameter recovery.

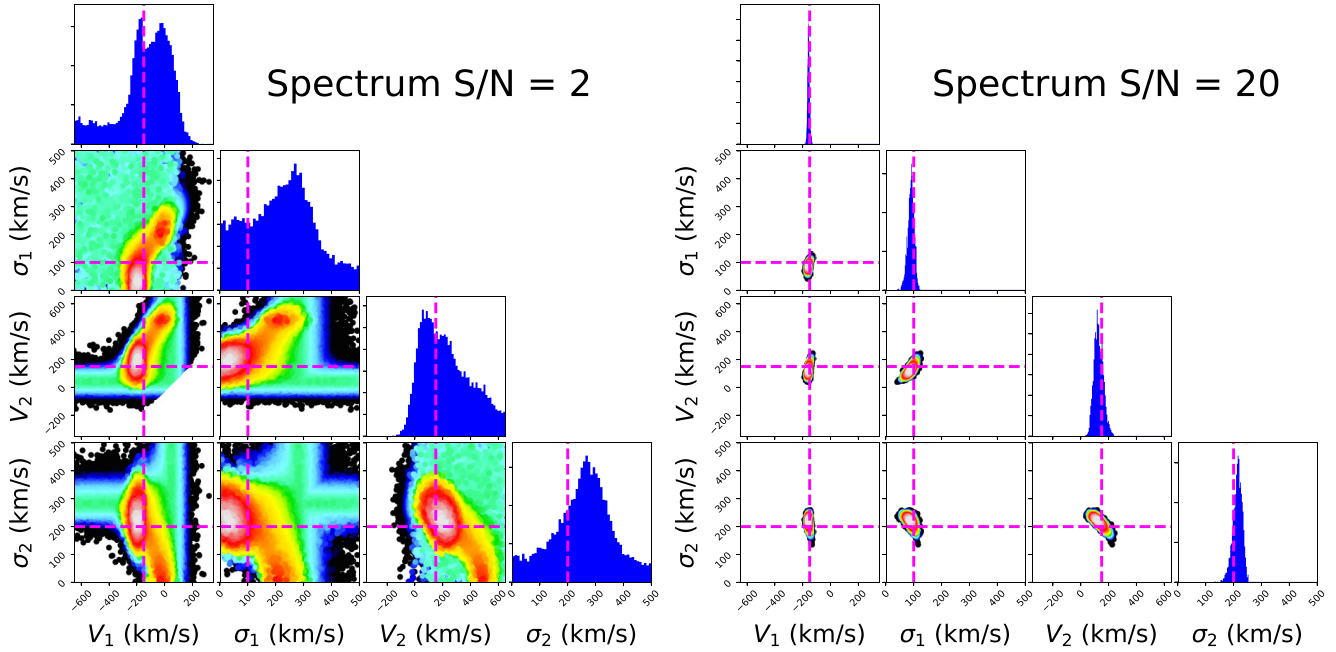
For this test, I took two model spectra from the MILES library (A. Vazdekis et al. 2010), both with solar metallicity but with different ages (12.6 and 1.0 Gyr). The spectra were logarithmically rebinned to a velocity scale of  $70 \text{ km s}^{-1}$ , typical of large surveys like SDSS (e.g. Abdurro'uf et al. 2022), and normalized to contribute equally to the total flux in the fitted region (354–741 nm). I assigned distinct kinematics to each component:  $(V_1, \sigma_1) = (-150, 100) \text{ km s}^{-1}$  for the old population and  $(V_2, \sigma_2) = (150, 200) \text{ km s}^{-1}$  for the young one.

I then used the Penalized PiXel Fitting (PPXF) software<sup>2</sup> (M. Cappellari & E. Emsellem 2004; M. Cappellari 2017, 2023) to recover the four kinematic parameters  $(V_1, \sigma_1, V_2, \sigma_2)$ . To explore the parameter space, I coupled PPXF with the Adaptive Metropolis MCMC sampler by (H. Haario, E. Saksman & J. Tamminen 2001) as implemented in the ADAMET package<sup>3</sup> (M. Cappellari et al. 2013). Assuming a uniform prior, the posterior probability is  $P \propto \exp(-\chi^2/2)$ . I ran a chain of  $10^5$  steps for two cases: a low- $S/N$  case ( $S/N = 2$  per pixel) and a high- $S/N$  case ( $S/N = 20$ ).

The results are shown in Fig. 1. At  $S/N = 2$  (left panel), the posterior distribution is highly complex, non-Gaussian, and biased away from the true input values (magenta lines). The posterior is hard to interpret and one cannot meaningfully quote a single best-fitting value or its error. Averaging such biased results from many low- $S/N$  spaxels would not recover the true average kinematics. In contrast, at  $S/N = 20$  (right panel), the posterior is unimodal, symmetric, and correctly centred on the true values. The right panel unambiguously

<sup>2</sup>Python version 9.4 from <https://pypi.org/project/ppxf/>.

<sup>3</sup>Python version 2.0 from <https://pypi.org/project/adamet/>.



**Figure 1.** Posterior probability distributions for the four kinematic parameters ( $V_1, \sigma_1, V_2, \sigma_2$ ) of two stellar components, recovered with PPXF coupled to an Adaptive Metropolis MCMC sampler ( $10^5$  steps). Left: spectrum with  $S/N = 2$  per pixel. The joint and marginal posteriors are highly non-Gaussian, multimodal and biased away from the true input values (magenta dashed lines), illustrating that parameter estimates at low  $S/N$  are unreliable. Right: spectrum with  $S/N = 20$  per pixel. The posteriors become well behaved, approximately Gaussian and centred on the true values, allowing robust inference. Contours mark the 68 and 95 per cent credible regions; diagonal panels show marginal histograms. This comparison demonstrates why one must bin spaxels to reach sufficient  $S/N$  before fitting non-linear spectral models.

reveals the two kinematic components and the parameters are well-constrained. This demonstrates that reaching a sufficient  $S/N$  by binning is essential before attempting such a measurement.

## 2.2 Star formation history from full-spectrum fitting

My second example concerns the recovery of a galaxy’s star formation history (SFH), a problem that involves fitting a spectrum with a large linear combination of template spectra representing stellar populations of different ages and metallicities.

Here, I constructed a synthetic galaxy spectrum with a simple SFH, dominated by a single burst of star formation 0.3 Gyr ago. The mock spectrum was built from a linear combination of 25 solar-metallicity templates from the MILES models (A. Vazdekis et al. 2010), with ages spaced logarithmically between 0.063 and 15.8 Gyr. I then attempted to recover the SFH using PPXF at three different  $S/N$  levels: 5, 50, and 500. For each  $S/N$  level, I ran 1000 Monte Carlo realizations, adding appropriate Gaussian noise to the spectrum in each run and fitting for the template weights. The SFH recovery was performed with a low level of regularization ( $\text{‘regul}=10$ ’; see M. Cappellari 2023, fig. 5, for visual reference). This example is not intended to illustrate the expected performance of PPXF in a realistic setting; for that, the reader is referred to J. Woo et al. (2024). Rather, its purpose is to demonstrate a generic feature of non-linear models, particularly those used in astrophysical problems where binning is common: at low  $S/N$ , the model posteriors can be biased and unrepresentative of the true solution.

Fig. 2 summarizes the Monte Carlo SFH recoveries. At  $S/N = 5$  the solutions are dominated by noise and a clear systematic bias: the fitter spuriously assigns weight to old populations and fails to recover the true single-burst history. At  $S/N = 50$  the correct peak is recovered but with substantial scatter, while only at  $S/N = 500$  does the

ensemble reliably reproduce the input SFH with high fidelity. These tests demonstrate that reaching a problem-dependent minimum  $S/N$  is essential for trustworthy spectral inference; adaptive binning is therefore a necessary pre-processing step, not merely a cosmetic choice.

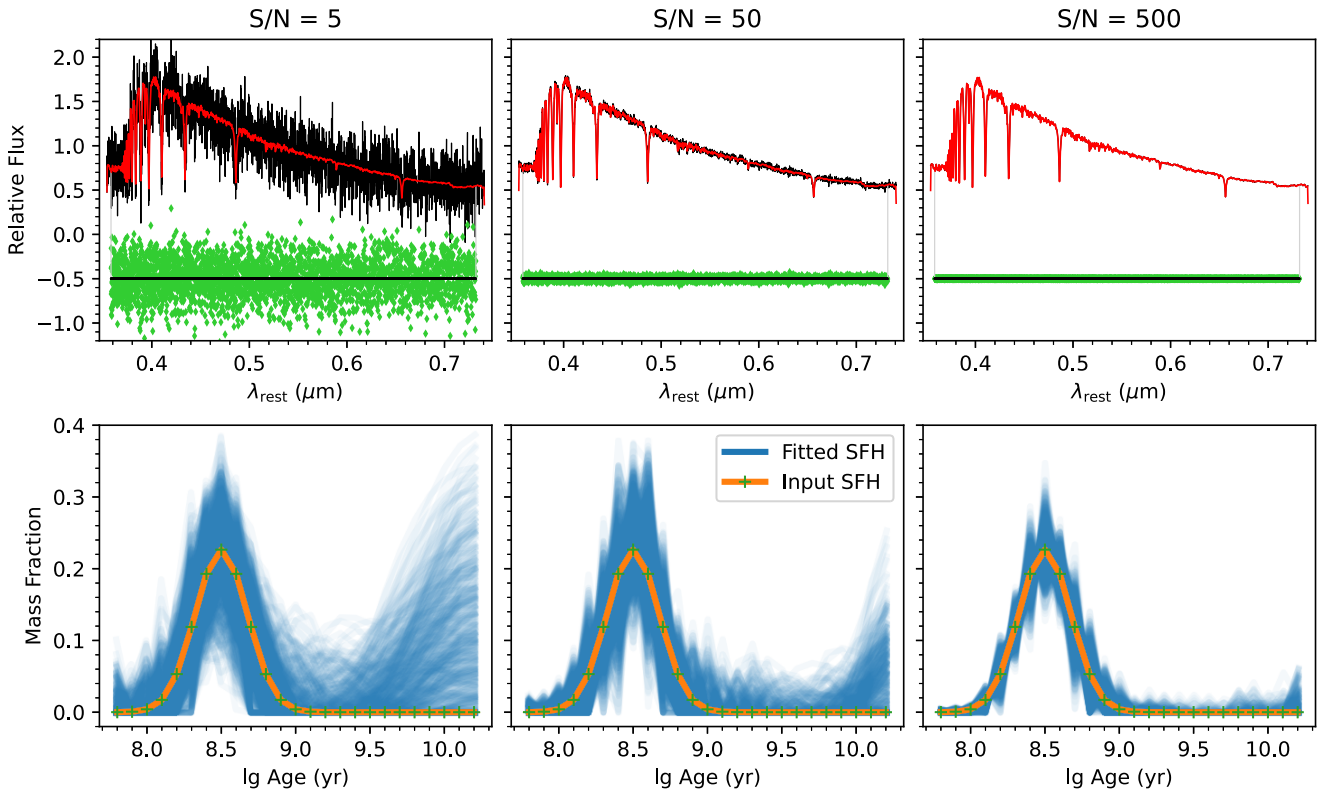
## 3 GENERALIZATIONS OF VORONOI DIAGRAMS

The original Voronoi-binning algorithm (M. Cappellari & Y. Copin 2003) used the ordinary Voronoi tessellation, the simplest member of a broader family of weighted Voronoi diagrams. Later, S. Diehl & T. S. Statler (2006) extended the regularization phase by adopting the multiplicatively weighted variant. In order to motivate the adoption of a different tessellation for adaptive binning, it is useful to place these and other variants side by side. This section therefore reviews the principal types of weighted Voronoi diagrams, following the classic treatments of A. Okabe et al. (2000, section 3) and F. Aurenhammer, R. Klein & D.-T. Lee (2013), and uses Fig. 3 to give a geometric interpretation of each. This comparison will make clear that one particular member of the family occupies a special position for our purposes.

### 3.1 Ordinary Voronoi diagram

Given a set of  $n$  distinct points  $\mathbf{G} = \{\mathbf{g}_1, \dots, \mathbf{g}_n\}$  in a  $d$ -dimensional Euclidean space  $\mathbb{R}^d$ , called generators, the ordinary Voronoi diagram is a partition of the space into regions based on the nearest-neighbour rule. The Voronoi cell  $\mathcal{V}(\mathbf{g}_j)$  associated with a generator  $\mathbf{g}_j$  contains all points in  $\mathbb{R}^d$  that are closer to  $\mathbf{g}_j$  than to any other generator  $\mathbf{g}_k$ . Using the Euclidean distance

$$d(\mathbf{x}, \mathbf{g}_j) = \|\mathbf{x} - \mathbf{g}_j\| \quad (1)$$



**Figure 2.** Recovery of a simple input star-formation history from full-spectrum fitting at three signal-to-noise levels. Top row: mock spectrum (black), PPXF best-fitting model (red) and residuals (green) for  $S/N = 5, 50,$  and  $500$  (left to right). Bottom row: the distribution of 1000 Monte Carlo recovered SFHs (transparent blue lines) compared to the input single burst at 0.3 Gyr (orange line with markers). At low  $S/N$  the recovered SFHs are biased and highly scattered; increasing  $S/N$  progressively reduces bias and scatter, and by  $S/N = 500$  the input burst is recovered with high fidelity. The fits employ 25 solar-metallicity MILES templates and 1000 Monte Carlo realizations per  $S/N$  level.

the cell is defined as

$$\mathcal{V}(\mathbf{g}_j) = \{\mathbf{x} \mid \|\mathbf{x} - \mathbf{g}_j\| \leq \|\mathbf{x} - \mathbf{g}_k\|, \forall k \neq j\}. \quad (2)$$

The boundaries of the Voronoi cells are formed by segments of perpendicular bisectors between pairs of generators. Consequently, the cells are always convex polytopes. Efficient algorithms exist for computing ordinary Voronoi diagrams, with a worst-case time complexity of  $\mathcal{O}(n \log n)$  in 2D (e.g. chapter 3 of F. Aurenhammer et al. 2013).

Geometrically, the ordinary Voronoi diagram can be visualized as the projection onto the 2D plane of the intersections of a set of identical 3D cones, whose apices are located at the generator positions (Fig. 3a). The diagram can be seen by looking at the cones from above.

### 3.2 Multiplicatively weighted Voronoi diagram

The MWV diagram assigns a weight  $w_j > 0$  to each generator  $\mathbf{g}_j$ , with the weight acting as a scaling factor on the distance. This allows one to change the influence of each generator on the partitioning of space. The weighted distance is

$$d_{\text{MW}}(\mathbf{x}, \mathbf{g}_j) = \|\mathbf{x} - \mathbf{g}_j\|/w_j. \quad (3)$$

The cell is defined as

$$\mathcal{V}_{\text{MW}}(\mathbf{g}_j) = \{\mathbf{x} \mid \|\mathbf{x} - \mathbf{g}_j\|/w_j \leq \|\mathbf{x} - \mathbf{g}_k\|/w_k, \forall k \neq j\}. \quad (4)$$

The boundary between two cells,  $\|\mathbf{x} - \mathbf{g}_j\|/\|\mathbf{x} - \mathbf{g}_k\| = w_j/w_k$ , is a hypersphere (known as a Circle of Apollonius in 2D). A key

characteristic of MWV diagrams is that their cells are not necessarily convex and in general can be disconnected. Fig. 3(b) shows an example of a MWV diagram with a cell contained inside another. Crucially, the optimal computation time for its generation scales as  $\mathcal{O}(n^2)$  (F. Aurenhammer & H. Edelsbrunner 1984; F. Aurenhammer et al. 2013, section 7.4.2).

The geometric interpretation of the boundaries of a MWV diagram is the projection of the intersections of cones with different inclinations (slopes), determined by their weights  $w_j$  (Fig. 3b).

### 3.3 Additively Weighted Voronoi diagram

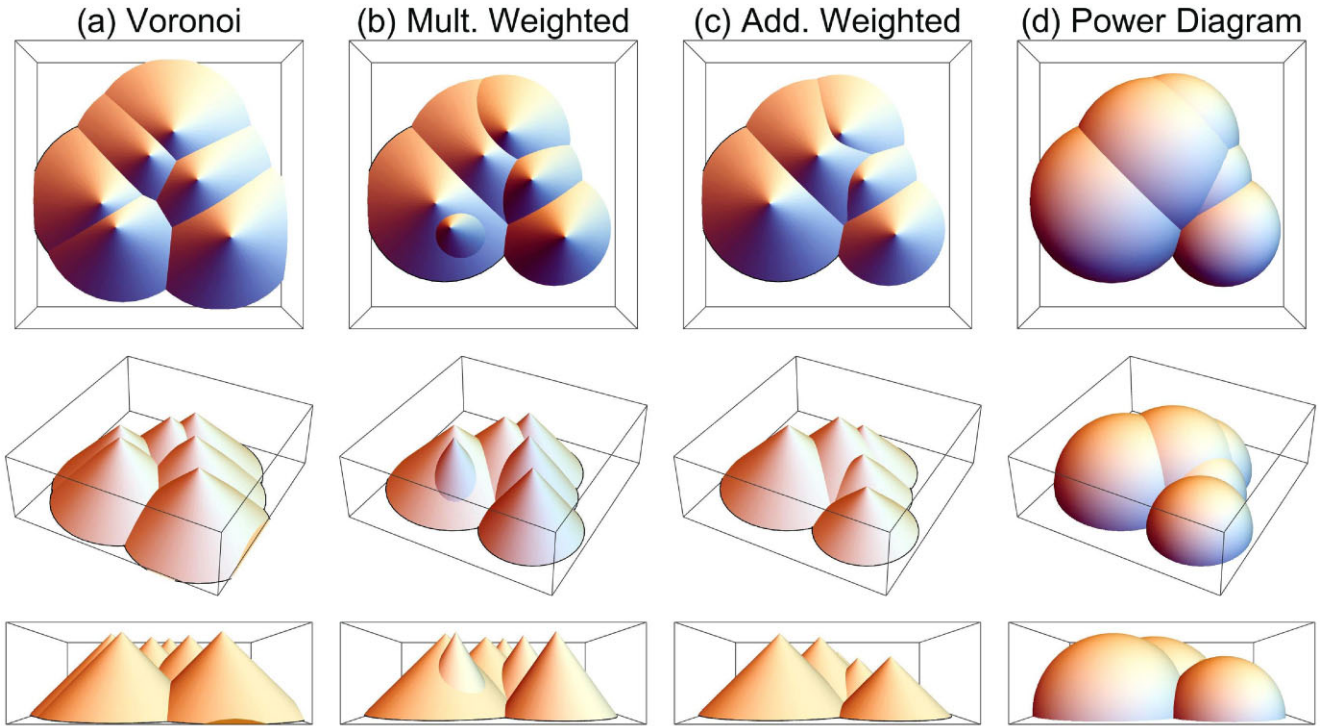
The additively weighted Voronoi (AWV) diagram also assigns a real-valued weight  $w_j$  to each generator  $\mathbf{g}_j$ . The distance metric is modified by subtracting this weight, defining a weighted distance

$$d_{\text{AW}}(\mathbf{x}, \mathbf{g}_j) = \|\mathbf{x} - \mathbf{g}_j\| - w_j. \quad (5)$$

The corresponding cell is

$$\mathcal{V}_{\text{AW}}(\mathbf{g}_j) = \{\mathbf{x} \mid \|\mathbf{x} - \mathbf{g}_j\| - w_j \leq \|\mathbf{x} - \mathbf{g}_k\| - w_k, \forall k \neq j\}. \quad (6)$$

The boundary between two cells, defined by  $\|\mathbf{x} - \mathbf{g}_j\| - \|\mathbf{x} - \mathbf{g}_k\| = w_j - w_k$ , is a sheet of a hyperboloid of revolution. In 2D, the boundaries are hyperbolic arcs. Like for MWV diagrams, the cells of AWV diagrams are not guaranteed to be convex. The computational complexity is like for the ordinary Voronoi diagrams, scaling as  $\mathcal{O}(n \log n)$  (S. Fortune 1986; F. Aurenhammer et al. 2013, section 7.4.1).



**Figure 3.** Geometric interpretation of several Voronoi generalizations. Each column illustrates a different tessellation. The top row shows the view from above: the boundaries of the 2D tessellation are the vertical projections on to the  $(x, y)$ -plane of the pairwise intersection curves of the 3D surfaces (cones or spheres). The middle (inclined) and bottom (side-on) rows show the underlying 3D surfaces. In all cases, the surfaces are centred at the same  $(x, y)$  locations (the generators), while their shapes are modified by weights. (a) *Ordinary Voronoi diagram*: Boundaries come from intersections of identical right circular cones (same slope and same apex height). The projected bisectors are straight lines, yielding convex cells. (b) *MWV (Apollonius) diagram*: Boundaries come from intersections of cones with different slopes but the same apex height. The weight  $w_j$  controls the cone slope, so at any fixed height the base radius scales with  $w_j$ . The projected bisectors are circular arcs (Apollonius circles), and cells may be non-convex. The figure shows an example where one small cell is contained entirely within a larger one. (c) *Additively weighted Voronoi (Johnson–Mehl) diagram*: Boundaries come from intersections of cones with the same slope but different apex heights. The weight  $w_j$  is encoded in the apex height, and  $w_j$  still represents the radius of the cone base. The projected bisectors are hyperbolic arcs, and cells are not guaranteed to be convex. In this example, one cone lies entirely below another, so its corresponding cell is empty and the tessellation has one fewer region than in (a) or (b). (d) *Power (Laguerre) diagram*: Boundaries come from intersections of spheres centred at  $(\mathbf{g}_j, 0)$  with radii  $r_j = \sqrt{w_j}$ . The projected bisectors are straight lines (radical axes), so cells are convex. As in (c), one sphere lies entirely below another, so its corresponding cell is empty and the tessellation has one fewer region.

This diagram can be visualized as the projection of the intersections of cones with identical slopes but different heights, where the apex of each cone is shifted vertically by its weight  $w_j$  (Fig. 3c).

### 3.4 Power diagram: the ideal candidate

Among the family of weighted Voronoi diagrams, the power diagram (also known as the Laguerre–Voronoi diagram) is not just another variation; it possesses a unique combination of geometric and computational properties that make it the ideal mathematical structure for the adaptive binning problem. It is defined using the ‘power distance’, where each generator  $\mathbf{g}_j$  is associated with a real-valued weight  $w_j$ . The power of a point  $\mathbf{x}$  with respect to a generator  $\mathbf{g}_j$  is given by

$$\text{pow}(\mathbf{x}, \mathbf{g}_j) = \|\mathbf{x} - \mathbf{g}_j\|^2 - w_j. \quad (7)$$

The power cell  $\mathcal{V}_{\text{pow}}(\mathbf{g}_j)$  consists of all points whose power with respect to  $\mathbf{g}_j$  is less than or equal to their power with respect to any other generator:

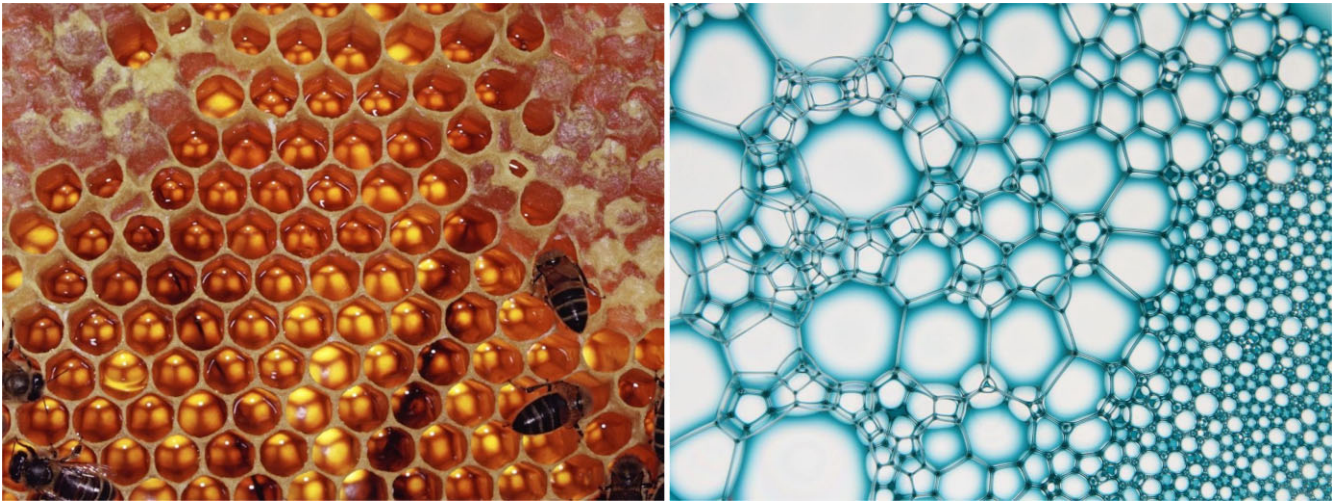
$$\mathcal{V}_{\text{pow}}(\mathbf{g}_j) = \{\mathbf{x} \mid \|\mathbf{x} - \mathbf{g}_j\|^2 - w_j \leq \|\mathbf{x} - \mathbf{g}_k\|^2 - w_k, \forall k \neq j\}. \quad (8)$$

The boundary condition,  $\|\mathbf{x} - \mathbf{g}_j\|^2 - w_j = \|\mathbf{x} - \mathbf{g}_k\|^2 - w_k$ , simplifies to a linear equation. This means the boundaries are hyper-

planes (straight lines in 2D), which in turn guarantees that the cells are always convex polytopes—a critical property that both AWV and MWV diagrams lack.

Furthermore, this linear boundary property allows power diagrams to be computed with optimal efficiency. By transforming the problem into a convex hull construction in one higher dimension, the tessellation can be found in  $\mathcal{O}(n \log n)$  time in 2D (H. Imai, M. Iri & K. Murota 1985; F. Aurenhammer 1987), matching the speed of the simplest ordinary Voronoi diagram.

Geometrically, the boundaries of a power diagram are the vertical projection of the intersections of a set of upward-opening paraboloids  $z = \|\mathbf{x} - \mathbf{g}_j\|^2 - w_j$ . Equivalently, the boundaries of a 2D power diagram can be viewed as the vertical projection of the intersection of a set of 3D spheres with different radii, whose centres lie on the  $z = 0$  2D plane (Fig. 3d). To see this, consider a sphere  $j$  centred at  $(\mathbf{g}_j, 0)$  with radius  $r_j$ . Its equation is  $\|\mathbf{x} - \mathbf{g}_j\|^2 + z^2 = r_j^2$ . The intersection of two spheres  $j$  and  $k$  lies on a plane (the radical plane) defined by equating their equations, which yields  $\|\mathbf{x} - \mathbf{g}_j\|^2 - r_j^2 = \|\mathbf{x} - \mathbf{g}_k\|^2 - r_k^2$ . This is precisely the boundary condition for a power diagram with weights  $w_j = r_j^2$ . This sphere-based interpretation provides a direct link to the physical analogy of packed soap bubbles (Fig. 4), which inspires the fast heuristic algorithm presented in this paper. As I will show in Section 4, this



**Figure 4.** Natural analogues for optimal data binning, illustrating the principles of compactness and capacity uniformity. *Left:* A honeycomb demonstrates the optimal packing of equal-area cells (hexagons). *Right:* A 2D foam (soap bubbles) provides a physical model for a capacity-constrained tessellation. By minimizing surface energy, the bubbles form a structure equivalent to a power diagram, where different bubble sizes correspond to different cell capacities. This illustrates the physical principle behind the POWERBIN algorithm. Image courtesy of Professor Simon Cox, Aberystwyth University.

specific geometric form makes the power diagram the natural solution to the problem of optimal transport, providing a rigorous mathematical foundation for capacity-constrained binning. This unique combination of guaranteed convexity, computational efficiency, and a direct link to optimal transport theory sets the power diagram apart as the ideal choice for our application.

#### 4 CENTROIDAL POWER DIAGRAMS FOR OPTIMAL-TRANSPORT BINNING

In this section, I show that among the various generalizations of Voronoi diagrams, power diagrams are uniquely suited for the adaptive binning of empirical data. This is because they provide a natural solution to a class of problems known as optimal transport, which offers a rigorous mathematical foundation for the binning criteria I outlined in Section 1.

##### 4.1 The optimal transport problem

The theory of optimal transport, first formulated by G. Monge (1781), provides a mathematical framework for finding the most efficient way to remap one distribution of ‘mass’ (or any density) to another, given a specified transport cost. For comprehensive reviews of the theory and its computational methods, see B. Lévy & E. L. Schwindt (2018) and G. Peyré & M. Cuturi (2019).

For the adaptive binning problem, I consider the ‘semi-discrete’ case: transporting a continuous density distribution  $\rho(\mathbf{x})$ , which is approximated by our  $N$  pixels of data, to a discrete set of  $n$  target locations, the bin generators  $\{\mathbf{g}_j\}_{j=1}^n$ . I define the transport cost as the squared Euclidean distance,  $\|\mathbf{x} - \mathbf{g}_j\|^2$ . The goal is to find a partition of the data into a set of  $n$  bins  $\{\mathcal{V}_j\}$  that minimizes the total transport cost,

$$\mathcal{E}(\{\mathbf{g}_j\}, \{\mathcal{V}_j\}) = \sum_{j=1}^n \int_{\mathcal{V}_j} \|\mathbf{x} - \mathbf{g}_j\|^2 \rho(\mathbf{x}) \, d\mathbf{x}, \quad (9)$$

while ensuring each bin  $j$  contains a prescribed amount of mass, or ‘capacity’,  $v_j$ . That is, the partition must satisfy the constraint

$$m_j = \int_{\mathcal{V}_j} \rho(\mathbf{x}) \, d\mathbf{x} = v_j, \quad \forall j. \quad (10)$$

It can be shown that the optimal partition  $\{\mathcal{V}_j\}$  for this problem is a power diagram (e.g. F. Aurenhammer et al. 1998).

##### 4.2 Energy functional and Centroidal Power Diagrams

While the primal energy  $\mathcal{E}$  in equation (9) is intuitive, finding the partition  $\{\mathcal{V}_j\}_{j=1}^n$  that minimizes it under capacity constraints is difficult. The problem becomes tractable by considering the dual problem, which can be expressed using a Lagrangian functional  $\mathcal{F}$  that depends on the generator positions  $\{\mathbf{g}_j\}$  and a set of real-valued weights  $\{w_j\}$  acting as Lagrange multipliers (F. Aurenhammer et al. 1998; Q. Mérigot 2011; F. De Goes et al. 2012; B. Lévy 2015):

$$\mathcal{F}(\{\mathbf{g}_j\}, \{w_j\}) = \mathcal{E}(\{\mathbf{g}_j\}, \{\mathcal{V}_j\}) - \sum_{j=1}^n w_j(m_j - v_j). \quad (11)$$

Here,  $\mathcal{E}$  is the primal energy from equation (9), where the partition  $\{\mathcal{V}_j\}$  is now explicitly shown to be the power diagram defined by the weights  $\{w_j\}$ . The term  $m_j$  is the current capacity of cell  $\mathcal{V}_j$  from equation (10), and  $v_j$  is the target capacity, which for the Voronoi-binning problem I generally assume to be constant  $v$ , although this is not a requirement for the method.

The key insight is that finding the optimal binning is equivalent to finding a saddle point of  $\mathcal{F}$ . The gradients of  $\mathcal{F}$  with respect to the weights and generator positions reveal its utility:

(i) **Gradient w.r.t. weights:** The gradient with respect to a weight  $w_j$  is simply the difference between the cell’s target capacity and its current capacity (F. Aurenhammer et al. 1998; F. De Goes et al. 2012):

$$\nabla_{w_j} \mathcal{F} = v_j - m_j. \quad (12)$$

For a fixed set of generators, finding the weights  $\{w_j\}$  that maximize the dual functional  $\mathcal{F}$  is a convex optimization problem (F. Aurenhammer et al. 1998). This is a crucial property, as it guarantees the existence of a unique global maximum. Standard and efficient algorithms, such as Newton’s method, can be used to find this solution by driving the gradient to zero, thus ensuring the capacity constraints  $m_j = v_j$  are satisfied.

(ii) **Gradient w.r.t. generators:** The gradient with respect to a generator position  $\mathbf{g}_j$  is (F. De Goes et al. 2012):

$$\nabla_{\mathbf{g}_j} \mathcal{F} = 2m_j(\mathbf{g}_j - \mathbf{b}_j), \quad (13)$$

where  $\mathbf{b}_j$  is the barycentre (density-weighted centroid) of the cell  $\mathcal{V}_j$ . Setting this gradient to zero implies that the generator must coincide with its cell's barycentre:  $\mathbf{g}_j = \mathbf{b}_j$ .

A configuration that is a stationary point of  $\mathcal{F}$  – simultaneously satisfying the capacity constraints and the barycentric condition – is called a CPD. A CPD corresponds to a (local) minimum of the original transport energy  $\mathcal{E}$ , thus providing a complete and principled solution to the adaptive binning problem.

### 4.3 Challenges in applying to astronomical data

While the CPD framework is theoretically ideal, its direct application to the binning of astronomical data faces two main practical challenges:

(i) **Discrete Data versus Continuous Theory:** The optimal transport theory is formulated for continuous density functions. When applied to discrete data, integrals are replaced by sums over pixels. This approximation is valid when bins are large, but it breaks down when bins contain only a few pixels, leading to numerical instabilities. While one could interpolate the discrete data to create a continuous density, this is only rigorously applicable when the capacity function is additive.

(ii) **Non-Additive Capacity:** The most significant challenge is that the bin ‘capacity’ is often not a simple additive quantity. For example, when binning to a target signal-to-noise ratio ( $S/N$ ), the bin's total  $(S/N)^2$  is only the sum of the pixel  $(S/N)^2$  if the noise is uncorrelated (see M. Cappellari & Y. Copin 2003, section 2). In practice, instrumental effects and data reduction steps (e.g. dithering, resampling) introduce significant covariance between pixels (see K. B. Westfall et al. 2019, section 6.2). This makes the capacity  $m_j$  a non-linear, non-additive function of its constituent pixels. As a consequence, the dual functional  $\mathcal{F}$  loses its convenient convexity at fixed generator positions, and the analytic gradients become invalid. As a result, standard gradient-based optimization methods become unstable and fail to converge.

I confirmed this limitation through numerical experiments using the formalisms of F. Aurenhammer et al. (1998) and F. De Goes et al. (2012). While these variational approaches perform well for additive capacities in the continuum limit (i.e. large bins with many pixels), they fail catastrophically for the non-additive capacities typical of real data with correlated noise.

Because of these issues, a direct implementation of a mathematically exact CPD solver is not robust for real-world data binning. In the following section, I introduce a new algorithm, POWERBIN, which is inspired by the optimal transport framework but uses a fast and robust heuristic to handle these complexities.

## 5 FAST CENTROIDAL POWER-DIAGRAM SOLVER

The previous section established that while CPDs provide a theoretically ideal framework for adaptive binning, formal solvers based on gradient-based optimization of the dual energy functional are impractical for real astronomical data. The non-additive nature of capacity measures like  $S/N$  with correlated noise violates the assumptions required for these methods to converge reliably. This

section introduces the core of the POWERBIN algorithm: a fast, robust, and physically-motivated heuristic that bypasses these problems. Instead of relying on complex and fragile numerical optimization, I develop a simple iterative scheme inspired by the geometry of packed cells, which proves highly effective at enforcing capacity constraints while maintaining computational efficiency and bin convexity.

### 5.1 A physical heuristic for the weight-capacity relation

The central challenge in constructing a capacity-constrained power diagram is to find the set of weights  $\{w_j\}$  that yields cells with the desired capacities  $\{v_j\}$ . As noted by experts in the field, ‘the relation between the weights and the measures of the power cells is non-trivial’ (B. Lévy 2015, section 2.4). This complexity arises from two main factors. First, the weights are defined only up to a common additive constant; adding the same value to all weights leaves the tessellation unchanged. Second, for an arbitrary arrangement of generators, there is no simple, direct relationship between a weight  $w_j$  and the area  $A_j$  of its corresponding cell. The cells can be highly elongated, and some generators may even have empty cells.

However, the problem simplifies dramatically if one considers the specific geometry of a *centroidal* tessellation, where each generator is close to the centre of its cell. My approach is based on two key insights:

(i) **Power weights as squared radii.** As shown in Section 3, a power diagram can be defined by associating a circle of radius  $r_j$  with each generator  $\mathbf{g}_j$ , such that the weight is  $w_j = r_j^2$ . This provides a direct geometric interpretation: the cell boundaries are the radical axes of these circles, and their sizes control the tessellation.

(ii) **The ‘packed bubbles’ approximation.** In a centroidal configuration, the generators are close to the cell centroids, which naturally produces compact, nearly-round cells. In this limit, the tessellation resembles a foam of packed bubbles (Fig. 4). For such a packing, the area  $A_j$  of a cell is well-approximated by the area of its defining circle:

$$A_j \approx \pi r_j^2. \quad (14)$$

This simple approximation provides the crucial link between a cell's area ( $A_j$ ) and its generator's weight ( $w_j = r_j^2$ ).

With this physical model, I can derive a simple update rule. The goal is to adjust the radii  $\{r_j\}$  until the measured capacity  $m_j$  of each cell matches a target value  $v$ . I start by assuming that a cell's capacity is, to first order, proportional to its area:  $m_j \approx \rho_j A_j$ , where  $\rho_j$  is an effective local capacity density.

To achieve the target capacity  $v$ , cell  $j$  would need a target area  $A_j^* \approx v/\rho_j$ . Using our approximation from equation (14), the corresponding target squared radius would be  $r_j^{2*} \approx A_j^*/\pi = v/(\pi \rho_j)$ . The unknown density  $\rho_j$  can be eliminated by substituting its value from the current iteration,  $\rho_j \approx m_j/A_j$ . This yields a simple update rule for the target squared radius based entirely on measurable quantities from the current tessellation:

$$r_j^{\text{new}} \leftarrow \sqrt{\frac{v}{m_j} \frac{A_j}{\pi}} = \sqrt{\frac{f_j A_j}{\pi}}, \quad \text{where } f_j \equiv \frac{v}{m_j}. \quad (15)$$

In terms of the power weights themselves, the update is  $w_j^{\text{new}} \leftarrow f_j A_j/\pi$ .

This leads to the simple yet powerful iterative algorithm summarized in Algorithm 1. The update rule in equation (15) is formally similar to the WVT rule of S. Diehl & T. S. Statler (2006), but it operates on the radii of a power diagram, not the weights of an

MWV diagram. This is a crucial distinction: for MWV diagrams, multiplying all weights by a common factor has no effect, whereas for a power diagram, it changes the tessellation. This implies that for a MWV the constant factors are irrelevant, while for a Power Diagram they are essential. By combining the natural area-radius relation ( $A \approx \pi r^2$ ) with centroidal recentering, this heuristic achieves stable, few-iteration convergence for capacity equalization while preserving the convexity and compactness of the bins.

## 5.2 Implementation details

The success and speed of the regularization stage of the POWERBIN algorithm rely on a few crucial implementation choices, which are detailed in Algorithms 1–3.

---

### Algorithm 1 PowerBin

---

**Input:** spaxels  $\{\mathbf{x}_i\}_{i=1}^N$ , capacity function  $C(I)$ , target  $\nu$

**Output:** Bin map  $\{b_i\}$ , generators  $\{\mathbf{g}_j\}$ , radii  $\{r_j\}$

#### Initialization:

Estimate pixelsize if not provided

$\{\mathbf{x}_i\} \leftarrow \{\mathbf{x}_i\}/\text{pixelsize}$  ▷ *Normalize coordinates*

$\{\mathbf{g}_j\} \leftarrow \text{BINACCRETION}(\{\mathbf{x}_i\}, \nu, C)$  ▷ *Algorithm 4*

$r_j \leftarrow 1$  ▷ *Initialize power-cell radii for all bins j*

#### Bins Regularization:

for  $t = 1$  to  $\text{maxiter}$  do

$\mathbf{g}_j^{\text{old}} \leftarrow \mathbf{g}_j; r_j^{\text{old}} \leftarrow r_j$  ▷ *Store state for all bins j*

$(\{\mathbf{g}_j\}, \{A_j\}, \{m_j\}) \leftarrow \text{UPDATEBINS}(\{\mathbf{x}_i\}, \{\mathbf{g}_j\}, \{r_j\}, C)$

$r_j \leftarrow \sqrt{\nu A_j / (m_j \pi)}$  ▷ *For each bin j*

    if oscillation detected then

$\mathbf{g}_j \leftarrow \mathbf{g}_j^{\text{old}} + d \cdot (\mathbf{g}_j - \mathbf{g}_j^{\text{old}})$  ▷ *Apply to all j*

$r_j \leftarrow r_j^{\text{old}} + d \cdot (r_j - r_j^{\text{old}})$  ▷ *Under-relaxation d ∈ (0, 1)*

$d_{j,k} \leftarrow \|\mathbf{g}_j - \mathbf{g}_k\|$  for nearest neighbour  $k$  ▷ *For each bin j*

$r_j \leftarrow \max(0.5, \min(r_j, d_{j,k} - 0.5))$  ▷ *Prevent empty cells*

$\Delta \leftarrow \text{norm}(\{\mathbf{g}_j - \mathbf{g}_j^{\text{old}}\})$

        if  $\Delta$  is small or cycling then

            break

#### Finalize:

$\{b_i\} \leftarrow \text{POWERDIAGRAM}(\{\mathbf{x}_i\}, \{\mathbf{g}_j\}, \{r_j\})$

Rescale  $\{\mathbf{g}_j\}$  and  $\{r_j\}$  by pixelsize

return  $\{b_i\}, \{\mathbf{g}_j\}, \{r_j\}$

---

### Algorithm 2 UpdateBins

---

**Input:** Spaxels  $\{\mathbf{x}_i\}$ , generators  $\{\mathbf{g}_j\}$ , radii  $\{r_j\}$ , capacity  $C(I)$

**Output:** Updated centroids  $\{\mathbf{g}_j\}$ , areas  $\{A_j\}$ , capacities  $\{m_j\}$

$\{b_i\} \leftarrow \text{POWERDIAGRAM}(\{\mathbf{x}_i\}, \{\mathbf{g}_j\}, \{r_j\})$

$I_j \leftarrow \{i \mid b_i = j\}; A_j \leftarrow |I_j|$  ▷ *For each bin j*

$\mathbf{g}_j \leftarrow \text{mean}\{\mathbf{x}_i \mid i \in I_j\}$

$m_j \leftarrow C(I_j)$

return  $\{\mathbf{g}_j\}, \{A_j\}, \{m_j\}$

---

### Algorithm 3 PowerDiagram

---

**Input:** Spaxels  $\{\mathbf{x}_i\}$ , generators  $\{\mathbf{g}_j\}$ , radii  $\{r_j\}$

**Output:** Bin map  $\{b_i\}$

$r_{\text{max}} \leftarrow (1 + \varepsilon) \cdot \max_j \{r_j\}$

$z_j \leftarrow \sqrt{r_{\text{max}}^2 - r_j^2}$  ▷ *For each generator j*

build KDTree on lifted points  $\{(\mathbf{g}_j, z_j)\} \in \mathbb{R}^3$

$b_i \leftarrow \text{NN of } (\mathbf{x}_i, 0) \text{ in KDTree}$  ▷ *For each pixel i*

return  $\{b_i\}$

---

#### 5.2.1 Efficient power tessellation

A key advantage of power diagrams over other weighted Voronoi diagrams is their computational efficiency. A power diagram can be computed with  $\mathcal{O}(N \log N)$  complexity (F. Aurenhammer 1987), whereas an MWV diagram requires  $\mathcal{O}(N^2)$  operations (F. Aurenhammer & H. Edelsbrunner 1984). I achieve this efficiency for my discrete dataset by implementing the geometric lifting technique described in H. Imai et al. (1985, section 5), as detailed in Algorithm 3. Each 2D generator  $\mathbf{g}_j$  with radius  $r_j$  is ‘lifted’ to a 3D point  $(\mathbf{g}_j, z_j)$  where  $z_j^2 = r_{\text{max}}^2 - r_j^2$ . The power-diagram assignment for a 2D spaxel  $\mathbf{x}_i$  is then equivalent to finding the nearest 3D neighbour to the point  $(\mathbf{x}_i, 0)$  among the lifted generators. This 3D nearest-neighbour search is performed efficiently using a standard KD-Tree (`scipy.spatial.KDTree`) in the SCIPY library (P. Virtanen et al. 2020), which implements the algorithm of S. Maneewongvatana & D. M. Mount (1999).

#### 5.2.2 Geometric centroids versus barycentres

A key choice in my algorithm is the update of the generators. Instead of moving them to the capacity-weighted barycentre of each cell, I update them to the cell’s unweighted, geometric centroid (the mean position of its constituent pixels). This means I do not strictly minimize the optimal transport energy functional from equation (9). However, this choice is deliberate and crucial for robustness. The main reason is that it allows the method to handle data with negative values (e.g. background-subtracted X-ray data), where the definition of a barycentre breaks down. This choice was adopted in the standard VORBIN algorithm for the same reason. Moreover, this choice makes the algorithm a more direct implementation of the physical ‘soap bubble’ analogy (Fig. 4) that inspired my area-weight relation, as the cells are centred geometrically rather than by mass. Furthermore, for real data with non-additive capacities, the energy functional itself is ill-defined, making its formal minimization moot. For positive data, I find that my algorithm is not particularly sensitive to this choice, unlike the standard unweighted CVT which relies on the barycentric condition.

#### 5.2.3 Failure of formal optimization

The use of a heuristic update rule and geometric centroids is a direct consequence of the practical failure of formal optimization methods. During development, I implemented a version of the algorithm that directly minimized the energy functional from F. Aurenhammer et al. (1998) and F. De Goes et al. (2012) using their analytic gradient and a quasi-Newton optimizer. This approach is not just slower; it fails completely. Even in the continuum limit with many spaxels per bin, the method fails to converge to a sensible result (i.e. the capacity is not equalized) when the capacity function is non-additive, as is generally the case with real data. The fast, physically motivated heuristic of POWERBIN proved to be far superior in all practical tests, converging quickly and reliably to a high-quality solution where formal methods could not.

#### 5.2.4 Robust convergence

Iterative methods can sometimes stall or enter a cycle. To prevent this, I use under-relaxation when I detect cycling, and employ a robust early-stopping heuristic. This method monitors the sequence of generator shifts and terminates the loop if the improvement stagnates or if it detects that the values are oscillating without any significant downward trend. This ensures reliable termination even in difficult cases.

5.2.5 *Generality of capacity function*

The combination of my robust heuristic and the use of geometric centroids make the POWERBIN algorithm extremely versatile. The capacity function is not limited to the specific forms discussed in this paper and can be adapted to various applications. This flexibility allows for the incorporation of complex, non-additive constraints, making the algorithm applicable to a broad range of problems.

## 6 A LINEAR-TIME BIN-ACCRETION ALGORITHM

A crucial, and perhaps under-appreciated, aspect of all successful adaptive-binning schemes is the quality of the initial tessellation. The iterative refinement stages, whether based on a CVT, a WVT, or the CPD presented here, are all local optimizers. They are variants of S. Lloyd (1982) algorithm, which is known to be sensitive to the initial placement of the generators. Unlike the optimization of weights at fixed generator’s location, the optimization of the energy functional of equation (11) with respect to the generator positions is not convex and presents a large number of secondary minima (see B. Lévy 2015, fig. 4). This means that the final result can vary significantly depending on the starting configuration.

---

### Algorithm 4 BinAccretion

---

**Input:** spaxels  $\{\mathbf{x}_i\}_{i=1}^N$ , capacity function  $C(I)$ , target  $\nu$

**Output:** Initial bin generators  $\{\mathbf{g}_j\}$

**Initialization:**

$\rho_i \leftarrow C(\{i\})$  *► For each pixel  $i$*   
 $b_i \leftarrow 0$  *► Initialize all pixels  $i$  as unbinned*  
 Pre-compute Delaunay adjacency graph for all  $\{\mathbf{x}_i\}$   
 Build max-heap of all pixels, prioritized by  $\rho_i$   
 $j \leftarrow 1$  *► Initialize bin counter*

**Accretion Loop:**

**while** unbinned pixels exist **do**

Pop unbinned pixel  $k$  with max  $\rho_k$  from heap as new seed  
 $I_j \leftarrow \{k\}; \quad b_k \leftarrow j$   
 $m_j \leftarrow \rho_k; \quad \mathbf{c}_j \leftarrow \mathbf{x}_k$   
 $\mathcal{R}_j \leftarrow 0$  *► Initialize roundness metric*  
 $F \leftarrow$  unbinned Delaunay neighbours of  $k$   
**while**  $m_j < \nu$  and  $F$  is not empty **do**  
   Find pixel  $p \in F$  closest to centroid  $\mathbf{c}_j$   
   Update  $\mathbf{c}_j$  and  $\mathcal{R}_j$  with pixel  $p$  using Welford’s algorithm  
   **if**  $\mathcal{R}_j$  exceeds roundness threshold **then**  
     **break**  
    $m_j^{\text{old}} \leftarrow m_j; \quad m_j \leftarrow C(I_j \cup \{p\})$   
   **if**  $m_j + m_j^{\text{old}} > 2\nu$  **then** *► Capacity diverging from target*  
     **break**  
    $I_j \leftarrow I_j \cup \{p\}; \quad b_p \leftarrow j$   
   Update frontier  $F$  with neighbours of  $p$   
 $j \leftarrow j + 1$

**Finalize:**

Identify bins that failed to reach a fraction of  $\nu$   
 Reassign their pixels to the nearest successful bin  
 Compute final generators  $\{\mathbf{g}_j\}$  as centroids of the final bins  
**return**  $\{\mathbf{g}_j\}$

---

One cannot, for instance, initialize the generators with points drawn randomly from the underlying signal or  $\mathcal{S}/\mathcal{N}$  distribution and expect the iterations to converge to a satisfactory result. The

discrete nature of the data and the non-convexity of the optimization landscape mean that such an approach will invariably become trapped in a poor local minimum, yielding a tessellation with suboptimal bin shapes and poor capacity uniformity. Consequently, the bin-accretion algorithm, first introduced in M. Cappellari & Y. Copin (2003, section 4.2), has always been the indispensable foundation of the entire procedure. It provides an excellent initial guess that already satisfies the capacity constraint, allowing the subsequent refinement to focus solely on improving the bin morphology.

With the development of the fast CPD solver, which has a time complexity of  $\mathcal{O}(N \log N)$  for  $N$  pixels, the original bin-accretion algorithm became the computational bottleneck. To fully realize the performance gains of the new method, it was essential to devise an accretion algorithm with a comparable, near-linear time complexity. I achieve this through four key improvements, while the complete revised method is described in Algorithm 4:

(i) **Delaunay Adjacency:** I begin by pre-computing a single Delaunay triangulation of all input pixel coordinates. This is an  $\mathcal{O}(N \log N)$  operation that provides a static adjacency graph for the entire data set. For any given pixel, its neighbours are instantly known without requiring any further geometric searches. The computation is done using `scipy.spatial.Delaunay`, which is based on the QHull library (C. B. Barber, D. P. Dobkin & H. Huhdanpaa 1996).

(ii) **Frontier-Based Growth:** To efficiently find the next pixel to add, the algorithm restricts its search to the bin’s ‘frontier’. This frontier is defined as the set of unbinned pixels that are Delaunay neighbours to any pixel already in the current bin. By only considering this small subset of pixels at each step, the search for the one closest to the bin’s centroid becomes extremely fast.

(iii) **Incremental Updates:** All quantities required to assess the validity of adding a new pixel to a bin—namely its centroid and mean squared radius—are updated incrementally using B. P. Welford (1962) algorithm. The total capacity is also updated incrementally if it is additive; for an additive capacity, this simply means adding the capacity of the new pixel to the running total for the bin. A recomputation for the entire bin is still required in the case of a black-box generic capacity function. Adding a pixel involves a simple update to a running sum, an  $\mathcal{O}(1)$  operation, rather than a full recomputation over all pixels in the growing bin.

(iv) **Heap-Managed Seeding:** To efficiently select the starting point for each new bin, I use a max-heap data structure containing all unbinned pixels, prioritized by their initial capacity density (or ‘brightness’, or  $\mathcal{S}/\mathcal{N}$ ). After a bin is completed, the next seed is retrieved from the top of the heap in an  $\mathcal{O}(\log N)$  operation. This approach is more efficient than the original method of M. Cappellari & Y. Copin (2003), which started a new bin from the unbinned pixel closest to the centroid of all remaining unbinned pixels. The brightness of a pixel is a static property, allowing for a single heap construction at the start, whereas the centroid of unbinned pixels changes after every bin is created, requiring a costly recalculation.

In many common astronomical applications, such as binning the surface brightness of a galaxy, this change in seeding strategy has a minimal impact on the final result. In such cases, the brightest unbinned pixels are naturally located near the galaxy’s centre, which is also where the centroid of the remaining light is. The old method would accrete bins in nearly spherical shells around the galaxy’s core, while the new ‘brightest-pixel’ approach will still start bins near the centre but will tend to grow them along isophotes. In general, both methods provide an excellent starting point for the subsequent regularization. However, the new approach appears superior in the case of multiple distinct objects, as illustrated in Section 7.2.

Apart from these significant algorithmic optimizations, the new implementation aims to reproduce the logic of the original bin-accretion algorithm from M. Cappellari & Y. Copin (2003, section 5.1). There are, however, two minor differences in the acceptance criteria. (a) I employ a different definition of roundness, based on the mean squared radius of the pixel coordinates, which is faster to update incrementally. (b) The precise conditions for accepting a new pixel into a bin have been slightly adjusted. The new algorithm therefore represents a much faster, but functionally very similar, version of the well-tested accretion method. It is worth noting that the roundness criterion is not critical for the success of the overall algorithm; the final tessellation after the regularization stage is quite similar, though not identical, even if this test is omitted entirely from the accretion phase.

## 7 POWER BINNING EXAMPLES

In this section, I demonstrate the compactness and uniformity performance, and the versatility of the POWERBIN algorithm. I apply it to a range of test cases, including simulated galaxies with different morphologies and noise properties, real integral-field spectroscopic data, and a large, complex image to showcase its scalability and applicability beyond standard astronomical use cases.

### 7.1 Application to mock single galaxies

I first test the algorithm on mock galaxy data to assess its quality performance under controlled conditions. I created two types of galaxy images: one with an exponential disk profile (Sérsic  $n_{\text{Ser}} = 1$ ) and another with a highly concentrated, elliptical-like profile (e.g. J. Kormendy et al. 2009) described by a J. L. Sérsic (1968) profile ( $n_{\text{Ser}} = 8$ ). These two cases test the algorithm’s ability to handle both shallow and steep signal gradients. For each galaxy, I consider two scenarios for the noise: uncorrelated or correlated.

The results are shown in Fig. 5. To aid visualization, the tessellation is coloured using the `networkx.coloring.greedy_color` algorithm from the NetworkX package (A. A. Hagberg, D. A. Schult & P. J. Swart 2008) to approximately four-colour the Delaunay graph of the bin generators. The top two panels show the ideal case of uncorrelated, Poissonian noise. In this scenario, the bin signal-to-noise is calculated using the standard capacity function for uncorrelated noise

$$\mathcal{C}(\{i\}) = (S/\mathcal{N}_{\text{bin}})^2 = \frac{(\sum_i S_i)^2}{\sum_i \mathcal{N}_i^2}, \quad (16)$$

where  $S_i$  and  $\mathcal{N}_i$  are the signal and noise of the individual pixels in the bin. For Poissonian noise,  $\mathcal{N}_i^2 = S_i$ , and the bin capacity becomes  $(S/\mathcal{N}_{\text{bin}})^2 = \sum_i S_i$ , which is additive. The left sub-panels show the resulting power diagram tessellation. The bins are compact and convex, and their sizes adapt smoothly to the underlying signal gradient, becoming larger in the low- $S/\mathcal{N}$  outskirts. The right subpanels confirm that the final bin  $S/\mathcal{N}$  (blue points) clusters tightly around the target value (dashed line), demonstrating excellent uniformity.

The bottom two panels illustrate a more realistic scenario where noise covariance is present. I simulate this by making the capacity function non-additive, using an empirical formula derived from real IFS data to penalize the  $S/\mathcal{N}$  of bins with many pixels. Specifically, the bin  $S/\mathcal{N}$  is modified as

$$S/\mathcal{N}_{\text{bin}} \rightarrow \frac{S/\mathcal{N}_{\text{bin}}}{1 + 1.07 \lg N_{\text{pix}}}, \quad (17)$$

where  $N_{\text{pix}}$  is the number of spaxels in the bin. This formula is not general but depends on the data under study. It was derived for CALIFA data (fig. 11 of R. García-Benito et al. 2015) and is used here for illustrative purposes. This test highlights a key strength of my heuristic approach: despite the non-linear and non-additive nature of the capacity, the algorithm converges robustly and still produces bins with excellent  $S/\mathcal{N}$  uniformity. A formal gradient-based optimizer would fail to converge in this regime, but the physically motivated update rule of POWERBIN, combined with the bin-accretion initialization, handles it with ease.

### 7.2 Application to mock sky-subtracted data

A crucial consideration for modern IFS data is that they are often background-limited rather than purely Poissonian. In this regime, the noise is roughly constant across the field, meaning that accreting spaxels with very low signal can decrease a bin’s  $S/\mathcal{N}$  instead of increasing it. This raises the question of how the algorithm performs on large mosaics that may contain multiple sources of interest within a vast field of mostly ‘empty sky’.

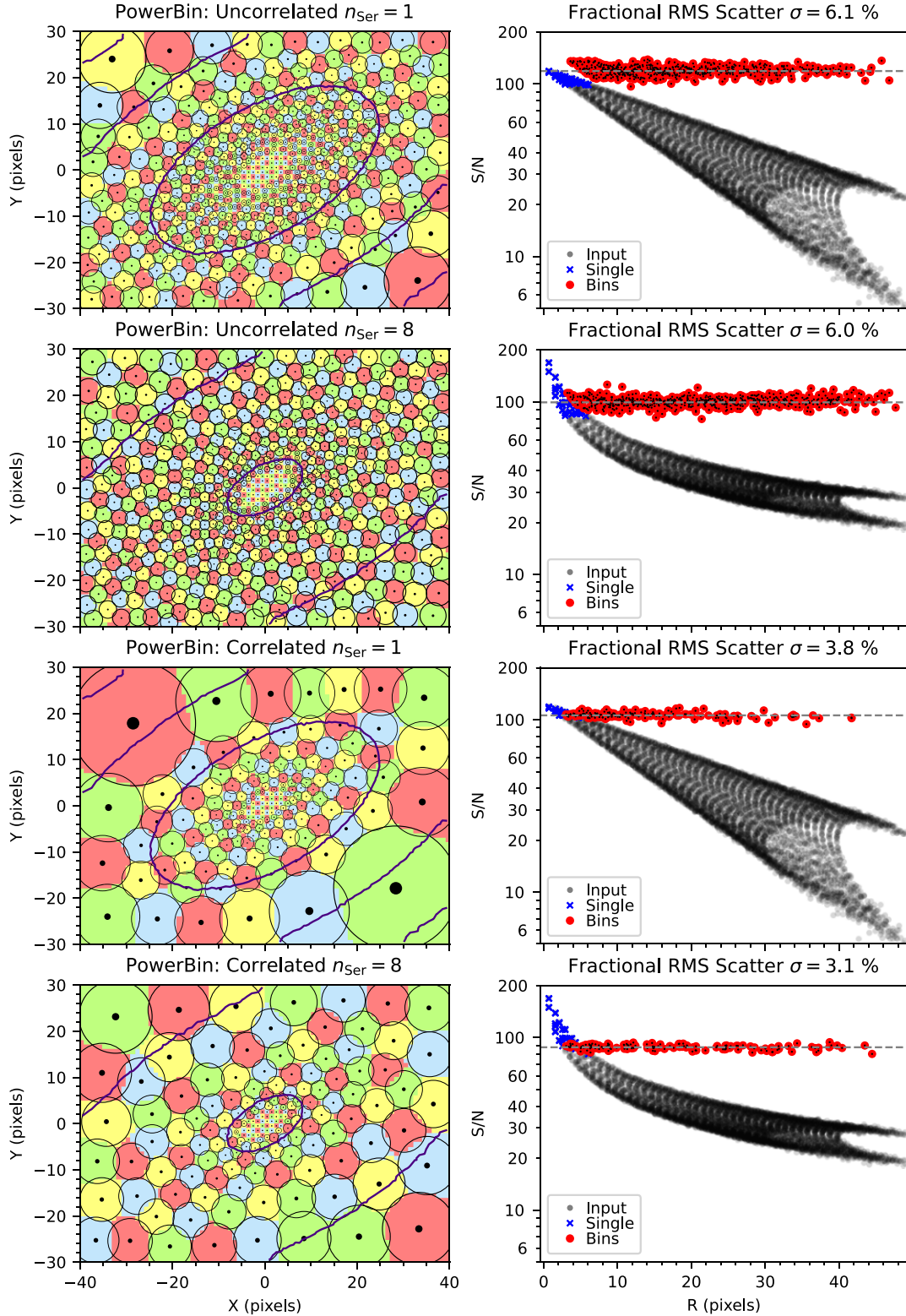
To address this scenario, I simulated a mock galaxy group image, shown in Fig. 6. The simulation generates an image by adding four exponential profile galaxies onto a constant sky background of 100 counts. The galaxy models have peak surface brightness ranging from 500 to 1000 counts. The combined image is then subjected to Poisson noise, and a slightly imperfect sky subtraction is performed to mimic a realistic data reduction process. This creates a background-limited image where the noise is nearly constant, dominated by the sky contribution. The results demonstrate that POWERBIN handles this challenging case robustly. It correctly identifies and tessellates the regions with galaxy signal, with the resulting bins clustering tightly around the target  $S/\mathcal{N}$ . The empty, noise-dominated regions are efficiently grouped into very large bins, which a user would typically discard in a subsequent analysis step.

The algorithm’s success in this regime is due to two key design choices. First, the bin-accretion stage seeds each new bin from the unbinned pixel with the highest  $S/\mathcal{N}$ . This allows the algorithm to effectively grow bins ‘in parallel’ around multiple, disconnected sources. This contrasts with the classic VORBIN approach, which started new bins from the barycentre of all remaining pixels and could risk a large noise-bin growing to engulf a faint, distant source. Second, unlike some previous implementations of VORBIN, the new accretion algorithm does not stop if adding a pixel temporarily decreases the bin’s  $S/\mathcal{N}$ . Instead, it continues to accrete neighbours until the target capacity is met or the bin’s roundness criterion is violated. This makes the process more robust in the background-dominated regime.

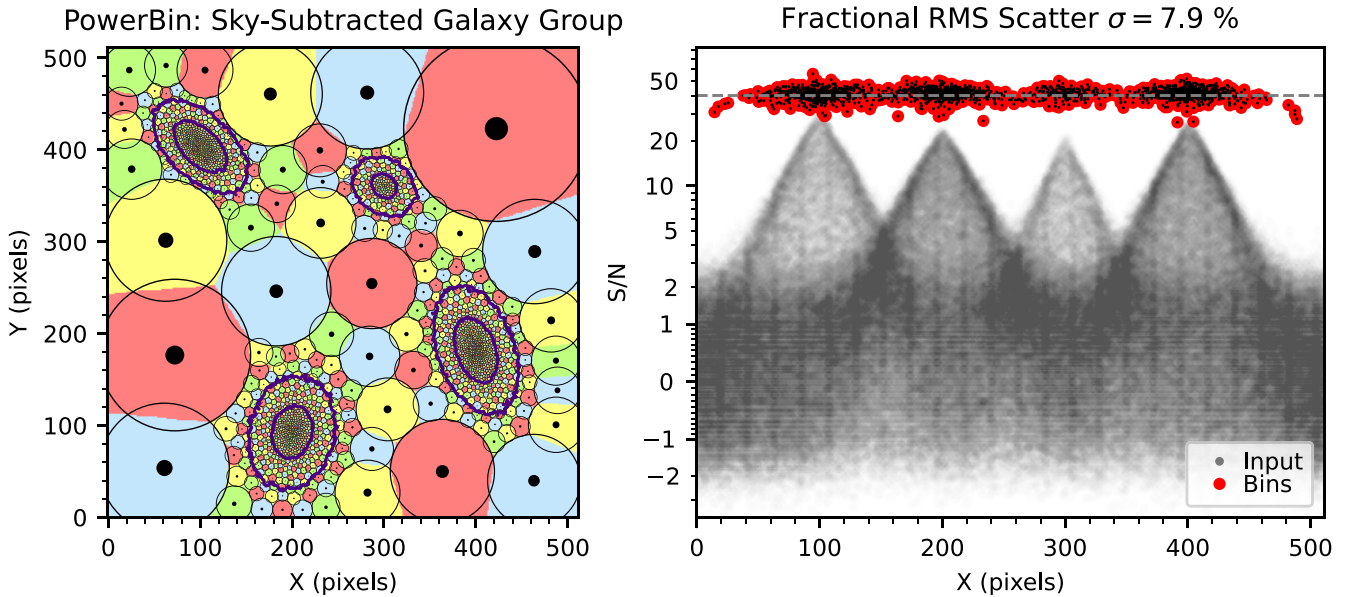
This example suggests that one could apply POWERBIN to an entire data cube and select the scientifically useful bins afterward. However, for fields with extremely sparse signals, a conservative approach of pre-selecting regions of interest before binning may still be a sensible choice.

### 7.3 Application on real IFS data

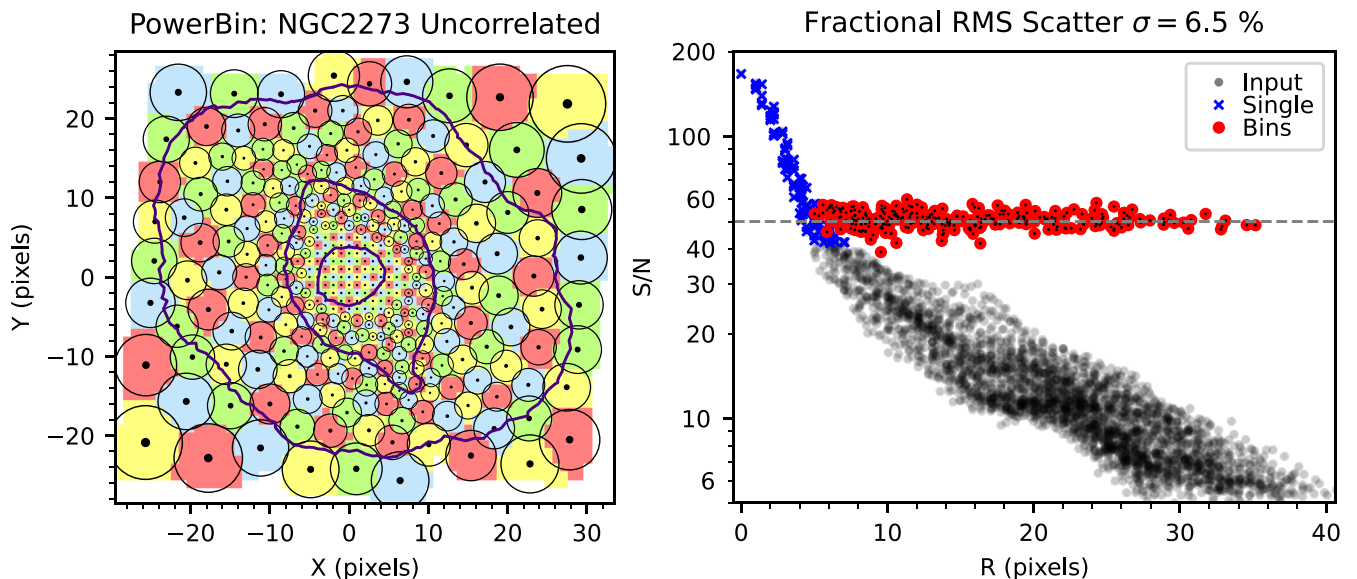
To provide a direct comparison with previous work, I apply POWERBIN to the SAURON integral-field data of the galaxy NGC 2273, shown in Fig. 7. This data set served as the primary test case for the original Voronoi-binning method and its subsequent multiplicatively weighted extension, making it a benchmark for two decades. The application of those previous methods to this same data set is shown in M. Cappellari & Y. Copin (2003, fig. 8) and S.



**Figure 5.** The performance of the POWERBIN algorithm is shown for two mock galaxies ( $n_{\text{Ser}} = 1$  and  $n_{\text{Ser}} = 8$ ). Both simulations have  $R_{\text{eff}} = 10$  pixels and an axial ratio of  $q' = 0.5$ . The target signal-to-noise ratio ( $S/N_{\text{T}}$ ) was calibrated to ensure  $\approx 40$  unbinned spaxels near the centre. *Left:* Resulting Power Diagrams; circle radii are  $r_j = \sqrt{w_j}$ , with small black circles marking the generator points. Contours show the galaxy's  $S/N$ , spaced logarithmically by 1 mag. *Right:*  $S/N$  distribution. Original spaxel  $S/N$  values are grey circles;  $S/N_{\text{T}}$  is the dashed line. Blue crosses are single spaxels ( $S/N > S/N_{\text{T}}$ ), and large red circles are the final bin  $S/N$ . Top panels use a simple Poissonian noise model; bottom panels show robustness to non-additive capacities from correlated noise. In all cases, the algorithm optimizes the capacity function  $\mathcal{C} = (S/N)^2$  (which is an additive capacity in the Poissonian case), but I plot the square root  $\sqrt{\mathcal{C}} = S/N$ .



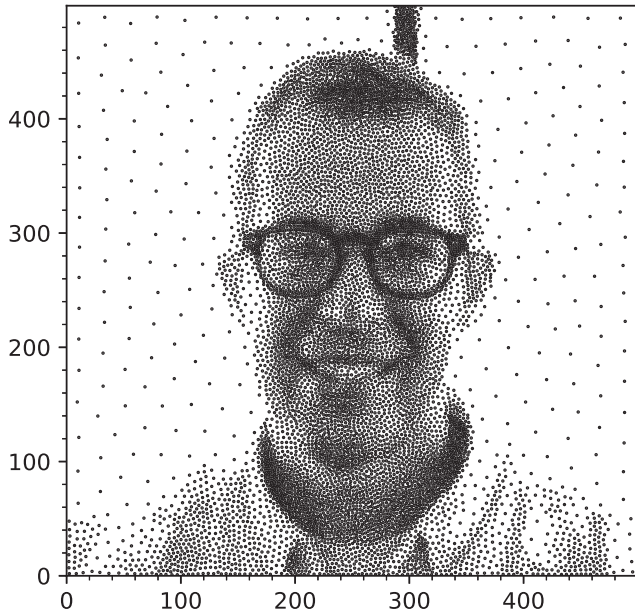
**Figure 6.** Application of the POWERBIN algorithm to a mock galaxy group, simulating a background-limited observation with multiple sources in a large, noisy field. For the large majority of the image spaxels, the signal is essentially zero, with only some nearly constant random noise remaining after imperfect sky subtraction. The plot format is the same as in Fig. 5. *Left panel:* The final power diagram tessellation. Circles indicate the bin radii ( $r_j = \sqrt{w_j}$ ), and small black disks mark the generators. Galaxy isophotes are spaced by 1 mag in  $S/N$ . The algorithm correctly bins the galaxy signal while filling the empty background with large, low-signal bins. *Right panel:* The  $S/N$  distribution, showing the original spaxel  $S/N$  (grey filled circles), the target  $S/N_T$  (dashed line), and the final bins  $S/N$  (red filled circles). The bins are clustered tightly around the target value, with the exceptions of some pure-noise outer bins falling slightly below  $S/N_T$ , demonstrating the algorithm’s robustness. Here the spaxels capacity can be negative and I equalize the capacity function  $C = S/N$  directly (not squared).



**Figure 7.** Application of the POWERBIN algorithm to the SAURON integral-field data of the galaxy NGC 2273. This data set was used as the primary test case in the original Voronoi-binning paper (M. Cappellari & Y. Copin 2003) and has been included for reference in the public VORBIN software package for two decades. The plot format is the same as in Fig. 5, assuming uncorrelated noise. *Left panel:* The final power diagram tessellation, with circles indicating the bin radii ( $r_j = \sqrt{w_j}$ ) and black disks marking the generators. Galaxy isophotes are spaced by 1 mag in  $S/N$ . *Right panel:* The  $S/N$  distribution, showing the original spaxel  $S/N$  (grey filled circles), the target  $S/N_T$  (dashed line), the unbinned spaxels (blue crosses), and the final bin  $S/N$  (red filled circles).

Diehl & T. S. Statler (2006, fig. 5) respectively, and can be directly compared with my Fig. 7. The figure shows that the new algorithm performs flawlessly on this classic data set. It produces a clean, convex tessellation that adapts to the galaxy’s morphology, and the

resulting bin  $S/N$  is highly uniform around the target value. The example included in the public VORBIN package, using the WVT regularization, gives a fractional rms scatter of 7.3 per cent, which is slightly larger than the 6.5 per cent scatter produced by POWERBIN



**Figure 8.** Demonstration of the POWERBIN algorithm’s ability to handle large data sets with sharp, irregular discontinuities. For this test, I used a  $500 \times 500$  pixel greyscale self-portrait. The flux was inverted so that dark regions correspond to high signal, and the algorithm was tasked with partitioning the image into  $10^4$  bins of equal integrated flux. The figure shows the final positions of the bin generators. As expected, the generators form a ‘blue noise’ distribution, with their density tracing the underlying signal. This illustrates the connection between capacity-constrained power diagrams and stippling algorithms used in computer graphics. This example highlights the versatility and scalability of POWERBIN for a wide range of applications beyond astronomical data.

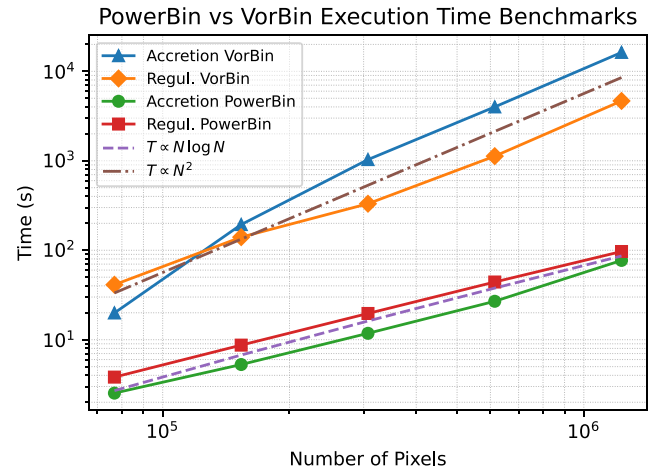
on the same input data. This confirms that POWERBIN successfully matches the results of the original method on real astronomical data.

This paper focuses on the algorithm and does not present new scientific applications, such as the generation of kinematic maps. This is because, while POWERBIN resolves the critical issues of non-convex bins and slow computation, the resulting tessellations are visually very similar to those from the classic VORBIN method. The latter has been successfully applied to hundreds of data sets over many years.

Instead of reproducing similar science results, I refer the reader to existing work for examples of the high-quality scientific products that can be derived from this binning approach. For beautiful maps of stellar kinematics and populations, particularly from high-quality MUSE data, see, for instance, D. Krajnović et al. (2015), M. Mitzkus, M. Cappellari & C. J. Walcher (2017), D. A. Gadotti et al. (2019), D. A. Gadotti et al. (2020), and A. Bittner et al. (2020). For the largest applications of VORBIN to date on the ever-increasing samples from major IFS surveys, see the results from the ATLAS<sup>3D</sup>, CALIFA, SAMI, and MaNGA surveys in M. Cappellari et al. (2011), J. Falcón-Barroso et al. (2017), J. van de Sande et al. (2017), and K. B. Westfall et al. (2019), respectively.

#### 7.4 General-purpose tessellation

Finally, to demonstrate the scalability and versatility of POWERBIN on a non-astronomical problem, I apply it to a task common in computer graphics: creating a stipple drawing from an image. I took a  $500 \times 500$  pixel greyscale self-portrait (Fig. 8) and tasked the



**Figure 9.** Benchmark comparison of the computational time for the classic VORBIN algorithm and the new POWERBIN method. The plot shows the execution time as a function of the number of input pixels,  $N$ , for a simulated galaxy image. The four curves represent the two main stages of each algorithm: bin accretion and iterative regularization. The classic method (VORBIN, top two curves) shows a steep scaling that approaches the theoretical  $\mathcal{O}(N^2)$  complexity of multiplicatively-weighted Voronoi diagrams. The new method (POWERBIN, bottom two curves) demonstrates a significantly improved performance, closely following the optimal  $\mathcal{O}(N \log N)$  scaling expected for power diagrams. Crucially, the bin-accretion stage of POWERBIN was also dramatically improved to follow a similar scaling.

algorithm with partitioning it into  $10^4$  bins of equal integrated flux. To achieve the desired artistic effect, the image was inverted so that dark regions correspond to high signal.

The result, shown in Fig. 8, plots the final positions of the bin generators. The algorithm successfully handles this large and complex input, which features sharp, irregular discontinuities. As expected from the connection to optimal transport, the generators form a ‘blue noise’ point distribution, where their density traces the underlying signal structure. This example showcases the computational efficiency and robustness of POWERBIN on large data sets. The entire process took just 20 seconds on a standard laptop, with 12 seconds for the bin-accretion phase and 8 seconds for the CPD regularization. This highlights its potential as a general-purpose tool for capacity-constrained tessellation in a wide variety of scientific and technical fields.

## 8 EXECUTION TIME BENCHMARKS

To quantify the performance improvement of the new algorithm, I conducted a series of benchmark tests comparing the execution time of POWERBIN against the classic VORBIN package. All tests were performed on a standard laptop with an Intel i7-1355U processor. The process was running on a single core at a sustained frequency of about 3GHz. The results are presented in Fig. 9.

For these tests, I generated a sequence of mock galaxy images of increasing size. The input signal for all tests was a simulated galaxy with an exponential surface brightness profile (Sérsic  $n_{\text{Ser}} = 1$ ), a fixed axial ratio of 4/3, and Poisson noise, corresponding to the uncorrelated noise case shown in the top panel of Fig. 5. I created five images, starting at  $320 \times 240$  pixels and progressively doubling the total number of pixels up to  $1280 \times 960$ . For each image, the target signal-to-noise was chosen to make the number of bins,  $n$ , scale

proportionally with the total number of pixels,  $N$ . This approach, which keeps the average number of pixels per bin constant, simulates a common scientific goal: exploiting a larger number of pixels to increase the spatial sampling of the binned map (i.e. more bins). This setup provides a realistic benchmark of how the algorithm's performance scales as both the number of input pixels and output bins increase. I adopted a number of bins logarithmically spaced from  $n = 1600$  to  $n = 25600$ .

Fig. 9 plots the execution time versus the number of input pixels on a log-log scale for the two main computational stages: the accretion and regularization steps for both the old VORBIN and the new POWERBIN algorithms. The performance difference is dramatic and confirms the expected theoretical scaling laws.

The classic VORBIN method, shown by the upper two curves, exhibits a computational time that scales significantly more steeply than  $\mathcal{O}(N \log N)$ . At large  $N$ , both its accretion (blue triangles) and regularization (orange diamonds) stages approach a scaling consistent with  $\mathcal{O}(N^2)$  (dot-dashed brown line). This is the expected behaviour, as the regularization is based on a MWV diagram, which has a quadratic time complexity. Similarly, the classic bin-accretion algorithm also performs operations on average linear in  $N$  for every pixel, leading to a  $\mathcal{O}(N^2)$  time complexity.

In stark contrast, the new POWERBIN algorithm, shown by the lower two curves, demonstrates vastly superior performance. Both the new fast bin-accretion stage (green circles) and the CPD regularization (red squares) follow a trend that is nearly perfectly described by the theoretical  $\mathcal{O}(N \log N)$  scaling (dashed purple line). This is the optimal complexity for this class of geometric problem and is a direct result of the algorithmic improvements described in Sections 5 and 6. The bottom line is that for a data set with one million pixels, the new algorithm is approximately two orders of magnitude faster than the previous standard, turning a computation that would take 6 h into one that takes 3 min. This efficiency gain is critical for the practical analysis of large-scale astronomical surveys.

It is important to emphasize that this benchmark compares the *relative* performance and algorithmic scaling of the two methods. Both the classic VORBIN and the new POWERBIN are implemented entirely in PYTHON, relying on standard scientific libraries like NUMPY (C. R. Harris et al. 2020) and SCIPY (P. Virtanen et al. 2020). The absolute execution times could be substantially reduced by porting the computationally intensive parts to a compiled language or specialized hardware like Graphics Processing Units (GPUs). However, such optimizations would not change the fundamental time complexity of the algorithms. The key result of this comparison is the difference in scaling –  $\mathcal{O}(N \log N)$  versus  $\mathcal{O}(N^2)$  – which demonstrates the inherent efficiency of the new approach, independent of the specific implementation.

## 9 CONCLUSIONS

In this paper, I have introduced POWERBIN, a new algorithm for the adaptive binning of two-dimensional data. This work was motivated by the increasing scale of modern astronomical surveys and the limitations of existing methods, which are either too slow or lack guarantees of bin convexity. The main contributions of this work can be summarized as follows:

(i) **A New Theoretical Framework:** I have framed the adaptive binning problem within the mathematical theory of optimal transport. The natural solution in this framework is a CPD, a generalization of a CVT that rigorously accommodates capacity constraints while guaranteeing convex bins.

(ii) **A Fast and Robust Heuristic Solver:** Formal CPD solvers, based on gradient-based optimization of a dual energy functional, are ill-suited to real astronomical data, which is discrete and often has non-additive noise properties. I have introduced a novel heuristic algorithm that circumvents these issues. It is based on a simple physical insight into the geometry of packed cells, which provides a direct, non-linear update rule for the power diagram weights to enforce the target capacity. This approach is fast, robust, and converges reliably even when the capacity function is non-additive, a regime where formal methods fail.

(iii) **An Optimized Bin-Accretion Algorithm:** The bin-accretion stage, which provides the crucial starting point for the iterative refinement, would have become the computational bottleneck of the current methods. Therefore, I developed a new implementation with near-linear time complexity,  $\mathcal{O}(N \log N)$ . This is achieved by using a pre-computed Delaunay triangulation for adjacency information, a frontier-based growth strategy, incremental updates for bin properties, and a heap-managed frontier to efficiently select pixels.

(iv) **Superior Performance and Scalability:** The combination of the fast CPD solver and the optimized bin-accretion algorithm results in a dramatic performance improvement. Benchmark tests show that the entire POWERBIN algorithm scales as  $\mathcal{O}(N \log N)$ , in stark contrast to the  $\mathcal{O}(N^2)$  scaling of previous methods. For a data set of one million pixels, the new algorithm is approximately two orders of magnitude faster than the widely-used VORBIN package.

I have demonstrated through a series of tests on simulated and real data that POWERBIN produces high-quality, convex tessellations with excellent capacity uniformity. It successfully handles a wide range of signal distributions and is robust to the challenges of correlated noise. Its robustness was further demonstrated on challenging background-subtracted mock observations containing multiple objects, where the noise is highly non-Poissonian. Its performance on the classic SAURON data of NGC 2273 confirms that it reproduces and improves upon the results of the original Voronoi-binning method.

By addressing the key limitations of speed and convexity, POWERBIN provides a powerful and scalable tool for the analysis of the massive data sets generated by current and future astronomical surveys. Its versatility, demonstrated on a non-astronomical imaging problem, also suggests its potential for broad application in other scientific and technical fields requiring capacity-constrained tessellation. Notably, both the bin-accretion and regularization stages of POWERBIN extend naturally to higher dimensions with minimal code changes and no change in the underlying concept. The Python implementation of the algorithm is publicly available to the community in the POWERBIN package.<sup>4</sup>

## ACKNOWLEDGEMENTS

I thank the referee for a constructive and insightful report that helped improve the clarity and completeness of this paper.

## DATA AVAILABILITY

No new data were generated in support of this research. The POWERBIN package is available at <https://pypi.org/project/powerbin/>.

<sup>4</sup><https://pypi.org/project/powerbin/>.

## REFERENCES

- Abdurro'uf et al., 2022, *ApJS*, 259, 35
- Aurenhammer F., 1987, *SIAM J. Comput.*, 16, 78
- Aurenhammer F., Edelsbrunner H., 1984, *Pattern Recogn.*, 17, 251
- Aurenhammer F., Hoffmann F., Aronov B., 1998, *Algorithmica*, 20, 61
- Aurenhammer F., Klein R., Lee D.-T., 2013, *Voronoi Diagrams and Delaunay Triangulations*, 1st edn. World Scientific Publishing Co., Inc., River Edge, NJ. Available at: <https://doi.org/10.1142/8685>
- Bacon R. et al., 2010, in McLean I. S., Ramsay S. K., Takami H., eds, Proc. SPIE Conf. Ser. Vol. 7735, Ground-based and Airborne Instrumentation for Astronomy III. SPIE, Bellingham, p. 8. Available at: <https://doi.org/10.1117/12.856027>
- Bacon R. M. et al., 2024, in Marshall H. K., Spyromilio J., Usuda T., eds, Proc. SPIE Conf. Ser. Vol. 13094, Ground-based and Airborne Telescopes X. SPIE, Bellingham, p. 61. Available at: <https://doi.org/10.1117/12.3018093>
- Barber C. B., Dobkin D. P., Huhdanpaa H., 1996, *ACM T. Math. Softw.*, 22, 469
- Bittner A. et al., 2020, *A&A*, 643, A65
- Bryant J. J. et al., 2015, *MNRAS*, 447, 2857
- Bundy K. et al., 2015, *ApJ*, 798, 7
- Cappellari M., 2017, *MNRAS*, 466, 798
- Cappellari M., 2023, *MNRAS*, 526, 3273
- Cappellari M., Copin Y., 2003, *MNRAS*, 342, 345
- Cappellari M., Emsellem E., 2004, *PASP*, 116, 138
- Cappellari M. et al., 2011, *MNRAS*, 413, 813
- Cappellari M. et al., 2013, *MNRAS*, 432, 1709
- Congiu E. et al., 2025, *A&A*, 700, A125
- De Goes F., Breeden K., Ostromoukhov V., Desbrun M., 2012, *ACM T. Graph.*, 31, 1
- Diehl S., Statler T. S., 2006, *MNRAS*, 368, 497
- Diehl S., Statler T. S., 2007, *ApJ*, 668, 150
- Du Q., Faber V., Gunzburger M., 1999, *SIAM Rev.*, 41, 637
- Emsellem E. et al., 2022, *A&A*, 659, A191
- Falcón-Barroso J. et al., 2017, *A&A*, 597, A48
- Fortune S., 1986, in Proceedings of the second annual symposium on Computational geometry – SCG '86. SCG '86. ACM Press, New York, p. 313. Available at: <https://doi.org/10.1145/10515.10549>
- Fraser-McKelvie A. et al., 2025, *A&A*, 700, A237
- Gadotti D. A. et al., 2019, *MNRAS*, 482, 506
- Gadotti D. A. et al., 2020, *A&A*, 643, A14
- García-Benito R. et al., 2015, *A&A*, 576, A135
- Gelman A., Carlin J. B., Stern H. S., Dunson D. B., Vehtari A., Rubin D. B., 2014, *Bayesian Data Analysis*, 3rd edn. CRC Press, Boca Raton. Available at: <https://doi.org/10.1201/b16018>
- Haario H., Saksman E., Tamminen J., 2001, *Bernoulli*, 7, 223
- Hagberg A. A., Schult D. A., Swart P. J., 2008, in Proceedings of the 7th Python in Science Conference. SciPy, Pasadena, CA, p. 11. Available at: <https://doi.org/10.25080/twv9851>
- Harris C. R. et al., 2020, *Nature*, 585, 357
- Imai H., Iri M., Murota K., 1985, *SIAM J. Comput.*, 14, 93
- Kormendy J., Fisher D. B., Cornell M. E., Bender R., 2009, *ApJS*, 182, 216
- Krajinović D. et al., 2015, *MNRAS*, 452, 2
- Lévy B., 2015, *ESAIM: M2AN*, 49, 1693
- Lévy B., Schwindt E. L., 2018, *Comput. Graph.*, 72, 135
- Lloyd S., 1982, *IEEE T. Inf. Theor.*, 28, 129
- Lu S., Zhu K., Cappellari M., Li R., Mao S., Xu D., 2023, *MNRAS*, 526, 1022
- Maneewongvatana S., Mount D. M., 1999, *Analysis of Approximate nearest Neighbor Searching with Clustered Point Sets*. preprint (arXiv:cs/9901013)
- McDermid R. M. et al., 2015, *MNRAS*, 448, 3484
- Mérigot Q., 2011, *Compu. Graph. Forum*, 30, 1583
- Mitzkus M., Cappellari M., Walcher C. J., 2017, *MNRAS*, 464, 4789
- Monge G., 1781, *Mem. Math. Phys. Acad. Royale Sci.*, 1781, 666 (available at <https://gallica.bnf.fr/ark:/12148/bpt6k35800/f796>)
- Okabe A., Boots B., Sugihara K., Chiu S. N., 2000, *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*. Wiley Series in Probability and Statistics Vol. 501. John Wiley & Sons, Chichester. Available at: <https://doi.org/10.1002/9780470317013>
- Peyré G., Cuturi M., 2019, *Found. Trends Mach. Learn.*, 11, 355
- Sánchez S. F. et al., 2012, *A&A*, 538, A8
- Sanders J. S., Fabian A. C., Allen S. W., Schmidt R. W., 2004, *MNRAS*, 349, 952
- Sarzi M., Spiniello C., La Barbera F., Krajinović D., van den Bosch R., 2018, *MNRAS*, 478, 4084
- Scott N. et al., 2017, *MNRAS*, 472, 2833
- Sérsic J. L., 1968, *Atlas de Galaxias Australes*. Obs. Astron. Univ. Nacional de Córdoba. Córdoba (translation at <https://doi.org/10.5281/zenodo.2562394>)
- van de Sande J. et al., 2017, *ApJ*, 835, 104
- Vazdekis A., Sánchez-Blázquez P., Falcón-Barroso J., Cenarro A. J., Beasley M. A., Cardiel N., Gorgas J., Peletier R. F., 2010, *MNRAS*, 404, 1639
- Virtanen P. et al., 2020, *Nat. Methods*, 17, 261
- Weilbacher P. M. et al., 2015, *A&A*, 582, A114
- Welford B. P., 1962, *Technometrics*, 4, 419
- Westfall K. B. et al., 2019, *AJ*, 158, 231
- Woo J., Walters D., Archinuk F., Faber S. M., Ellison S. L., Teimoorinia H., Iyer K., 2024, *MNRAS*, 530, 4260

This paper has been typeset from a  $\text{\TeX}/\text{\LaTeX}$  file prepared by the author.