

Deep learning for filament segmentation: Hyperparameter estimation and limitations

L. Berthelot^{1,2,*}, T. Artières¹, A. Zavagno^{2,3}, D. Arzoumanian⁴, E. Schisano⁵, F.-X. Dupé¹, and M. Gray²

¹ Aix Marseille Univ, CNRS, LIS, Marseille, France

² Aix Marseille Univ, CNRS, CNES, LAM Marseille, France

³ Institut Universitaire de France, 1 rue Descartes, 75005 Paris, France

⁴ National Astronomical Observatory of Japan, Osawa 2-21-1, Mitaka, Tokyo 181-8588, Japan

⁵ INAF-IAPS, Via Fosso del Cavaliere 100, Rome, Italy

Received 19 November 2025 / Accepted 5 August 2026

ABSTRACT

Context. Detecting filamentary structures in molecular clouds is essential for understanding star formation, yet remains difficult due to the evolving nature of filaments, their wide variety, and in practice incomplete datasets (partially labeled data). Despite promising results with deep learning, the filament detection field lacks standardized, ground-truth–complete benchmarks for method comparison leading to method improvement.

Aims. We use a synthetic benchmark dataset to study how filament detection methods behave when their parameters are selected automatically from training and validation sets. The objective is to test the scalability of a single, globally optimized configuration (hyper-parameter set) to a large and heterogeneous dataset. We also assess the impact of incomplete annotations – common in observational datasets – on model bias, focusing on low-contrast and low-density filaments.

Methods. We introduce a synthetic benchmark based on Plummer-like radial profiles with tunable parameters and spatially coherent backgrounds. This is motivated by observational results of filament properties (Herschel PACS/SPIRE). The synthetic dataset yields realistic column-density maps with fully known ground truth. We leverage this dataset to apply a common training-validation-test protocol to three approaches: UNet for semantic segmentation, Mask R-CNN for instance segmentation, and FilFinder as a morphology-based method. For FilFinder, the experiment should be understood as an automatic grid-search study over global parameters, rather than as an assessment of the method under manual, field-specific tuning.

Results. Under this global automatic-tuning protocol, FilFinder is difficult to summarize with a single parameter set across the full heterogeneity of the synthetic dataset. In contrast, UNet and Mask-RCNN adapt their internal parameters during training and achieve stronger aggregate scores. Semantic and instance segmentation paradigms exhibit complementary strengths. Crucially, training on observationally biased labels induces pronounced failures on low-contrast (low- p) and low-density filaments; unbiased training restores performance across these regimes.

Conclusions. The proposed Plummer-based benchmark provides a rigorous and flexible framework for studying both automatic parameter tuning and learning biases in filament detection. More broadly, the results emphasize the need for complete, physically informed supervision and caution against training on incomplete catalogs. Synthetic, physics-aware data therefore provide a useful controlled environment for method development, parameter-selection studies, and bias diagnosis.

Key words. stars: formation – ISM: clouds – ISM: structure – infrared: ISM

1. Introduction

Filamentary structures are a ubiquitous component of the interstellar medium (ISM), playing a central role in the star formation process (Schneider & Elmegreen 1979; André et al. 2010; Arzoumanian et al. 2011; Hacar et al. 2018; Arzoumanian et al. 2019; Shimajiri et al. 2019; Clarke et al. 2020; Priestley & Whitworth 2022; Hacar et al. 2023; Pillsworth & Pudritz 2024). Recent observational surveys, particularly those conducted by the *Herschel* Space Observatory (Molinari et al. 2010), have revealed that dense filaments pervade both low- and high-mass star-forming regions, often acting as the main sites of prestellar core formation (Molinari et al. 2010; André et al. 2014; Könyves et al. 2020). To understand filaments, we must not only detect them but also track their evolution over time – especially during their formation, when they contain little material and thus appear

as low-density, low-contrast structures. The ability to detect and characterize these filaments is thus a prerequisite for understanding the large-scale organization of the ISM and the initial conditions of star formation, with particular care for low-density and -contrast filaments. Numerous filament detection algorithms have been proposed over the past decades, spanning a wide range of techniques from local pixel-based gradient filters to global and multiscale topological approaches (Sousbie 2011; Koch & Rosolowsky 2015) (see Berthelot et al. (2024) for a comprehensive presentation). While these methods have been successfully applied to both observational and simulated data, they usually require extensive hyperparameter tuning and remain sensitive to factors such as noise, background fluctuations, and data resolution. In recent years, machine-learning-based approaches have emerged as powerful alternatives, offering the ability to optimize hyperparameters based on previous detection and extract filamentary structures with limited manual intervention (Alina et al. 2022; Xu et al. 2023; Zavagno et al. 2023; Berthelot et al. 2024).

* Corresponding author: loris.berthelot@lis-lab.fr

Evaluating filament detection remains difficult because the true filament population is unknown in observations, precluding objective ground truth. While numerical simulations play a central role in understanding the formation and evolution of molecular clouds and their filamentary substructures, they suffer from the same limitations. These simulations incorporate a variety of physical processes, including gravity, turbulence, magnetic fields, and stellar feedback, to replicate the complex behavior of the ISM. Magnetohydrodynamic simulations (Seifried et al. 2017; Walch et al. 2015; Suin et al. 2022) have been particularly instrumental in demonstrating how supersonic flows and magnetic fields contribute to filament formation. For instance, Inoue & Inutsuka (2012) showed that molecular clouds arise from repeated episodes of compression in a magnetized medium, while Kritsuk & et al. (2013) emphasized the role of coupled chemistry and radiative cooling in cloud evolution. Comprehensive reviews such as Hennebelle & Inutsuka (2019) highlight the importance of magnetic support and turbulence in shaping cloud morphology and fragmentation. On galactic scales, simulations such as the Cloud Factory project (Smith et al. 2019) have resolved filamentary clouds within large-scale dynamics, offering a bridge between cloud formation and star formation environments. Other studies such as Wareing et al. (2016) have explored the interplay between stellar feedback and filamentary structure, demonstrating how shocks and ionization fronts reorganize dense gas into elongated features. Although these simulations offer valuable physical insights, filaments are not prescribed but emerge from matter evolution. As a result, identifying them still requires filament detection algorithms akin to those used on observations, with the same ambiguities as observational studies (subjected to hyperparameter tuning).

As a consequence, this situation introduces a systematic labeling bias in observational catalogs: labeled filaments are dominated by bright, well-defined filaments that experts (or tuned algorithms) can reliably extract. As a result, training neural networks on such catalogs yields models that are biased toward high-contrast, high-density structures and that underperform on faint, low-contrast (and low-density) filaments. Moreover, existing studies generally apply a single method to a specific dataset (Green et al. 2017; Men'shchikov 2021), with performance assessments based on visual inspection or expert judgment rather than ground-truth-based metrics. This absence of an objective comparison severely limits our ability to quantify progress and identify the most effective approaches. We argue that advancing the field of filament detection critically depends on the development of one or more reference datasets.

In the context of machine learning, the previously described setting – whereby the complete ground truth is not available for training and evaluation – is typically framed as a semi-supervised learning problem (Jiao et al. 2024). Classical semi-supervised approaches often rely on techniques such as pseudo-labeling with teacher–student models (Lee 2013), entropy minimization (Grandvalet & Bengio 2004), or consistency-based bootstrapping strategies (Qiao et al. 2018). While these methods have shown great success in natural image domains, applied machine learning in scientific fields often adopts a different strategy when confronted with limited labeled data. In particular, a common and effective approach is to generate synthetic yet realistic datasets that reflect properties of real-world studied objects (Sekh et al. 2021; Hong et al. 2021).

In this work, we adopt this physics-informed approach of using prior knowledge to model the target and its environment, thereby fixing label positions and properties and avoiding incomplete annotations. We construct a fully synthetic dataset in which

filament geometry and physical parameters are known by design. Motivated by observations *Herschel* PACS and SPIRE observations (Molinari et al. 2010; André et al. 2010; Arzoumanian et al. 2019); Schisano et al. (2020) and references therein) that high-column-density, high-contrast filaments are well described by Plummer-like radial profiles (Nutter et al. 2008; Arzoumanian et al. 2011, 2019; Mannfors et al. 2025) and by theoretical works (Ostriker 1964; Smith et al. 2014; Myers 2017), we define a filament as a skeleton whose radial profile is well fit by a Plummer-like function. We choose to extend this very same filament definition for low-contrast and low-density filaments, which are not yet well studied in observations. Each synthetic filament is characterized by a central spine (skeleton) and a set of three physical parameters: the central density (n_0), the flat radius (R_{flat}), and the power-law index (p). This model provides a physically meaningful and flexible framework for simulating both low- and high-density and low- and high-contrast filaments. By generating filaments with Plummer-like radial profiles embedded in realistic backgrounds, we create controlled and diverse data for training and testing. This creates (i) a rigorous benchmark with full ground truth and (ii) a clean framework to quantify biases against low-density and low-contrast filaments.

For the purpose of this study, we selected three filament detection approaches, each corresponding to a distinct methodological category. First, FilFinder (Koch & Rosolowsky 2015) was used as a morphology-based method whose parameters are normally selected by the user according to the map and scientific objective. Here, we deliberately placed it in a supervised-style training-validation-test protocol in order to test whether a single automatically selected parameter set can scale to a heterogeneous synthetic dataset. This choice should therefore be interpreted as a study of automatic global parameter tuning, not as a claim about the best performance attainable by FilFinder under expert, case-by-case tuning. Second, a standard U-Net architecture (Ronneberger et al. 2015) was employed to illustrate semantic segmentation techniques based on deep learning. Finally, the Mask-RCNN framework (He et al. 2017) was adopted to evaluate performance in the context of instance segmentation (see Gu et al. (2022) for a review).

The paper is organized as follows. Section 2 provides an overview of filament detection methods. Section 3 introduces the physical model adopted for filament generation, followed by a detailed description of the dataset construction process. Section 4 presents the experimental setup, including the common automatic parameter-selection protocol used for the benchmark. Benchmark, parameter-sensitivity, and bias-study results are presented in Sect. 5. Finally, Sect. 6 summarizes the main findings and outlines future directions.

2. Filament detection methods

Filament detection in astronomical maps has developed along four main methodological lines: (i) derivative-based ridge enhancement, (ii) morphological and skeleton extraction, (iii) template–pattern matching at multiple scales, and (iv) learning-based segmentation. Each family makes different trade-offs between robustness to background fluctuations, scale selectivity, ability to recover topology, and computational cost. Overall, classical methods offer interpretability and strong orientation diagnostics, but at a high computational and tuning cost (see Appendix F for details), while learning-based methods provide adaptability and state-of-the-art accuracy given sufficient training data. Persistent challenges include sensitivity to background

and contrast, parameter tuning, projection effects (coincidence of structures along the line of sight), and the need for consistent, bias-aware evaluation datasets.

2.1. Derivative-based methods

First- and second-order image derivatives are used to highlight elongated intensity ridges. Gradient fields have been leveraged to analyze filament orientation with respect to the magnetic field (Soler et al. 2013; Planck Collaboration Int. XXXII 2016). Hessian-based approaches classify pixels via the eigenvalues of the local curvature matrix, often across multiple Gaussian scales; thresholding the most negative eigenvalue isolates ridge-like structures (Polychroni et al. 2013; Schisano et al. 2014, 2020; Pezzuto et al. 2021; Fiorellino et al. 2021). The multiscale Frangi “vesselness” filter (Frangi et al. 1998) further enhances line-like features and has been successfully applied to continuum maps (Salji et al. 2015a,b). These methods are scale-aware but can be sensitive to noise, threshold choice, and background gradients.

2.2. Morphological and skeleton-based methods

Topological skeletonization recovers the network of filament crests and their connectivity. DisPerSE uses critical points and thresholds to extract robust skeletons from 2D and 3D scalar fields (Sousbie 2011), and has been widely adopted on Herschel maps and spectral cubes (Arzoumanian et al. 2011; Peretto et al. 2012; Palmeirim et al. 2013; Panopoulou et al. 2014). Morphology-driven pipelines such as FilFinder combine adaptive thresholding, morphological cleaning, and medial-axis thinning to produce one-pixel-wide skeletons together with basic morphological descriptors (Koch & Rosolowsky 2015). These approaches provide interpretable topology (lengths, branches), but their results depend on persistence, pruning and the signal-to-noise ratio.

2.3. Template matching and pattern recognition

Nonlocal detectors correlate the image with oriented templates or accumulate orientation evidence locally. Template matching (TM) rotates bar-like kernels over band-pass filtered maps to produce significance and orientation fields from which coherent ridges are traced (Juvela 2016). The rolling Hough transform (RHT) estimates local linear orientations within sliding windows and has been extensively used to map filament alignments with the magnetic field across diverse datasets and scales (Clark et al. 2014, 2015; Clark & Hensley 2019; Jelić et al. 2018; Malinen et al. 2016; Panopoulou et al. 2016; Boe et al. 2020; Ma et al. 2023; Halal et al. 2024). FilDReaMS generalizes RHT to a multi-scale, statistically calibrated framework that jointly infers orientations and prevalent widths while mitigating grid and threshold biases (Carrière et al. 2022b,a; Oers et al. 2023). In position-position-velocity (PPV) cubes, CRISPy performs generalized ridge estimation to recover velocity-coherent filaments, disentangling projection overlaps (Chen et al. 2014, 2015, 2020; Arzoumanian et al. 2022).

2.4. Multiscale statistical filtering

Wavelet-like decompositions separate structures by characteristic size before recombining significant elongated components. `getfilaments` and the unified `getsf` framework decompose images into single-scale maps, apply scale-wise cleaning and

thresholding, and reconstruct filament masks and skeletons jointly with compact source extraction (Men’shchikov 2013, 2021). These pipelines have been validated on synthetic and *Herschel* data and used to quantify widths, contrasts, and source–filament deblending in a variety of environments (Pineda et al. 2016; Louvet et al. 2022; Motte et al. 2022; Cox et al. 2016; Rivera-Ingraham et al. 2016). Complementary anisotropic wavelet analyses provide scale–orientation spectra to quantify filament anisotropy and alignment (Ossenkopf-Okada & Stepanov 2019).

2.5. Machine learning approaches

Supervised convolution neural networks (CNN) and transformer models (e.g., UNet variants, instance segmentation) learn data-driven priors for filament masks and instances, often achieving strong performance on curated datasets; however, they require labeled training data and inherit observational biases (e.g., missing low-contrast filaments) unless complemented by simulation-based augmentation or physics-informed constraints, as shown in Sect. 5.2. Machine learning approaches can further be split into two categories: semantic segmentation (see Sect. 2.5.1) and instance segmentation (see Sect. 2.5.2).

2.5.1. Semantic segmentation

Semantic segmentation refers to the task of assigning a class label to each pixel in an image, thereby enabling a fine-grained understanding of spatial structures (Long et al. 2015). It has become a cornerstone task in computer vision, with widespread applications in autonomous driving, medical imaging, and satellite analysis. The advent of deep learning has led to significant advances in segmentation performance, with early convolutional architectures such as fully convolutional networks (FCNs; Long et al. 2015) followed by encoder–decoder structures such as UNet (Ronneberger et al. 2015) (and its many variations such as UNet++ (Zhou et al. 2018), Attention-UNet (Oktay et al. 2018), ResUNet (Zhang et al. 2018) and UNet3+ (Huang et al. 2020)), DeepLab (Chen et al. 2017), and more recently Vision Transformers (ViTs) adapted for dense prediction tasks (Chen et al. 2021; Xie et al. 2021; Cao et al. 2022). These architectures leverage hierarchical feature extraction, skip connections, and multi-scale attention to achieve high accuracy in various semantic segmentation benchmarks.

In astrophysics, semantic segmentation has been increasingly adopted for identifying structures such as galaxies (Domínguez Sánchez et al. 2018), star-forming regions (Vafaei Sadr et al. 2022), and interstellar filaments (Alina et al. 2022; Xu et al. 2023; Zavagno et al. 2023; Berthelot et al. 2024) where the UNet architecture (or one of its extensions) is predominantly used. However, a key limitation of semantic segmentation lies in its pixel-wise formulation, which does not explicitly capture object-level structure. In the context of astrophysics, this is particularly problematic: due to the 2D projection of inherently 3D environments and because of filament merging, multiple filaments may overlap along the line of sight, and segmentation models – lacking explicit disentangling mechanisms – struggle to distinguish such mixed structures.

2.5.2. Instance segmentation

To overcome the limitations of semantic segmentation in representing discrete and potentially overlapping structures, instance segmentation has emerged as a more expressive alternative.

Instead of assigning a single label per class, instance segmentation distinguishes and separates individual objects of the same category, generating a distinct mask for each occurrence. Classical instance segmentation approaches can broadly be categorized into two families: two-stage and one-stage methods. Two-stage methods, initiated by the R-CNN framework (Girshick et al. 2014), first generate region proposals and then perform pixel-level classification and segmentation within each region. This line of work has led to successive improvements, including Fast R-CNN (Girshick 2015), Faster R-CNN (Ren et al. 2015), and the widely adopted Mask-RCNN (He et al. 2017). In contrast, one-stage methods prioritize speed and real-time performance by eliminating the proposal stage. The most notable example is the YOLO family (Redmon et al. 2016), which has evolved through numerous versions; see Jegham et al. (2024) for a comprehensive review. More recently, transformer-based architectures such as Mask2Former (Cheng et al. 2022) and MaskFormer (Cheng et al. 2021) have introduced unified frameworks that seamlessly integrate object detection and segmentation, offering state-of-the-art performance in complex and cluttered scenes.

In astrophysics, instance segmentation remains underexplored compared to semantic methods. These approaches offer promising avenues for studying filament morphology, orientation, and physical parameters, especially in crowded regions where traditional skeletonization fails. The lack of instance segmentation use is mainly due to the lack of large-scale annotated datasets with filament-level instance labels, which has hindered the broader adoption of instance segmentation in astrophysics.

3. Filament model to generate a synthetic benchmark dataset for filament identification

In this section we detail the model used to represent the filaments in our synthetic dataset that defines the scope of this study and the experimental results' validity. We start by presenting the Plummer radial profile model in Sect. 3.1 then we give a formal definition of what a filament is in Sect. 3.2. Details about the dataset generation procedure can be found in Appendix A.

3.1. Plummer radial profile

The radial structure of interstellar filaments, as revealed by *Herschel* observations of nearby molecular clouds, is well modeled using a Plummer-like profile (Arzoumanian et al. 2011; André et al. 2014). This functional form offers a physically and theoretically (Ostriker 1964; Smith et al. 2014; Myers 2017) motivated description of the transition from a central flat-density core to a power-law decline in the outer regions. It provides a convenient and flexible framework to characterize the observed filamentary structures and to compare them across different environments (Arzoumanian et al. 2011; André et al. 2014).

The Plummer-like density profile is given by

$$N(r) = n_0 \left[1 + \left(\frac{r}{R_{\text{flat}}} \right)^2 \right]^{-(p-1)/2}, \quad (1)$$

where $N(r)$ is the column density at a radial distance r , from the filament's central axis, n_0 is the central density, R_{flat} is the characteristic radius of the inner flat region, and p is the power-law exponent governing the slope of the profile at large radii. This form ensures a central plateau followed by a gradual steepening of the profile, and asymptotically becomes a power law, $N(r) \propto r^{-p}$ for $r \gg R_{\text{flat}}$. A few examples of

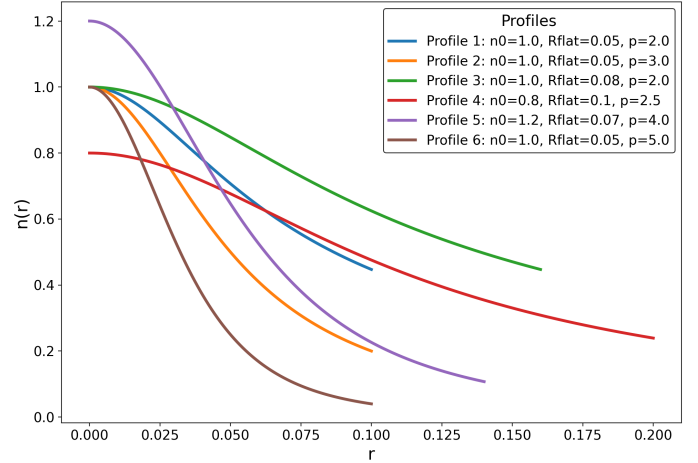


Fig. 1. Panels of randomly generated Plummer profiles and their associated parameters.

Plummer radial profiles can be found in the right column of Fig. 1. This model has been widely applied to filaments in regions such as IC 5146, Aquila, and Polaris (Arzoumanian et al. 2011). The fit parameters typically indicate power-law indices in the range of $p \sim 1-3$ and flat inner radii of $R_{\text{flat}} \sim 0.03-0.25$ pc (Arzoumanian et al. 2019). In more recent work, Mannfors et al. (2025) want to assess how a higher angular resolution impacts Plummer-profile fits of massive filaments and find that the measured width (FWHM) largely tracks resolution: nearby filaments appear at ~ 0.10 pc with *Herschel* alone but contract to ~ 0.05 pc in the combined *Herschel*+ArTéMiS maps, while at larger distances the apparent width inflates with the beam. Following this work, André et al. (2025) use JWST/MIRI absorption imaging to resolve the inner profiles and demonstrate convergence of the width to a characteristic ~ 0.1 pc scale once adequately resolved. Their multi-resolution consistency check (Spitzer, ArTéMiS, *Herschel*, JWST) indicates that the width is not a mere beam artifact, thereby clarifying the regime in which FWHM is a robust physical quantity rather than a resolution-driven parameter.

3.2. Filament definition

In our dataset, each synthetic filament is defined by a parametric representation that captures both its geometry and radial density structure. The geometric backbone of the filament is specified by a one-pixel-wide skeleton, represented as a connected set of discrete coordinates tracing the filament's central axis. The radial structure of the filament around this skeleton is governed by a set of Plummer-like profile parameters: the central density, n_0 , the flat radius, R_{flat} , and the power-law index, p . These parameters control the radial density profile as described in Sect. 3.1, and are treated as ground truth values for each individual filament instance.

The filament's physical properties can be directly retrieved from the Plummer parameters. We defined the contrast of a filament as the ratio between its central column density, n_0 , and the background level evaluated at the edge of the filament mask. Using the Plummer profile defined in Eq. (1) and adopting the mask definition from Sect. 3.1 (radius extending to $2R_{\text{flat}}$), the background level at the mask boundary is $N(2R_{\text{flat}}) = n_0 5^{-(p-1)/2}$. Hence the contrast is

$$\text{contrast}_{(n_0, R_{\text{flat}}, p)} = 5^{\frac{p-1}{2}}. \quad (2)$$

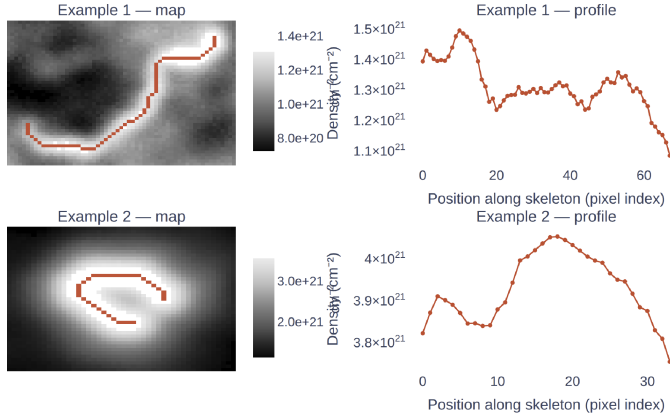


Fig. 2. Panels of randomly generated skeletons based on pseudo-random walk (on the left) and their column density value along the skeleton (on the right).

This expression depends only on the slope parameter, p (the factors n_0 and R_{flat} cancel), increasing monotonically with p . For this reason, we refer to low-contrast filaments as low- p filaments.

To construct a binary mask associated with each filament, used for segmentation supervision, we defined the filament region as the set of all pixels whose Euclidean distance to the skeleton is less than or equal to $2R_{\text{flat}}$ (Arzoumanian et al. 2019). This radius-dependent criterion ensures that the mask encompasses both the flat inner region and a significant portion of the surrounding power-law tail. Formally, for a skeleton, $\mathcal{S}$, and a given flat radius, R , the filament mask, $\mathcal{M}$, is defined as

$$\mathcal{M} = \left\{ \mathbf{x} \in \Omega \mid \min_{\mathbf{s} \in \mathcal{S}} \|\mathbf{x} - \mathbf{s}\|_2 \leq 2R_{\text{flat}} \right\}, \quad (3)$$

where Ω denotes the spatial domain of the image. This construction yields a mask that is consistent with the underlying physical scale of the filament and can adapt to varying widths across the dataset.

To better match survey observations, the central column density, n_0 , along each filament’s skeleton is not fixed but follows a smooth random profile, with absolute variations capped at 25%. Examples of such longitudinal n_0 variations are shown in the right column of Fig. 2.

3.3. Dataset generation

The synthetic dataset was generated through a three-stage pipeline designed to produce physically plausible filamentary structures embedded in realistic diffuse backgrounds. First, each filament was defined by a smooth, one-pixel-wide skeleton generated with a constrained random walk. This procedure yielded single connected and unbranched structures with variable lengths and limited curvature, providing a diverse set of morphologies while preserving the visual properties of observed interstellar filaments. Second, a filament density map was constructed around each skeleton using a Plummer-like radial profile parameterized by the triplet $(n_0, R_{\text{flat}}, p)$. For computational efficiency, the density assigned to each pixel was computed from its Euclidean distance to the nearest skeleton point, rather than from explicit orthogonal cuts along the filament axis. This approximation preserves the essential radial behavior of the profile, while ensuring full image coverage and efficient large-scale generation. Third, the filament was combined with a synthetic diffuse background intended to better reproduce the complexity of real

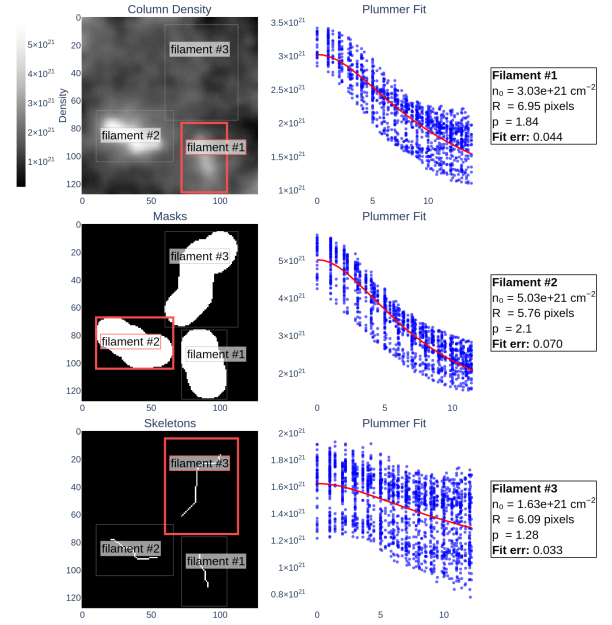


Fig. 3. *Top left:* example of a synthetic column-density map. *Middle left:* corresponding ground-truth filament mask. *Bottom left:* Ground-truth skeleton (a one-pixel-wide crest), from which the mask was derived using Eq. (3). *Right panels:* radial column-density profile, $N(r)$, as a function of the Euclidean distance, r , to the skeleton (blue), with the Plummer model, $N_p(r)$, evaluated at the ground-truth parameters (n_0, R_{flat}, p) overlaid in red.

interstellar environments. The background was generated from a smoothed version of the filament map together with Gaussian-filtered noise, followed by a nonlinear intensity transformation and rescaling. This produces spatially structured backgrounds correlated with the filament morphology, yielding more realistic maps than simple parametric background models. More details about the dataset generation method can be found in Appendix A.1.

3.4. Filament and dataset properties

The generated dataset consists of 50 000 synthetic column density maps 128×128 pixels in size for training, and 10 000 maps for evaluation with a theoretical resolution of $18''.2$. Some examples of column density maps and their corresponding mask maps can be found in Fig. 3. Each map contains between two and five filaments, with the number drawn uniformly at random, with no overlapping filaments. Histograms of the benchmark dataset parameters are shown in Fig. 4, and further details on the Plummer parameter estimation are provided in Appendix A.2. The parameter distributions are drawn from previous observational works (Arzoumanian et al. 2011; André et al. 2014; Arzoumanian et al. 2019), except for the p parameter, where we used the distribution from Arzoumanian et al. (2019) while adding a Gaussian distribution centered at 1.2 to include low-contrast filaments.

4. Methodology to compare filament detection methods

In this section, we first describe the filament detection methods evaluated in the study, followed by a presentation of the metrics used to assess their performance. Lastly, we present the

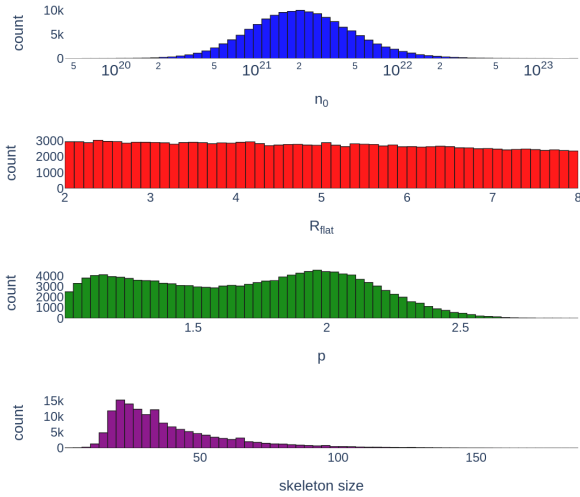


Fig. 4. Distributions of filament fit properties in the generated train dataset. *Top to bottom:* n_0 , R_{flat} , p , and skeleton lengths.

experimental setup used for the automatic parameter-selection protocol.

4.1. Filament detection methods' description

In this section, we introduce the filament detection methods evaluated on our benchmark dataset. As a representative of semantic segmentation approaches, we included the widely used UNet architecture. For instance segmentation, we adopted the Mask-RCNN framework. Finally, we incorporated FilFinder as a morphology-based method to study how a non-learning algorithm behaves when its parameters are selected automatically through a global validation-set grid search.

4.1.1. UNet

The UNet architecture (Ronneberger et al. 2015) is a neural network designed for semantic segmentation, structured similarly to an autoencoder. While a classical autoencoder is trained to reconstruct its input through a compressed latent representation, UNet is trained to produce a segmentation map from the input image. It achieves this through an encoder-decoder structure: the encoder progressively reduces spatial resolution via convolution and pooling layers to extract high-level features, while the decoder restores the original resolution using convolution and transposed convolution layers.

What distinguishes UNet from standard autoencoders is the use of skip connections, which link corresponding layers of the encoder and decoder. These connections enable the network to retain and reuse fine-grained spatial information from early layers during decoding, facilitating precise localization and boundary reconstruction. The final layer typically consists of a 1×1 convolution to produce a per-pixel class prediction map. The UNet architecture has proven to perform well on the filament detection task, as highlighted in Zavagno et al. (2023); Berthelot et al. (2024).

4.1.2. Mask-RCNN

Mask-RCNN (He et al. 2017) is a widely used deep learning architecture for instance segmentation, which extends Faster R-CNN (Ren et al. 2015) by adding a parallel branch for pixel-wise

object mask prediction. It follows a two-stage pipeline: in the first stage, a region proposal network (RPN) scans the image to identify candidate object regions (regions of interest, or RoIs); in the second stage, each proposed region is classified and refined via bounding box regression, while a third branch simultaneously predicts a binary mask for the object within the region.

The architecture builds on a convolutional backbone (ResNet50; He et al. 2016) to extract shared feature maps, which are then used by the RPN to generate proposals. For each RoI, features are extracted using RoIAlign – a bilinear interpolation technique that preserves spatial alignment – and fed into the classification, bounding box, and mask heads. The mask head consists of a small FCN that outputs a segmentation mask of fixed size for each detected object.

Mask-RCNN combines the strengths of object detection and semantic segmentation into a unified framework capable of identifying and separating multiple overlapping instances of the same class. This makes it particularly suitable for tasks where individual object delineation is crucial, such as filament detection in dense and crowded astronomical fields. The Mask-RCNN architecture has been applied with success to filament detection in Alina et al. (2022).

4.1.3. FilFinder

FilFinder is a non-machine-learning algorithm developed to identify filamentary structures in astronomical images by leveraging morphological and topological operations (Koch & Rosolowsky 2015). It operates on 2D intensity maps and detects filaments as elongated, connected regions through a combination of thresholding, adaptive masking, and medial axis skeletonization.

The algorithm begins with an adaptive thresholding step that isolates bright, filamentary regions by computing local intensity thresholds across subregions of the image, allowing it to adapt to varying background levels and highlight locally significant structures that may be missed by global thresholding methods. These regions are then refined using a series of morphological operations (such as closing and dilation) to remove noise and connect fragmented components. Unlike deep learning methods, FilFinder is fully deterministic and parameter-driven, making it interpretable and effective for qualitative and quantitative structural analysis of relatively clean and high-contrast images. The hyper-parameter grid-search setup can be found in Appendix B.

4.2. Metrics

For semantic segmentation (pixel-level) performance evaluation, we used standard classification metrics such as the Dice similarity coefficient and area under the curve Receiver Operating Characteristic (AUC) ROC (see below). In addition to those, we used instance segmentation metrics as implemented in the cocoEval evaluation suite (COCO Consortium 2017; Bolya et al. 2019; Lin et al. 2014) to attest instance level performance.

AUC ROC. The receiver operating characteristic (ROC) curve summarizes a classifier's behavior across all decision thresholds. First, we defined the pixel-level detection outcomes. Let Ω be the set of image pixels, $P \subset \Omega$ the binary prediction mask, and $G \subset \Omega$ the ground-truth mask. A **true positive (TP)** is any pixel in both P and G (filament correctly predicted), $TP = |P \cap G|$. A **false positive (FP)** is any pixel in P but not in G (background mislabeled as filament), $FP = |P \setminus G|$. A **false negative (FN)** is

any pixel in G but not in P (missed filament), $FN = |G \setminus P|$. A **true negative (TN)** is any pixel in neither P nor G (background correctly predicted), $TN = |\Omega \setminus (P \cup G)|$.

Area under the curve Receiver Operating Characteristic (AUC ROC) is the parametric curve $\{(FPR(\tau), TPR(\tau)) : \tau \in [0, 1]\}$, where for a given threshold, τ ,

$$TPR = \frac{TP}{TP + FN} \quad (\text{Recall}), \quad (4)$$

$$FPR = \frac{FP}{FP + TN}. \quad (5)$$

The AUC ROC compresses the ROC into a single scalar and admits a ranking interpretation: it is the probability that the model assigns a higher score to a randomly chosen positive (filament) than to a randomly chosen negative (background). Scores range from 0 to 1, with 1 indicating perfect separability, 0.5 a random classifier, and values closer to 0 reflecting inverted rankings.

Dice similarity coefficient (DSC). The Dice coefficient is a widely used metric for evaluating the overlap between predicted and ground-truth segmentation masks. It is defined as

$$DSC = \frac{2TP}{2TP + FP + FN}, \quad (6)$$

where TP denotes the number of true positives (pixels correctly classified as filaments), FP the number of false positives (background pixels incorrectly classified as filaments), and FN the number of false negatives (filament pixels incorrectly classified as background). The DSC score ranges from 0 (no overlap) to 1 (perfect detection).

COCO-style average precision (AP, AP50, AP75). The COCO protocol (COCO Consortium 2017; Bolya et al. 2019; Lin et al. 2014) measures instance-segmentation quality by averaging precision over multiple intersection-over-union (IoU) thresholds. For a fixed IoU threshold, τ , detections are ranked by confidence and matched to ground truth; let P_n and R_n denote precision and recall after the n -th ranked detection. The average precision (AP) at τ is the area under the precision–recall curve,

$$AP(\tau) = \sum_n (R_n - R_{n-1}) P_n. \quad (7)$$

The overall COCO AP is the mean of $AP(\tau)$ computed at IoU thresholds $\tau \in \{0.50, 0.55, \dots, 0.95\}$:

$$AP = \frac{1}{10} \sum_{\tau=0.50}^{0.95 \text{ step } 0.05} AP(\tau). \quad (8)$$

Two commonly reported single-threshold variants are:

- **AP50**: $AP(0.50)$, favoring coarse localization.
- **AP75**: $AP(0.75)$, emphasizing stricter spatial alignment.

These metrics are standard for benchmarking instance-segmentation models on datasets such as COCO. They all range from 0 to 1: 0 means no successful detections at the specified IoU threshold (area under the precision–recall curve is zero), while 1 indicates perfect performance (precision and recall equal to 1 across recall).

Metric computation. Because UNet produces pixel-wise (semantic) masks rather than discrete instances, we used a watershed method (Vincent & Soille 1991; Meyer 1994; Soille 2003) to obtain instance-level outputs from pixel-segmentation. It has been widely used and was popularized in computer vision by Bai & Urtasun (2017) and tools such as *ilastik* for cell and nuclei segmentation Berg et al. (2019). By design, *FilFinder* returns a single thresholded binary mask (no per-instance masks or confidence scores). Instance AP-style metrics require confidence-ranked predictions and are highly threshold-dependent, making them ill-posed for *FilFinder*; accordingly, we do not report instance-level results for this method. For a cross-family comparison at the pixel level, we additionally computed pixel-based scores for Mask-RCNN by binarizing the logical union of its predicted instances.

Note that pixel-level metrics can be meaningfully compared across models; however, comparisons based on instance-level metrics are not valid. First, the APs of Mask R-CNN are penalized when multiple overlapping proposals correspond to the same ground-truth filament: only the highest-IoU proposal is counted as a true positive, while the others are treated as false positives, thereby reducing the APs. Second, converting UNet outputs into instances through thresholding and watershed segmentation typically yields a single detection per object, which avoids duplicate predictions and can artificially inflate APs compared to a native instance segmentation model.

In summary, instance-level metrics reported for UNet are provided for qualitative insight only, and comparisons at the instance level should not be performed. Proper instance-level evaluation requires a dedicated instance segmentation architecture, such as Mask R-CNN.

4.3. Experimental setup

Neural networks. All experiments were conducted using a five-fold cross-validation scheme to ensure the robustness and generalization of our results. For each fold, the grid search was performed over the learning rate ($1e^{-2}$, $1e^{-3}$, $1e^{-4}$, $1e^{-5}$), and the optimal value was selected based on the highest Dice similarity coefficient obtained on the validation set.

We used a batch size of 256 for all models. The optimization was performed using the Adam optimizer with weight decay of $1e^{-5}$ for regularization. A learning rate scheduler was applied, reducing the learning rate by a factor of 10 every ten epochs to ease convergence. All experiments were run on three NVIDIA A40 GPUs (computation times are discussed in Appendix F).

FilFinder. *FilFinder* is a morphology-based filament identification method designed to extract connected filamentary networks from astronomical maps through intensity flattening, global and adaptive filtering, morphological cleaning, and skeletonization. In its usual application, the relevant parameters are selected by the user in light of the map properties, scientific target, and expected filament population. In this work, we intentionally ask a different question: whether a single parameter set can be selected automatically from a training-validation split and then applied unchanged to a held-out test set, analogously to the way neural-network hyperparameters are selected. We therefore adopted the same protocol used for neural networks: a global *FilFinder* hyperparameter set was selected by grid search on the training-validation data, using the DSC as the selection criterion, and the chosen configuration was then frozen and applied to the test set. Grid-search ranges and settings

Table 1. Comparative results of segmentation models.

Model	AUC ROC	DSC	AP	AP75	AP50
UNet	0.990 ± 0.000	0.926 ± 0.001	0.798 ± 0.004	0.819 ± 0.002	0.760 ± 0.004
Mask-RCNN	0.958 ± 0.003	0.739 ± 0.010	0.708 ± 0.009	0.848 ± 0.003	0.756 ± 0.010
FilFinder	0.61	0.40	–	–	–

Notes. Comparative results across complementary metrics. Pixel-level metrics (AUC–ROC, DSC) evaluate dense classification, while instance-level metrics (AP, AP50, AP75) assess detection quality across IoU thresholds. Displayed values are mean scores across the five folds with their respective standard deviation. For cross-family comparison, pixel-level scores shown for Mask-RCNN and instance-level scores shown for UNet are obtained via post-processing (union binarization and instance separation, respectively); these approximations are not strictly comparable to each model’s native output type (see Sect. 4.2). Because FilFinder produces binary results, we do not display instance-level metrics, which are threshold-dependent.

are detailed in Appendix B. This procedure prevents test leakage and provides a reproducible survey-scale parameter-tuning experiment. It should not be interpreted as replacing the expert, map-specific tuning for which classical methods are commonly used.

4.4. Normalization

Normalization is commonly applied to neural networks’ inputs to improve optimization convergence, generalization, and robustness to outliers. In our case, however, normalization of synthetic maps requires special care. We generate filaments from a parameterized model (see Sect. 3.1), and these parameters directly encode the filament’s physical properties. Consequently, normalization should not alter these parameters – or, at a minimum, it should allow us to recover the original values from those fit on the normalized maps.

To meet this requirement, we adopted a simple rescaling by the map’s maximum column density. Using Eq. (1), the normalized column density is

$$N_{\text{norm}}(r) = \frac{N_p(r)}{n_{\text{max}}} = n'_0 \left[1 + \left(\frac{r}{R_{\text{flat}}} \right)^2 \right]^{-(p-1)/2}, \quad (9)$$

where n_{max} is the maximum column density in the map and $n'_0 = n_0/n_{\text{max}}$. This rescaling leaves R_{flat} and p unchanged, while the new amplitude, n'_0 , is trivial to compute (and the original n_0 is recovered via $n_0 = n_{\text{max}} \times n'_0$).

5. Results and discussion

In this section we present the experimental results obtained with the benchmark dataset introduced in Sect. 3. We begin by reporting the outcome of the common automatic parameter-selection protocol in Sect. 5.1. We then identify failure regimes of the learning-based detectors in Sect. 5.2 and, motivated by these findings, investigate in Sect. 5.3 how training on incomplete annotations biases learning, focusing on low-contrast and low-density filaments. Qualitative results about the computation time can be found in Appendix F.

5.1. Benchmark of detection methods

To evaluate the behavior of different filament detection approaches under a common, reproducible protocol, we conducted a benchmark on our synthetic dataset with a known ground truth. The central purpose of this comparison is to study

how each approach scales when its parameters are selected automatically from training-validation data and then fixed on the test set.

We report the results obtained under the evaluation protocol described in Sect. 4.3 and with the metrics described in Sect. 4.2. Table 1 summarizes the average performance across five cross-validation folds. Column 1 lists the methods. Columns 2 to 5 provide the average values of the pixel-level metrics: AUC ROC and DSC. Columns 6 to 8 report instance-level detection metrics: AP, AP50, and AP75, capturing the performance across varying IoU thresholds.

We emphasize once again (see Sect. 4.2 for details) that a direct comparison between UNet and Mask-RCNN cannot be made using instance-level metrics. The instance-level results reported for UNet are provided only as indicative values, obtained through post-processing steps as described in Sect. 4.2. A proper instance-level evaluation can only be achieved using a dedicated instance segmentation model such as Mask-RCNN.

Under this automatic global parameter-selection protocol, UNet obtains the strongest pixel-level performance across our benchmark, while Mask-RCNN remains competitive and yields a good performance on instance-level metrics. The FilFinder configuration selected by grid search produces lower aggregate pixel-level scores on the full heterogeneous test set. It shows that a single global configuration, selected once from a training-validation set, does not transfer as effectively across our synthetic maps as the learned parameters of neural networks (see Sect. 4.3).

A visual inspection of UNet and Mask-RCNN outputs reveals complementary failure modes (Fig. 5). UNet exhibits under-segmentation: it misses faint, low-contrast, and low-density filaments, yielding many false negatives and depressing Recall and DSC. In contrast, Mask-RCNN tends toward over-segmentation: it frequently labels bright background patches and ridges as filaments – even when their radial profiles are not Plummer-like – thereby inflating false positives and reducing both pixel- and instance-level metrics (we explore whether unmatched predictions from Mask-RCNN are genuine false positives or not in Appendix E). As a consequence, Mask-RCNN also tends to accrete background around genuinely faint or low-density filaments, further degrading instance-level and pixel-level scores. Although both behaviors lower the metrics, they stem from different causes: UNet’s pixel-wise decision rule is conservative in low-signal-to-noise regimes, whereas Mask-RCNN’s proposal-and-scoring pipeline can hallucinate coherent instances in high-intensity backgrounds and occasionally fragment long filaments into overlapping detections.

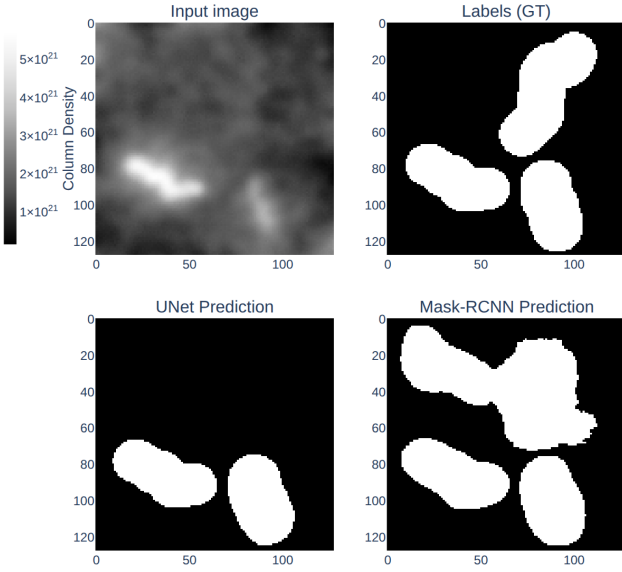


Fig. 5. *Top left:* input column-density map (before normalization; see Sect. 4.4). *Top right:* corresponding ground-truth segmentation. *Bottom left:* UNet prediction. *Bottom right:* Mask-RCNN output.

5.2. Performance in regard to Plummer parameters

Motivated by our visual inspection of the benchmark outputs – in which both UNet and Mask-RCNN systematically struggle on faint structures – we explicitly targeted two challenging regimes: low-contrast filaments or low-density filaments. We quantified the low-contrast filaments with the Plummer slope, p (shallower profiles, i.e., lower p , yielding a weaker contrast at larger distance to the skeleton; see Sect. 3.2), and we quantified the low-density filaments with the density proxy $\chi = n_0/n_{\max}$, where $n_{\max}$ is the image’s maximum column density. These parameters delineate filament populations that are common in observations yet frequently missed by standard pipelines (Arzoumanian et al. 2019; Mannfors et al. 2025): low-contrast and low-density filaments, respectively. Guided by this observation, we stratified results by p and by χ to assess each model’s sensitivity precisely in these difficult regimes. Operationally, for each ground-truth filament we selected the single predicted mask with the highest IoU and computed pairwise scores (e.g., IoU, Dice, AP hits at fixed IoU thresholds) for that match; if no overlap was found, the filament contributed as a miss (zero score). We then studied these per-pair measurements by the Plummer slope, p , and by the density proxy, χ , to obtain the performance as a function of the filament properties. Note that this conditioning on best matches downplays the impact of false positives (predictions with no corresponding ground truth), whereas the global benchmark in Sect. 5.1 penalizes such errors, and therefore yields lower, more conservative aggregate scores. We report results only for the Mask-RCNN in this section. UNet results can be found in Appendix D. We believe that the metrics alone are of limited interest. More important is understanding how the method’s behavior and performance vary as a function of two key factors: contrast, parameterized by p , and density scale, parameterized by χ .

5.2.1. Performance versus p

One of the core properties of filaments is their contrast, defined as the ratio between the central column density, n_0 , and the background level. Given the Plummer radial profile, the contrast

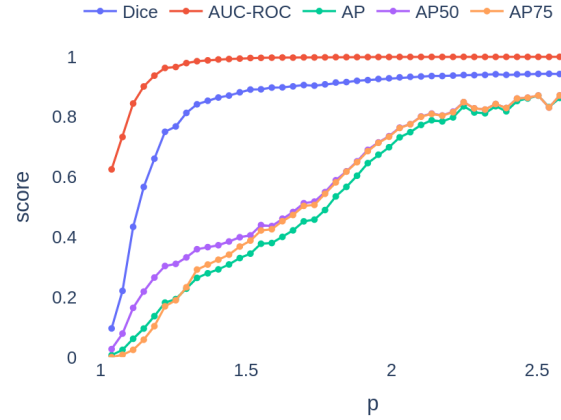


Fig. 6. Mask-RCNN’s performance as a function of Plummer power index p : AUC ROC, DSC, AP, AP50, and AP75 reported per p bin.

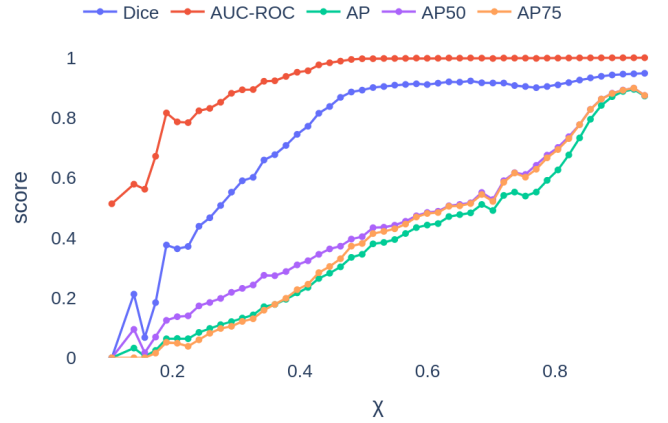


Fig. 7. Mask-RCNN’s performance versus the density proxy χ : AUC ROC, DSC, AP, AP50, and AP75 are reported for each χ bin.

is solely defined by p (see Sect. 3.2 for details). Because the contrast plays a central role in the filament evolution, we propose to study the filament detection performance in regard to their contrast.

Figure 6 presents Mask-RCNN performance as a function of p across multiple metrics. While the scores remain comparatively stable for $p > 1.5$, all metrics degrade sharply as $p \rightarrow 1$ (the low-contrast regime), consistent with the qualitative failure modes identified in Sect. 5.1. It appears that, as p tends to 1, performance drastically drops. For pixel-level metrics, the critical point appears to be around $p = 1.3$, while for instance-level metrics, it is $p = 2$.

5.2.2. Performance versus $(n_0/n_{\max})$

We also considered a complementary density-based proxy, χ , defined at the image level. By construction, $\chi \in [0, 1]$ ranks filaments by relative central density, independently of their Plummer shape parameters; in particular, χ is agnostic to R_{flat} and p . Low values of χ therefore identify low-density filaments that are likely to exhibit a poor filament-to-background ratio (on the image level) once a structured background is added, even if their intrinsic Plummer slopes are not especially shallow, due to the normalization (see Sect. 4.4).

Figure 7 presents Mask-RCNN performance as a function of χ across multiple metrics. While scores remain comparatively

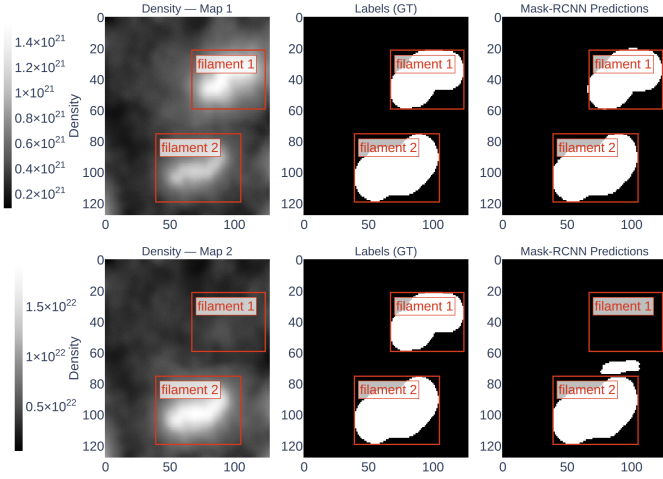


Fig. 8. Two synthetic filament maps with matched geometry. The columns show (left to right): the density map, ground-truth annotation, and Mask-RCNN prediction. The rows correspond to Map 1 (top) and Map 2 (bottom). Red rectangles indicate ground-truth bounding boxes. Each density panel displays its color bar on the left; predictions are shown as masks.

stable for moderate-to-high ($\chi > 0.6$), all metrics degrade as χ decreases (i.e., as filaments become progressively fainter relative to the brightest structures in the image), consistent with the qualitative failure modes identified in Sect. 5.1. Figure 7 indicates that Mask-RCNN performance depends not only on a filament’s intrinsic parameters but also on the content of the entire map. Because filaments are not isolated, the properties of other filaments modulate detectability.

This is illustrated in Fig. 8. Filament 1 has comparable Plummer parameters in both Map 1 ($n_0 = 1.5 \times 10^{21}$, $R = 6$, $p = 1.5$) and Map 2 ($n_0 = 4.5 \times 10^{21}$, $R = 6$, $p = 1.8$). By contrast, Filament 2 changes markedly between Map 1 ($n_0 = 1.1 \times 10^{21}$, $R = 6$, $p = 1.8$) and Map 2 ($n_0 = 1.8 \times 10^{22}$, $R = 6$, $p = 1.9$), especially in n_0 . The large-scale background morphology is identical in both maps (with intensities scaled to the filaments’ densities). As is shown in the third column, this difference has a strong impact on the Mask-RCNN: in Map 1, both filaments are recovered, whereas in Map 2, Filament 1 is missed entirely. These examples highlight that detection is driven not only by a filament’s own parameters but also by its relative context within the map.

5.3. Study of biases across Plummer parameters

In previous studies (Alina et al. 2022; Zavagno et al. 2023; Berthelot et al. 2024), machine-learning models were trained and evaluated on observational data where the ground truth was incomplete – some filaments, particularly faint ones, were not annotated, and thus absent from both the training and testing sets. We propose to study two distinct biases: the low-contrast bias and the low-density bias. To obtain instance-level metric insights about the UNet (without requiring post-processing steps which may bias results), we constructed an auxiliary dataset drawn from the same parameter distributions as our main benchmark but containing exactly one filament per image. In this single-instance setting, the semantic output coincides with a unique object, allowing us to derive instance-level results without ambiguity from overlaps or mergers. Minor adjustments to the generation pipeline were required to robustly probe the low-density regime; these modifications are described in Appendix C. Given

Table 2. Mask-RCNN’s performance analysis in regard to p bias.

Dataset	AUC ROC	DSC	AP	AP50	AP75
Biased	0.87	0.74	0.61	0.67	0.65
Unbiased	0.93	0.30	0.73	0.83	0.77

Notes. Column 1 indicates which training set (biased vs. unbiased) was used for Mask-RCNN; Columns 2 and 3 report pixel-level metrics (AUC ROC and DSC), and Columns 4–6 list the instance-level metrics (AP, AP50, and AP75). In the p -biased training set, only high-contrast filaments (i.e., $p > 1.5$) are annotated as positives.

the low variability observed across folds in the cross-validation benchmark, the following experiments were conducted without performing a grid search or cross-validation (see Sect. 5.1). Instead, we adopted a learning rate of 1×10^{-4} . Once again, we report results only for Mask-RCNN, while the results for UNet can be found in Appendix D.

5.3.1. Low-contrast bias

The missed filaments by neural networks are typically low-contrast, meaning their column density is only marginally higher than that of the surrounding background, making them harder to detect. From a physical standpoint, and referring to the Plummer function defined in Eq. (1), such filaments correspond to low values of the power-law index, p , close to but greater than 1 (as detailed in Sect. 5.2.1). As a result, current detection neural networks are biased against these low-contrast filaments, since they are underrepresented – or entirely absent – in existing datasets. We refer to this phenomenon as the p bias, as the contrast is governed by the p parameter in the Plummer function. Building on this observation, we aim to assess how this bias affects the detection capabilities of various methods, especially in the presence of filaments with low p .

To quantify the impact of the p bias, we trained Mask-RCNN on two training sets that differ only in their annotations. (i) A p -biased set that mirrors observational practice: low- p (low-contrast, $p < 1.5$) filaments are present in the column-density maps but not annotated. (ii) An unbiased set in which all filaments, including low- p structures, are labeled. Evaluation is performed on a single, test set containing a mix of low- and high-contrast filaments to enable a controlled comparison. This experiment highlights a key risk of training on incomplete observational annotations: excluding low-contrast filaments degrades performance precisely in the regime where detection is most needed, underscoring the importance of comprehensive labeling for reliable filament detection.

Mask-RCNN’s results are given in Table 2, where the first column indicates the dataset on which the Mask-RCNN was trained, columns 2 and 3 give the mean value for pixel-level metrics (AUC ROC and DSC), and columns 4 to 6 list instance-level metrics (AP, AP50, and AP75). One can note a drop in the performance of every metric (except DSC) between the Mask-RCNN trained on the biased dataset compared to the one trained on the unbiased dataset.

We stratified the Mask-RCNN performance by the Plummer slope, p . Figure 9 reports AUC ROC, DSC, AP, AP50, and AP75 across p bins for models trained on the p -biased (top panel) and p -unbiased sets (bottom panel). One trend emerges: excluding low- p (low-contrast) filaments during training selectively degrades performance in the shallow-profile regime, whereas

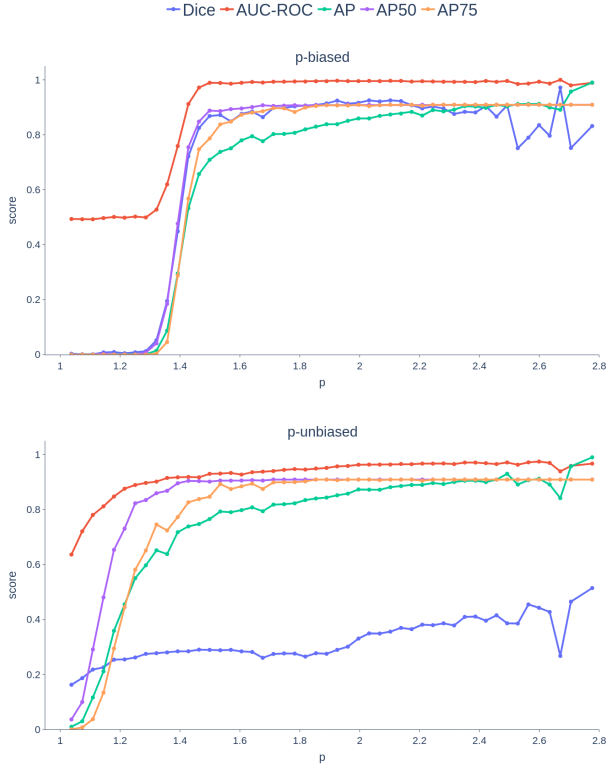


Fig. 9. Mask-RCNN’s performance as a function of Plummer power index, p : AUC ROC, DSC, AP, AP50, and AP75 reported per p bin for Mask-RCNN trained on the biased dataset (*top*) and for Mask-RCNN trained on the unbiased dataset (*bottom*).

unbiased training yields markedly more stable scores across low p .

Table 2 shows that training Mask-RCNN on a p -biased dataset produces a marked degradation in performance across every metric, except DSC, compared to the same architecture trained on an unbiased dataset. This comparison is made on the same test set (containing both low- and high- p filaments), confirming that the bias in training annotations directly harms generalization to low-contrast regimes. We hypothesize that the drop in DSC is primarily due to our thresholding choice. In these experiments, both network outputs were binarized using the same default threshold, rather than a value tuned on the validation set to select each model’s optimal operating point. For this reason, we place less emphasis on the DSC decrease and focus instead on threshold-independent metrics – AUC-ROC and APs – which better reflect overall ranking performance.

From Fig. 9, performances are broadly comparable for $p > 1.5$. In contrast, as we enter the low-contrast regime (lower p), models trained on the p -biased dataset exhibit a pronounced drop across every metric, whereas models trained on the unbiased dataset maintain substantially higher scores.

These results highlight the limitations of models trained on incomplete ground truth. While the model maintains high performance on high-contrast filaments that match the training distribution, its ability to generalize to low- p , low-contrast structures is severely impaired.

5.3.2. Low-density bias

In addition to low-contrast filaments, low-density filaments (compared to their local environment) are frequently overlooked

Table 3. Mask-RCNN’s performance analysis in regard to low-density bias.

Dataset	AUC ROC	DSC	AP	AP50	AP75
Biased	0.74	0.39	0.67	0.79	0.71
Unbiased	0.93	0.30	0.73	0.83	0.77

Notes. Column 1 indicates which training set (density-biased vs. unbiased) was used for Mask-RCNN, Columns 2 and 3 display the pixel-level metrics (AUC ROC and DSC), and Columns 4–6 list the instance-level metrics (AP, AP50, and AP75). In the density-biased training set, only high-density filaments ($\chi = n_0/n_{\max} > 0.5$) are annotated as positives, while low-density filaments ($\chi \leq 0.5$) present in the images remain unlabeled.

by traditional detection algorithms due to the effects of normalization. This normalization, often applied on a per-image basis to enhance contrast and stabilize learning, can inadvertently suppress the visibility of faint structures by rescaling all pixel values relative to the brightest features. As a result, filaments with low column densities (relative to the image) become nearly indistinguishable from the background and are less likely to be detected or included in training datasets.

To quantify the impact of a density bias, we mirrored the p -bias protocol with two training sets and a single shared test set. We defined a density-based density proxy, $\chi = n_0/n_{\max}$ (Sect. 5.3.2), and set the boundary between low- and high-density filaments at $\chi = 0.5$. The density-biased training set annotates only high-density filaments ($\chi > 0.5$); low-density filaments ($\chi \leq 0.5$) remain present in the images but are unlabeled, mimicking incomplete observational catalogs. The unbiased training set annotates all filaments irrespective of χ . Both models are evaluated on the same test set that contains a mixture of low- and high-density filaments.

Mask-RCNN’s results are given by Table 3 where the first column indicates the dataset on which the Mask-RCNN was applied, columns 2 and 3 report pixel-level metrics (AUC ROC, DSC), and columns 4 to 6 list instance-level metrics (AP, AP50, and AP75). We can highlight a drop in the performance of every metric (except DSC) between the Mask-RCNN trained on the biased dataset compared to the one trained on the unbiased dataset.

To pinpoint the source of the gap, we stratified metrics by the density–density proxy, χ . On the test set, Fig. 10 (top panel) shows AUC ROC, DSC, AP, AP50, and AP75 as a function of χ for a Mask R-CNN trained on a density-biased dataset. The bottom panel presents the same analysis for a model trained on an unbiased set. Comparing these curves indicates that omitting low- χ (low-density) filaments during training selectively harms performance in the faint-structure regime, whereas unbiased training maintains substantially higher scores across the full χ dynamic range.

Table 3 shows that training Mask-RCNN on a density-biased dataset – i.e., one that annotates only high-density filaments ($\chi = n_0/n_{\max} > 0.5$) – yields a pronounced degradation in performance relative to the same architecture trained on an unbiased dataset. The comparison was performed on the same test set (containing both low- and high- χ filaments), confirming that bias in the training annotations directly harms generalization to faint, low-density regimes. As discussed in Sect. 5.3.1, the observed drop in DSC likely stems from our thresholding strategy; accordingly, we place greater emphasis on threshold-independent metrics such as AUC-ROC and APs.

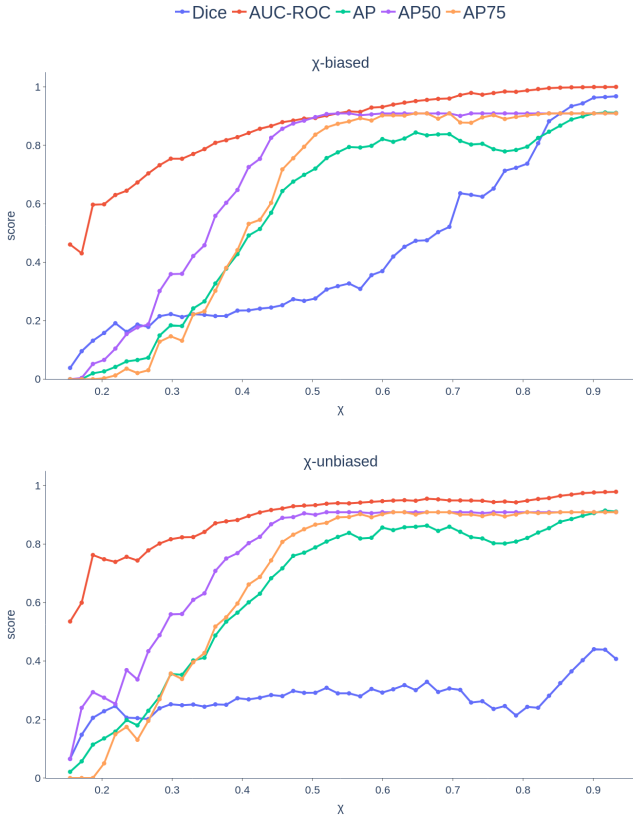


Fig. 10. Mask-RCNN’s performance as a function of $\chi = n_0/n_{\max}$: AUC ROC, DSC, AP, AP50, and AP75 per χ -bin for Mask-RCNN trained on the biased dataset (*top*) and for the Mask-RCNN trained on the unbiased dataset (*bottom*).

From Fig. 10, performance is broadly comparable in the high-density regime (large $\chi = n_0/n_{\max}$, e.g. $\chi > 0.5$). In contrast, as we enter the low-density regime (small χ), the model trained on the density-biased dataset exhibits a pronounced drop across all metrics, whereas the model trained on the unbiased dataset maintains substantially higher scores. This finding underscores a key limitation of training on incomplete ground truth: while the biased model performs well on high-density filaments that mirror its training distribution, its ability to generalize to faint, low- χ structures is markedly degraded.

6. Conclusions

In this work, we introduced a synthetic Plummer-based dataset to study filament detection in a controlled setting with complete ground truth. The dataset was used for two complementary objectives. First, we examined how different filament detection approaches behave under an automatic parameter-selection protocol based on training, validation, and test splits. In this context, UNet and Mask-RCNN were treated as learning-based methods whose parameters are optimized from data, whereas FilFinder was used as a morphology-based method placed in a global grid-search framework. This comparison should therefore be understood as a study of automatic parameter tuning and scalability, not as a definitive ranking of FilFinder against neural networks in the expert-tuned regime for which it was designed. UNet and Mask-RCNN show complementary strengths, while being subject to the same difficulties, in particular not performing as well on low-density and -contrast filaments. On the

other hand, FilFinder proves to not be fit for an automatic hyper-parameter selection.

Second, we used the controlled dataset to quantify how incomplete training annotations affect learning-based filament detection. By introducing biases with respect to contrast and density, we show that models trained on incomplete labels suffer significant degradation when confronted with underrepresented filament types. It demonstrates that observational catalogs missing faint or low-contrast structures can induce systematic blind spots in neural networks, even when those networks perform well on the annotated part of the distribution.

Overall, our results advocate for using synthetic, physics-aware data as a diagnostic tool for both parameter-selection strategies and training-set biases. Future work will extend the benchmark to 3D simulations, incorporate additional physical effects such as temperature variations and projection complexities, and consider additional non-learning filament-identification methods, including approaches with a more explicit treatment of background emission. We hope that the dataset and benchmarking framework introduced here will serve as a foundation for developing filament detection methods that are robust, interpretable, and transferable across instruments and observing conditions.

References

- Alina, D., Shomanov, A., & Baimukhametova, S. 2022, *IEEE Access*, **10**, 74472
- André, P., Men’schikov, A., Bontemps, S., et al. 2010, *A&A*, **518**, L102
- André, P., Di Francesco, J., Ward-Thompson, D., et al. 2014, Protostars and Planets VI [arXiv:1312.6232]
- André, P., Di Francesco, J., Ward-Thompson, D., et al. 2014, in *Protostars and Planets VI*, eds. H. Beuther, R. S. Klessen, C. P. Dullemond, & T. Henning, 27
- André, P., Arzoumanian, D., Könyves, V., Shimajiri, Y., & Palmeirim, P. 2019, *A&A*, **629**, L4
- André, P., Mattern, M., Arzoumanian, D., et al. 2025, *ApJ*, **984**, L59
- Arzoumanian, D., André, P., Didelon, P., et al. 2011, *A&A*, **529**, L6
- Arzoumanian, D., André, P., Könyves, V., et al. 2019, *A&A*, **621**, A42
- Arzoumanian, D., Russeil, D., Zavagno, A., et al. 2022, *A&A*, **660**, A56
- Bai, M., & Urtasun, R. 2017, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 5221
- Berg, S., Kutra, D., Kroeger, T., et al. 2019, *Nat. Methods*, **16**, 1226
- Berthelot, L., Zavagno, A., Artières, T., et al. 2024, *A&A*, **692**, A41
- Blum, H. 1967, *Models for the Perception of Speech and Visual Form*, 362
- Boe, B., Habbal, S., & Druckmüller, M. 2020, *ApJ*, **895**, 123
- Bolya, D., Zhou, C., Xiao, F., & Lee, Y. J. 2019, *Proc. IEEE/CVF Int. Conf. Comput. Vis.*
- Cao, H., Wang, Y., Chen, J., et al. 2022, in *European Conference on Computer Vision* (Springer), 205
- Carrière, J.-S., Ferrière, K., Ristorcelli, I., & Montier, L. 2022a, *A&A*, **668**, A42
- Carrière, J.-S., Montier, L., Ferrière, K., & Ristorcelli, I. 2022b, *A&A*, **668**, A41
- Chen, Y.-C., Genovese, C. R., & Wasserman, L. 2014, arXiv e-prints [arXiv:1406.1803]
- Chen, Y.-C., Ho, S., Tenneti, A., et al. 2015, *MNRAS*, **454**, 3341
- Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., & Yuille, A. L. 2017, in arXiv e-prints [arXiv:1706.05587]
- Chen, M. C.-Y., Di Francesco, J., Rosolowsky, E., et al. 2020, *ApJ*, **891**, 84
- Chen, J., Lu, Y., Yu, Q., et al. 2021, arXiv e-prints [arXiv:2102.04306]
- Cheng, B., Schwing, A., & Kirillov, A. 2021, arXiv e-prints [arXiv:2107.06278]
- Cheng, B., Misra, I., Girdhar, R., Gkioxari, G., & Kirillov, A. 2022, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 1290
- Clark, S., & Hensley, B. S. 2019, *ApJ*, **887**, 136
- Clark, S., Peek, J., & Putman, M. 2014, *ApJ*, **789**, 82
- Clark, S. E., Hill, J. C., Peek, J. E., Putman, M. E., & Babler, B. L. 2015, *Phys. Rev. Lett.*, **115**, 241302
- Clarke, S. D., Williams, G. M., & Walch, S. 2020, *MNRAS*, **497**, 4390
- COCO Consortium. 2017, pycocotools: Python API for the COCO dataset and evaluation, <https://github.com/cocodataset/cocoapi>
- Cox, N. L. J., Arzoumanian, D., André, P., et al. 2016, *A&A*, **590**, A110
- Dominguez Sánchez, H., Huertas-Company, M., Bernardi, M., Tuccillo, D., & Fischer, J. 2018, *MNRAS*, **476**, 3661

- Fiorellino, E., Elia, D., André, P., et al. 2021, *MNRAS*, **500**, 4257
- Frangi, A. F., Niessen, W. J., Vincken, K. L., & Viergever, M. A. 1998, in *Medical Image Computing and Computer-Assisted Intervention—MICCAI'98: First International Conference Cambridge, MA, USA, October 11–13, 1998 Proceedings 1* (Springer), 130
- Girshick, R. 2015, in *Proceedings of the IEEE International Conference on Computer Vision*, 1440
- Girshick, R., Donahue, J., Darrell, T., & Malik, J. 2014, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 580
- Grandvalet, Y., & Bengio, Y. 2004, *Adv. Neural Inf. Process. Syst.*, **17**
- Green, C.-E., Cunningham, M. R., Dawson, J. R., et al. 2017, *ApJ*, **840**, L17
- Gu, W., Bai, S., & Kong, L. 2022, *Image Vis. Comput.*, **120**, 104401
- Hacar, A., Tafalla, M., Forbrich, J., et al. 2018, *A&A*, **610**, A77
- Hacar, A., Clark, S., Heitsch, F., et al. 2023, in *Astronomical Society of the Pacific Conference Series*, 534, 153
- Halal, G., Clark, S. E., Cukierman, A., et al. 2024, *ApJ*, **941**, 155
- He, K., Zhang, X., Ren, S., & Sun, J. 2016, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 770
- He, K., Gkioxari, G., Dollár, P., & Girshick, R. 2017, in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2961
- Hennebelle, P., & Inutsuka, S.-i. 2019, *Front. Astron. Space Sci.*, **6**, 5
- Hong, S. E., Jeong, D., Hwang, H. S., & Kim, J. 2021, *ApJ*, **913**, 76
- Huang, H., Lin, L., Tong, R., et al. 2020, in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (IEEE)*, 1055
- Inoue, T., & Inutsuka, S.-i. 2012, *ApJ*, **759**, 35
- Jegham, N., Koh, C. Y., Abdelatti, M., & Hendawi, A. 2024, *Yolo11, and Their Previous Versions*
- Jelić, V., Prelogović, D., Haverkorn, M., Remeijn, J., & Klindžić, D. 2018, *A&A*, **615**, L3
- Jiao, R., Zhang, Y., Ding, L., et al. 2024, *Comput. Biol. Med.*, **169**, 107840
- Juvela, M. 2016, *A&A*, **593**, A58
- Koch, E. W., & Rosolowsky, E. W. 2015, *MNRAS*, **452**, 3435
- Könyves, V., André, P., Arzoumanian, D., et al. 2020, *A&A*, **635**, A34
- Kritsuk, A. G., et al. 2013, *MNRAS*, **428**, 2311
- Lee, D.-H. 2013, Pseudo-Label: The Simple and Efficient Semi-Supervised Learning Method for Deep Neural Networks, Workshop on Challenges in Representation Learning, ICML, https://deeplearning.net/wp-content/uploads/2013/03/pseudo_label_final.pdf
- Lin, T.-Y., Maire, M., Belongie, S., et al. 2014, in *European Conference on Computer Vision* (Springer), 740
- Liu, D. C., & Nocedal, J. 1989, *Math. Program.*, **45**, 503
- Long, J., Shelhamer, E., & Darrell, T. 2015, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 3431
- Louvet, F., Motte, F., Maury, A., & et al. 2022, *A&A*, **658**, A111
- Ma, Y. K., McClure-Griffiths, N. M., Clark, S. E., et al. 2023, arXiv e-prints [arXiv:2302.04880]
- Malinen, J., Montier, L., Montillaud, J., et al. 2016, *MNRAS*, **460**, 1934
- Mannfors, E., Juvela, M., Liu, T., & Pelkonen, V.-M. 2025, *A&A*, **695**, A242
- Men'shchikov, A. 2013, *A&A*, **560**, A63
- Men'shchikov, A. 2021, *A&A*, **649**, A89
- Meyer, F. 1994, *Signal Process.*, **38**, 113
- Molinari, S., Swinyard, B., Bally, J., et al. 2010, *A&A*, **518**, L100
- Motte, F., Bontemps, S., Csengeri, T., et al. 2022, *A&A*, **662**, A8
- Myers, P. C. 2017, *ApJ*, **838**, 10
- Nutter, D., Kirk, J., Stamatellos, D., & Ward-Thompson, D. 2008, *MNRAS*, **384**, 755
- Oers, J., Carrière, J.-S., Ferrière, K., & others. 2023, in *SF2A-2023: Proceedings of the Annual French Society of Astronomy and Astrophysics Conference*, 313, <https://sf2a.eu/proceedings/2023/2023sf2a.conf.3130.pdf>
- Oktay, O., Schlemper, J., Folgoc, L. L., et al. 2018, arXiv e-prints [arXiv:1804.03999]
- Ossenkopf-Okada, V., & Stepanov, R. 2019, *A&A*, **621**, A5
- Ostriker, J. 1964, *ApJ*, **140**, 1056
- Palmeirim, P., André, P., Kirk, J., et al. 2013, *A&A*, **550**, A38
- Panopoulou, G. V., Tassis, K., Goldsmith, P., & Heyer, M. 2014, *MNRAS*, **444**, 2507
- Panopoulou, G., Psaradaki, I., & Tassis, K. 2016, *MNRAS*, **462**, 1517
- Peretto, N., André, P., Könyves, V., et al. 2012, *A&A*, **541**, A63
- Pezzuto, S., Benedettini, M., Di Francesco, J., et al. 2021, *A&A*, **645**, A55
- Pillsworth, R., & Pudritz, R. E. 2024, *MNRAS*, **528**, 209
- Pineda, J. E., Palmeirim, P., André, P., et al. 2016, *A&A*, **590**, A102
- Planck Collaboration Int. XXII. 2016, *A&A*, **586**, A135
- Polychroni, D., Schisano, E., Elia, D., et al. 2013, *ApJ*, **777**, L33
- Priestley, F., & Whitworth, A. 2022, *MNRAS*, **512**, 1407
- Qiao, S., Shen, W., Zhang, Z., Wang, B., & Yuille, A. 2018, in *Proceedings of the European Conference on Computer Vision (ECCV)*, 135
- Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. 2016, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 779
- Ren, S., He, K., Girshick, R., & Sun, J. 2015, *Adv. Neural Inf. Process. Syst.*, **28**
- Rivera-Ingraham, A., Ristorcelli, I., Juvela, M., et al. 2016, *A&A*, **591**, A90
- Ronneberger, O., Fischer, P., & Brox, T. 2015, in *Medical Image Computing and Computer-Assisted Intervention (MICCAI)* (Springer), 234
- Salji, C., Richer, J., Buckle, J., et al. 2015a, *MNRAS*, **449**, 1782
- Salji, C., Richer, J. S., Buckle, J. V., et al. 2015b, *MNRAS*, **449**, 1769
- Schisano, E., Rygl, K., Molinari, S., et al. 2014, *ApJ*, **791**, 27
- Schisano, E., Molinari, S., Elia, D., et al. 2020, *MNRAS*, **492**, 5420
- Schneider, S., & Elmegreen, B. G. 1979, *ApJS*, **41**, 87
- Seifried, D., Walch, S., Girichidis, P., et al. 2017, *MNRAS*, **472**, 4797
- Sekh, A. A., Opstad, I. S., Godtliebsen, G., et al. 2021, *Nat. Mach. Intell.*, **3**, 1071
- Shimajiri, Y., André, P., Ntormousi, E., et al. 2019, *A&A*, **632**, A83
- Smith, R. J., Glover, S. C., & Klessen, R. S. 2014, *MNRAS*, **445**, 2900
- Smith, R. J., Treß, R. G., Sormani, M. C., et al. 2019, arXiv e-prints [arXiv:1911.05753]
- Soille, P. 2003, *Morphological Image Analysis: Principles and Applications*, 2nd edn. (Berlin, Heidelberg: Springer)
- Soler, J. D., Hennebelle, P., Martin, P., et al. 2013, *ApJ*, **774**, 128
- Sousbie, T. 2011, *MNRAS*, **414**, 350
- Suin, P., Shore, S. N., & Pavlík, V. 2022, *A&A*, **667**, A69
- Vafaei Sadr, A., Noroozi, A., Teyssier, R., & Arzoumanian, D. 2022, *A&A*, **666**, A153
- Vincent, L., & Soille, P. 1991, *IEEE Trans. Pattern Anal. Mach. Intell.*, **13**, 583
- Walch, S., Girichidis, P., Naab, T., et al. 2015, *MNRAS*, **454**, 238
- Wareing, C. J., Pittard, J. M., & Falle, S. A. E. G. 2016, *MNRAS*, **457**, 76
- Xie, E., Wang, W., Yu, Z., et al. 2021, in *Advances in Neural Information Processing Systems* (NeurIPS)
- Xu, D., Kong, S., Kaul, A., Arce, H. G., & Ossenkopf-Okada, V. 2023, *ApJ*, **955**, 113
- Zavagno, A., Dupé, F. X., Bensaid, S., et al. 2023, *A&A*, **669**, A120
- Zhang, Z., Liu, Q., & Wang, Y. 2018, *IEEE Geosci. Remote Sens. Lett.*, **15**, 749
- Zhou, Z., Rahman Siddiquee, M. M., Tajbakhsh, N., & Liang, J. 2018, in *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support: 4th International Workshop, DLMIA 2018, and 8th International Workshop, ML-CDS 2018, held in conjunction with MICCAI 2018, Granada, Spain, September 20, 2018, proceedings 4* (Springer), 3

Appendix A: Dataset generation and properties

A.1. Dataset generation

In this section we describe the different steps of the dataset generation. A step-by-step illustration of the pipeline is presented in Fig. A.1. Each map is generated as a two-dimensional square image containing several non-overlapping synthetic filaments. Unless stated otherwise, the number of filaments is fixed at generation time, and each filament is independently assigned either a Plummer-like profile or a non-Plummer profile. For maps containing at least two filaments, the number of Plummer-like filaments is drawn from a binomial distribution with probability 0.5, while enforcing that at least one filament of each type is present. This construction allows the dataset to contain both target filaments and contaminating filamentary structures with different radial behavior.

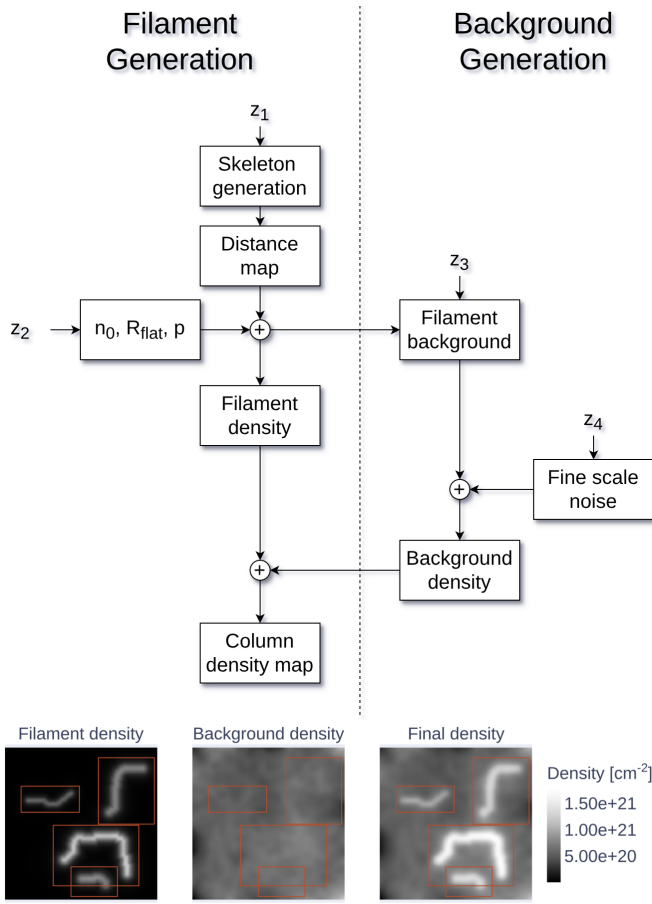


Fig. A.1: Overview of the dataset generation pipeline. The left branch presents the filament generation process. The right branch describes the background generation. The z_i represent random generation.

A.1.1. Skeleton generation

The geometric structure of each synthetic filament is defined by a one-pixel-wide skeleton, representing the central axis along which the filament profile is later constructed. Since there is currently no established statistical model that captures the exact morphology of interstellar filament skeletons, we adopt a random generation approach. Our objective is to produce a diverse,

yet physically plausible, ensemble of skeletons that reflects the qualitative properties observed in real molecular clouds.

To avoid generating highly turbulent or curved structures, we employ a pseudo-random process based on a constrained random walk. Each skeleton is constructed by iteratively stepping from an initial position in a random direction, while maintaining smoothness through limited angular variation between successive steps. The initial position is sampled uniformly in the image, and the initial direction is sampled uniformly in $[0, 2\pi]$. At each step, the direction is updated by adding a random angular perturbation uniformly drawn in $[-\pi/8, \pi/8]$, and the walk is advanced with a fixed step size of two pixels. The segment connecting two successive positions is rasterized with a one-pixel thickness, ensuring a connected skeleton at the pixel level. At each iteration, a new orientation is sampled from a uniform distribution centered on the previous direction, with a bounded angular deviation. A fixed-length segment is then drawn in this direction, and its coordinates are recorded to ensure path continuity and pixel-level connectivity.

The algorithm begins at a randomly selected position within the image and proceeds to construct a skeleton of a randomly chosen length. This choice is motivated by observational studies suggesting that filament lengths approximately follow log-normal distribution (see Fig. A.1 in André et al. 2019). In practice, the proposed length is sampled from a log-normal distribution with mean parameter 3.5 and standard deviation parameter 1 in logarithmic space. Skeletons shorter than 20 pixels or longer than 160 pixels are rejected and regenerated. For simplicity and consistency, all generated skeletons are unbranched, forming single, connected paths without bifurcations. After generation, each skeleton is recentered in the image before being randomly placed in the final map. To avoid overlapping structures, the corresponding mask is cropped and inserted at a random location only if it does not intersect any previously placed filament mask. Examples of skeletons can be found in the left column of Fig. 2.

This method balances morphological diversity with physical realism, producing skeletons that are statistically varied but smooth and continuous, in line with the large-scale structure of filaments observed in molecular clouds.

A.1.2. Filament generation from skeleton and Plummer parameters

Once the skeleton is generated, the corresponding filament is constructed by assigning a radial density profile around its central axis. Each filament is parameterized by a randomly sampled triplet $((n_0, R_{\text{flat}}, p))$ (see Sect. 3.4 for bound description), corresponding respectively to the central density, the flat radius, and the power-law index of a Plummer-like profile. The flat radius is sampled uniformly between 3 and 8 pixels. The power-law index is drawn from a Gaussian distribution with mean 5 and standard deviation 1, with rejection of values smaller than 2. The central column-density normalization is drawn from a log-normal distribution with median scale $9 \times 10^{20} \text{ cm}^{-2}$ and logarithmic standard deviation 1.1. To avoid unrealistically constant filaments, the central column density is allowed to vary along the skeleton through a smooth one-dimensional random curve. This curve is generated as a sum of Gaussian components, including two components preferentially located close to the filament ends and additional components whose number scales approximately with the skeleton length. The resulting curve is smoothed with a Gaussian kernel of standard deviation 3 pixels and then rescaled to set the local value of $n_0(s)$ along the skeleton. These parameters

define the theoretical density distribution transverse to the filament, as introduced in Sect. 3.1.

Rather than explicitly constructing orthogonal radial profiles at each skeleton point and evaluating the Plummer function along them—which is computationally expensive and may result in incomplete coverage—we adopt an efficient approximation. For every pixel of the image, we compute its Euclidean distance to the closest point on the skeleton. This scalar value is then used as the radial argument in the Plummer function. The local Plummer parameters are propagated from the nearest skeleton pixel to the rest of the image using the Euclidean distance transform. The resulting parameter maps are then smoothed with a Gaussian kernel of standard deviation 4 pixels, which avoids discontinuities between neighbouring skeleton segments and produces smoothly varying filament properties.

In two dimensions, the resulting density field is given by

$$N(\mathbf{x}) = n_0(s) \left[1 + \left(\frac{d(\mathbf{x})}{R_{flat}} \right)^2 \right]^{-(p-1)/2}, \quad (\text{A.1})$$

where $d(\mathbf{x}) = \min_{s \in S} \|\mathbf{x} - \mathbf{s}\|_2$ is the distance from position $\mathbf{x}$ to the nearest skeleton point $\mathbf{s}$ in the set S defining the filament axis and $n_0(s)$ the central column density of the nearest skeleton point $\mathbf{s}$ in the set S .

In addition to Plummer-like filaments, the generation procedure can introduce non-Plummer filamentary contaminants. These structures share the same skeleton-generation and mask-placement procedure, but their radial profile is defined by a linear decrease from n_0 at the skeleton to the value of the Plummer profile at $2R_{flat}$, followed by a constant plateau beyond this radius. This produces elongated structures that are morphologically filamentary but do not follow the target Plummer profile.

For each filament, the associated instance mask is defined as the set of pixels whose distance to the skeleton is smaller than or equal to $2R_{flat}$. Consequently, $2R_{flat}$ corresponds to the mask radius and the full transverse mask width is approximately $4R_{flat}$. With $R_{flat} \in [3, 8]$ pixels, this corresponds to mask radii between 6 and 16 pixels and approximate full mask widths between 12 and 32 pixels. At a pixel scale of $18''/2$ and a distance of 260 pc, one pixel corresponds to approximately 2.29×10^{-2} pc. The sampled R_{flat} range therefore corresponds to 0.069–0.183 pc, while the full mask width spans approximately 0.275–0.733 pc.

This approximation offers two key advantages. First, it is significantly less computationally intensive than computing local orthogonal profiles at each skeleton segment, especially in large datasets or higher dimensions. Second, it guarantees full spatial coverage: every point in the image is assigned a density value based on its distance to the filament, thereby avoiding the presence of unfilled or undefined pixels that often result from sparse radial sampling.

Although this method introduces minor geometric distortions compared to a strictly orthogonal profile evaluation, it preserves the essential radial symmetry of the Plummer distribution around the filament axis and remains a valid approximation in the context of large-scale statistical studies.

A.1.3. Background generation

In addition to modeling the intrinsic structure of filaments, a crucial component of synthetic dataset generation is the inclusion of a background field that mimics the local environment surrounding the filament. This background should be both realistic (observationally-wise) and spatially coherent with the fila-

ments, in order to provide meaningful conditions for supervised learning or physical inference.

While constant, linear, and low-order polynomial models are widely used in the literature due to their simplicity and compatibility with Plummer fitting techniques (Arzoumanian et al. 2019; André et al. 2014; Mannfors et al. 2025), none of these models adequately capture the complexity and spatial variability of backgrounds observed in real interstellar environments. To address this limitation, we propose to use a novel background generation method designed to produce more realistic, filament-correlated diffuse structures.

Our approach begins by leveraging the filament density map to define regions of enhanced large-scale structure. A smoothed version of the filament map is first produced via Gaussian filtering, which informs the global structure of the background. More specifically, the filament map is normalized by its maximum value and smoothed with a Gaussian kernel of standard deviation 10 pixels. This smoothed filament map is then used as a lower envelope for a uniform random field, ensuring that the generated background remains spatially correlated with the filament distribution. This is combined with a layer of fine-scale Gaussian-filtered noise to introduce local fluctuations, mimicking the multiscale nature of the ISM. The large-scale component is obtained by smoothing the random field with a Gaussian kernel of standard deviation 10 pixels, while the fine-scale component is generated from an independent random field smoothed with a Gaussian kernel of standard deviation 3 pixels. The fine component is added with a relative weight of 0.2. The resulting map is then nonlinearly transformed via exponentiation to enhance contrast and spatial coherence. We use an exponent of 1.2 for this nonlinear transformation. The background is finally min–max normalized to the same intensity range as the filament-density map before being added to the synthetic filament emission. Finally, the background is rescaled to match the intensity range of the filament map, ensuring a physically plausible superposition. Compared to observation, we do not take into consideration, hence do not model, bright sources in our synthetic maps.

Importantly, this method guarantees that the generated background is spatially correlated with the filament morphology while remaining non-trivially structured. Unlike parametric models, this generative procedure can simulate realistic diffuse emission without imposing a predefined functional form, making it a powerful tool for training and benchmarking filament detection and characterization algorithms in more observationally faithful scenarios.

A.2. Plummer parameter fitting

Although the Plummer parameters are initially sampled from observation-inspired distributions and the filament density is synthesized directly from the Plummer profile based on Eq. 1, the subsequent addition of filament and background components (see Fig. A.1) can alter the effective parameters in the final maps. We therefore re-estimate the Plummer parameters after image synthesis, directly on the final density maps.

Concretely, we first extract the filament skeleton from the binary mask using a Medial Axis Transform (MAT) skeletonization (Blum 1967). For each pixel within the mask, we then compute its Euclidean distance to the nearest skeleton pixel, yielding paired samples $\{(r_i, n_i)\}_{i=1}^M$ of distance-to-skeleton r_i and observed density n_i . Let $N(r; n_0, R_{flat}, p)$ denote the Plummer function (Eq. 1). We fit the parameters by minimizing the mean

squared error (MSE):

$$\min_{n_0, R_{\text{flat}}, p} \frac{1}{M} \sum_{i=1}^M (n_i - N(r_i; n_0, R_{\text{flat}}, p))^2, \quad (\text{A.2})$$

using the L-BFGS optimizer (Liu & Nocedal 1989). This procedure returns the triplet $(\tilde{n}_0, \tilde{R}_{\text{flat}}, \tilde{p})$ along with an error summary, *err*, defined as the final mean MSE of the fit Plummer model on the sampled (r_i, n_i) pairs. Because R_{flat} define the mask (see Sect. 3.2), we do not optimize the $\tilde{R}_{\text{flat}}$ parameter and force $\tilde{R}_{\text{flat}} = R_{\text{flat}}$. Throughout this paper, we exclusively refer to the fit Plummer parameters, denoted for simplicity as $(n_0, R_{\text{flat}}, p)$. A few examples of Plummer profile fits can be found in Fig. 3.

Appendix B: FilFinder gridsearch

In this section we detail the methodology adopted to apply FilFinder to our synthetic dataset under an automatic parameter-selection protocol. The purpose is not to reproduce the full expert-guided workflow for which FilFinder is often used, but to test whether the method can be deployed in a training-validation-test framework similar to that used in machine learning. In this framework, model or algorithm parameters are selected on a validation set and then frozen before final evaluation on a held-out test set.

Although FilFinder is a non-learning-based algorithm, it depends on several user-defined parameters that influence the mask and skeleton returned by the algorithm, such as the global threshold, adaptive-thresholding scale, smoothing scale, and minimum retained structure size. To avoid introducing manual tuning on the test set, we perform a systematic grid search over this parameter space using only the training and validation splits. The combination of parameters that yields the best average validation DSC is selected, and this optimal configuration is then applied unchanged to the test set for final evaluation.

This procedure ensures a leakage-free and reproducible experiment. However, it also deliberately restricts FilFinder to a single global configuration across maps with different filament contrasts, densities, and background realizations. The experiment therefore probes the scalability of automatic global parameter selection, rather than the performance attainable by local, expert-guided tuning.

FilFinder requires the following hyperparameters:

1. **flatten_percent**: The percentile of the data used to set the normalization of the arctan transform.
2. **smooth_size**: Patch size (in pixels) used to smooth the flattened image before adaptive thresholding.
3. **size_thresh**: Lower area threshold (in pixels) on objects found by adaptive thresholding.
4. **glob_thresh**: Percentile of the data used for global masking; all intensities below this percentile are excluded from the filamentary mask.
5. **adapt_thresh**: Patch size (in pixels) for adaptive thresholding.

Table B.1 gives a description of the hyperparameter value study during the gridsearch scheme for FilFinder. To remain coherent with our dataset generation procedure (see Sect. 3.4), we set the beamwidth to 18''2 and the distance to 260pc. We carefully selected the FilFinder parameter grid to remain consistent with the spatial scales present in our synthetic data while keeping the grid sufficiently compact for large-scale automatic tuning. In particular, smooth_size and adapt_thresh control

the characteristic spatial scale, in pixels, of the smoothing and adaptive-thresholding operations used during mask construction. With a pixel scale of 18''2 and a distance of 260 pc, one pixel corresponds to approximately 2.29×10^{-2} pc. Therefore, the values explored in the grid, 8, 16, and 32 pixels, correspond to physical scales of approximately 0.184, 0.367, and 0.735 pc, respectively. In our synthetic dataset, the filament mask is defined as the set of pixels whose distance to the skeleton is smaller than $2R_{\text{flat}}$. The full transverse mask width is therefore approximately $4R_{\text{flat}}$. For $R_{\text{flat}} \in [0.030, 0.121]$ pc, this gives a mask-width range of $4R_{\text{flat}} \in [0.121, 0.484]$ pc, corresponding to approximately 5.3 to 21.1 pixels. The values of 8 and 16 pixels therefore fall within the expected filament-mask width range, while the 32-pixel value explores a broader smoothing and thresholding scale. This compact grid also addresses the scalability constraint of applying automatic parameter search to thousands of maps; a much denser grid, although possible, would rapidly dominate the computational cost.

Table B.1. Grid of FilFinder hyperparameters explored.

Dataset	Candidate values		
flatten_percent	0.97	0.98	0.99
smooth_size	8,	16	32
size_thresh	16	32	64
glob_thresh	0.85	0.90	0.95
adapt_thresh	8	16	32

Notes. The table lists the hyperparameter grid explored for FilFinder. The Cartesian product of these candidate values yields $3^5 = 243$ configurations. The best hyperparameter set is highlighted in bold.

Appendix C: Single-filament generation

To compare UNet and Mask-RCNN on instance-level metrics as a function of Plummer parameters, we require per-image predictions that contain a single, unambiguous instance. Rather than post-processing UNet's semantic maps into approximate instances (which would introduce biases), we construct a single-filament test set in which each image contains exactly one labeled filament. Crucially, we preserve the same Plummer parameter distributions and a realistic, filament-coherent background so that low-density (faint) filaments remain represented and measurable.

Pipeline. The single-filament map is produced as follows:

1. **Propose a full scene.** Generate several candidate filaments (skeletons) and assign per-skeleton Plummer parameters $(n_0, R_{\text{flat}}, p)$. Compute the distance transform to the union of all skeletons and propagate $(n_0, R_{\text{flat}}, p)$ to every pixel by nearest-skeleton assignment, followed by light smoothing. Evaluate the Plummer field to obtain a dense map of the sum of all filaments, denoted density_{all}.
2. **Create a scene-coherent background.** Build the diffuse background from density_{all} (as in the multi-filament case), ensuring that large-scale structure and contrast resemble realistic conditions.
3. **Select one filament.** Randomly choose a filament index k . Recompute a distance transform to the chosen skeleton only and propagate that filament's $(n_0, R_{\text{flat}}, p)$ to pixels (all other

filaments are ignored at this stage). Evaluate the corresponding Plummer profile to obtain density_k .

4. **Compose and finalize.** Form the image as density_k + background, apply a mild Gaussian smoothing, and add additive noise scaled to the image's dynamic range.
5. **Export labels.** Return the binary mask, skeleton mask, and bounding box for the selected filament only ($N=1$); provide the distance map to the selected skeleton and the reference $n_{\max} = \max(\text{density}_{\text{all}} + \text{background})$ (useful for $n_0/n_{\max}$ analyses).
6. **Normalization.** Normalization is applied based on $n_{\max}$, to simulate low-density filaments (important for low-density studies).

Differences from the multi-filament generator.

- Only one filament is rendered and labeled.
- The distance map is computed with respect to the selected skeleton (not the union).
- The background is still derived from the full candidate scene to keep realistic spatial correlations.
- The outputs include exactly one instance label per image.

This design yields unambiguous instance-level supervision for UNet and FilFinder, while retaining the same physical parameter ranges and background statistics as in the multi-filament benchmark.

Appendix D: Benchmark UNet results

In this section we summarize UNet performance for the same parameter and bias studies discussed for Mask-RCNN in Sects. 5.2 and 5.3.

D.1. Performance with respect to Plummer parameters

D.1.1. Performance versus p

Figure D.1 shows UNet performance as a function of the Plummer slope p . As for Mask-RCNN, metrics remain stable for $p > 1.5$ but rapidly degrade toward $p \rightarrow 1$, corresponding to the low-contrast regime.

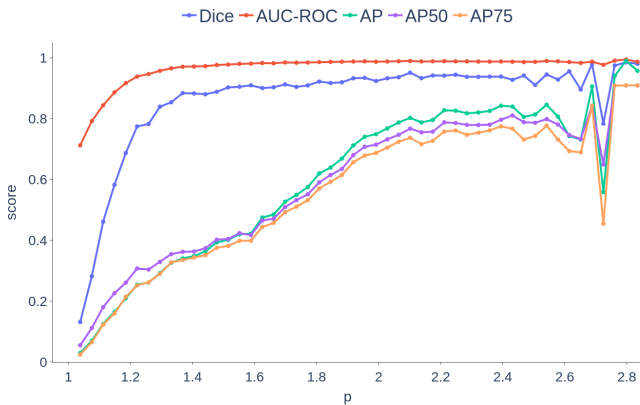


Fig. D.1: UNet performance as a function of the Plummer index p : AUC ROC, DSC, AP, AP50 and AP75 per p bin.

D.1.2. Performance versus $n_0/n_{\max}$

Similarly, Fig. D.2 reports performance versus the contrast proxy $\chi = n_0/n_{\max}$. Metrics remain high for $\chi > 0.5$ and drop systematically in low-density regimes, reproducing trends observed with Mask-RCNN.

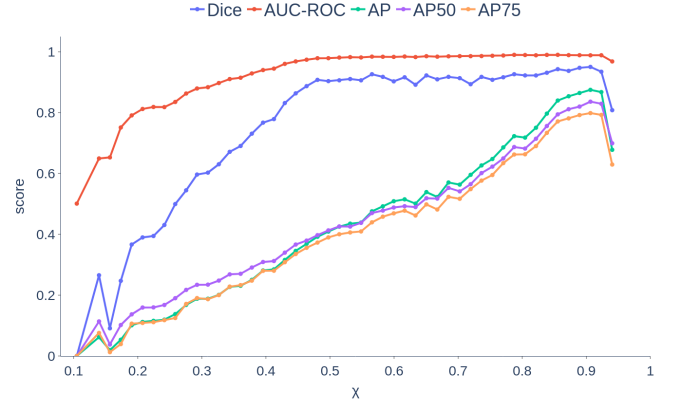


Fig. D.2: UNet performance versus $\chi = n_0/n_{\max}$: AUC ROC, DSC, AP, AP50 and AP75 per χ -bin.

Overall, UNet displays similar dependencies on the Plummer parameters as Mask-RCNN, confirming that detectability primarily depends on intrinsic filament contrast and the global map normalization.

D.2. Bias analysis across Plummer parameters

D.2.1. Low-contrast bias

Table D.1. UNet performance under p -bias conditions.

Dataset	AUC ROC	DSC	AP	AP50	AP75
biased	0.87	0.79	0.64	0.61	0.60
unbiased	0.98	0.73	0.87	0.85	0.82

Notes. Comparison between p -biased and unbiased training sets. In the biased case, only high-contrast filaments ($p > 1.5$) were annotated.

As shown in Table D.1 and Fig. D.3, excluding low- p filaments during training significantly reduces performance in the low-contrast regime. Unbiased training yields consistent scores across p , confirming that missing annotations primarily limit generalization rather than architecture choice.

D.2.2. Low-density bias

Figure D.4 shows the same trend: models trained on biased datasets perform well on bright, high- χ filaments but fail to generalize to faint structures. Unbiased training preserves performance across the full dynamic range.

D.3. Summary

UNet exhibits the same sensitivity to annotation bias as Mask-RCNN: when trained on incomplete ground truth (missing low- p or low- χ structures), performance collapses in those regimes

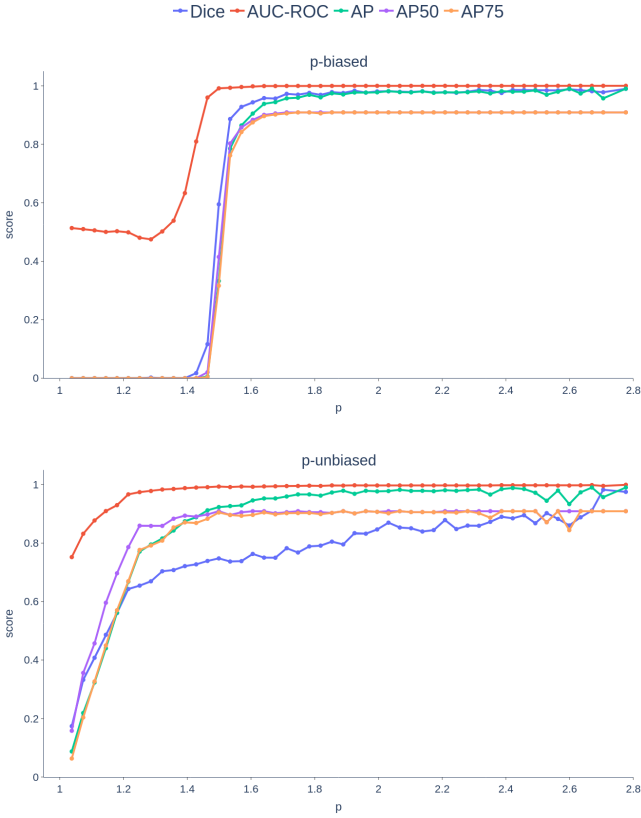


Fig. D.3: UNet performance versus p for biased (top) and unbiased (bottom) training.

Table D.2. UNet performance under density bias.

Dataset	AUC ROC	DSC	AP	AP50	AP75
biased	0.89	0.80	0.67	0.63	0.62
unbiased	0.98	0.73	0.87	0.85	0.82

Notes. Comparison between density-biased and unbiased training. In the biased case, only filaments with $\chi > 0.5$ were annotated.

despite strong results on the training distribution. This confirms that dataset completeness, rather than model architecture, is the dominant factor governing generalization to faint, low-contrast filaments.

Appendix E: Mask-RCNN unmatched instances

Motivation. We examine whether our dataset may unintentionally contain unlabeled filamentary structures. Specifically, we investigate whether instances predicted by *Mask-RCNN* that do not match any ground-truth mask correspond to genuine filaments overlooked during labeling or simply represent false positives.

Hypothesis. Our working hypothesis is that unmatched predictions can be interpreted as false positives. To test this hypothesis, we designed the following protocol. For each unmatched instance, we fit a Plummer profile to obtain the parameters $\theta = (n_0, R_{\text{flat}}, p, \text{err})$ (following the Appendix A.2 method). We

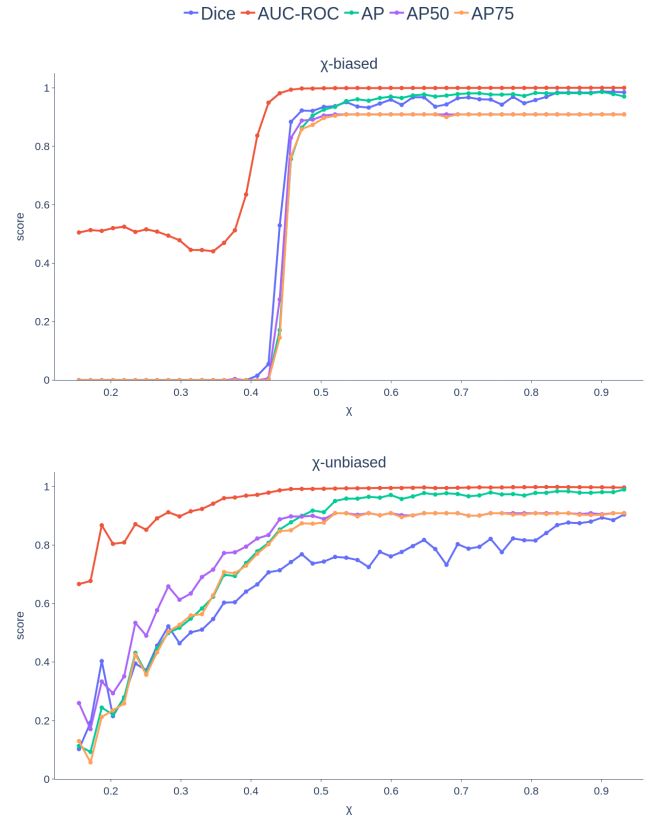


Fig. D.4: UNet performance versus $\chi = n_0/n_{\text{max}}$ for biased (top) and unbiased (bottom) training.

then compare the distributions of θ for unmatched predictions and for ground-truth filaments. Two outcomes are possible: (i) if the distributions are clearly distinct, unmatched predictions are unlikely to belong to the ground-truth distribution and should be regarded as false positives; (ii) if the distributions overlap substantially, unmatched predictions are statistically consistent with ground-truth filaments and should be considered as filaments.

Estimating $(n_0, R_{\text{flat}}, p, \text{err})$. Using a *Mask-RCNN* model trained on the benchmark dataset (see Sect. 5.1), we collect all predicted instances on the benchmark test set. We define an unmatched instance as any prediction with $\text{IoU} \leq 0.5$ to every ground-truth mask (consistent with the benchmark configuration). For each unmatched instance, we perform skeletonization followed by Plummer-profile fitting to obtain $(n_0, R_{\text{flat}}, p, \text{err})$ (with the described method in Appendix A.2). For ground-truth filaments, the corresponding tuples are retrieved from metadata stored during generation. This produces two empirical distributions of $(n_0, R_{\text{flat}}, p, \text{err})$: one for unmatched predictions and one for ground-truth filaments.

Classifier. We use a small fully connected network to classify tuples $(n_0, R_{\text{flat}}, p, \text{err})$ as unmatched versus ground truth. The architecture is composed of linear layers (4–64–64–64–1) with ReLU activations after each hidden layer, trained with a binary cross-entropy objective and the Adam optimizer. We estimate generalization via 5-fold out-of-fold (OOF) cross-validation: for each fold, the model is trained on four folds and evaluated on

the held-out fold. Classifier’s performance is reported as its accuracy (fraction of correctly classified tuples).

Results. Table E.1 summarizes the OOF accuracy. The classifier achieves a mean accuracy of 0.90 with a standard deviation across folds of 0.059, indicating both strong separability and stability.

Table E.1. Classifier accuracy under 5-fold OOF cross-validation.

Metric	Value
Mean accuracy	0.90
Variance accuracy	0.059

Notes. Mean and Variance of OOF accuracy over 5 folds.

To probe where separability is strongest, we examine accuracy as a function of p (Fig. E.1). Accuracy is about 100% for instances (unmatched and groundtruth) with $p > 1.9$, while it drops noticeably for $p \in [1.00, 1.5]$. The worst accuracy being 76% for instances (unmatched and groundtruth) with $p \in [1.00, 1.2]$.

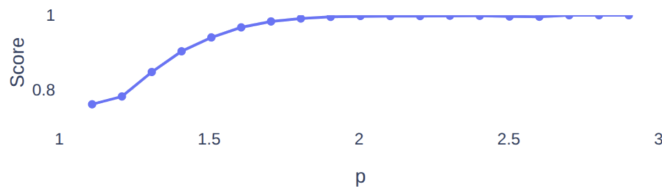


Fig. E.1: Classifier accuracy across p bins.

Conclusion. The two populations—unmatched predictions and ground-truth filaments—are distinguishable in the $(n_0, R_{\text{flat}}, p, \text{err})$ space. Consequently, unmatched Mask-RCNN predictions can be interpreted as false positives in approximately 90% of cases. The classifier accuracy is about 100% for instances with $p > 1.9$. However, when p tends towards 1, specifically with $p < 1.5$, distributions appear to blur as the classifier performance decreases to 76% accuracy.

Appendix F: Computation time

While Sect. 5.1 focused on the outcome of the common automatic parameter-selection protocol, another critical aspect of practical deployment is computational efficiency. Beyond the challenge of incomplete ground truth, filament detection methods must also be scalable and computationally feasible, especially when applied to large-scale astronomical datasets. This is particularly important when a non-learning method is placed in a grid-search framework, because the cost grows with both the number of maps and the number of candidate parameter combinations.

In this section we compare the computational cost associated with each detection method. Specifically, we report both the time required for hyperparameter optimization—which corresponds to the training phase for neural networks and the grid search for FilFinder—and the inference time required to process the test

set. All timing results are summarized in Table F.1, offering insights into the resource demands and scalability of the different approaches under study. Computation times were measured as part of the benchmarking protocol in Sect. 5.1, using the experimental setup detailed in Sect. 4.3.

Before interpreting runtimes, we emphasize that FilFinder and the neural networks were executed on different hardware (Sect. 4.3). Because neural networks benefit from GPU hardware acceleration, they were trained on three GPUs in parallel, whereas FilFinder ran on 32 CPU cores since it does not benefit from GPU computation. In addition, because of computation time constraints, we limited the number of grid search samples to 10000 for FilFinder instead of the 50000 used for neural networks training (see Appendix B for details about FilFinder grid-search). Moreover, the benchmark ran on a shared system with concurrent jobs, which may have introduced resource contention and intermittent slowdowns. Wall-clock times are therefore not directly comparable and should be only viewed as indicative. Our goal is to highlight practical scaling considerations—agnostic hyperparameter optimization and large-scale inference—when deploying a given filament detection method to extensive datasets, rather than to provide an exact speed benchmark. Table F.1 summarize the computation times.

Table F.1. Computation time of models.

Method	UNet	Mask-RCNN	FilFinder
Optimization	10h58min	24h18min	110h40min
Inference	3min	4min	9h15min

Notes. Computation times for each filament detection method during the benchmarking procedure described in Section 5.1. The table reports separately the time required for hyperparameter optimization (training for neural networks, grid search for FilFinder on only 10k maps) and inference (plus final training for neural networks) on the test set. These values provide insight into the computational cost and scalability of each approach, but do not compare directly.

Neural networks front-load most of the cost into a one-off optimization that amortizes well as datasets grow; once trained, per-image inference is inexpensive. In contrast, a grid-search deployment of FilFinder scales with both the number of images and the size of the candidate parameter grid. This makes large automatic searches costly unless the parameter space is pruned, parallelized, or informed by prior knowledge such as expected background level and filament width. Therefore, the computation-time comparison should be read primarily as a scalability analysis of the adopted automatic tuning protocol. FilFinder remains especially useful for targeted analyses where interpretability, skeleton morphology, and expert adjustment to a specific map are desired. As noted in the table footnote, these numbers depend on hardware and parallelization (GPUs for neural networks versus CPUs for FilFinder) and therefore should be read as indicative rather than strictly comparable. The FilFinder optimization time was measured on a 10k-images grid search (as opposed to neural networks trained on 50k maps); different grid sizes or early-stopping criteria would change that figure proportionally.