



HAL
open science

Sketchpose: Learning to Segment Cells with Partial Annotations

Clément Cazorla, Nathanaël Munier, Renaud Morin, Pierre Weiss

► To cite this version:

Clément Cazorla, Nathanaël Munier, Renaud Morin, Pierre Weiss. Sketchpose: Learning to Segment Cells with Partial Annotations. *Journal of Machine Learning for Biomedical Imaging*, 2025, <10.59275/j.melba.2025-f7b3>. <hal-04330824v2>

HAL Id: hal-04330824

<https://hal.science/hal-04330824v2>

Submitted on 25 Aug 2025

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons CC BY 4.0 - Attribution - International License

Sketchpose: Learning to Segment Cells with Partial Annotations

Clément Cazorla ^{1,2}, Nathanaël Munier ¹, Renaud Morin ², Pierre Weiss ¹

¹ Institut de Recherche en Informatique de Toulouse (IRIT), Institut de Mathématiques de Toulouse (IMT), Centre de Biologie Intégrative (CBI), Laboratoire de Biologie Moléculaire, Cellulaire et du Développement (MCD), Université de Toulouse, CNRS, Université Toulouse III – Paul Sabatier, Toulouse, France

² Imactiv-3D, Centre Pierre Potier, 1 place Pierre Potier, 31100 Toulouse, France

Abstract

The most popular networks used for cell segmentation (e.g. Cellpose, Stardist, HoverNet,...) rely on a prediction of a distance map. It yields unprecedented accuracy but hinges on *fully annotated* datasets. This is a serious limitation to generate training sets and perform transfer learning. In this paper, we propose a method that still relies on the distance map and handles partially annotated objects. We evaluate the performance of the proposed approach in the contexts of frugal learning, transfer learning and regular learning on regular databases. Our experiments show that it can lead to substantial savings in time and resources without sacrificing segmentation quality. The proposed algorithm is embedded in a user-friendly Napari plugin.

Keywords

Cellpose, Deep learning, Distance Map, Frugal learning, Napari, Segmentation

Article informations

©2025 Cazorla, Munier, Morin and Weiss. License: CC-BY 4.0

Corresponding author: clement.cazorla31@gmail.com

1. Introduction

Image segmentation plays a fundamental role in the analysis of biological images. It enables the extraction of quantitative information on diverse objects ranging from molecules, droplets, membranes, nuclei, cells, vessels or other structures. In modern biological research, accurate segmentation is often pivotal to better understand the mechanisms of life. The increasing availability of high-throughput imaging technologies has led to a surge in the quantity and complexity of image data, raising significant challenges and opportunities. Manual annotation of the resulting images is labor-intensive, time-consuming, and often impractical for large-scale datasets. Automated segmentation is therefore widely accepted as a critical step in biological research.

A simplified history of cell segmentation Image segmentation has long been dominated by handcrafted algorithms. The processing pipelines typically combine popular tools such as linear filtering, thresholding Otsu (1979), morphological operations Serra and Soille (2012); Legland et al. (2016), active contour models (Snake) Kass et al. (1988) or watershed Vincent and Soille (1991). A significant issue with handcrafted approaches is that they are usually image-specific and rely on the manual tuning of a few complicated hyper-parameters. Although excellent performance can be

achieved, it is often the work of a handful of talented people and these techniques are not broadly applicable.

The introduction of machine learning and especially random forests made image segmentation accessible to a much larger range of researchers. These techniques automatically combine and tune elementary image processing bricks. They are driven by a few easily interpretable user annotations. Embedded in well conceived software such as Ilastik Berg et al. (2019) or Labkit Arzt et al. (2022), these techniques heavily contributed to democratize image segmentation and classification.

Deep learning and convolutional neural networks played an important role in improving the segmentation performance around 2015. For instance, the popular U-Net architecture Ronneberger et al. (2015) increased the accuracy on some cell segmentation challenges by more than 10%, which can be considered as a small revolution. This type of neural network architecture seems to be a good prior for segmenting “natural” images, as suggested by the so-called Deep Image Prior principle Lempitsky et al. (2018). However, it can sometimes demonstrate limited effectiveness when it comes to separating nearby or touching objects. Many applications in biology involve densely packed objects (e.g. cells, nuclei) and a pixel-classification U-Net is often insufficient to perform a satisfactory analysis. To address

this issue, new architectures coming from computer vision such as Mask R-CNN He et al. (2017) have been developed and continued improving the performance.

Roughly at the same time, a few approaches (Deep watershed transform Bai and Urtasun (2017), Deep Regression of the Distance Map Naylor et al. (2018); Kumar et al. (2019), StarDist Schmidt et al. (2018), Hover-Net Graham et al. (2019), Cellpose Stringer et al. (2021), Omnipose Cutler et al. (2022)) have been developed and generated results with an unprecedented quality. Despite certain differences, they all share a common underlying principle. The idea is to make a regression with respect to some *distance function*. Given a set of annotated objects, a distance function to the objects centers or boundaries is computed. A convolutional neural network is then trained to predict the distance function rather than a binary map of the objects. The gradient of this distance function points in opposite directions on each side of the boundary, which makes it possible to determine them with much greater precision. This principle created a new gap in the segmentation accuracy, especially for objects with touching boundaries.

A current trend consists in involving the user in the training procedure. This “human in the loop” principle was incorporated in CellPose 2.0 Pachitariu and Stringer (2022). Users fully annotate patches of the segmented image, to adapt the neural network weights to the image at hand.

It would be hazardous to call these approaches the current “state-of-the-art”, since this field is expanding extremely quickly. However – as of 2025 – we can safely claim that algorithms based on the distance map are at the basis of some of the most popular and efficient cell segmentation methods.

Contributions This work stems from a practical observation: methods which rely on a regression to the distance function currently require exhaustive annotations. As the distance function is a global geometrical property, it is impossible to compute it using just a few sketches. Hence, it is *a priori* unclear how partial annotations can be used in this framework, see Figure 1. Cellpose 2 gets around this problem by allowing the user to annotate patches of interest in their entirety. Similarly, Sugawara (2023) recently proposed a simple extension of Stardist and Cellpose by training the networks on a subset of completely annotated objects. This is a time-consuming process that does not allow the expert to focus on local spots (e.g. a part of boundary) where the network clearly missed the segmentation.

In this paper, we introduce a novel idea that allows us to use the distance function even with partially annotated objects. After drawing just a few regions and boundaries, the user can train a task-aware neural network. This approach capitalizes on the generalization capacity of neural networks, reducing the overall annotation effort without

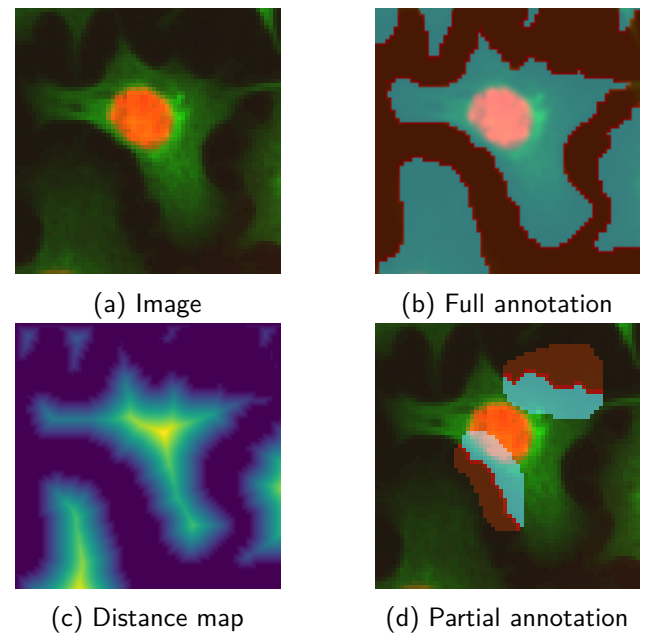


Figure 1: Partial VS full annotation. In (b), the complete annotation is used to compute the distance map shown in (c). In (d), it is unclear how to compute it from the partial object annotation.

sacrificing accuracy. We explore the performance of the proposed architecture in 3 different settings:

- *Few-shot learning*: starting from random weights, we show that just a few *partial* annotations are already enough to quickly realize complex cell segmentation analyses. This is interesting when faced with a problem for which no close pre-trained model exists.
- *Transfer learning*: starting from Omnipose’s optimized weights, we show that just a few clicks at locations where the segmentation is inaccurate lead to improved weights and fast adaptation to out-of-distribution images. This is the traditional field of transfer learning, domain adaptation, e.g.. Our contribution here is to show that this can be done with only a few scattered annotations.
- *Large databases*: finally, we show that large, but partially annotated sets can also be used to train high performance neural networks. This is important since it can significantly accelerate the design of segmentation databases.

This evaluation on both small and large-scale dataset, overall showcases the advantages of our approach in terms of time and resource savings. We developed a Napari plugin Chiu et al. (2022) named Sketchpose, to assess its potential, ensure reproducibility of the results and provide an additional tool to the community. It relies on a modified version of the Omnipose Cutler et al. (2022) algorithm. The plugin is currently being downloaded regularly, with 638 downloads to date.

2. Methodology

2.1 Preliminary definitions and notations

In all the paper \mathcal{X} refers to the image domain, which can be understood as a discrete set of coordinates, or as a continuous domain depending on the context. In the discrete setting, we let $|\mathcal{X}|$ denote the number of pixels of \mathcal{X} .

Definition 1 For an arbitrary set $S \subset \mathcal{X}$, we let ∂S denote its boundary. We use the 4-connectivity (top, bottom, left, right) in the discrete setting.

Definition 2 (Point to set distance) The distance from a point $\mathbf{x} \in \mathcal{X}$ to a set $S \subseteq \mathcal{X}$ is defined by

$$\text{dist}(\mathbf{x}, S) \stackrel{\text{def}}{=} \inf_{\mathbf{x}' \in S} \|\mathbf{x} - \mathbf{x}'\|_2. \quad (1)$$

2.2 Omnipose

Our work is based on the Omnipose cell segmentation architecture Cutler et al. (2022). In this section, we justify this choice, explain its founding principles and then demonstrate how they can be adapted to deal with partial annotations.

2.2.1 Why Omnipose

Cellpose Stringer et al. (2021) has now become a standard in cell segmentation. Its excellent performance, processing speed, and ergonomic graphical interface make it a handy tool for every day cell biology image analysis. However, it occasionally fails in scenarios involving complex and elongated objects. In such cases, it tends to produce over-segmentation, where neighboring objects are split in smaller fragments.

The Omnipose algorithm Cutler et al. (2022) was conceived in order to address this limitation. The main difference between Omnipose and Cellpose is the fact that the distance map is defined as the distance to the cell boundaries in Omnipose, while it is defined as a distance to a cell ‘‘centroid’’ in Cellpose. A weakness of the latter is that there is no canonical choice to define this center, hence Omnipose’s choice seems more principled. This explains our decision to choose and base our work on its architecture.

2.2.2 The main principles

Figure 2 summarizes the main ideas behind the Omnipose architecture and its training. Omnipose is based on a regular convolutional neural network (CNN), with a U-Net like architecture Ronneberger et al. (2015). Given an input 2D image with N pixels, the CNN can be seen as a mapping N_w^{omni} of the form

$$N_w^{\text{omni}} : \mathbb{R}^N \rightarrow \mathbb{R}^N \times \mathbb{R}^N \times \mathbb{R}^{2N} \\ u \mapsto (N_w^b(u), N_w^d(u), N_w^v(u)). \quad (2)$$

It depends on weights w that should be optimized during a training stage. It returns 3 different outputs (illustrated on the top of Figure 2):

- $N_w^b(u) \equiv$ *boundary probability*: at every pixel, the value of this image can be interpreted as a probability of being a boundary between the objects to segment.
- $N_w^d(u) \equiv$ *distance map*: at a given pixel, the value of this map is equal to:
 - The distance of the pixel to the closest object boundary, if the pixel is inside an object.
 - 0 (or a fixed negative value) elsewhere.
- $N_w^v(u) \equiv$ *flow field*: can be interpreted as the gradient of the distance map. It is an essential feature of the Cellpose and Omnipose architectures. Ultimately, the flow is used through a procedure called Euler integration to generate a segmentation mask. This is illustrated on the top right of Figure 2.

2.2.3 The original loss definition

The original training stage involves a collection of $K \in \mathbb{N}$ images $(u_k)_{1 \leq k \leq K}$ together with their *exhaustive* segmentation masks. For every image u_k in the dataset, an algorithm creates the gold standard boundary probability b_k^* , distance map d_k^* and flow field \mathbf{v}_k^* . This is illustrated on the bottom of Figure 2.

The weights w of the neural network are then optimized so as to minimize a loss function that compares the output of the CNN with the gold standard:

$$\inf_w \text{loss}^{\text{omni}}(w) \stackrel{\text{def}}{=} \frac{1}{K} \sum_{k=1}^K \ell_{\mathcal{B}}(b_k, b_k^*) + \ell_{\mathcal{D}}(d_k, d_k^*) + \ell_{\mathcal{V}}(\mathbf{v}_k, \mathbf{v}_k^*), \quad (3)$$

where $b_k = N_w^b(u_k)$, $d_k = N_w^d(u_k)$, $\mathbf{v}_k = N_w^v(u_k)$.

In the original Omnipose implementation available on GitHub, the different losses were defined as follows:

- **Boundary loss** $\ell_{\mathcal{B}}$:

This term compares the predictions b to b^* using the following loss:

$$\ell_{\mathcal{B}}(b, b^*) \stackrel{\text{def}}{=} \frac{\lambda_{\mathcal{B}}}{|\mathcal{X}|} \sum_{\mathbf{x} \in \mathcal{X}} g(b[\mathbf{x}], b^*[\mathbf{x}]), \quad (4)$$

where $g : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$ combines a sigmoid and a binary cross entropy loss.

- **Distance loss** $\ell_{\mathcal{D}}$:

This loss calculates a weighted mean squared error between the predicted distance fields and the ground truth distance fields. It is defined as

$$\ell_{\mathcal{D}}(d, d^*) \stackrel{\text{def}}{=} \frac{\lambda_{\mathcal{D}}}{|\mathcal{X}|} \sum_{\mathbf{x} \in \mathcal{X}} (d[\mathbf{x}] - d^*[\mathbf{x}])^2 \cdot \rho[\mathbf{x}],$$

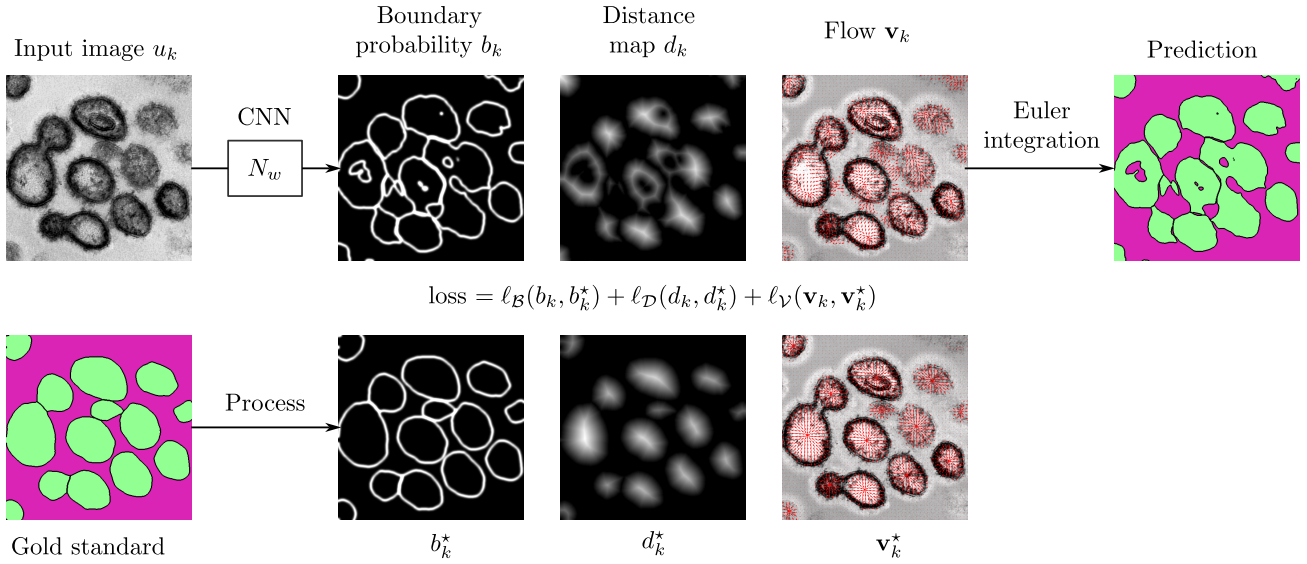


Figure 2: A sketch of the Omnipose training procedure

where $\rho \in \mathbb{R}^N$ is a weight image with higher values around the gold standard boundaries.

▪ **Flow loss** $\ell_{\mathcal{V}}$:

This loss is defined as a weighted sum of three losses $\ell_{\mathcal{V}} = \ell_{\mathcal{V},1}^1 + \ell_{\mathcal{V},2}^2 + \ell_{\mathcal{V},3}^3$. The first one is a mean squared error loss:

$$\ell_{\mathcal{V}}^1(\mathbf{v}, \mathbf{v}^*) \stackrel{\text{def}}{=} \frac{\lambda_{\mathcal{V},1}}{|\mathcal{X}|} \sum_{\mathbf{x} \in \mathcal{X}} \|\mathbf{v}[\mathbf{x}] - \mathbf{v}^*[\mathbf{x}]\|_2^2 \cdot \rho[\mathbf{x}]. \quad (5)$$

The second one compares the norms of the vector fields:

$$\ell_{\mathcal{V}}^2(\mathbf{v}, \mathbf{v}^*) \stackrel{\text{def}}{=} \frac{\lambda_{\mathcal{V},2}}{|\mathcal{X}|} \sum_{\mathbf{x} \in \mathcal{X}} (\|\mathbf{v}[\mathbf{x}]\|_2 - \|\mathbf{v}^*[\mathbf{x}]\|_2)^2 \cdot \rho[\mathbf{x}]. \quad (6)$$

The third one aims to minimize the distance between trajectories generated through the ground truth and predicted flows. Trajectories starting from an initial point \mathbf{z} can be generated by simple explicit Euler discretization:

$$\begin{aligned} \mathbf{x}_0(\mathbf{z}) &\stackrel{\text{def}}{=} \mathbf{z} \\ \mathbf{x}_{l+1}(\mathbf{z}) &\stackrel{\text{def}}{=} \mathbf{x}_l(\mathbf{z}) + \Delta t \cdot \mathbf{v}[\mathbf{x}_l(\mathbf{z})] \\ \mathbf{x}_0^*(\mathbf{z}) &\stackrel{\text{def}}{=} \mathbf{z} \\ \mathbf{x}_{l+1}^*(\mathbf{z}) &\stackrel{\text{def}}{=} \mathbf{x}_l^*(\mathbf{z}) + \Delta t \cdot \mathbf{v}^*[\mathbf{x}_l^*(\mathbf{z})]. \end{aligned}$$

where Δt is a step-size. Letting $L \in \mathbb{N}$ denote an integration time, the ‘‘Euler’’ loss then becomes:

$$\ell_{\mathcal{V}}^3(\mathbf{v}, \mathbf{v}^*) \stackrel{\text{def}}{=} \frac{\lambda_{\mathcal{V},3}}{|\mathcal{X}|} \sum_{\mathbf{z} \in \mathcal{X}} \sum_{l=1}^L \|\mathbf{x}_l(\mathbf{z}) - \mathbf{x}_l^*(\mathbf{z})\|_2^2. \quad (7)$$

It measures how two trajectories generated by Euler integration using the ground truth and predicted vector

fields deviate. This loss is implemented in the torchVF library by Peters (2022). For more information, we refer the reader to the related report.

An inspection of the code reveals that the different weights have been set empirically as: $\lambda_{\mathcal{B}} = 10$, $\lambda_{\mathcal{D}} = 2$, $\lambda_{\mathcal{V},1} = 2$, $\lambda_{\mathcal{V},2} = 2$, $\lambda_{\mathcal{V},3} = 1$.

Remark 3 *The different losses have probably been combined by trial and error to produce the best possible results. However, there are clear redundancies in the definitions of the losses, for instance $\ell_{\mathcal{V}}^1$, $\ell_{\mathcal{V}}^2$ and $\ell_{\mathcal{V}}^3$ are all measuring the distance between flows using different metrics. In our implementation, we tried to simplify the losses as much as possible, while still maintaining a good performance.*

2.3 Adapting to partial annotations

All the principles described above heavily depend on an exhaustive segmentation of the cells. Indeed, the distance functions and gradient flows – which are instrumental to define the loss functions – are global properties which do change heavily if the objects boundaries are incomplete. In this section, we describe the main methodological contribution of this paper, which will allow us to handle partial boundaries.

2.3.1 The gold standard

The notations are summarized in Table 1. We assume that the domain $\mathcal{X} = \mathcal{X}_0 \sqcup \mathcal{X}_1$ is partitioned with the background set \mathcal{X}_0 and the foreground set \mathcal{X}_1 . A difficulty in instance segmentation is that multiple objects may exist within the connected components of a region \mathcal{X}_i . To differentiate them, we let $(\mathcal{X}_{i,j})_{1 \leq j \leq J_i}$ denote a partition of the set \mathcal{X}_i as different objects within a similar class. For instance in

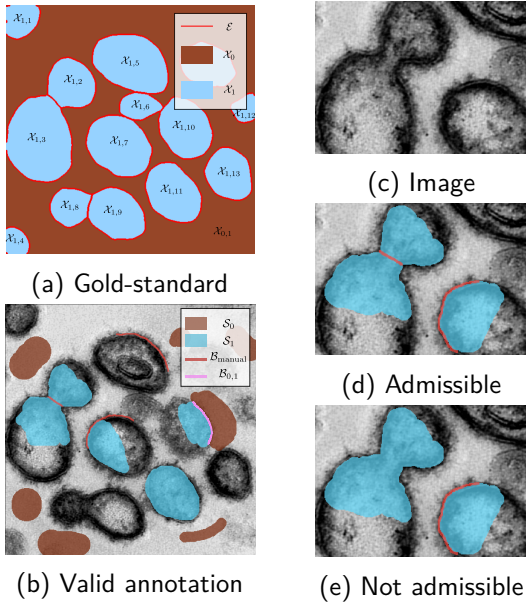


Figure 3: (Left) Ground-truth and an admissible annotation set. (Right) If a stroke contains multiple objects, the object boundaries have to be drawn. In this example, two nuclei are present under the blue bow-tie-shaped region. Therefore, a manual boundary has to be added in the center.

Table 1: Summary of notations.

Notation	Description
$\mathcal{X} = \mathcal{X}_0 \cup \mathcal{X}_1$	Image domain
\mathcal{X}_0	True background
\mathcal{X}_1	True foreground
\mathcal{S}_0	Background strokes
\mathcal{S}_1	Foreground strokes
\mathcal{E}	True boundaries
$\mathcal{B} = \mathcal{B}_{\text{manual}} \cup \mathcal{B}_{0,1}$	User-defined boundaries
\mathcal{D}	Valid distance set

Figure 3a, the foreground set \mathcal{X}_1 is split in 13 components. A connected component of \mathcal{X}_1 can be split as $\mathcal{X}_{1,2} \cup \mathcal{X}_{1,3}$. The background set \mathcal{X}_0 is split in a single component $\mathcal{X}_{0,1}$.

We let

$$\mathcal{E} \stackrel{\text{def}}{=} \bigcup_{i \in \{0,1\}} \bigcup_{j=1}^{J_i} \partial \mathcal{X}_{i,j} \quad (8)$$

denote the set of all *edges* (or object boundaries) within the image. It is depicted in red in Figure 3a.

2.3.2 The annotation set

The input of our neural network is a set of “sketches” or strokes drawn by the user. We let \mathcal{S}_0 and \mathcal{S}_1 denote the strokes describing the background and foreground respectively. They are depicted in brown and blue respectively in

Figure 3b. The intersection of the brown and blue strokes define natural boundaries. We can indeed construct the *touching boundaries* $\mathcal{B}_{0,1}$ between different strokes as

$$\mathcal{B}_{0,1} \stackrel{\text{def}}{=} \overline{\mathcal{S}_0} \cap \overline{\mathcal{S}_1},$$

where $\overline{\mathcal{X}}$ is the closure of \mathcal{X} in the continuous setting and the interface between neighboring pixels in the discrete setting.

In addition, the user can delineate other boundaries, denoted $\mathcal{B}_{\text{manual}}$, to separate touching objects within a class. We can concatenate all the boundaries to obtain a complete boundary set \mathcal{B} defined as

$$\mathcal{B} = \mathcal{B}_{\text{manual}} \cup \mathcal{B}_{0,1}. \quad (9)$$

For the algorithm to work properly, we require the following set of assumptions.

Assumption 1 (Assumptions on the strokes)

- *The strokes correctly separate the background and foreground: $\mathcal{S}_0 \subseteq \mathcal{X}_0$ and $\mathcal{S}_1 \subseteq \mathcal{X}_1$.*
- *The strokes do not overlap: $\mathcal{S}_0 \cap \mathcal{S}_1 = \emptyset$. This is actually forced by our Napari interface.*
- *The boundaries \mathcal{B} are a subset of the exact boundaries \mathcal{E} , that is:*

$$\mathcal{B} \subseteq \mathcal{E}. \quad (10)$$
- *If the stroke \mathcal{S}_i contains multiple objects, then the boundaries between the objects need to be completely drawn with $\mathcal{B}_{\text{manual}}$ (see Figure 3d). Letting $\mathcal{S} \stackrel{\text{def}}{=} \mathcal{S}_0 \cup \mathcal{S}_1$ denote the complete stroke set drawn by the user, this condition reads:*

$$\mathcal{E} \cap \mathcal{S} \subseteq \mathcal{B}. \quad (11)$$

2.3.3 The main observation

The main result we will use to define and certify our algorithm is summarized in the following theorem.

Theorem 4 (The valid distance set) *Let $\mathcal{CB} \stackrel{\text{def}}{=} \partial \mathcal{S}_0 \cup \partial \mathcal{S}_1 \cup \mathcal{B}$ denote the complete set of annotation boundaries and define the valid distance set \mathcal{D} as*

$$\mathcal{D} \stackrel{\text{def}}{=} \{x \in \mathcal{S}, \text{dist}(x, \mathcal{B}) \leq \text{dist}(x, \mathcal{CB})\}. \quad (12)$$

The following relationships hold:

$$\begin{aligned} \text{For all } x \in \mathcal{D}, \quad & \text{dist}(x, \mathcal{E}) = \text{dist}(x, \mathcal{B}). \\ \text{For all } x \in \mathcal{S}_0 \cup \mathcal{S}_1, \quad & \text{dist}(x, \mathcal{CB}) \leq \text{dist}(x, \mathcal{E}). \end{aligned}$$

The proof of this theorem is given in Appendix A. This theorem should be understood as follows. The first identity informs us that we can compute the *exact distance map* $\text{dist}(x, \mathcal{E})$ to the set of exact boundaries \mathcal{E} on the valid set \mathcal{D} . This set can be computed using only the partial annotations of the boundaries $\mathcal{B} \subseteq \mathcal{E}$ and the different semantic regions $\mathcal{S}_i \subseteq \mathcal{X}_i$. The second inequality tells us that we have an information everywhere on the strokes \mathcal{S}_0 and \mathcal{S}_1 . Moreover, in the case of total annotations, we get $\mathcal{D} = \mathcal{X}$ and the proposed idea will lead to a training equivalent to the one in Omnipose and we can see the proposed setting as a generalization. Figure 4 schematically summarizes Theorem 4.

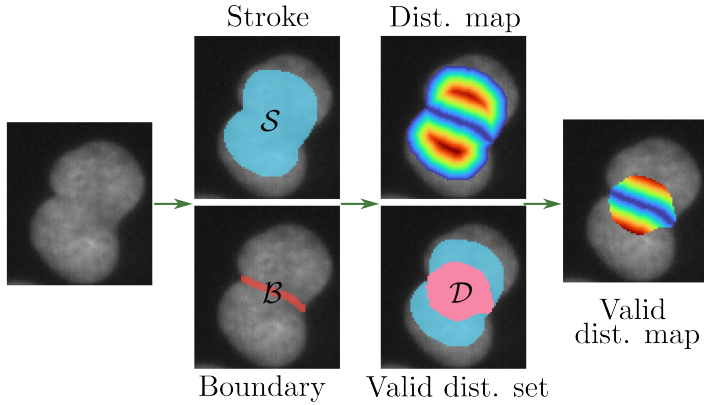


Figure 4: Illustration of the valid distance set theorem. All the pixels in \mathcal{D} are closer to \mathcal{B} than to the boundaries of \mathcal{S} . The colormap used to represent the distance map exhibits a progressive shift in colors, transitioning from blue to red.

2.3.4 Adapting the training

A simpler architecture The prediction of the boundaries $N_w^b(u)$ is not necessary, since only the distance map $N_w^d(u)$ and the flow field $N_w^y(u)$ are needed to compute the final masks. Hence, we keep the same U-Net architecture, but remove the channel associated to the boundaries:

$$N_w^{\text{sketch}} : \mathbb{R}^N \rightarrow \mathbb{R}^N \times \mathbb{R}^{2N} \quad (13)$$

$$u \mapsto (N_w^d(u), N_w^y(u))$$

Different summation sets Equipped with the valid distance set \mathcal{D} , we are ready to adapt the losses to cope with partial annotation. In Omnipose, the losses $\ell_{\mathcal{B}}$, $\ell_{\mathcal{D}}$, $\ell_{\mathcal{V}}^1$ and $\ell_{\mathcal{V}}^2$ are defined by summation over the set \mathcal{X} (see paragraph 2.2.3). With partial annotation, the gold standard is not properly defined on this set and we therefore need to change the summation sets.

Based on Theorem 4, the losses related to the distance

set and to the flows become:

$$\ell_{\mathcal{D}}^{\text{partial}}(d, d^*) \stackrel{\text{def}}{=} \frac{\lambda_{\mathcal{D}}}{|\mathcal{D}|} \sum_{\mathbf{x} \in \mathcal{D}} (d[\mathbf{x}] - d^*[\mathbf{x}])^2,$$

$$\ell_{\mathcal{V}}^{\text{partial}}(\mathbf{v}, \mathbf{v}^*) \stackrel{\text{def}}{=} \frac{\lambda_{\mathcal{V},1}}{|\mathcal{D}|} \sum_{\mathbf{x} \in \mathcal{D}} \|\mathbf{v}[\mathbf{x}] - \mathbf{v}^*[\mathbf{x}]\|_2^2.$$

We replaced $\sum_{\mathbf{x} \in \mathcal{X}}$ by $\sum_{\mathbf{x} \in \mathcal{D}}$ in section 2.2.3 and discarded the weights ρ . This means that we compare the ground truth and prediction only where it makes sense to do so.

Dealing with inequalities Until now, we just used the first identity in Theorem 4, but the second inequality brings some additional information. We propose to integrate it in the training through the additional asymmetric loss:

$$\ell_{\mathcal{D}}^{\text{ineq}}(d, d^*) \stackrel{\text{def}}{=} \frac{\lambda_{\mathcal{D},\text{ineq}}}{|\mathcal{S}_1 \cup \mathcal{S}_2|} \sum_{\mathbf{z} \in \mathcal{S}_1 \cup \mathcal{S}_2} \text{ReLU}^2(d[\mathbf{z}] - d^*[\mathbf{z}]). \quad (14)$$

Putting it all together We can now define the total sketchpose loss as:

$$\text{loss}^{\text{sketch}}(w) = \ell_{\mathcal{V}}^{\text{partial}}(\mathbf{v}, \mathbf{v}^*) + \ell_{\mathcal{D}}^{\text{partial}}(d, d^*) + \ell_{\mathcal{D}}^{\text{ineq}}(d, d^*) \quad (15)$$

where d, \mathbf{v} are obtained by the simplified neural network N_w^{sketch} .

2.4 The Sketchpose plugin

A significant part of this work lies in the development of a user-friendly graphical interface to train and use the neural network. It is integrated in Napari Chiu et al. (2022), which is well suited to embedding the Python/PyTorch codes at the core of our approach.

Sketchpose can be easily installed through either the pip package manager or Napari’s Chiu et al. (2022) built-in interface. A detailed documentation can be accessed by clicking on this hyperlink. It offers step-by-step instructions illustrated by short videos, to assist users in effectively testing all the capabilities of the plugin.

The user directly draws a few strokes for the background, foreground and boundaries. The brush size can be adjusted similarly to usual paint software. An entire stroke boundary can be added to the boundary set by a double right-click. The training and drawing can be achieved in parallel to target the places where the segmentation is inaccurate.

The networks can be initialized by random weights or existing pre-trained weights. The multi-threaded plugin’s architecture makes it possible to annotate, train and observe the current segmentation results simultaneously. The user can annotate regions where the segmentation is inaccurate in priority, hence reducing the annotation time. The predictions can be restricted to a bounding box at each epoch of the training process, to reduce the processing time, which

is particularly helpful for large scale images. Finally, users can work with a single image or a set of images for the inference and training steps.

3. Experiments

In this paragraph, we conduct several experiments to explore three distinct use cases of the method.

- Learning from a limited set of annotations on a single image with randomly initialized neural network weights.
- Learning from a limited set of annotations on a single image, starting from a pre-trained neural network.
- Learning with randomly initialized weights using a large dataset with sparse annotations. We study the impact of the percentage of labeled pixels (10%, 25%, 50% and 100%) on the segmentation quality, when training on thousands of cells.

After describing the metrics used for validation, we will turn to the practical results. For all the experiments, we used a single Nvidia RTX5000 GPU with 16 Gb.

3.1 Evaluation metrics

To quantify the predictions quality, we enumerate the true positives (TP), the true negatives (TN) and the false positives (FP). A true positive is an object in the gold standard that can be matched to an object in the prediction with an Intersection over Union (IoU) criterion higher than a threshold τ . We let $TP(\tau)$ denote the total number of true positives. The total number of estimated objects without matches is denoted $FP(\tau)$ (for false positives). The total number of gold standard objects without valid matches is denoted $FN(\tau)$ (for false negatives). Utilizing these values, we compute the object detection accuracy metric ($OA(\tau)$) Caicedo et al. (2019) for each image using the formula:

$$OA(\tau) = \frac{TP(\tau)}{TP(\tau) + FP(\tau) + FN(\tau)}.$$

The reported object detection accuracy is then computed as the average over all images in the test set.

Additionally, we computed the average DICE and the Aggregated Jaccard Index defined as follows:

$$J_{\text{aggregated}}(A, B) = \frac{1}{N} \sum_{i=1}^N \frac{|A_i \cap B_i|}{|A_i \cup B_i|},$$

$$\text{average DICE}(A, B) = \frac{1}{N} \sum_{i=1}^N \frac{2 \cdot |A_i \cap B_i|}{|A_i| + |B_i|},$$

where A and B are a dataset and its groundtruth.

3.2 Training from scratch on a single image

In this section, we will showcase several results achieved while training from scratch on a small set of images. The tests are made on a variety of biological structures (dendritic cells, osteoclasts, bacteria, insect eggs, adipose tissue, artistic image of cells).

3.2.1 Training details

For this experiment, the model have been trained for 100 epochs (≈ 2 minutes) with a batch size of 16 and image flips for data augmentation.

3.2.2 *Staphylococcus aureus*

In the example of Figure 5, we use a microscopy image of methicillin-resistant *Staphylococcus aureus* (MRSA) infections, from European Commission, *Horizon Magazine* (2020), “Can we reverse antibiotic resistance?”. It is reused under the European Commission’s reuse policy.

After drawing for less than one minute and training for 100 epochs ($\approx 2'$), we achieve a much better result than the trained model of Omnipose (see Figure 5). The quality metrics is shown in Figure 5d.

3.2.3 Eggs on a tree leaf

In this section, we picked an image from the Omnipose dataset, which likely represents eggs of an insect on a tree leaf. At first sight, the segmentation task is uneasy, since the objects are tightly connected, with identical textures and blurry boundaries. We first annotated a subset of 5 eggs in Figure 6b with a minimal amount of background. The segmentation result after training is already surprisingly good in Figure 6d, but some objects are not detected, and others are merged. We annotated 2 extra eggs in Figure 6c. With this extra information, retraining the network now produces a near perfect segmentation mask, with a single error (2 pink eggs on the left). This experiment illustrates a unique feature of Sketchpose: it is possible to interactively annotate while training. This offers a possibility to label a minimum amount of regions to reach the desired output. This principle sometimes called “active learning” or “human-in-the-loop” Budd et al. (2021) is significantly enhanced by using partial annotations and the user-friendly Napari interface.

3.3 Transfer learning on a single image

In this section, we explore the feasibility of improving pre-trained weights using transfer learning.

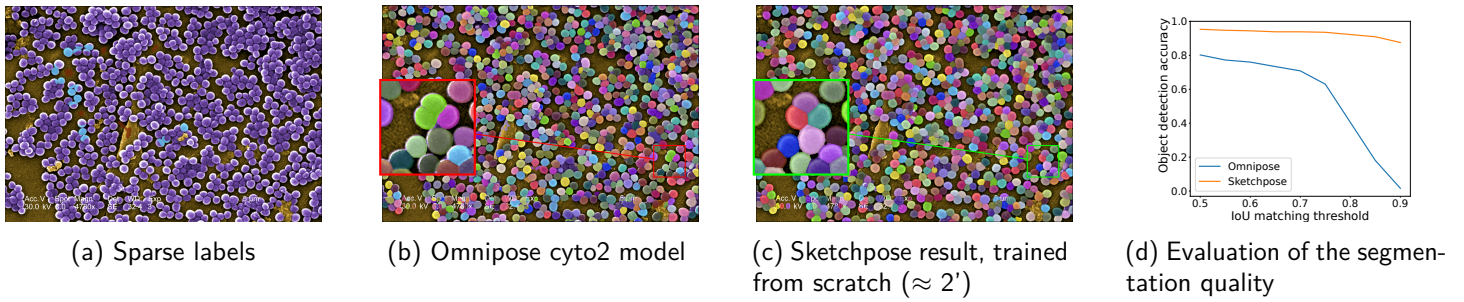


Figure 5: (a,b,c) A training from scratch with sparse labels (in blue on the left image). Image credit: Janice Carr, Jeff Hageman, USCDCP. (d) Evaluation of segmentation quality. Omnipose results are: DICE=0.92, Jaccard index=0.82. Sketchpose: DICE=0.99, Jaccard index=0.96.

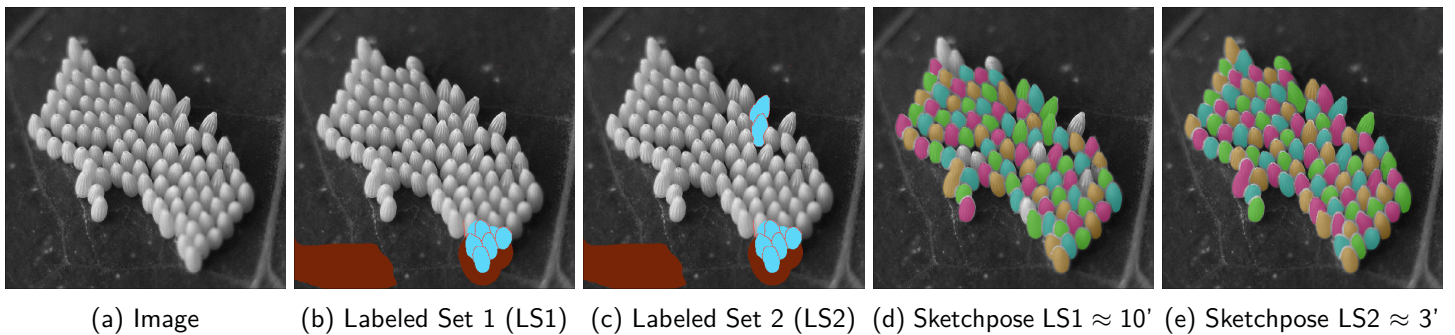


Figure 6: Progressive training in Sketchpose. In this example, we show that it is possible to improve the segmentation performance of Sketchpose by progressively annotating at places where the network failed. Here, a quite minimal annotation set is enough to near perfectly separate the eggs on the leaf.

3.3.1 Training details

As for the previous experiment, the model have been trained for 100 epochs (≈ 2 minutes) with a batch size of 16 and image flips for data augmentation.

3.3.2 Bacteria segmentation

Bacteria are often used as biological models (e.g. in DNA studies). A precise segmentation can be difficult to achieve, because they have elongated shapes and can be clustered.

The Omnipose Cutler et al. (2022) model was initially conceived to address the shortcomings of Cellpose for this task.

Figure 7 shows how transfer learning with sparse annotations can improve the Omnipose results by separating touching bacterias. Figure 7d shows a quantitative comparison of both methods. As can be seen, Sketchpose's adapted weights provide much higher performance. A visual inspection indicates that all objects have been correctly separated, apart from the cluster touching the boundary on the bottom left.

3.3.3 Adipocytes segmentation

The image in Figure 7e shows a crop of a very large image of a skin explant provided by DIVA Expertise. One can see

a part of the dermis (in pink) and above it, adipose tissue (large white circular cells). Adipose tissue is the third skin layer after the epidermis and dermis, also known as the hypodermis. Hypodermal cells (adipocytes) secrete specific molecules (e.g. adiponectin, leptin) which have a direct impact on the biology of fibroblasts present in the dermis, and also on keratinocytes present in the epidermis. They are the subject of numerous studies (see Bourdens et al. (2019) and Sadick et al. (2015) for instance). For most of the studies where skin explants are imaged, we first need to count the adipocytes number in the image, and remove any potential outliers detected in the dermis and epidermis.

While Omnipose cyto2 results in some undersegmentation for this task, the adapted weights provided by Sketchpose yields significantly enhanced results. Annotating 6 cells and a training for 100 epochs (less than 1 minute) were sufficient to significantly improve the quality of the segmentation and to remove the outliers from the dermis (see Figure 7c). Figure 7 shows a quantitative comparison between Omnipose and Sketchpose on this example.

3.3.4 Osteoclasts segmentation

Osteoclasts are responsible for bone resorption, and are widely studied (see Labour et al. (2016) for instance) as being responsible for certain pathologies such as osteoporosis.

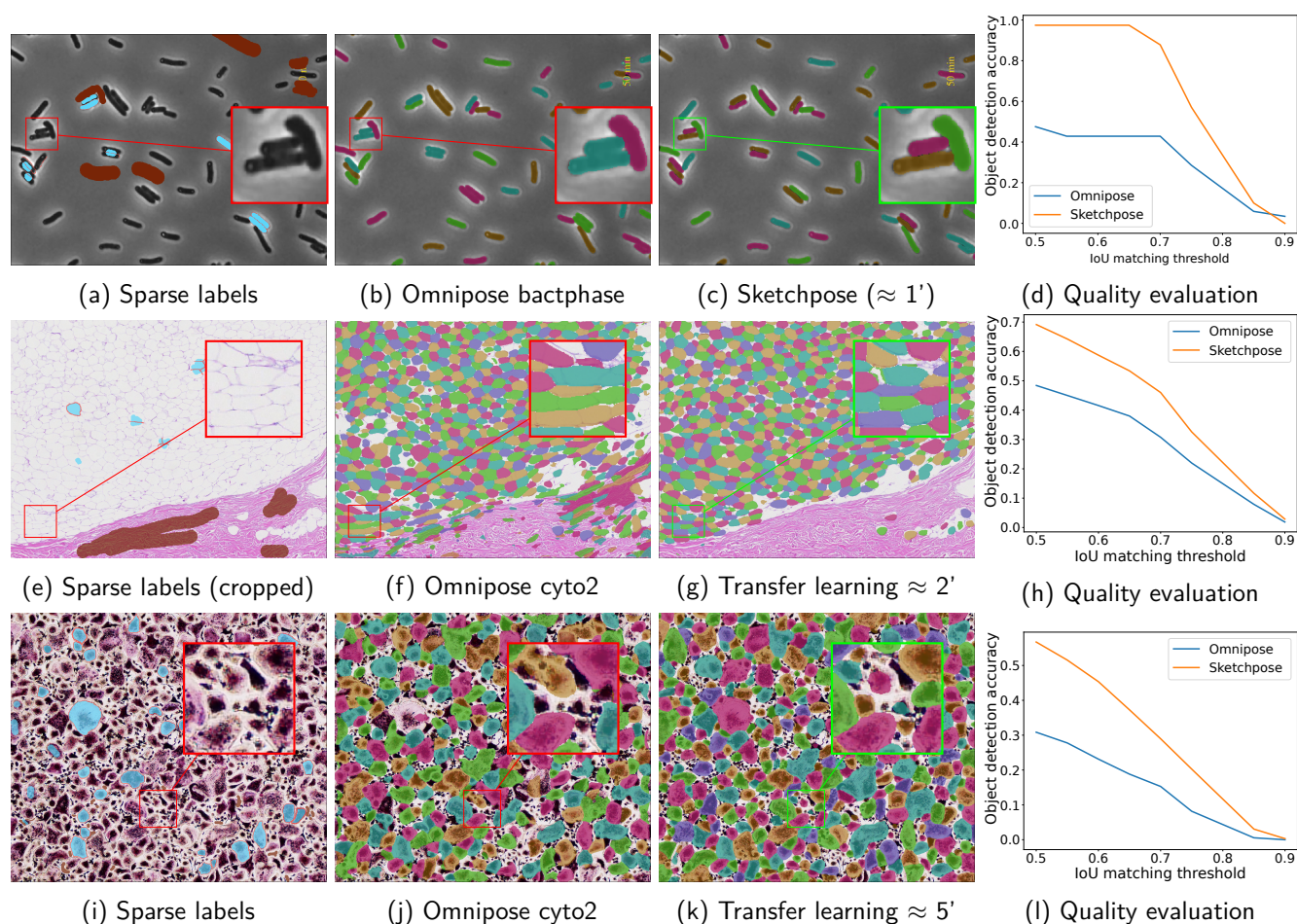


Figure 7: Transfer learning experiment. In all cases, just a few strokes are enough to significantly improve the segmentation quality. **(a–d) Bacteria.** (a–c) Training with a few sparse labels. (d) Evaluation of segmentation quality. Omnipose: DICE = 0.81, Jaccard = 0.53. Sketchpose: DICE = 0.90, Jaccard = 0.72. **(e–h) Adipocytes.** (e–g) Transfer learning from the Omnipose bact_phase model. (h) Evaluation of segmentation quality. Omnipose: DICE = 0.89, Jaccard = 0.69. Sketchpose: DICE = 0.89, Jaccard = 0.79. Image adapted from Zhu et al. (2021). **(i–l) Osteoclasts.** (i–k) Transfer learning from the Omnipose cyto2 model. (l) Evaluation of segmentation quality. Omnipose: DICE = 0.90, Jaccard = 0.68. Sketchpose: DICE = 0.95, Jaccard = 0.80.

sis when dysfunctional. Their differentiation goes through several stages, culminating in the activated osteoclast. The latter is generally large and contains numerous nuclei. Atlantic Bone Screen (ABS) company is investigating the effect of different drugs in inducing either proliferation or cell death in these activated osteoclasts, in order to regulate their population. To do so, they extract osteoclasts from biopsies, culture them, apply the drugs and image them under a bright-field microscope.

The studied image is a crop of an image containing around 20,000 cells. We can see touching cells presenting a great variety in size, shape color. The image is complex to segment and poses a real challenge. What is more, ABS does not want to count pre-osteoclasts (small black nuclei), but only the mature cells (according to specific nuclei criteria). Each study comprises around sixty images, hence manual counting task performed at ABS is costly and laborious.

In Figure 7, we present a qualitative depiction that underscores the enhancement in segmentation accuracy attained through transfer learning with just a few labels. Labeling required approximately 2 minutes, while the training process took about 5 minutes. A quantitative comparison is available in Figure 7l.

3.4 Training from scratch on large datasets

The aim of this experiment is to highlight the possibility to train our model on large datasets with sparse annotations. We use two different datasets as illustrated in Figure 8.

Microbeseg dataset The *Microbeseg* dataset Scherr et al. (2022) contains 826 fluorescent microscopy images of bacteria with about 30,000 manually annotated objects). It contains a mix of datasets from Omnipose and the Cell Tracking Challenge. It is publicly available here.

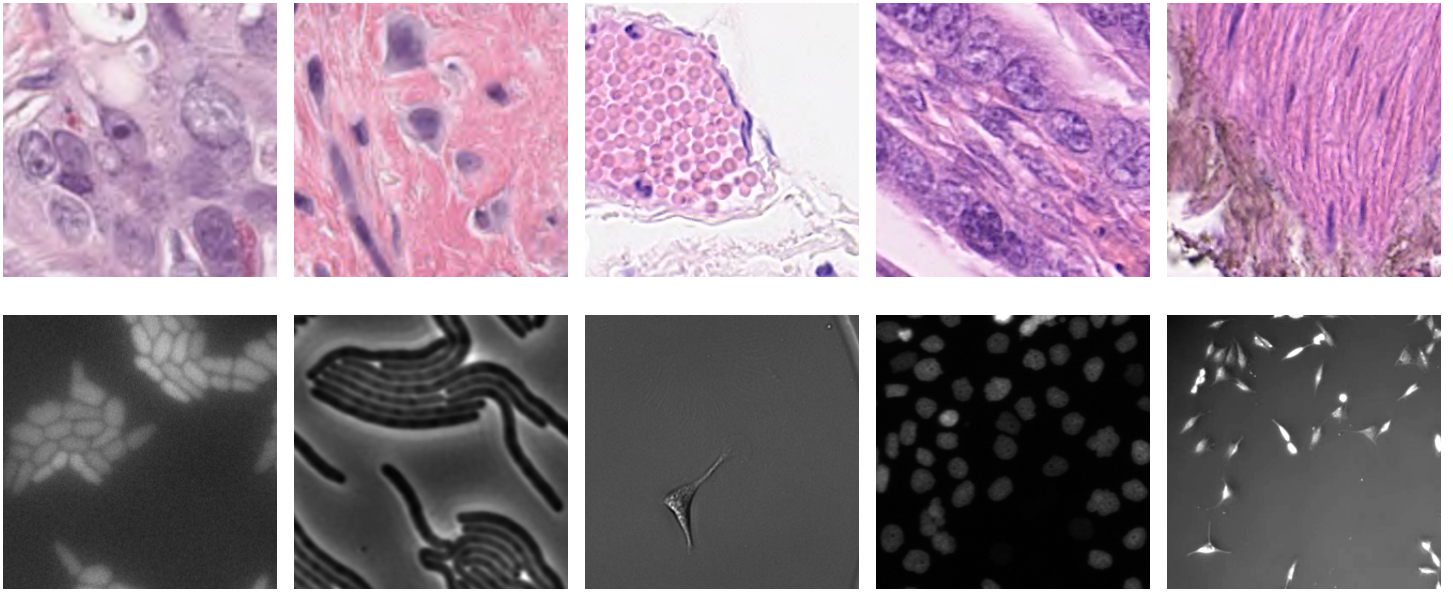


Figure 8: Top: examples of PanNuke images. Bottom: examples of MicrobeSeg images.

PanNuke dataset The PanNuke dataset Gamper et al. (2019) contains 7,904 image tiles of histopathology slides stained with H&E across 19 tissue types, each with nuclear instance segmentations and five-class nuclear type annotations. It is publicly available from Kaggle and is widely used for benchmarking nucleus segmentation and classification algorithms.

3.4.1 Selecting annotation subsets

In this section, we investigate the model robustness across various annotation levels each characterized by a different percentage of annotated pixels: 10%, 25%, 50%, and 100%. We generate randomly binary masks by thresholding white Gaussian noise with a Gaussian filter of variance σ^2 . The resulting Gaussian process is then thresholded to keep only a given proportion of pixels.

While the model is stochastic in nature, the generated data is created once and for all, enabling its deterministic reuse across multiple training sessions. Figure 9 shows an image accompanied by four corresponding label masks illustrating decreasing levels of annotation sparsity.

3.4.2 Training details

For each dataset, we trained the Sketchpose model for 1000 epochs. This takes about 5 hours for the Microbeseg dataset and 30 hours for the PanNuke dataset using our Nvidia RTX5000. Each model was trained with the four percentages of annotated pixels we described above. There was no data augmentation, except random cropping of the images to 224x224 pixels. This is the size which was used in the original Omnipose model.

3.4.3 Results

We evaluated and compared the performance using two alternative models. The first one is the Cellpose 3.0 model Stringer and Pachitariu (2025), which regresses a distance to the objects centroids. The second one is the LKCell model Cui et al. (2024), which is a better performing variant of CellVit Hörst et al. (2024), itself a variant of HoverNet Graham et al. (2019). These models are among the most popular and best performing for histopathology images such as the PanNuke dataset. While they perform classification and segmentation, we will just compare their ability to segment objects, as we are not interested in the classification task here. We trained both LKCell and Cellpose 3.0 from scratch on each of the two datasets with complete annotations for 1000 epochs.

We compare the performance using other standard quality metrics used in instance segmentation. In all the metrics below, an IoU threshold of 50% is used.

- **Precision:** measures how many of the predicted positives are actually correct (i.e., the fraction of predicted segments that are true).
- **Recall:** how many of the actual positives were correctly predicted (i.e., how complete the prediction is).
- **F1-Score:** harmonic mean of precision and recall, balancing both.
- **Detection Quality (DQ):** evaluates object-level detection performance, penalizing missed or extra objects. It evaluates the ability to detect object instances correctly, regardless of segmentation quality.
- **Segmentation Quality (SQ):** measures how well matched

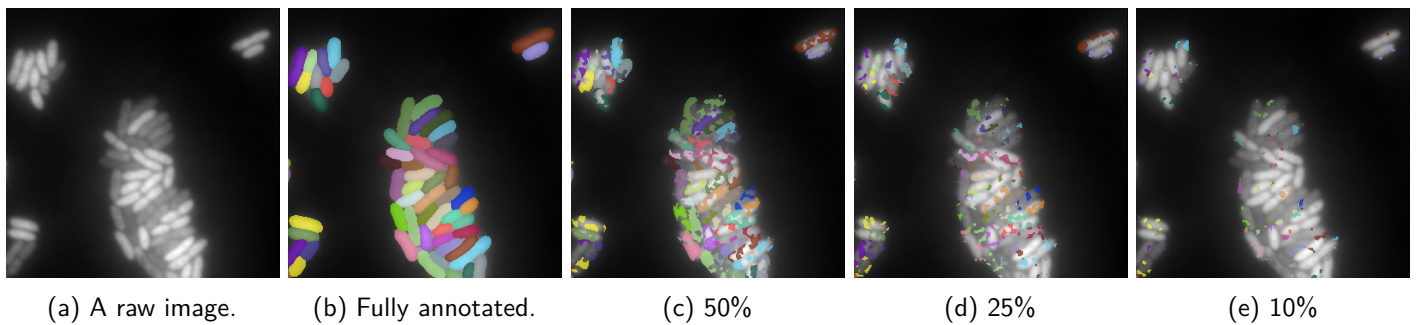


Figure 9: Annotations example with four different sparsity levels and a domain regularity $\sigma = 50$.

objects are segmented, assuming correct pairing, reflecting the quality of the predicted segment.

The main results are reported in Table 2, and several key observations emerge.

Best-performing methods On the MicrobeSeg dataset, Cellpose and Sketchpose (100% annotations) deliver the best performance, while LKCell lags behind with a 15% lower F1-score. Conversely, on PanNuke, LKCell outperforms both competitors with a 4–6% F1-score gain, confirming its suitability for this dataset.

Impact of annotation density For MicrobeSeg, reducing annotation density leads to a significant performance drop for Sketchpose: around 10% at 50–25% annotations, and up to 20% with only 10%. Depending on the application, such degradation may or may not be acceptable.

The situation is more favorable for PanNuke. Sketchpose maintains stable performance, with only a 4% drop when reducing annotations from 100% to 10%. Given that sparse annotations likely reduce annotation time by a factor of ten, this is a promising result—especially since random sampling was used. In practice, targeted annotations by an expert would likely yield even better outcomes.

This contrast may stem from dataset characteristics: PanNuke contains simpler, roughly convex objects, while MicrobeSeg features elongated or irregular shapes, making annotation density more critical.

IoU matching thresholds To provide a refined view of the performance, we also plot the F1-score as a function of the IoU matching threshold in Figure 10. There, we see that the ranking between the methods is stable up to a matching threshold of 70%, which is usually considered a high precision segmentation in biological imaging. A surprising phenomenon is that Sketchpose trained with 25% of annotations performs better than the 50% model on the MicrobeSeg dataset. Similarly, the 50% model performs better than the 100% model on the PanNuke dataset. This might indicate that carefully selected annotations can lead to better results than complete annotation, or helps reducing the influence of errors in the gold-standard database.

Sketchpose is the first distance-based method allowing to take advantage of this observation.

4. Discussion & conclusion

We introduced Sketchpose, an open-source plugin to extend the applicability of Omnipose to partial annotations. From a methodological aspect, we developed a theory making it possible to use distance functions, despite having only access to partial information on the objects boundaries. From a more practical viewpoint, we developed an interactive interface within Napari, which facilitates efficient online learning with a real-time visualization of the training progress. The multi-threaded implementation allows users to continue annotating while the neural network trains or infers.

The new training procedure was tested in three different frames: i) training a neural network from scratch and just a few strokes, ii) improving the weights of a pre-trained network (a.k.a. transfer learning or human in the loop), iii) training with massive, but partial annotations.

For point i), frugal annotation works surprisingly well on a few test cases despite really limited information. A dozen strokes are already enough to provide results on par – or better – than pre-trained networks.

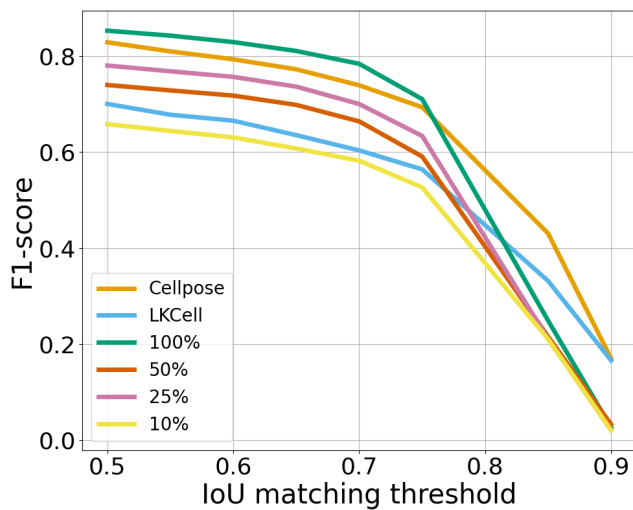
For point ii), our experiments demonstrated the potential benefits of using transfer learning. That is, starting with a pre-trained Omnipose models, we can further refine it using our methodology.

As for point iii), the conclusions are diverse. For datasets containing simple object shapes, such as PanNuke, it seems that a limited number of annotations (down to 25%) is sufficient to achieve results on par or even better than complete annotations. For more complex objects, it seems that complete annotations are still preferable. These conclusions should be validated on a case by case basis, but the ability to annotate while training make it possible to take the minimal amount of annotation time for a given task.

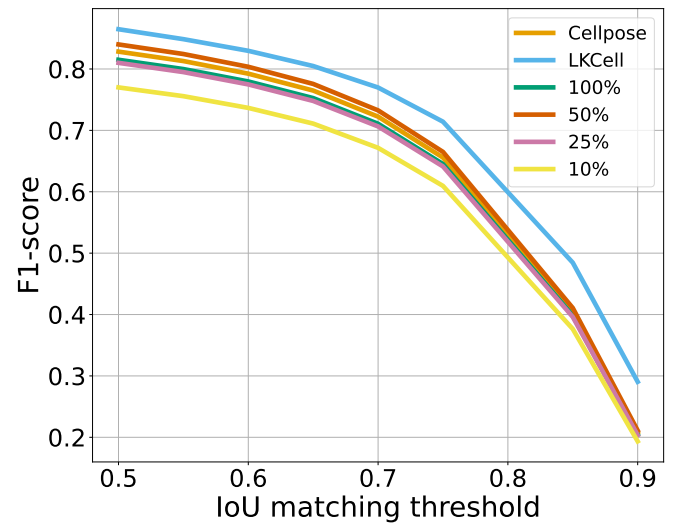
The method also shows a few limitations. First, it would benefit from faster training times to make the method even more interactive. We plan to improve this aspect in the

Table 2: Comparison of segmentation methods trained and tested on the MicrobeSeg and PanNuke datasets using various metrics. All values are computed with an IoU threshold of 50%.

	<i>MicrobeSeg dataset</i>						<i>PanNuke dataset</i>					
	LKCell	Cellpose	100%	50%	25%	10%	LKCell	Cellpose	100%	50%	25%	10%
Precision	0.73	0.88	0.77	0.63	0.68	0.55	0.82	0.88	0.86	0.84	0.78	0.75
Recall	0.65	0.75	0.89	0.86	0.83	0.78	0.84	0.71	0.71	0.75	0.77	0.73
F1-Score	0.66	0.79	0.81	0.69	0.72	0.60	0.83	0.78	0.77	0.79	0.77	0.73
DICE	0.82	0.86	0.85	0.85	0.83	0.80	0.88	0.87	0.87	0.88	0.87	0.87
Jaccard	0.73	0.79	0.75	0.75	0.72	0.69	0.81	0.79	0.79	0.80	0.79	0.77
Det. Quality (DQ)	0.73	0.88	0.77	0.63	0.68	0.55	0.82	0.88	0.86	0.84	0.78	0.75
Seg. Quality (SQ)	0.73	0.79	0.75	0.75	0.72	0.69	0.81	0.79	0.79	0.80	0.79	0.77



(a) F1-score over the Microbeseg test dataset.



(b) F1-score over the PanNuke test dataset.

Figure 10: Evaluation of segmentation F1-score over Microbeseg’s and PanNuke’s test datasets as a function of the percentage of annotated pixels compared to Cellpose and LKCell.

forthcoming versions. Second, it is important to mention that our formalism is currently restricted to the two dimensional setting with two labels (background / foreground). Extending the methodology to numerous classes is rather straightforward, and the proposed ideas extend directly to this case. However, the proposed strategy do not extend to 3D directly. It could be used if the user was able to delineate a surface surrounding the objects of interest, but not just curves in 2D. Indeed, this would result in an empty valid distance set (see Theorem 4) and unadapted loss functions. This limitation of the method must be put into perspective by the fact that even the Cellpose 3D model is based on 2D predictions only, which are aggregated in post-processing.

In summary, the proposed method demonstrated numerous qualities in 2D for partial annotations. We showed that it is possible to train complex networks with a few sketches, reducing the annotation burden significantly. Further developments are needed to accelerate the training process and for a multi-class extension in 3D.

Acknowledgments

C. Cazorla was a recipient of ANRT (Agence Nationale pour la Recherche et la Technologie) in the context of the CIFRE Ph.D. program (N°2020/0843) with Imactiv-3D and Institut de Mathématiques de Toulouse (IMT). P. Weiss acknowledges a support from ANR-3IA Artificial and Natural Intelligence Toulouse Institute ANR-19-PI3A-0004 and from the ANR Micro-Blind ANR-21-CE48-0008. This work was performed using HPC resources from GENCI-IDRIS (Grant 2021-AD011012210R1).

We are grateful for the information provided by Kevin John Cutler about the original Omnipose implementation. The authors acknowledge Atlantic Bone Screen for providing the osteoclasts image and DIVA Expertise for providing the adipocytes image.

Ethical Standards

The work follows appropriate ethical standards in conducting research and writing the manuscript, following all applicable laws and regulations regarding treatment of animals or human subjects.

Conflicts of Interest

We declare we don't have conflicts of interest.

Data availability

The code is available there: <https://bitbucket.org/koopaa31/napari-sketchpose/src/master/>. The documentation is available there: <https://sketchpose-doc.readthedocs.io/en/latest/>.

References

- Matthias Arzt, Joran Deschamps, Christopher Schmied, Tobias Pietzsch, Deborah Schmidt, Pavel Tomancak, Robert Haase, and Florian Jug. Labkit: labeling and segmentation toolkit for big image data. *Frontiers in computer science*, 4:10, 2022.
- Min Bai and Raquel Urtasun. Deep watershed transform for instance segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5221–5229, 2017.
- Stuart Berg, Dominik Kutra, Thorben Kroeger, Christoph N Straehle, Bernhard X Kausler, Carsten Haubold, Martin Schiegg, Janez Ales, Thorsten Beier, Markus Rudy, et al. Ilastik: interactive machine learning for (bio) image analysis. *Nature methods*, 16(12):1226–1232, 2019.
- Marion Bourdens, Yannick Jeanson, Marion Taurand, Noémie Juin, Audrey Carrière, Franck Clément, Louis Casteilla, Anne-Laure Bulteau, and Valérie Planat-Bénard. Short exposure to cold atmospheric plasma induces senescence in human skin fibroblasts and adipose mesenchymal stromal cells. *Scientific reports*, 9(1):8671, 2019.
- Samuel Budd, Emma C Robinson, and Bernhard Kainz. A survey on active learning and human-in-the-loop deep learning for medical image analysis. *Medical Image Analysis*, 71:102062, 2021.
- Juan C. Caicedo, Jonathan Roth, Allen Goodman, Tim Becker, Kyle W. Karhohs, Matthieu Broisin, Csaba Molnar, Claire McQuin, Shantanu Singh, Fabian J. Theis, and Anne E. Carpenter. Evaluation of deep learning strategies for nucleus segmentation in fluorescence images. *Cytometry Part A*, 95(9):952–965, 2019. . URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/cyto.a.23863>.
- Chi-Li Chiu, Nathan Clack, et al. napari: a python multi-dimensional image viewer platform for the research community. *Microscopy and Microanalysis*, 28(S1):1576–1577, 2022.
- Ziwei Cui, Jingfeng Yao, Lunbin Zeng, Juan Yang, Wenyu Liu, and Xinggong Wang. Lkcell: Efficient cell nuclei instance segmentation with large convolution kernels. *arXiv preprint arXiv:2407.18054*, 2024.
- Kevin J Cutler, Carsen Stringer, Teresa W Lo, Luca Rappozzi, Nicholas Stroustrup, S Brook Peterson, Paul A Wiggins, and Joseph D Mougous. Omnipose: a high-precision morphology-independent solution for bacterial cell segmentation. *Nature methods*, 19(11):1438–1448, 2022.
- Johanna Gamper, Navid Alemi Koohbanani, Katarzyna Benet, Adnan Khuram, and Nasir Rajpoot. Pannuke: an open pan-cancer histology dataset for nuclei instance segmentation and classification. In *European Congress on Digital Pathology (ECDP)*, pages 11–19. Springer, 2019. . URL https://doi.org/10.1007/978-3-030-23937-4_11.
- Simon Graham, Quoc Dang Vu, Shan E Ahmed Raza, Ayesha Azam, Yee Wah Tsang, Jin Tae Kwak, and Nasir Rajpoot. Hover-net: Simultaneous segmentation and classification of nuclei in multi-tissue histology images. *Medical Image Analysis*, 58:101563, 2019.
- Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.
- Fabian Hörst, Moritz Rempe, Lukas Heine, Constantin Seibold, Julius Keyl, Giulia Baldini, Selma Ugurel, Jens Siveke, Barbara Grünwald, Jan Egger, et al. Cellvit: Vision transformers for precise cell segmentation and classification. *Medical Image Analysis*, 94:103143, 2024.
- Michael Kass, Andrew Witkin, and Demetri Terzopoulos. Snakes: Active contour models. *International journal of computer vision*, 1(4):321–331, 1988.
- Jitendra Kumar, R. Srivastava, R. Srivastava, A. Mehrotra, R. Srivastava, R. Srivastava, and R. Srivastava. Deep learning framework for recognition of diabetic retinopathy using monocular color fundus images. *Computers in Biology and Medicine*, 109:283–293, 2019. .
- Marie-Noëlle Labour, Mathieu Riffault, Søren T Christensen, and David A Hoey. Tgf β 1-induced recruitment of human

- bone mesenchymal stem cells is mediated by the primary cilium in a smad3-dependent manner. *Scientific reports*, 6(1):35542, 2016.
- David Legland, Ignacio Arganda-Carreras, and Philippe Andrey. Morpholibj: integrated library and plugins for mathematical morphology with imagej. *Bioinformatics*, 32(22):3532–3534, 2016.
- Victor Lempitsky, Andrea Vedaldi, and Dmitry Ulyanov. Deep image prior. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9446–9454. IEEE, 2018.
- Peter Naylor, Marick Laé, Fabien Reyal, and Thomas Walter. Segmentation of nuclei in histopathology images by deep regression of the distance map. *IEEE transactions on medical imaging*, 38(2):448–459, 2018.
- Nobuyuki Otsu. A threshold selection method from gray-level histograms. *IEEE transactions on systems, man, and cybernetics*, 9(1):62–66, 1979.
- Marius Pachitariu and Carsen Stringer. Cellpose 2.0: how to train your own model. *Nature methods*, 19(12):1634–1641, 2022.
- Ryan Peters. Torchvf: Vector fields for instance segmentation. 2022.
- Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention—MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer, 2015.
- Neil S Sadick, Andrew S Dorizas, Nils Krueger, and Amer H Nassar. The facial adipose system: its role in facial aging and approaches to volume restoration. *Dermatologic Surgery*, 41:S333–S339, 2015.
- Tim Scherr, Johannes Seiffarth, Bastian Wollenhaupt, Oliver Neumann, Dietrich Kohlheyer, Hanno Scharr, Katharina Nöh, and Ralf Mikut. microbeseg models, 2022. URL <https://zenodo.org/record/7221151>.
- Uwe Schmidt, Martin Weigert, Coleman Broaddus, and Gene Myers. Cell detection with star-convex polygons. In *Medical Image Computing and Computer Assisted Intervention—MICCAI 2018: 21st International Conference, Granada, Spain, September 16-20, 2018, Proceedings, Part II 11*, pages 265–273. Springer, 2018.
- Jean Serra and Pierre Soille. *Mathematical morphology and its applications to image processing*, volume 2. Springer Science & Business Media, 2012.
- Carsen Stringer and Marius Pachitariu. Cellpose3: one-click image restoration for improved cellular segmentation. *Nature methods*, 22(3):592–599, 2025.
- Carsen Stringer, Tim Wang, Michalis Michaelos, and Marius Pachitariu. Cellpose: a generalist algorithm for cellular segmentation. *Nature methods*, 18(1):100–106, 2021.
- Ko Sugawara. Training deep learning models for cell image segmentation with sparse annotations. *BioRxiv*, pages 2023–06, 2023.
- Luc Vincent and Pierre Soille. Watersheds in digital spaces: an efficient algorithm based on immersion simulations. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 13(06):583–598, 1991.
- Lillian Zhu, Manohary Rajendram, and Kerwyn Casey Huang. Effects of fixation on bacterial cellular dimensions and integrity. *iScience*, 24(4):102348, 2021. ISSN 2589-0042. URL <https://www.sciencedirect.com/science/article/pii/S2589004221003163>.

Appendix A. Proof of the valid distance set theorem

We start with a basic observation.

Proposition 5 (Properties of the distance function)

- $\mathcal{A}_1 \subset \mathcal{A}_2 \Rightarrow \forall \mathbf{x} \in \mathcal{X}, \text{dist}(\mathbf{x}, \mathcal{A}_2) \leq \text{dist}(\mathbf{x}, \mathcal{A}_1)$.
- $\mathcal{A}_1 \subset \mathcal{A}_2$ and $\mathbf{x} \in \mathcal{A}_1 \Rightarrow \text{dist}(\mathbf{x}, \partial \mathcal{A}_1) \leq \text{dist}(\mathbf{x}, \partial \mathcal{A}_2)$.

Proof The first item is direct:

$$\begin{aligned} \text{dist}(\mathbf{x}, \mathcal{A}_1) &= \inf_{\mathbf{x}' \in \mathcal{A}_1} \text{dist}(\mathbf{x}', \mathbf{x}) \\ &\geq \inf_{\mathbf{x}' \in \mathcal{A}_2} \text{dist}(\mathbf{x}', \mathbf{x}) = \text{dist}(\mathbf{x}, \mathcal{A}_2). \end{aligned}$$

Here is one proof of the second item by separating the two cases: either $x \in \mathring{\mathcal{A}}_1$ or $x \in \partial \mathcal{A}_1$.

- *Case 1:* $x \in \partial \mathcal{A}_1$. This case is trivial since $\text{dist}(\mathbf{x}, \partial \mathcal{A}_1) = 0 \leq \text{dist}(\mathbf{x}, \partial \mathcal{A}_2)$ by positivity of the distance.
- *Case 2:* $x \in \mathring{\mathcal{A}}_1$. In that case, the key argument is to show that the open ball of radius $\text{dist}(\mathbf{x}, \partial \mathcal{A}_1)$ centered in x is included in $\mathring{\mathcal{A}}_1$. Precisely

$$\tilde{\mathcal{B}} \stackrel{\text{def}}{=} \mathcal{B}(x, \text{dist}(\mathbf{x}, \partial \mathcal{A}_1)) \subseteq \mathring{\mathcal{A}}_1 \subseteq \mathring{\mathcal{A}}_2. \quad (16)$$

Indeed having Equation (16) established implies by contraposition that

$$\partial \mathcal{A}_2 \subseteq \mathring{\mathcal{A}}_2^c \subseteq \tilde{\mathcal{B}}^c \quad (17)$$

where the first inclusion is given by $\partial\mathcal{A}_2 \stackrel{\text{def}}{=} \bar{\mathcal{A}}_2 \setminus \mathring{\mathcal{A}}_2 \subseteq \mathring{\mathcal{A}}_2^c$. Therefore, taking infimum with respect to these sets, implies the following inequalities and by the way the intended result.

$$\begin{aligned} \text{dist}(\mathbf{x}, \partial\mathcal{A}_2) &= \inf_{z \in \partial\mathcal{A}_2} \|z - x\| \\ &\geq \inf_{z \in \mathring{\mathcal{A}}_2^c} \|z - x\| \\ &\geq \inf_{z \in \mathring{\mathcal{B}}^c} \|z - x\| = \text{dist}(\mathbf{x}, \partial\mathcal{A}_1). \end{aligned}$$

So let's prove Equation (16): by contradiction, assume that there exists $z \in \mathring{\mathcal{B}} \cap \mathring{\mathcal{A}}_1^c$. Notice that $[x, z] \cap \mathring{\mathcal{A}}_1^c$ is a compact set (here $[x, z]$ denotes the closed segment between the points x and z). Thus

$$z^* \stackrel{\text{def}}{=} \arg \min_{x' \in [x, z] \cap \mathring{\mathcal{A}}_1^c} \|x - x'\|$$

is well defined and the semi-open segment $[x, z^*[$ is included in $\mathring{\mathcal{A}}_1$. This implies that $z^* \in \partial\mathcal{A}_1$ since the sequence $z_n \stackrel{\text{def}}{=} x + (1 - \frac{1}{n})(z^* - x) \in [x, z^*[\subseteq \mathring{\mathcal{A}}_1$ converges to z^* . The contradiction comes from

$$\text{dist}(\mathbf{x}, \partial\mathcal{A}_1) \leq \|z^* - x\| \leq \|z - x\| < \text{dist}(\mathbf{x}, \partial\mathcal{A}_1).$$

The first inequality holds because $z^* \in \partial\mathcal{A}_1$, the second one because $z^* \in [x, z]$ and last one because $z \in \mathring{\mathcal{B}}$.

By proof by contradiction, Equation (16) holds.

In conclusion, in all cases the inequality is verified. \blacksquare

Theorem 4 can be proven in two steps. First, notice that the inclusion $\mathcal{B} \subseteq \mathcal{E}$ (Assumption 1) and the first bullet in Proposition 5 implies that $\text{dist}(\mathbf{x}, \mathcal{E}) \leq \text{dist}(\mathbf{x}, \mathcal{B})$ for any $\mathbf{x} \in \mathcal{X}$.

Let's establish the converse inequality. Let \mathbf{x} denote an arbitrary point in \mathcal{D} . Aiming for a proof by contradiction, assume that $\text{dist}(\mathbf{x}, \mathcal{E}) < \text{dist}(\mathbf{x}, \mathcal{B})$. We can proceed by separating two cases:

- *Case 1:* $\text{dist}(\mathbf{x}, \mathcal{E}) = 0$. This implies that $\mathbf{x} \in \mathcal{E}$ since the set \mathcal{E} is closed as a finite union of closed sets $\partial\mathcal{X}_{i,j}$. Moreover, as \mathbf{x} belongs to \mathcal{D} , in particular \mathbf{x} belongs to \mathcal{S} . It is sufficient to apply (11) and obtain $\mathbf{x} \in \mathcal{E} \cap \mathcal{S} \subseteq \mathcal{B}$ which is inconsistent with $\text{dist}(\mathbf{x}, \mathcal{B}) > 0$.
- *Case 2:* $r \stackrel{\text{def}}{=} \text{dist}(\mathbf{x}, \mathcal{E}) > 0$. The point \mathbf{x} verifies $\text{dist}(\mathbf{x}, \mathcal{B}) \leq \text{dist}(\mathbf{x}, \mathcal{CB})$ as $\mathbf{x} \in \mathcal{D}$. Let us define $r \stackrel{\text{def}}{=} \text{dist}(\mathbf{x}, \mathcal{E}) > 0$ and

$$\begin{aligned} \varepsilon &\stackrel{\text{def}}{=} \text{dist}(\mathbf{x}, \mathcal{CB}) - \text{dist}(\mathbf{x}, \mathcal{E}) \\ &\geq \text{dist}(\mathbf{x}, \mathcal{B}) - \text{dist}(\mathbf{x}, \mathcal{E}) > 0 \end{aligned}$$

by assumption. Since \mathbf{x} belongs to \mathcal{S} , there exists $i_0 \in \{0, 1\}$ such that $\mathbf{x} \in \mathcal{S}_{i_0}$. Because $\text{dist}(\mathbf{x}, \mathcal{E}) = r$, there exists a point $\mathbf{z} \in \mathcal{E}$ such that $r \leq \|\mathbf{x} - \mathbf{z}\|_2 = r + \varepsilon/2$

- *Case 2.a:* $\mathbf{z} \in \mathcal{S}_{i_0}$. By assumption (11), the contradiction comes quickly since now

$$\mathbf{z} \in \mathcal{S}_{i_0} \cap \mathcal{E} \subseteq \mathcal{S} \cap \mathcal{E} \subseteq \mathcal{B} \subseteq \mathcal{CB} \quad (18)$$

and this implies the contradictive inequality

$$r + \varepsilon = \text{dist}(\mathbf{x}, \mathcal{CB}) \leq \|\mathbf{x} - \mathbf{z}\|_2 \leq r + \varepsilon/2.$$

- *Case 2.b:* $\mathbf{z} \notin \mathcal{S}_{i_0}$. In that case, we may define the point \mathbf{y} on the line $[\mathbf{x}, \mathbf{z}]$ which is the nearest from the point \mathbf{x} and also in $\partial\mathcal{S}_{i_0}$. Since $\mathbf{y} \in \partial\mathcal{S}_{i_0} \subseteq \mathcal{CB}$, it implies a contradiction as intended:

$$\begin{aligned} r + \varepsilon = d(\mathbf{x}, \mathcal{CB}) &\leq \|\mathbf{x}, P(s)\|_2 \\ &= s\|\mathbf{x} - \mathbf{z}\|_2 \leq \|\mathbf{x} - \mathbf{z}\|_2 = r + \varepsilon/2. \end{aligned}$$

The point \mathbf{y} is defined as $P(s)$ where the map $P : t \mapsto t\mathbf{z} + (1 - t)\mathbf{x}$ assigns to each scalar $t \in [0, 1]$ a point $P(t)$ of the line and set

$$s \stackrel{\text{def}}{=} \inf_{P(t) \in \mathcal{S}_{i_0}} t. \quad (19)$$

Since $P(0) = \mathbf{x} \in \mathcal{S}_{i_0}$ and $P(1) = \mathbf{z} \notin \mathcal{S}_{i_0}$, the scalar s is well defined. The remaining task is to show that $P(s) \in \partial\mathcal{S}_{i_0}$. The argument works by construction and with a topological argument. Indeed, by definition of the infimum, there exists a sequence $0 < \eta_n \rightarrow 0$ such that $P(s + \eta_n) \notin \mathcal{S}_{i_0}$, thus $P(s) \notin \mathring{\mathcal{S}}_{i_0}$. Also by definition, for all $0 \leq \eta < s$, $P(\eta) \in \mathcal{S}_{i_0}$, thus $P(s) \in \bar{\mathcal{S}}_{i_0}$.

In both cases, the assumption $\text{dist}(\mathbf{x}, \mathcal{E}) < \text{dist}(\mathbf{x}, \mathcal{B})$ leads to a contradiction. We deduce that $\text{dist}(\mathbf{x}, \mathcal{E}) \geq \text{dist}(\mathbf{x}, \mathcal{B})$.

The second inequality in Theorem 4 is a consequence of the property (10). Indeed, this property implies that we can separate the strokes \mathcal{S}_i into connected components $\mathcal{S}_{i,j}$. These are subsets of the connected components $\mathcal{X}_{i,j'}$ for some j' depending on j . The inequality is then just a consequence of Proposition 5.