
LUMIN: Light-sheet Microscopy Analysis Unified with Distributed and Domain-Randomized Generative Models

UNDERGRADUATE THESIS

*Submitted in partial fulfillment of the requirements of
BITS F421T Thesis*

By

Chaitanya Kapoor
ID No. 2020A3TS1219P

Under the supervision of:

Dr. Satrajit Ghosh
&
Dr. S. M. Zafaruddin



BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE PILANI, PILANI CAMPUS

July 2024

Declaration of Authorship

I, Chaitanya Kapoor, declare that this Undergraduate Thesis titled, “LUMIN: Light-sheet Microscopy Analysis Unified with Distributed and Domain-Randomized Generative Models” and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

Date:

Certificate

This is to certify that the thesis entitled, “*LUMIN: Light-sheet Microscopy Analysis Unified with Distributed and Domain-Randomized Generative Models*” and submitted by Chaitanya Kapoor ID No. 2020A3TS1219P in partial fulfillment of the requirements of BITS F421T Thesis embodies the work done by him under my supervision.

Supervisor

Dr. Satrajit Ghosh
Director of Open Data in Neuroscience,
MIT
Date:

Co-Supervisor

Dr. S. M. Zafaruddin
Associate Professor,
BITS-Pilani Pilani Campus
Date:

“And here, poor fool! With all my lore, I stand no wiser than before.”

—Johann Wolfgang von Goethe

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE PILANI, PILANI CAMPUS

Abstract

Bachelor of Engineering (Hons.)

LUMIN: Light-sheet Microscopy Analysis Unified with Distributed and Domain-Randomized Generative Models

by Chaitanya Kapoor

Cell segmentation is a foundational, but non-trivial task for analyzing microscopy images. It often serves as the first step to quantify biological phenomenon (*e.g.*, cellular behavior). As the neuroscience field works towards mapping whole-brains of larger organisms with microscopy, the datasets increase proportionately in size. With this high influx of data, there is a growing need in the field for software tools that can process such large-scale datasets. In this work, we develop a distributed image segmentation workflow that leverages cloud and high-performance computing infrastructure to reduce inference times. Further, existing deep learning segmentation networks require a large pool of human-annotated labels to perform segmentation on real light-sheet microscopy datasets. To mitigate this, we build on an existing domain-randomized generative model, **AnyStar**, to synthesize training datasets that account for observed image acquisition variances in light-sheet microscopy data, which we term **AnyStar++**. We train a zero-shot deep learning models on our datasets, and use it to perform inferences of cellular structures in unseen light-sheet microscopy data. We also conduct preliminary analyses to compare the segmentation results to existing models (*i.e.* Cellpose, **AnyStar**). Through this work, we lay a foundation for developing zero-shot deep learning models that can robustly and accurately perform segmentation on whole brain light-sheet microscopy datasets.

Acknowledgements

There are countless people I am grateful to have in my life, and thank for without whom I would not be a fraction of what I am today.

Over the past semester, I feel like I have grown significantly both as an aspiring scientist as well as an individual. This has opened my eyes to the exciting and collaborative world of science, that I hope to pursue a career in, in the near future. Words cannot express my gratitude to my thesis advisor, Prof. Satra Ghosh, who is one of the kindest and most patient individuals I have ever met. He is a true inspiration of what I (and everyone should!) aspire to be. The vast breadth of resources that he opened up for me and let me, as a naïve undergraduate student discover is truly phenomenal, and I am extremely thankful to him for this. Thank you Satra, for allowing me to attend COSYNE in Portugal this year, in the middle of my project—it was perhaps the most transformative experience till date!

Thank you Yaël, for literally carrying me through the second half of my thesis—there is no way that I would have made a dent in it without you. Your insights, knowledge and love for anything “*synth*” is truly infectious and inspiring :P

Thank you Kabi, for meeting with me everyday, after ditching the first project and being patient with my frivolous and impatient thoughts. You have pushed me to do more in the last month, than I could have imagined. For this, I am extremely grateful to you.

Thank you to all members of SIG! I am going to miss it all. I have been very fortunate to have met some of the kindest and most hardworking people here. You guys have left me with so much to cherish and learn from. Thank you Sabeen and Isaac for being super fun office partners, and listening to my free food obsessions. Thank you Alina, for showing me cool stuff around campus and informing me about free food in the atrium. Thank you Meral for appreciating my love for freebies, and so kindly offering me an old retreat tshirt! I love it so much, and wear it every so often now. Thank you Aaron for getting us free tickets to the Red Sox game. Above all, thank you Building 46 for the (almost) unlimited supply of free food and free icecream and cookies every Monday—it definitely gave me significant motivation to get out of bed in the morning.

Harsha—I have so much to say about our mostly limited time that we got to spend. You definitely look super intimidating in the beginning but are such a softy when I actually spoke to you. I really value the time you took to give me unhinged career advice and making me believe that I took the right choices. I really hope that we meet again sometime in the near future, either skydiving from an aeroplane or just surfing in San Diego.

Of course, I cannot forget to thank Prof. Zafar in BITS for being so kind and pulling a few strings—you know what you helped me do :-)

This would not be complete without expressing my thankfulness and love I have for Professor Donglai Wei, to whom I attribute all my successes till date. If not for him, I would not have the faintest clue of *what* research is, let alone be capable of producing an ounce of work. In the later days of my continued project in 2022, I got introduced to Professor Ed Boyden at the Synthetic Neurobiology Group, MIT.

Ed—your mentorship, support, and reductionist philosophy of approaching scientific questions is something I try to inculcate in everything I have done since. I perhaps can never quantify what it meant for you to take a chance on me, and show me the cutting edge of science right before my eyes and opening up a whole new world of possibilities that I never knew could have existed.

Talmo—I don’t know the number of times that I have screwed up, but it is definitely way more than I can count! You have been an incessant support throughout and made me question (in a good way!), myself and my decisions. You have been ever so kind, friendly, and giving to me, despite my (seemingly) stochastic research trajectory. Thank you for all the flexibility you have given me while I have moonlighted across labs! I don’t think *most* people would be so cognoscente or accommodating about this fact.

Corban—meeting, and becoming such close friends with you was perhaps the most unlikely and best thing to happen to me in Boston. From discussing two-photon microscopy, to sitting me through an arduous live Zebrafish imaging session, to the airport ride in Summer 2023, it has been a crazy one! I am so grateful for our friendship. Thank you for inviting me to Aletheia every so often, and your Bible study sessions.

Jack—you were my first friend in Boston and I can never forget that. We have had so many memories here in the last 2 years, that I can never forget. From slandering Arsenal, to almost always beating you at a game of FIFA, I don’t think that I could have asked for a better friend. You have been so supportive and instrumental in guiding my baby steps into the beautiful world of Neuroscience and behavior, and for that I am extremely grateful to you.

Margaret—I have an immense amount of gratitude and respect for all the help that you provided all along the way, despite your busy schedule. Be it my NSF application or the multiExR project, thank you for your immense patience all along.

Nivi—you are the most patient, kind-hearted and selfless person I have ever met. None of this would be possible without you, be it discussing ideas over icecream, coming over for cookies, and countless other memories. You know how much of a support you have been through everything. I am extremely thankful to you for all you do.

Words cannot articulate the love and emotion that I feel for my parents and brother, who, at every step of my journey have been ever so supportive of my questionable, and rather risky career choices. Thank you Parth for being my constant FIFA partner—those have been a core

memory for me. Thank you Maiyya, for all the random conversations you are always eager to have at any hour of the day. You have been a constant source of inspiration to me for staying level-headed even in the most adverse of situations. Thank you Papa for brainstorming even the most random ideas over call, and being so enthusiastic yet encouraging on any given day. Thank you for believing in me even when I didn't know what was possible—it has shaped me into the person I am today.

Contents

Declaration of Authorship	i
Certificate	ii
Abstract	iv
Acknowledgements	v
Contents	viii
List of Figures	x
List of Tables	xiii
Abbreviations	xiv
1 Introduction	1
1.1 Light-Sheet Microscopy	2
1.1.1 Artifacts in Light-Sheet Microscopy	3
1.2 Image Segmentation	5
1.3 Domain Randomization	6
1.4 Problem Formulation	7
2 Large-Scale Volumetric Image Segmentation	8
2.1 Image Segmentation Models	8
2.1.1 Cellpose	8
2.1.2 StarDist	9
2.1.3 AnyStar	10
2.2 Data Processing	10
2.2.1 Dataset	10
2.2.2 Workflow	11
2.2.3 Connected-Component Labeling	13
2.2.4 Structuring Elements	13
2.2.5 Boundary Effects	14

2.3	Stitching Results	15
2.3.1	Qualitative Analysis	16
2.3.2	Putative Analysis	16
2.4	Workflow Profiling	19
2.5	Discussion	20
3	Zero-Shot Segmentation of Light-Sheet Microscopy Data Using Domain-Randomized Generative Models	21
3.1	Image-Label Pair Synthesis	21
3.1.1	Label Synthesis	22
3.1.2	Background Synthesis	22
3.1.3	Alpha Blending	22
3.1.4	Augmentation Sequence	23
3.1.5	Synthesized Datasets	24
3.2	Spherical Harmonics	24
3.2.1	Visualization	26
3.2.2	Orthogonality	27
3.2.3	Parity	27
3.2.4	Rotations	28
3.3	Perlin Noise	28
3.4	Steerable Filters	29
3.4.1	Gaussian Filters	30
3.4.2	Spherical Harmonic Filters	31
3.5	Segmentation of Light-Sheet Microscopy Data	32
3.5.1	Modeling Training	33
3.5.2	Model Inferences	33
3.5.3	Putative Analysis	34
3.6	Code and Data Availability	36
3.7	Discussion	36
4	Discussion and Conclusions	37
A	Proofs and Derivations	39
A.1	Spherical Polynomials	39
A.2	Orthogonality of Spherical Harmonics	41
Bibliography		43

List of Figures

1.1	Light Sheet Microscopy. Successive planes of a biological specimen are illuminated by exciting fluorophores.	2
1.2	Oblique Light-Sheet Microscopy. An oblique plane of light is launched at the specimen of interest, and fluorescence is captured by the same objective lens.	3
1.3	Mechanism and Effect of Spherical Aberrations in Light-Sheet Microscopy Images. (Top) Path of light rays that results in spherical aberrations, due to mismatched focal lengths. (Bottom) Example images before and after correction of the aberration, which results in a sharper image. Figure from [19].	4
1.4	Stripe Artifacts in Light-Sheet Microscopy Images. (a) Stripe artifacts in an imaged section. (b) Post-processing removes the stripes from the region of interest. Figure from [98].	4
1.5	Shadow Artifacts in Light-Sheet Microscopy Images. (a) Shadow artifacts make parts of the acquired image appear incomplete and obscured. (b) Increasing the numerical aperture of the microscope minimizes shadows by focusing light into a finer beam. Figure from [56].	5
1.6	U-Net Architecture. A typical characteristic of the U-Net architecture is its contracting path for capturing context and a symmetric expanding path for precise localization by learning latent vector representations at its bottleneck.	6
2.1	Example of Star-Convex Polygons. (a) illustrates a line segment connecting two points in the set S (denoted by green), connected by a line segment. On the other hand, (b) shows an annular ring where the line segment connecting two arbitrary points does not belong to the set S	10
2.2	Light-Sheet Microscopy Data. (a) Cross-section of light-sheet data along the yz axis. (b) Magnified view of an arbitrary region of interest to show the presence of cellular structures stained with anti-NeuN antibody markers.	11
2.3	Data Processing Workflow. The workflow of processing data, by streaming Zarr files from the DANDI Archive as Dask arrays to a local computing system. Since the data is large (≈ 50 GiB) to efficiently load into system memory and run computations, we <i>chunk</i> and distribute it across multiple parallel threads. Each of the individual chunks are then segmented, and stitched (Sec. 2.2.2) together to reconstruct the original image stack. We qualitatively verify our stitching and segmentation algorithms by visualizing it in popular image viewers (<i>e.g.</i> , Neuroglancer).	12
2.4	Connected Component Labeling Example. An image grid where the white pixels correspond to background and foreground pixels are denoted by 1. The pixels grouped in blue and red illustrate the presence and absence of a neighborhood, respectively, corresponding to the 2D structuring element in Equation 2.1.	14

2.5 Stitching of Cellpose Model Inference. Panel (a) shows a randomly chosen, zoomed-in ROI in the segmented volume. The process of stitching adjacent, distributed voxels is shown in panels (a1) - (a3) for a single slice. Each of the red arrows highlight the volumes pre- and post-stitching. (b) Shows a 3D visualization of the stitched volume, where each distributed voxel is mapped to a different color.	16
2.6 Stitching of AnyStar++ Spherical Model Inference. Panels (a1) - (a3) illustrate the process of stitching adjacent, distributed voxels. Each “section” is labeled with a different color. Again, nuclei instances pre- and post-stitching are denoted by red arrows. (b) Shows a 3D visualization of the stitched volume, where each voxel is mapped to a different color.	17
2.7 Distribution of Nuclei Count. Variation of nuclei count with different chunking dimensions. We plot this distribution for 4 different models—Cellpose, AnyStar, AnyStar++ Gaussian, AnyStar++ Spherical.	18
2.8 Intersection over union for different chunk dimensions. Disribution of IoU computed across 4 different models—Cellpose, AnyStar, AnyStar++ Gaussian, AnyStar++ Spherical. We compute these values for a range of chunked dimensions: {32, 64, 128, 256}.	18
2.9 Time Profile Plot. Run-time differences for three different tasks in the segmentation workflow for the Cellpose and AnyStar-X models.	19
3.1 Alpha Blending Example. (a) Background images (blue square), (b) foreground images (red circle), and (c) composite α blended images for three α values (rows).	23
3.2 Image Composition Comparison. We illustrate differences in the technique used by AnyStar to compose foreground and background, with our generative model. Row (a) replaces all non-foreground pixels with a weighted background to compose the resultant image. On the other hand, in (b), we alpha blend the foreground and background to create a final image.	24
3.3 Light-Sheet Microscopy and Synthesized Training Data. (a) Representative slice from a light-sheet microscopy dataset of the human brain. Example synthetic images generated using the (b) AnyStar , (c) AnyStar++ Gaussian, and (d) AnyStar++ Spherical models. The AnyStar++ Gaussian and AnyStar++ Spherical models created the background using a steerable Gaussian kernel and steerable kernel in the spherical harmonic basis, respectively.	25
3.4 Spherical Coordinate System. The parameters of spherical harmonics are defined on a sphere, and hence follow a similar coordinate convention as spherical coordinates. Note that for a harmonic function, the magnitude of $r = 1$, since these functions are orthonormalized.	26
3.5 Spherical Harmonics in a Cartesian Plane. We visualize an example set of spherical harmonic functions for ($\ell = 2$). The colors here denote the analytical values of the corresponding harmonic at each point.	26
3.6 Fade Function and Perlin Noise. The left panel here shows a plot of a standard fade function $x(t) = 6t^5 - 15t^4 + 10t^3$ which is used to smooth interpolation of image grids. The right panel shows an example of Perlin Noise generated with a random seed = 1, in the 3 rd octave, which is used to determine the amount of texture observed.	29

3.7	Label Synthesis using Perlin Noise. (a) Randomly generated spheres having a mean radius $r = 12$. (b) Perlin noise used for corrupting the uniformly generated spheres. (c) Corrupted labels after using Perlin noise which provides a more realistic topological appearance to the cell bodies.	29
3.8	Orthogonal Components. (Left – Right) We show a symmetrically distributed Gaussian filter of dimensions (9×9) , as represented by Eq. 3.1. The next panels illustrate the orthogonal components of this filter, oriented at 0° and 90° respectively.	30
3.9	Steered 3D Gaussian Kernel. (Left – Right) A slice of a symmetrically distributed 3D Gaussian. We steer this kernel by arbitrary rotation angles $(\alpha, \beta, \gamma) = (70.31^\circ, -1.62^\circ, 174.56^\circ)$ in the middle column. Finally, we use this kernel to convolve a $(128 \times 128 \times 128)$ voxel of white noise to create background corruptions which are characteristic to light-sheet microscopy.	31
3.10	Random Normal Steered Gaussian. Instead of using a uniformly distributed Gaussian for the kernel’s parameters, we sample values from a random normal distribution to form a kernel of dimensions $(7 \times 7 \times 7)$. We notice that a steered convolution using this kernel gives directionally filtered <i>salt and pepper</i> noise.	32
3.11	Spherical Basis Kernel. (a) A steered Gaussian kernel projected to a spherical harmonic basis. (b) Result of convolving the resultant kernel with random Gaussian noise.	33
3.12	Distribution of Nuclei Volume. We show the distribution of nuclei volume estimated from its convex hull, for the previously described models. The unit of volume measure here is in px ³	35
3.13	Distribution of Major Axis Length. We show the variation of the lengths of major axes across the different models summarized earlier. The unit of measure of major axis length is in px.	35

List of Tables

2.1	Workflow Profiling Across Models. We compute the mean runtime for each of the steps described in our workflow—data streaming from the DANDI Archive, distributed segmentation of individual image chunks, followed by stitching of the chunked image volume.	19
3.1	Augmentation Sequence. We provide a list of all augmentations, along with their corresponding stochastic probability (p) of occurrence. Along with this, we also list augmentations that are applied to <code>images</code> and <code>image-label</code> pairs.	25
3.2	Summary of Models and Corresponding Synthetic Datasets.	33
3.11	Qualitative Comparison of Zero-Shot Segmentation Models.	34
3.4	Summary of Segmentation of Light-Sheet Microscopy.	34

Abbreviations

SPIM	Selective Plane Illumination
NA	Numerical Aperture
GAN	Generative Adversarial Network
DANDI	Distributed Archives for Neurophysiology Data Integration
LSM	Light-Sheet Microscopy
IoU	Intersection over Union
NMS	Non-Maximum Suppression
GMM	Gaussian Mixture Model
PV	Partial Voluming

To Donglai Wei, for giving me my first shot at science.

Chapter 1

Introduction

Neuroscience is the study of the nervous systems of living organisms to answer *affective* (*e.g.*, influence of emotions) and *effective* (*e.g.*, influence of stimuli) questions of the brain, its structure, and what it does in response to both internal and external stimuli.

Why study the brain? Because *nobody* understands how it works (yet!) [46]. Modern computing and software has scaled at a tremendous pace, which has fueled advancements in the field [88] at an increasing rate. For instance, a suite of tools has enabled an ongoing effort, at the nanoscale, to construct the connectome, or simply put, neuron connectivity diagram of fruitfly brains [24, 25, 35, 50, 54, 73].

On the other spectrum of scales, **macro-scale** studies draw attention towards tasks such as whole-brain imaging [86], large-scale brain networks [9, 78], and cross-region brain interactions. Scientists believe that this level of analysis could be climactic to understand how large-scale brain activity patterns give rise to cognitive functions, behavior and consciousness.

Undoubtedly, we will someday be able to simulate brains of primates *in-silico*, or perhaps grow brains in petri-dishes. It has been speculated that the growth of deep learning will come to a gradual halt [79, 63, 66], only to turn to neuroscientific insights to design biologically-constrained neural networks, that better emulate brain function.

This makes the study of brains in the 21st century particularly exciting. Neuroscience is unique, in a way that it ties some of the most abstruse disciplines—chemistry, biology, deep learning to answer elusive questions about the brain. Being a nascent field, the rate of development of tools to answer such questions is far outstripped by the deluge of data [10, 80] that is generated by labs around the world.

In this project, we turn our focus to build tools that enable large-scale image segmentation, to aid biologists in tasks such as cell phenotyping [96, 68, 48], neuronal circuit mapping [82,

[49] and *in-situ* transcriptomic profiling [52, 2, 14, 62]. We begin by describing the physics governing Light-Sheet Microscopy, followed by a review of state-of-the-art techniques for image segmentation (Sec. 1.2), and domain randomization (Sec. 1.3). In light of the aforementioned challenges, we conclude by formulating our problem statement, and the tools designed as part of this thesis to advance the field.

1.1 Light-Sheet Microscopy

Imagine we have with us a “*magical*” flashlight, which has the power to shine a super thin and (almost) flat sheet of light, like a piece of paper. Now let us also pretend that we have an opaque toy block with lots of small objects inside it, some of which reflect light, that we wish to look at. We now shine our magical flashlight through this toy block—this thin sheet of light only illuminates a single plane of the block at a time. Since this plane has been illuminated, we can now capture a picture of the visible objects (i.e. those reflecting light). We then sequentially step through the entire toy block, till we have captured all individual layers. We can now stack up images from each of the individual layers, one on top of the other, to create a 3D volume of the toy block, without having to cut it open.

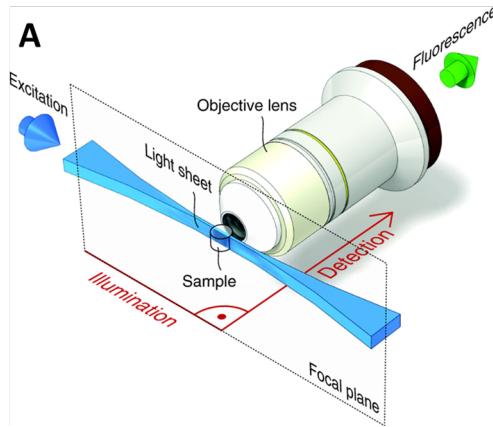


FIGURE 1.1: **Light Sheet Microscopy.** Successive planes of a biological specimen are illuminated by exciting fluorophores.

The above thought experiment, in essence, describes the process of light-sheet microscopy (LSM). The “*light-sheet*” allows a selected plane of a biological sample to be illuminated, by exciting fluorophores in its selected focal volume, and is hence also known as **Selective Plane Illumination Microscopy (SPIM)**, which allows one to visualize, at high-resolutions (~ 250 nm) [1, 26, 12, 37]. However, resolutions often vary based on the wavelength of light emitted by the fluorophores.

Practically, in a light-sheet microscope, a plane of laser light is used to illuminate the selected plane (Fig. 1.1). This allows for optical sectioning, that is, reduction of out-of-focus fluorescence and background noise [1].

A key advantage of LSM is its ease with which it allows *in-vivo* imaging of live animals, which allows one to gain insights into neural activities during the course of biological processes [43, 28, 89].

Several variants of LSM have since been developed, addressing experimental requirements, and hence having their own set of tradeoffs and associated challenges. For instance lattice light-sheet microscopy [12], allows for rapid image acquisition of samples, and hence reduces the effect of phototoxicity. However, as a result of its limited field size, acquisition deteriorates in quality over larger sample depths. Similarly, Swept Confocally-Aligned Excitation (**SCAPE**) microscopy [8, 85], solves the problem of resolution by illuminating the focal plane of interest with an oblique light-sheet (Fig. 1.2). These advancements have expanded the capabilities of LSM, allowing us to image thicker tissue samples and increase speeds of acquisition among others, hence increasing its utility in diverse biological applications.

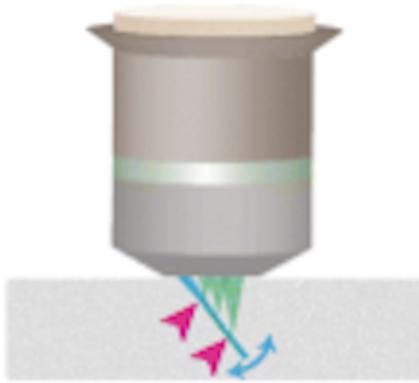


FIGURE 1.2: **Oblique Light-Sheet Microscopy.** An oblique plane of light is launched at the specimen of interest, and fluorescence is captured by the same objective lens.

1.1.1 Artifacts in Light-Sheet Microscopy

Despite having cleared tissue (i.e. make a biological sample transparent), subsequent analysis of neuronal bodies still remains a challenge for several reasons. For one, LSM images are usually anisotropic, engendering a slew of different artifacts along each pair of 3D planes. We describe and illustrate some of these below.

Spherical Aberrations. In LSM, as in any other optical system, aberrations lower the overall quality and accuracy of microscopy data. This is typically a result of the lens having different focal lengths throughout, as a result of its intrinsic curvature. As a result, the formed image appears to be *smeared*, and out-of-focus as shown in Figure 1.3.

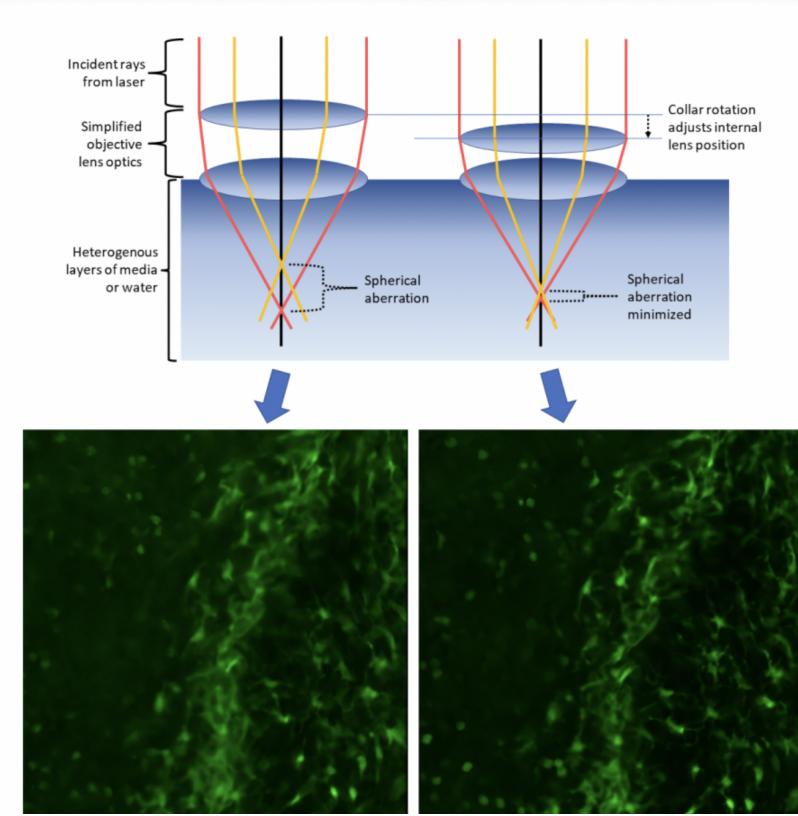


FIGURE 1.3: Mechanism and Effect of Spherical Aberrations in Light-Sheet Microscopy Images. (Top) Path of light rays that results in spherical aberrations, due to mismatched focal lengths. (Bottom) Example images before and after correction of the aberration, which results in a sharper image. Figure from [19].

Stripe Artifacts. A key characteristic of stripe artifacts is the appearance of linear, streaky patterns superimposed on the image (Fig. 1.4). Typically, this is a result of either heterogeneous illumination, or the scattering and absorption of different wavelengths of light by fluorophores in a tissue sample, that is, dense and opaque tissue regions of a specimen would interact differently with light.

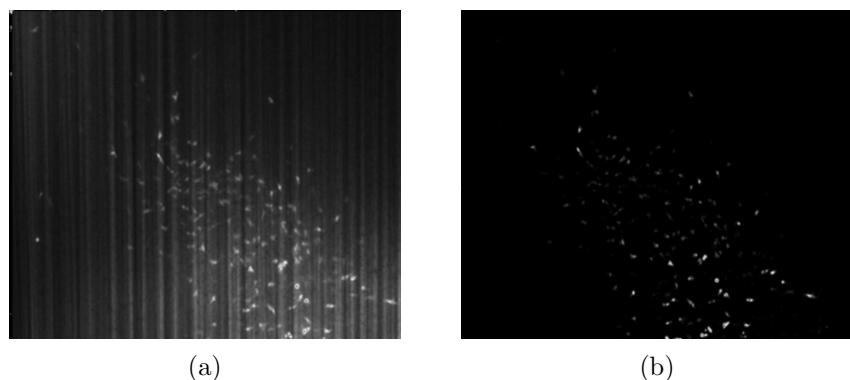


FIGURE 1.4: Stripe Artifacts in Light-Sheet Microscopy Images. (a) Stripe artifacts in an imaged section. (b) Post-processing removes the stripes from the region of interest. Figure from [98].

Shadow Artifacts. Shadow artifacts manifest themselves as either completely dark region or streaks, obscuring parts of the sample. This gives an appearance of having incomplete data (Fig. 1.5). Variations in refractive index within a tissue sample often cause light to bend, and potentially scatter, exacerbating the issue. This is especially common in heterogeneous samples, where different components possess different refractive indices.

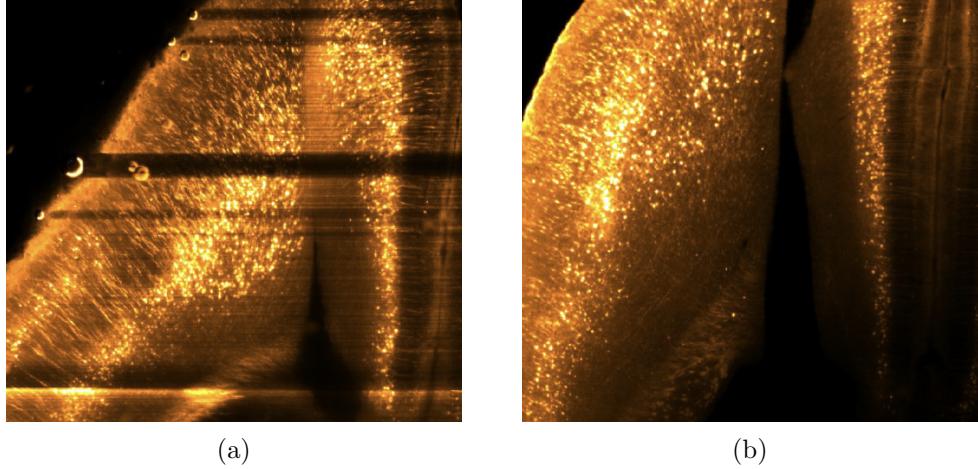


FIGURE 1.5: **Shadow Artifacts in Light-Sheet Microscopy Images.** (a) Shadow artifacts make parts of the acquired image appear incomplete and obscured. (b) Increasing the numerical aperture of the microscope minimizes shadows by focusing light into a finer beam. Figure from [56].

1.2 Image Segmentation

Image segmentation, is one a foundational task in computer vision. The objective is to partition, or segment, an image into multiple hierarchical regions. One prominent task in image segmentation is that of *instance* segmentation, where, the objective is to separate not only different object entities in an image, but also identify each instance of the concerned object. This is unlike *semantic* segmentation, where the task objective is to group image pixels into semantically meaningful categories without distinguishing between different instances of the same object category. This gives instance segmentation the ability of precise localization and recognition. As a result, this spatial information is particularly applicable across a vast range of applications, such as autonomous driving [99, 21], medical imaging [100, 16, 87] and robotics, where accurate identification and delineation of objects are crucial for decision-making and interaction with the environment.

Through the lens of deep learning techniques, instance segmentation is typically accomplished using Convolutional Neural Networks (CNNs) [72, 17]. These networks leverage their intrinsic ability to learn hierarchical representations of visual features extracted using image kernels, as the task entails both fine-grained spatial reasoning and high-level semantic understanding.

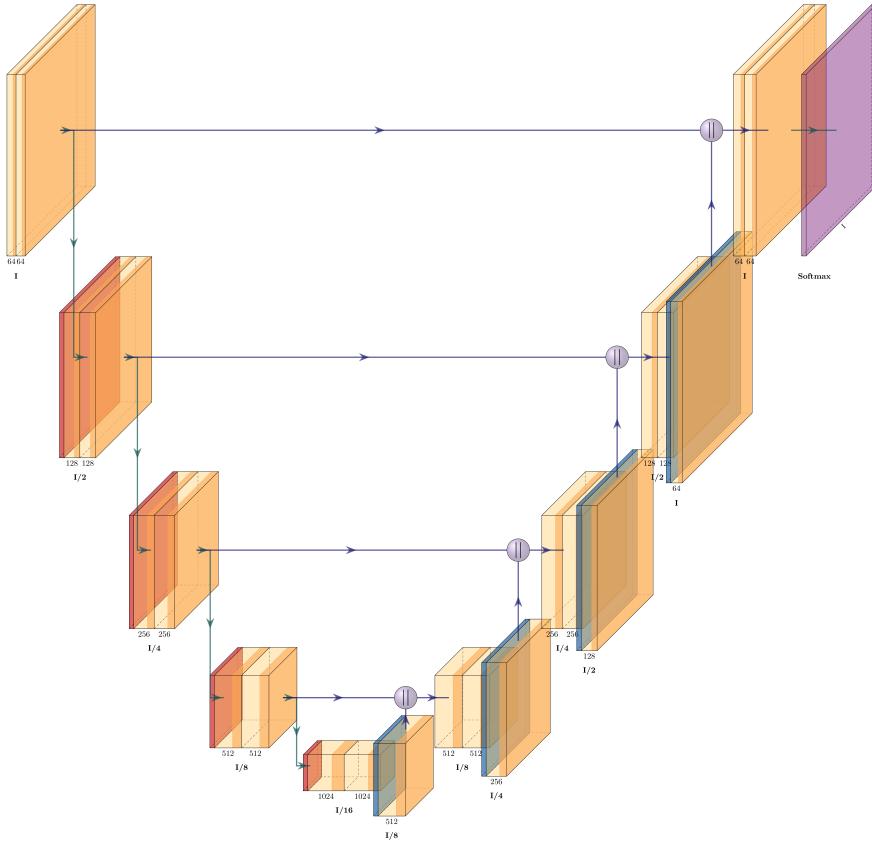


FIGURE 1.6: **U-Net Architecture.** A typical characteristic of the U-Net architecture is its contracting path for capturing context and a symmetric expanding path for precise localization by learning latent vector representations at its bottleneck.

However, despite these multitudes of advancements, few works exist which are able to lend themselves for useful neuroscientific applications, from the perspective of imaging. In particular, most works make use of networks which are *extremely* data hungry, making them less than ideal for applications where there is a lack of ground-truth annotations.

In case of LSM nuclei instance segmentation, which is the focus of this thesis, state-of-the-art methods depend heavily on a vast amount of annotated data. For instance, Yin *et al.* [98] train a 3D U-Net (Fig. 1.6) for vessel segmentation in Zebrafish. This method fails to scale well with lesser ground truth data. On the other hand, Yang *et al.* [97] propose the use of a Generative Adversarial Network (GAN) to address lack of data. However, this does not translate effectively when one wants to use these models for human brain cell analysis. Several factors are responsible for this, which we have elucidated in Section 1.1.1.

1.3 Domain Randomization

Bridging the “*reality-gap*” which distinguishes synthetic from real world data potentially holds a lot of promise for large-scale, automated image analysis, particularly in biology. At its core,

domain randomization is a technique where a large variety of data is created for the models to train on. This, in theory, exposes the model to a large variation of data, and if the variability is significant enough, the models should generalize to the real world with no additional training.

A useful analogy to this could be the following—imagine we have a robot, and we want to teach it to find a red ball, but in different rooms. If we teach the robot to search for this ball in a single room, it may not be able to do well in other rooms, only because the surrounding environment has now changed. However, now imagine that we have lots of different rooms with different colors, shapes etc. This way, our robot learns to find the red ball no matter where it is hidden. Now, when we finally play in a new room, the robot is really good at finding the red ball because it had practiced in so many different rooms.

In a similar manner, we randomize, using a set of image augmentations, image parameters such as intensity, shear, resolution etc. to force our neural network to learn essential features of the objects of interest. This is inspired by a relatively new suite of tools and techniques in the field of biomedical image segmentation [23, 20, 36, 6] among others. In this thesis, we simulate cortical sections of the human brain acquired via LSM, and domain randomize the dataset to train a deep learning model. This enables us to perform inferences on real-world data without the need to retrain or fine-tune our network, hence using the idea of what is known as transfer learning [83].

1.4 Problem Formulation

The complexity and size of LSM datasets require modern infrastructure and methods for analysis. To process such large datasets, we develop a framework that is capable of segmenting cell bodies in terabyte-scale LSM images by distributing the processing and then stitching the results. To improve the accuracy of segmentation of LSM data, we build on an existing generative framework to synthesize training data with domain randomized features. We leverage topological similarities (i.e. star-convex polygons) of nucleic bodies to synthesize shapes. This allows us to train a deep learning network on the synthetic dataset, and hence perform inferences on unannotated, real-world data.

Manual annotation of datasets is time and resource intensive. As a consequence, it is critical to establish a robust set of metrics, or quantifiable measures to assess the quality of image segmentation models. Along these lines, we provide a preliminary set of test metrics which we compute to assess model performance. This has direct implications, in enabling one to gain insights into model scalability, generalization and applicability for real-world imaging tasks.

Chapter 2

Large-Scale Volumetric Image Segmentation

In this chapter, we describe a workflow to perform cell body segmentation on large-scale, volumetric images. We demonstrate results from our pipeline on a 1mm^3 section of the human brain acquired using light-sheet microscopy. Our framework is capable of processing terabyte-scale images, and addresses the challenge of performing *distributed* segmentation, and stitching voxelized chunks to compose a volume at the original resolution.

We describe models that are supported by our framework in section 2.1. In section 2.2, we describe the data processing workflow, which includes methods to perform distributed stitching of individual voxels (section 2.2.2). We conclude with post-hoc analysis on evaluating our stitching algorithm in the absence of ground-truth labels.

2.1 Image Segmentation Models

In this section, we present a description of the cell segmentation models that have been used for this thesis. A common claim across all models discussed here is their capability to perform *generalist* cell segmentation, i.e.: segment cells without the need for model retraining or parameter tuning. These characteristics make them a suitable choice for conducting fair model evaluations.

2.1.1 Cellpose

Cellpose is a deep-learning based cellular segmentation algorithm that is capable of segmenting cellular structures across a multiple imaging modalities [81, 59]. To learn a standardized cell representation, manually annotated masks are transformed to an intermediate representation by

simulating a *reversible* diffusion process. A key advantage of this representation is that extrema of a large variety of shapes lie at the geometric center of their segmentation masks. As a result, the point of convergence of the horizontal and vertical flow gradients is used to recover the original masks.

A neural network (in this case, a U-Net [72]) is used to predict the spatial flows as computed by simulating a diffusion process. In addition to the flow, the neural network also predicts an object localization probability, which enables prediction of cell boundaries. The final segmentation masks are retrieved by thresholding these localization probabilities.

2.1.2 StarDist

Cell nuclei typically possess a roundish shape in microscopy images. Traditional cell segmentation approaches either localize individual instances, assuming an approximate shape representation which is refined iteratively. Such (*top-down*) methods [53, 69, 70] rely on accurate refinement of shapes to yield accurate mask predictions.

On the other hand, a different (*bottom-up*) set of approaches [11, 95, 77, 15] first classify every pixel in an image to an instance. These predictions are then fused with one another using methods such as connected-component labeling (Sec. 2.4) to predict a single nucleic instance. However, in the presence of a high-density of nucleic cells, the fusion of misclassified pixels are aggravated as errors are propagated across a large number of segmented nuclei. To alleviate these issues, StarDist [74, 92, 91] bakes in an implicit geometric prior which constrains a neural network to learn localized shapes in the manifold of star-convex polygons. This class of polygons is capable of expressing shapes observed across most cellular structures. We provide a brief description of this below.

Star-Convex Polygons. For an arbitrary set S in a Euclidean space \mathbb{R}^n , a star-domain is said to exist if $s_0 \in S$, such that for all $s \in S$, a line segment from s_0 to s lies in S .

This definition can be extended to define a star-convex **polygon**, to be a star-domain whose boundary is a sequence of connected line segments. For example, an annular ring (Fig. 2.1b) is not a star domain since a line segment across the diameter of the annulus does not lie in its star-domain S , which defines the polygon.

For every pixel (i, j) , the network regresses a distance $\{r_{i,j}^k\}_{n=1}^k$ to the predicted boundary of the cell along a set of k precomputed radial directions. Along with this, the model also predicts an object probability, which is filtered using non-maximum suppression to arrive at the final segmentation mask.

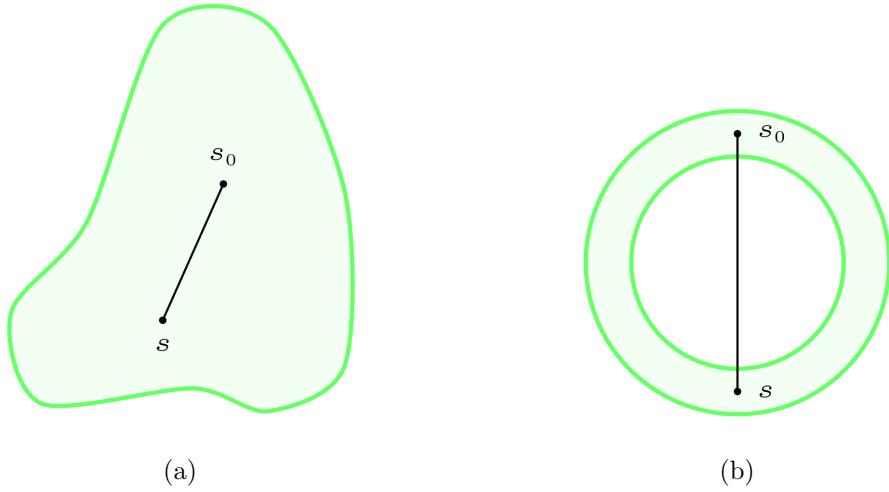


FIGURE 2.1: **Example of Star-Convex Polygons.** (a) illustrates a line segment connecting two points in the set S (denoted by green), connected by a line segment. On the other hand, (b) shows an annular ring where the line segment connecting two arbitrary points does not belong to the set S .

2.1.3 AnyStar

Borrowing from the idea of implicitly enforcing star-convex geometric priors, AnyStar aims to synthesize photorealistic blob-like shapes to train a star-convex deep-learning network [23]. AnyStar additionally performs physically-informed augmentations of the images (Sec. 1.3) to simulate imaging artifacts (*e.g.*, Gibbs ringing [84, 65]). By virtue of sampling multiple probability distributions to synthesize augmentations, such a scheme can potentially generate infinite synthetic data and its corresponding labeled pairs to train a deep learning network. These factors makes the learned network robust to naturally occurring imaging artifacts and potentially out-of-distribution examples. It also helps to tackle the issue of a lack of annotated data, by readily synthesizing it.

2.2 Data Processing

In this section, we describe the dataset used for analysis, and the workflow that streams the dataset, performs distributed computation, and visualization.

2.2.1 Dataset

We make use of a publicly available, light-sheet microscopy (LSM) dataset available on the **DANDI** Archive [60, 40]. This dataset contains 1mm^3 sections of a human brain, imaged using LSM. The sections were stained with several markers including anti-NeuN antibody for

neuronal cells [33], YOYO for nuclei, and Lectin for blood vessels. Each of the sections were SHIELD-processed [61] to enable high-throughput screening and imaging of various proteins. The resolution of the tissue sample is $2.5\mu\text{m} \times 3.6\mu\text{m} \times 2.5\mu\text{m}$, along the (x, y, z) axes, respectively. We show a representative cross-sectional view from this dataset in Figure 2.2, which can be visualized in Neuroglancer.

Note that for our analysis, we make use of a singular channel, having anti-NeuN stains.

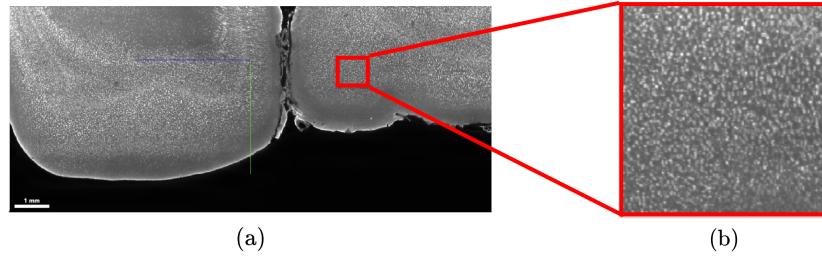


FIGURE 2.2: Light-Sheet Microscopy Data. (a) Cross-section of light-sheet data along the yz axis. (b) Magnified view of an arbitrary region of interest to show the presence of cellular structures stained with anti-NeuN antibody markers.

2.2.2 Workflow

The workflow to segment large-scale image volumes has several key steps that are shown in Figure 2.3 and are described below:

- Stream data from DANDI Archive
- Lazy evaluation of data using Dask arrays
- Distributed cell segmentation of chunks
- Chunk stitching
- Visualization

Data Streaming from DANDI Archive. Due to large data sizes and limited local storage, we stream Zarr [58] arrays that are stored remotely on the DANDI Archive to the local computing environment. A major advantage of this approach is that it avoids local caching of data, hence allowing for flexible downstream data processing.

Lazy Evaluation of Data using Dask Arrays. This refers to the technique of deferring any computations until they are needed (i.e. triggered by a specific user command). We achieve this

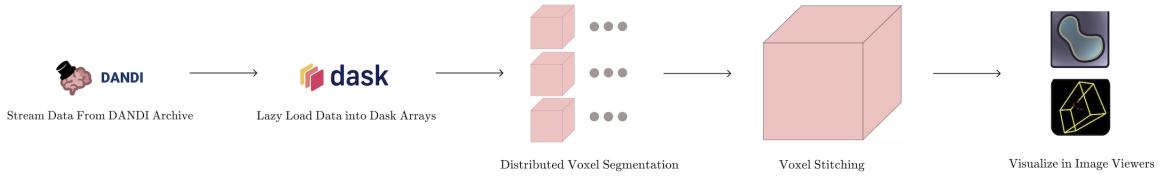


FIGURE 2.3: Data Processing Workflow. The workflow of processing data, by streaming Zarr files from the DANDI Archive as Dask arrays to a local computing system. Since the data is large (≈ 50 GiB) to efficiently load into system memory and run computations, we *chunk* and distribute it across multiple parallel threads. Each of the individual chunks are then segmented, and stitched (Sec. 2.2.2) together to reconstruct the original image stack. We qualitatively verify our stitching and segmentation algorithms by visualizing it in popular image viewers (e.g., Neuroglancer).

functionality by (lazy) loading the streamed Zarr files as Dask arrays and queuing computations, which in this case is performing image segmentation. This process allows Dask to lazy construct an internal *task* graph, which queues all necessary computations that need to be performed. As an example, we demonstrate lazy evaluation with toy functions for multiplying two numbers in Listing 1.

```

1 import dask
2
3 def multiply(num1: int, num2: int):
4     return num1 * num2
5
6 # use a delayed function decorator to enable lazy evaluation
7 @dask.delayed
8 def multiply_dask(num1: int, num2: int):
9     return num1 * num2
10
11 # "eager" computation
12 res = multiply(num1=2, num2=3)
13
14 # a computation is explicitly triggered on a delayed object
15 res_dask = multiply(num1=2, num2=3)
16 res_dask = res_dask.compute()

```

LISTING 1: Lazy Evaluations. The code listing here demonstrates a lazy evaluation. We define two functions to perform identical tasks (multiplication). In case of the `dask.delayed` wrapped function, an explicit `.compute()` object call is needed to evaluate the desired value.

Distributed Segmentation of Chunks. Since the entire light-sheet volume is too large to fit into any computing systems RAM, we divide the volume into groups of voxels (i.e. chunks) which are distributed across computing threads. A specific model then fetches independent voxel chunks, and performs segmentation. This process has two key advantages:

1. Maximum compute usage, since all cores and threads used. This form of chunking can be done since each voxel has locally independent image features.
2. There is **no** local caching of results, as described in the [lazy evaluation](#) section.

After segmenting each individual chunk, a composite volume, that matches the original image resolution, is stitched while preserving relative locations to each other in the original volume. This is done by storing offsets of each voxel in the pixel space. As a result, we can systematically arrange our voxels in the desired order.

Chunk Stitching. A critical part of segmenting large volumes is stitching individual chunks. We describe the process of stitching voxels accompanied with an explanation of boundary effects that need to be addressed to preserve the original resolution of the volume while stitching.

In this section, we describe the main components that have been used to stitch chunks (i.e. groups of voxels), after chunking them for distributed computation. We begin by describing the idea of connected-component labeling (Sec. 2.2.3). We follow this by detailing certain nuances that arise when one tries to stitch faces of voxels, whilst preserving the resolution of the original voxel (Sec. 2.2.5). Last, we introduce the idea of structuring elements, which is used to determine neighborhood connectivity of pixels.

2.2.3 Connected-Component Labeling

The goal of connected-component labeling is to group pixels in an image into local, individualized components that share similar intensity values. The topological landscape of grouped pixels is governed by a structuring matrix (Sec. 2.2.4). We extend this idea to stitch instance segmentation masks *across* chunks, when they maybe split as a result of chunking the original image volume, thus ensuring label consistency.

2.2.4 Structuring Elements

Briefly, a structuring element is used to define a neighborhood of a pixel by defining a local topology. For example, in the case of connected-component labeling, a commonly used structuring element for 2D images is a 4-connected neighborhood. Pixels are considered as *neighbors* if they share an edge between each other. This can be formalized by the following matrix structure:

$$\mathcal{Q} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{pmatrix} \quad (2.1)$$

where \mathcal{Q} is the structure matrix. We illustrate the effect of using such an element on a 2D grid in Figure 2.4. Since the connections (edges) formed while labeling the adjacency graph are undirected, it necessitates the structure matrix to be **centrosymmetric** [90].

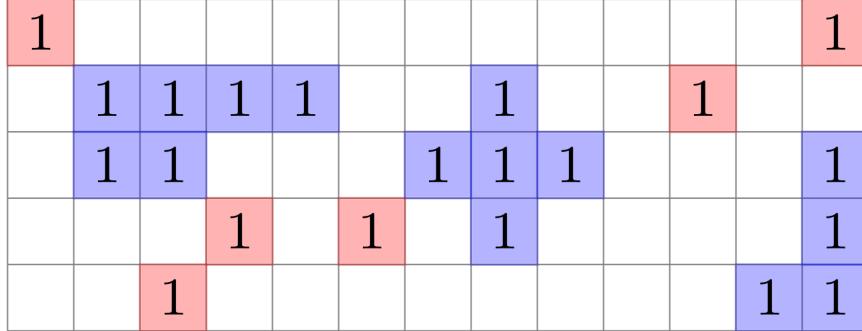


FIGURE 2.4: **Connected Component Labeling Example.** An image grid where the white pixels correspond to background and foreground pixels are denoted by **1**. The pixels grouped in **blue** and **red** illustrate the presence and absence of a neighborhood, respectively, corresponding to the 2D structuring element in Equation 2.1.

We extend this idea to 3D, and use a cube as a structuring element.

$$\tilde{\mathcal{Q}} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

Note that here, each element of the matrix represents a *voxel*. Through this, we attempt to describe and capture a spatial relationship that describes a cube in $3 \times 3 \times 3$ neighborhood.

2.2.5 Boundary Effects

A major caveat of stitching individual voxels is preserving the original image volumes dimensions. Directly stitching chunks together can result in a smaller volume than the original due to local label connectivity. Since we operate on the chunk boundaries during connected-component labeling, we must ensure that each chunk is slightly expanded along its borders to maintain dimensions. One must also be careful to trim the borders by the same amount they were expanded. This process can be broken into three key steps:

1. Pad borders to align chunked shapes
2. Map a function (connected-component labeling) over a series of chunks having overlap
3. Trim excess padding to maintain image resolution

Note that while padding, we make use of the `reflect` boundary condition in Dask. This handles the chunks borders by reflecting the values at the edges outwards, hence creating a mirroring effect.

```

1  import numpy as np
2  import dask.array as da
3
4  # example array
5  array = np.array([[1, 2, 3],
6                  [4, 5, 6],
7                  [7, 8, 9]])
8
9  # chunk array into (2, 2) blocks
10 dask_array = da.from_array(array, chunks=(2, 2))
11
12 # reflection boundary condition
13 padded_array = da.pad(dask_array, pad_width=1, mode="reflect")
14
15 result = padded_array.compute()

```

LISTING 2: Reflection Boundary Condition. We use the `reflect` boundary condition to pad our chunked array. This ensures that volume dimensions are preserved, and prevent our segmentation model from predicting false negatives. The result of this computation is shown in Equation 2.2.

This is illustrated with the help of a small example in Listing 2. This padding would yield a result equivalent to:

$$\text{array} = \begin{pmatrix} 5 & 4 & 5 & 6 & 5 \\ 2 & 1 & 2 & 3 & 2 \\ 5 & 4 & 5 & 6 & 5 \\ 8 & 7 & 8 & 9 & 8 \\ 5 & 4 & 5 & 6 & 5 \end{pmatrix} \quad (2.2)$$

An advantage of following such a padding scheme is that it prevents a segmentation model from predicting phantom background pixels, which would then result in spurious connected components.

Visualization. For interactive visualization of the 3D cell segmentation and stitching algorithm results, we use Neuroglancer [38] to view the entire volume and Napari [18] to view sub-volumes.

2.3 Stitching Results

We explore both qualitative (Sec. 2.3.1) and quantitative verification (Sec. 2.3.2) of our stitching algorithm. While we do not have annotated ground truth labels, we make use of certain heuristics

to quantify the efficacy of stitching.

2.3.1 Qualitative Analysis

We illustrate an example visualization of segmented nuclei using the Cellpose and **AnyStar++ Spherical** models. The stitching algorithm is model agnostic, and can potentially be extended to any out-of-the box framework. Note that for both visualizations shown (Fig. 2.5, Fig. 2.6), we use a randomly selected ROI of an isotropic dimension of 128^3 from the dataset. The volumes are re-chunked to an isotropic dimension of 64^3 voxels to demonstrate our algorithms results.

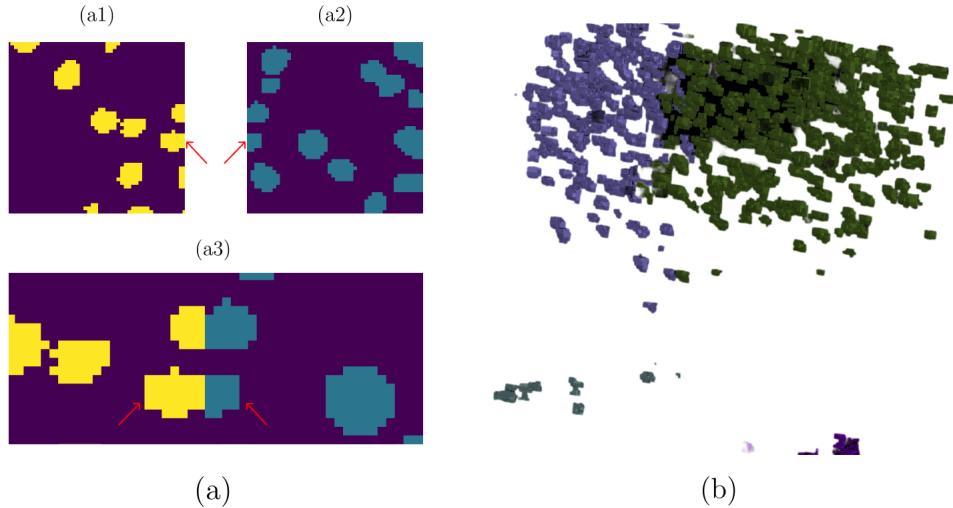


FIGURE 2.5: Stitching of Cellpose Model Inference. Panel (a) shows a randomly chosen, zoomed-in ROI in the segmented volume. The process of stitching adjacent, distributed voxels is shown in panels (a1) - (a3) for a single slice. Each of the red arrows highlight the volumes pre- and post-stitching. (b) Shows a 3D visualization of the stitched volume, where each distributed voxel is mapped to a different color.

Each of the individual, re-chunked voxels are highlighted with a different color to illustrate *partitions*, and show voxel boundaries and spatial offsets.

2.3.2 Putative Analysis

Inspired by the comparison techniques of spike sorting algorithms [93, 29], where there is a lack of standardized, ground-truth bench mark, we propose the use of some heuristics to quantify the performance of our stitching algorithm.

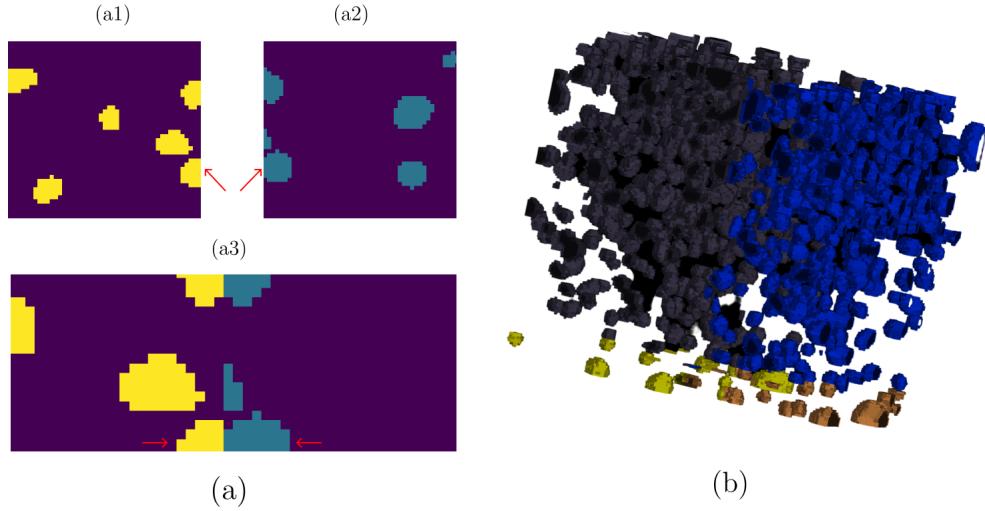


FIGURE 2.6: Stitching of AnyStar++ Spherical Model Inference. Panels (a1) - (a3) illustrate the process of stitching adjacent, distributed voxels. Each “section” is labeled with a different color. Again, nuclei instances pre- and post-stitching are denoted by red arrows. (b) Shows a 3D visualization of the stitched volume, where each voxel is mapped to a different color.

More concretely, let us assume that we use a model \mathcal{M} for segmenting a volume V of data. To segment this volume V , we can potentially use different re-chunked voxel sizes. However, it is critical to note here that all the re-chunked voxel sizes would correspond to the same original volume V . This would imply that once stitched back to its original resolution, both volumes must be identical. That is, suppose the that we chunk V into i voxels.

$$V = \{v_1, v_2, \dots, v_i\}$$

Now we conjecture that once all v_i^s are stitched using an *ideal* algorithm, the original volume should be formed. Or, concretely, we must have

$$V_s = V$$

where V_s is the image volume stitched from the re-chunked voxels. To be able to quantify and identify this similarity, we propose the use of two metrics:

1. Nuclei Count
2. Intersection over Union (IoU)

Nuclei Count. We count the number of nuclei instances for every stitched volume, and plot these results in Figure 2.7. Here, we use a randomly selected 512^3 volume, and label it as our

“ground-truth”, which arguably, serves as a proxy for one. We run inferences with this volume size, without stitching, to assess the base performance of our algorithm. We perform analysis with varied chunk sizes of $\{32, 64, 128, 256\}$.

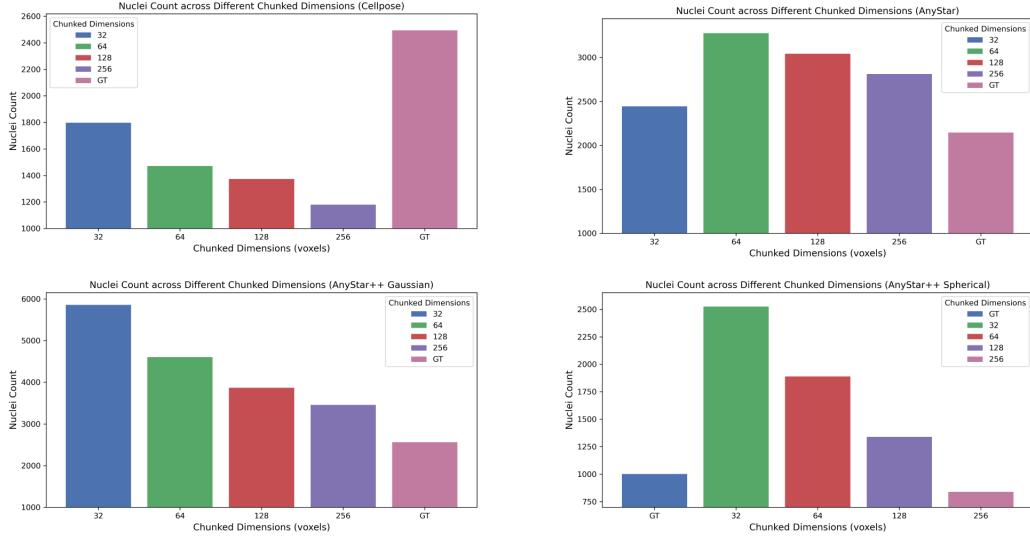


FIGURE 2.7: **Distribution of Nuclei Count.** Variation of nuclei count with different chunking dimensions. We plot this distribution for 4 different models—Cellpose, AnyStar, AnyStar++ Gaussian, AnyStar++ Spherical.

Intersection over Union. We compute the IoU against a randomly selected 512^3 instance mask volume. It should be noted that each of the instance masks are **binarized** before computing IoU. While this eliminates the problem of matching instances across volumes, it fails to capture the splitting or merging of nuclei across voxel boundaries.

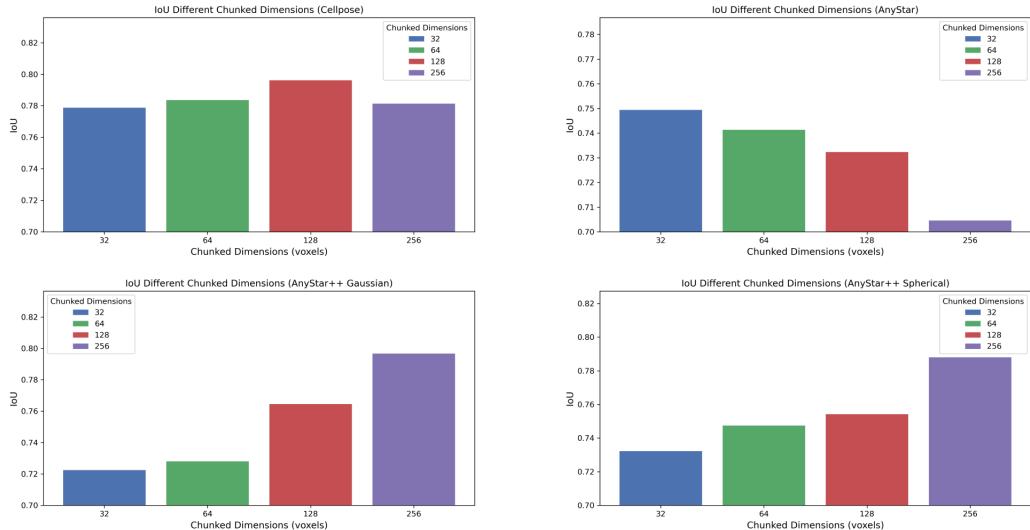


FIGURE 2.8: **Intersection over union for different chunk dimensions.** Disribution of IoU computed across 4 different models—Cellpose, AnyStar, AnyStar++ Gaussian, AnyStar++ Spherical. We compute these values for a range of chunked dimensions: $\{32, 64, 128, 256\}$.

2.4 Workflow Profiling

We profile each of the steps described in the previously mentioned workflow, and describe it in Table 2.1. Note that we conduct this analysis for two separate models—Cellpose and **Anystar-X**, where the latter is used to describe any variant of the StarDist3D-based networks that, which we train on our synthetic data (Sec. 3.1.1).

Segmentation Algorithm	Streaming (sec.)	Segmentation (sec.)	Stitching (sec.)
Cellpose	0.92	30.53	0.132
Anystar-X	0.89	37.66	0.183

TABLE 2.1: **Workflow Profiling Across Models.** We compute the mean runtime for each of the steps described in our workflow—data streaming from the DANDI Archive, distributed segmentation of individual image chunks, followed by stitching of the chunked image volume.

For our analysis here, we use a 128^3 volume sectioned from the LSM Dandiset. This volume is chunked isotropically along all 3 axes, having dimensions $\{64, 64, 64\}$. It also follows logically that mean runtime for both segmentation and stitching will change corresponding to the chunk sizes. We show a visual comparison of time profiles for the above models in Figure 2.9.

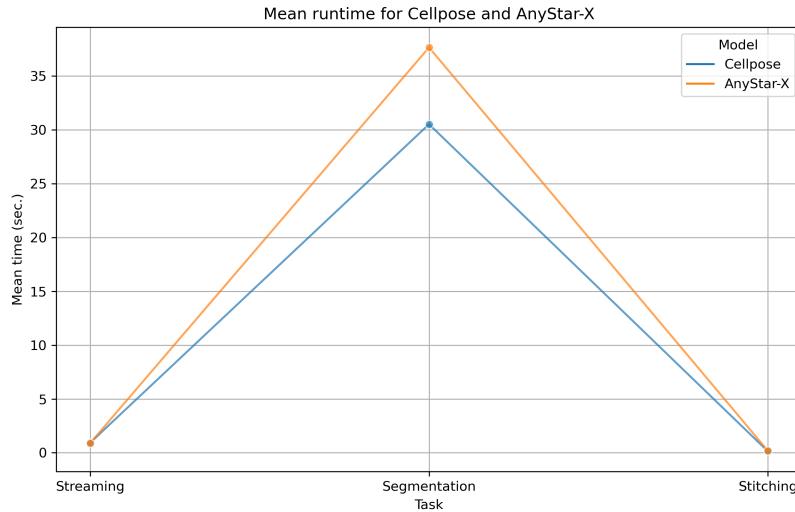


FIGURE 2.9: **Time Profile Plot.** Run-time differences for three different tasks in the segmentation workflow for the Cellpose and AnyStar-X models.

This workflow was run on a system having an Intel(R) Xeon(R) CPU E5-2698 v4 @ 2.20GHz, 2 processing units and a Tesla V100-SXM2-32GB-LS GPU.

2.5 Discussion

In this chapter, we design an image segmentation framework which is capable of segmenting terabyte-scale image volumes. We have made the code for this framework available publicly on [GitHub](#). We describe in detail our workflow for performing distributed segmentation, and stitching a chunked segmentation mask through connected-component labeling. This leverages the power of high-performance computing systems which allows one to significantly reduce inference times. We also perform a qualitative and quantitative test, based on heuristics, of the proposed stitching algorithm in the absence of ground-truth labels for comparison. We conclude by profiling runtimes for each of the individual steps in our proposed framework. These observations allow us to conclude that our stitching algorithm is model agnostic, hence, potentially increasing its applicability.

Chapter 3

Zero-Shot Segmentation of Light-Sheet Microscopy Data Using Domain-Randomized Generative Models

In this chapter, we describe our algorithm to synthesize image-label pairs that are used to train a 3D StarDist deep network for performing zero-shot inferences on real world light-sheet microscopy data. Our image synthesis method builds upon `AnyStar` [23] to generate more realistic backgrounds by leveraging steerable filters to *directionally* spread and sharpen noise. For each image we generate background and foreground separately and alpha-blend them to produce a final composite image, which is subject to the same intensity and artifact augmentation routine as `AnyStar`. We then train the StarDist3D model, and qualitatively compare the inference results to `AnyStar` and Cellpose.

3.1 Image-Label Pair Synthesis

We synthesize image-label pairs in a similar manner as `AnyStar`, whereby the foreground and background are created separately and then alpha blended to form a composite image (Sec. 3.1.3). For the foreground, we begin by synthesizing spheres having uniformly sampled radii corrupted by Perlin noise (Sec. 3.3). For these foreground labels, corresponding backgrounds are synthesized by sampling intensities from a conditional Gaussian mixture model (GMM) which is modulated by directionally steered noise (Sec. 3.1.2). We project Gaussian filters to a spherical harmonic basis, which brings to bear its properties of orthogonality, rotational symmetry and

parity (Sec. 3.2.2, Sec. 3.2.4, Sec. 3.2.3). This is particularly useful in modeling observed anisotropy of signal noise during image acquisition. Finally, an augmentation sequence is applied to the synthetic images (Sec. 3.1.4). We also demonstrate example fluorescence microscopy slices using **AnyStar**'s synthesis structure, and show a comparison with those synthesized by our proposed algorithm, which we call **AnyStar++**.

3.1.1 Label Synthesis

We generate n spheres of radii r , which are sampled from a uniform distribution $\mathcal{U} \sim [4, 20]$. The centers of these spheres are initially placed at the vertices of a regularly-spaced 3D grid, which are then randomly translated. Since cellular bodies are typically non-spherical, the distances d_j^i between each voxel j and sphere i is corrupted by additive Perlin noise (Sec. 3.3). The resultant image grid is now (randomly) zero or reflection padded for simulating varied nuclei distribution. An initial instance maps is then assigned to each of the corrupted spheres.

3.1.2 Background Synthesis

For a label map L having n instances, an initial image is synthesized by sampling foreground intensities from a conditional, n -component GMM, where each of its parameters $\{\mu_i, \sigma_i\}$ are sampled from a uniform distribution for every image. We simulate variable backgrounds to emulate characteristic staining patterns observed in light-sheet microscopy. To synthesize bright foreground instances, we repeat the process described previously, of sampling instances from a n -component GMM.

However, to simulate accurate geometric and topological features in the background, we rely on a generative model as described in **SynthMorph** [36] and **Neurite** [20]. Next, we synthesize a directionally sharpened background using a steered filter having a spherical harmonic basis as described in Section 3.4.2. The composite image is formed by alpha blending the foreground and background images, with the value of α chosen empirically (Sec. 3.1.3).

3.1.3 Alpha Blending

Alpha blending is used to composite a foreground and background image to give an appearance of partial transparency. Simply put, this technique can be formalized as the following linear combination

$$I = \alpha F + (1 - \alpha) B$$

where I is the composed image, constructed from the foreground (F), background (B) and a weighting term (α). Intuitively, $\alpha = 0$ would correspond to an opaque image, whereas $\alpha = 1$ would be completely transparent. This is demonstrated in Fig. 3.1.

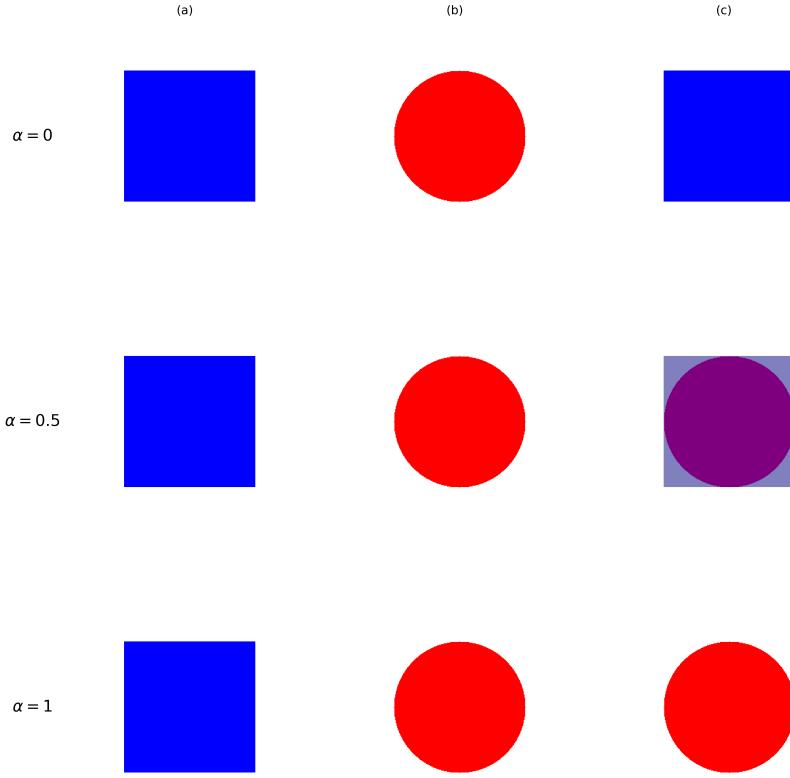


FIGURE 3.1: **Alpha Blending Example.** (a) Background images (blue square), (b) foreground images (red circle), and (c) composite α blended images for three α values (rows).

Often, separate alpha channels are introduced in images, where an alpha value is encoded for every pixel in the image. However, in our case, we apply α uniformly to an entire image matrix.

In contrast to AnyStar, instead of replacing non-foreground pixels, we alpha blend the background with the foreground to better capture artifacts of fluorescence microscopy. Figure 3.2 clearly shows the difference between using the two approaches to form the corresponding composite image.

3.1.4 Augmentation Sequence

The final images in our dataset are obtained after passing them through an augmentation sequence, as detailed in AnyStar [23]. Briefly, each of the volumes obtained from the previous step are cropped to a 64^3 subvolume. They are then subject to a series of affine deformations,

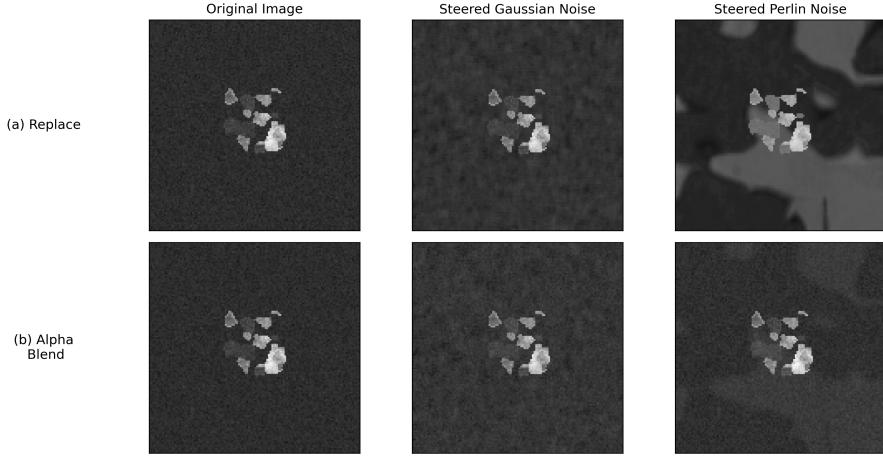


FIGURE 3.2: Image Composition Comparison. We illustrate differences in the technique used by AnyStar to compose foreground and background, with our generative model. Row (a) replaces all non-foreground pixels with a weighted background to compose the resultant image. On the other hand, in (b), we alpha blend the foreground and background to create a final image.

applied uniformly across the entire image matrix. As described previously, the type of image padding used (mirror or reflection) is chosen stochastically to simulate different cell densities.

This is followed by augmenting images with a large set of intensity transforms, which are inspired by image acquisition artifacts in MRIs—bias fields [39, 32], k -space spikes [41], Gibbs ringing [84], Rician noise [31], gamma adjustments and cutout augmentation [22]. Each of the axes of the synthesized voxels are then subject to a Gaussian blur for simulation of partial voluming (PV) [5, 4], i.e. voxels containing multiple tissue classes which have intensities that are not accurate for the instance they represent.

Lastly, lightweight, on-the-fly augmentations are applied to the dataset, such as axis-aligned flips, 90° rotations and elastic deformations. The probabilities and type of augmentation used are summarized in Table 3.1.

3.1.5 Synthesized Datasets

We generate a dataset that closely resembles LSM data using synthesis and domain randomization techniques. Here, we show example images (Fig. 3.3) using each of the methods, and show a qualitative comparison with real LSM sections described in Section 2.2.1.

3.2 Spherical Harmonics

The spherical harmonics $Y_\ell^m(\theta, \phi)$ are a set of functions which are defined on the surface of a sphere. Here, θ is defined as the polar coordinate where $\theta \in [0, \pi]$ and ϕ is the azimuthal

Augmentation	Probability (p)	Type
Spatial Crop	0.50	Image-Label
Affine Deformation	1.0	Image-Label
Bias Field	1.0	Image
Gaussian Noise	0.25	Image
k -space Spike	0.20	Image
Gamma Adjustment	0.80	Image
Gaussian Blur	0.80	Image
Rician Noise	0.20	Image
Gibbs Ringing	0.50	Image
Gaussian Sharpening	0.25	Image
Histogram Shift	0.10	Image
Axis-aligned Flip	1.0	Image-Label
90° Rotation	1.0	Image-Label
Elastic Deformation	1.0	Image-Label
Cutout Augmentation	0.20	Image-Label

TABLE 3.1: **Augmentation Sequence.** We provide a list of all augmentations, along with their corresponding stochastic probability (p) of occurrence. Along with this, we also list augmentations that are applied to [images](#) and [image-label](#) pairs.

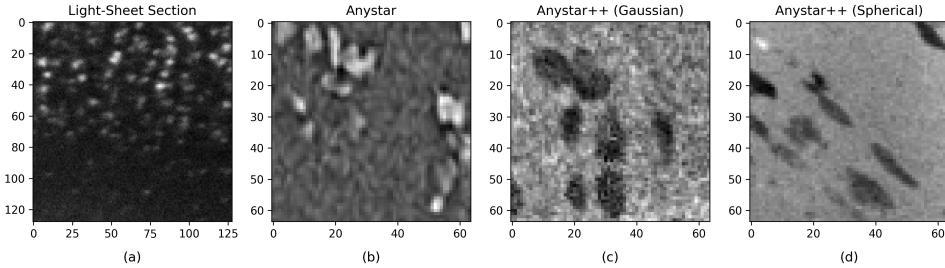


FIGURE 3.3: **Light-Sheet Microscopy and Synthesized Training Data.** (a) Representative slice from a light-sheet microscopy dataset of the human brain. Example synthetic images generated using the (b) AnyStar , (c) AnyStar++ Gaussian, and (d) AnyStar++ Spherical models. The AnyStar++ Gaussian and AnyStar++ Spherical models created the background using a steerable Gaussian kernel and steerable kernel in the spherical harmonic basis, respectively.

coordinate which has its domain defined as the half-open interval $\phi \in [0, 2\pi)$. These definitions are identical to that of a canonical spherical coordinate system, as illustrated in Figure 3.4.

These functions satisfy the spherical harmonic differential equation, which is obtained from the angular component of Laplace's equation [55, 76, 51]. A detailed proof of a closed form representation of $Y_\ell^m(\theta, \phi)$ is described in Appendix A.1.

In the following subsections, we visualize a set of spherical harmonic functions (3.2.1). We then move to describing some important properties of orthogonality (3.2.2) and rotational symmetry (3.2.4) which we leverage for realistic synthesis of light-sheet microscopy data.

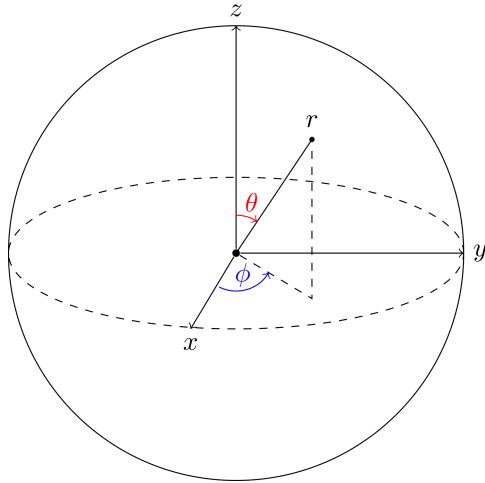


FIGURE 3.4: Spherical Coordinate System. The parameters of spherical harmonics are defined on a sphere, and hence follow a similar coordinate convention as spherical coordinates. Note that for a harmonic function, the magnitude of $r = 1$, since these functions are orthonormalized.

3.2.1 Visualization

In this section, we graphically visualize the spherical harmonics (Fig. 3.5), where each of the phases are color coded. We transform the spherical coordinates to a Cartesian representation for visualization given by

$$x = r \sin \theta \cos \phi$$

$$y = r \sin \theta \sin \phi$$

$$z = r \cos \theta$$

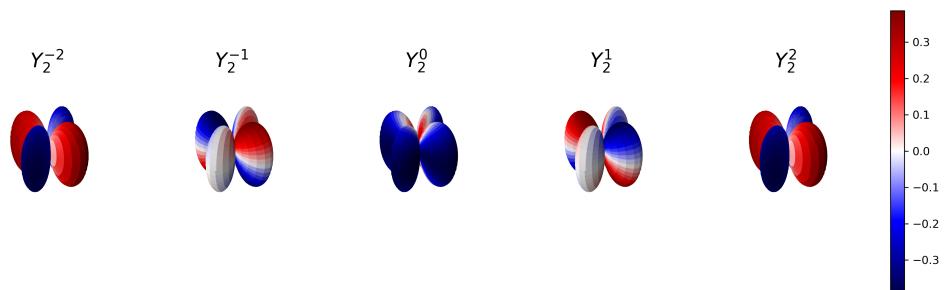


FIGURE 3.5: Spherical Harmonics in a Cartesian Plane. We visualize an example set of spherical harmonic functions for ($\ell = 2$). The colors here denote the analytical values of the corresponding harmonic at each point.

Note that we consider the **real** part of the associated Legendre polynomials¹ for visualization.

¹Refer to Appendix A.1.

An intuitive understanding of the surface plots can be thought of as plotting a surface which satisfies the condition

$$Y_\ell^m(\mathbf{r}) \text{ such that } \|\mathbf{r}\|_2 = 1$$

where $\mathbf{r} = (\theta, \phi)$. These parameters modulate how far we are situated from the origin. As a result, when \mathbf{r} is small, the surface is effectively *shrunk* closer to the origin, whereas it is expanded away from the origin at larger values of \mathbf{r} .

3.2.2 Orthogonality

Orthogonality of spherical harmonics is particularly useful since for a selected degree ℓ , no polynomials are equivalent. That is, $(Y_\ell^m \neq Y_\ell^n)$ where $(m \neq n)$. As a result, it is possible to construct a unique basis of harmonics with desired directional properties by choosing the appropriate multipliers. A detailed proof of this property is illustrated in Appendix A.2.

3.2.3 Parity

When inverted about the origin, certain algebraic properties of spherical harmonics of spatial inversion are either even or odd. More generally, we can intuitively see that under inversion about the origin, the radius r of the sphere remains invariant. However, the angles are transformed to

$$\begin{aligned} \theta &\rightarrow \pi - \theta \\ \phi &\rightarrow \phi + \pi \end{aligned}$$

This parity transformation gives us

$$\begin{aligned} Y_\ell^m(\theta, \phi) &\rightarrow Y_\ell^m(\pi - \theta, \phi + \pi) \\ &= e^{i\ell\phi} e^{im\pi} \sin^m \theta \\ &= e^{i\ell\pi} Y_\ell^m(\theta, \phi) \\ &= (-1)^\ell Y_\ell^m(\theta, \phi) \end{aligned}$$

Here, $(-1)^\ell$ is the *parity* of the harmonic. This property comes of importance particularly when we wish to represent steerable filters having antipodal symmetries. In particular, this allows us to independently manipulate even-parity and odd-parity samples to achieve our desired directional responses to filtered outputs.

3.2.4 Rotations

Spherical harmonics form the set of eigenfunctions for the $\text{SO}(3)$ rotation group. Formally, consider a rotation $\mathcal{R} \in \text{SO}(3)$ applied to a spherical harmonic of degree and order (ℓ, m) respectively. This operation yields a set of spherical harmonics that can be represented as a linear combination of spherical harmonics having the same degree.

$$Y_\ell^m(\theta', \phi') = \sum_{m'=-\ell}^{\ell} \zeta_{mm'} Y_\ell^{m'}(\theta, \phi)$$

where $\zeta_{mm'} \in \mathbb{R}^{(2\ell+1)}$ is a matrix containing constant multiples. This property [67, 42, 7] allows spherical harmonics to encode rotational symmetries, hence lending a natural representation for designing orientation-adaptive steerable filters.

3.3 Perlin Noise

Contrary to Gaussian noise, Perlin noise [64] is a type of gradient noise which is used for the generation of a pseudo-random appearance. For this reason, Perlin noise has found an increasing relevance in simulations of natural environments (*e.g.*, terrain generation) in computer graphics and video games [34, 47, 44, 94].

The key idea of Perlin noise is to generate a set of gradient vectors over a grid, which determine a direction of deformation.

Without loss of generality, we consider a 2D point (x, y) , which is then surrounded by 4 lattice points:

$$\mathbf{p} = \{(i, j), (i + 1, j), (i, j + 1), (i + 1, j + 1)\}$$

where $i = \lfloor x \rfloor$ and $j = \lfloor y \rfloor$. Each of these lattice points has a corresponding gradient vector \mathbf{g}_{ij} associated with them. For each of the corners, a dot product is computed between the gradients and their corresponding distance vector.

$$\sigma = \mathbf{g} \cdot \mathbf{x}$$

The corners of each of the grids is now interpolated using the dot products (σ). Instead of a linear interpolation, a smoother gradient interpolation is achieved by what is known as a *fade function*. This result is illustrated in Fig. 3.6.

For the label synthesis, we use Perlin noise to corrupt perfect spheres, which is illustrated in Figure 3.7, for a randomly selected plane from a 128^3 volume.

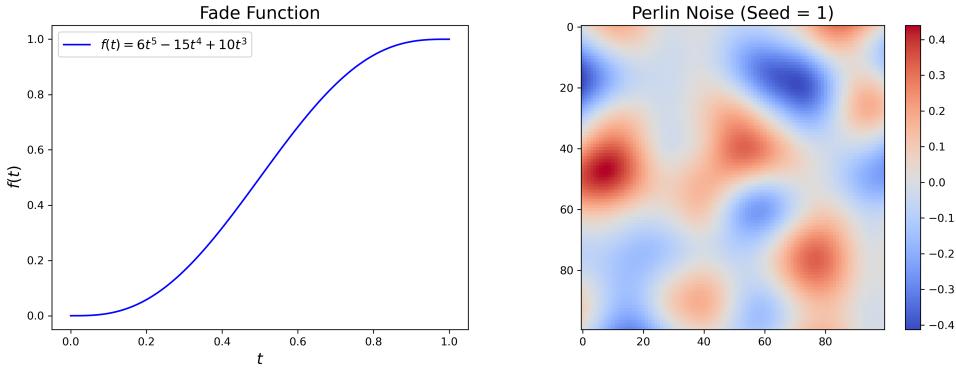


FIGURE 3.6: Fade Function and Perlin Noise. The left panel here shows a plot of a standard fade function $x(t) = 6t^5 - 15t^4 + 10t^3$ which is used to smooth interpolation of image grids. The right panel shows an example of Perlin Noise generated with a random seed = 1, in the 3rd octave, which is used to determine the amount of texture observed.

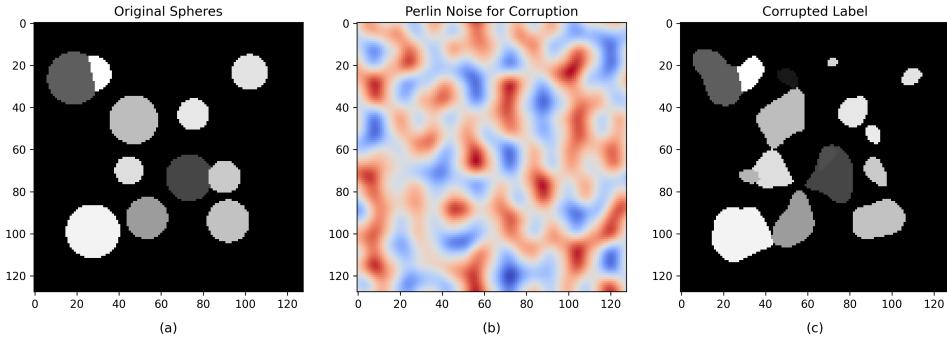


FIGURE 3.7: Label Synthesis using Perlin Noise. (a) Randomly generated spheres having a mean radius $r = 12$. (b) Perlin noise used for corrupting the uniformly generated spheres. (c) Corrupted labels after using Perlin noise which provides a more realistic topological appearance to the cell bodies.

3.4 Steerable Filters

Several low-level computer vision tasks require greater adaptive control over phase and orientation in image filters. To achieve this, one computes filter responses at various orientations. A “basis” of such *steered* filters is obtained by computing a linear combination of these. This allows for computation of impulse responses from these filters at any arbitrary orientation in a n -dimensional space.

We leverage these critical properties of steerable filters [27], by using them as directional smoothing kernels when convolved with noise. This step helps us in synthesizing realistic backgrounds for images acquired by light-sheet microscopy.

In the following sections, we show the effects of steering such filters. We experiment with a Gaussian basis, as well as a spherical harmonic basis to take advantage of its properties of

parities (Sec 3.2.3) and symmetricity.

3.4.1 Gaussian Filters

A 2-dimensional, symmetric Gaussian function G can be described in Cartesian coordinates by the following expression:

$$G(x, y) = e^{-(x^2+y^2)} \quad (3.1)$$

To construct an orthogonal basis from G , we compute a directional derivative along two orthogonal axes positioned at 0° and 90° with respect to one another in a traditional right-handed coordinate system. We denote these (first) derivatives as $G_1^{0^\circ}$ and $G_1^{90^\circ}$

$$\begin{aligned} G_1^{0^\circ} &= \frac{\partial}{\partial x} G(x, y) = -2xG(x, y) \\ G_1^{90^\circ} &= \frac{\partial}{\partial y} G(x, y) = -2yG(x, y) \end{aligned}$$

Thus, for an arbitrary orientation of G , say θ , we can form a basis from the above orthogonal components that spans the entire set of 2D filters, which we denote as G_1^θ .

$$G_1^\theta = \cos(\theta)G_1^{0^\circ} + \sin(\theta)G_1^{90^\circ}$$

Note that care must be taken to normalize the basis function (in this case, G), to have a unity integral over all space. In our case, we assume a scaling factor of 1, without loss of generality. Fig. 3.8 shows a plot of the orthogonal impulse responses of G .

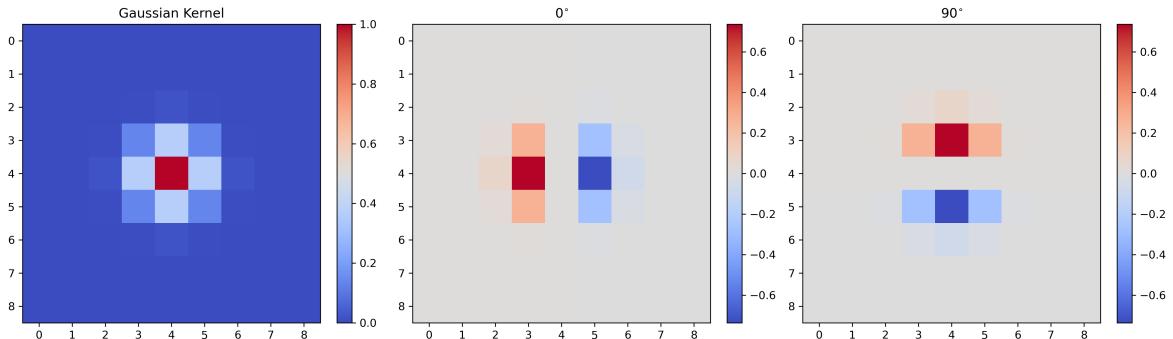


FIGURE 3.8: Orthogonal Components. (Left – Right) We show a symmetrically distributed Gaussian filter of dimensions (9×9) , as represented by Eq. 3.1. The next panels illustrate the orthogonal components of this filter, oriented at 0° and 90° respectively.

We extend this idea to a 3D Gaussian, since our goal is to synthesize image voxels. Consider the following basis function with unit scaling factor:

$$G(x, y, z) = \exp\left(-\frac{x^2 + y^2 + z^2}{2\sigma^2}\right)$$

To construct an orthogonal basis, we consider three arbitrary Euler angular positions - $\{\alpha, \beta, \gamma\}$. Note that this is useful in defining a rotation matrix $R \in \text{SO}(3)$, to directionally orient any filter by transforming the described filters coordinates [75]. Hence, a 3D filter that spans the set of all filters G_1^θ can be represented in spherical coordinates as

$$G_1^\theta = R_x^\alpha G_x + R_y^\beta G_y + R_z^\gamma G_z$$

Where each of the G_x, G_y, G_z denote the first orthogonal directional derivatives. Here, the rotation matrices (R) are defined by standard angular rotations.

$$R_x^\alpha = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha \\ 0 & \sin \alpha & \cos \alpha \end{pmatrix} \quad R_y^\beta = \begin{pmatrix} \cos \beta & 0 & \sin \beta \\ 0 & 1 & 0 \\ -\sin \beta & 0 & \cos \beta \end{pmatrix} \quad R_z^\gamma = \begin{pmatrix} \cos \gamma & -\sin \gamma & 0 \\ -\sin \gamma & \cos \gamma & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

We then convolve the steered filter G_1^θ with Gaussian white noise to produce a background that highlights visual features seen in brightsheet microscopy (Fig. 3.9). Note that the steering angle used here is also randomly chosen, within the appropriate angular domains for each of the individual axes.

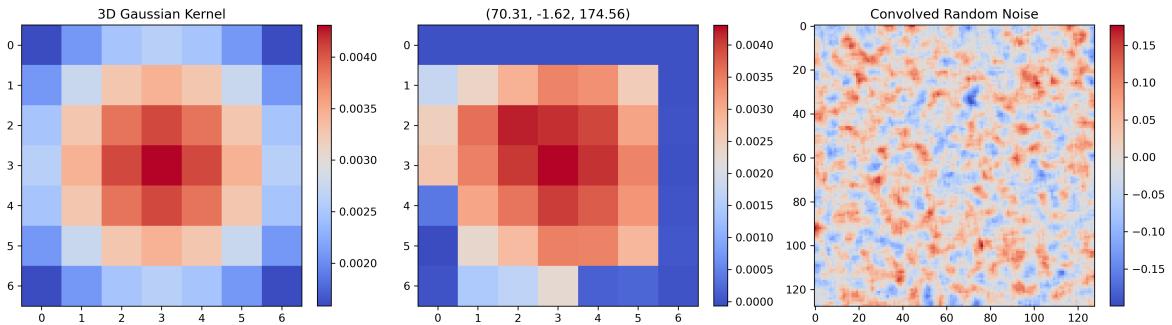


FIGURE 3.9: Steered 3D Gaussian Kernel. (Left – Right) A slice of a symmetrically distributed 3D Gaussian. We steer this kernel by arbitrary rotation angles $(\alpha, \beta, \gamma) = (70.31^\circ, -1.62^\circ, 174.56^\circ)$ in the middle column. Finally, we use this kernel to convolve a $(128 \times 128 \times 128)$ voxel of white noise to create background corruptions which are characteristic to light-sheet microscopy.

To synthesize a larger distribution of images, we add an additional degree of randomness by sampling a 3D Gaussian kernels parameters from a random normal distribution (Fig. 3.10).

3.4.2 Spherical Harmonic Filters

To make effective use of the properties of spherical harmonics as described in Sec. 3.2, we project a Gaussian kernel to a spherical harmonic basis. We describe the process of generating such a

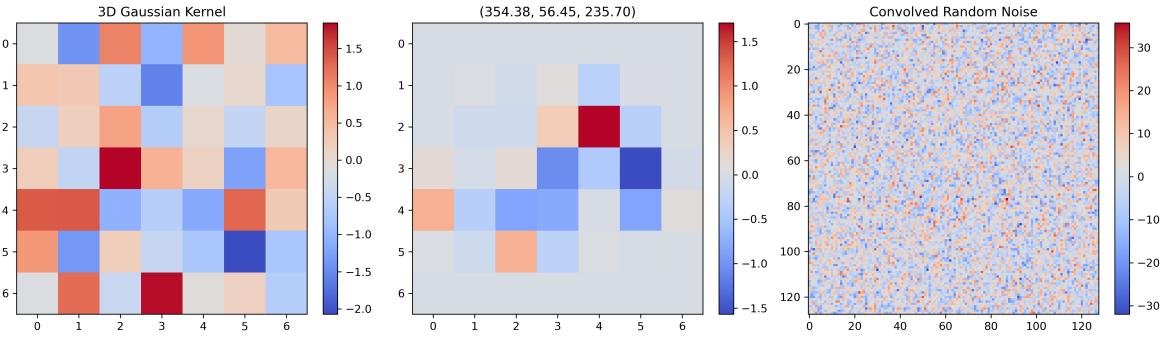


FIGURE 3.10: **Random Normal Steered Gaussian.** Instead of using a uniformly distributed Gaussian for the kernel’s parameters, we sample values from a random normal distribution to form a kernel of dimensions $(7 \times 7 \times 7)$. We notice that a steered convolution using this kernel gives directionally filtered *salt and pepper* noise.

kernel in Algorithm 1.

Algorithm 1 Generation of Steered Spherical Harmonic Filters

- 1: $(G_x, G_y, G_z) \leftarrow \mathcal{N}(0, 1)$ ▷ Sample kernel from a random normal distribution
 - 2: Sample random angles (α, β, γ) and construct \mathbf{R}
 - 3: $G^{\theta'} \leftarrow \mathbf{G} \cdot \mathbf{R}$ ▷ Steer the kernel
 - 4: $r \leftarrow \sqrt{x^2 + y^2 + z^2}$ ▷ Convert Cartesian to Spherical coordinates
 - 5: $\theta \leftarrow \arccos \frac{z}{r}$
 - 6: $\phi \leftarrow \text{sgn}(y) \cdot \arccos \frac{x}{\sqrt{x^2+y^2}}$
 - 7: Compute $Y_\ell^m(\theta, \phi)$
 - 8: $G^{\theta, \phi} \leftarrow G^{\theta'} \cdot \Re(Y_\ell^m(\theta, \phi))$ ▷ Project to Spherical Harmonic Basis
-

Briefly, we begin by steering a Gaussian kernel with randomly sampled angles (α, β, γ) as described in Sec. 3.4.1. The Cartesian representation of the Gaussian is then converted to a spherical coordinate system, which is used to construct the corresponding harmonic basis. Now, we project our steered Gaussian onto the harmonic basis by computing its matrix dot product. Note that we only consider the **real** part of the harmonic basis to maintain a physically realizable representation of the projected kernel. We illustrate a convolution performed by this kernel on random noise in Figure 3.11.

3.5 Segmentation of Light-Sheet Microscopy Data

In this section, we train a StarDist3D model with the **AnyStar++ Gaussian** and **AnyStar++ Spherical** datasets. We subsequently perform segmentation on an example LSM dataset with the Cellpose, **AnyStar**, **AnyStar++ Gaussian**, and **AnyStar++ Spherical** models, and qualitatively compare the results. The source code and model weights are made publicly available.

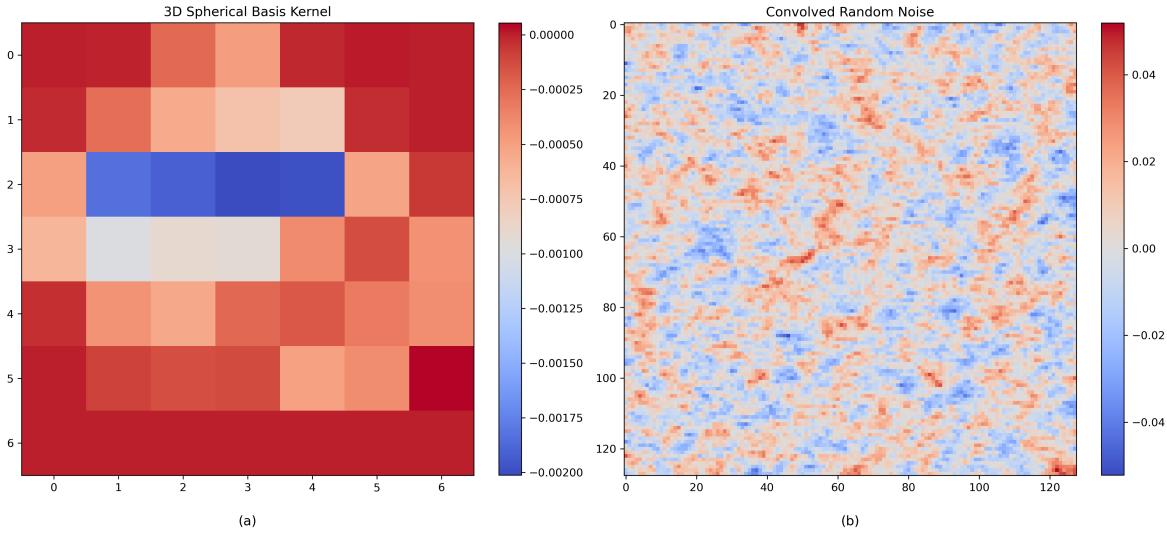


FIGURE 3.11: **Spherical Basis Kernel.** (a) A steered Gaussian kernel projected to a spherical harmonic basis. (b) Result of convolving the resultant kernel with random Gaussian noise.

3.5.1 Modeling Training

We train a StarDist3D network for with the AnyStar++ Gaussian and AnyStar++ Spherical datasets. In particular, we train a 5-resolution 3D U-Net [72] having 32 convolutional channels per axis at the highest resolution (here, 64^3). A list of model names, with their corresponding training dataset is summarized in Table 3.2. Each of the subsequent resolutions are halved from the previous convolutional layer, i.e: $\{64^3, 32^3, 16^3, 8^3, 4^3\}$.

Data Generation Method	Number of Images Generated	Model Trained
AnyStar	27,000	StarDist3D
AnyStar++ Gaussian	27,000	StarDist3D
AnyStar++ Spherical	27,000	StarDist3D

TABLE 3.2: Summary of Models and Corresponding Synthetic Datasets.

We make use of AnyStar's publicly available weights called AnyStar-Mix. For the remaining models (AnyStar++ Gaussian, AnyStar++ Spherical), we train the model from scratch for a total of 250,000 iterations using the Adam optimizer [45] with an initial learning rate of $\alpha = 2 \times 10^{-4}$, which is linearly decayed to 0 over the course of all iterations.

3.5.2 Model Inferences

In this section, we show a qualitative comparison of results obtained from randomly selected slices from the dataset described in Section 2.2.1. This comparison includes Cellpose, AnyStar-Mix, AnyStar++ Gaussian which simulates background using a steerable Gaussian

filter and AnyStar++ Spherical where the background is simulated using a steered spherical harmonic basis filter (Fig. 3.11).

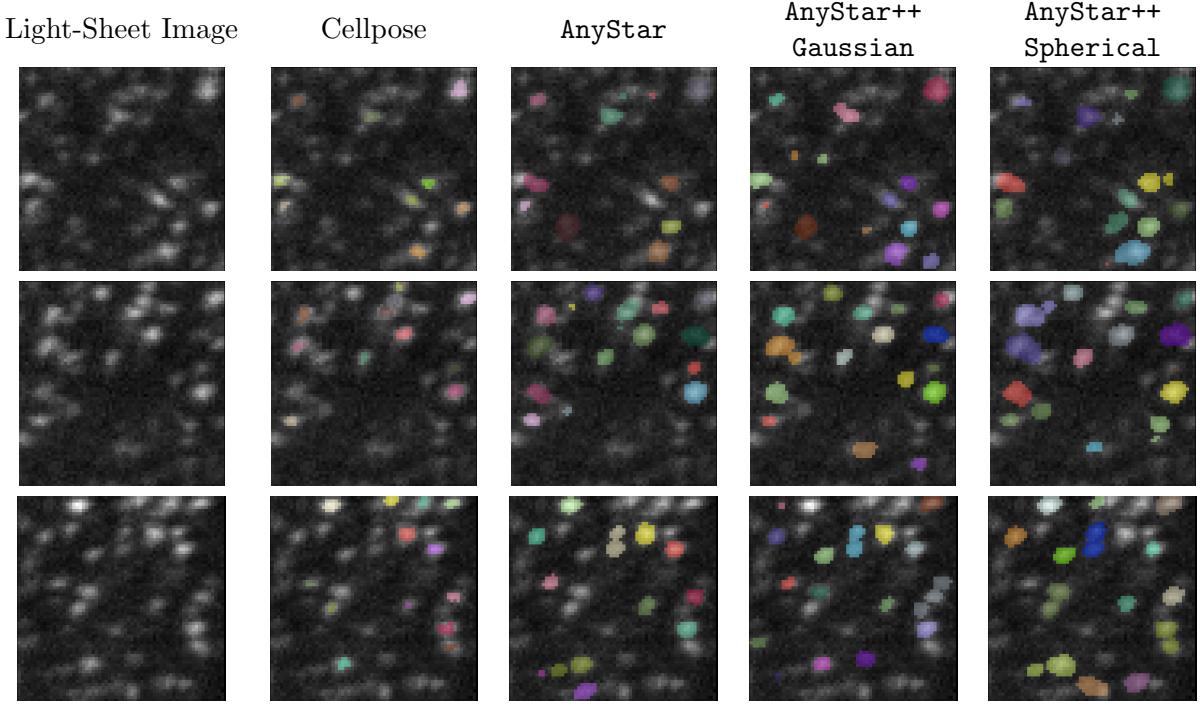


FIGURE 3.11: Qualitative Comparison of Zero-Shot Segmentation Models.

We also summarize the list of trained segmentation models, alongside their analyzed datasets in Table 3.4.

Data Generation Method	Model	Data Analyzed
AnyStar-Mix	StarDist3D	Dandiset 000108
AnyStar++ Gaussian	StarDist3D	Dandiset 000108
AnyStar++ Spherical	StarDist3D	Dandiset 000108
Cellpose	Cellpose Nuclei Model	Dandiset 000108

TABLE 3.4: Summary of Segmentation of Light-Sheet Microscopy.

3.5.3 Putative Analysis

Since we do not have ground truth annotations for nuclei masks in the chosen dataset, we analyze some of the mean geometric measures of the instance masks:

1. Nuclei Volume
2. Major Axis Length

While we do not have access to the mean values of the aforementioned metrics for the given dataset, we believe that appropriately re-scaling pixel space values to physical units could serve as a plausible sanity check of segmentation quality. Note that all analyses done below is for a randomly selected ROI of dimensions $(64 \times 64 \times 64)$.

Nuclei Volume. As we do not know the exact shapes of nuclei *a priori*, we compute the volume of each mask by estimating the volume of its convex hull. A distribution of nuclei volume is plotted across all different segmentation models in Figure 3.12.

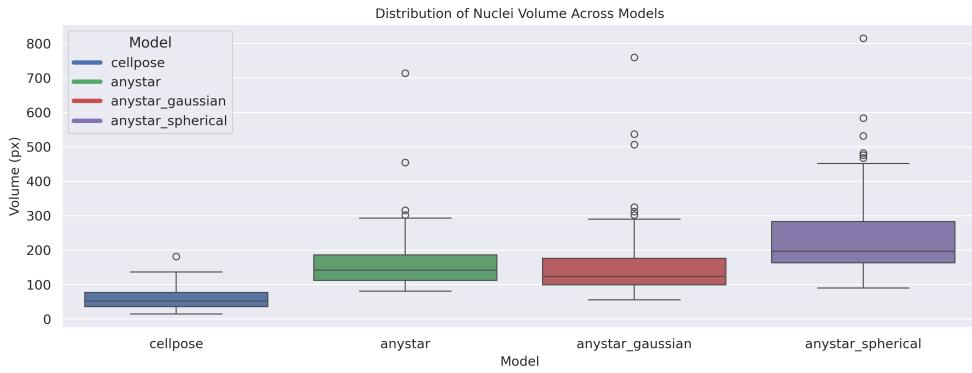


FIGURE 3.12: **Distribution of Nuclei Volume.** We show the distribution of nuclei volume estimated from its convex hull, for the previously described models. The unit of volume measure here is in px^3 .

Major Axis Length. This is computed using length of the major axis of the ellipse that has an equivalent normalized second central moments as the described instance mask. The distribution of major axis lengths of the computed instance masks is shown in Figure 3.13.

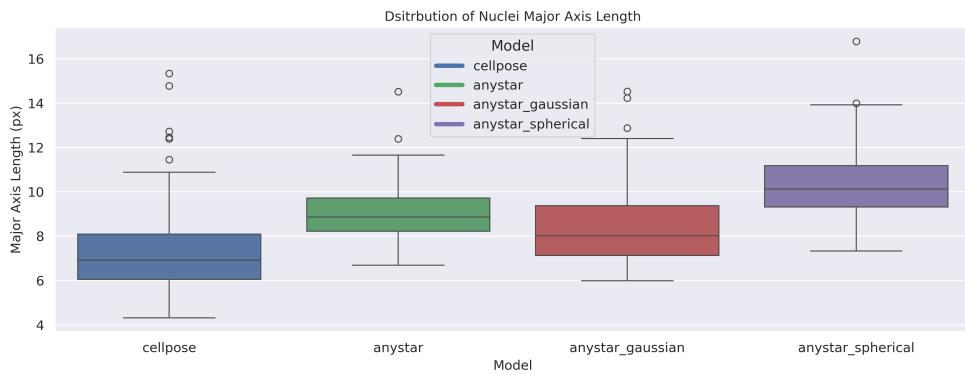


FIGURE 3.13: **Distribution of Major Axis Length.** We show the variation of the lengths of major axes across the different models summarized earlier. The unit of measure of major axis length is in px.

3.6 Code and Data Availability

The source code for this project is available on GitHub at <https://github.com/lincbrain/lsm-segmentation>.

3.7 Discussion

In this chapter we began by describing the image-label pair synthesis process using different intensity models (Secs. 3.1.1, 3.1.2). We then moved to discuss some preliminary ideas about spherical harmonics (Sec. 3.2) and described their properties which make them useful for designing steerable filters (Sec. 3.4), capable of achieving symmetricity and forming a flexible basis vector representation set. We then proposed a steerable filter based on a spherical harmonic basis for synthesizing light-sheet microscopy images, and describe our algorithm in Sec. 3.4.2. Finally, we conduct a qualitative analysis of the same model (StarDist3D [91]) trained with our synthetic data and those by `AnyStar`, and show that our approach better captures the variances inherent to light-sheet microscopy through directional and symmetric manipulation of background signal noise. We also perform a rudimentary qualitative analysis of some geometric measures of the segmented instance masks, and note that this could be a potential technique to quantify the accuracy of segmentation models in the absence of ground-truth annotations.

Chapter 4

Discussion and Conclusions

In this thesis, we have developed and evaluated a distributed image segmentation workflow tailored for large-scale datasets. In this effort, our primary objective was to reduce inference times significantly, which is crucial for handling the ever-increasing volume of biological imaging data. This is especially useful for scientists, when they wish to have faster feedback to iterate over their experiments. We demonstrate, albeit with baby steps, the scalability of our approach, indicating its potential for broader applications in HPC clusters (*e.g.*, distributing computation over multiple nodes). We also show some preliminary experimentation, profiling different tasks in our proposed workflow, providing some insights into its performance.

While evaluating our stitching algorithm, we made use of IoU and nuclei count for evaluating the quality of the algorithm. However, the caveat to this technique is that it is based on certain heuristics, and hence may not necessarily translate to an accurate representation of an algorithms performance. For instance, when we binarize volumes for computing IoU, we are no longer able to quantify whether or not a *false-join* or a *false-merge* has taken place, since all instances have been collapsed to a single identity. However, having some form of robust statistical measurements that can operate in scenarios where there is **no** annotated data would be extremely crucial in the field.

We also introduce a domain-randomized generative model for synthesizing a realistic light-sheet microscopy dataset for training segmentation models. Here, we leverage some useful properties of spherical harmonics to capture the variance in image acquisition of such data. We demonstrate qualitatively how our synthesized dataset allows models to perform **zero-shot** segmentation on unseen datasets, and surpass current state-of-the-art. However, due to lack of ground truth, it becomes non-trivial to quantify our models performance. We make an attempt to quantify this by estimating some key geometric characteristics of the instance masks which is less than ideal for robust quantification.

In summary, we believe that the following could be useful future avenues to explore:

- Distribution of computation across computation nodes in an HPC, and see how this scales with dataset size.
- Robust quantification measures for estimating algorithmic performance in the absence of ground-truth data.

Appendix A

Proofs and Derivations

Much of the proofs and derivations here have ideas that have been directly (or indirectly) borrowed from Griffith's textbook on Quantum Mechanics [30].

A.1 Spherical Polynomials

Consider the Laplacian in spherical coordinates.

$$\nabla^2 = \frac{1}{r^2} \frac{\partial}{\partial r} \left(r^2 \frac{\partial}{\partial r} \right) + \frac{1}{r^2 \sin^2 \theta} \frac{\partial}{\partial \theta} \left(\sin \theta \frac{\partial}{\partial \theta} \right) + \frac{1}{r^2 \sin^2 \theta} \frac{\partial^2}{\partial \phi^2} \quad (\text{A.1})$$

We denote the RHS of Equation A.1 as a function f parameterized by (r, θ, ϕ) . Hence, this gives us the following:

$$\nabla^2 f(r, \theta, \phi) = 0$$

To solve this, we observe that the LHS is can be decomposed into independent, individual functions of each of the variables. Using the method of separation of variables [57] yields

$$f(r, \theta, \phi) = g(r)h(\theta)k(\phi)$$

Substituting this in A.1 and multiplying throughout with $r^2/g(r)h(\theta)k(\phi)$ gives

$$\frac{1}{g(r)} \frac{d}{dr} \left(r^2 \frac{dg}{dr} \right) + \frac{1}{h(\theta) \sin^2 \theta} \frac{d}{d\theta} \left(\sin \theta \frac{dh}{d\theta} \right) + \frac{1}{k(\phi) \sin^2 \theta} \frac{d^2 k}{d\phi^2} = 0$$

This allows us to separate variables into two unique ordinary differential equations. As a result, we note that each of the equations must be equivalent to a constant.

$$\frac{1}{k} \frac{d^2 k}{d\phi^2} = \frac{1}{g} \frac{d}{dr} \left(r^2 \frac{dg}{dr} \right) + \frac{1}{h \sin^2 \theta} \frac{d}{d\theta} \left(\sin \theta \frac{dh}{d\theta} \right) = -m^2 \quad (\text{A.2})$$

For reasons that become evident later, we transform the separation constant m , to $\ell(\ell + 1)$.

$$\frac{1}{g} \frac{d}{dr} \left(r^2 \frac{dg}{dr} \right) = -\frac{1}{h \sin \theta} \frac{d}{d\theta} \left(\sin \theta \frac{dh}{d\theta} \right) + \frac{m^2}{\sin^2 \theta} = \ell(\ell + 1)$$

Now, the resultant radial equation reduces to

$$r^2 \frac{d^2 g}{dr^2} + 2r \frac{dg}{dr} - \ell(\ell + 1)g = 0 \quad (\text{A.3})$$

This is recognized to be an Euler equation [71], where the analytic solution is of the form $g = r^s$. We substitute g in Eq. A.3 to obtain the value of the exponent s .

$$\begin{aligned} r^2 \cdot s(s-1)r^{s-2} + 2r \cdot sr^{s-1} - \ell(\ell+1)r^s &= 0 \\ \therefore s(s+1) = \ell(\ell+1) &\implies s = \ell \text{ or } s = (-\ell-1) \end{aligned}$$

Applying a similar idea to the angular equation, we have

$$\frac{1}{\sin \theta} \frac{d}{d\theta} \left(\sin \theta \frac{dh}{d\theta} \right) + \left(\ell(\ell+1) - \frac{m^2}{\sin^2 \theta} \right) h = 0$$

Changing variables from $x \rightarrow \cos \theta$ and $y \rightarrow h(\theta)$ reduces Equation A.1 to

$$(1-x^2) \frac{d^2 y}{dx^2} - 2x \frac{dy}{dx} + \left(\ell(\ell+1) - \frac{m^2}{\sin^2 \theta} \right) y = 0$$

This is a differential equation representing associated Legendre polynomials [13]:

$$y(x) = P_\ell^m(x); \quad \text{for } \ell \in \{0, 1, 2, \dots\} \text{ and } m \in \{-\ell, -\ell+1, \dots, \ell-1, \ell\}$$

where each of the $P_\ell^m(x)$ have the following closed form representation

$$P_\ell^m(x) = \frac{(-1)^m}{2^\ell \ell!} (1-x^2)^{m/2} \frac{d^{\ell+m}}{dx^{\ell+m}} (x^2 - 1)^\ell \text{ for } m \in \{-\ell, -\ell+1, \dots, \ell-1, \ell\}$$

Since Eq. A.2 is invariant to the sign of m , it follows that $P_\ell^m(x)$ and $P_\ell^{-m}(x)$ have equivalent solutions. Using this, it can be shown that

$$P_\ell^{-m}(\cos \theta) = (-1)^m \frac{(\ell-m)!}{(\ell+m)!} P_\ell^m(\cos \theta)$$

which is then used to construct a general solution to Eq. A.1.

$$f(r, \theta, \phi) = \begin{Bmatrix} r^\ell \\ r^{\ell-1} \end{Bmatrix} P_\ell^m(\cos \theta) \begin{Bmatrix} e^{im\phi} \\ e^{-im\phi} \end{Bmatrix}$$

where ℓ and m exist in their respective domains. In obtaining this solution, we can introduce the spherical harmonics $Y_\ell^m(\theta, \phi)$ as a function of the associated Legendre polynomials

$$Y_\ell^m(\theta, \phi) = (-1)^m \sqrt{\frac{(2\ell+1)}{4\pi} \frac{(\ell-m)!}{(\ell+m)!}} P_\ell^m(\cos \theta) e^{im\phi}$$

A.2 Orthogonality of Spherical Harmonics

In this section, we prove that a pair of spherical harmonics are orthogonal to one another over a sphere.

Theorem A.1. *For a set of spherical harmonics $\{Y_\ell^m, Y_{\ell'}^n\}$, of degrees n and m respectively, defined on a sphere of radius r the following must hold:*

$$\int_{\omega} Y_\ell^m Y_{\ell'}^n d\omega = 0 \quad (m \neq n)$$

Or, spherical harmonics of varying degrees are orthogonal over a sphere, where $d\omega = \sin \theta d\theta d\phi$, which is the differential solid angle in spherical coordinates.

Proof. To trivialize the integral, we consider representing the spherical harmonics using Euler's formula [3]. Hence, we would like to express a spherical harmonic function in the form

$$Y_\ell^m(\theta, \phi) = (\cos \theta + i \sin \theta)^m \cdot P_\ell^m(\cos \phi) \tag{A.4}$$

where $P_m^\ell(\cdot)$ is the associated Legendre polynomial. We note that a sphere of radius r centered at the origin can be parameterized as

$$\psi(\theta, \phi) = \langle r \cos \theta \sin \phi, r \cos \phi, r \sin \theta \sin \phi \rangle \text{ for } (\theta, \phi) \in [0, \pi] \times [0, 2\pi]$$

Using this, we can rewrite Eq. A.4. Without loss of generality, we set the radius of the sphere $r = 1$, which gives us the following

$$\begin{aligned} Y_\ell^m &= \left(\frac{x}{\sin \phi} + i \frac{z}{\sin \phi} \right)^m \cdot P_\ell^m(y) \\ &= (x + iz)^m \cdot \frac{P_\ell^m(y)}{\sin^m \phi} \\ &= (x + iz)^m \cdot \frac{P_\ell^m(y)}{(1 - y^2)^{m/2}} \end{aligned}$$

Now, we can expand the domain ω across the domains of (θ, ϕ) to yield

$$I = \int_{\omega} Y_\ell^m Y_{\ell'}^n d\omega = \int_0^\pi \int_0^{2\pi} \left(e^{im\theta} \cdot P_m^\ell(\cos \phi) \right) \cdot \left(e^{in\theta} \cdot P_{\ell'}^n(\cos \phi) \right) d\theta \sin \phi d\phi$$

This can integral can be grouped into 2 independent integrals, I_1 and I_2 .

$$I = \underbrace{\left(\int_0^\pi P_\ell^m(\cos \phi) \cdot P_{\ell'}^n(\cos \phi) \sin \phi d\phi \right)}_{I_1} \cdot \underbrace{\left(\int_0^{2\pi} e^{i(m-n)\theta} d\theta \right)}_{I_2}$$

Now, we observe that $I_2 = 0$ whenever $m \neq n$:

$$I_2 = \frac{1}{i(m-n)} e^{i(m-n)\theta} \Big|_0^{2\pi} = 0$$

and the result for $m \neq n$ follows. □

Bibliography

- [1] Misha B Ahrens et al. “Whole-brain functional imaging at cellular resolution using light-sheet microscopy”. In: *Nature methods* 10.5 (2013), pp. 413–420.
- [2] Shahar Alon et al. “Expansion sequencing: Spatially precise in situ transcriptomics in intact biological systems”. In: *Science* 371.6528 (2021), eaax2656.
- [3] Tom M Apostol. “Another elementary proof of Euler’s formula for $\zeta(2n)$ ”. In: *The American Mathematical Monthly* 80.4 (1973), pp. 425–431.
- [4] Benjamin Billot et al. “A learning strategy for contrast-agnostic MRI segmentation”. In: *arXiv preprint arXiv:2003.01995* (2020).
- [5] Benjamin Billot et al. “Partial volume segmentation of brain MRI scans of any resolution and contrast”. In: *Medical Image Computing and Computer Assisted Intervention-MICCAI 2020: 23rd International Conference, Lima, Peru, October 4–8, 2020, Proceedings, Part VII 23*. Springer. 2020, pp. 177–187.
- [6] Benjamin Billot et al. “SynthSeg: Segmentation of brain MRI scans of any contrast and resolution without retraining”. In: *Medical image analysis* 86 (2023), p. 102789.
- [7] Miguel A Blanco, Manuel Flórez, and Margarita Bermejo. “Evaluation of the rotation matrices in the basis of real spherical harmonics”. In: *Journal of Molecular structure: THEOCHEM* 419.1-3 (1997), pp. 19–27.
- [8] Matthew B Bouchard et al. “Swept confocally-aligned planar excitation (SCAPE) microscopy for high-speed volumetric imaging of behaving organisms”. In: *Nature photonics* 9.2 (2015), pp. 113–119.
- [9] Ed Bullmore and Olaf Sporns. “Complex brain networks: graph theoretical analysis of structural and functional systems”. In: *Nature reviews neuroscience* 10.3 (2009), pp. 186–198.
- [10] Danilo Bzdok and BT Thomas Yeo. “Inference in the age of big data: Future perspectives on neuroscience”. In: *Neuroimage* 155 (2017), pp. 549–564.
- [11] Juan C Caicedo et al. “Evaluation of deep learning strategies for nucleus segmentation in fluorescence images”. In: *Cytometry Part A* 95.9 (2019), pp. 952–965.

- [12] Bi-Chang Chen et al. “Lattice light-sheet microscopy: imaging molecules to embryos at high spatiotemporal resolution”. In: *Science* 346.6208 (2014), p. 1257998.
- [13] Chang-Yuan Chen et al. “Exact solutions to a class of differential equation and some new mathematical properties for the universal associated-Legendre polynomials”. In: *Applied Mathematics Letters* 40 (2015), pp. 90–96.
- [14] Fei Chen et al. “Nanoscale imaging of RNA with expansion microscopy”. In: *Nature methods* 13.8 (2016), pp. 679–684.
- [15] Hao Chen et al. “DCAN: deep contour-aware networks for accurate gland segmentation”. In: *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. 2016, pp. 2487–2496.
- [16] Hao Chen et al. “DCAN: Deep contour-aware networks for object instance segmentation from histology images”. In: *Medical image analysis* 36 (2017), pp. 135–146.
- [17] Liang-Chieh Chen et al. “Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs”. In: *IEEE transactions on pattern analysis and machine intelligence* 40.4 (2017), pp. 834–848.
- [18] Chi-Li Chiu and Nathan Clack. “napari: a Python Multi-Dimensional Image Viewer Platform for the Research Community”. In: *Microscopy and Microanalysis* 28.S1 (2022), 1576–1577. DOI: [10.1017/S1431927622006328](https://doi.org/10.1017/S1431927622006328).
- [19] Pina Colarusso and Craig Brideau. *Two-Photon Microscopy: How to Minimize Optical Aberrations – Bliq Photonics — bliqphotonics.com*. <https://bliqphotonics.com/two-photon-microscopy-how-to-minimize-optical-aberrations/>. [Accessed 25-06-2024].
- [20] Adrian V Dalca, John Guttag, and Mert R Sabuncu. “Anatomical priors in convolutional networks for unsupervised biomedical segmentation”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 9290–9299.
- [21] Bert De Brabandere, Davy Neven, and Luc Van Gool. “Semantic instance segmentation for autonomous driving”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 2017, pp. 7–9.
- [22] Terrance DeVries and Graham W Taylor. “Improved regularization of convolutional neural networks with cutout”. In: *arXiv preprint arXiv:1708.04552* (2017).
- [23] Neel Dey et al. “AnyStar: Domain randomized universal star-convex 3D instance segmentation”. In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2024, pp. 7593–7603.

- [24] Sven Dorkenwald et al. “Neuronal wiring diagram of an adult brain”. In: *bioRxiv* (2023). DOI: [10.1101/2023.06.27.546656](https://doi.org/10.1101/2023.06.27.546656). eprint: <https://www.biorxiv.org/content/early/2023/07/11/2023.06.27.546656.full.pdf>. URL: <https://www.biorxiv.org/content/early/2023/07/11/2023.06.27.546656>.
- [25] Nils Eckstein et al. “Neurotransmitter Classification from Electron Microscopy Images at Synaptic Sites in *Drosophila Melanogaster*”. In: *bioRxiv* (2023). DOI: [10.1101/2020.06.12.148775](https://doi.org/10.1101/2020.06.12.148775). eprint: <https://www.biorxiv.org/content/early/2023/05/10/2020.06.12.148775.full.pdf>. URL: <https://www.biorxiv.org/content/early/2023/05/10/2020.06.12.148775>.
- [26] Christoph J Engelbrecht and Ernst H Stelzer. “Resolution enhancement in a light-sheet-based microscope (SPIM)”. In: *Optics letters* 31.10 (2006), pp. 1477–1479.
- [27] William T Freeman, Edward H Adelson, et al. “The design and use of steerable filters”. In: *IEEE Transactions on Pattern analysis and machine intelligence* 13.9 (1991), pp. 891–906.
- [28] Ruixuan Gao et al. “Cortical column and whole-brain imaging with molecular contrast and nanoscale resolution”. In: *Science* 363.6424 (2019), eaau8302.
- [29] Sarah Gibson, Jack W Judy, and Dejan Markovic. “Comparison of spike-sorting algorithms for future hardware implementation”. In: *2008 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE. 2008, pp. 5015–5020.
- [30] David J Griffiths and Darrell F Schroeter. *Introduction to quantum mechanics*. Cambridge university press, 2018.
- [31] Hákon Gudbjartsson and Samuel Patz. “The Rician distribution of noisy MRI data”. In: *Magnetic resonance in medicine* 34.6 (1995), pp. 910–914.
- [32] Régis Guillemaud and Michael Brady. “Estimating the bias field of MR images”. In: *IEEE Transactions on Medical imaging* 16.3 (1997), pp. 238–251.
- [33] VV Gusel’Nikova and DE4463411 Korzhevskiy. “NeuN as a neuronal nuclear antigen and neuron differentiation marker”. In: *Acta Naturae* 7.2 (25) (2015), pp. 42–47.
- [34] Mark J Harris et al. “Simulation of cloud dynamics on graphics hardware”. In: *Proceedings of the ACM SIGGRAPH/EUROGRAPHICS conference on Graphics hardware*. 2003, pp. 92–101.
- [35] Larissa Heinrich et al. “Synaptic Cleft Segmentation in Non-isotropic Volume Electron Microscopy of the Complete *Drosophila* Brain”. In: *Medical Image Computing and Computer Assisted Intervention – MICCAI 2018*. Ed. by Alejandro F. Frangi et al. Cham: Springer International Publishing, 2018, pp. 317–325. ISBN: 978-3-030-00934-2.
- [36] Malte Hoffmann et al. “SynthMorph: learning contrast-invariant registration without acquired images”. In: *IEEE Transactions on Medical Imaging* 41.3 (2022), pp. 543–558.

- [37] Ying S Hu et al. “Single-molecule super-resolution light-sheet microscopy”. In: *ChemPhysChem* 15.4 (2014), pp. 577–586.
- [38] Google Inc. *Neuroglancer*. <https://github.com/google/neuroglancer>. 2016.
- [39] Jaber Juntu et al. “Bias field correction for MRI images”. In: *Computer Recognition Systems: Proceedings of the 4th International Conference on Computer Recognition Systems CORES’05*. Springer. 2005, pp. 543–551.
- [40] Lee Kamentsky et al. *Light sheet imaging of the human brain (Version draft. jv. [Data set]. DANDI archive*. 2023. doi: [10.80507/dandi.123456/0.123456.1234](https://doi.org/10.80507/dandi.123456/0.123456.1234). URL: <https://doi.org/10.80507/dandi.123456/0.123456.1234>.
- [41] Yi-Hsuan Kao and James R MacFall. “Correction of MR k-space data corrupted by spike noise”. In: *IEEE transactions on medical imaging* 19.7 (2000), pp. 671–680.
- [42] Michael Kazhdan, Thomas Funkhouser, and Szymon Rusinkiewicz. “Rotation invariant spherical harmonic representation of 3 d shape descriptors”. In: *Symposium on geometry processing*. Vol. 6. 2003, pp. 156–164.
- [43] Philipp J Keller and Hans-Ulrich Dodt. “Light sheet microscopy of living or cleared specimens”. In: *Current opinion in neurobiology* 22.1 (2012), pp. 138–143.
- [44] George Kelly and Hugh McCabe. “A survey of procedural techniques for city generation”. In: *The ITB Journal* 7.2 (2006), p. 5.
- [45] Diederik P Kingma and Jimmy Ba. “Adam: A method for stochastic optimization”. In: *arXiv preprint arXiv:1412.6980* (2014).
- [46] *Lab Notes — Why don’t we understand the brain? — alleninstitute.org*. [Accessed 24-06-2024].
- [47] Ares Lagae et al. “A survey of procedural noise functions”. In: *Computer Graphics Forum*. Vol. 29. 8. Wiley Online Library. 2010, pp. 2579–2600.
- [48] Ed Lein, Lars E Borm, and Sten Linnarsson. “The promise of spatial transcriptomics for neuroscience in the era of molecular cell typing”. In: *Science* 358.6359 (2017), pp. 64–69.
- [49] Talia N Lerner, Li Ye, and Karl Deisseroth. “Communication in neural circuits: tools, opportunities, and challenges”. In: *Cell* 164.6 (2016), pp. 1136–1150.
- [50] Albert Lin et al. “Network Statistics of the Whole-Brain Connectome of Drosophila”. In: *bioRxiv* (2024). doi: [10.1101/2023.07.29.551086](https://doi.org/10.1101/2023.07.29.551086). eprint: <https://www.biorxiv.org/content/early/2024/02/28/2023.07.29.551086.full.pdf>. URL: <https://www.biorxiv.org/content/early/2024/02/28/2023.07.29.551086>.
- [51] Peter Lindqvist. *Notes on the p-Laplace equation*. 161. University of Jyväskylä, 2017.
- [52] Songlei Liu et al. “Barcoded oligonucleotides ligated on RNA amplified for multiplexed and parallel in situ analyses”. In: *Nucleic acids research* 49.10 (2021), e58–e58.

- [53] Wei Liu et al. “Ssd: Single shot multibox detector”. In: *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*. Springer. 2016, pp. 21–37.
- [54] Arie Matsliah et al. “Neuronal ‘parts list’ and wiring diagram for a visual system”. In: *bioRxiv* (2024). DOI: [10.1101/2023.10.12.562119](https://doi.org/10.1101/2023.10.12.562119). eprint: <https://www.biorxiv.org/content/early/2024/04/15/2023.10.12.562119.full.pdf>. URL: <https://www.biorxiv.org/content/early/2024/04/15/2023.10.12.562119>.
- [55] Dagmar Medková. “The Laplace equation”. In: *Boundary value problems on bounded and unbounded Lipschitz domains*. Springer, Cham (2018).
- [56] *mesoSPIM features — mesospim.org*. <https://mesospim.org/features/>. [Accessed 25-06-2024].
- [57] Willard Miller Jr. “Symmetry and separation of variables”. In: (1977).
- [58] Josh Moore et al. “OME-Zarr: a cloud-optimized bioimaging file format with international community support”. In: *Histochemistry and Cell Biology* 160.3 (2023), pp. 223–251.
- [59] Marius Pachitariu and Carsen Stringer. “Cellpose 2.0: how to train your own model”. In: *Nature methods* 19.12 (2022), pp. 1634–1641.
- [60] Juhyuk Park et al. “Integrated platform for multi-scale molecular imaging and phenotyping of the human brain”. In: *BioRxiv* (2022), pp. 2022–03.
- [61] Young-Gyun Park et al. “Protection of tissue physicochemical properties using polyfunctional crosslinkers”. In: *Nature biotechnology* 37.1 (2019), pp. 73–83.
- [62] Andrew C Payne et al. “In situ genome sequencing resolves DNA sequence and structure in intact biological samples”. In: *Science* 371.6532 (2021), eaay3446.
- [63] Pietro Perconti and Alessio Plebe. “Deep learning and cognitive science”. In: *Cognition* 203 (2020), p. 104365.
- [64] Ken Perlin. “An image synthesizer”. In: *ACM Siggraph Computer Graphics* 19.3 (1985), pp. 287–296.
- [65] Daniele Perrone et al. “The effect of Gibbs ringing artifacts on measures derived from diffusion MRI”. In: *Neuroimage* 120 (2015), pp. 441–455.
- [66] Alessio Plebe and Giorgio Grasso. “The unbearable shallow understanding of deep learning”. In: *Minds and Machines* 29.4 (2019), pp. 515–553.
- [67] Adrien Poulenard et al. “Effective rotation-invariant point cnn with spherical harmonics kernels”. In: *2019 International Conference on 3D Vision (3DV)*. IEEE. 2019, pp. 47–56.
- [68] Aditya Pratapa, Michael Doron, and Juan C Caicedo. “Image-based cell phenotyping with deep learning”. In: *Current opinion in chemical biology* 65 (2021), pp. 9–17.

- [69] Joseph Redmon et al. “You only look once: Unified, real-time object detection”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 779–788.
- [70] Shaoqing Ren et al. “Faster r-cnn: Towards real-time object detection with region proposal networks”. In: *Advances in neural information processing systems* 28 (2015).
- [71] Philip L Roe. “Characteristic-based schemes for the Euler equations”. In: *Annual review of fluid mechanics* 18.1 (1986), pp. 337–365.
- [72] 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. Springer. 2015, pp. 234–241.
- [73] Philipp Schlegel et al. “Whole-brain annotation and multi-connectome cell typing quantifies circuit stereotypy in Drosophila”. In: *bioRxiv* (2023). DOI: [10.1101/2023.06.27.546055](https://doi.org/10.1101/2023.06.27.546055). eprint: [https://www.biorxiv.org/content/early/2023/07/15/2023.06.27.546055](https://www.biorxiv.org/content/early/2023/07/15/2023.06.27.546055.full.pdf). URL: <https://www.biorxiv.org/content/early/2023/07/15/2023.06.27.546055>.
- [74] Uwe Schmidt et al. “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*. 2018, pp. 265–273. DOI: [10.1007/978-3-030-00934-2_30](https://doi.org/10.1007/978-3-030-00934-2_30).
- [75] Ken Shoemake. “Animating rotation with quaternion curves”. In: *Proceedings of the 12th annual conference on Computer graphics and interactive techniques*. 1985, pp. 245–254.
- [76] George H Shortley and Royal Weller. “The numerical solution of Laplace’s equation”. In: *Journal of Applied Physics* 9.5 (1938), pp. 334–348.
- [77] Christoph Sommer et al. “Ilastik: Interactive learning and segmentation toolkit”. In: *2011 IEEE international symposium on biomedical imaging: From nano to macro*. IEEE. 2011, pp. 230–233.
- [78] Olaf Sporns. “Structure and function of complex brain networks”. In: *Dialogues in clinical neuroscience* 15.3 (2013), pp. 247–262.
- [79] Katherine R Storrs and Nikolaus Kriegeskorte. “Deep learning for cognitive neuroscience”. In: *arXiv preprint arXiv:1903.01458* (2019).
- [80] Alexandros Stougiannis et al. “Data-driven neuroscience: enabling breakthroughs via innovative data management”. In: *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*. 2013, pp. 953–956.
- [81] Carsen Stringer et al. “Cellpose: a generalist algorithm for cellular segmentation”. In: *Nature methods* 18.1 (2021), pp. 100–106.

- [82] Jessica L Swanson et al. “Advancements in the quest to map, monitor, and manipulate neural circuitry”. In: *Frontiers in Neural Circuits* 16 (2022), p. 886302.
- [83] Lisa Torrey and Jude Shavlik. “Transfer learning”. In: *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques*. IGI global, 2010, pp. 242–264.
- [84] Jelle Veraart et al. “Gibbs ringing in diffusion MRI”. In: *Magnetic resonance in medicine* 76.1 (2016), pp. 301–314.
- [85] Venkatakaushik Voleti et al. “Real-time volumetric microscopy of in vivo dynamics and large-scale samples with SCAPE 2.0”. In: *Nature methods* 16.10 (2019), pp. 1054–1062.
- [86] CL Walsh et al. “Imaging intact human organs with local resolution of cellular structures using hierarchical phase-contrast tomography”. In: *Nature methods* 18.12 (2021), pp. 1532–1541.
- [87] Jie Wang et al. “Instance segmentation of anatomical structures in chest radiographs”. In: *2019 IEEE 32nd International Symposium on Computer-Based Medical Systems (CBMS)*. IEEE. 2019, pp. 441–446.
- [88] Xiao-Jing Wang et al. “Computational neuroscience: a frontier of the 21st century”. In: *National science review* 7.9 (2020), pp. 1418–1422.
- [89] Zeguan Wang et al. “Imaging the voltage of neurons distributed across entire brains of larval zebrafish”. In: *bioRxiv* (2023), pp. 2023–12.
- [90] James R Weaver. “Centrosymmetric (cross-symmetric) matrices, their basic properties, eigenvalues, and eigenvectors”. In: *The American Mathematical Monthly* 92.10 (1985), pp. 711–717.
- [91] Martin Weigert and Uwe Schmidt. “Nuclei Instance Segmentation and Classification in Histopathology Images with Stardist”. In: *The IEEE International Symposium on Biomedical Imaging Challenges (ISBIC)*. 2022. DOI: [10.1109/ISBIC56247.2022.9854534](https://doi.org/10.1109/ISBIC56247.2022.9854534).
- [92] Martin Weigert et al. “Star-convex Polyhedra for 3D Object Detection and Segmentation in Microscopy”. In: *The IEEE Winter Conference on Applications of Computer Vision (WACV)*. 2020. DOI: [10.1109/WACV45572.2020.9093435](https://doi.org/10.1109/WACV45572.2020.9093435).
- [93] Scott B Wilson and Ronald Emerson. “Spike detection: a review and comparison of algorithms”. In: *Clinical Neurophysiology* 113.12 (2002), pp. 1873–1881.
- [94] Steven Worley. “A cellular texture basis function”. In: *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*. 1996, pp. 291–294.
- [95] Weidi Xie, J Alison Noble, and Andrew Zisserman. “Microscopy cell counting and detection with fully convolutional regression networks”. In: *Computer methods in biomechanics and biomedical engineering: Imaging & Visualization* 6.3 (2018), pp. 283–292.

- [96] Bin Yang et al. “Single-cell phenotyping within transparent intact tissue through whole-body clearing”. In: *Cell* 158.4 (2014), pp. 945–958.
- [97] Canyu Yang, Dennis Eschweiler, and Johannes Stegmaier. “Semi-and self-supervised multi-view fusion of 3D microscopy images using generative adversarial networks”. In: *International Workshop on Machine Learning for Medical Image Reconstruction*. Springer. 2021, pp. 130–139.
- [98] Jingyi Yin et al. “Optimized U-Net model for 3D light-sheet image segmentation of zebrafish trunk vessels”. In: *Biomedical Optics Express* 13.5 (2022), pp. 2896–2908.
- [99] Ziyu Zhang, Sanja Fidler, and Raquel Urtasun. “Instance-level segmentation for autonomous driving with deep densely connected mrfss”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016, pp. 669–677.
- [100] Yanning Zhou et al. “Cia-net: Robust nuclei instance segmentation with contour-aware information aggregation”. In: *Information Processing in Medical Imaging: 26th International Conference, IPMI 2019, Hong Kong, China, June 2–7, 2019, Proceedings 26*. Springer. 2019, pp. 682–693.