Title

Data-Driven Information-Optimal Computational Microscopy

Author

Pinkard, Henry

Publication Date

2022

Peer reviewed|Thesis/dissertation

Data-Driven Information-Optimal Computational Microscopy

by

Henry Baker Pinkard

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Computational Biology

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Laura Waller, Co-chair
Professor Jennifer Listgarten, Co-chair
Associate Professor Aaron Streets
Professor Jitendra Malik

Summer 2022

Data-Driven Information-Optimal Computational Microscopy

Abstract

Data-Driven Information-Optimal Computational Microscopy

by

Henry Baker Pinkard

Doctor of Philosophy in Computational Biology

University of California, Berkeley

Professor Laura Waller, Co-chair

Professor Jennifer Listgarten, Co-chair

Optical microscopes have been an indispensable tool in biology and medicine for over three centuries. Unlike their simple predecessors, contemporary microscopes often employ complex robotic automation and customized algorithms. In the past decade, advances in high-performance computer processors, the ease of collecting massive datasets, and machine learning have created many new possibilities for data-driven approaches to microscope control and image analysis.

This dissertation covers the challenges and opportunities in modern microscopy. First, it shows how neural networks can be used to create microscopes that adapt to the samples they are imaging in real time. For example, this paradigm can be used to quickly focus microscopes using inexpensive hardware or visualize developing immune responses at large scales. Next, new open-source software that facilitates development of these and other microscopy techniques is presented. Next, it turns to how microscopes can make measurements of the intrinsic optical properties of cells, from which their biological function can be inferred. Development of techniques that do so requires comparing approaches on standardized datasets, and the creation of such a dataset containing hundreds of thousands of images of single cells is described. Finally, a new theoretical framework for modeling the information transmission of both microscopes and image-processing algorithms is introduced. This perspective provides a new set of engineering principles for microscopes and opens a range of new research questions.

To my family,

Without whom I would be nowhere.

An investment in knowledge pays the best interest.
- Benjamin Franklin

Contents

List of Figures

Acknowledgments

The task of expressing the full depth of my gratitude to all those who have helped me along the way to the end of this journey cannot be done full justice in a few written paragraphs. But I would be remiss if I did not attempt to express my deepest thanks to the the many people who have made this journey possible.

Over the years it has resonated with me to hear others describe the process of "following their curiosity". However in retrospect, it seems to me that in fact I had less agency in the process than the phrase implies. A more apt description would be that my curiosity *compelled* me to follow the path I took (though I certainly enjoyed the ride). This would not have been possible without those who showed me to the fields of my own ignorance where I could plant the seeds of interest, as well as the support, patience, and encouragement of those who allowed them to grow into the plants of knowledge.

First and foremost is my advisor, Laura Waller. My first encounter with her was seeing her talk at a microscopy conference at Berkeley two years before I entered grad school. I remember being dumbfounded by her presentation. On the surface, I understood *what* she had done, but it was clear to me that not only did I not know *how* she had done it, I wasn't even aware of the existence of the areas of knowledge from which she was drawing. A couple years later, I had the privilege of interacting with her directly–first as a student in two of her classes, and later as a member of her lab. She has been and incredible role model and mentor all these years. I've never ceased to be amazed and appreciative of the clarity and simplicity with which she can cut through layers of noise and imprecision to express complex ideas; Her patience in answering my stream of random questions; the culture of collaboration and cooperation she fosters; her integrity and ability to treat others fairly and with kindness; her ability to adapt and work with others from a wide variety of styles and backgrounds; and her subtle yet delightful sense of humor. I am incredibly grateful for her trust and encouragement to delve deeply into the unknown with an uncertain path and uncertain destination.

Two other mentors deserve a special thanks for creating the path that led me to Berkeley in the first place. Nico Stuurman gave me my first job out of college (for which my qualifications were dubious, at best). This was my first introduction to microscopes and set the stage for everything that followed. His generosity to others with his time and expertise continues to inspire me today, and have allowed me to see how rewarding it can be to be a part of a scientific project larger than oneself. Max Krummel gave me my second job, and showed me how fun science can be. Whether building new microscopes or shattering bananas after submerging them in liquid nitrogen, his enthusiasm is infectious. I had never seriously considered getting a PhD until seeing him in action.

Matt Thomson played a small but significant role in my trajectory, when, during a grad school interview at UCSF, he informed me about the existence of information theory, and recommended the best textbook I've ever read, "Information Theory, Inference, and Learning Algorithms" by David MacKay. My immense confusion after reading the first five pages of this book created a North Star of knowledge to strive towards over the next several years, and later the basis for the final third of this dissertation.

The Waller lab has been full of so many people from whom I have learned immensely and shared many good times. I cannot express the depth of my gratitude to have worked with so many talented, fantastic people. I absorbed so much optimization and optics knowledge from my cubicle-mates, Shwetadwip Chowdhury and Emrah Bostan; Regina Eckert and Kristina Monakhova's generosity with their time and commitment to improving the department culture was inspirational and instructive; Michael Kellman and Eric Markley were and are great rock climbing partners; Kyrollos Yanny never failed to make me laugh during our experience GSIing together; Li-Hao Yeh, Michael Chen, Zack Phillips were patient and generous in sharing their expertise during my early years in the lab; Tiffany Chien, Leyla Kabuli, and Amit Kohli have done more than anyone to understand and help me untangle and organize the chaotic pile of ideas in my head.

Collaboration and exchanges with a many talented peers and mentors have enriched my time and Berkeley. This started from day 1 at Berkeley, when I was fortunate to be in a life raft of study group with Suzanne Dufault and Ivana Malenica in the tempestuous ocean of first semester probability and statistics classes. Soon after, I began the first of many independent study groups with Chenling Antelope Xu, whose thought-provoking questions and unique perspectives to this day make me examine my assumptions and come to a deeper understand of topics. I learned the ins and outs of algorithms with Aviva Bulow and Nick Everetts. Eli Rosas was a wonderful and uplifting partner not only in optical engineering, but also in learning to play tennis. I cannot recall a discussion I've had with Connor Bybee where I haven't learned something new. Likewise with Anastasios Angelopoulos. I never ceased to be impressed by the undergrads I worked with: Arman Babakhani, Fanice Nyatigo, Cherry Liu, Grace Zhang, and Ryan Mei. The Biological Imaging Development Center at UCSF was always a welcoming home away from home thanks to Adam Fries, Jordan Briscoe, Kyle Marchuk, and Hratch Baghdassarian. Neil Switz was a filled my head with all sorts of random microscope facts. Steve Conolly was a fantastic professor to GSI for, and the depth with which he cared about students was always an inspiration. I always had something to learn from Mark Tsuchida and Nick Anthony about open source software development. Likewise for Stéfan van der Walt, who taught me to fully appreciate the power of Python. My thesis committee and qualifying exam committees, Jennifer Listgarten, Aaron Streets, Jitendra Malik, and Nir Yosef were excellent role models and I was always inspired by their work.

Being in the third ever cohort of the Computational Biology Graduate Group, it has been exciting to watch the program and the many wonderful people in it grow over the years. The founding students of the program, especially James Kaminski, Robert Tunney, Jeff Spence, Shaked Afik were incredibly welcoming, and set a extremely positive culture in motion. The unsolicited kindness of Brooke Rhead was particularly personally uplifting. It's been a joy to watch the work of Diana Aguilar-Gomez, Maya Lemmon-Kishi, Isabel Serrano, Sandra Hui, Amanda Mok, and Tal Ashuach build more robust formal mentoring structures for the benefit of future students. I am eternally grateful to Kate Chase and Xuan Quach, who make the program possible in the first place and navigating through the otherwise complex and confusing structures of Berkeley a breeze.

Funding from the National Science Foundation, the Berkeley Institute for Data Science, and the UCSF Bakar Computational Health Sciences Institute made my PhD possible and opened countless opportunities possible. I'm especially grateful to Stacey Dorton, Marsha Fenner, Angela Rizk-Jackson, and Elizabeth Brashers for making these institutes run.

Last but not least, I want to thank my family. Without their constant love and support none of this would have been possible. Especially one very very special and important person. She knows who she is.

Chapter 1

Introduction

For the past few centuries microscopy has played an important role in science, engineering, and medicine. Traditionally, microscopes have consisted of a series of lenses and a detector: a human retina, photographic film, or a digital camera sensor. Recent decades have seen the emergence of *computational* microscopy, in which optics, detectors, and algorithms are designed in tandem. In a computational microscope, the detected image (henceforth called the "measurement") is no longer the final product of the imaging system. It is merely the input to an image processing algorithm, which may produce an image that is in some sense "better" than the raw measurement, or it may perform some other type of inference.

The emergence of computational microscopes opens a wide range of new design possibilities. For example, the measurement no longer needs to be human interpretable–the algorithm can take care of creating a human interpretable image instead, or bypass image-formation altogether and simply make a decision or perform some task. This frees the imaging system to be designed according to other physical or computational constraints. Another possibility is the creation of adaptive imaging systems, which use measurements as intermediate feedback in order to optimize parameters of the imaging system to produce a better final image.

The mathematical foundations of computational microscopy in many cases predate the widespread availability of computers. An early development was deconvolution algorithms, which attempt to invert the low-pass filtering operation imparted by an imaging system and form and computationally synthesize sharper images. These date back to the work of Norbert Wiener in the 1940s, and now find widespread adoption in microscopy, where they enable the production of sharper, cleaner images.

More recently, algorithms used in computational imaging systems have been designed based not only on human knowledge (e.g. the convolution imparted by an imaging system) distilled into a series of explicit, rules-based steps (e.g. a deconvolution algorithm), but also learned from data. In general, the former process is time-intensive and requires expert knowledge to design algorithms that perform well.

The latter approach falls under the umbrella of **machine learning**. Pure machine learning approaches remove some of the requirements for expert conception and design of rules-based image processing algorithms. Instead, relevant rules can be inferred from data.

This offers potential advantages when the process of coming up with rules-based algorithms is time-intensive or even impossible. In the past decade, one particular class of machine learning algorithms has come to dominate machine learning: deep neural networks.

1.1 The remarkable rise of neural networks

The power of neural networks stems from their ability to approximate functions. Specifically, a sufficiently large neural network can approximate *any* function [55]. For image processing algorithms, this is especially useful because functions can be complex and difficult to write out analytically.

The origins of neural networks date back to at least the 1950s in research attempting to model properties of human brains [129]. However, their rise to dominance in machine learning occurred only over the last decade. This paradigm shift is usually attributed to three factors: 1) The invention of a general purpose algorithm for training neural networks to approximate a function from training data called "Backpropagation" [133]. 2) A tipping point in the amount of data available in different domains. 3) The availability of computer processors, specifically graphics processing units (GPUs), capable of training and evaluating neural networks quickly.

In imaging, a significant event in this rise was the creation of the ImageNet dataset [26] in 2009, which contained millions of natural images labelled into a number of categories (cats, dogs, airplanes, cars, etc.). An annual competition began to see what algorithms could perform best in classifying images into their respective categories. In 2012, a neural network [72] won this competition for the first time, dramatically surpassing the previous state-of-the-art. This triggered an enormous interest and surge of new research in neural networks. Concurrent with the work described in this dissertation, neural networks continued their expansion into state-of-the-art performance on many tasks, including beating the best human players at the board game Go [145] and learning to predict the 3d structure of proteins from their amino acid sequence [63] far better than any existing algorithm. At the present moment, the end does not appear to be in sight for their impressive performance.

1.2 Data-driven computational microscopy

The work in this dissertation began in 2015, near the beginning of this flurry of interest in and new possibilities provided by neural networks. This opened up a variety of new ways of performing existing computational microscopy techniques and entirely new techniques that could be quickly learned from data.

However, challenges remain within this process. Deep neural networks generally require large amounts of training data, presenting difficulties on problems when such data is difficult or expensive to acquire. They also have a tendency to produce catastrophically incorrect predictions under certain conditions [39]. On scientific problems where images are the final

output, this can present a particularly pernicious "hallucination" problem, in which images are produced which look convincingly realistic, but are unrepresentative of the underlying reality. The final chapters of this dissertation aim to develop new tools that address this and other challenges presented by the widespread availability of flexible learning machines.

1.3 The arc of this dissertation

In an era where algorithms can be part of the imaging system, control imaging systems themselves, produce incredible state-of-the-art solutions, and produce catastrophically bad and misleading solutions with no apparent underlying reason, many new tools and ways of thinking are needed. This dissertation makes contributions in many areas:

Chapter 2 describes a modern data-driven approach to an age-old problem in microscopy: focusing correctly on the sample. This is an example of adaptive microscopy, in which the imaging system is controlled by an algorithm based on feedback from an intermediate measurement.

Chapter 3 takes the same ideas of adaptive control and applies them to a more challenging problem: controlling illumination while imaging a large biological specimen on a two-photon fluorescence microscope.

Adaptive microscopy relies on a large stack of hardware control and image processing code, much like many other modern types of microscopy. Recognizing the bottleneck imposed by researchers developing their own bespoke software solutions, Chapter 4 describes the development of software package for this purpose based on Micro-Manager [28, 29], a popular open source software for controlling a variety of types of microscope hardware.

Another bottleneck for data-driven approaches is the availability of data specific to certain domains. Chapter 5 addresses this need by describing the creation and curation of a large, structured dataset of microscope images.

The flexibility and design possibilities in computational microscopy, particularly when neural networks are involved, strains the assumptions underlying traditional performance metrics and design principles. In Chapter 6, a new framework based in information theory for reasoning about and evaluating computational microscopes is presented to address these needs.

Finally, Chapter 7 describes some possibilities for future research that builds upon this work.

Chapter 2

Learned single-shot autofocus

In this chapter, we present a modern approach to an age-old problem when utilizing microscopes: correctly focusing on the sample. This work also provides an example of a general framework that will be explored more in the next chapter: creating adaptive microscopes that use the data they collect to control themselves automatically.

2.1 Overview

Many biological experiments involve imaging samples in a microscope over long time periods or large spatial scales, making it difficult to keep the sample in focus. For example, when observing a sample over time periods of hours or days, thermal fluctuations can induce focus drift [70]. Or, when scanning and stitching together many fields-of-view (FoV) to form a high-content high-resolution image, a sample that is not sufficiently flat necessitates refocusing at each position [177]. Since it is often experimentally impractical or cumbersome to manually maintain focus, an automatic focusing mechanism is essential.

A variety of solutions have been developed for autofocus. Broadly, these methods can be divided into two classes: hardware-based schemes that attempt to directly measure the distance from the objective lens to the sample [100, 178, 43, 5, 180], and software-based methods that take one or more out-of-focus images and use them to determine the optimal focal position [143, 174, 83, 82]. The former usually require hardware modifications to the microscope (e.g. an infrared laser interferometry setup, additional cameras or optical elements), which can be expensive and place constraints on other aspects of the imaging system. Software-based methods, on the other hand, can be slow or inaccurate. For example, a software-based method might require a full focal stack, then use some measure of image sharpness to compute the ideal focal plane [143]. More advanced methods attempt to reduce the number of images needed to compute the correct focus [174], or use just a single out-of-focus image [83, 82]. However, existing single-shot autofocus methods either rely on nontrivial hardware modifications such additional lenses and sensors [82] or are limited in their application to specialized regimes (i.e. can only correct defocus in one direction within

a certain range) [83].

Here, we demonstrate a new computational imaging-based single-shot autofocus method that does not suffer from the limitations of previous methods. The only hardware modification it requires is the addition of one or more off-axis LEDs as an illumination source, from which we correct defocus based on a single out-of-focus image. Alternately, it can be used with no hardware modification on existing coded-illumination setups, which have been demonstrated for super-resolution [182, 103, 154], quantitative phase [182, 155, 103], and multi-contrast microscopy [181, 86].

The central idea of our method is that a neural network can be trained to predict how far out of focus a microscope is, based on a single image taken at arbitrary defocus under spatially coherent illumination. A related idea has recently been used to achieve fast, post-experimental digital refocusing in digital holography [171, 127]. Our work addresses autofocusing in more general microscope systems, with both incoherent and coherent illumination. Intuitively, we believe this works because coherent illumination yields images with sharp features even when the sample is out of focus. Thus, there is sufficient information in the out-of-focus image that an appropriate neural network can learn a function that maps these features to the correct defocus distance, regardless of the structural details of the sample. To test this idea we collected data using a Zeiss Axio Observer microscope (20×, 0.5 NA) with the illumination source replaced by a programmable quasi-dome LED array [111]. The LED array provides a flexible means of source patterning, but is not necessary to implement this technique (see Sec 2.2).

Though our experimental focus prediction requires only one image, we do need to collect focal stacks for training and validation. We use Micro-Magellan [119] for software control of the microscope, collecting focal stacks over 60 μm with 1 μm spacing, distributed symmetrically around the true focal plane. For each part of the sample, we collect focal stacks with two different types of illumination: spatially coherent (i.e. a single LED) and (nearly) spatially incoherent (i.e. many LEDs at once).

The incoherent focal stack is used for computing the ground truth focal position, since the reduced coherence results in sharp images only when the sample is in focus. Sharpness can be quantified for each image in the stack by summing the high-frequency content of its radially averaged log power spectrum. The maximum of the resultant curve was chosen as the ground truth focal position for the stack (Fig. 2.1a, left). Because this ground truth value is calculated by a deterministic algorithm, this paradigm scales well to large amounts of training data. For transparent samples, the incoherent image stack was captured with asymmetric illumination in order to create phase contrast [92]. In our case, this was achieved by using the LED array to project a half annulus source pattern [155]; however, any asymmetric source pattern should suffice.

The coherent focal stack is used one image at a time as the input to the network, which is trained to predict the ground truth focal position (Fig. 2.1). Since the network only takes a single image as its input, each image in the stack represents a separate training example. In our case, the coherent focal stack was captured by illuminating the sample with a single off-axis LED. In the case of arbitrary illumination control (e.g. with an LED array) different

illumination angles or patterns may perform differently for a given amount of training data. Supplementary Fig. 2.5 compares performance for varying single-LED illumination angles as well as multi-LED patterns. For simplicity, here we consider only the case of a single LED positioned at an angle of 24 degrees relative the optical axis.

Our neural network architecture for predicting defocus (described in detail in Sec. 2.4), which we call the fully connected Fourier neural network (FCFNN), differs substantially from the convolutional neural networks (CNNs) typically used in image processing tasks [127, 171, 173] (Sec. 2.5). We reasoned that singly-scattered light would contain the most useful information for defocus prediction, and thus we designed the FCFNN to exclude parts of the captured image's Fourier transform that are outside the single-scattering region for off-axis illumination (Fig. 2.6). This results in 2-3 orders of magnitude fewer free parameters and memory usage during training than state-of-the-art CNNs (Table 2.1). Hence, our network can be trained on a desktop CPU in a few hours with no specialized computing hardware, making our method more reproducible, without sacrificing quality.

Briefly, the FCFNN (Fig. 2.1a, right) begins with a single coherent image. This image is Fourier transformed, and the magnitude of the complex-valued pixels in the central part of the Fourier transform are reshaped into a single vector, which is used as the input to a trainable fully connected neural network. After the network has been trained, it can be used to correct defocus during an experiment by capturing a single image at an arbitrary defocus under the same coherent illumination. The network predicts defocus distance, then the microscope moves to the correct focal position (Fig. 2.1b).

Training with 440 focal stacks took 1.5 hours on a desktop CPU or 30 minutes on a GeForce GTX 1080 Ti GPU, in addition to 2 minutes per focal stack for pre-computing ground truth focal planes and Fourier transforms. A single prediction from a 2048x2048 image takes $\sim$50 ms on a desktop CPU. We were able to train FCFNNs capable of predicting defocus with root-mean-squared error (RMSE) smaller than the axial thickness of the sample (cells). Figure 2.2 shows how this performance varies based on the number of focal stacks used to train the network, where each focal stack contained 60 planes spaced 1 μm apart, distributed symmetrically around the true focal plane. Note that this curve could be quite different depending on the sample type and quality of training data.

To test the performance of our method across different samples, we collected data from two different sample types (Fig. 2.3a): white blood cells attached to coverglass, and an unstained 5 μm thick mounted histology tissue section. When the network is *trained* on images of cells, then *tested* on different images of cells, it performs very well (Fig. 2.3b). However, when the network is trained on images of cells, then tested on a different sample type (tissue), it performs poorly (Fig. 2.3c). Hence, the method does not inherently generalize to new sample types. To solve this problem, we diversify the training data. We add a smaller amount of additional training data from the new sample type (here 130 focal stacks of tissue data, in addition to the 440 stacks of cell data it was originally trained on). With this training, the network performs well on both tissue and cell samples. Hence, our method can generalize to other sample types, without sacrificing performance on the original sample type (Fig. 2.3d). The best performing neural networks in other domains are typically trained

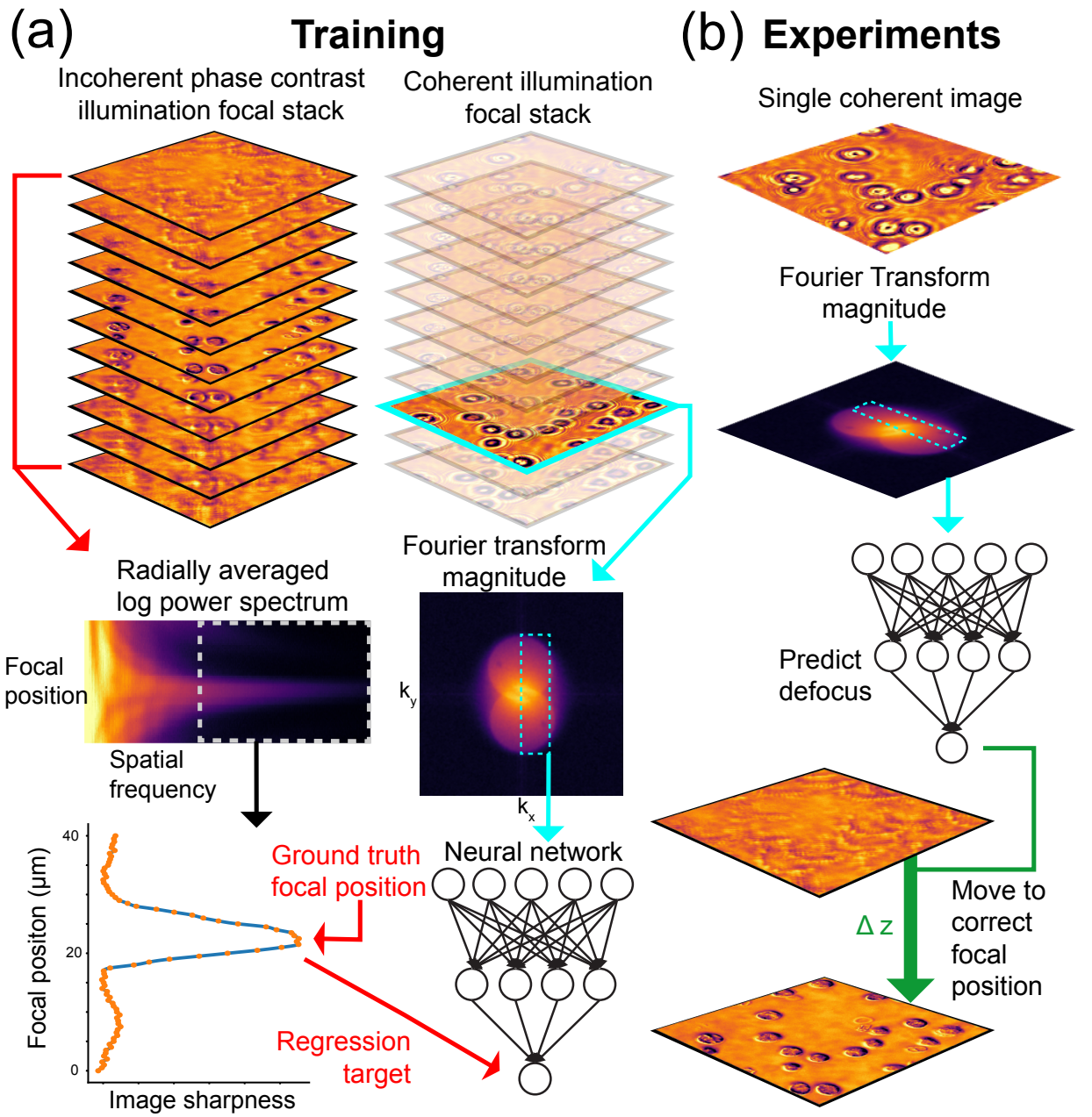

Figure 2.1: **Training and defocus prediction.** a) Training data consists of two focal stacks for each part of the sample, one with incoherent (phase contrast) illumination, and one with off-axis coherent illumination. Left: The high spatial frequency part of each image's power spectrum from the incoherent stack is used to compute a ground truth focal position. Right: For each coherent image in the stack, the central pixels from the magnitude of its Fourier transform are used as input to a neural network that is trained to predict defocus. The full set of training examples is generated by repeating this process for each of the coherent images in the stack. b) After training, experiments need only collect a single coherent image, which is fed through the same pipeline to predict defocus. The microscope's focus can then be adjusted to correct defocus.

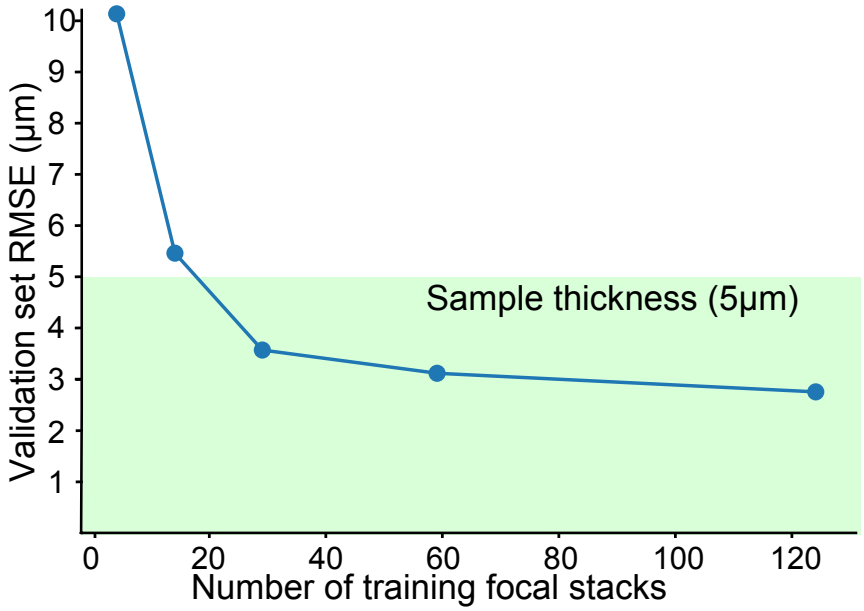

Figure 2.2: **Performance vs. amount of training data.** Defocus prediction performance (measured by validation RMSE) improves as a function of the number of focal stacks used during the training phase of the method.

on large and varied datasets [72]. Thus, if the FCFNN is trained on defocus data from a variety of sample types, it should generalize to new types more easily.

Empirically, we discovered that discarding the phase of the Fourier transform and using only the magnitude as the input to the network dramatically boosted performance. To illustrate, Fig. 2.4a compares networks trained using the Fourier transform magnitude as input vs. those trained on the argument of the Fourier transform phase. Not only were networks using magnitude able to better fit the training data, they also generalized better to a validation set. This suggests useful information for predicting defocus in a coherent intensity image is relatively more concentrated in the magnitude compared to the phase of its Fourier transform. We speculate that this is because the phase of the intensity image generally relates more to spatial position of features (which is unimportant for focus prediction), whereas the magnitude contains more information about how they are transformed by the imaging system.

In order to understand what features of the images the network learns to make predictions from, we compute a saliency map for a network trained using the entire uncropped Fourier transform, shown in Fig. 2.4b. The saliency map attempts to identify which parts of the input the network is using to make decisions, by visualizing the gradient of a single unit within the neural network with respect to the input [146]. The idea is that the output unit is more sensitive to features with a large gradient and thus these have a greater influence

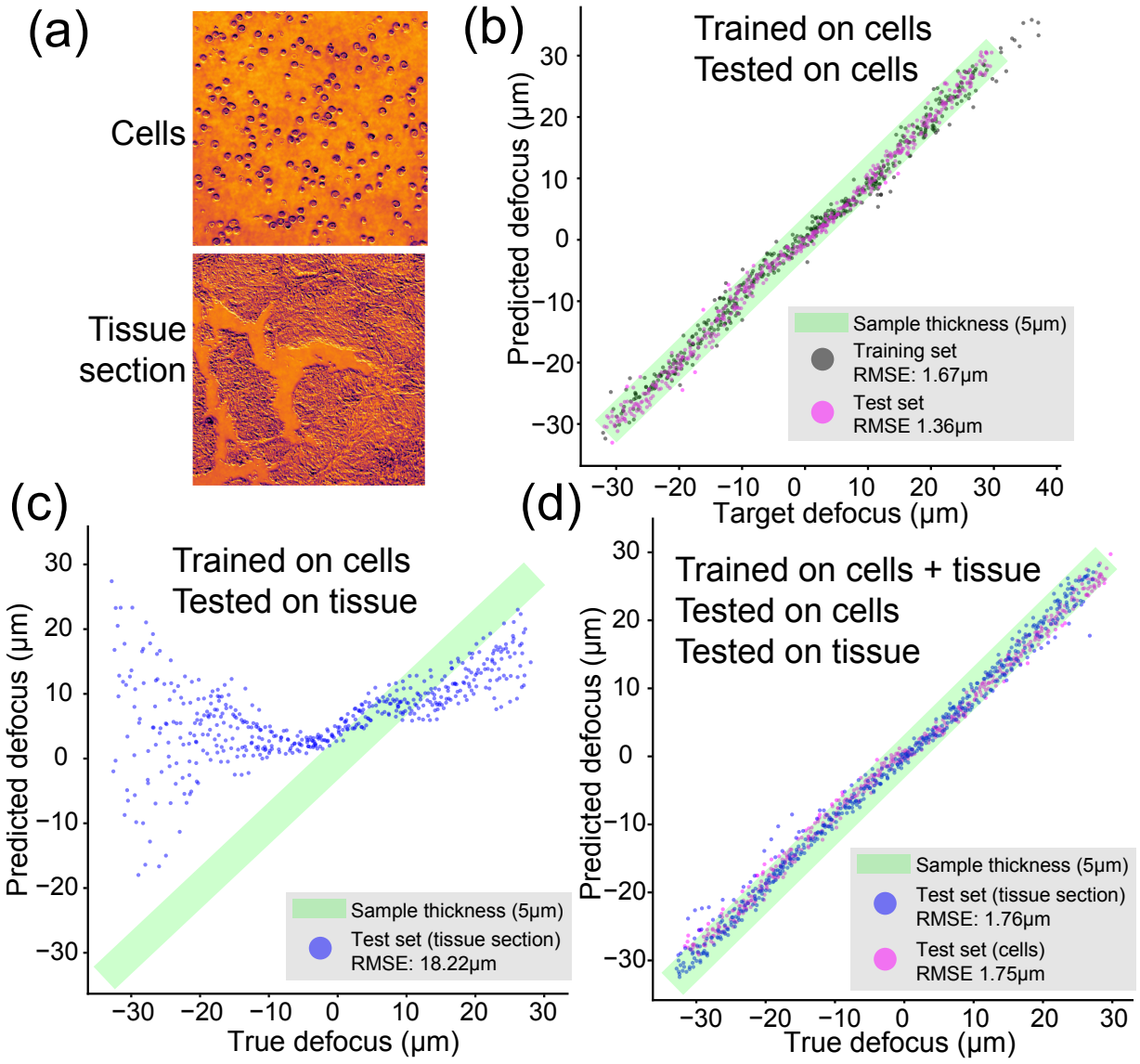

Figure 2.3: **Generalization to new sample types.** a) Representative images of cells and tissue section samples. b) A network trained on focal stacks of cells predicts defocus well in other cell samples, c) but fails at predicting defocus in tissue sections. d) After adding limited additional training data on tissue section samples, however, the network can learn to predict defocus well in both sample types.

on prediction. In our case, the gradient of the output (i.e. the defocus prediction) was computed with respect to the the Fourier transform magnitude. Averaging the magnitude of the gradient image over many examples clearly shows that the network recognizes specific parts of the the overlapping two-circle structure (Fig. 2.4b) that is typical for an image formed by coherent off-axis illumination (Fig. 2.6) [27]. In particular, the regions at the edges of the circles have an especially large gradient. These areas correspond to the highest angles of light collected by the objective lens. Intuitively, this makes sense because changing the focus will lead to proportionally greater changes in the light collected at the highest angles (Fig. 2.4b).

To summarize, we have demonstrated a method for training and using neural networks for single-shot autofocus, with analysis of design principles and practical trade-offs. The method works with different sample types and is simple to implement on a conventional transmitted light microscope, requiring only the addition of off-axis illumination and no specialized hardware for training the neural network. We introduced the FCFNN, a neural network architecture that incorporates knowledge of the physics of the imaging system into its design, thereby making it orders of magnitude more efficient in terms of parameter number and memory requirements during training than general state-of-the-art approaches for image processing.

The code needed to implement this technique and reproduce all figures in this manuscript can be found in the Jupyter notebook: 1. H. Pinkard, "Single-shot autofocus microscopy using deep learning–code," (2019), `https://doi.org/10.6084/m9.figshare.7453436.v1`. Due to its large size, the corresponding data is available upon request.

2.2 Practical aspects of implementing on a new microscope

Hardware/Illumination In order to generate data using our method, the microscope must be able to image samples with two different contrast modalities: one with spatially incoherent illumination for computing ground truth focal position from image blur, and a second coherent or nearly coherent illumination (i.e. one or a few LEDs) as input to the neural network. The incoherent illumination can be accomplished with the regular brightfield illumination of a transmitted light microscope in the case of samples that absorb light. In the case of transparent phase-only samples (like the ones used in our experiments), incoherent phase contrast can be created by using any asymmetric illumination pattern. We achieved this by using a half-annulus pattern on a programmable LED illuminator, but this specific pattern is not necessary. The same effect can be achieved by blocking out half of the illumination aperture of a microscope condenser with a piece of cardboard[92] or other means of achieving asymmetric illumination. The asymmetric incoherent illumination is only needed for the generation of training data, so it does not need to be permanent.

For the spatially coherent illumination, a single LED pointed at the sample from an

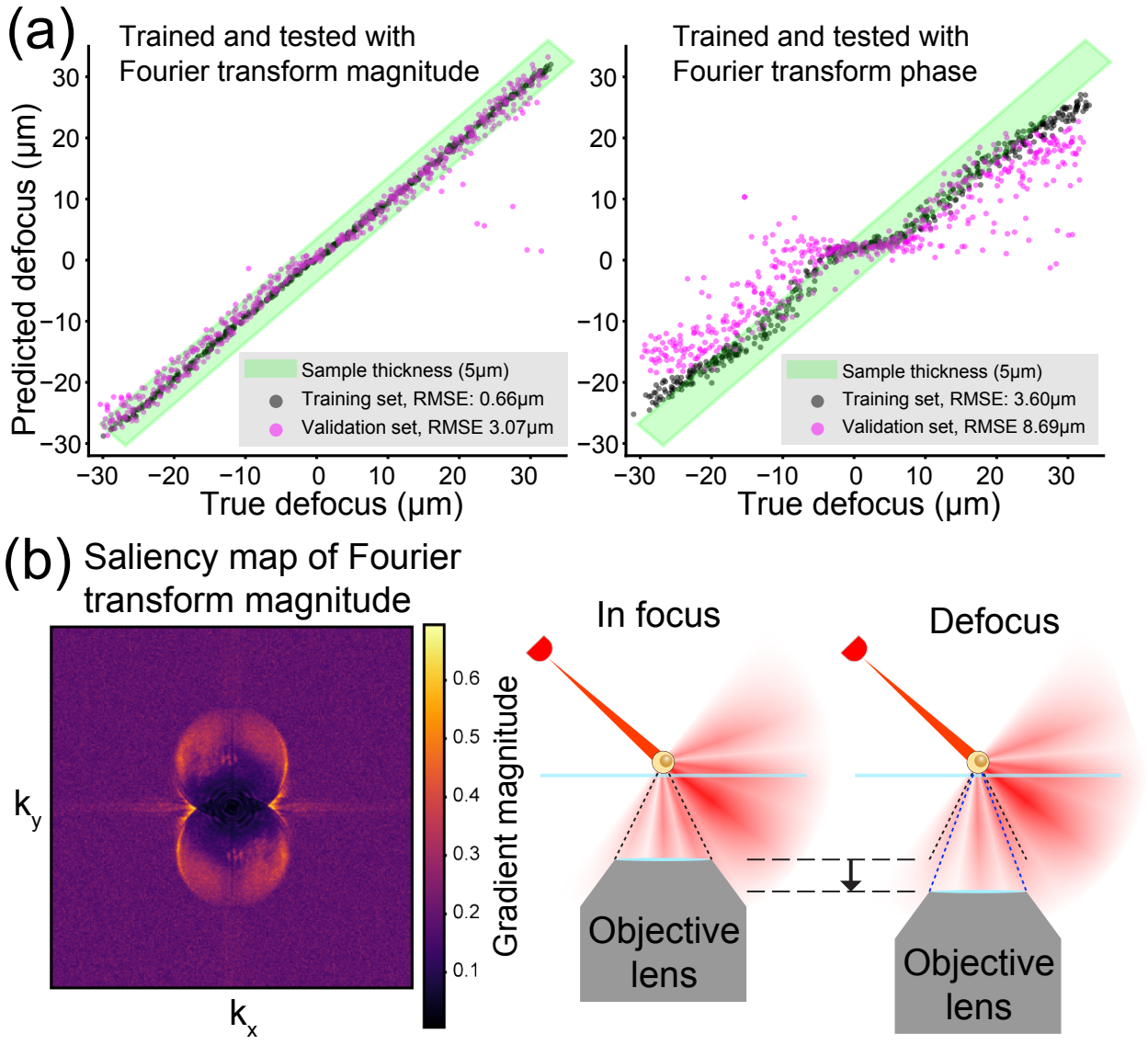

Figure 2.4: **Understanding how the network predicts defocus.** a) A network trained on the magnitude of the Fourier transform of the input image performs better than one trained on the argument of the phase of the Fourier transform. b) Left: a saliency map (the magnitude of the defocus prediction's gradient with respect to the Fourier transform magnitude) shows the edges of the object spectrum have the strongest influence on defocus predictions. Right: edges correspond to high-angle scattered light, which may not be captured off-focus, providing significant changes in the input image with defocus.

oblique angle (i.e. not directly above) generates sufficient contrast. However, our experiments with different multi-LED patterns (see Sec. 2.3) indicate that a series of LEDs arranged in a line might be even better for this purpose.

Software Our implementation used a stack of open source acquisition control software based on Micro-Manager [28] and the plugin for high-throughput microscopy, Micro-Magellan [119]. Both are agnostic to specific hardware, and can thus be implemented on any microscope to easily collect training data. Automated LED illumination in Micro-Manager can be configured using a simple circuit connected to an Arduino and the Micro-Manager device adapter to control digital IO (`https://micro-manager.org/wiki/Arduino#Digital_IO`). Large numbers of focal stacks can be collected in an automated way using the 3D imaging capabilities of Micro-Magellan, and a Python reader for the NDTiff files created [120] allows for easy integration of data into deep learning frameworks. Examples of this can be seen in the Jupyter notebook: 1. H. Pinkard, "Single-shot autofocus microscopy using deep learning–code," (2019), `https://doi.org/10.6084/m9.figshare.7453436.v1`.

Other imaging geometries Although we have demonstrated this technique on a transmitted light microscope with LED illumination, in theory there is no reason why it couldn't be applied to other coherent illuminations and geometries. For example, using a laser instead of an LED as a coherent illumination source should be possible with minimal modification. We've also demonstrated the technique using relatively thin samples. Autofocusing methods like ours are generally not directly applicable to thick samples, since it is difficult to define the ground truth focal plane of a thick sample in a transmitted light configuration. However, in principle it is possible that these methods could be used in a reflected light geometry, where the "true" focal plane corresponds to the top of the sample.

2.3 Choosing an illumination pattern

Although the network is capable of learning to predict defocus from images taken under the illumination of a single off-axis LED, different angles or combinations of angles of illumination might contain more useful information for prediction. Better performance can make the prediction task more accurate, easier and able to be learned with less training data. Since our experimental setup uses a programmable LED array quasi-dome as an illumination source [111], we can choose the source patterns at will to test this. First, restricting the analysis to one LED at a time, we tested how the angle of the single-LED illumination affects performance (Fig. 2.5a). We found that performance improves with increasing angle of illumination, up to a point where performance rapidly degrades. This drop-off occurs in the 'darkfield' region (where the illumination angle is larger than the objective's NA), likely due to the low signal-to-noise ratio (SNR) of the higher-angle darkfield images (see inset images in Fig. 2.5a). This drop in SNR could plausibly be caused by either a decrease in the number of photons hitting the sample from higher-angle LEDs, or a drop in the content of

the sample itself at higher frequencies. To rule out the first possibility, we compensated for the expected number of photons incident on a unit area of the sample, which is expected to fall off approximately proportional to $\frac{1}{cos(\theta)}$, where θ is the angle of illumination relative to the optical axis [112]. The dataset used here increases exposure time in proportion to $cos(\theta)$ in order to compensate for this. Thus, the degradation of performance at high angles is most likely due to the amount of high frequency content in the sample itself at these angles and therefore might be sample-specific.

Next, we tested 18 different single or multi-LED source patterns chosen from within the distribution of x and y axis-aligned LEDs available on our quasi-dome (Fig. 2.5b,c). Since the light from any two LEDs is mutually incoherent, single-LED images can be added digitally to synthesize the image that would have been produced with multiple-LED illumination. This enabled us to computationally experiment with different illumination types on the same sample. Figure 2.5c shows the defocus prediction performance of various patterns of illumination. The best performing patterns were those that contained multiple LEDs arranged in a line. Given that specific parts of the Fourier transform contain important information for defocus prediction and that these areas will move to different parts of Fourier space with different angles of illumination, we speculate that the line of LEDs helps to spread relevant information for defocus prediction into different parts of the spectrum. Although this analysis demonstrates more and higher angle LED patterns seem to yield superior performance, there are potential caveats: In the former case, it could fail to hold when applied to a denser sample (i.e. not a sparse distribution of cells). In the latter, there is the cost of the increase in exposure time needed to acquire such images.

2.4 Fully connected Fourier neural network architecture

The fully connected Fourier neural network (FCFNN) begins with a single coherent intensity image captured by the microscope. This image is Fourier transformed, and the magnitude of the complex-valued pixels in the central part of the Fourier transform are reshaped into a single vector. The useful part of the Fourier transform is directly related to the angle of coherent illumination (Fig. 2.6). A coherent illumination source such as an LED that is within the range of brightfield angles for the given objective (i.e. at an angle less than the maximum captured angle as determined the objective's NA) will display a characteristic 2-circle structure in its Fourier transform magnitude. The two circles contain information corresponding to the singly-scattered light from the sample and move farther apart as the angle of the illumination increases. The neural network input should consist of half of the pixels in which these circles lie, because as the saliency map in Fig. 4b of the main text demonstrates, they contain the useful information for predicting defocus. Only half the pixels are needed because the Fourier transform of a real-valued input (i.e. an intensity image) has symmetric magnitudes, so the other half contain redundant information. These circles move

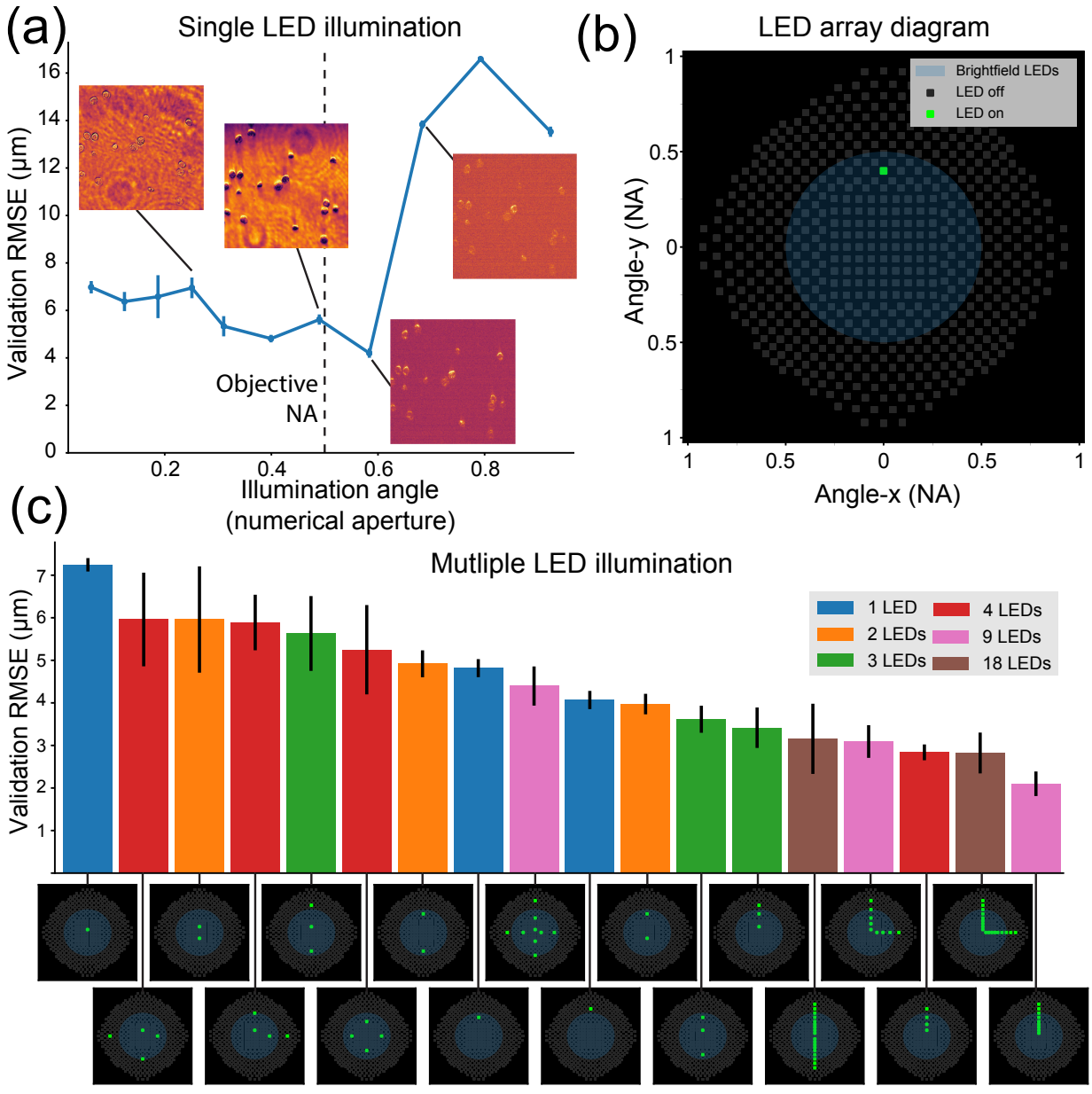

Figure 2.5: **Illumination design.** a) Increasing the numerical aperture (NA) (i.e. angle relative to the optical axis) of single-LED illumination increases the accuracy of defocus predictions, up to a point at which it degrades. b) Diagram of LED placements in NA space for our LED quasi-dome. c) Defocus prediction performance for different illumination patterns. Patterns with multiple LEDs in an asymmetric line show the lowest error.

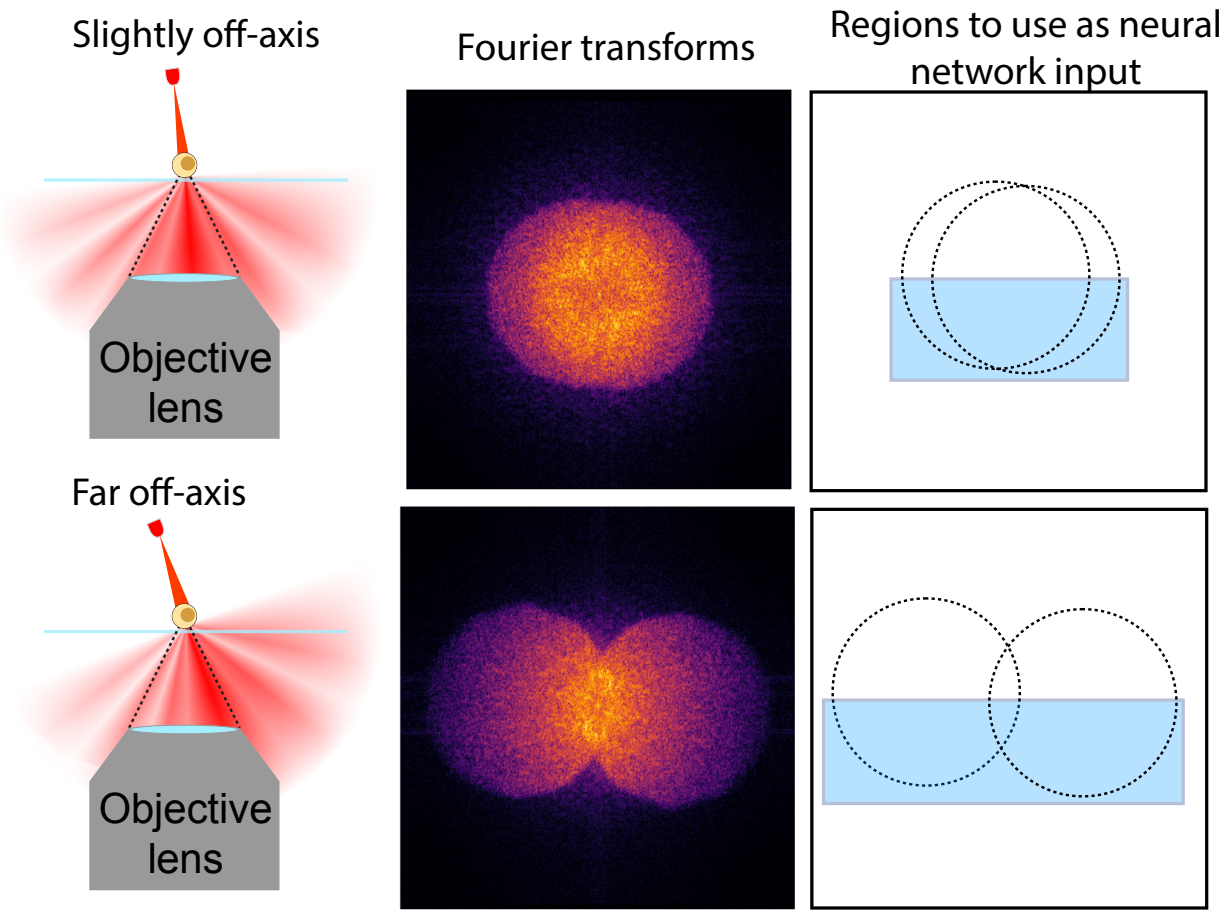

Figure 2.6: **Fourier Transform regions to use as network input.** Off-axis illumination with a coherent point source at an angle within the numerical aperture of the collection objective produces a characteristic two-circle structure in the log magnitude of the Fourier transform of the captured image. As the angle of illumination increases, these circles move further apart. Information about single-scattering events is confined within these circles. The blue regions represent the pixels that should be cropped out and fed into the neural network architecture.

with changing illumination angle, so they angle of illumination and relevant pixels must be selected together.

After cropping out the relevant pixels and reshaping them into a vector, the vector is normalized to have unit mean in order to account for differences in illumination brightness, and it is then used as the input layer of a neural network trained in TensorFlow [1]. The learnable part of the FCFNN consists of a series of small (100 unit) fully connected layers, followed by a single scalar output (the defocus prediction).

We experimented with several hyperparameters and regularization methods to improve performance on our training data. The most successful of these were: 1) Changing the number and width of the fully connected layers. We started small and increased both until this ceased to improve performance, which occurred with 10 fully connected layers of 100 units each. 2) Applying dropout [148] to the vectorized Fourier transform input layer (but not other layers) to prevent overfitting to specific parts of the Fourier transform. 3) Dividing the input image into patches and averaging the predictions over each patch. This gave best performance when we divided the 2048x2048 image into 1024x1024 patches. 4) Using only the central part of the Fourier transform magnitude as an input vector. We manually tested how much of the edges to crop out. 5) Early stopping - when loss on a held out validation set ceased to improve - helped test performance.

In general, we observed better performance training on noisier and more varied inputs (i.e. cells at different densities, particularly lower densities, and different exposure times). This is consistent with other results in deep learning, where adding noise to training data improves performance [162].

Table 2.1: Comparison of number of learnable weights and memory usage

Architecture	Image size	# of learnable weights	memory per training example (MB or KB)
FCFNN (ours)	1024x1024	$6.3x10^6$	18 KB
CNN (Ren et. al[127])	1000x1000	$2.5x10^8$	111 MB
FCFNN (ours)	1000x1000	$6.0x10^6$	17 KB
CNN (Yang et. al [173])	84x84	$2.9x10^7$	1.3 MB
FCFNN (ours)	84x84	$3.6x10^4$	3 KB

2.5 Comparison of FCFNNs and CNNs

The FCFNN differs substantially from the convolutional neural networks (CNNs) used as the state-of-the-art in image processing tasks. Typically, to solve a many-to-one regression task of predicting a scalar from an image, as in the defocus prediction problem here, CNNs first use a series of convolutional blocks with learnable weights to learn to extract relevant features from the image and then often will pass those features through a series of fully connected layers to generate a scalar prediction [72]. Here, we have replaced the feature learning part of the network with a deterministic feature extraction module that uses only the physically-relevant parts of the Fourier Transform.

Deterministically downsampling images into feature vectors early in the network reduces the required number of learnable weights and the memory used by the backpropagation algorithm to compute gradients during training by 2-3 orders of magnitude. Table 2.1 shows a comparison between our FCFNN and two CNNs used for comparable tasks. The architecture used by Ren et al. is used for post-acquisition defocus predicition in digital holograpy and the architecture of Yang et al. is used for post-acquisition classification of images as in-focus or out-of-focus. Both use the conventional CNN paradigm of a series of convolutional blocks followed by one or more fully connected layers.

Similar to CNNs, our FCFNN can also incorporate information from different parts of the full image. CNNs do this with a series of convolutional blocks that gradually expand the size of the receptive fields. The FCFNN does this inherently by use of the Fourier transform. Each pixel in the Fourier transform corresponds to a sinusoid of a certain frequency and orientation in the original image, so its magnitude draws information from every pixel.

Chapter 3

Learned adaptive multiphoton illumination

In the previous chapter an adaptive microscopy technique was presented in which images are fed into a neural network and the output of that network is used to control the microscope by correctly focusing onto the sample. In this chapter, the same paradigm is applied to a different modality: *in vivo* multiphoton microscopy. Instead of focus, the parameter controlled by the network's output is excitation laser power.

3.1 Introduction

Imaging of cells *in vivo* is an essential tool for understanding the spatiotemporal dynamics that drive biological processes. For highly scattering tissues, multiphoton microscopy is unique in its ability to image deep into intact samples (200 μm - 2 mm, depending on the tissue). Because of the nonlinear relationship between excitation light power and fluorescence emission, scattered excitation light contributes negligibly to the detected fluorescence emission. Thus, localized fluorescent points can be imaged deep in a sample in spite of a large fraction of the excitation light scattering away from the focal point, by simply increasing the incident excitation power [51] (**Fig. 3.1a**).

The dual problems photobleaching and photodamage are an inescapable part of every fluorescence imaging experiment. The concept of a "photon budget" is often used to express the inherent trade-offs between sample health, signal, spatial resolution, and temporal resolution, and a widely pursued goal is to make microscopes that are as gentle as possible on sample while still generating the contrast necessary for biological discovery [137]. These problems are an especially acute concern in multiphoton microscopy since, unlike in single-photon fluorescence, they increase supra-linearly with respect to the intensity of fluorescence emission [19, 106].

When imaging deep into a sample using multiphoton microscopy, excitation light focusing to different points in the sample will be subjected to different amounts of scattering, and

the excitation laser power must be increased in order to maintain signal. Failing to increase sufficiently will lead to the loss of detectable fluorescence. Increasing too much subjects the sample to unnecessary photobleaching and photodamage, with the potential to disrupt or alter the biological processes under investigation. If done improperly, this can even result in visible burning or destruction of the sample. This problem is especially pronounced in highly scattering tissue (e.g. in lymph nodes) because the appropriate excitation power has more rapid spatial variation compared to less scattering tissues.

Adaptive optics (AO) represents one strategy for addressing this challenge [61, 152]. By pre-compensating the shape of the incident excitation light wavefront based on the scattering properties of the tissue, the fraction of incident light that reaches the focal point increases, lessening the need to increase power with depth. However, AO still suffers from an exponential decay of fluorescence intensity with imaging depth when using constant excitation [51, 152], so an increase in incident power with depth is still necessary.

Alternatively, instead of minimizing scattering with AO, adaptive illumination techniques modulate excitation light intensity to ensure the correct amount reaches the focus. To make the best use of a sample's photon budget, these methods should increase power to the minimal level needed to yield sufficient contrast, but no further than this to avoid the effects of photobleaching and photodamage.

Most commercial and custom built multiphoton microscopes have some capability to increase laser power with depth, either using an exponential profile or an arbitrary function. For a flat sample (e.g. imaging into brain tissue through a cranial window), these techniques work well. The profile of fluorescence decay with depth can be approximated by an exponential or heuristically defined for an arbitrary function, by focusing to different depths in a sample and manually specifying increases. However, this task is more complex for a curved or amorphous sample, in which such profiles shift as the height of the sample varies and change shape in different areas of the sample.

A more advanced class of of methods for adapting illumination uses feedback from the sample during imaging. This strategy has been employed previously in both confocal [54] and multiphoton [21, 81] microscopy. The basic principle is to implement a feedback circuit between the microscope's detector and excitation modulation, such that excitation light power is turned off at each pixel once a sufficient number of photons have been detected. However, this approach doesn't account for fluorophore brightness and labelling density; thus, it is impossible to disambiguate weak fluorophores (e.g. a weakly expressed fluorescent protein) receiving a high dose of incident power from strong fluorophores (e.g. a highly expressed fluorescent protein) receiving a low dose. Not only does this run the risk of unnecessarily depleting the photon budget, it can also lead to over-illumination and photodamage if left unchecked. To prevent photodamage, a heuristic user-specified upper bound is set to cap the maximum power. Such an upper bound can vary by over an order-of-magnitude when imaging into highly-scattering thick samples. Thus, applying this approach to image 100s of μms deep in such samples still requires additional prior knowledge about the attenuation of fluorescence in different parts of a sample.

The difficulties of adaptive illumination in non-flat samples thus creates several prob-

lems. First, the range over which sufficient contrast can be generated is limited to the sub-region where an appropriate function to modulate power can be ascertained and applied by the hardware. Second, incorrect modulations can deplete the photon budget and cause unnecessary photodamage, with unknown effects on the processes under observation.

In intravital imaging of the popliteal lymph node, an important model system for studying vaccine responses, the constraint on imaging volumes imparts an unfortunate bias. Previous studies of T cell dynamics in intact lymph nodes have increased the density of transferred monoclonal T cells in order to achieve sufficient numbers for visualization (10^6 or more) within the limited imaging volume of multiphoton microscopy. This number is 2-3 orders-of-magnitude more than the number of reported clonal precursor T cells ($10^3 - 10^4$) under physiological conditions [125, 9]. It is well-established that altering precursor frequencies changes the kinetics and outcome of the immune responses [90, 37, 4, 50], but it is unknown how these alterations might have affected the conclusions of previous studies.

Here, we describe a data-driven technique for learning the appropriate excitation power as a function of the sample shape, and provide a simple hardware modification to an multiphoton microscope that enables its application. Our method can provide 10-100$\times$ increase in the volume to which appropriate illumination power can be applied in curved samples such as lymph nodes, and a reproducible way to automatically apply the minimal illumination needed to observe structures of interest, thereby conserving the photon budget and minimizing the perturbation to the sample induced by the imaging process. Significantly, our method neither requires the use of additional fluorescence photons to perform calibration on each sample, nor specialized sample preparation to introduce fiducial markers.

The method uses a one-time calibration experiment to learn the parameters of a physics-based machine learning model that captures the relationship between fluorescence intensity and incident excitation power in a standardized sample, given the sample's shape. On subsequent experiments, this enables continuous adaptive modulation of incident excitation light power as a focal spot is scanned through each point in the sample. We describe a simple hardware modification to an existing multiphoton microscope that enables modulation of laser power as the excitation light is scanned throughout the sample. This modification costs <\$50 for systems that already have an electro-optical or accousto-optic modulator, as most modern multiphoton systems do. We call our technique learned adaptive multiphoton illumination (LAMI).

Our central insight is inspired by the idea of "standard candles" in astronomy [110], where the fact that an object's brightness is known *a priori* allows its distance to Earth to be inferred based on its apparent brightness. Analogously, we hypothesize that by measuring the fluorescence emission of identically-labelled cells ("standard candles") at different points in a sample volume under different illumination conditions, we could use a physics-based neural network to learn an appropriate adaptive illumination function that could predicted from sample shape alone.

Applying LAMI to intravital imaging of the mouse lymph node, we first show that the learned function generalizes across differently-shaped samples of the same tissue type (e.g. one mouse lymph node to another). Moving to a new tissue type, which would attenuate

light differently, would require a new calibration experiment. After a one-time calibration experiment, the trained neural network can be used to automatically modulate excitation power to the appropriate level at each point in new samples, enabling dynamic imaging of the immune system with single-cell resolution across volumes of tissue more than an order-of-magnitude larger than previously described. Unlike previous studies which artificially increased the number of monoclonal precursor T cells to $> 10^6$ (2 orders-of-magnitude greater than typical physiological conditions) in order to visualize them in a small imaging volume [93, 13], we image physiologically realistic (5×10^4 transferred) cell frequencies.

3.2 Results

Learning illumination power as a function of shape The detected fluorescence intensity at a given point results from a combination of two factors: 1) The sample-dependent physics of light propagation (e.g. scattering potential of the tissue, fraction of emitted photons that are detected, etc.), which are difficult to model *a priori* due to heterogeneity in sample shapes. 2) The fluorescent labelling (e.g. the type and local concentration of fluorophores), a nuisance factor that makes it difficult to disambiguate weak fluorophores receiving a high dose of incident power from strong fluorophores receiving a low dose.

Our method relies on the fact that, if fluorescence labelling of distinct parts of the sample are, on average, constant (i.e. "standard candles"), we can separate out the effects of fluorescence strength and tissue-dependent physics by performing a one-time calibration to learn the effect of *only* the tissue-dependent physics for a given tissue type. The calibrated model captures the effects of the physics relating excitation power, detected fluorescence, local sample curvature, and position in the XY field-of-view, which includes optical vignetting effects. By generating a dataset consisting of points with random distributions over these variables, we can learn the parameters of a statistical model to predict excitation power as a function of detected fluorescence, sample shape, and position. On subsequent experiments in different samples of the same type, the model can predict the excitation power required to achieve a desired level of detected fluorescence for each point in the sample based only on sample shape and XY position.

The standard candle fluorophores are only necessary during the calibration step. In the mouse lymph node, we introduce them by transferring genetically identical, identically-labelled (with either cytosolic fluorescent protein or dye) lymphocytes, which then migrate into lymph nodes and position themselves throughout its volume. Although there are certainly stochastic differences in labeling density between individual cells (e.g. noise in expression of fluorescent proteins), the neural network estimates the population mean, so as long as these differences are not correlated with the cells' spatial locations, they will not bias the calibration.

An important consideration is what type of statistical model will be used to predict excitation power. One possibility is a purely physics-based model. We developed such a model using principles of ray optics by computing the length each ray travels through

the sample and its probability of scattering before reaching the focal point (**Fig. 3.11**). When one must predict excitation in real time, however, this model is too computationally intensive (∼1 s per focal point). To circumvent the problem, the model parameters can be pre-computed, but this requires the assumption of an unrealistic, simplified sample shape, thus introducing a sample-dependent source of model mismatch. On top of this, there may be additional sources of model-mismatch, such as a failure to account for wave-optical effects, inhomogeneous illumination across the field-of-view, spatial variation in attenuation of fluorescence signal, etc.

Given this model-mismatch, we found that a physics-based neural network was a better solution. Unlike the purely physics-based model, a physics-based neural network is a flexible function approximator that can be easily adapted to incorporate additional relevant physical quantities into its predictions. For example, accounting for variations in brightness across a single field-of-view would require building optical vignetting effects into a physical model, whereas a neural network can simply take position in the field-of-view as an input and learn to compensate for these effects. Importantly, a small neural network can make predictions quickly (∼1 ms per focal point) and is thus suitable for real-time application.

The neural network makes its predictions based on measurements of the sample shape that capture important parameters of the physics of fluorescence attenuation. To measure these parameters, points were hand selected on the sample surface in XY images of a focal stack to generate a set of 3D points representing the outer shape of the sample (**Fig. 3.1b**). These points were interpolated in 3D in a piece-wise linear fashion to create a 3D mesh of the sample surface. In multiphoton microscopy, the distance light travels through tissue is an important quantity, as both the fraction of excitation light that attenuates from scattering/absorption and the fraction of fluorescence emission that absorbs are proportional to the negative exponential of this distance [51], assuming homogeneous scattering. We thus reasoned that measuring the full distribution of path lengths (i.e. every ray within the objective's numerical aperture - the same starting point of the ray optics model), would provide an informative parameterization to predict fluorescence attenuation. Empirically, we found that the full distribution of distances was not needed to achieve optimal predictive performance (based on error on a held out set of validation data during neural network training), and that measuring 12 distances along lines with a single angle of inclination relative to the optical axis was sufficient (**Fig. 3.1c**, green box). We encode the assumption that the optical system is rotationally symmetric about the optical axis by binning the measured distances into a histogram. The counts of this histogram were used in the feature vector fed into the neural network.

The neural network takes inputs of mean standard candle brightness, local sample shape, and position within the XY field-of-view and outputs a predicted excitation power (Fig 1c, orange box). The network is trained using a dataset with a single standard candle cell that was imaged with a random, known amount of excitation power. Neural networks are excellent interpolators and poor extrapolators, so we ensured that the random excitation power used in training induced a range of brightnesses spanning too-dim-to-see to unnecessarily bright (**Fig. 3.1c**, top middle. Unlike contemporary deep neural networks [75], the prediction only

requires a very small network with a single hidden layer (a $10^4 - 10^6$ reduction in number of parameters compared to state-of-the-art deep networks). Once trained, the network can then be used with new samples to predict the point-wise excitation power needed for a given level of brightness (Fig 3.1c, bottom). In a shot noise-limited regime, the signal-to-noise ratio (SNR) is proportional to $\sqrt{N}$, where N is the number of photons collected, while brightness is proportional to N (assuming a detector with a linear response). Thus this brightness level can be interpreted at SNR^2 for a constant level of labeling density. After the one-time network training with standard candles, experiments can be fluorescently labeled without standard candles, and only the sample shape is needed to predict excitation power.

Modulating excitation light across field-of-view The appropriate excitation power often varied substantially across a single 220x220μm field-of-view – visibly so when imaging curved edges of the lymph node where the sample was highly inclined relative to the optical axis, thereby including both superficial and deep areas of the lymph node (**Fig. 3.2**). The trained network predicted very different excitation powers from one corner of the field-of-view to another in such cases. In order to be able to deliver the correct amount of power, we need to be able to spatially pattern excitation light at different points within a single field-of-view as the microscope scans through all points in 3D. To accomplish this, we designed a time-realized spatial light modulator (TR-SLM) capable of modulating excitation laser power over time as it raster scans a single field-of-view (**Fig. 3.2, 3.3, 3.4**). Unlike a typical SLM, we leverage the point scanning nature of multi-photon microscopy to achieve 2D spatial patterning by changing the voltage of an electro-optic modulator (EOM) at a rate faster than the raster scan rate in order to spatially pattern the strength of excitation. 3D spatial patterning is achieved by applying different 2D patterns when focused to different depths. This method has the advantage or avoiding reflection or transmission losses associated with SLMs, thereby maintaining use of the full power of the excitation laser. The TR-SLM was built using an Arduino-like programmable micro-controller connected to a small op-amp circuit that output a voltage to an EOM, allowing it to retrofit an existing multiphoton microscope for less than $50.

Generalization across samples To validate the performance of LAMI and demonstrate that it can generalize across samples, we trained the network on a single lymph node and tested on a new, differently-shaped lymph node (**Fig. 3.7a**). The test lymph node was seeded with a variety of fluorescent labels and imaged *ex vivo* to eliminate the possibility of motion artifacts associated with intravital microscopy. The surface of the test lymph node was mapped as described previously (**Fig. 3.1b**). Several different desired brightness levels were tested to find one with appropriate signal. For comparison, we imaged the test lymph node with a constant excitation power, with an excitation power predicted by a ray optics model, and with LAMI (**Fig. 3.7b**). Since a full ray optics model was too computationally intensive to be computed at each point in real time, we made the *a priori* assumption of a perfectly spherical sample for our ray optics model comparison. With constant excitation, fluorescence

intensity rapidly decayed after the first 25-50 μm. The ray optics model, which modulated illumination based on both depth and curvature, provided visualization of a much larger area, but still exhibited visible heterogeneity, including areas with little to no detectable fluorescence. This makes sense given that the lymph node was not perfectly spherical, which the model had assumed. LAMI provided clear visualization of cells throughout the volume of the lymph node (**Fig. 3.7b**), up to the depth limit imposed by the maximum power of the excitation laser on our system of around 300-350 μm (**Fig. 3.7e**). In interpreting this data, it is important to note that the images were taken sequentially, so some movement of individual cells between images is expected. Similar performance was maintained even on lymph nodes with irregular, multi-lobed shapes.

Unlike a flat sample, where fluorescence attenuates with depth following an exponential function [51], a curved, convex sample such as a lymph node has a sub-exponential decay with depth (**Fig. 3.14**). To better understand how the appropriate excitation power changes across the sample, we visualized the predictions of the neural network across space (**Fig. 3.7c**). This prediction can be used to make a quantitative comparison of volumes of the sample to which appropriate excitation power can be delivered with LAMI vs. other common adaptive excitation strategies in multiphoton microscopy (**Fig. 3.5**).

In order to conduct LAMI experiments on *in vivo* samples, we added two additional data processing steps: 1) Correcting motion artifacts, which are an inescapable feature of intravital imaging (**Fig. 3.12, 3.13**) and 2) developing a pipeline for identifying and tracking multiple cell populations across time (**Fig. 3.13, 3.6**). The latter used an active machine learning [71] framework to amplify manual data labelling, which led to a 40$\times$ increase in the efficiency of data labeling compared to labeling examples at random(**Fig. 3.13**).

***In vivo* lymph node imaging under physiological conditions** Using our system, we conducted a biological investigation of a common model system for response to vaccination, *in vivo* imaging of a murine popliteal lymph node in anesthetized mouse. Subunit vaccines are a clinically used subset of vaccines in which patients are injected with both a part of a pathogen (the antigen/subunit) and an immunogenic molecule to elicit a protective immune response (the adjuvant). A common model system for these consists of mice being immunized with Ovalbumin, a protein in egg whites, as a model antigen and lipopolysaccharide adjuvant. Before immunization, fluorescently labelled T cells that specifically respond to Ovalbumin (monoclonal OT-I and OT-II T cells) are also transferred to the host mouse so that their antigen-specific behavior can be observed in relation to antigen-presenting cells in the local lymph node where the initial immune response occurs.

Typically, these experiments can only image a small volume of the lymph node at once. In order to visualize a sufficient number of antigen-specific T cells, previous studies transferred 2-3 orders-of-magnitude more monoclonal cells than would typically exist under physiological conditions, a modification that is well established to alter the dynamics and outcomes of immune responses [90, 37, 4, 50]. With our LAMI techniques, we can deliver the correct excitation power to 10-100$\times$ larger volumes of tissue (the exact number depends on what

baseline, as described in**Fig. 3.5, Methods**), so the perturbation of introducing a physiologically unrealistic number of cells is no longer needed. We use an endogenous population of fluorescently labelled antigen-presenting cells, type I conventional dendritic cells labeled with Venus under the XCR1 promoter [172].

24 hours after immunization with lipopolysaccharide, the type I conventional dendritic cell network exhibited a marked reorganization (**Fig. 3.8**,), with XCR1+ cells clustering closer to each other and moving from a more even distribution throughout various areas of the lymph node into primarily the paracortex. We found that these clusters of dendritic cells were located primarily around OT-I (CD8 T cells specific to Ovalbumin) rather than OT-II (CD4 T cells specific to Ovalbumin) or polyclonal T cells, and closer to high endothelial venules than in the control condition.

Imaging and tracking dendritic cells in a control condition and at 24 hours after immunization revealed that this reorganization was accompanied by a change in motility, with dendritic cells at the 24 hour time point moving both more slowly and in a subdiffusive manner, thus confining themselves to smaller areas compared to the more exploratory behavior of the control condition (**Fig. 3.10a**). This decrease in average motility appeared global with respect to anatomical subregions and the local density of other dendritic cells (**Fig. 3.9**). These changes in dendritic cell motility were also accompanied by changes in T cell motility in an antigen-specific manner. OT-I T cells, which appeared at the center of dense clusters of dendritic cells, showed the most confined motility compared to polyclonal controls, while OT-II cells were often found on the edges of these clusters with slightly higher motility (**Fig. 3.10b**,).

To understand how this reorganization takes place, we next imaged lymph nodes 5 hours after immunization. Although dendritic cell motility has not yet changed at this time point, the increasing formation of clusters is detectable on the timescale of an hour (**Fig. 3.10c**). Over time, new clusters appeared to form both from spatially separated dendritic cells moving towards one another, and from isolated dendritic cells moving towards and joining larger existing clusters.

These findings reveal that there is a marked difference in the location and behavior of dendritic cell networks encountered by T cells that enter an inflamed lymph node at the beginning versus the later stages of an immune response. Notably, they also show that the larger-scale dendritic cell reorganization precedes the T cell activation-induced motility arrest that we and others observe amongst antigen-specific T cells at the 24 hour time point.

We speculate that this increased local concentration of dendritic cells may be necessary for rare, antigen-specific T cells to find one another and form the homotypic clusters required for robust immunological memory [38]. The reorganized environment could be an important factor in the difference in differentiation fate of T cells that enter lymph nodes early vs late in immune responses [25].

3.3 Discussion

We have demonstrated how a computational imaging multiphoton microscopy approach, learned adaptive multiphoton illumination (LAMI), provides a rigorous, data-driven approach for adapting illumination to achieve sufficient signal-to-noise without over-illuminating the sample. This removes an important source of human bias, heuristic adjustment of illumination, and thereby enables automated, reliable, and reproducible imaging experiments. Significantly, it neither requires specialized sample preparation nor additional calibration images that deplete the sample's photon budget. This technology enables imaging experiments with more physiological conditions. In this work, we demonstrate an example of lymph node imaging with 100× lower T cell frequencies.

LAMI is most useful for highly-scattering samples with non-flat surfaces (e.g. lymph nodes, large organoids or embryos), which have complicated functions mapping shape to excitation. Applying LAMI to other tissues will require development of sample-specific standard candles. There are many possibilities for these–the only strict conditions are having a labelling density that is not location specific and that individual standard candles can be spatially resolved. Some possibilities for standard candles include genetically-encoded cytoplasmic fluorophores or organelles or fluorescent beads. Other samples will also require a means of building a map of the sample surface. Though this work uses second harmonic generation signal at the sample surface, reflected visible light might be better suited for this purpose. This process could also be automated to improve imaging speed.

Although scattering of excitation light is likely the largest factor responsible for the drop in fluorescence with depth, absorption of emission light may also play a role, especially when imaging deeper into the sample. The fact that far-red fluorphores can be seen at greater depths than those in the visible spectrum supports this possibility (e.g. eFluor670 cells in Fig. 3.7b). Since the neural network makes no distinction between a loss of fluorescence from scattered excitation light and one from absorbed emission light, it is possible that the network learns to compensate for some combination of the two. Compensating for absorbed excitation light would imply that fluorescence emission and photobleaching increase at greater depths (which anecdotally seemed to be the case). It is also possible that the sample does not have a spatially-uniform scattering potential, but that the neural network learns to implicitly predict and compensate.

There are many areas in which LAMI could be improved. The biggest issue in delivering the correct amount of power to each point in intravital imaging of lymph nodes was the map of the sample surface becoming outdated as the sample drifted over time. To combat this, we employed both a drift correction algorithm and periodically recreated the surface in between time points based on the most recent imaging data. We note that our system used a modified multiphoton system not explicitly designed for this purpose, and building a system from scratch with better hardware synchronization between scanning mirrors, focus, and excitation power would increase temporal resolution several fold and lessen the impact of temporal drift. Using state-of-the art image denoising methods [168] would also allow for faster scanning.

The maximum depth of LAMI in our experiments was limited by the maximum excitation laser power that could be delivered. A more powerful excitation laser could push this limit deeper, or using 3-photon, rather than 2-photon excitation. Another improvement to depth could be made by coupling adaptive illumination with adaptive optics (AO). Incorporating AO could lessen the loss in resolution with depth and potentially restore diffraction-limited resolution deep in the sample. Combining ideas form LAMI with adaptive optics could be especially powerful. One limitation of adaptive optics in deep tissue multiphoton microscopy is the need for feedback from fluorescent sources to pre-compensate for scattering [164, 96, 61], making the achieved correction dependent on the brightness and distribution of the fluorescent source being imaged. We have demonstrated in this work that it is possible to predict the appropriate excitation amplitude from sample shape alone. We speculate that a similar correction might be possible for the phase of excitation light, since scattering is caused by inhomogeneities in refractive index, and the largest change in refractive index seen by the excitation wavefront is likely to be at the surface of the sample when it passes from water into tissue. Deterministic corrections based on the shape of the sample surface have indeed shown to improve resolution in cleared tissue [91], and the additional flexibility of a neural network could provide room for further improvement.

In contrast to contemporary techniques based on deep learning [75], the neural network we employ is simple and shallow (1 hidden layer with 200 hidden units). Adding layers did not increase the performance of this network on a test set. We believe this is a consequence of the relatively small training set sizes we used ($10^4 - 10^5$ examples). Larger and more diverse training sets and larger networks would likely improve performance and potentially allow for additional output predictions such as wavefront corrections.

In conclusion, LAMI is a powerful technique for adaptive illumination in multiphoton imaging, with the potential for opening a range of biological investigations. We were able to implement it on an existing two-photon microscope using only an Arduino-like programmable micro-controller and a small op-amp circuit for less than $50. A tutorial on how to implement LAMI using exclusively open-source hardware and software can be found on Zenodo[115].

3.4 Methods

Quantifying the increased volume imaged with LAMI

In order to understand the increases in volume provided by LAMI, we must first consider the problem of signal decay in multiphoton microscopy, the commonly used techniques for addressing this problems, and the unique challenges that arise when applying these techniques to curved samples.

Challenge 1: non-exponential decay profile The literature reports that two-photon fluorescence decays exponentially with depth at constant power, and thus requires an exponential increase of power with constant depth in order to compensate and achieve uniform signal [51]. A simple geometric argument demonstrates why this is not true of curved samples. The exponential increase in attenuation is based upon the assumption that the path

lengths through the sample of excitation rays increase linearly with depth. However, in a curved sample this is not the case. Specifically, in the case of convex samples like lymph nodes, the distance traveled through the sample by marginal rays increase sublinearly as a function of the distance focused into the sample (Fig 3.14a, b). As a result, the required power to compensate and achieve uniform fluorescence must be sub-exponential with depth (Fig. 3.14c).

Challenge 2: decay profile must be relative to top of sample. In multiphoton systems where an arbitrary function (i.e. not just an exponential) can be set to increase power with depth, the power increase profiles are usually a function of the microscopes Z-axis. For a curved sample, this means that the power profile will not be applied from the top of the sample itself. Thus, in order to properly apply a decay profile, the microscope must incorporate some knowledge of the position of the top sample and offset the decay profiles appropriately.

Challenge 3: functional form of decay profile changes across curved samples. Even with the ability to apply arbitrary offsets depending on the the location of the top of the sample, the function mapping depth to fluorescence decay can change across the sample depending on the local curvature being imaged through. To be able to image a curved lymph node in full, one must know an ensemble of such profiles, such that the appropriate one can be applied based on the local shape.

Estimating the increase in volume using LAMI. Before getting into the details of the calculations, an important point must be clarified: For a given object in the sample, there are a range of laser powers that might appear to be acceptable. That is, anything above the threshold where it becomes visible and below the threshold at which visible heat damage occurs. However, photobleaching and photodamage are occurring well below the upper threshold where the sample can be clearly seen burning. Thus, our criteria is not to end up anywhere in this range, but rather to be at its very bottom: generating enough emission light for visualization and analysis with the minimal possible excitation power.

Figure 3.5a shows a comparison of various potential strategies for spatially modulating illumination in a popliteal lymph node, our calculations for the volume that each would be able to image, and the parameter values used in those calculations. The 3D volume of illumination power predictions made by LAMI is used as the target value for excitation power at each point.

The top row shows the simplest strategy: constant illumination. With this strategy, a small strip on the upper portion on the lymph node, where the required power is approximately constant can be made visible. This strip does not extend to the lower portion because shadowing of half of the excitation light requires greater power here. It is not possible to image larger areas deeper in the lymph node without overexposing adjacent areas. We model this as a spherical shell with a 25 μm thickness. We multiply the resultant volume by a factor of $\frac{1}{4}$ as a rough estimate of the fraction of the hemispherical shell not affected by this shadowing.

Most multiphoton systems are equipped with an ability to modulate laser power with depth. In many cases, this consists of setting a decay constant to modulate power according

to an exponential function. Flat samples often have such an exponential profile with depth, but curved samples such as lymph nodes do not (as shown theoretically in **Figure 3.5)**. Based on the empirical fit in **Figure 3.5b)**, we conclude that such a strategy only works up to 140μm deep, and thus set the h parameter for this method equal to 140.

Many multiphoton systems are not limited to only an exponential increase, but can increase power with depth according to an arbitrary, user-specified function. In this case, the microscope can image up to the depth limit according to the maximum laser power, which on our system is $\sim$300μm. Thus, we set the h parameter for this method equal to 300.

In either case, a typical multiphoton system will do this modulation as a function of the coordinate of the position of the Z-drive. Thus, these profiles only remain valid for XY shifts over which the top of the sample has not changed significantly in Z. The radius of this shift was used as the value of r in our in rows 2 and 4 of Figure 3.5a. To estimate it, we plotted a series of XZ profiles of the ground truth excitation at different lateral shifts (Fig. 3.5). These profiles began at the Z coordinate of the top of sample, and did not shift as the top of the sample changed (because MPMs without the ability to modulate power in X, Y, and Z during a scan, as achieved without TR-SLM, cannot do this). The corresponding line profiles stay constant for up to 250 μm when imaging the central part of the lymph node, giving us an estimate of 125 μm for the r parameter. This is a best case estimate, because the required profiles are only relatively constant with Z in the central, flatter part of the lymph node. In other areas of the lymph node (which researchers are often interested in imaging, since relevant biology can be quite location-specific in these structured organs), these profiles vary much more quickly, staying constant for no more than 100 μm (giving $r = 50$).

Imaging volumes larger than the previous cases require the ability not just to modulate power along the Z axis, but also to 1) modulate power as a function of X,Y, and Z, and 2) Have a map of the top surface of the sample. The latter is necessary so that the function for increasing laser power can be offset relative to the surface of the sample, rather than being a function of the microscope Z drive's global coordinate space. The former is necessary because the Z coordinate of the surface of the sample can change substantially over a single field of view, so offsetting this function within a single field of view requires the ability to apply different excitation profiles in Z at different XY locations.

Using these two technologies in concert, the illumination system is no longer limited by the shift of the sample surface relative to the Z coordinate of the microscope. That is, the function for increasing power with depth can now be applied with an arbitrary Z offset. In this regime, the lateral extent of what can be imaged is now limited by the distance over which shifted versions of that function remain valid. Closer to curved edges of the sample, the functions shape must change to avoid over-illuminating the sample. To estimate this value, we plotted a series of XZ profiles of the ground truth excitation at different lateral shifts, starting from the top of the sample at the given lateral location (Fig. 3.5d). From these, we estimate a value of 500 μm, and thus set r equal to 250 in row 4 of 3.5a.

To realize the full potential of the multiphoton microscope, we must not only be able to

apply an arbitrary amount of excitation power in X, Y, and Z, and have a robust method for both learning the function mapping shape to power and applying it in real time. The former is accomplished by the TR-SLM, and the latter by LAMI. Using both of these together the full volume of the lymph node (up to the excitation power limit) can be imaged, applying no more than the minimum necessary power. We calculate this as a spherical shell with an outer radius of 510μm and an inner radius given by the depth that can be imaged with the laser at maximum power: (510 minus 300).

Microscope

All imaging was performed on a custom-built two-photon microscope (with 20× 1.05 NA water immersion objective) equipped with two Ti:sapphire lasers, one MaiTai (Spectra-Physics) and one Chameleon (Coherent). The former was tuned to 810nm and the latter was tuned to 890 nm in order to provide a good combination of incident power and excitation for the set of fluorophores used. The microscope had 6 photomultiplier tube detectors in different bands throughout the visible spectrum, giving 6-channel images. All data was collected using Micro-Magellan [116] software to control the Prior Proscan II XY stage and two Z drives, a ND72Z2LAQ PIFOC Objective Scanning System with a 2000 μm range, which was used to translate the focus during data collection, and a custom built stepper-motor based Z drive, which was used to re-position the sample due to drift in between successive time points. All Z-stacks were collected with 4 μm spacing was used between successive planes.

Spatial light modulator

Because the appropriate excitation power varies as a function of X, Y, and Z, we need to modulate laser intensity over all of three dimensions. However, typical two-photon microscopes are equipped to only modulate intensity over Z–by changing the laser intensity between different focal planes. Thus, a custom time-resolved spatial light modulator (TR-SLM) was built to provide the ability to pattern illumination across a single XY focal plane. By applying different 2D patterns at each focal plane, the laser intensity could be modulated across X, Y, and Z. This TR-SLM takes advantage of the scanning nature of multiphoton microscopy (MPM)–that is, the final image is built up pixel-by-pixel in a raster scanning pattern. This scanning pattern is physically created inside of the microscope by the changing the angle of deflection of two scanning mirrors. One of these mirrors operates in resonant scanning mode, oscillating back and forth with sinusoidal dynamics to control X position within the image. The second mirror is a galvanometer which operates with linear dynamics to control the Y position within the image. Both mirrors are controlled by a custom built controller box (Sutter Instruments), which outputs TTL signals corresponding to completion of a single line and completion of a full frame (line-sync and frame-sync, respectively).

The basic operation of the TR-SLM is to take these TTL signals as input, determine where in the field-of-view (FoV) is currently being scanned, and apply appropriate modulation to the excitation laser based on a pre-loaded pattern. A circuit diagram for the TR-SLM

can be seen in **Figure 3.4**. The TR-SLM is built from a Teensy 3.2 (a programmable microcontroller) using the Arduino IDE. It connects to the controlling computer via USB, through which a low-resolution (8x8) XY modulation pattern is pre-loaded via serial communication. The TR-SLM is also connected to the mirror controller's frame-sync and line-sync TTL signals. Each time one of these signals is received, an interrupt fires, which initiates a corresponding timer. In between interrupts, current scanning position in X and Y is determined based on the elapsed time on these timers (using the appropriate inverse cosine mapping for the resonant scanner). The laser modulation is then determined for that point by bilinear interpolation of the low-resolution pattern. This ensures the ability to apply a smooth gradient of excitation across the field rather than a discretized one determined by the resolution of the supplied pattern.

The excitation laser's amplitude is controlled by an electro-optical modulator (EOM), which takes a logic-level input of 0-1.2V (where 0V is off and 1.2V is full power). The EOM's input is controlled by the Teensy's onboard digital-to-analog converter (DAC) via a voltage divider and voltage buffer circuit. The DAC output is put through a voltage divider to lower the logic level from 3.3V to 1.2V in order to utilize the full 12-bit analog control, and the signal is then run through a LM6142 rail-to-rail operational amplifier in a buffer configuration to isolate the DAC output from any downstream current-draw effects.

To validate the performance of the TR-SLM, a uniform fluorescent plastic slide was imaged with different patterns projected onto it (**Fig. 3.3**). Figure 3.3a shows a checkerboard pattern, which is not a realistic pattern that would be projected into a lymph node, but demonstrates the resolution capabilities of the TR-SLM. Along the vertical axis, the pattern can be precisely specified on a pixel-by-pixel basis. However, the pattern is blurred along the horizontal direction, resulting from the average of many images, each with a noisy pattern along that dimension due to the resonant scanning mirror moving along the horizontal axis faster than the vertical axis. The fundamental limitation is the clock speed of the Teensy, which limits how fast the voltage to the EOM that modulates the excitation laser can be updated. However, in practice, this noise is not a problem because the excitation power needed is a smoothly varying function, and thus a more realistic pattern for imaging into a sample is a gradient pattern (Fig. 3.3c).

Imaging experiment setup

The popliteal lymph node was surgically exposed in an anesthetized mouse. Because of the geometry of our surgical setup, only one half of the popliteal lymph node was visible (i.e. the axis running from top of cortex to medulla was perpendicular to the optical axis). Although we were able to image this half of the lymph node, to get a better view of the whole cortical side of the lymph node, we had to cut the afferent lymphatic, so that the lymph node could be reoriented with it's cortex facing the objective lens. The efferent lymphatic and blood vessels were left intact. We note that a better surgical technique might be able to circumvent this limitation.

To start the experiment, the microscope was focused to a point on the top of the lymph

node cortex using minimal excitation power and the signal visible from second harmonic generation (SHG). Micro-Magellan's explore mode was then used to rapidly map the cortex of the lymph node using a low excitation power, and interpolation points were marked on collagen signal from the SHG image. This surface was used not only to predict the modulated excitation power, but also to guide data acquisition: Using Micro-magellan's distance from surface 3D acquisition mode, only data within the strip of volume ranging from 10 μm above the lymph node cortex to 300μm was acquired, rather than the cuboidal volume bounding this volume. This avoided wasting time imaging areas that were either not part of the lymph node, or so deep within it that they are below the depth limit of 2-photon microscopy. Over time the volume being imaged tended to drift. This was partially compensated for by using the drift correction algorithm described below. However, we limited the use of this algorithm to drift in the Z direction where drift tended to be the most extreme (presumably because of thermal effects or the swelling of the lymph node itself). For XY drift, or for Z drift the algorithm did not correct, we periodically paused acquisition and marked new interpolation points on the cortex of the lymph node, in order to update both the physical area being imaged, and the automated control of the excitation laser.

Image denoising

All data were denoised using spatio-temporal rank filtering [116]. Two full scans of each field-of-view were collected at each focal plane before being fed into a 3×3 spatial extent rank filter. Because of the computational load of performing all the sorting operations associated with this filtering at runtime, a computer with a AMD RYZEN 7 1800X 8-Core 3.6 GHz processor was used for data collection, and the filtering operations were paralellized over all cores. In addition, the final reverse-rank filtering step was done offline to save CPU cycles during acquisition. Before processing data, an additional filtering step using a 2D Gaussian with a 2-pixel sigma kernel was applied to each 2D slice to improve signal-to-noise on downstream tasks. We note that while spatio-temporal rank filtering was designed specifically for the task of cell detection applied here, there may be room for further improvement of real-time denoising (and thus lower doses of excitation light) strategies based on deep learning [168].

Ray optics spherical excitation model

On the way to developing standard candle calibration, we experimented with a simulation framework in which lymph nodes were modeled as spheres with homogeneous scattering potential. This framework had several disadvantages, as detailed below. However, it was useful as a starting point for calculating the random excitation powers that were applied to generate the standard candle training data (even though this may not have been absolutely necessary for generating random excitation). It is also useful to understand why it is difficult to accurately make a physics-based model of this problem, and why machine learning is especially useful. The details of this model are described below.

For a single ray propagating through tissue the proportion of photons which remain unscattered and the two-photon fluorescence intensity decay exponentially with depth[51]:

$F = P_0 e^{-\frac{2z}{l_s}}$, where F is the two-photon fluorescence excitation, z is the distance of propagation, P_0 is the incident power, l_s is the "mean free path" for a given tissue at a given wavelength, which measures the average distance between scattering events.

We assume a beam with a Gaussian profile at the back focal plane of an objective lens, which implies that the amplitude and intensity of the cross-sectional profile of the focusing beam are also both Gaussian. We also assume the contribution from photons that are multiply scattered back to the focal point is negligible and that scattered light does not contribute to the two-photon excitation at the focal point. Using a geometric optics model with these assumptions, the attenuation of each ray propagating towards the focal point can be considered separately. Thus, we can calculate the amount of fluorescence emission at the focal point by numerically integrating over all rays in the numerical aperture of the objective, with a known tissue geometry and scattering mean free path (**Fig. 3.11a**). Figure 3.11b shows the output of such a simulation for a spherical lymph node of a given size. Relative excitation power is that factor by which input power would need to be increased to yield the same fluorescence as if there were no scattering. It is parameterized by the vertical distance from the focal point to the lymph node surface, and the normal angle at that surface.

This model suffers from three main drawbacks that preclude its usefulness for predicting excitation in real time: model calibration, model mismatch and speed.

First, in order to calibrate such a model, two difficult to measure physical parameters of the microscope and the sample must be estimated: the complex field of excitation light in the objective pupil plane and the scattering mean free path of the sample. The model is sensitive to miscalibrations of the former, because different angles travel through different lengths of tissue in the sample, so an overfilling or underfilling of the objective back aperture can have a major influence on the amount of light that reaches the focal point. Estimating this likely requires some kind of PSF measurement along with a phase retrieval algorithm (though our model instead used a estimate of a Gaussian profile with zero phase). The second needs to be measured empirically, as such values are not comprehensively available for different wavelengths and different tissue types in literature.

Second, even if the model can be calibrated, it will only work if the model captures all the relevant physics of the problem. Thus, if wave optical effects play an important role here (which they might, with a coherent excitation source such as laser), the model will fail to account for this. The model also assumes a homogeneous scattering potential throughout the lymph node, which we don't necessarily know to be true. In contrast, neural networks are a much more flexible class of models, with the ability to fit many different types of functions without being hindered by model mismatch.

Third, and most importantly, such a model is very computationally costly. It must integrate over the full 2D distribution of rays within the microscope's numerical aperture, calculating the propagation distance through the sample for each one. Our implementation of this took ∼1s per focal point, and 64 such calculations must be done for each field view, which is acquired in 60 ms. Such a model would need to be sped up 1000x in order to be applied in real time. Because of this, the implementation used to generate the data in our

figure had to be pre-computed, and thus couldn't know the actual shape of the sample, adding another source of model mismatch and potentially explaining the suboptimal performance. The neural network model, in contrast, could be evaluated in less than 1 ms, and could thus be applied in real time without extensive computational optimization.

Standard candle calibration

The training data for standard calibration was collected by imaging an inguinal lymph node *ex vivo*, which had previously been seeded with 2×10^6 lymphocytes from a Ubiquitin-GFP mouse and $2\text{x}10^6$ lymphocytes from a B6, which had been labelled in vitro with eFluor-670 (e670). Each population was used as a standard candle for one of the two excitation lasers on the system, which had their wavelengths tuned to 810nm and 890nm. Two separate images were recorded, one with each laser on. The lymph node was imaged by tiling multiple Z-stacks in XY to cover the full 3D volume. Each Z-stack was imaged with power determined using the output of the spherical model described above, multiplied by a randomizing factor drawn from a uniform distribution between 0.5 and 2. This randomly distributed brightness data was then fed into the cell segmentation and identification pipeline described below. The mean brightness was taken for all voxels within each segmented region as the brightness of the standard candle. The standard candle's spatial location was used to determine the EOM voltage applied at that point in space, its location in the XY FoV, and a set of statistics to serve as effective descriptors of the physics of light scattering and emission light absorption.

The physical parameters were computed by measuring 12 distances from the focal point of the standard candle to the top of the interpolation marking the cortex of the lymph node (**Fig 1** in main text). All 12 distances were measured along directions that had the same angle of inclination to the optical axis (ϕ), with equally spaced rotations about the optical axis. Taken in its raw form, each element of this feature vector is associated with a specific absolute direction in the coordinate space of the microscope. The microscope should be approximately rotationally-symmetric about its optical axis. We don't want the machine learning model to have to learn this symmetry from data, because it would needlessly increase the amount of training data needed. Thus, we explicitly build in this assumption by binning all distances into a histogram. We use nonlinearly-spaced bin edges for this histogram, based on the intuition that scattering follows exponential dynamics with propagation distance, so relative difference in short distances of propagation are more significant than those same differences at long distances. The bin edges of this histogram were calculated by taking 13 equally-spaced points from zero to one, and putting them through the transformation $f(x) = (x^{1.5})(350\mu m)$, where 350 μm is the propagation distance beyond which we don't expect excitation light to yield any fluorescence excitation.

Standard candle brightness, location in XY FoV, and the physical parameter vector were concatenated into a single feature vector. Each of these feature vectors corresponded to one standard candle cell and was associated with a scalar that stored the voltage of the EOM used to image that standard candle. The total number of these pairs were 4000 for the GFP standard candles and 14000 for the e670 standard candles. We standardized all feature

vectors by subtracting their element-wise mean and dividing by their element-wise standard deviation. We then trained a fully connected neural network with one 200-unit hidden layer and a single scalar output. The network was trained using the Adam optimizer, dropout with probability 0.5 at training, and a batch size of 1000. Training was continued until the loss on the validation set ceased to decrease.

The output of this network is the voltage on a particular EOM. Because the goal of this network is to deliver the right amount of excitation power, as opposed to voltage, we converted this voltage into an estimate of relative excitation power (in arbitrary units) before feeding it into a squared error loss function. We measured the function relating EOM voltage to incident power empirically by placing a laser power meter at the focal plane of the objective lens and measuring the incident power under several different voltages. We found this curve to be well approximated by a sinusoid, so we fit the parameters of this sinusoid and used it directly in the loss function.

We experimented with several different architectures before finding the one that worked best with our data. Neither adding additional hidden layers, nor increasing the width of the existing hidden layer beyond 200 improved performance. The best performing value of ϕ (The angle of inclination to the optical axis) was 20°. Neither other angles, nor using multiple angles improved performance on the validation set. This was somewhat surprising, as we would have expected more information about the local geometry to improve prediction. We suspect that this might be the case with a larger training set.

Using standard candle calibration to control laser power

On later experiments, we loaded the trained weights of the network, computed its output for 64 points in an 8×8 grid for each XY image, and sent these values to the TR-SLM through serial communication. The element of the vector corresponding to standard candle brightness must be chosen manually, and can be thought of a z-score of the distribution of brightnesses in the training set (since the training set was standardized prior to training). For example, picking a value of 0 means the network will provide the right laser power to achieve the mean brightness in the training set. Picking a value of -1 means it will aim for a brightness 1 standard deviation below the mean value of the training set.

To calculate the physical parameter feature vector, we computed the interpolation of the lymph node surface as described previously. This interpolation yields a function of the form $z(x, y)$, where there is a single z coordinate for every XY position (unless the XY position is outside the convex hull of the XY coordinates of all points, in which case it is undefined). To avoid having to repeatedly recalculate this function, it is evaluated automatically over a grid of XY test points and cached in RAM by Micro-Magellan. In order to fill out the physical parameter feature vector, we must calculate the distance from an XYZ location inside the lymph node to it's intersection with the interpolated surface. We measure this distance numerically, using a binary search algorithm. This algorithm starts with a value larger than any distance we expect to measure (i.e. 2000μm), tests whether the Z value for this XY position is above the surface interpolation or undefined (which means it is outside

the lymph node), halves the search space, and repeats this test until the distance is within some tolerance (we used μm). These calculations were all handled on a separate thread from acquisition so that they could be pre-computed and not slow down acquisition. We note that our strategy of sending each pattern out as a serial command certainly prevents the system from running as fast as it might otherwise be able. Sending out many such patterns at once and relying on a system that uses hardware TTL triggering should dramatically increase the temporal resolution of this technique.

Validating standard candle calibrated excitation

To validate the use of standard candle calibrated excitation, we transferred 2×10^6 GFP lymphocytes, 2×10^6 RFP lymphocytes, and 2×10^6 e670 lymphocytes and imaged its mediastinal lymph node *ex vivo*. We note that the mediastinal lymph node is quite different in size and shape than an inguinal lymph node. The lymph node was imaged with constant excitation, excitation predicted by the spherical ray optics model, and excitation predicted by the standard candle neural network (**Fig. 2**). The transferred lymphocytes included both T cells and B cells, meaning that there should be fluorescently labelled cells throughout the volume of the lymph node.

Drift correction

Focus drift, primarily in the Z direction, was present in all experiments at rates on the order of $\sim1\mu$m/min. This is unsurprising given the massive influx of cells to lymph nodes during inflammatory reactions. It was essential to compensate for this drift, because not doing so would lead to a mismatch between the coordinates of our interpolation marking the lymph node cortex and it's actual location, which would in turn mean the automated excitation would be misapplied. To compensate for this drift, we designed a drift compensation algorithm that ran after each time point, and changed the Z position of secondary Z focus drive (i.e. not one used to step through Z-stacks) after each time point. Estimates of drift were based on the second harmonic generation signal from the fibers in the lymph node cortex, which were a convenient choice because their contrast was not dependent on fluorescent labelling, and their spectral channel (violet) had relatively little cross-talk with other fluorophores. At each time point after the second, the cross-correlation of the 3D image in violet channel was taken with the corresponding image from the previous time point. The maximum of this function was taken within every 2D image corresponding to a single slice, and a cubic spline was fit to these maxima. The argmax of the resulting smooth curve was used to estimate the offset in Z between two successive timepoints with subpixel accuracy. This estimate was used to update an exponential moving average that estimated the rate of drift, so that both the existing drift from the previous timepoint could be corrected, and the expected future drift could be pre-compensated for. In practice, this algorithm worked well enough to stabilize the sample enough for the adaptive illumination to be correctly applied. Remaining drift in the imaging data was corrected computationally as described below.

Image registration

In order to conduct *in vivo* investigations, we must first address motion artifacts, which are an inescapable feature of intravital imaging and can compound when imaging large volumes over time. We thus develop a correction pipeline based on iterative optimization of maximum a posteriori (MAP) estimates of translations for image registration and stitching (**Fig. 3.12, 3.13**). These corrections enabled the recovery of stabilized timelapses in which cell movements can be clearly visualized and tracked.

We identified three types of movement artifacts that occurred during intravital imaging. 1) Due to the mouse's breathing, there were periodic movements of successive images relative to one another within each Z-stack. These movements could be well approximated by motion within the XY plane, in part because of the geometry of the imaging setup, and in part because of the heavily anisotropic resolution of the imaging system, in which objects were blurred out along the Z axis much more so than X and Y. 2) Individual Z-stacks were misaligned with each other in X, Y and Z. This seemed likely to be caused by physical movement of the sample as a result of some combination of thermally-induced focus drift and biological changes leading to small tissue movements. 3) Global movements of the entire sample over time. All three remained to some degree even after experimental optimizations to improve the system stability and pre-heating the objective lens to minimize thermal drift.

To correct these artifacts, we used a three stage procedure with each step corresponding to a type of movement artifact. Although cross-correlation is often the first choice for rigid image registration problems in the literature, it was found to be ineffective for solving two of the three of problems. Thus, we employed a more general framework, using iterative optimization to compute *maximum a posteriori* (MAP) estimates. This framework depends on the ability to transform and resample the image in a differentiable manner. As shown in Fig. 3.14a, we can set up a general image registration problem that can be solved by numerical optimization by creating a parametric model for how pixels move relative to one another, resampling the raw image based on the current parameters of this model, and then computing a loss function that describes how well the alignment based on this transformation is. This paradigm enables us to solve general MAP estimation problems of the following form with iterative optimization:

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} L(f_1(\boldsymbol{\theta}), f_2(\boldsymbol{\theta}), ...) + R(\boldsymbol{\theta})$$

Where $\boldsymbol{\theta}$ are the parameters to optimize, f_n is the transformation and resampling of the n^{th} subset of pixels, L the loss function, and $R(\boldsymbol{\theta})$ is a regularization term for the parameters that allows incorporation of prior knowledge. We used the deep learning library TensorFlow to set up these optimization problems. This had the advantage of being able to automatically calculate the derivatives needed for optimization using built-in automatic differentiation capabilities. Often, these problems used extremely large amounts of RAM, because all image pixels were stored in memory when performing optimization. We were able to do this by using virtual machines on Google Cloud Platform with extremely large amounts of memory (>1TB). However, we note that it would be possible to reduce the RAM

requirements by downsampling the images, or more carefully coding the optimization models to only use relevant parts of the images rather than every pixel.

For the first correction, movements in XY for each Z-stack, each Z stack was optimized separately. We observed that XY movements were almost always confined to a single z plane and that looking at an XZ or XY image of the stack, these movements were clearly visible as discontinuities along the Z axis. Thus we parameterized the model by a (number of Z-planes)×2 vector, corresponding to an XY shift for each plane. For this correction, all channels except for the channel corresponding to second harmonic generation were used. The loss function was taken as the sum over all pixel-wise mean-squared differences between consecutive z planes, normalized by the total squared intensity in the image (which was necessary to ensure that the learning rate of the optimization did not need to be adjusted to accommodate the total brightness of the Z-stack):

$$L(\boldsymbol{x}, \boldsymbol{y}) = \frac{\sum_{j=0}^{N-1} \sum_{x',y'} (\boldsymbol{I}(x' + x_j, y' + y_j, z_j) - \boldsymbol{I}(x' + x_{j+1}, y' + y_{j+1}, z_{j+1}))^2}{\sum_{x',y',z'} \boldsymbol{I}(x', y', z'_j)^2},$$

where $\boldsymbol{x}$ and $\boldsymbol{y}$ are vectors holding the translations at each slice, j is the index of the z plane, N is the total number of Z planes, $\boldsymbol{I(x, y, z)}$ is a pixel in the raw Z-stack, and x', y' are the coordinates of pixels in the raw image. The regularization in this problem was a quadratic penalty on the sizes of the translations (implicitly encoding a prior that these translations should be normally distributed about 0) multiplied by an empirically-determined weighting factor:

$$R(\boldsymbol{x}, \boldsymbol{y}) = \lambda(\|\boldsymbol{x}\|_2^2 + \|\boldsymbol{y}\|_2^2)$$

The value of lambda used was $8{\times}10^{-3}$. The model was optimized using the Adam optimizer and a learning rate of 1. Optimization proceeded until a the total loss had failed to decrease for 10 iterations.

The second correction, fixing movements over time, was computationally much easier to solve, because the strong signal of the similarity between consecutive time points in channels made registration not especially difficult if the correct channels were used. For this reason, this correction did not require iterative optimization, and could instead be solved with cross-correlation alone. The 3D cross-correlation was taken between every consecutive two time points for each Z-stack. The location of the maximum value of each of these cross-correlations gave the optimal 3D translation between consecutive time points, and taking a cumulative sum of these pairwise shifts gave an absolute shift for each stack over time.

The third correction, finding the optimal stitching alignment between each Z-stack, was the most computationally challenging of these problems. This is because it has a relatively small amount of signal (i.e. the overlapping areas of each Z-stack, which was less than 10% of the total volume of each Z-stack). Furthermore the signal in these areas was relatively weak, because it was most susceptible to photo-bleaching since it is exposed to excitation light multiple times at each point. Furthermore, since stacks were often taken a few minutes

apart, the content in these overlapping regions often changed. Compensating for these difficulties not only required using the iterative optimization framework with an appropriate loss function that accounted for variations in image brightness and proper regularization, but also carefully choosing which channels to use registration based on the presence of non-motile fiducial signals. Most of the datasets we collected had a channel with high endothelial venules, a large and immobile structure in the lymph node, fluorescently labelled, and these channels were often the most useful due to strong signal and lack of movement. We also found good performance by including the second harmonic generation channel that provided signal from the collagen fibers in the lymph node cortex. Finally, we noticed that autofluorescent cells were numerous and immobile throughout the lymph node. Because autofluorescence has a broad emission spectrum that appears across 3-4 channels at once, as opposed to the labelled structures which appear over only 1-2, we were able to isolate the signal from these cells by taking the minimum pixel value over several channels.

As shown in Fig. 6b, each Z-stack was parameterized by a 3 element vector that corresponded to its X,Y, and Z shifts. The loss function was taken as the mean of all of the correlation coefficients of the pixels in the overlapping regions of every pair of adjacent Z-stacks. Correlation coefficients are a better choice of loss for this task than cross-correlation, because they better account for variations in image brightness [79]. Optimization was performed using Newton's method with a trust region constraint. Rather than performing optimization on each time point separately, all time points were averaged together and a single optimization was performed taking all information into account. This was possible because relative movements between stacks that differed by time point had already been corrected by cross-correlations in step 2. Because of the strong signal afforded by averaging multiple time points together, no regularization was needed.

Cell identification–feature engineering

We developed a machine learning pipeline for tracking cell locations over time based on their fluorescent labels (**Fig. 3.13**). Automating this process was essential, as some datasets contain thousands of labelled cells at twenty different time points. Our pipeline enabled their detection across all datasets with the manual classification of no more than 500 cells for each time, a task that could be completed in a few hours of manual effort. Briefly, this pipeline consisted of two stages: a 3D segmentation algorithm to identify cell candidates, followed by a neural network that used hand-designed features (**Fig. 3.6**) to classify each candidate as positive or negative for a given fluorescent tag. We used an active learning [71] framework to efficiently label training data for this classification network, which led to a 40× increase in the efficiency of data labeling compared to labeling examples at random(**Fig. 3.13**).

Cells were detected in a two-stage pipeline that first utilized 3D segmentation to identify cell candidates, followed by machine learning to classify which of those cell candidates belonged to a population of interest. Candidate regions were generated using the segmentation algorithm **(Fig. 3.13)** built into Imaris 7.6.5 (i.e. the "surfaces" module), which includes a

filtering step to smooth the data, a local background subtraction step to account for variations in brightness, a thresholding step to generate segmented regions, and a splitting step, in which seed points of a certain size are generated, and segmented regions are split based on these seed points. Candidates were generated for each population of interest (i.e. each fluorescent label) through the ImarisXT Matlab interface. Next, each candidate region was "featurized" by computing a set of descriptive statistics about the pixels enclosed within it. By default, Imaris outputs a set of 97 such statistics for each candidate, including intensity means, standard deviations, minimums, maximums, as well as a number of morphological features. However, these features are specific neither to the biological or technical context of the data, and we found them to not be effective in all cases for training high-quality classifiers. Thus, we engineered a set of additional features to better capture the variations that are useful for classifying cells.

First, we reasoned that since all spectral channels are collected simultaneous in two-photon microscopy, the ratios of intensity in different channels contains important information. Treating each set of spectral statistics (e.g. intensity means for different channels) as a 6-dimensional vector (for a 6-channel image), we subtracted the background pixel value for each channel, and normalized to unit length. This "spectral normalization" takes advantage of the fact that that intensity measurements for a given fluorescent object are all proportional to the excitation power delivered to the focal point, and thus it normalizes intensity statistics while preserving their ratio. It also creates an additional feature from the magnitude of the vector prior to normalization, which captures the brightness of the object irrespective of it spectral characteristics.

We also designed several feature classes based on the observation that one of the failure modes of the segmentation algorithm in the candidate generation step was that it often created a single region around a cell of interest along with a second cell in close contact to it that expressed a different fluorophore, but had spectral bleed-through into the channel on which the segmentation was run. Thus, intensity weighted centers-of-mass (COMs) within each region would be expected to show greater variance among the different spectral channels compared to a surface that surrounds a structure single source of fluorescence intensity. This should hold true even if the the two objects surrounded by a single surface shared emission in the same channels, as long as the spectral profile of the two objects differs. With this in mind, we computed features for all pairwise distances between the intensity-weighted COM for different channels, as well as the distance from each intensity-weighted COM to the non-intensity weighted COM. With the same reasoning, we also added the correlation matrices containing the pairwise correlations between channels for all pixels within each candidate region as features (Figure 3.14a).

Finally, to more directly address the issue of overlapping, spectrally dissimilar cells (which are often the most biologically interesting case), we designed an algorithm to identify sub-regions of pixels within each candidate region that has a spectrum that is most similar to a reference spectrum (i.e. the spectrum of the fluorophore of the cell of interest). This algorithm is based on the normalized cut segmentation algorithm [144]. However, unlike that algorithm, which is designed for use on grayscale images, and builds an adjacency

matrix for all pixels based on a combination of their spatial and intensity differences, our algorithm segments regions based on differences in their spatial and spectral distances. This is accomplished by defining distances between each pair of pixels as:$d_{i,j} = \alpha\|\mathbf{r}_i, \mathbf{r}_j\|_2^2 + \beta\hat{\mathbf{s}}_i^T\hat{\mathbf{s}}_j$, where $\mathbf{r}_i$ and $\mathbf{r}_j$ are the spatial coordinates of the two pixels, $\hat{\mathbf{s}}_i$ and $\hat{\mathbf{s}}_j$ are their unit norm intensity vectors across all channels, and α and β are tuning parameters. Then, an adjacency matrix can be constructed by defining the adjacency between pixel i and pixel j as $w_{i,j} = e^{-d_{i,j}}$. The spectral clustering method defined in [99] can then be used to break all pixels into distinct regions, and the normalized cut region of interest (NC-ROI) most similar to a given reference intensity can be used for further downstream processing. Using this method, a number of additional features were calculated for the pixels within each NC-ROI.

To validate that these features were in fact useful, we performed two types of analyses. First, we looked at which types candidates cells the classifier repeatedly failed to correctly classify. By running k-fold cross validation on a ground truth set of labelled candidates, we were able to identify which candidates the classifier failed to correctly classify. Overlaying these results on principal component analysis plot of the spectral variation among the cells of interest (RFP labelled T cells), we found that the misclassified cells were often spectral outliers as a result of spatial overlap with some other fluorescent structure (Fig 3.5, left). However, including the engineered features in this experiment dramatically reduced the miscalssificaiton of these cells.

Next we ran a bootstrap analysis to identify the most useful features. We performed regularization and variable selection via the elastic net procedure. Elastic net is a useful method for identifying a sparse subset of useful predictors in a dataset with correlated predictors.

For the dataset examined, the number of labeled T cells was significantly lower than the number of non T cells (T cells: 204, non T cells: 38, 575). In order to keep a balanced training and test dataset, we partitioned the data so that the model performs training and testing on similar sample distributions. In particular, we performed 100 bootstrap resampling procedures from both T cell and non T cell data of approximately equal sample size. The model obtained was further tested on a smaller test dataset with equal T cell and non T cell ratio, set aside at the beginning of the procedure.

For each bootstrapped sample set, we further ran 1000 iterations of randomly picking test set/training set partitions. This step was performed in order to asses the stability and overall distribution of the lambda parameter picked for each model. Despite lambda being chosen through 5-fold cross-validation procedure, it is still specific to the prior decided training/test set partition. By performing additional random partitions of the bootstrapped dataset, we can break this dependence. Additionally, we can also look at the distribution of cross-validation error provided by the glmnet package, as well as the misclassification error from the test set. The final model was fitted using the lambda parameter with the lowest cross-validation and misclassification error for the bootstrapped sample set. We looked at the averaged probabilities across all the samples, as well the total number of times each predictor was chosen by the elastic net out of 100 bootstrap subsamples.

The final results of this analysis can be seen in Figure 3.5d. Many of the engineered

features are among the strongest predictors, further validating their usefulness for this task.

Cell identification – active learning

Having developed a useful set of task-specific engineered features that can train high quality classifiers with sufficient labelled training data, the last remaining piece of the pipeline is a scheme for generating labelled training data. Often, this can be the most time consuming piece of developing a machine learning system. To alleviate this bottleneck, we drew from the field of active learning, a paradigm in which a classifier chooses which data points receive labels, allowing them to learn more efficiently by seeing more informative examples [71] . Specifically, we use the strategy of "uncertainty sampling" [78], in which the classifier outputs a number between 0 and 1 for each example (with 0 being complete certainty of one class and 1 being complete certainty of the other), and the example with a value closest to 0.5 is then selected and sent to a human for labelling. This process is then repeated until enough training data is labelled to train a classifier that generalizes well to the remaining unlabelled data.

Applied specifically to the problem of classifying which candidate regions corresponded to cells, the workflow was a follows: After computing the engineered features for all candidate regions, we first labelled one example of a candidate that belong to the population and one that did not. These labels were used to train the classifier (a small, fully connected neural network with 12 hidden units). Because classification accuracy usually increased by averaging the predictions of multiple neural nets, 3 were trained in parallel and their predictions were averaged. When making final predictions of cell populations, 100 neural nets were averaged. The neural net was trained in Matlab, and the labelling interface for selecting cells was built with Imaris for data visualization, and Matlab script on the backend that communicated interactively with Imaris through the ImarisXT interface. We periodically predicted the identities of all candidates, in order to identify particularly difficult examples even faster and manually label them.

Although well justified from a theoretical standpoint as a means to make exponential gains in data labelling efficiency on idealized problems [140], there remained the question of whether active learning had the same effect on this problem. To answer this, we generated a ground truth set of labels by carefully manually labelling every cell on a limited dataset of candidates for RFP-labelled T cells. We reviewed each cell multiple times to be sure that its label was not a false positive, and searched manually through the data volume to identify any false negatives. Next, one positive and one negative candidate were randomly selected and assigned labels. This labelled set was used to simulate the uncertainty sampling procedure, drawing labels from the ground truth set rather than a human labeller. The accuracy on the remaining unlabelled examples was used to assess performance. As the plot in Figure 3.5d demonstrates, uncertainty sampling vastly outperformed random sampling for this classification task.

Statistics and Reproducibility

All data analysis was performed in using custom scripts written using tools from the Scientific Python stack [163]. For all displacement vs root time plots, curves were fit to the means of scatterplot data using locally weighted scatterplot smoothing (LOWESS) using locally linear regression [22] using a tricubic or Gaussian weighting functions. Sigma and alpha parameters were tuned manually for each comparison to capture the major trends in the data while smoothing out noise. Error bars represent 95% confidence intervals derived from bootstrap resampling of data with 500 iterations. Cells were tracked over multiple time points using the Brownian motion tracking algorithm in Imaris 7.6.5. In all cases the number of tracks used for quantification overestimates the number of cells since some cells move in and out of the frame or have breaks in their tracks where the algorithm fails. The visualizations of individual tracks used random subsamples of the total number of tracks for visual clarity.

Mouse Immune challenge

OTI and OT2 cells were isolated from lymph nodes of mice. Additionally, polyclonal (C57BL/6) or LCMV P14-specific CD8+ T-cells were isolated as negative controls. Selection was carried out with a negative selection EasySep mouse CD8+ or CD4+ isolation kit (STEMCELL Technologies, 19853 and 19852). If T-cells did not have a transgenic reporter (CD2-RFP or ubiquitin-GFP), they were fluorescently labelled with one of eFluor 670 (ThermoFisher Scientific, 65-0840-85), Violet Proliferation Dye 450 (BD, 562158), or CMTMR (Thermo Fisher Scientific, C2927). Dyes were diluted 1000-fold and incubated with isolated cells for 10-15 minutes minutes in a 37C, 5 percent CO2 incubator. T cells were injected retro-orbitally (r.o.) into Xcr1-Venus recipient mice in 50-100 uL volumes. The number of OT1 and OT2 transferred was $5x10^4$ except for experiments conducted at 5 hours post-infection, where $5x10^4$ cells were transferred in order to visualize more T cell-dendritic cell interactions; for each experiment, equal numbers of OT1 and OT2 cells were transferred. 1e6 control T-cells were transferred. Mice were given a 30 uL footpad injection containing 2.25 ug LPS (Sigma-Aldrich, L6529-1MG) and 20 ug OVA protein (Sigma, A5503-1G) 1-4 days after T-cell transfer; a 30 uL footpad injection of DPBS was used as a negative control to the infection model. To visualize high endothelial venules, in some imaging experiments, 15 ug Meca-79 Alexa Fluor 647 (Novus Biologicals, NB100-77673AF647) or Alexa Fluor 488 (Novus Biologicals, NB100-77673AF488) was transferred r.o. in a volume of 50 uL immediately before imaging.

3.5 Data availability

Source data are provided on FigShare [114].

3.6 Code availability

A streamlined Jupyter notebook that describes how to implement LAMI can be found on Zenodo [115].

All other code including data analysis code can be found on Zenodo [113].

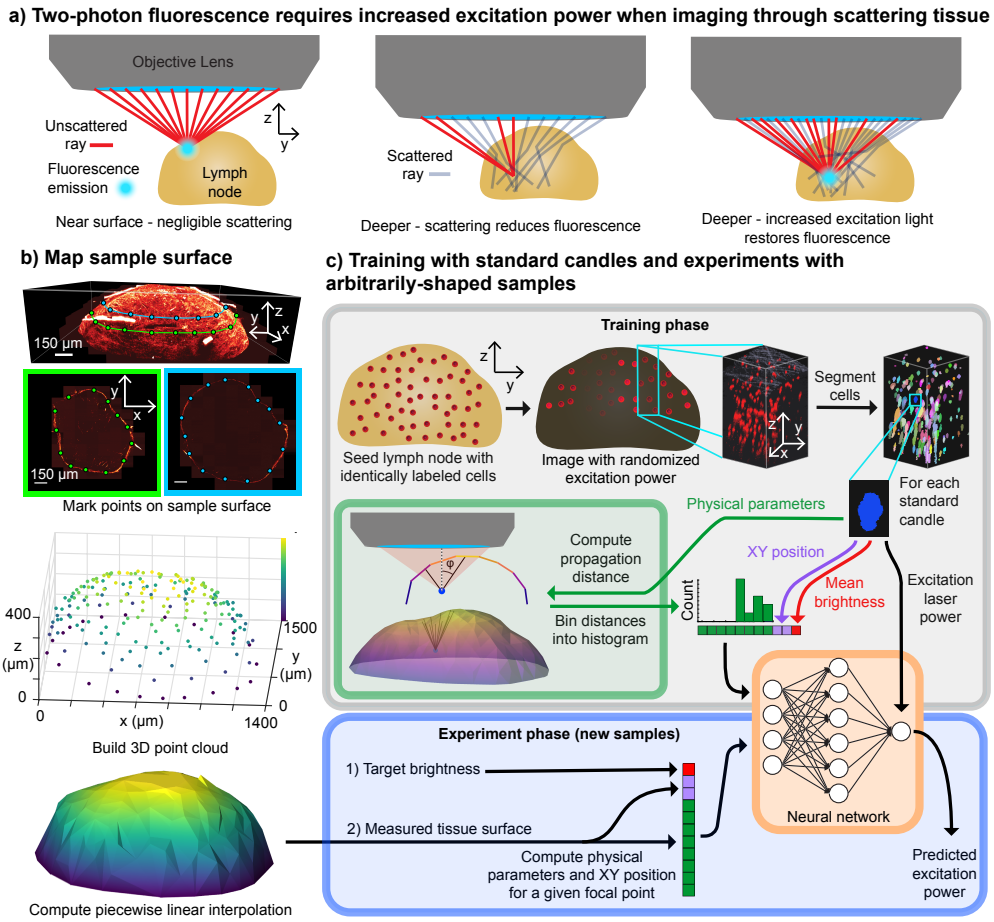

Figure 3.1: **Learned Adaptive Multiphoton Illumination (LAMI).** a) *In vivo* multiphoton microscopy requires increasing laser power with depth to compensate for the loss of fluorescence caused by excitation light being scattered b) Our LAMI method uses the 3D sample surface as input to its neural network. We map it by selecting points on XY image slices at different Z positions (top) to build up a 3D distribution of surface points (middle) that can be interpolated. c) Training uses samples seeded with cells with the same fluorescent label (standard candles), which is imaged with a random amount of power. A 3D segmentation algorithm then isolates the voxels corresponding to each standard candle. The mean brightness of these voxels, position in XY field-of-view, and a set of physical parameters (a histogram of propagation distances through the tissue to the focal point at a specific angle of inclination to the optical axis (ϕ)) are concatenated into a single vector for each standard candle. The full set of these vectors is used to train a neural network that predicts excitation laser power. (Bottom) After training, subsequent samples need not be seeded with standard candles. The network automatically predicts point-wise excitation power as a function of the sample geometry and a user-specified target brightness.

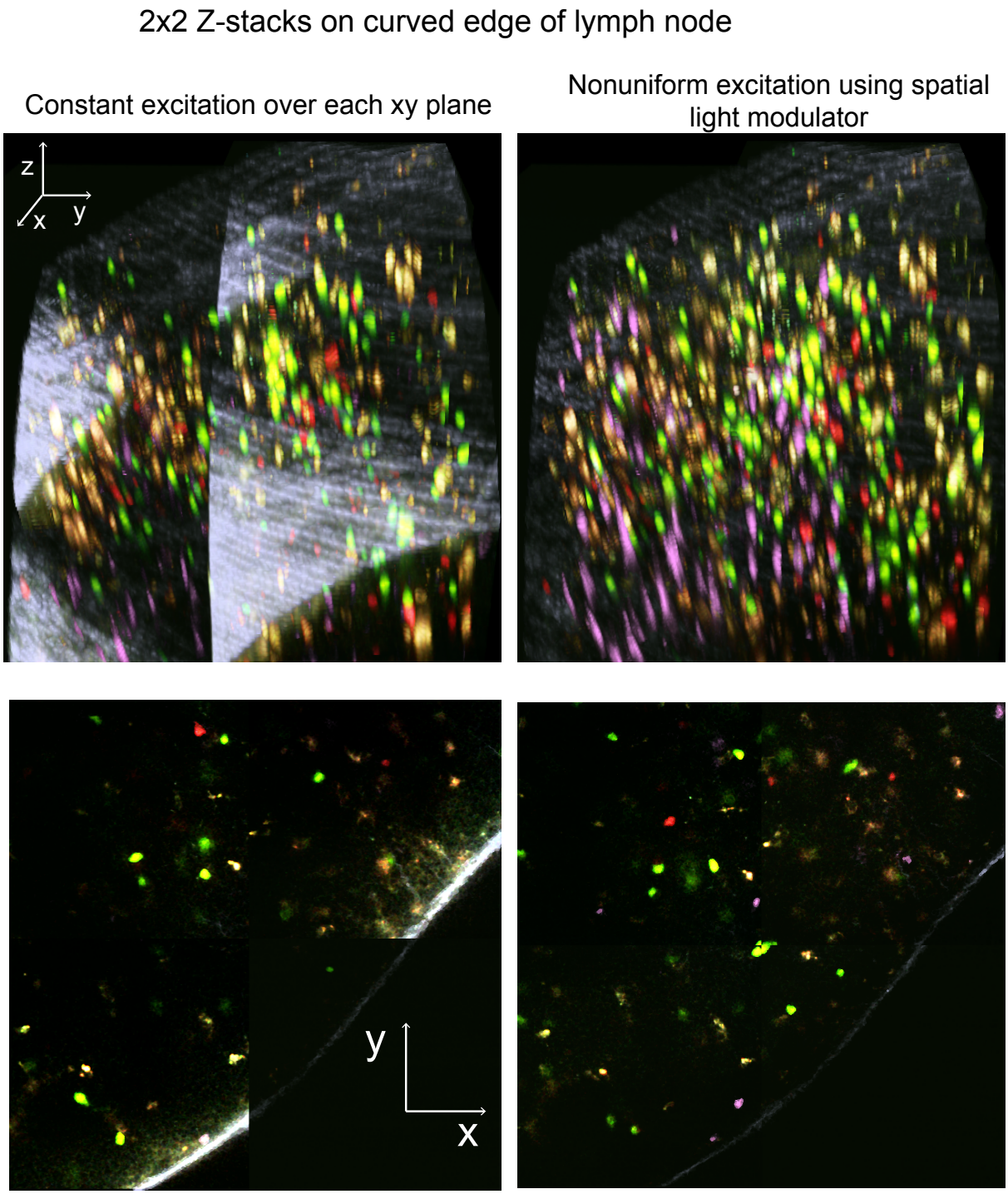

Figure 3.2: **Nonuniform excitation across field on curved tissue**. Imaging into to curved tissue such as the edge of a lymph node requires variable excitation over the XY field of view. 3D view (top) and 2D slice (bottom) of a 2x2 grid of Z-stacks. Left, constant excitation power within each XY plane in each Z stack. Right, variable excitation power allows excitation to be set correctly for each point in XYZ field of view.

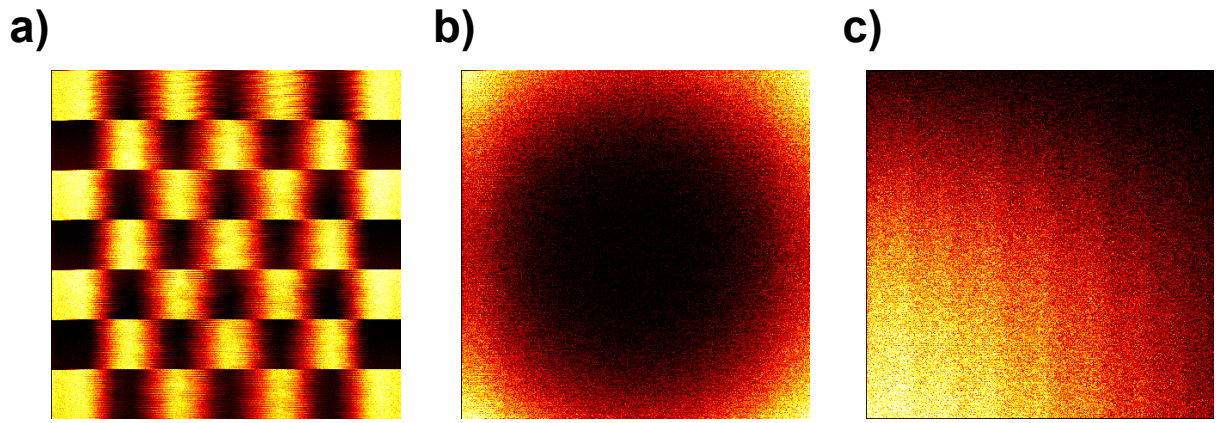

Figure 3.3: **Spatial Light Modulator Test Patterns.** Images taken on flat fluorescent test slide with different patterns of excitation light. a) A checkerboard pattern demonstrating the difference in horizontal vs. vertical resolution. b) A vignetting compensation pattern, with more excitation at the edges of the field of view. c) A gradient across the field of view pattern.

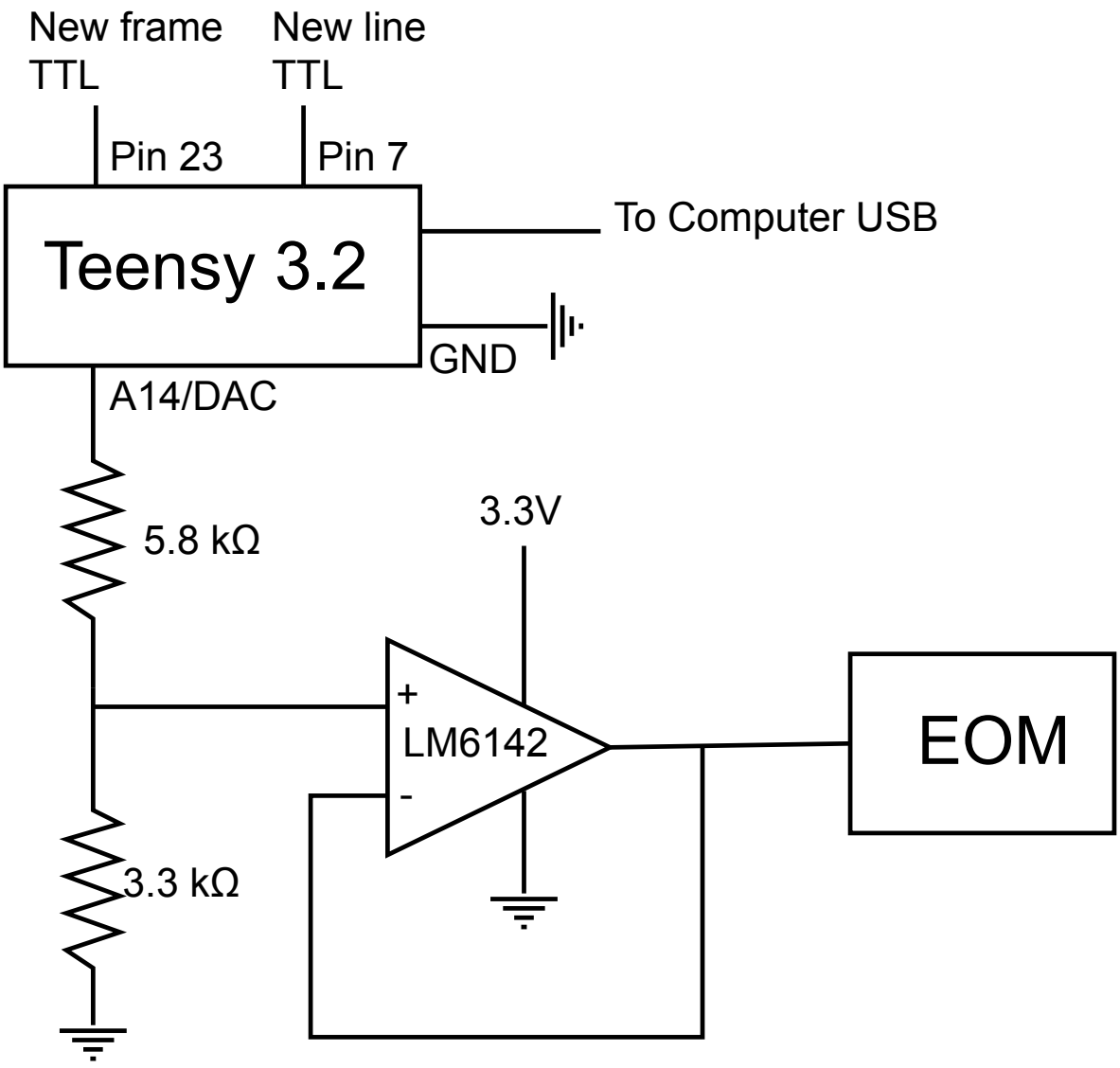

Figure 3.4: **Circuit diagram of time-realized spatial light modulator (TR-SLM)**. Wiring of circuit connecting Teensy 3.2 to electro-optic modulator (EOM) that controls excitation laser power via an op-amp. New frame TTL connects to a trigger that fires every time the raster scan pattern begins a new frame. New line TTL connects to a trigger that fires after resonant scanner completes a new line (which corresponds to two rows of pixels)

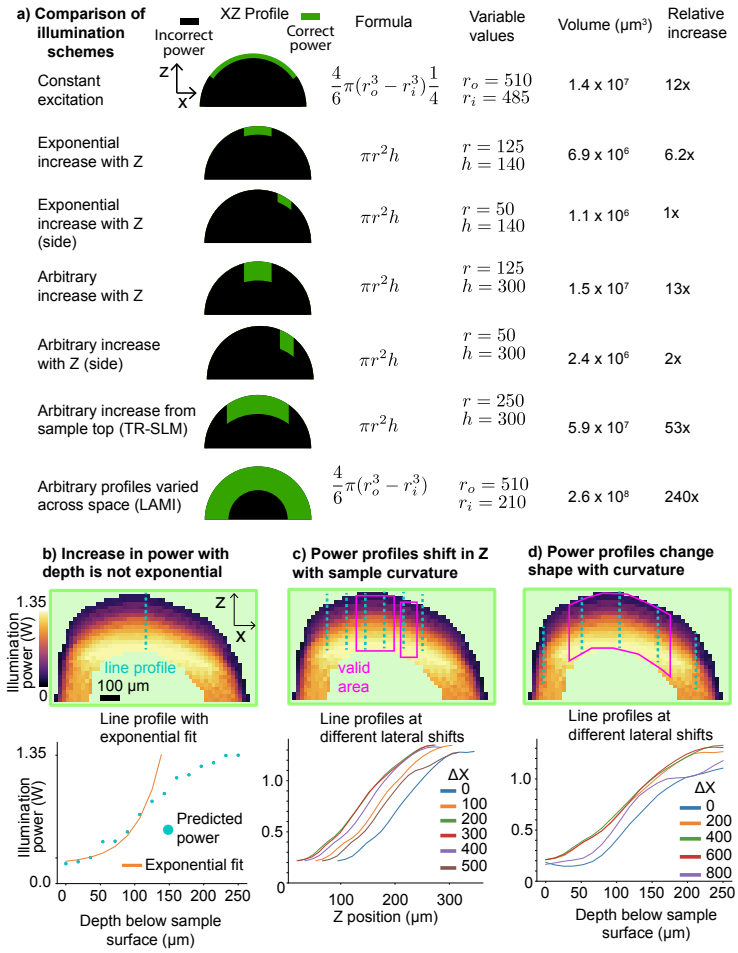

Figure 3.5: **A comparison of different adaptive excitation strategies**. a) Overview of various adaptive excitation strategies, including details of calculations for the total volume each can image. The top 5 rows are strategies that are employed on existing multiphoton microscopes. The bottom two are enabled by the development of time-realized spatial light modulator (TR-SLM) and TR-SLM + learned adaptive multiphoton illumination (LAMI), respectively. Various values used in the calculations are derived from measurements shown in b-d. b) Top, XZ slice of excitation predicted by LAMI in a popliteal lymph node. Cyan dashed line shows profile, which is plotted with an exponential fit on the bottom. The excitation only follows an exponential profile for ∼140 μm. c) Excitation power Z profiles spaced at 100 μm intervals (cyan dashed lines and bottom plot). Magenta boxes show areas that can be imaged with an approximately constant profile in Z. d) Excitation power profiles starting from the top of the sample (cyan dashed lines and bottom plot). Moving towards the edges, the shape of the profiles noticeably changes. The magenta outlined region shows the region that can be imaged with a single excitation profile, applied starting at the top of the sample.

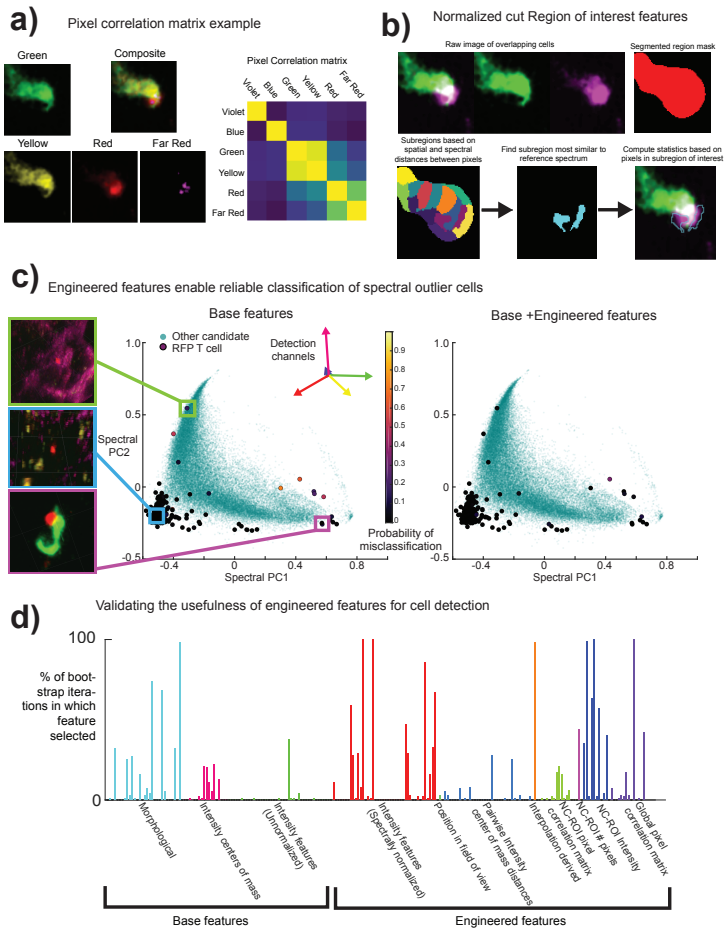

Figure 3.6: **Engineered features for cell classification** a) The pairwise correlations between pixels of different channels. This provides a clear signal (i.e. distinct clusters in the correlation matrix) when a GFP and RFP labelled cell do not entirely spatially overlap. b) Normalized cut features: by breaking down an area of masked pixels (red, top right) into subregions (bottom left), a subregion that is most similar to a reference spectrum (i.e. the magenta cell) can be identified). c) Including engineered features enables robust identification of spectral outliers. Plots show all candidate cells plotted over first two principal components of the average color spectrum of RFP T cells. Left, spectral outliers (representative images shown on left) from the main cluster of T cells also tend to misclassified. Right, adding in engineered features to classification vastly improves the misclassification probability of these spectral outliers. d) Elastic net bootstrap analysis colored by feature class. Many classes of bootstrapped features were selected a high proportion of the time, validating their usefulness in this classification problem.

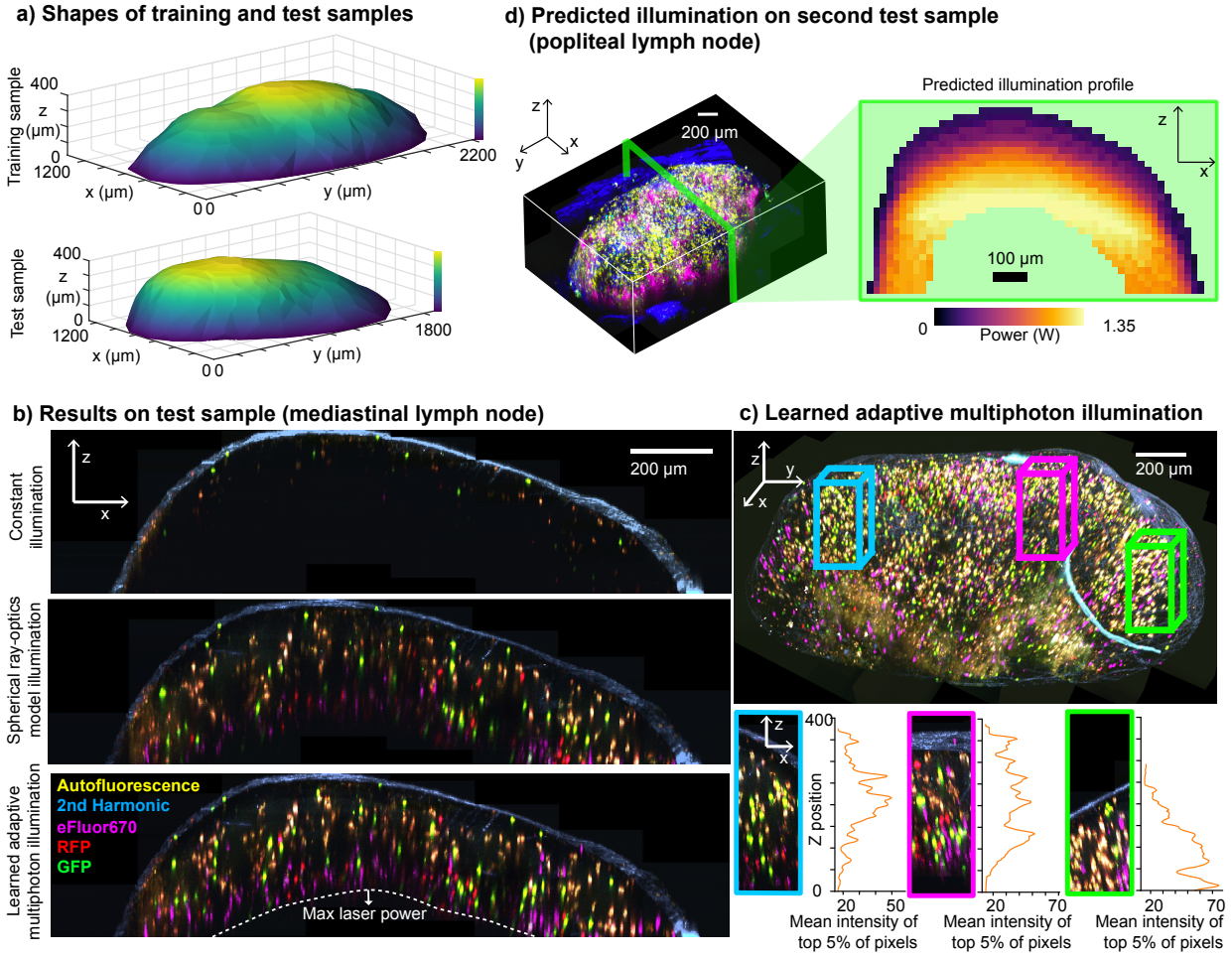

Figure 3.7: **Validation of LAMI on lymph node samples.** a) The surface shapes of lymph nodes used for (top) training with standard candles and (bottom) testing. b) Results with constant illumination power, illumination power predicted by the ray optics model that assumed a perfectly spherical shape, and illumination power predicted by LAMI in the test sample, which had been seeded with lymphocytes labelled with GFP (green), RFP (red) and eFluor670 (magenta). Constant illumination rapidly attenuates the signal with depth. The ray optics model generates contrast throughout the volume, but has visible non-uniformity and areas where the signal from cells is entirely missing. In contrast, LAMI gives good signal throughout the imaging volume up to the maximum excitation laser power. c) A 3D view of the LAMI-imaged lymph node, with several XZ projections of representative areas with different surface curvature. Plots show Z-position vs mean intensity of top 5% of pixels to demonstrate good signal is maintained with depth using LAMI. d) Popliteal lymph node imaged with LAMI along with XZ cross section of predicted illumination.

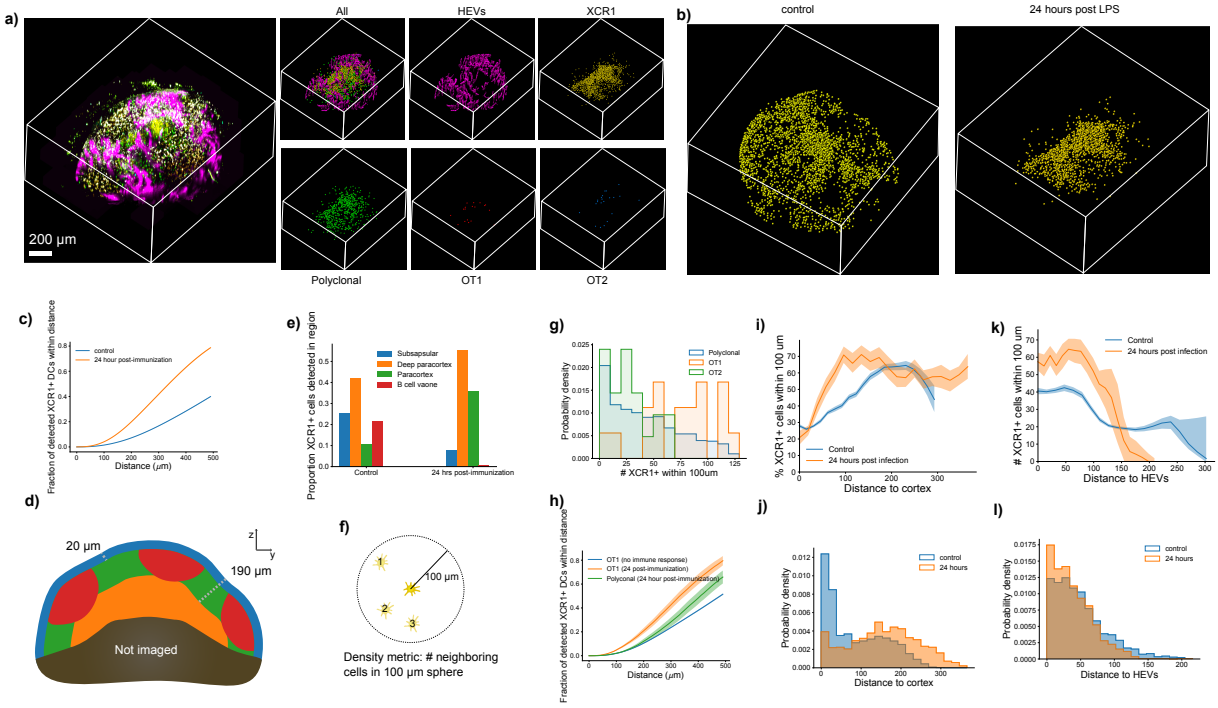

Figure 3.8: **Reorganization of cell population in the lymph node 24 hours after immunization** a) Image data (left) and localizations of XCR1, Polyclonal, OT1, and OT2 as well as 3D segmentation of high endothelial venules. b) Localization of XCR1 cells in control condition and 24 hours after immunization. c) Amount of clustering as assessed by the mean fraction of XCR1 cells within different distances of XCR1 cells. d) Schematic of how the different parts of the lymph node were defined for e), which shows the changes in localization of XCR1 cells from 0 to 24 hours. f) Schematic of the metric used to assess dendritic cell clustering. g) Histograms of DC cluster density at locations of different types of T cells. h) Mean fraction of detected XCR1 cells within distance of different types of T cells. Shaded area represents standard error. i) Mean percent of XCR1 cells within 100 μm vs. distance to cortex at 0 and 24 hours. Error bars represent bootstrapped 95% confidence interval. j) Histogram of XCR1 cell distances to cortex at 0 and 24 hours. k) Percent of XCR1 cells within 100 μm vs. distance to HEVs at 0 and 24 hours. Error bars represent bootstrapped 95% confidence interval. l) Histogram of XCR1 cell distances to HEVs at 0 and 24 hours.

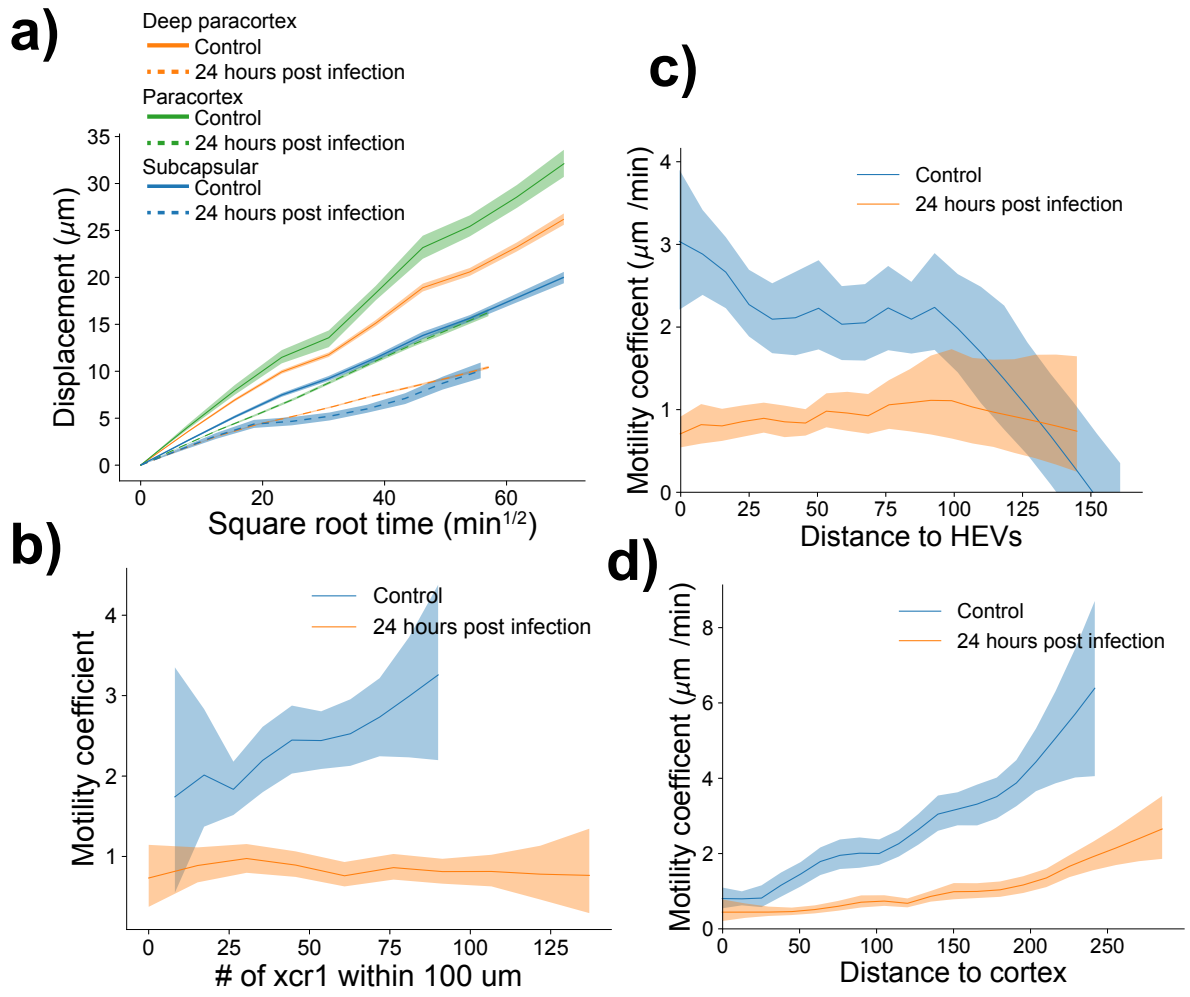

Figure 3.9: **Dendritic cell motility changes in different anatomical locations** a) Mean displacement vs. square root time plots for dendritic cells in different parts of the lymph node at 0 and 24 hours. b) Mean dendritic cell motility coefficients vs the number of other dendritic cells within 100 μm. c) Mean motility coefficient vs. distance to high endothelial venules. d) Mean motility coefficient vs. distance to cortex. Shaded regions in all plots represent bootstrapped 95% confidence intervals.

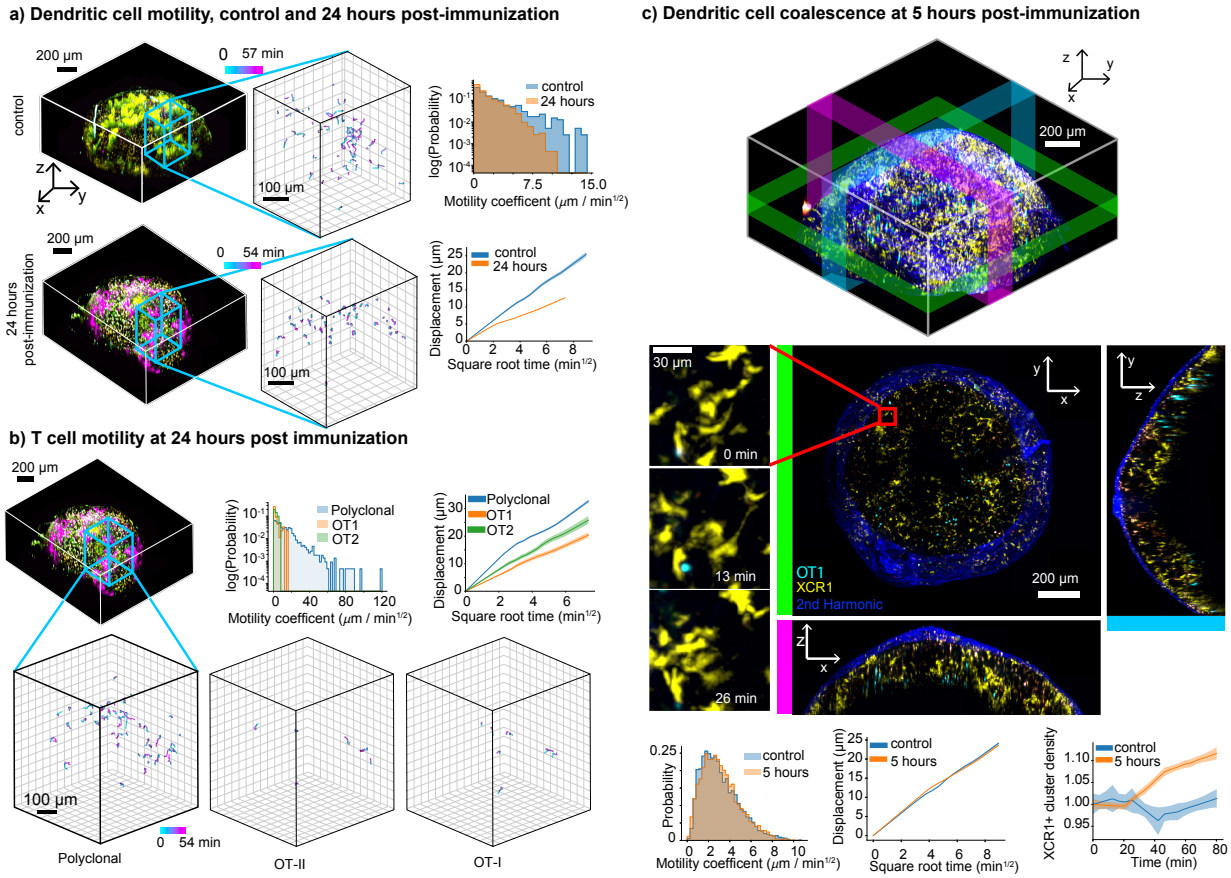

Figure 3.10: **Immune response under physiological conditions.** a) Distinct changes in global behavior of antigen presenting cells as measured by XCR1+ dendritic cell motility 24 hours after immunization show the cell behavioral correlates of developing immune responses. (Left) Tracks of motility in control and 24-hour post immunization, (right top) log histograms of motility coefficients, and (right bottom) displacement vs square root of time show that dendritic cells switch from faster random walk behavior in the control (i.e. straight line in bottom right plot) to slower, confined motility 24 hours post immunization. b) T cell motility at 24 hours post-immunization. (Top middle) log histograms of OT1, OT2, polyclonal T cells. (Top right) Displacement vs square root of time plots. (Bottom) tracks of T cell motility. c) dendritic cell clustering can be visualized and quantified on the whole lymph node level. (Top) 3D view with colored bars marking areas shown in 2D projections below. XY, YZ, XZ projections with zoomed-in area show an example of dendritic cell cluster forming over 26 minutes. (Bottom) Histograms of dendritic cell motility at 5 hours post-infection vs control, mean displacement vs square root of time, mean normalized density over time in 5 hours post-infection vs control dataset show that formation of dendritic cell clusters can be detected on the timescale of 1 hour, but without any detectable change in dendritic cell motility. Error bars on all plots show 95% confidence intervals.

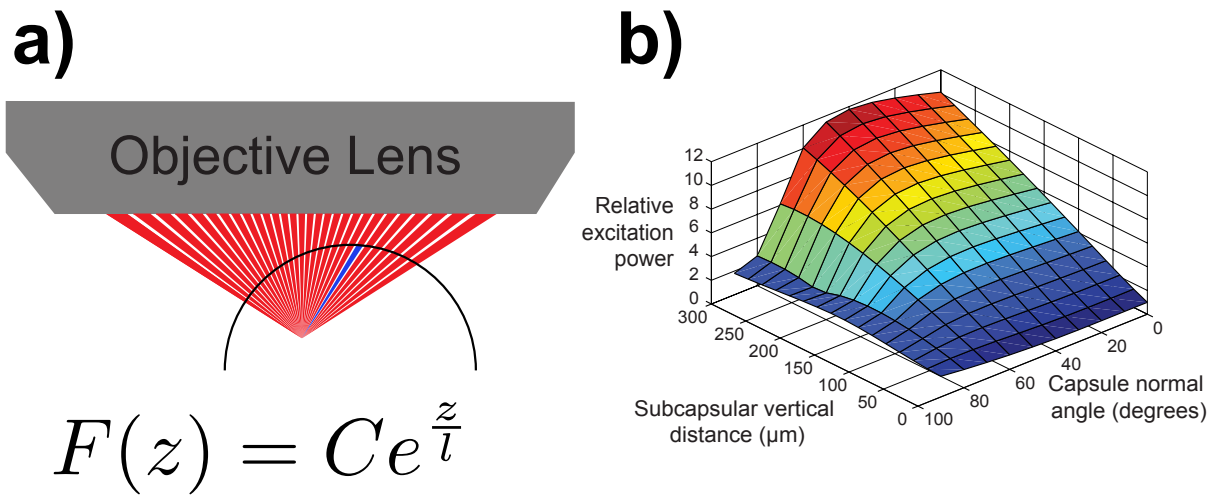

Figure 3.11: **Spherical tissue ray-optics scattering model.** A previous scattering model used on the way to developing standard candle calibration. In this model, the tissue is assumed to be a sphere with homogeneous scattering potential. a) Fluorescence at the focal point is computed by integrating the contribution from every ray within the cone of the objective's numerical aperture. The contribution of each ray drops off with its propagation distance through tissue (z) as shown in the equation. b) The predictions of the model with parameters estimated for lymph node tissue. Relative excitation power is the inverse of the fraction of input power that makes it to the focal point. It is indexed by the vertical distance from the focal point to the top of the tissue, and the normal angle of the sphere directly above the focal point.

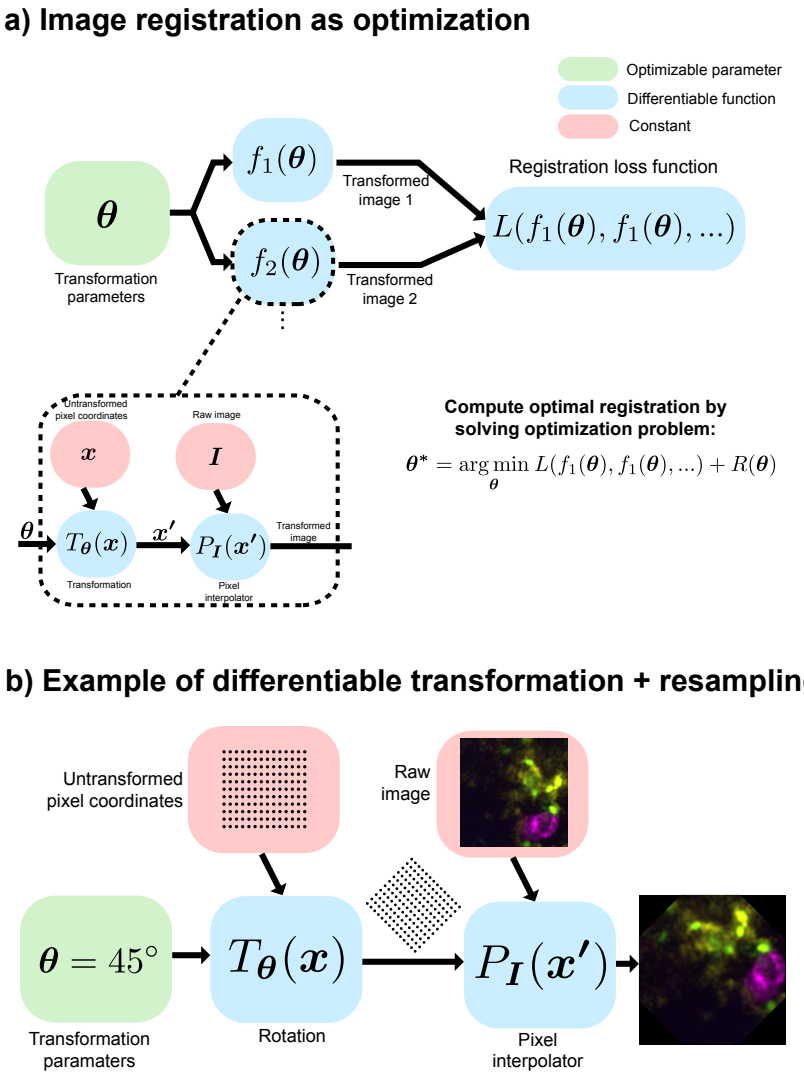

Figure 3.12: **Image registration formulated as iterative optimization**

a) Overview of *maximum a posteriori* estimation for image registration. Differentiable transformations specific to each correction are used to resample raw image pixels, and fed into a loss function that quantified the quality of solution. b) Rotation as an example of a differentiable transformation.

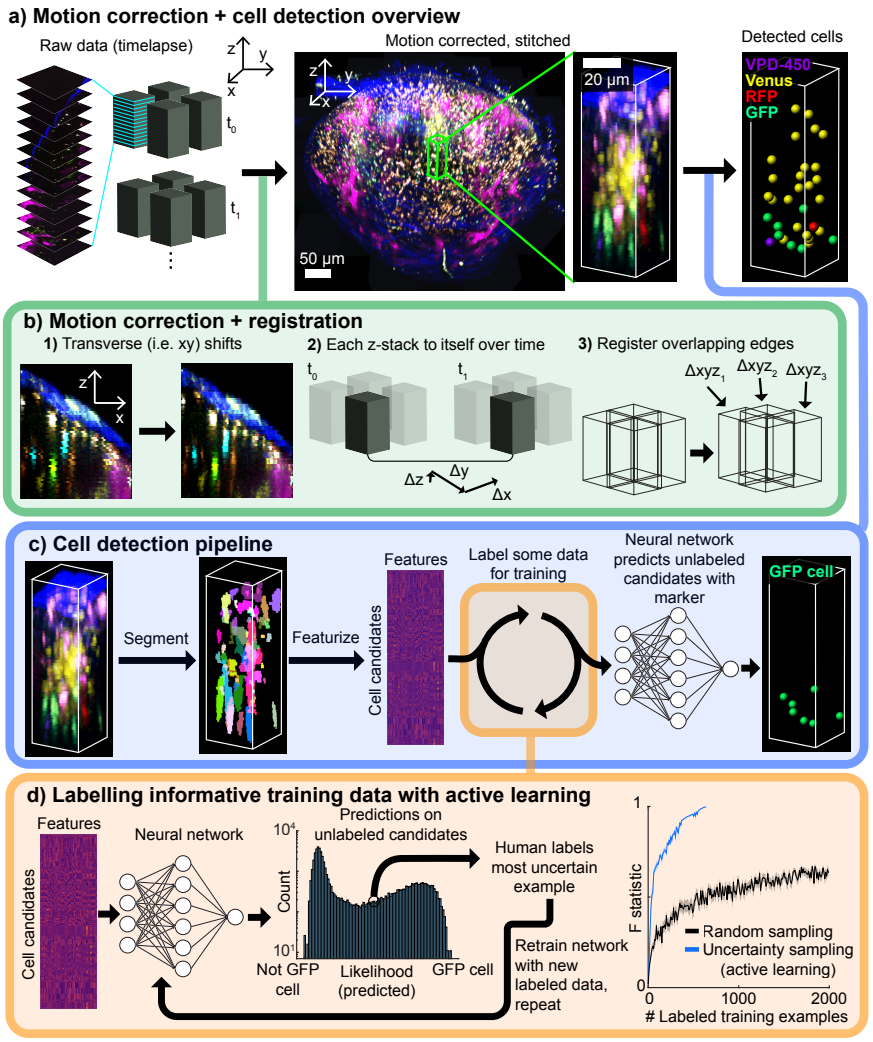

Figure 3.13: **Motion artifact correction and active learning-based cell detection** a) Overview of data processing converting raw data of separate z-stacks into a single stitched and motion-corrected volume, followed detecting cells based on individual fluorescent protein expression. b) Motion correction and registration consisted of three types of corrections: 1) the XY movements within each slice were optimized. 2) Stacks at consecutive timepoints were registered to one another using cross correlation. 3) The alignment between stacks was optimized. c) Cell identification began by computing a 3D segmentation algorithm to identify candidate cells. Features were then computed for each candidate cell and fed into a classification neural network that predicts which candidates belong to the population of interest d) Active learning was used to label an informative training set. In this paradigm the classification network outputs a measure of certainty that each candidate is a cell or not. The most uncertain of these examples is selected for human labelling, the classification network is retrained, and the procedure is repeated. This enables selection of which candidates belong to population of interest (e.g. GFP). Right, active learning data labelling dramatically boosts classifier accuracy compared to randomly sampling and labelling data.

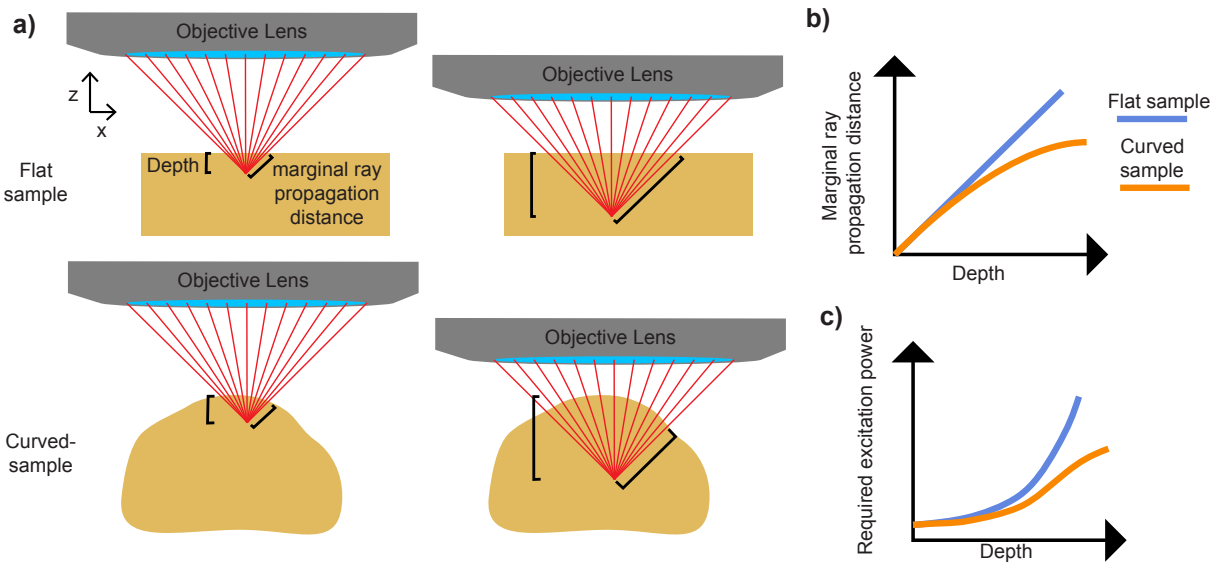

Figure 3.14: **Curved samples require sub-exponential power increases with depth.** a, b) When focusing into a flat sample, the distance from the focal point to the top of the sample ("depth") and the distance travelled by the marginal ray through the sample increase linearly, in a curved sample, the distance travelled by the marginal ray increases sub-linearly. c) As a result, the curved sample requires sub-exponential increases in laser power to maintain signal with depth.

Chapter 4

Microscope control software

The previous two chapters presented new adaptive microscopy techniques, in which images are collected and processed in real time to control parameters of the imaging system. Such experiments rely on a large stack of hardware control code, and although the techniques are in quite different application spaces, much of the code can be reused between techniques.

These examples represent a larger trend, in which microscopy experiments are increasingly growing to larger scales and becoming integrated with post-processing and image analysis strategies. This creates the critical need for software that is: 1) user-friendly, so that researchers can rapidly experiment with and apply new techniques, 2) abstracted from details of underlying hardware, so that new techniques can be easily shared and validated across microscopes. Many researchers already rely on the open-source μManager software for acquiring data, largely because of its ability to control a breadth of different hardware through a common interface. However, μManager is written in C++ and Java, presenting a barrier to combining its many strengths with Python, on which many fields have come to rely as the basis of a consistent, open ecosystem for data analysis.

Here, we present Pycro-Manager, a tool that bridges this gap and provides access to the wealth of existing tools within μManager through Python, while also creating a new, high-level programming interface to create complex experiments with concise, readable code. This interface both facilitates rapid development of new microscopy techniques and provides a way to communicate and share these developments across hardware platforms.

4.1 Overview

Cutting-edge innovations in biological microscopy are increasingly blurring the line between data acquisition and data analysis. Often, the captured data requires significant image processing to produce the final image, and users rely on pipelines of multiple programs or software libraries to get from the capture stage to the final image [16, 138]. Computational microscopy and machine learning-based methods take this paradigm to an extreme, often producing raw measurements that are not human-interpretable without post-processing [11,

47, 108, 73, 62]. Furthermore, new data-adaptive imaging methods rely on data processing during acquisition to actively control various hardware settings of the microscope [30].

Testing new ideas and applying them for biological discovery is often impeded by a lack of control software that is capable of meeting demands for speed and performance, integrating new and diverse types of hardware, providing the flexibility to adapt in real time to the data being captured, providing the flexibility to adapt in real time to the data being captured, and providing user-friendly programming interfaces. As a result, researchers often end up developing bespoke software that works only with specific instruments, using closed-source and/or proprietary programming languages. In addition to slowing development, this lack of a consistent software ecosystem creates barriers to reproduction and replication of new techniques. These include the increased likelihood of bugs in code that is not re-used and tested in different contexts, the burden on new users of understanding new code bases, and the extra work of disentangling high-level code that is not properly abstracted from the underlying hardware.

In many fields the Python programming language, in particular the NumPy/SciPy ecosystem [49, 163], has emerged as a common framework for reproducing research results that rely upon complex, multi-layer scientific workflows in a portable, scriptable form [121] that is accessible to researchers with various levels of programming experience.

The central challenge of such a consolidated approach in microscopy is diversity of hardware used in different types of microscopes, ranging from custom-built components on optical tables to turnkey commercial systems. μManager [28, 29] is an essential tool for controlling microscopes, thanks to an extensive library of 'device adapters' for controlling different types of hardware, from cameras to complete microscopes, from a single programming interface. Community contributions of device adapters, plugins, and scripts provide hundreds of developer-years of microscopy automation.

Despite the power of these libraries, which are written in C++ and Java, they are often difficult to integrate with the latest developments in computer vision and scientific computing, which most readily interface with NumPy. This not only increases the difficulty of developing techniques that rely on both customized data capture and data analysis, but also hinders the dissemination and adoption of these techniques by fragmenting them across multiple tools/programming languages and making them harder to understand and test in new contexts.

To address these needs, we present here Pycro-Manager. Pycro-Manager is built upon a layer that uses ZeroMQ (`https://zeromq.org/`), an open-source messaging library for transporting instructions and data between processes within a single physical machine or between machines on network (**Fig. 4.1a**). A ZeroMQ server running within the Java layer of Micro-Manager provides access to Java objects, functions, and data into language-agnostic messages, and a ZeroMQ client in Python dynamically translates these messages into Python objects/functions and NumPy arrays. As a result, the existing capabilities of μManager can be called as if they had been written in Python, without users having to learn Java, Java developers having to learn Python, or the abandonment of the relative strengths of either language (**Supplementary materials**).

In addition, Pycro-Manager provides a new library of high-level programmatic building blocks (in Python) for customizing data acquisition. These are designed to facilitate creation of complex data acquisitions with concise, readable code, while still being extensible enough to be customized and combined to enable new types of experiments.

The high-level API contains three important abstractions: *acquisition events*, *acquisition hooks*, and *image processors* (**Fig. 4.1b,c**). These building blocks can be used individually or combined for more complex applications. Each is designed to be accessed with minimal, readable code (**Fig. 4.1c)**. Acquisition events enable customized instructions of how to adjust hardware when acquiring images and how to index those images along arbitrary axes (e.g. time, z, etc.) for storage and display. For common microscopy workflows like timelapses, z-stacks, etc., events can be automatically generated by high-level functions (**Fig. 4.1c**). Alternatively, they can be created manually from nested Python lists and dictionaries to allow for a greater degree of customization. Acquisition hooks enable the execution of arbitrary Python code concurrently at various stages of the data acquisition process. Image processors give access to the image data as soon as it is acquired, for modification or for use in data-driven feedback loops on the acquisition process.

In order to keep up with the data rates of techniques such as light sheet microscopy, Pycro-Manager is designed with support for large datasets with fast writing speeds in mind. With appropriate hardware (e.g. NVMe solid state drives and RAID arrays), the current version has been tested at speeds of up to 1.2 GB/s, with dataset sizes on the order of several TB. However, these speeds are only supported when saving directly to disk. The Java-Python transport layer currently has a maximum throughput of about 100 MB/s, placing an upper limit on speed for acquisitions that use *image processors*. Both Java and Python readers are available for reading the data saved by Pycro-Manager.

The combination of Jupyter notebooks (`https://jupyter.org/`), interactive documents that consolidate text, code, equations, and results, and the Pycro-Manager high-level application programming interface (API) provides a means to describe the full workflow of a research project, from data acquisition to analysis to results, thereby facilitating understanding, dissemination, and reproduction of new microscopy technologies. For techniques that rely heavily on computation, these notebooks can be as (or even more) valuable than their corresponding research papers.

In the supplementary information and online documentation (`https://pycro-manager.readthedocs.io/en/latest/#`), we provide a wealth of tutorials describing the basic features of Pycro-Manager, as well as Jupyter notebooks outlining several sample applications. A brief overview of these applications follows:

Microscope alignment is often performed by optimizing hardware, guided by real-time feedback about changes in the point spread function or the image Fourier transform. For example, the real time changes in the 3D point spread function of a light sheet microscope can be viewed in visualization tools like Napari (`https://zenodo.org/record/3555620`).

Light sheet microscopes are dependent on having hardware components synchronized to each other with transistor-transistor logic (TTL) triggering and a high-performance pipeline for saving data. Acquisition hooks enable the integration of external timing de-

vices into acquisition workflow, and the performance of Pycro-Manager supports high data throughput applications. These features can be used, for example, to control an oblique plane illumination microscope and collect multi-terabyte high-resolution datasets of cleared tissue [134] or subcellular dynamics [136].

Integrated sample processing and imaging is often desirable for experiments with complex workflows using microfluidics. Cyclic immunofluorescence imaging [84] is one such example where tissue sections are imaged and stained with fluorophore-conjugated antibodies in multiple rounds, with Pycro-Manager enabling a synchronized, automated control through Python of multiple rounds of staining using microfluidics and image acquisition.

Computational microscopy and machine learning often require processing of intermediate images to get a final result. Image processors can be used for this purpose, enabling applications such as deep learning-based image denoising [73] and solving physics-based inverse problems to transform defocused images into quantitative phase measurements [62].

In the case of adaptive microscopy, the acquisition process itself can be controlled based on feedback from acquired images [30]. Various features of Pycro-Manager can combined to enable this. We provide examples such as image-guided, closed-loop optogenetic control [40], targeted multi-contrast pathology imaging guided by a neural network [80], single-shot image-based autofocusing using machine learning [117], and sample-adaptive multiphoton illumination using a combination of machine learning and physical modeling [118].

Microscope control over networks enables groups of networked microscopes or splitting out hardware control and data processing to different physical machines. The language-agnostic ZeroMQ messaging layer on which Pycro-Manager is built opens up the possibility of these applications. ImJoy [104] can be used to take this a step further and build interactive, browser-based plugins that can be easily deployed to others.

In summary, Pycro-Manager provides a much needed interface between more than a decade's worth of open-source microscopy development and the cutting edge of scientific computing in Python. It frees researchers from many of the burdens of developing low-level code and thereby enables rapid experimentation with new techniques. Finally, it enables concise, readable code for instrument control that is maximally abstracted from specific hardware, facilitating the understanding, dissemination, and reproduction of new techniques.

The source code and documentation for Pycro-Manager can be found at: `https://github.com/micro-manager/pycro-manager` and: `https://pycro-manager.readthedocs.io/en/latest/#`

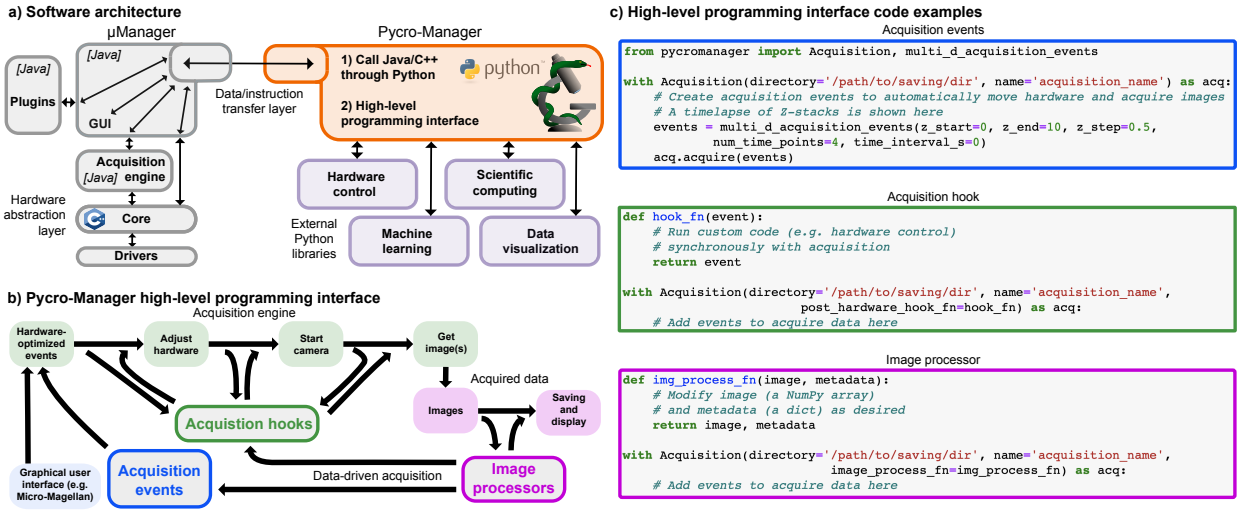

Figure 4.1: **a) Software architecture overview.** (Grey) The existing parts of μManager provide generic microscope control abstracted from specific hardware, a graphical user interface (GUI), a Java plugin interface, and an acquisition engine, which automates various aspects of data collection. (Orange) Pycro-Manager enables access to these components through Python over a network-compatible transport layer, as well as a concise, high-level programming interface for acquiring data. These provide integration of data acquisition with (purple) Python libraries for hardware control, data visualization, scientific computing, etc. **b) Pycro-Manager's high-level programming interface.** The data acquisition process in Pycro-Manager starts with (blue) a source of acquisition events (from either a programming or GUI). These events are passed to (green) the acquisition engine, which optimizes them to take advantage of hardware triggering where available, sends instructions to hardware, and acquires images. (Magenta) The resulting images are then saved and displayed in the GUI. The three main abstractions of the Pycro-Manager high-level programming interface (acquisition events, acquisition hooks, and image processors) enable fine-grained control and customization of this process. c) **Code examples.** Code snippets for implementing (blue) acquisition events, (green) acquisition hooks, and (magenta) image processors.

4.2 History and motivation

Over the past 15 years, μManager [28, 29] has emerged as a powerful software tool for microscope control. It contains a hardware abstraction layer that puts hundreds of hardware components from many different manufacturers under a common control interface. Its adoption has been driven primarily by two sub-communities: 1) end-users who need a familiar software for controlling various microscopes, some of which mix and match available equipment and 2) developers who make use of the APIs within μManager to create control abilities for customized experiments/applications. The former category is agnostic to the programming language in which μManager is implemented, because they are interacting only with its GUI. For the latter category, the underlying language is an important consideration, as it dictates how customized capabilities are programmed, and how easily external libraries can be integrated.

In the 15 years of μManager's existence, there has been a significant shift towards the Python programming language in the broader scientific community. In particular, the scientific Python ecosystem [163] has dramatically increased the ease and speed of implementing complex data analysis, increasingly supplanting proprietary languages like MATLAB [121], thanks in part to a wealth of community contributions of data analysis routines and the inherent advantages of working with Python, a free, open-source, fully-featured programming language. These trends make a Python interface for μManager a useful feature for the microscope developer community.

A Python interface for the low-level components of μManager (i.e. the core) already existed before the development of Pycro-manager, but it had significant limitations. This interface was generated by compiling bindings for the μManager core (written in C++) for Python (rather than Java). This provides access to the the low-level hardware control capabilities of μManager through Python. However, Python bindings and Java binding are mutually exclusive on the same instantiation of the μManager core. Thus, using these Python bindings precludes the use of μManager's higher-level Java libraries, which includes capabilities such as the GUI, the acquisition engine, and data saving code, which are essential for many experiments.

This naturally raises the question of whether it would make sense to recreate the capabilities that are currently in the μManager Java layer in Python. We opted against this, for two reasons. First, an enormous amount developer time and community testing went into creating the current μManager Java layer. Re-creating this would not be a small task with debatable pay-off as there is no guarantee that Python would not be supplanted by a new language (e.g. Julia) as the top choice of researchers down the line. Second, for all its functionality and success in data analysis, Python has several disadvantages compared to Java when used in a complex, multithreaded software like Micro-Manager.

Multi-threading. One weakness of Python in multi-threaded applications compared to Java results from its Global Interpreter Lock (GIL). The GIL prevents calls on different threads from simultaneously accessing their C implementations. This precludes a guarantee that multi-threaded applications will actually make use of multiple CPU cores, which can lead

to unexpected results like multi-threaded tasks taking longer than single threaded versions. Some libraries, like NumPy [160], explicitly release the GIL to prevent this difficulty, but this behavior varies on a library-by-library basis. Python attempts to work around this by providing the "multiprocessing" module, where simultaneous computations on different cores can take place by exposing multiple processes through a thread-like interface, but this too has drawbacks. For example, this module is not cross-platform since threads behave differently Windows and POSIX systems. There is also the overhead, both computationally and from the perspective of a developer, of having to specifically define how objects must be serialized when passing across processes.

Portability. Java's use of the Java Virtual Machine makes it much easier to write code that can easily be deployed across multiple platforms. Unlike Python/C/C++, developers don't need to worry about differences of how different platforms handle threading and memory management.

Speed. Java strikes a nice balance between execution speed and developer speed. In our experience most developers can code and debug significantly faster with Java than C/C++ and Java is significantly higher performance than Python.

Usage. Despite Python's continued ascension, Java remains in the top 2 or 3 of most popular programming languages (`https://pypl.github.io/PYPL.html`, `https://www.tiobe.com/tiobe-index/`). Developing purely in Python/C would substantially limit a large number of developers' ability to contribute.

For all these reasons, we opted to create a Python interface to the high level capabilities of μManager that made use of Java libraries when possible. This undertaking required two things. First, a transport layer that enables cross-language communication between Java and Python. Second, versions of the major features in the Java layer of μManager (specifically, a multi-dimensional image viewer, data saving library, and acquisition engine) that are amenable to inter-operability with external customization.

4.3 Architecture and design

Limitations

Pycro-Manager inherits its platform and hardware limitations from Micro-Manager. Since Java/Python/C++ can all be run on Windows/OSX/Linux, the full stack is cross platform. The limitation comes from which hardware can be controlled on which OS which is itself limited by which platforms hardware makers create drivers for. Windows is almost often the most widely supported. If drivers are made available for a certain platform, a Micro-Manager device adapter can be written that allows it to be controlled. Micro-Manager supports a wide variety of cameras and all kinds of motorized components, but support for scanning-based modalities (i.e. confocal microscopes) is much more limited (though it is certainly possible). Multiple layers of abstraction separate Pycro-Manager from low-level devices it controls (i.e. acquisition engine, Micro-Manager core, device adapters). It would thus be

possible to plug in a new and improved device layer in the future, with minimal changes required in Pycro-Manager itself.

Since Pycro-Manager is built upon a translation layer that passes data and instructions between Java and Python, which run in separate processes, there is some inherent performance overhead. Currently, the 100 MB/s data transfer rate may be limiting for some applications, and users may choose to do data processing in Java. Data processing libraries are more limited in Java. In this way, Pycro-Manager finds a balance between micro-manager hardware control and modern data analysis tools.

Java-Python transport layer design

Pycro-Manager is built upon a layer that uses ZeroMQ (`https://zeromq.org/`), an open-source messaging library for transporting instructions and data between processes within a single physical machine or between machines on network (**Fig. 1a in main text**). A ZeroMQ server running within the Java layer of Micro-Manager provides access to Java objects, functions, and data in the form of language-agnostic messages, and a ZeroMQ client in Python dynamically translates these messages into Python objects/functions and NumPy arrays.

The need to call Python code from Java is not a new problem. Indeed, Py4J (`https://www.py4j.org/`), an existing library for this task, already enables much of this functionality. However, a major limitation of this library is its speed in transferring large blocks of data, which is currently 10 MB/s, and thus would introduce significant lags in workflows that require passing large blocks of data (such as images) from one language to the other. In light of this, we designed a novel transport layer that uses ZeroMQ as a language-agnostic communications library, thereby realizing a 20x increase in speed. This was inspired by a similar strategy used in JuPyter notebooks [68]. This enables communication between Java and Python code running in separate processes.

The root of this transport layer exists within a server that runs on the Java side. This server is started automatically with μManager after a specific option is checked. On the Python side, a process can be started that connects to this server as a client. The client can then specify the Java classpath of a class to either instantiate or to get an existing instance. This allows new Java objects to be constructed from the Python side and then passed over by reference, or references to some special existing objects (e.g. the Java-wrapped μManager core) to be passed across. Before passing over the transport layer, a reference is stored on the Java side, so that it will not be garbage collected by Java. Next, Java's Reflection API is used to get information about the object's fields and methods (e.g. names, classes, arguments, return types, etc.), which are serialized into JavaScript Object Notation (JSON), an open, human-readable data interchange format.

For example, consider the following Java class:

```
package org.micromanager.some.classpath
```

```
public class TestJavaClass {

   private int iVal;
   private String sVal;

   public TestJavaClass(int i, String s) {
      iVal = i;
      sVal = s
   }

   private int getI() {
      return iVal;
   }

   private void setS(String newS) {
      sVal = newS;
   }
}
```

To construct a new instance of this object, the following message is sent from the Python side of the transport layer:

```
{
    "command": "constructor",
    "classpath": "org.micromanager.some.classpath",
    "argument-types": ["int", "java.lang.String"],
    "arguments": [2, "some_string"]
}
```

This message results in the creation of a new Java object. Information about that object is then sent back over the bridge

```
{
    "hash-code": "1e55f1c690a430da-f351-42d6-8516-15258c49c177", // Unique ID
       for this reference
    "interfaces": // Superclasses and interfaces implemented by this object
       ["org.micromanager.some.classpath","java.lang.Object"],
    "port": 4827, //The port over which the client-server pair operate
    "api": // public methods of the object
       [{"name": "getI", "arguments": [], "return-type": "int"},
        {"name": "setS", "arguments": ["java.lang.String"], "return-type":
           "void"},
        // Additional methods of java.lang.Object removed for clarity
       ],
    "type": // Whether it's an object passed by reference or a primitive
```

```
        passed by value
        "unserialized-object",
    "fields": ["sVal"], // Names of public fields of the object
    "class": "org.micromanager.some.classpath.TestJavaClass" // The object's
        class
}
```

Upon receipt on the Python side, this JSON string is deserialized and an instance of a Python object is dynamically created. This Python object has methods and fields that have the same names as the Java methods and fields. Any time the value of a field is get or set, or a method is invoked, a new message is sent to the Java side with instructions on what action to take on the shadowed Java object. When all references to the Python object have disappeared, and it is garbage collected on the Python side, a message is sent to the Java side to release the reference to the Java side so that it too can be garbage collected if appropriate. In this way, Java code can be called as if it were written in Python.

Acquisition engine, image viewer, data storage

The transport layer enables full usage of μManager's existing APIs in Python. However, there remain several desirable features not present in the Micro-Manager Java libraries: μManager's file formats are limited to four defined dimensions: channel, z, time, and position and don't support multi-resolution pyramid saving. Furthermore there are readers for the files they write in languages other than Java. The μManager viewer similarly does not support arbitrary dimensions or fast zooming by taking advantage of multi-resolution data sources. Furthermore, its integration with the ImageJ [139] viewer makes it difficult to implement custom interactive interfaces and limits the displayed data to the ImageJ data model. The μManager acquisition engine includes few possibilities for extensions/changes, and its code is difficult to modify and maintain since it was written in a programming language (Clojure) used by few scientists/programmers.

Because of these limitations, we created three libraries corresponding to three logical components: a multidimensional image viewer ("NDViewer"), a multipage tiff storage class with support for multi-resolution pyramids ("NDTiffStorage"), and an acquisition engine ("AcqEngJ"). The former two used pieces of code from μManager 1.4), Micro-Magellan, and ImageJ, while the latter took inspiration from the analogous component in μManager 1.4), but was written from scratch to enable a multitude of new features.

NDViewer. Real time visualization of acquired data is a crucial component of setting up and performing most microscopy experiments. The μManager 1.4 and 2.0 viewers are built on top of or tightly integrated with the image viewer of ImageJ. This gives the advantage that ImageJ tools can be used directly on the data in the μManager viewer. However, it requires further engineering to achieve good performance, since ImageJ was not designed with real time acquisitions in mind. It also introduces the limitations that 1) it is difficult to create customized interactive layers on top of the viewer without colliding with ImageJ code

in unexpected ways and 2) the data being displayed must fit into the ImageJ data model.

For these reasons, we developed NDViewer separately from ImageJ, taking inspiration and bits of code from μManager 1.4/2.0 and ImageJ. NDViewer makes minimal assumptions on the format of the data–it can have multiple channels, an arbitrary number of named axes of any size, and the size of the source data is dynamic (e.g. could be changed as more data is added). Missing data along different axes are permitted, and data can arrive in any order. It also supports fast visualization at multiple scales with a mutli-resolution pyramid data backend. It supports the addition of custom control panels and overlays. The source code can be found at: https://github.com/micro-manager/NDViewer.

NDTiffStorage. Flexible and performant data writing are key to enabling the widest range of microscopy workflows possible. A fork of the μManager multipage TIFF storage format was used as the basis for the saving format for Micro-Magellan [119] and has since continued to evolve into its own distinct library (used by both Micro-Magellan and Pycro-Manager). Its evolution has been driven by the need to maximize flexibility while maintaining data writing performance for microscopy applications with high data thorughput (¿1 GB/s) as well as the ability to seamlessly handle multi-TB datasets. This evolution began with the ability to support multi-resolution pyramids, which was accomplished by creating a different multipage tiff dataset for each resolution level of the pyramid. It has since added support for arbitrary N-dimensional named data axes (i.e. not just channel, z, time, position) in any order, without the need to know which axes are present at the start of the experiment. The source code and the file format specification can be found at: https://github.com/micro-manager/NDTiffStorage.

AcqEngJ. An acquisition engine is an important component of a complex microscopy acquisition where multiple hardware components need to be synchronized. Its function is to take in high level instructions about the desired type of data acquisition and transform these into a sequence of low level hardware instructions, optimized for the specific hardware components being used. This protocol is then executed, and acquired data are dispatched to downstream components such as data visualization and saving.

AcqEngJ takes as input a series of "acquisition events". Each event contains the instructions to acquire an image and/or to adjust one or more hardware components. There is an implicitly defined syntax for converting these events to/from JSON, such that they can be easily produced from a variety of sources (e.g. Java, Python, a text file). An example acquisition event is shown below (see Pycro-Manager documentation for a full and up to date version):

```
{
   // A dictionary with the positions along various axes (e.g. time point
      index, z-slice
   // index, etc) a 'channel' axis is not required as it is inferred
      automatically
   'axes': {'z': integer_value, 'time': integer_value2},
```

```
    // The channel's configuration group and preset
    'channel': {
         'group': 'name_of_micro_manager_config_group',
         'config': 'setting_of_micro_manager_config_group'
    },

    'exposure': exposure_time_in_ms,

    // For z stacks
    'z': z_position_in_um,
}
```

While acquisition events describe *what* to acquire, they do not specify *how* to acquire it. Upon receiving a list of acquisition events, AcqEngJ will parse them to create a new set of instructions optimized for the specific hardware being used. This process consists primarily of implementing hardware triggering wherever applicable using the μManager core's sequencing API.

Hardware triggering can dramatically increase the speed of acquisition by replacing communication between the computer and the hardware in between the acquisition of each image with TTL triggering between different hardware components. For example, when collecting a Z-stack, hardware components are loaded with sequences of instructions (e.g. physical positions on a stage or a sequence of exposures on a camera). The sequence can then be executed independently of the computer, except for frames being read off a camera as fast as possible. There are more complex and potentially useful capabilities such as explicitly accounting for the different timings of different components, though at present these are not implemented in AcqEngJ.

Finally, since new and customized microscopy setups often depart from conventional acquisition workflows, it is extremely useful to be able to modify different parts of the acquisition with custom code and data/metadata processing routines. This capability is provided by Pycro-Manager's acquisition hook and image processor functionality. Thus, AcqEngJ allows execution of arbitrary code at different points in the acquisition cycle, and addition/deletion/modification of acquired images before they are sent to downstream data storage/visualization.

The Python Acquisition class

Since AcqEngJ, NDViewer, NDTiffStorage, and the Java-Python transport layer are each separate libraries that do not depend on one another, there remains one final component to glue them all together and expose them to the Python side. On the java side, the **org.micromanager.remote** package does this work. On the Python side, the **Acquisition** class exposes a simple Python interface for the various features of of the three libraries. To do so, it communicates with a small number of Java classes in **org.micromanager.remote**. Together, these packages handle the creation of acquisitions in AcqEngJ, the addition of

acquisition hooks and image processors as appropriate, and the ZeroMQ sockets to handle passing relevant objects back and forth between Java and Python. More information about the **Acquisition** class can be found in the Pycro-Manager documentation.

Chapter 5

A benchmark dataset for single-cell computational microscopy

Like the robust, user-friendly software for microscope control described in the previous chapter, standardized benchmark datasets are an important tool for accelerating data-driven microscopy. This chapter addresses that need; we create the Berkeley Single Cell Computational Microscopy (BSCCM) dataset, which contains over 400,000 images of single white blood cells imaged under various illumination conditions on an LED array microscope. These images are also paired with more conventional cell type markers: immunofluorescence measurements of surface proteins and images of histology staining of cells. We hope this dataset will provide a valuable resource for the development and testing of new algorithms in computational microscopy and computer vision with practical biological applications.

5.1 Introduction

In computational microscopy, an image is formed through the action of not only physical optics such as lenses, but also algorithms that process the raw measurements recorded by the physical system. Generally, the algorithmic side of such systems utilize some knowledge of the physics of the image formation process. The resultant images are thus a function of both the raw measurements fed into them and the computational model of the system's physics. Compared to conventional microscopy techniques, computational microscopy has the potential to create imaging systems that utilize less expensive hardware, capture information more effectively, and provide more robust, quantitative images.

As the number and variety of algorithmic image formation techniques continues to grow, standardized performance benchmarks are essential for both guiding new research and understanding the utility of new techniques. This need has been compounded by the fact that, like many other fields that rely on image processing, the past decade has seen an explosion of techniques that use data-driven machine learning methods, the most prominent example being convolutional neural networks [75] . Unlike traditional physics-based algorithms,

which often possess some degree of physical interpretabiliy, vanilla neural networks operate as "black boxes" that can yield images with excellent perceptual quality, but also fail unpredictably without obvious explanations as to why [151].

Since data-driven machine learning models depend heavily on the data fed into them, developing and benchmarking new algorithms depends upon the existence of easily-accessible benchmark datasets. The benefits of these standarized datasets are two-fold: they allow new advances to be compared against existing approaches without the confounding effects of different data, and they speed the development of new techniques by freeing researchers from the burden of collecting and processing their own data. In the computer vision field, datasets such as MNIST [76], and ImageNet [26], provide simplified, easily understood problems (handwritten digit and natural image classification, respectively), and algorithms developed using these datasets have gone on to be quite successful of generalizing to a variety of diverse applications from diagnosing skin cancer [31] to mapping poverty from satellite imagery [3].

Unlike most computer vision techniques, computational microscopy encompasses the design of imaging systems, in addition to the processing of the images they produce. This often requires formulating models of the imaging system's physics and/or acquiring additional calibration data. For example, deconvolution algorithms usually require knowledge of the system's point spread function (PSF); Diffraction tomography, in which a 3D image of a sample is reconstructed from 2D projections, requires a physical model of how those projections were formed. It is in some cases possible to learn these extra parameters by estimating them alongside the image reconstruction (i.e. blind deconvolution [77] or illumination angle learning [27]). However, the development of such self-calibration algorithms would similarly benefit from reference datasets with available ground truth.

Another important consideration is what metrics should be used to characterize a "good" algorithm. There are multiple potentially useful performance characteristics. These include generating an image with high perceptual quality, performance on a downstream prediction task, or the distortion compared to a known or simulated ground truth. Often, researchers will use a USAF resolution target or a fabricated 3D object with known physical properties as ground truth [183]. While useful as benchmarks, these are arguably quite different from the ultimate applications of many computational microscopy algorithms. These are especially ill-suited to techniques that optimize imaging systems for a particular task or type of sample [57, 64].

To address this need, we created the Berkeley Single Cell Computational Microscopy (BSCCM) dataset, which 1) is large enough to be used with contemporary data-hungry neural networks 2) has multiple orthogonal readouts for assessing algorithmic performance 3) contains the structured metadata and calibration required by computational microscopy algorithms 4) is applicable to real-world biological applications. The dataset contains $400,000$ images of single white blood cells taken with an LED-array microscope [181, 111], as well as paired measurements of the same cells with the two traditional measurements used in phenotyping assays: histology staining and surface protein readout from fluorophore-conjugated antibodies.

The outline of this chapter is as follows: Section 5.2 gives relevant background about

LED-array microscopy and single-cell phenotyping. Section 5.3 discusses specifics of the different versions of the dataset and what they contain, with the aim of providing all the information required to get started using the dataset. Subsequent sections may provide additional insight about how the data was collected, which may be useful for advanced use-cases or replication. Section 5.4 describes how the data was collected and processed. Section 5.5 gives an overview of the underlying organization of the data itself: how files are organized, what metadata is available, etc. We note that knowledge of the specifics in not needed for users using the high-level wrapper code we provide for accessing the data. Finally, we include in the appendix a Jupyter notebook demonstrating how to access and use this data.

5.2 Background

LED array microscopy

Label-free microscopy uses intrinsic optical properties of cells (such as spatial variations in refractive index) instead exogenous chemical contrast agents (such as light absorbing or emitting dyes that bind to specific molecules). Although label-free techniques have been around for close to a century [179], recently advances have enabled the extraction of quantitative estimates of physical properties, such the phase delay caused by the sample [124] and polarization [102]. As a result, physical properties like the relative protein/lipid content or the polymeric structure of molecules can be spatially resolved [175, 95], providing useful phenotypic information about biological samples.

The LED array microscope [181] offers a simple yet powerful system for label-free imaging of biological samples. It is realized by replacing the illumination lamp on a traditional, transmitted light microscope with an LED array, so that the sample can be illuminated with different angles of light by turning on different LEDs. This can done using a planar LED array or, as in our case, an LED array quasi-dome [111]. The advantage of the latter is that it yields increased intensity at high illumination angles compared to the planar array. Specifically, the intensity of incident light on the sample given an angle θ relative to the optical axis is proportional to $\frac{1}{cos^4\theta}$ for the planar array, whereas the quasi-dome's dropoff with angle is $\frac{1}{cos\theta}$.

By capturing images with different illumination patterns and using physics-based models of the image formation process, a variety of techniques can be implemented. These include multi-contrast imaging [181, 86], the synthesis of high-resolution images from low-resolution inputs [182, 103, 157], estimation of 3D structure from 2D measurements [156, 154, 56, 85], quantitative phase imaging [103, 155], and calibration of important physical parameters of the imaging systems directly from data [18, 27].

Achieving optimal performance with these techniques usually requires the use of pre-processing steps such as shading corrections, in which raw measurements are divided by an image taken on the same system under the same illumination, but without a sample. This

pre-processing accounts for inhomogenous illumination caused by imperfections in the optical system. As a result, it is important for standardized datasets to include the calibration images needed to make such corrections.

One important consideration in choosing illumination patterns is the tradeoff between a single LED (spatially coherent illumination) vs. multiple LEDs at once (partially coherent illumination). The latter minimizes the time and data needed to capture all information, but potentially at the expense of accuracy. Finding optimal illumination strategies remains an active area of research. However, the physics of LED array illumination provides a convenient way to test different illumination strategies: Since each LED is incoherent with all other LEDs, images corresponding to illumination with multiple patterns can be synthesized *in silico* by adding together the corresponding single-LED illuminated images. Though these two aren't perfectly equivalent due to factors such as the camera offset, read noise, etc., they are practically similar enough that computationally optimized illumination in fact does generalize to improved imaging on experimental systems [57, 64].

Single cell biology

Next, we turn to the question of what samples to image. Among potential biological applications, imaging single cells is particularly appealing for three reasons. First, single-cell phenotyping assays already have ubiquitous clinical application. For example, counting the number of different white blood cell types in a blood sample from a patient (peripheral blood leukocyte differential count) is one of most frequently used clinical laboratory tests[14]. Furthermore, cutting edge methods that measure the RNA [66] or protein [65] composition of single cells continue to yield new biological knowledge and clinical applications. Second, label-free computational microscopy has the potential to augment or replace many single cells methods (microscopy-based or otherwise), thanks to its speed, relatively low cost, and non-toxicity to cells. Third, many imaging techniques that can be performed using LED array illumination have already been shown to capture rich, biologically significant information for characterizing cells[166, 170, 17, 32, 176].

Realizing the full potential of label-free computational microscopy for single cell imaging relies upon finding ways to benchmark and then optimize the performance of these methods for biological assays. The majority of single cell assays capture information by measuring the levels of various known molecular markers on single cells (e.g. specific messenger RNAs or proteins). As a result, being able to compare the information captured by a label-free method with the information present in existing assays is an important step.

White blood cells (leukocytes) are an ideal model system for this task. They are clinically significant in a variety of disease processes [14, 53]: changes in the frequencies of these cell types in peripheral blood can be informative about a number of diseases including infections, inflammatory disease, and cancer. They are morphologically-diverse, containing substantial variation in their size, the shape of their nuclei, and their cytoplasmic composition. Finally, there are multiple, ubiquitously-used assays for phenotypically characterizing them [48]: characterization of expression of proteins that serve as markers of cell lineage (usually based

on antibody binding on a flow cytometer [33, 2, 48]) or using a histology stain and manual examination on a brightfield microscope. This redundancy provides multiple means of benchmarking the performance of label-free microscopy.

5.3 Dataset overview

This section gives an overview of the different versions of the BSCCM dataset, with the aim of providing the most important information for getting started.

All data were collected on a Zeiss Axio Observer microscope with its illumination source replaced by a programmable quasi-dome LED array [111] (**Fig. 5.1a**). This microscope was also equipped with an epifluorescence light path consisting of a mercury arc coupled to a 405nm bandpass filter, and 6 switchable dichroic mirror-emission filter combinations. After LED array/fluroescence imaging, a subset of cells was stained with a light absorbing histology stain (Wright's stain) and imaged again using the LED array using red, green, and blue brightfield patterns to produce an RGB image. The LED array and fluorescence images were collected using a 20× 0.5 NA objective lens. The histology images were collected with a 63× 1.4 NA oil immersion lens. **Figure 5.1b** shows an example of the collected data, including LED array, fluorescence, and histology patterns.

Since the spectra of the fluorophores used overlap and bleed through into multiple channels, the fluorescence data was subsequently processed in order to estimate the relative abundance of each antibody's protein target, the measurement of which provide a metric with visible morphological correlates (Fig. 5.1c)

BSCCM, BSCCMNIST, and BSCCM-coherent

Given the large size of the dataset, we split it into two versions: BSCCM, BSCCMNIST, which contain images taken with multi-LED illumination patterns; BSCCM-coherent, which utilized single-LED patterns. The size of the former is further reduced into BSCCMNIST, which is a spatially cropped, downsampled version of BSCCM with image dimensions (28×28) and bits-per-pixel (8) matched to those of the commonly used MNIST dataset of handwritten digits [76]. In addition, corresponding smaller versions of each of these datasets (BSCCM-tiny, BSCCMNIST-tiny, BSCCM-coherent-tiny) with a subset of cells provide easy-to-work-with versions (**Table 5.1**).

In addition to LED array illumination patterns, each dataset also contains fluorescence images under different antibody staining conditions (**Fig. 5.2**). For BSCCM/BSCCMNIST these conditions include either a single antibody at a time, all antibodies together, or no antibodies. BSCCM-coherent contains only all-antibodies and no antibody conditions.

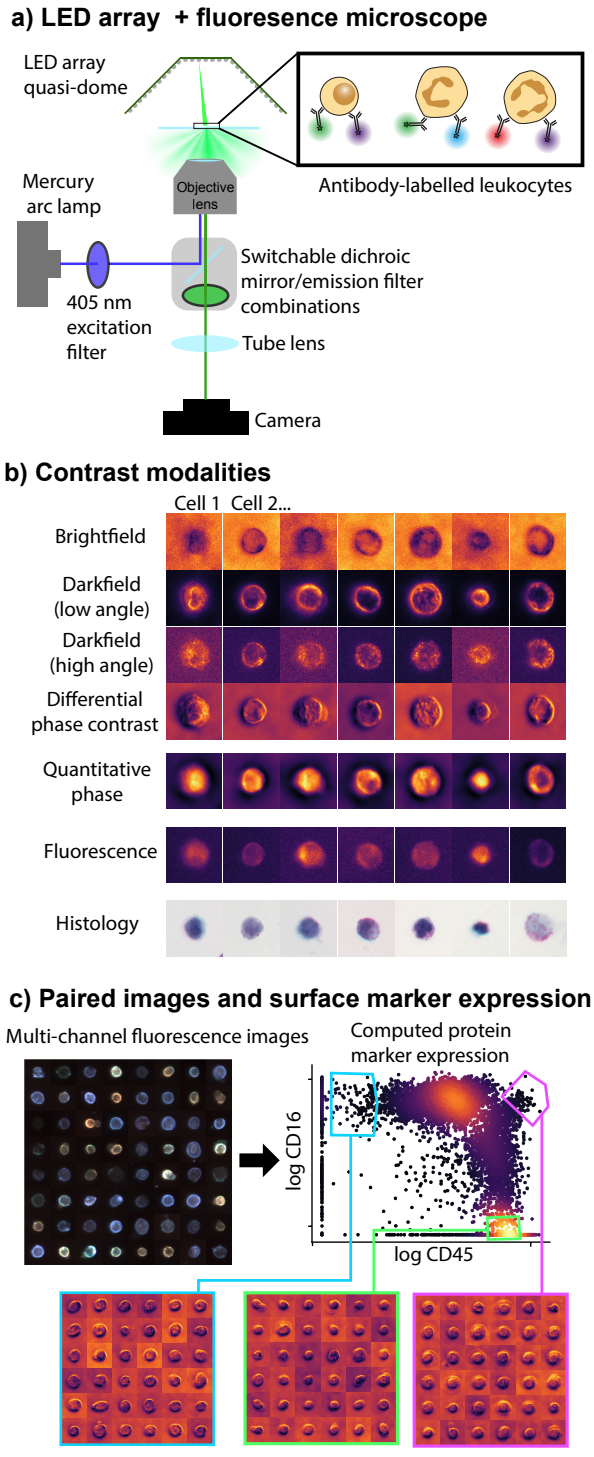

Figure 5.1: **BSCCM overview a)** Schematic of the microscope used in data collection: a commercial microscope with its trans-illumination lamp replaced with a programmable LED array quasi-dome. **b)** Example of the contrast modalities present in the dataset, including LED array illumination, fluorescence, and histology-stained. **c)** Multi-channel fluorescence images were processed to derive the levels of different surface proteins, the levels of which correlate distinct morphological phenotypes

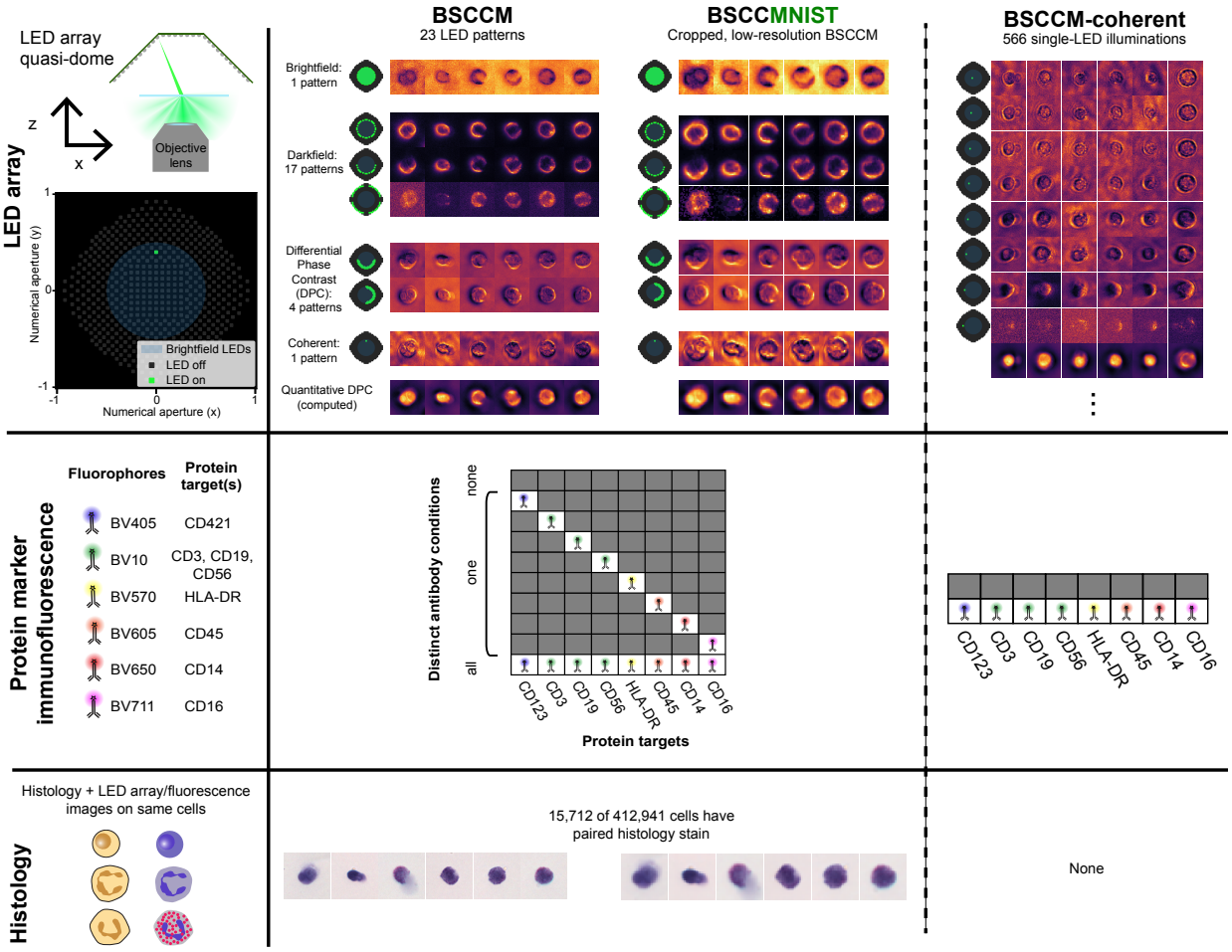

Figure 5.2: **Comparison of BSCCM and BSCCM-coherent datasets (Left)** XY and XZ diagrams of the LED array quasi-dome, full set of fluorescent antibodies and protein targets, and diagram of histology stained cells. **(Middle)** The BSCCM/BSCCM/BSCCMNIST datasets, which includes 23 LED array illumination patterns per each cell, 2 identical batches of slides images with either none, one, and all antibodies, and representative examples of histology images present for a subset of cells. **(Right)** The BSCCM-coherent dataset, which includes 566 single-LED illumination patterns, no antibody and all antibody staining conditions, and no matched histology contrast cells

Name	Size	# cells	# LED patterns	image size (LED-array/fluor, histology)	bits per pixel (LED-array/fluor, histology)
BSCCM	228 GB	412,941	23	128×128, 398×398	12, 36
BSCCMNIST	58 GB	412,941	23	28×28, 28×28	8, 24
BSCCM-coherent	30 GB	4,304	566	128×128, N/A	12, N/A
BSCCM-tiny	576 MB	1000	23	128×128, 398×398	12, 36
BSCCMNIST-tiny	1.4 GB	1000	23	28×28, 28×28	8, 24
BSCCM-coherent-tiny	58 MB	10	566	128×128, N/A	12, N/A

Table 5.1: **Comparison of BSCCM datasets**

Metadata and calibration

In addition to the image data, the BSCCM datasets also contain detailed metadata corresponding to each cell. This includes information about the microscope: pixel sizes of the sensor, all wavelengths, angles of each LED, etc. There are also background images of each LED illumination pattern, computed by taking the pixel-wise median (smaller percentiles are available as well) across many fields of view, which effectively removes the contribution of the cells themselves and captures the microscope's illumination pattern.

5.4 Methods

In this section we described how the data were generated and processed into it's final form.

Sample preparation + imaging

Assembling imaging chambers Imaging chambers were specially designed to 1) hold cells in an aqueous environment 2) be stored for a period of weeks in between cell isolation and imaging 3) be disassembled so that a subsequent histology staining step could be applied, without moving cells. Chambers were constructed with 600 μm acrylic spacer in between a poly-L lysine-coated #1.5 coverslip and a standard glass microscope slide, creating a chamber

with a volume of $\approx 450\mu$L (Fig. 5.3a). The shape of the spacer was empirically optimized for loading cells at one end of the chamber without the formation of unwanted bubbles, and it was cut from a larger sheet of acrylic using a laser cutter. The coverslips were cleaned with milliQ water then isopropanol then milliQ again, coated with poly-L lysine by placing them in a plasma cleaner, and then allowing 1mL of poly-L lysine solution to sit on the cleaned surface for one hour. The chambers were assembled by melting paraffin wax in onto the spacer and sandwiching it between two pieces of glass.

Cell isolation and staining 12 mL of blood were drawn by venipuncture and added to 50mL tubes containing red blood cell lysis buffer (Fisher Scientific, #NC9067514), which had been diluted with 10× deionized water as specified in manufacturer instructions. Cells were incubated for 10-15min at room temperature, while wrapped in aluminum foil to protect cells from light. Tubes were then centrifuged at 350g for 5 minutes. The supernatants were aspirated without disturbing the pellets, and all cells were then concentrated into a single 15 mL tube. This tube was filled with additional red blood cell lysis buffer and incubated at room temperature for another 10 min to get rid of remaining red blood cells. This tube was then centerfuged at 350g for 5 minutes and resuspended in 1mL IgG normal mouse serum control (Invitrogen #PI31880) and put on ice. Cells were counted on a hemocytometer and resuspended at 2×10^7 per mL in IgG buffer.

Antibodies were added and samples stained on ice for 30 min. An additional 700 μL were added to each tube to fill them, and they were centerfuged at 350g for 5 min. The supernatant was discarded, and they were filled with PBS+EDTA, and centerfuged again using the same setting. The supernatant was discarded, and they were resuspended in 200μL PBS+EDTA and counted on a hemocytometer. Cells were then resuspended with 50-300k cells (depending on the number remaining in each tube) in 450μL of PBS+EDTA.

The following antibodies were used at the following concentrations :

- Brilliant Violet 510 anti-human CD19 #302241 (5% dilution)
- Brilliant Violet 570 anti-human HLA-DR #307637 (6% dilution)
- Brilliant Violet 605 anti-human CD45 #368523 (5% dilution)
- Brilliant Violet 711 anti-human CD16 #302043 (4% dilution)
- Brilliant Violet 510 anti-human CD56 #318339 (6% dilution)
- Brilliant Violet 510 anti-human CD3 #317331 (5% dilution)
- Brilliant Violet 421 anti-human CD123 #306017 (5% dilution)
- Brilliant Violet 650 anti-human CD14 #301835 (5% dilution)

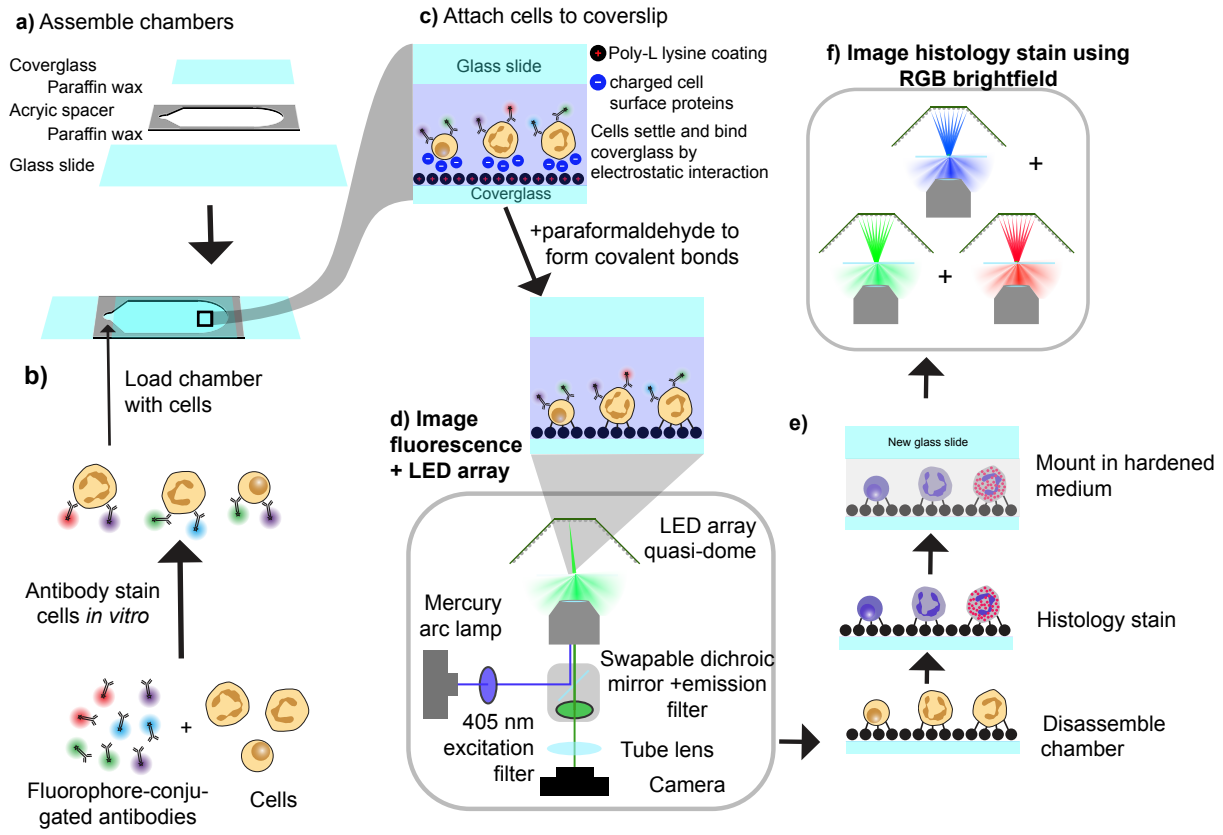

Figure 5.3: **Sample preparation and imaging a)** Imaging chambers were assembled by attaching an acrylic spacer between a microscope slide and cover glass using paraffin wax. **b)** Cells were stained with fluorophore-conjugated antibodies and loaded into the chamber by an opening at its end. **c)** Cells were attched to the coverslip first by binding through electrostatic interactions and then covalently using paraformaldehyde. **d)** The slide was then imaged using LED array and fluorescence illumination. **e)** After imaging the slide was disassembled, a Wright's (histology) stain was applied, and the cells were mounted on a new slide with hardening mounting medium. **f)** RGB histology images were collected by illuminating with each color of the LED array in series

Filling imaging chambers with cells Imaging chambers were filled by pipetting in suspended cells (Fig. 5.3b), and placed with coverglass facing down to allow the cells to settle on the coverglass and bind by electrostatic interaction. After 30 min, 22.5 μL were removed from the chamber using a pipette, taking care to keep the slide tilted and not let the resultant air bubble move throughout the chamber, as its surface tension tended to rip cells off the coverglass. Next it was replaced with 22.5μL of 16% paraformaldehyde to form covalent bonds between the cells and the coverslip. This process was repeated 4 times to increase the effective concentration of paraformaldehyde (Fig. 5.3c). The opening in the chambers were sealed to prevent evaporation by using epoxy to attach a small piece of acrylic to block the opening. Finally, the chambers were stored vertically (to prevent shear forces of air bubbles ripping away cells from the coverglass) at room temperature and protected from light until they were sequentially imaged over the next 2 weeks (Fig. 5.3d).

Histology staining After imaging in fluorescence and LED-array modes, some chambers were disassembled for histology staining (Fig. 5.3e). It was necessary to do the histology staining in a separate step, since having a histology stain on the cells during imaging with the LED array would alter their absorption/scattering properties, and could potentially alter the fluorescence signal as well. Imaging chambers were submerged under water while opened, in order to avoid the surface tension from air bubbles ripping cells off of the coverslip. The coverslip was first released from the acryclic space by sliding a razor blade underneath it. The top portion of it where the epoxied sealant was present was then removed by cutting off a piece of the coverslip with a diamond tipped knife. The slides were then stained with Wright's stain (Sigma-Aldrich # 45253) by dipping in coplin jars (Grainger #F44208-1000). They were submerged in the stain for one minute, followed by 5 dips in and out of a 75% water 25% ethanol phosphate buffer solution at pH 6.65. They were then dipped twice in a 75% water 25% ethanol low concentration (0.83 mM) phosphate buffer solution at pH 6.65. The 25% ethanol was used to reduce the surface tension of the solution and minimize the chance of cells be dislodged from the coverslip.

The remaining stain on the slide was blotted off using a Kimwipe, and the slides were left to dry overnight. The next day, excess wax was scraped off the coverslip with a razor blade, 2 drops of anhydrous mounting medium were added (Milliport Neo-Mount #109016), and a fresh microscope slide was attached. After drying, the top surface of the coverslip was cleaned with methanol prior to imaging (Fig. 5.3f).

Microscope and data collection Samples were imaged on Zeiss Axio Observer microscope using its standard fluorescence illumination path and its trans-illumination lamp replaced by a programmable quasi-dome LED array [111]. The fluorescence excitation source was a mercury arc lamp with a single band-pass filter selecting for the 405 nm peak. Fluorescence and LED array images were collected with a 20$\times$ 0.5 NA air objective and histology images were collected using a 63$\times$ 1.4 NA oil objective. All images were taken on a Basler Ace acA2440-75um USB3 Monochrome camera, using the central 2056$\times$2056 pixel region.

The microscope was controlled using Micro-Magellan[119] with some additional customized modifications to its source code. These modifications have since been generalized, and are now part of Pycro-Manager [120]. Micro-Magellan's explore feature was used to map out the imaging chamber, and its surface feature was used to mark the approximate position of the cells.

Full scanning a single slide took ∼16 hours, in part because many channels were collected, and also because there was a several second delay at each XY position in order to allow the XY stage to fully settle and prevent loss of resolution due to motion blur. Because of this long acquisition time, the focus could drift by tens of μm over the course of an imaging session away from it's originally marked position. The microscope was not equipped with a laser-based hardware autofocus system, so there was a need for an autofocus mechanism that could be executed quickly and many times over the course of imaging. We developed a single-shot autofocusing method to accomplish this, which is described in full in chapter 2 and elsewhere [117]. This autofocusing routine was executed at each XY position to provide a more precise focus after each move of the XY stage.

Slide scanning of the histology slides was performed in a similar fashion. However, the single-shot autofocus algorithm did not give satisfactory results on the histology sample. This was in part due to the much smaller depth of focus of the histology objective, but also may have been inherent to the algorithm itself. As a result, an alternative autofocus algorithm was developed, in which focal stacks were taken and the sharpest plane was computed from them. Since this was more time consuming, it was only performed on a subset of XY positions, and a running average was kept to fill in the focus offset at other positions.

Histology stained samples are typically imaged with brightfield illumination using an RGB camera. In our case, we achieved the same effect by collecting the three channels independently with a monochrome sensor, by lighting up red, green, and blue brightfield LEDs sequentially.

Exposures The use of multiple fluorophores that were all excited by a common wavelength meant that there was in many cases substantial bleedthrough from one fluorescent channel to another (we note that this was a cost-saving decision meant to minimize the number of excitation filters needed). In order to maximize our ability to later unmix data into measurements of the component fluorophores, we tuned the exposure that each fluorescence channel was collected with. First, examples of strongly stained cells were collected for each fluorophore, with a constant exposure over all channels. The summed fluorescence intensity of each cell yielded a 6-element vector, containing the brightness of each fluorophore in each channel. These vectors were each multiplied by a unique scalar, and then stacked together to form a 6×6 matrix. Using the condition number of this matrix (ratio of largest to smallest eigenvalue) as an objective function, these multipliers were optimized by gradient descent to find the best multipliers, which were then used to determine the ratio between exposures of different channels.

Exposures for the LED array channels (including the histology imaging) were picked

for each channel by empirically finding a setting that effectively made use of the camera's dynamic range.

Raw data to single-cell crops

A combination of manual data cleaning and machine learning was used to process the raw data from large slide-scans to a collection of single-cell crops. The raw data, full slide-scans in multiple channels (Fig. 5.4a), were visualized using a custom-programmed multi-resolution viewer written in Python. This enabled the identification and exclusion of visible debris on the slide and areas near the edges where optical artifacts were visible (Fig. 5.4b). Next, the 4 differential phase contrast (DPC) images at each field of view were used as input to an inverse algorithm that computed quantitative phase [155] (qDPC) (Fig. 5.4c). A difference of Gaussians blob-finding algorithm [87] implemented in Scikit-Image [161] was used to identify candidates for single cells. The parameters of this algorithm were intentionally set somewhat permissively to be sure to capture small cells, and as a result many false positive were present and these cell candidates required extensive algorithmic filtering.

Since the end goal was to find isolated single cells, this filtering began by removing all cells that were too close to another detected cell (Fig. 5.4d). In spite of the paraformaldehyde treatment, there were some cells that remained unsecured to the coverslip. This was clearly visible when watching a timelapse of the cells, as well as the fact that movement could be seen between different channels. To facilitate removal of these cells, the first and last channel during imaging ($\sim$18s apart) used the same darkfield illumination pattern on the LED array. By aligning these two images using a cross-correlation algorithm, cells that were moving could be removed (Fig. 5.4e). Next, cells that were outside the fluorescence illumination fieldstop were excluded, so that all cells in the final dataset would have valid fluorescence measurements.

At this point, the cell candidates consisted of a mix of centered, in-focus single cells and empty areas, small acellular debris, out-of-focus cells, and clumps of cells (Fig. 5.4f). A training set of $\sim$1000 cells was manually given binary labels for keeping and excluding. This labelled training set was then used to train a convolutional neural network capable of predicting whether unlabelled examples should be kept or discarded. The network architecture consisted of a 3 channel image (consisting of quantitative differential phase contrast image, a 0.5-0.6 NA darkfield image, and a brightfield image) fed into a DenseNet201 [59] architecture, followed by a 400-unit fully connected layer with ReLu activation, a 0.5 probability dropout layer, another 400-unit fully connected layer with ReLu activation, another 0.5 probability dropout layer, and 2-unit fully connected layer with a softmax output. The network was trained in Keras[20] with the Adam optimizer[67] with a learning rate of 3×10^{-6} and a batch size of 8. Training was continued until loss stopped decreasing on a held-out validation set. The performance of this network was optimized by using Keras-Tuner[101] to perform Bayesian optimization over its hyperparameters to achieve optimal performance on a held out set of validation data.

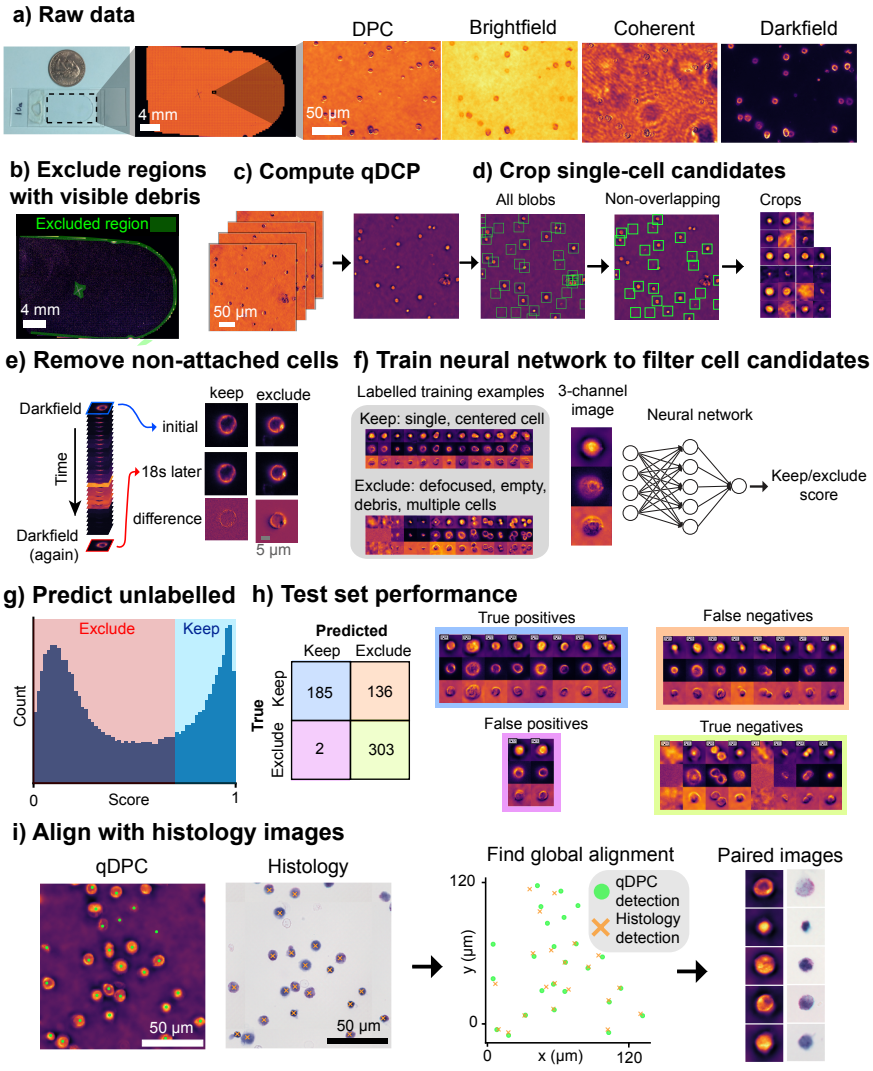

Figure 5.4: **From raw data to single-cell crops a)** a single imaging chamber, an image a full slide scan, and zoom-ins in four different illumination patterns. **b)** Regions with visible debris were manually excluded from further processing. **c)** Quantitative differential phase contrast (qDPC) images were calculated for each field of view, and **d)** a blob-finding algorithm was employed to find and crop out candidate images for isolated single-cells. **e)** Candidates that were not attached to the coverslip, as measured by movement between the first and last darkfield image were removed. **f)** A manually labelled training set of cells to include or exclude was created and used to train a neural network that predicted whether to keep cells. **g)** Histogram of predictions on unlabelled cell candidates **h)** Performance of the trained network on the labelled test set. **i)** Detected cells in differential phase contrast and histology stain contrast were aligned and matched.

Examining the performance of this network on a test set of held-out labelled examples, it was clear that many failure cases occurred when the network misidentified a clump of two cells as one. To compensate for this, the same blob finding algorithm used to originally locate cells, but with different parameters that erred on the side of detecting multiple cells when only one was present, was used to identify unlabelled cell candidate crops that potentially contained more than one cell. These examples were then labelled, thereby filling the training set with many more examples of multi-cell images, which improved its performance on these types of images on the test set. One factor that made this especially difficult was that many lymphocytes, during the time in between when they were isolated and fixed by paraformaldehyde, had begin to undergo apoptosis. The resultant expulsion of their cytoplasm gave them a characteristic dumbbell-like shape, which looked very much like two cells stuck together and was in some cases even difficult for the human-labeller. Because of this, the labels erred on the side of excluding these cases, and as a result cells that looked like this were more likely to be excluded from the final dataset.

Finally, the trained network was applied to all unlabelled data to score each cell candidate for whether it should be removed or kept. Labelling cell candidates for keeping or removing was ultimately subjective, especially when it came to how well in focus each cell was. Because of this, after labelling a certain amount of training data, classification accuracy on the test set showed asymptotic improvement with more data at around $\sim$80% accuracy. This behavior suggested that no further improvement was possible given the inherent noise in the human-provided labels. In addition, most errors on the test set appeared to be drawn from the harder to distinguish examples where this noise would be expected to be most prominent. We decided it was more important to reduce false positives than false negatives in the final dataset. That is, excluding non-centered/out of focus/non-cell images was more important than ensuring all high-quality images included. Thus, we used a conservative cutoff for the keep/exclude score of 0.7. Figure 5.4g shows the histogram of predictions on unlabelled data, as well as results on the test set.

Cells in the histology images were identified by a similar procedure. Cell candidates were first identified by a difference of Gaussians blob finding algorithm. Next, these candidates were filtered by summing all the gradients in each image and removing the ones low-values, which reliably removed out-of-focus cells or empty crops. This was the only exclusion step needed, since finding a global alignment over the full slide between histology and LED array/fluorescence images enabled one-to-one matching of histology cell candidates to LED array/fluorescence ones, and the latter had already been filtered to match the desired criteria. The histology and LED-array/fluorescence slides were aligned by computing an objective function that measured how close each histology cell was to a corresponding fluorescence cell, given global translation and rotation parameters applied to all detected cells. Specifically, for each histology cell the closest LED array/fluorescence cell was found, and the sum of these minimum distances over all histology cells was computed. A grid search over all possible values of translation and rotation of the two slides was performed, using Jax[12] for GPU-acceleration. The optimal value was confirmed by looking at the visual alignment of the two contrast modalities and the structurally similar features between the two contrast modalities

(Fig. 5.4h).

Fluorescence processing/demixing

After computing a finalized set of single, isolated, in-focus cells, the fluorescence images of each of these cells was processed in order compute estimates of the relative amounts of the protein targets of each antibody. First the raw fluorescence was measured in the foreground and background of each cell by summing all the pixels inside or outside (Fig. 5.5a) of a circle inscribed in the square crop. This size was chosen so that the pixels outside of it captured the local background intensity of that area of the slide/field of view, while the pixels inside contained the background plus fluorescence from the cell. Background subtraction and shading correction are a common practice in quantitative fluorescence microscopy due to the nonuniform pattern of excitation light across the field of view as well as the nonuniform collection of fluorescence on the edges of the image[147, 109]. In these data, there were also variations in background fluorescence between slides, likely due to antibodies in the chamber that were unbound to cells. The spatial pattern of the background for each slide was first visualized using a spatial histogram, and then a smoothed version of this was calculated using a locally weighted scatterplot smoothing (LOWESS) using locally linear regression [22] (Fig. 5.5b). The foreground brightness was calculated using the same procedure on the brightest 10% of cells after subtracting the smoothed background, yielding a shading correction. Taking the brightest cells gave a better estimate since only a fraction of cells actually had antibody for a given fluorophore bound. Finally the foreground of all cells was corrected for spatial variations by subtracting the smoother background and dividing by the shading correction, yielding a spatial histogram that no longer displayed obvious spatial variation (Fig. 5.5b, bottom right). This process was repeated over each fluorescent channel and both batches of data.

These spatially corrected fluorescence measurements were then used to solve a demixing inverse problem–going from fluorescence intensities to the relative levels of antibody-bound proteins or autofluorescent molecules that gave rise them. To solve this problem the spectra of the various fluorescent species were measured by comparing scatter plots of cells' fluorescence in conditions where they were treated with a single antibody vs the condition were they were treated with no antibody (Fig. 5.5c). Taking the difference of the means of the antibody-positive and antibody-negative populations gave a vector that measured that fluorophores' spectrum. Cells also showed varying amounts of autofluorescence. To capture the spectrum of this autofluorescence, the same procedure was repeated with the brightest autofluorescence and the rest of the population with these excluded. These measured spectra were then used to form "mixing matrices", which could be used as part of the demixing problem (Fig. 5.5d).

Fluorescence unmixing with non-negative matrix factorization Computing a spectral unmixing inverse problem, in which the levels of individual proteins can be found from the levels of overlapping spectra, can be posed as a non-negative matrix factorization problem. This models the physics of the imaging process since neither the spectra nor the

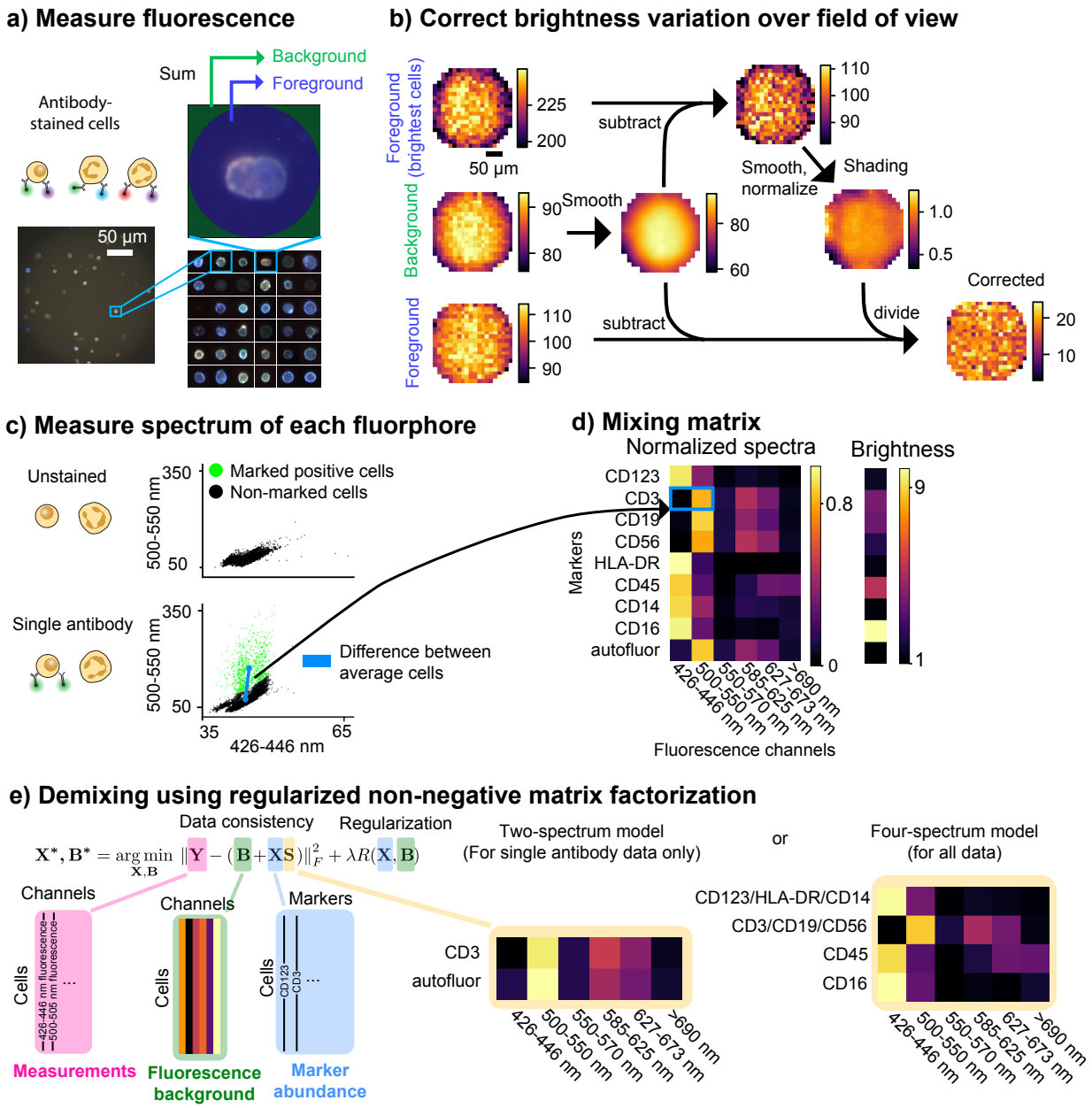

Figure 5.5: **Raw fluorescence to protein estimates a)** Fluorescent cells in a single field of view, and the areas of each crop used to compute foreground and background fluorescence estimates. **b)** The background subtraction and shading correction procedure used to correct for spatial variation in brightness across the field of view. **c)** The spectrum of each fluorophore was computed by looking at cells stained with the corresponding antibody vs. no antibodies and taking difference of the means of antibody-positive and antibody-negative cells. **(d)** The normalized spectra and relative brightness for each antibody and the autofluorescence. **e)** The regularized non-negative matrix factorization optimization problem that was solved to give estimated of the relative abundance of each protein. This problem utilized a two-spectra model (for single antibody conditions) or a four-spectra model (for every antibody condition)

fluorphore density can be negative. A simplified version of the optimization problem can be seen in Fig. 5.5e. The exact problem including normalizing constants was:

$$\mathbf{X^*, B^*} = \arg\min_{\mathbf{X},\mathbf{B}} \frac{1}{NC} \|\mathbf{Y} - (\mathbf{XS} + \mathbf{B})\|_F^2 + \lambda \frac{1}{NM} \|\text{vec}(\mathbf{X}\text{diag}(\mathbf{w}))\|_1 + \beta \frac{1}{NC} \|\text{vec}(\mathbf{B})\|_1$$

Where

$$\|\mathbf{A}\|_F^2 = \sum_{i,j} a_{ij}^2 \quad \text{(Frobenius norm)}$$

$$\|\text{vec}(\mathbf{A})\|_1 = \sum_{i,j} |a_{ij}| \quad \text{(Elementwise L1 norm)}$$

N is the number of cells, C is the number of fluorescence channels, and M is the number of spectra (i.e. one for each unique fluorophore or group of similar fluorophores). $\mathbf{X}$ is an $N \times M$ matrix where each row contains the levels of each protein for a given cell, $\mathbf{Y}$ is an $N \times C$ matrix containing the observed fluorescence of each protein for a given cell, $\mathbf{S}$ is an $M \times C$ matrix containing the fluorescence spectrum for each protein as a row, and $\mathbf{B}$ is an $N \times C$ matrix containing the background level fluorescence in each channel. Each row of $\mathbf{B}$ was constrained to be identical for cells from the same physical microscope slide, thereby enforcing the constraint of a global level of background fluorescence specific to each slide.

diag($\mathbf{w}$) is a diagonal matrix formed by putting the entries of the $M \times 1$ column vector $\mathbf{w}$ along the diagonal, where $\mathbf{w}$ is a weighting vector that enables regularizing the levels of different fluorophores independently. This was useful since different fluorophores varied by over an order of magnitude of absolute brightness. We found that a useful heuristic for setting the value of each element of $\mathbf{w}$ was to project the normalized spectrum of each fluorophore onto the first right singular vector of $\mathbf{S}$ and divide the result by spectrum's magnitude.

λ and β are global regularization tuning parameters for $\mathbf{X}$ and $\mathbf{B}$, respectively, and have values: $\lambda = 7 \times 10^{-1}$ $\lambda = 5 \times 10^{-2}$.

The optimization problem was solved using gradient descent with momentum with a learning rate of 1×10^3 and a momentum of 0.9. This was implemented computationally using Jax [12].

An important choice is which spectra will be included in the unmixing matrix. For the cases in which only a single antibody was used, more accurate results can be obtained by building this knowledge into the optimization problem and excluding spectra of fluorophores that aren't present, leaving only the antibodies spectrum and the ever-present autofluoescence spectrum forming a two-spectra mixing matrix (Fig. 5.5e).

In the case where the cells were stained with all antibodies at once, this problem becomes more complex. Ideally all of the spectra will be linearly independent and there would be more measurements (fluorescent channels) than fluorescent species to unmix. For these data, neither of these are true. Some fluorophores are similar in spectra, and the presence of autofluorescence in addition to the six fluorophores means that this is an underdetermined problem, with only six measurement channels. To compensate and come to a reasonable

solution, it is necessary to combine multiple similar spectra with insufficient signal-to-noise to be individually distinguished into one. The spectra for CD14, HLA-DR, and CD123 were combined into a single spectrum in order to achieve this. This is only possible because of the similarity among these spectra (Fig. 5.5d). Though the CD16 fluorophore also appears to have a similar spectra to these three, we note that its faint fluorescence in the >690 nm channel, combined with it having the highest absolute brightness enabled it to be effectively identified by the algorithm. CD3, CD19 and CD56 were also combined into a single channel, since the antibodies that targeted them all used the same fluorophore. Merging spectra in this way yielded the four-spectrum mixing matrix (Fig. 5.5e).

For the two-spectra model, validating the correct regularization level was performed by visualizing how well a known antibody-positive population (Fig. 5.5a) is separated from a known antibody-negative population (Fig. 5.6a). Ideally, a horizontal line separating these populations could be drawn (Fig. 5.6a, optimally-regularized). If the regularization is too weak, this line no longer runs perpendicular to the axis of the unmixed protein (Fig. 5.6a, under-regularized). If too strong, the levels of the second fluorescent species are all zero (Fig. 5.6a, over-regularized).

For the four-spectra unmixing model, results were validated by unmixing data that came from a single antibody staining condition. In this scenario, it is known that the correct answer should assign variation only along the the axis of the protein target of that particular antibody. Figure 5.6b shows the result of this experiment for four single-antibody conditions corresponding to the four spectra in the top four rows, as well as the condition with all antibodies in the bottom rows. This experiment shows that some unmixing results using this model were very reliable. For example, this can be seen in the fact that unmixed CD3 only data barely registered non-CD3 signal, and similarly for the cross-talk between CD45 and CD16. CD123, CD45 and CD16 all gave rise to signal in the CD3/CD19/CD56 spectra, which may be due in part due to the face that this model didn't explicitly account for autofluorescence.

After developing and validating these models, the unmixing inverse problem was solved on all applicable data to obtain the levels of protein abundance on each cell. The two-spectrum model was applied to all cells stained with a single antibody (with the spectrum of that antibody inserted as appropriate). The four-spectrum model was applied to all data. Though two-spectrum results are likely more accurate when available, single antibody staining conditions were also unmixed with the four-spectrum for comparison purposes.

Known imperfections

In the construction of this dataset, several technical errors were made along the way, which we describe here:

- For the cells in batch 1 that were stained with all antibodies, the amount CD19 antibody used was 35 percent of the amount used in other CD19 stains

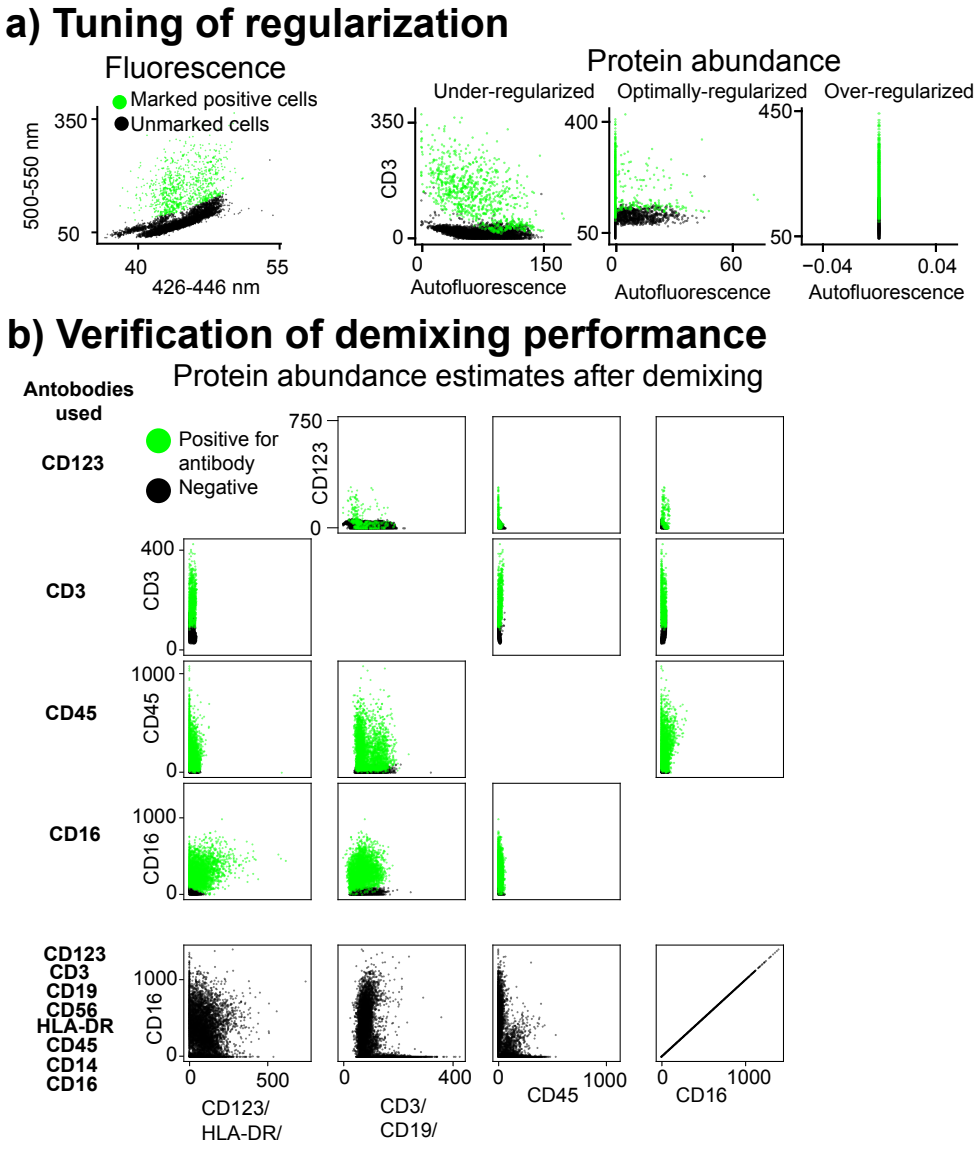

Figure 5.6: **Analysis of demixing performance a)** The effect of choosing different regularization levels on the two spectrum model. Under-regularizing fails to separate marked antibody-positive cells (green) from unmarked cells (black). Over-regularizing separates the two, but collapses all autofluorescence values to 0. Optimally regularizing balances these two. **b)** The 4-spectrum demixing model applied to single-antibody stained data (top 4 rows) or all antibody stained data (bottom row). For the single-stain cases, the algorithm successfully separates marked cells from non-marked cells with only small estimated amounts for antibodis not present in most cases, though there is there is some error for certain antibodies: for example, CD16 and CD45 into the CD3/CD19/CD56 channel

- There was a small drop of oil on an internal lens element in the objective lens used for the LED-array/fluorescence imaging. When imaging with certain low-NA single LEDs, this created a strongly visible artifact across the field of view. Because of this, these LEDs are excluded from BSCCM-coherent.

- All data were imaged with a region of interest (ROI) set on the camera to a central 2056×2056 region. However, for cells in the no antibody, batch 1, replicate 1 slide, this ROI was not set. This may have caused differences in fluorescence exposure and spatial variations compared to other slides.

- As discussed in Section 5.4, the fluorophores/filters chosen were not ideal for multi-channel imaging. This can be compensated somewhat as described in that section, but a far better choice would have been to use fluorophores with unique excitation spectra and their own matched excitation filters. We opted not to do this in an effort to minimize cost. We note that hindsight is 20/20.

5.5 Data organization

This section describes how the data is organized: which files contain which data, what metadata is available, etc. We note that a full understanding of these details is not necessary for using the dataset, as the Python package we provide abstracts away many of these implementation details.

File structure and organization

All image data are stored in Zarr [94] datasets using Blosc/zstd compression. Tabular metadata (i.e. per-cell metadata) are stored in .csv files. Global metadata, which contains information that is not specific to individual cells, but rather pertains to the whole dataset is stored in text files in Javascript Object Notation (JSON) format.

Each top-level BSCCM (regular, coherent, tiny, or coherent-tiny) contains (up to) 5 items:

- **BSCCM_images.zarr**: A Zarr dataset containing all the images of cells

- **BSCCM_backgrounds.zarr**: A Zarr dataset containing the background intensity over the full field of view, for each LED array illumination pattern.

- **BSCCM_global_metadata.json**: A text file containing metadata about the full dataset (pixel size, wavelength, channel names, etc.) in JSON format

- **BSCCM_index.csv**: A comma separated value (CSV) file containing metadata specific to each cell in the dataset

- **BSCCM_surface_markers.csv**: A comma separated value (CSV) file containing information about the surface protein marker levels of each cell, along with many measurements derived from the fluorescence images and intermediate values used in computing these levels

BSCCM_images.zarr Zarr datasets contain a hierarchy of directories. For the BSCCM_images.zarr file, this has the following structure:

```
+-- antibodies_CD16
|   +-- batch_0
|   |   +-- slide_replicate_0
|   |       +-- dpc
|   |           +-- cell_0
|   |           +-- cell_1
|   |           ...
|   |       +-- fluor
|   |           ...
|   |       +-- led_array
|   |           ...
|   |       +-- histology
|   ...
+-- antibodies_CD45
...
```

The outermost directory contains "antibodies_" followed by the name of the antibody used to stain the cells, or "unstained"/"all" for the no-antibody and all-antibody conditions, respectively.

The next level contains directories named "batch_" followed by the batch index and is either 0 or 1 for BSCCM/BSCCMNIST or 1 for BSCCM-coherent. The batch index captures the two biological replicates (i.e. distinct cell isolations) on two different dates. There is presumably some degree of biological variation between these two isolations, in addition to variation in antibody staining/processing/etc.

The next level contains directories named "slide_replicate" followed by a 0 or 1. Due to the number of available cells, some conditions (i.e. a given batch/antibody) were split among multiple physical microscope slides and imaged on different dates. The vast majority of cells/slides did not have a second replicate.

The next level contains directories that identify the contrast modality and is one of:

- "dpc": Quantitative differential phase contrast images (which were computed from raw DPC led-array images)
- "fluor": Fluorescence images, either from the fluorphores attached to antibodies, or the cells' inherent autofluorescence

- "led_array": Images taken with different LED array illumination patterns
- "histology" RGB images of histology stained cells

The next level contains "cell_" followed by the cell's global index. Each cell in each dataset has a unique global index, which allows them to be matched with per-cell metadata (described below).

Finally, within each cell directory are the Blosc/zstd-compressed binary data, split into individual files per each channel in order to maximize performance when reading only a single channel.

BSCCM_backgrounds.zarr This file contains the background images for each channel across the full field of view (2056x2056 pixels). The top level directory contains the channel name. The subdirectories are different versions depending on which pixel-wise percentile was taken over 200 images, with possible options of 5, 10, 20, 40, and 50 (median). The structure is as shown below.

```
+-- Brightfield
|   +-- 5\_percentile
|   +-- 10\_percentile
|   +-- 20\_percentile
|   +-- 40\_percentile
|   +-- 50\_percentile
+-- DF\_50
...
```

BSCCM_global_metadata.json This file contains metadata specific to the full dataset, including names of channels, collection settings like exposure, and useful information for calibration like wavelength, objective NA, etc. It is a text file with JSON structure, as shown below:


```
{"led_array":
    {"channel_names": [
        "Brightfield", "DF_50", "DF_50_Bottom", "DF_50_Right", "DF_55",
            "DF_60", "DF_60_Bottom", "DF_60_Right", "DF_65", "DF_70",
            "DF_70_Bottom", "DF_70_Right", "DF_75", "DF_80", "DF_80_Bottom",
            "DF_80_Right", "DF_85", "DF_90", "DPC_Bottom", "DPC_Left",
            "DPC_Right", "DPC_Top", "LED119"],
    "channel_indices": {
        "Brightfield": 0, "DF_50": 1, "DF_50_Bottom": 2, "DF_50_Right": 3,
            "DF_55": 4, "DF_60": 5, "DF_60_Bottom": 6, "DF_60_Right": 7,
            "DF_65": 8, "DF_70": 9, "DF_70_Bottom": 10, "DF_70_Right": 11,
            "DF_75": 12, "DF_80": 13, "DF_80_Bottom": 14, "DF_80_Right": 15,
            "DF_85": 16, "DF_90": 17, "DPC_Bottom": 18, "DPC_Left": 19,
```

```
            "DPC_Right": 20, "DPC_Top": 21, "LED119": 22}, "exposure_ms":
            {"Brightfield": 8, "DF_50": 29, "DF_50_Bottom": 58, "DF_50_Right":
            58, "DF_55": 46, "DF_60": 62, "DF_60_Bottom": 124, "DF_60_Right":
            124, "DF_65": 79, "DF_70": 84, "DF_70_Bottom": 168, "DF_70_Right":
            168, "DF_75": 93, "DF_80": 142, "DF_80_Bottom": 284,
            "DF_80_Right": 284, "DF_85": 228, "DF_90": 510, "DPC_Bottom": 17,
            "DPC_Left": 17, "DPC_Right": 17, "DPC_Top": 17, "LED119": 200},
    "wavelength_nm": 515,
    "pixel_size_um": 0.166,
    "objective": {"NA": 0.5, "magnification": 20}},
"fluorescence":
    {"channel_names": [
    ...
```

BSCCM_index.csv This contains per-cell metadata in a single, large CSV file with one row per each cell and the following columns:

```
"global_index": An integer uniquely identifying the cell
"position_in_fov_y_pix"/"position_in_fov_x_pix":Location of the cell center
    within the image field of view
"detection_radius": The radius reported by the blob finding algorithm that
    initially located the cell, which gives a rough estimate of its size
"has_matched_histology_cell": Whether or not the cell has a matching cell in
    histology contrast
"fov_center_x"/"fov_center_y"/"fov_center_z": the microscope stage
    coordinates of the field of view from which the cell was drawn
"batch": the index of the cell isolation experiment the cells were drawn from
    (either 0 or 1)
"antibodies": the name or the single antibody used to stain the cells, or
    'all' or 'unstained' if every antibody or no antibodies were used
"imaging_date": the date the slide of cells was imaged
"data_path": the path to the image data with the BSCCM\_images.zarr file
"slide_replicate": the index of the slide replicate within the same
    antibody/batch conditions (either 0 or 1)
```

BSCCM_surface_markers.csv This contains per-cell calculations about fluorescence surface marker levels. It is entirely derived from the fluorescence imaging data (as described in the methods). It contains metadata in a single, large CSV file with one row per each cell and the following columns:

```
"global_index": An integer uniquely identifying the cell
// raw measurements derived from fluorescence images
```

```
"Fluor_426-446_total_raw" //Raw foreground fluorescence in 426-446nm channel
"Fluor_500-550_total_raw" //Raw background fluorescence in 500-550nm channel
...
"Fluor_426-446_background" //Raw background fluorescence in 426-446nm channel
...
(Many more intermediate calculations)
...
// protein levels estimates from unmixing procedure
"CD45_single_antibody_model_unmixed" // CD45 protein levels using 2 spectrum
   unmixing model
"CD123_single_antibody_model_unmixed"
...
"CD123/HLA-DR/CD14_full_model_unmixed" // Combined CD123/HLA-DR/CD14 protein
   levels using 4 spectrum unmixng model
...
```

Chapter 6

Information Optimal Microscopy

All models are wrong, but some are useful.
- George Box

With the flexibility to design both imaging hardware and algorithms, computational microscopy opens up a much larger design space than traditional microscopy. This strains the assumptions underlying traditional performance metrics and design principles used in microscopy, thus necessitating the development of new set of tools for the the design and evaluation of computational microscopes.

Information theory [141] contains the mathematical tools suitable for the development of such a theory. It was originally developed for communications engineering, in which a receiver is trying to decode an unknown random message from a sender. A microscope can similarly be analyzed in a similar framework. The message is the sample under observation, and the sender is some process of Nature. The communication system over which messages are sent is the microscope. It is our job to design microscopes that gather as much information about the sample as possible, so that the processes of Nature can be decoded.

This chapter presents a high-level overview of a framework of the information-theoretic computational microscope, laying the foundation for future work to further develop new engineering principles for microscopy. A comprehensive introductory tutorial to the principles of information theory can be found in Appendix A.

6.1 Introduction

For the past few centuries microscopy has played an important role in science, engineering, and medicine. Traditionally, microscopes have consisted of a series of lenses and a detector: a human retina, photographic film, or a digital camera sensor. Recent decades have seen the emergence of *computational* microscopy, in which optics, detectors, and algorithms are designed in tandem.

The emergence of computational microscopy enables a range of new design possibilities, while also revealing limitations of the traditional ways of characterizing performance. For example, microscopes have traditionally been characterized in terms of their resolution. Different criteria have been proposed for what it means for an object to be "resolved" when the image formation process can be described by an incoherent model (i.e. when illuminating with an extended source or performing fluorescence imaging). For example, the Abbe criterion (also known as the "diffraction limit") defines resolution as the maximum spatial frequency the imaging system can collect. The more conservative Sparrow criterion characterizes resolution as the minimum separation between two point sources at which a dip in intensity between them can be seen. However, both rest on the assumption that resolution is an inherent property of the imaging system itself. More recently, these resolution criteria have been generalized to nonlinear image formation models (e.g. those using coherent illumination) [58] by using a standardized sample with a known phase delay. None of these methods provide a means of comparing across imaging systems with different computational post-processing or using different samples.

In a computational microscope, achievable resolution is often not only a function of the system, but also of the class of objects being imaged. For example, many computational imaging systems are designed using principles of compressed sensing [15], in which the ability to algorithmically reconstruct an image from a sometimes human-uninterpretable raw measurement rests on the object being sparse–that is, capable of being transformed such that it can be represented by a small number of values compared to its original dimensionality. For objects that meet this condition, it may be possible to exceed the limits implied by the traditional resolution criteria. For example, localization microscopy techniques are able to infer the position of single molecules with greater precision than the diffraction limit by capturing multiple images with only a small number of fluorescent molecules in each [8].

It is often said that the ultimate performance of such systems depends on the signal-to-noise ratio of the captured images. However, this too is ambiguous on a computational imaging system: denoising algorithms are becoming increasingly adept at generating clean images from noisy raw measurements. Is it safe to assume that algorithms that boost signal-to-noise ratio similarly boost a microscope's resolution? And when design constraints prevent both from being maximized simultaneously, is it more important to boost signal-to-noise ratio or resolution?

In the past decade, deep neural networks have had enormous success on many imaging tasks (as well as in non-imaging applications), further complicating the situation. Deep neural networks are flexible and powerful function approximators, which enables them to produce a wide range of images from raw measurements that would be otherwise impossible using a physical system alone. However, this is a double-edged sword: they can also "hallucinate" realistic-looking images that do not represent the underlying physical reality [169]. On a traditional imaging system, it is often implicitly assumed "seeing is believing." That is, the detected image is representative of the true object being imaged, even if it somewhat distorted or blurred. This may not strictly be true in practice–it is possible for imaging systems to create images with features that are incorrectly interpreted. However, this problem

is made even worse with algorithmic processing by deep neural networks. How should we describe a computational imaging system that sometimes produces images that exceed the capabilities of physical systems, but sometimes hallucinates images that do not reflect the underlying physical reality?

One potential solution to these issues is to focus not on the image produced by a computational microscope, but on the performance on a downstream task that uses the image. Often, computational microscopes are designed for a particular task, which has its own performance metric. For example, this could be the accuracy of classifying diseased versus healthy cells or the mean squared error in estimating the spatial distribution of a cell's refractive index. When ground truth data is available, these performance metrics provide absolute measurements of a computational imaging system's performance. However, they are generally measured in units that cannot be compared with each other or with the traditional performance measurements of resolution and signal-to-noise ratio. If a system is known to be able to classify diseased versus healthy cells with 90% accuracy, what mean squared error could be expected when estimating the spatial distribution of refractive index of those same cells? And what is the minimum resolution needed to reach such performance thresholds?

The issues raised by the shortcomings of traditional performance metrics in computational microscopy necessitate new ones. These new performance metrics may give rise to entirely new ways of thinking about the design of microscopes and the principles used to engineer them.

Ultimately, all of the ways of characterizing the performance of the computational imaging system reflect a universally desirable characteristic: the ability to discriminate between different possible states of an object. If two points are optically resolved, they can be discriminated from a single, larger object; a noisy image of an object might be confused for another object, while a clean image of the same object can be uniquely identified; Classification accuracy and mean squared error both measure the ability to discriminate an object from other, similar objects.

Information: universal and absolute

The quantity that measures how well different possible states can be discriminated is called **information**. While the word "information" is often used colloquially, it was given mathematical precision in 1948 in the seminal work of Claude Shannon [141], who later described that "information can be treated very much like a physical quantity, such as mass or energy." [150] This theory was originally developed in the context of transmitting messages over a unreliable, noisy communication channel, but its claims and ideas are applicable to the analysis of any human-made or natural system.

Information quantifies limits of how certain one can be about an unknown, random event. For example, suppose a fair coin has been flipped twice, but the result is unknown to us. There are four possible outcomes (`TH`, `HT`, `TT`, `HH`) and we cannot guess which one has occurred with better than $\frac{1}{4}$ chance. Now we are told that result of the first flip was `T`. How

much information have we gained? This knowledge allowed us to eliminate $\frac{1}{2}$ of the possible outcomes, thereby halving our uncertainty. This corresponds to 1 bit of information.

More generally, each bit of information allows us to eliminate half of the probability mass of a unknown, random event. Suppose instead of the coin being fair, it was weighted such that it yielded `T` with only $\frac{1}{4}$ probability. Now, knowing that the first flip yielded `T` allows us to eliminate more than half of the probability mass, because `HT` (which occurs with $\frac{3}{16}$ probability) and `HH` ($\frac{9}{16}$) can both be ruled out. Thus, we can eliminate $\frac{3}{4}$ of the total probability and only $\frac{1}{4}$ remains. We have halved the probability mass, twice. This means we have gained 2 bits of information.

Information in computational microscopy

In a computational microscope, the unknown random event is the process of Nature yielding an unknown object that we are imaging, and the detected image is analogous to being told something about the result of the coin flip. The image contains information that will allow us to reduce our uncertainty about the identity of this object, but not necessarily all the information that exists in the object. We cannot be more certain about the outcome than the amount of information captured allows.

It is important to note that the information itself is independent of how well we utilize it to make inferences about the object's identity. Information is a fixed quantity that derives from the probability distribution of possible objects that we might be imaging. Designing a good inference procedure to make use of this captured information is a separate step (although knowing something about the underlying distribution can be quite helpful in this process).

That information is related to all other performance metrics (resolution, signal-to-noise ratio, classification accuracy, etc.) is not merely an intuition–it is a mathematical fact. A sub-field of information theory called **rate-distortion theory** tells us that to achieve a given level of average performance of one of these metrics, there is a minimum amount of information required. This level of information is necessary, but not sufficient. It has to be both the "right" information and utilized in the correct way. Different pieces of information, though they may have the same value in bits, can describe different features of the data, and since different performance metrics quantify different features of the data, obtaining information relevant to those features will be most beneficial to their performance. For example, mean squared error can be improved by being able to differentiate possibilities with large differences in their values, as opposed to being able to discriminate between values with similar magnitudes.

Another important fact about information is that once lost, it cannot be recovered by subsequent physical or computational processing. Formally, this is known as the Data Processing Inequality (Sec. A.3). Practically, it means that the amount of information present in a (theoretical) noiseless image on a detector is greater than or equal to the amount of information in the detected noisy image. Similarly, the amount of information after some algorithmic inference procedure has been applied to the detected image (e.g. to classify an

image cell or denoise a noisy image) is less than or equal to the information in the detected image.

Considering an imaging system in the context of its information-gathering capabilities allows us to reconcile some of the ambiguities and limitations of traditional ways of characterizing performance. The apparent ambiguity in the idea of resolution–that it is limited by physical processes such as diffraction, but that it is possible to exceed this limit under certain circumstances–is more clear when considered from the perspective of information. If we've gathered the necessary information to determine what object we're imaging under a microscope, we can use an algorithm to create a high-resolution image of that object. Directly forming a high-resolution image of that object using a physical system is one, but not the only, way to gather this information.

A similar intuition applies to signal-to-noise ratio. A detected image has a certain signal-to-noise ratio, which we may be able to improve by algorithmically denoising it. This is only possible when the detected image contains enough information for us to discriminate from other, similar objects. Thus, an image can be generated that more closely resembles the original object. We can do this perfectly only when we have all the information needed to discriminate between every possible object of the class of sample being imaged. If we don't have all this information, any prediction made about the object will have inherent, irreducible uncertainty, and attempts to make more precise predictions than this uncertainty allows will lead to errors.

Another performance metric often used for imaging systems is perceptual quality. However, just because a computational imaging system produces images of high perceptual quality, it cannot be concluded that these images will be useful for subsequent tasks. A neural network may produce a handful of realistic looking images that match the ground truth. However, when looking at its performance over the entire distribution of possible objects and measured images, if distinct objects map to the same convincingly realistic-looking hallucination, information has been lost and the full imaging system should not be considered "good." If the information were lost at a previous step due to physical optics or detection noise, no algorithm will be able to recover it.

Information is absolute Another advantage of information over traditional performance metrics is that it is absolute. For example, the signal to noise ratio of an image is dependent on its parameterization. If we have some noisy detected image, along with a corresponding ground truth image, we could measure the quality of the image by computing its average signal-to-noise ratio. But if we then applied some invertible nonlinear transformation (like squaring the intensity values of each pixel) to both images and calculated the signal-to-noise once again, we would get a different value. The quality of the image wouldn't really have changed in this scenario, since we could always invert the transformation to get back the original (though the image *would* look different). Such transformations are ubiquitous in computational imaging systems where algorithms can process detected images in a multitude of ways. In this same scenario, the information contained in the detected image remains

unchanged. Information has an absolute significance within and across imaging systems, while many other metrics do not.

In summary, information is a powerful tool for understanding, characterizing, and improving computational imaging systems. Here, we explore how to understand the information gathering capabilities of a computational microscope, how to use information-theoretic tools to quantify its performance, and how to utilize principles of information theory to design better systems.

6.2 Related work

Several past works have considered the application of information theory to microscopy. The focus of much of this work has been to analytically calculate the upper limit on the information transmission capabilities of a microscope, using continuous representations of the images and noise involved. One of the earliest works in this area was by Fellgett and Linfoot [34], who calculated the information capacity of a microscope under the assumption of Gaussian noise. Lukosz [88] later described the number of samples/degrees of freedom needed to define an optical image based on sampling theorem arguments, suggesting that the invariance of this number of degrees of freedom explained why super-resolution imaging that surpassed the diffraction limit was possible. Bershad [7] proposed an object-independent best-case relationship between signal-to-noise, resolution, and information collected, relying on the simplifying assumption of Gaussian objects and Gaussian noise. Saleh [135] and Neifield [98] performed a similar analysis to estimate the amount of information in an image. Cox [24] proposed a "Theorem of invariance of information capacity," which extended previous work to 3D + time imaging with multiple states of polarization and calculating theoretical limits of information capacity under these circumstances under Gaussian noise assumptions, with a specific emphasis on imaging beyond the diffraction limit. More recently, Gureyev [45, 44, 46] estimated upper limits of information and information per photon for coherent imaging systems imaging weakly scattering samples. Narimanov [97] related the information and resolution limits for a restricted class of samples in far field microscope.

Another line of work nominally applying information theory to microscopy has focused on the performance limits of estimating the position of a single fluorescent point source [107, 142, 41, 126]. This work uses the Cramer-Rao lower bound, which specifies the minimum achievable variance of a statistical estimator (to estimate the molecule's position). This work utilizes the Fisher Information, an approximation of the sensitivity of a statistical model to the data it uses to make inferences. Confusingly, Fisher Information is an entirely different concept than that information/entropy that are the basis of information theory, and in fact predates Shannon's work that is widely credited for creating the field of information theory.

By making restrictive assumptions on the class of objects being imaged, and assuming Gaussian statistics to simplify theoretical analyses, it is possible to estimate what the upper limit of information transmission by an imaging system is, but not how to achieve it in practice. Cox and Sheppard [24] addressed this noting that, "it is significantly more difficult

to achieve the theoretical performances predicted by information theory applied to optics" than best case performance would predict.

In order to develop practical engineering principles to optimize information transmission, it is thus necessary to develop tools that address not only how much information *could* be transmitted, but how much information *is* transmitted. Doing so will require developing a probabilistic model of a generic imaging system, so that the probabilities of different objects and images can be reasoned about and calculated for general classes of objects being imaged.

6.3 Probabilistic model for computational microscopy

In order to apply the tools of information theory, we must first formulate a probabilistic model of a computational microscope. That is, we will define random entities (i.e. variables/vectors/matrices/functions) for each stage of the imaging system. This model will allow us both to compute and reason about the flow of information as it is extracted from the object as an optical field(s) and transformed into a digital representation on a computer. Before describing the model, we briefly describe the mathematical tools used in its construction.

Stochastic processes

Stochastic processes (also known as **random processes**) are an important mathematical tool that will be used throughout the model. Stochastic processes generalize the concept of a scalar random variable to vectors and functions. Intuitively, these vectors/functions can be be thought of as random time series. However, here they will be applied both over time and space. Different types of stochastic processes will be useful for modelling the successive stages of a computational imaging system.

For example, a simple stochastic process can be formed by taking a random variable X with some distribution, and defining a **random vector** $\mathbf{X}^N$ consisting of an ordered set of independent and identically distributed instances of X:

$$\mathbf{X}^N = (\mathrm{X}_1, \mathrm{X}_2, \ldots \mathrm{X}_N).$$

This random vector is not a particularly interesting stochastic process, since each constituent random variable is independent. More generally, $\mathrm{X}_1, \mathrm{X}_2, \ldots$ need not be independent or identically distributed, and the stochastic process can be distributed according to some joint distribution $p_{\mathrm{X}_1,\mathrm{X}_2,\ldots\mathrm{X}_N}(x_1, x_2, \ldots x_N)$.

In addition to random vectors, another type of stochastic process is the **random function**, in which a single realization is not a vector, but a function that can be evaluated on any point within its domain. In a sense, a random function is a random vector $\mathbf{X}_N$ with $N \to \infty$, since a function can be evaluated at an infinite number of points over its domain.

In other words, random functions generalize random vectors from a discrete to a continuous space.

A particularly important type of random function is a **random field**, which is a random function over a specified domain. By choosing a domain with some physical meaningful definition, like $\mathbb{R}^3$ to represent 3-dimensional Euclidean space, random fields are a useful mathematical object for describing unknown phenomena in that space. For example, a random field over $\mathbb{R}^3$ could be used to model local variations in the density of gas. Each realization of the random field would give a deterministic function $f(x, y, z)$ that can be evaluated to find the density at any point in space. Certain functions would occur with much greater probability due to their physical plausibility.

Stationary stochastic processes

Certain stochastic processes have a property called **stationarity** that can greatly simplify their mathematical description and analysis. Stationarity means that the stochastic process is shift-invariant–the joint probability distribution does not change in different subsets of its domain. For example, for a stochastic process modeling a time series this might represent different offsets in time. For example, if a random vector $\mathbf{X}^N$ is stationary, the joint distributions of $\mathrm{X}_1, \mathrm{X}_2$ and $\mathrm{X}_{N-1}, \mathrm{X}_N$ must be identical. More generally, this can be stated as, for any integer k:

$$p_{\mathrm{X}_1,\mathrm{X}_2,\ldots\mathrm{X}_k} = p_{\mathrm{X}_{1+d},\mathrm{X}_{2+d},\ldots\mathrm{X}_{k+d}} \tag{6.1}$$

Stationarity is an especially useful property in the context of information theory, because it allows the information in a random process to be summarized by a scalar **entropy rate** (Sec. A.2). Entropy rate is the average additional information provided by each increment of the stochastic process (i.e. a single element of a random vector, or an infinitesimal change dx for a random function $f(x)$).

The objects being imaged under a microscope can be naturally described as stationary stochastic processes, since they have no inherent absolute system of spatial coordinates. However, the images produced by a microscope are inextricably linked to different locations within the field of view, which usually have different performance characteristics (e.g. resolution, brightness, etc.). As a result, stationarity should be considered an approximation, albeit one that is not far from the truth. In either case, it provides substantial benefits in terms of mathematical and computational simplicity.

Model overview

We will model a generic computational microscope in four sequential steps (**Fig. 6.1**):

1. An **object** modeled by a random function $\mathbf{O}^*$ is determined by some process of Nature unknown to us.

2. An **encoder** modeled by a random function $\mathbf{E}^*$, which represents the illumination/lenses/etc., and maps the object to a **noiseless image** on the sensor–a pattern of incident energy modeled by the random vector/matrix $\mathbf{X}^*$. The choice to make this object random even though it corresponds to a deterministic optical system will be further explored below.

3. The noiseless image is measured by a **detector**, yielding a noisy image modeled by a random vector/matrix $\mathbf{X}$ with the same dimensionality as $\mathbf{X}^*$.

4. The noisy image is processed by a decoder algorithm $\mathbf{D}$ to estimate a solution $\hat{\mathbf{T}}$ for some **task** with true value $\mathbf{T}^*$, which can be modeled as a random variable/vector/-matrix/function depending on the task in question.

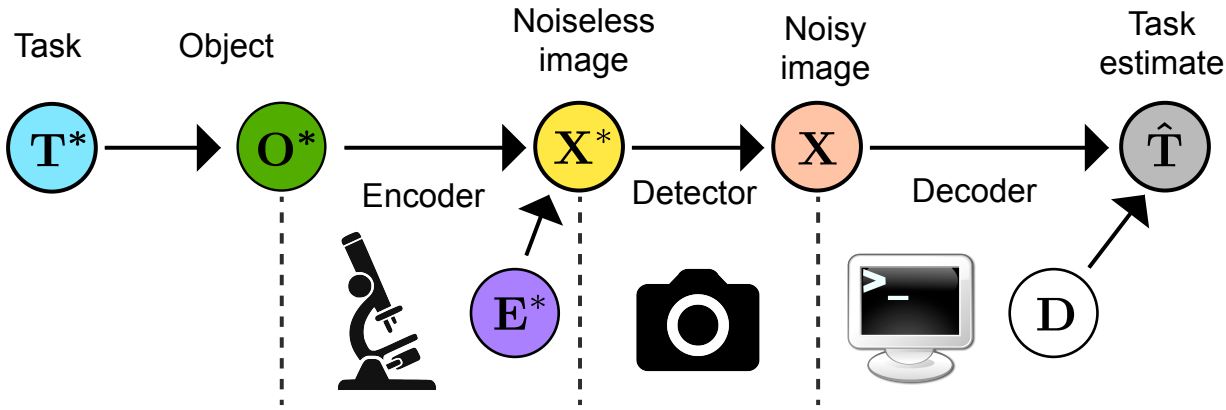

Figure 6.1: **Probabilistic model of a computational microscope** Each circle represents a random variable/vector, and arrows represent the conditional independence of implied by a Markov chain structure.

Each random entity has a distribution which captures its possible values and corresponding probabilities for a specific class of objects and microscopes. For example, when imaging white blood cells and classifying them into a discrete sets of cell types (T cell, B cell, Neutrophil, Macrophage, etc.): $\mathbf{O}^*$ is the distribution of all white blood cells; $\mathbf{X}^*$ represents the distribution of possible images on the sensor with no noise; $\mathbf{X}$ represents all possible detected noisy images; $\hat{\mathbf{T}}$ is the distribution of class labels assigned by some classification algorithm, and $\mathbf{T}^*$ represents the true class labels.

The arrows in Figure 6.1 represent conditional independence between successive stages of the microscope. For example, the detected noisy image is independent of the object given the noiseless image. In other words, these variables form a Markov chain. Mathematically:

$$p_{\mathbf{X},\mathbf{O}^*|\mathbf{X}^*} = p_{\mathbf{X}|\mathbf{X}^*} p_{\mathbf{O}^*|\mathbf{X}^*}.$$

The Data Processing Inequality (Sec. A.3) states that information can only be preserved or destroyed (but not gained) in successive steps of a Markov chain. This yields the important insight that the imaging system can be thought of as a series of independent information channels (Sec. A.10), each of which may incur some loss of information that cannot be recovered at a subsequent step. For example, if information is not present in the detected image, no algorithmic processing will be able to recover it, though many researchers claim to (we won't point fingers here). Thus, to maximize the information throughput of an imaging system, it is essential to identify and address potential losses of information at each stage.

Each mapping from one stage of the imaging system to the next (e.g. the mapping from $\mathbf{X}^* \rightarrow \mathbf{X}$ or $\mathbf{X} \rightarrow \hat{\mathbf{T}}$) can be modeled as a channel A.10, which has an associated **channel capacity** that defines the maximum amount of information it could transmit under the best of circumstances (i.e. the best input distribution). These arise from limitations such as the number of pixels on a sensor, the physical constraints on how optical fields can be transformed, and noise that destroys information relevant to the object. A well-engineered system will be able transmit information at or close to its capacity at each stage. We refer to such systems as being **information-optimal**.

Having given the necessary mathematical background and an overview of the model's interpretation, we now proceed to discussing each stage in greater depth.

6.4 Objects and tasks

The object $\mathbf{O}^*$ is the sample that the microscope is imaging. Specifically, it is the physical characteristics of its matter that alter or create optical field(s) that form an image on a detector. For example, if we are imaging cells on a brightfield microscope, in which contrast is generated by variations in the absorbance of across the sample, then the object is a full mathematical description of the cell's light absorption. This could be a function over 3D space that specifies the rate of light absorption at each point in the sample. This function could arise as a result of the intrinsic optical properties of the sample (**label-free imaging**), or it could be the result of the application of a chemical contrast agent.

Objects are generated by some process of Nature that we cannot fully predict, and thus it appears random to an observer (though not maximally random–there remains some redundancy). This is why we use imaging systems: to gather information about the object. The more information we can gather, the better we will be able to: 1) discriminate the current object being imaged from other possible objects that might have instead been present; 2) accurately estimate the structure and properties of the object; 3) solve some downstream task that is related to the object's identity or properties.

In some cases, the task of interest is really to estimate an image of the object itself; this is called an **inverse problem**. For example, in quantitative phase microscopy, we may be interested in estimating the 3D distribution of a cells refractive index, which is the same physical property that gives rise to the contrast in collected images.

In other cases, the object is merely a physical property that has some statistical relationship with this task. Continuing with the previous example of imaging a cell's absorption of light, the task might be to classify the cell as healthy or diseased. Imaging its absorption of light may provide information about its microscopic structure that is relevant to this task. The object will also contain additional information that is not relevant to this task.

Objects

Some examples of objects are:

- **A fluorescence example**: When imaging cells that have been treated with a fluorescent dye that binds DNA (e.g. the Hoechst stain), the object is the 3D distribution of individual fluorescent molecules.
- **A label-free example**: When performing brightfield imaging of unlabelled cells, the object is the 3D distribution of complex refractive index of the cells, which governs the transmission and phase changes of incident light

In either case (along with many others), the object can be represented mathematically as a random field.

The fluorescence case : In the fluorescence case, a particular instance of an object is modeled as a function of 3D space + time $f(x, y, z, t) : \mathbb{R}^4 \to \{0, 1\}$. The value of this function is 1 if the (x, y, z) location is the centroid of a fluorescent molecule at time t; otherwise it is 0. Integrating over any particular volume/time interval of $f(x, y, z, t)$ gives the number of fluorescent molecules occurring in this interval.

Since we don't know which particular object will occur, each possible function occurs with some probability, giving a random field, denoted $\mathbf{O}^*$.

The label-free case : The label-free case is similar in that it is a random field over the domain 3D space + time. However, unlike the number of molecules, refractive index is represented as a complex number, representing the absorption of and phase delay imparted to a certain wavelength of light in an infinitesimal volume and time increment. Mathematically, a single instance of this random field is the function $f(x, y, z, t) : \mathbb{R}^4 \to \mathbb{C}$. Integrating over a finite volume/time interval gives the average complex refractive index in that interval.

Once again, $\mathbf{O}^*$ represents the random field describing the ensemble of possible spatiotemporal configurations of refractive index and their corresponding probabilities.

Tasks

Examples of tasks include:

- Classifying cells as diseased or healthy

- Predicting the abundance (number of copies) of a particular protein from images of cells
- Predicting the spatial distribution of a particular protein from images of cells
- Estimating the physical properties of the object. This represents a special case where object and task are identical known as in **inverse problem**.

Tasks are denoted by $\mathbf{T}^*$, which depending on the particular type of task, could be a random variable/vector/matrix/function/field. Corresponding to its mathematical representation, $\mathbf{T}^*$ will be distributed according to a probability density function, mass function, or distribution over functions.

The image of an object will be useful in estimating the value of $\mathbf{T}^*$ if there is some shared information between $\mathbf{T}^*$ and $\mathbf{O}^*$. However, all of the information needed to estimate the value $\mathbf{T}^*$ will not necessarily be present in $\mathbf{O}^*$. As a result, even with complete knowledge of the exact object being imaged, it may not be possible to estimate the task value exactly. Conversely, knowing the exact value of the task would not necessarily allow one to estimate the object exactly. Mathematically, this is represented by the existence of a stochastic mapping from object to task and task to object. The conditional entropy $H(\mathbf{T}^* \mid \mathbf{O}^*)$ quantifies the average task uncertainty given complete knowledge of the object, and is a function of the conditional distribution $p_{\mathbf{T}^*|\mathbf{O}^*}$. $H(\mathbf{O}^* \mid \mathbf{T}^*)$ represents additional information in the object that is not relevant to the task.

It is important to note that $\mathbf{T}^*$ does not represent our estimate of the task, but rather its true value. In a subsequent section, a random entity describing the estimates we produce for the task, $\hat{\mathbf{T}}$, will be introduced. $p_{\mathbf{T}^*|\mathbf{O}^*}$ represents the limit of how much uncertainty can be reduced with the best possible imaging system.

Examples of tasks If the task is to classify a cell or diseased or healthy, T^* (which is not bold because it is scalar rather than vector valued) is a random variable that takes a Bernoulli distribution. There are two possible outcomes: 0, which means the cell is healthy, or 1, which means the cell is diseased. If the task can be predicted exactly given knowledge of the object, then $p_{\mathrm{T}^*|\mathbf{O}^*}$, the remaining uncertainty in the task after learning of the object, will have all its probability mass on a single outcome, meaning there is no uncertainty: it will be either $Bernoulli(0)$ or $Bernoulli(1)$. Otherwise, the parameter of the distribution will lie in between 0 and 1, indicating residual uncertainty.

In the case that the task is to predict the expression of a particular protein in a particular cell, T^* will be a probability mass function over nonnegative integers that correspond to the number of copies of that protein in the cell. The distribution $p_{\mathrm{T}^*|\mathbf{O}^*}$ once again defines the upper limit of how well the true number of proteins can be estimated given complete knowledge of the object.

Information content of tasks and objects

Tasks and objects both contain information, and some of that information is shared between them. Since better imaging systems will in general capture more information, it is natural to wonder how much information there is to capture. Is it the case that there is a finite amount of information, which, if completely acquired, will have achieved the upper limit of performance? While noting the limitation that our model is an approximation to the underlying physical reality, which could be more accurately described by quantum information theory, the answer to this question lies in the type of mathematical object used to represent objects and tasks.

For the random fields described above to model objects, there definitely is not a finite amount of information that could completely describe them. In the case of representing the locations of fluorescent molecules with a 3D + time function $f(x, y, z, t) : \mathbb{R}^4 \to \{0, 1\}$, though each function has a discrete range (fluorophore or not a fluorophore) that can contain no more than 1 bit of information, there are an infinite number of such points that would need to be described in order fully specify the function. In the latter example of a refractive index object, there are additionally an infinite number of possible values at each point in each function, since the range of each function is the set of complex numbers $\mathbb{C}$. Random functions used to described objects will always have either a continuous range, a continuous domain, or both, so these two examples are representative of the general situation.

Since tasks may also be mathematically represented as random field, they too may contain an infinite amount of information. However, for tasks that are distributed according to a probability mass function on a discrete outcome space, there may be only a finite amount of information present. For example, in the case of classifying diseased versus healthy cells, the average information in the task $H(\mathbf{T}^*)$ can be no more than 1 bit (and it will be less than 1 bit if cells are diseased with greater or less than $\frac{1}{2}$ chance).

If the object distribution contains infinite information, but the task distribution contains finite information, it follows that there is also infinite task-irrelevant information in the object. For example, if trying to classify a cell as healthy or diseased, certain structural features of the cell may be highly indicative of disease, while other structural features are irrelevant. In addition, there may be redundancy among task-relevant features: The same information used to infer healthy/diseased may be contained in multiple, distinct structural features of the object.

Even in the case where object and task both contain infinite information, it is possible that the information they share $I(\mathbf{T}^*, \mathbf{O}^*)$ is finite. This can only occur if there is not a deterministic relationship between task and object, and in general is complicated to calculate mathematically [35, 69].

For example, suppose the task is the 3D distribution of a particular protein on some type of cell. To measure this, cells are treated with antibodies that bind that protein, each of which is attached to a fluorophore. The object is then the 3D distribution of fluorophores that are attached to the antibodies. These are not identical because of the random nature of the chemical process of antibody binding. Some targets will have multiple antibodies bound,

some may have none, and some antibodies will bind to non-target molecules. Additionally, for correctly bound antibodies, there will be some spatial separation between the center of the target protein and the center of the fluorophore, which might be oriented randomly. As a result, only a finite amount of task-relevant information may be present in the object, and no more than this amount could be recovered without developing a better chemical labelling strategy. There are chemical limits to how precisely this antibody-labelling can be performed. This would define the channel capacity for the mapping between task and object.

Some bits are more useful than others Every bit of task information gained is not necessarily equally useful. With each bit of information gained, half of the probability mass of the task distribution can be ruled out. However, different bits of information can correspond to ruling out different outcomes. Depending on the real-world context of the task, it may be relatively more advantageous to rule out certain outcomes. For example, when predicting the level of protein expression from images of a cell, it may be more biologically significant to rule out extremely high values of protein expression than to discriminate between similar expression levels. The amount of information required is unrelated to the actual values taken and only depends on probabilities.

Information rates

Discrete representations

Though the random function/field representations are likely more representative of the underlying physical reality, approximating this reality with a stochastic process that can be indexed over a discrete set (e.g. a random vector) will greatly simplify their analysis. For example, a 1-dimensional object at a fixed point in time that would most accurately be modeled as a random function $f(x) : \mathbb{R} \to \mathbb{R}$ can instead be modeled as a random vector representing a sequence of samples from that function $(\mathrm{X}_1, \mathrm{X}_2, \dots)$. The tools of information theory are much more fully developed on the latter case, and the interpretations in many cases are more clear.

This discrete representation allows the the application of an important concept in information theory: **Entropy rates** (Sec. A.2). The entropy rate of a random vector is the average amount of information contained in the next random event, knowing the outcome of all previous random events. Mathematically:

$$H(\mathrm{X}_N \mid \mathrm{X}_{N-1}, \mathrm{X}_{N-2}, \dots)$$

This is simplest to consider in a stationary stochastic process, where the entropy rate does not depend on the value of N.

Entropy rate quantifies the uncertainty of the next event in a random process as it unfolds over time, space, or some other dimension. For example, when watching a living cell under

a microscope, there will always be some uncertainty about its likely future behavior. There are an ensemble of possible behaviors it could exhibit at the next time point, each of which occurs with some probability. If it is hard to predict what will happen next, the entropy rate will be high. If the entropy rate is 0, there is no chance the cell will move or change at the next time point, which means the cell is probably dead.

For a given object, its entropy rate over space does not necessarily equal its entropy rate over time. There can even be different entropy rates over different dimensions of space. For example, cells imaged on a microscope are usually placed on a piece of glass that is oriented perpendicular to the Earth's gravitational field. As a result, many types of cells will be characteristically splayed across this glass and much wider than they are tall. It may be easier or harder to predict the physical composition of a cell along this perpendicular axis compared to the horizontal axes, meaning there are different entropy rates.

One reason entropy rates are important to consider is because there is always some kind of finite budget for imaging that can be allocated to collecting different dimensions of entropy. We don't have infinite time to image an object, nor can we always image its full spatial extent. High performance imaging systems will be able to gather information at a high rate relative to this budget. How to do this is the subject of the following sections.

6.5 Encoders

Having defined *what* is being imaged, the object $\mathbf{O}^*$, *and* why it is being imaged, to perform a task $\mathbf{T}^*$, we can now discuss *how* to image it. The first step of this process is performed by an **encoder**, which is the physical optical system (illumination, lenses, etc.) that forms a noiseless image $\mathbf{X}^*$ on the surface of detector. $\mathbf{X}^*$ is a **noiseless image**. The noise introduced in the detection process will be considered in a subsequent detection step, but the mathematical structure of $\mathbf{X}^*$ depends on properties of the detector like pixel size, shape, etc.

Light incident on the detector can carry information extracted from the object in several of its physical properties. Assuming a fixed, deterministic optical system that uses coherent, monochromatic light, any variation in properties such as polarization, phase, amplitude, spatial distribution, and temporal dynamics is a consequence of the object, and thus carries some information about the object.

However, much of this information may be lost in the encoded noiseless image. The continuous optical field incident on the detector is discretized into an array of pixels, each of which integrates some finite spatial and temporal extent of the incident intensity. Properties such as polarization and phase will not influence the integrated intensity, which only depends on the amplitude of the incident electric field, and thus any unique information carried in their variations will be lost.

The discrete nature of the encoded noiseless images means that they contain a finite amount of information. Calculating the amount of information contained, either through

exact or approximate methods, will be important for assessing the performance of the optical components of a computational imaging system.

Extractors

Before an image can be formed on the detector, information must be extracted from an object in the form of a scattered/absorbed/emitted optical field. This field can never be measured directly with existing technology, but it is very likely that it contains information. However, it does not necessarily contain *all* the object's information. If multiple objects produce the same extracted field, then the information that enables those two objects to be discriminated will be lost. It is thus useful to study extractors in a theoretical context, to determine what information is lost and how, because if information is lost at the extraction stage, no subsequent steps can recover it.

Like an object, an extracted field can be mathematically modelled as a random field. For example, a monochromatic coherent field could be represented by a function: $f(x, y, t) : \mathbb{R}^3 \to \mathbb{C}^2$, where the function's range is $\mathbb{C}^2$ to represent the two polarization components. The random field would be an ensemble of such functions corresponding to the ensemble of possible objects. Like the object, this representation implies that an extracted field contains infinite information.

Despite containing infinite information about the object, some information about the object may nonetheless be lost in the extracted field (a fraction of infinity is still infinity). Theoretically analyzing this loss of information may be informative in the design of systems.

A trivial example of task-relevant information being lost at the extraction stage would be imaging the spatial distribution of a particular protein using fluorophore-conjugated antibodies in a cell, but using the wrong wavelength of excitation light. There will likely still be light incident on the detector (from autofluorescence of other molecules), but this light will not contain any information about the task of interest. Modifying other parts of the system (e.g. reducing optical aberrations) will not be able to recover task-relevant information.

If information has successfully been extracted by producing variations in an optical field that correspond to variations in an object, the next challenge will be to encode that information into something that can be detected (i.e. variations in time and space-integrated intensity). For example, when illuminating a class of objects that vary in their refractive index with coherent light, information will be extracted in the phase of light that has passed through the object. When measuring the integrated intensity of this light, this information will be lost, because the light's amplitude is unaffected. However, if the light that has passed through the object interferes with light from the same source that did not pass through the object (e.g. as in a Michelson interferometer), these phase variations are converted to integrated intensity variations, and the information can be recovered.

We cannot directly measure the information contained in an extracted field, and since it is usually infinite anyway, it is unlikely future detectors will enable this. Thus, extracted fields can be analyzed in two ways. First, physics-based mathematical models can be used to determine what variations in optical fields are lost by idealized imaging systems. For

example, the numerical aperture of an imaging system will determine what spatial frequencies are not collected by the imaging optics (in the absence of illumination trickery). Objects that vary only in high-frequency details outside of this limit will not be able to be discriminated.

Second, the information present in encoded noiseless images can be analyzed to determine not what information has been lost, but rather what information has been recovered.

Encoders

The **encoder** maps an extracted field onto a detector, forming a noiseless image consisting of a discrete array of pixels. There will undoubtedly be a loss of information at this stage, since the noiseless image formed on the detector will integrate over several properties of the incident field [169]. For example, a monochromatic camera will integrate over wavelength spectrum, incident angle, polarization states, temporal variations within the exposure time, and spatial variations smaller than the size of a pixel. Any object information that is carried only by variations in these properties in the field incident on the sensor will be lost.

Though extracted fields can be thought of as containing infinite information, discrete noiseless images always contain finite information. Good imaging systems will maximize the amount of task-relevant information present in the encoded noiseless image. Mathematically, $I(\mathbf{T}^*; \mathbf{X}^*)$ should be as high as possible.

The maximum value of $I(\mathbf{T}^*; \mathbf{X}^*)$ is limited by the number of unique images that could be recorded on the detector, a limit which we refer to as the **sensor capacity** C_{sensor}. Mathematically, the set of possible noiseless images $\mathcal{X}$ can be written as:

$$\mathcal{X} = \{0, 1, \ldots W-1\} \times \{0, 1, \ldots H-1\} \times \{0, 1, \ldots 2^{\#(\text{bits per pixel})-1}\}$$

Where W and H are the width and height of the sensor in pixels, each of which can represent a finite number of brightness values determined by the bits per pixel of the camera. The maximum number of distinct images that can be formed on the sensor $|\mathcal{X}|$ is $W \times H \times 2^{\#(\text{bits per pixel})}$. The sensor capacity is determined by the maximum entropy over the space $\mathcal{X}$, $H_{\max}(\mathcal{X})$.

$$C_{\text{sensor}} = H_{\max}(\mathcal{X}) = \log_2 |\mathcal{X}|$$
(The sensor capacity)

It means that on average, images cannot contain more than:

$$I(\mathbf{T}^*; \mathbf{X}^*) \leq \log_2 |\mathcal{X}| = \log_2(W \times H \times 2^{\#(\text{bits per pixel})})$$

For example, a 1024×1024 pixel sensor with 8 bits per pixel (data bits, which upper limit bits of information) has 256 possible brightness values at each pixel and will be able to

represent $1024 \times 1024 \times 256 = 2^{28}$ possible images. As a result, a distribution of such images can contain no more than $\log_2 2^{28} = 28$ bits of information.

However, it is unlikely that the sensor capacity can be achieved in practice, because of physical limitations of the optics forming the image. 28 bits would only be achievable if the encoder could map a distribution of objects to a uniform distribution over every possible image. Images collected on a microscope generally have a characteristic appearance that is distinct from the uniform random noise images that would be needed to maximize encoded information, meaning they carry less information than the sensor capacity. Thus, this is unlikely to be achievable in general, because of the physical constraints of optical systems. This can be mathematically described by a new limit: the **encoder capacity** C_{encoder}.

The encoder capacity will usually depend on both the degrees of freedom of the imaging system, and the objects being imaged. This is because most optical systems cannot perform arbitrary transformation, like mapping an object white noise, that would be required to map different objects to different noiseless images on the sensor. For example, a linear-shift invariant system imaging a 2D object will only be able to produce images that are a convolution of the field exiting this object.

Mathematically, an encoder function can be written $e_{\boldsymbol{\theta}}(\mathbf{o}^*) : \boldsymbol{\mathcal{O}} \to \boldsymbol{\mathcal{X}}$, where $\boldsymbol{\theta}$ represents the degree(s) of freedom of the optical system, $\boldsymbol{\mathcal{O}}$ is the space of possible objects, and $\boldsymbol{\mathcal{X}}$ is the space of possible noiseless images. The encoder maps objects to noiseless images:

$$\mathbf{X}^* = e_{\boldsymbol{\theta}}(\mathbf{O}^*)$$

The encoder capacity can be written as.

$$C_{\text{encoder}} = \max_{\boldsymbol{\theta}} I(\mathbf{T}^*, e_{\boldsymbol{\theta}}(\mathbf{O}^*))$$

(The encoder capacity)

By construction, the encoder capacity contains information less than or equal to the sensor capacity, since

$$I(\mathbf{T}^*, \mathbf{X}^*) \leq H(\mathbf{X}^*) \leq H_{\max}(\boldsymbol{\mathcal{X}}) \tag{6.2}$$

Calculating and understanding the physical limits in information encoding for a particular imaging system is important both to determine its best-case performance, as well as to develop strategies for approaching that best-case performance. Doing so requires an understanding the physics of the optical system and their interactions with the object. This can be approached through simulation or approximated empirically if it is possible to capture nearly noiseless images of an object. The latter corresponds to the situation of taking an infinitely long exposure of a static object, allowing the recorded image to converge to the true noiseless image by the Law of Large Numbers.

With either simulation or empirical approximation, calculating the best-case and actual performance will encounter difficulties due to the "curse of dimensionality." Computing information theoretic quantities such as entropy and mutual information requires summing over the relevant outcome space. In the example above that space is the space of possible images, which has 2^{28} elements. Thus, this computation will require computing a sum over a 2^{28}-dimensional space, a feat that is too difficult for current computers [1]. As a result, we must resort to approximations, which despite being inexact, provide important insights into how to produce information-optimal encoders.

Encoder uncertainty

The above discussion has been based on the assumption that $e_{\boldsymbol{\theta}}(\cdot)$ is a deterministic function: there is a fixed value of $\boldsymbol{\theta}$ that determines the optical characteristics of the encoder (e.g. the imaging system's point spread function). Even if this is true, it is also true that we can never be completely certain what the value of $\boldsymbol{\theta}$ is for a particular imaging system. For example, point spread functions can be estimated or measured, but there will always be some error due to the presence of noise.

Thus our inability to know the exact functional form of the encoder will introduce additional uncertainty into the measurement of the object. For example, a blurry-looking patch on a noiseless fluorescence image could equally well be the result of an object with diffusely spread fluorophores, or a concentrated spot of fluorophores imaged through a highly aberrated imaging system. Knowing that the imaging system is not abberated enables the presence of diffusely spread fluorophores to be inferred. In the absence of this knowledge about the imaging system, the two possibilities will not be able to be discriminated. Since we can never be completely certain of this knowledge of the imaging system, there will always be residual uncertainty.

To model this uncertainty, the degrees of freedom of the encoder $\boldsymbol{\theta}$ must themselves be considered random. As a result, $e_{\boldsymbol{\theta}}(\cdot)$ can be considered a random function, which we denote as $\mathbf{E}^*$. $\mathbf{E}^*$ represents an ensemble of functions that map from object $\mathbf{O}^*$ to noiseless image $\mathbf{X}^*$, each with corresponding probability. For example, these functions might be different point spread functions of a microscope, and their probabilities determine optical aberrations introduced by common misalignments in optical components. It is important to emphasize that this uncertainty is not uncertainty in our *beliefs* about the how the microscope is aligned (as would arise in a Bayesian interpretation), but rather the uncertainty inherent to the problem, which is usually limited by either the variability in the physical process of the optical system's construction, or our inability to calibrate exactly.

Some sources of encoder uncertainty may include:

- Variations in the point spread function at different points in the field of view due to field-varying aberrations

[1] Quantum computers to the rescue?

- Variations in the position of the object relative to the microscope's focal plane due to imperfect performance of mechanical hardware
- Variations in illumination across the field of view
- Background speckle patterns/coherence artifacts across the field of view when using coherent light due to imperfections in the optical system

Uncertainty vs. noise This uncertainty is conceptually different from detection noise because the uncertainty is, in principle, knowable. If a perfect characterization of the imaging system became available, the effects of the imaging system on the noiseless image would no longer appear random–there would be no uncertainty about the physical system that produced the image. In contrast, uncertainty introduced by "noise" does not have this property. As will be described in the next section, this noise arises from quantum phenomena that are inherently stochastic. There is no calibration that could be performed to account for noise in the detected image.

Digging deeper, it is worth noting that a perfect characterization of an imaging system is impossible. It will always rely on some empirical calibration, such as measuring the system's point spread function, and this measurement can never be perfectly precise due to the presence of noise. So in the end, encoder uncertainty may have the same effect as noise: potentially causing loss of information about the object, and introducing additional variations, which, if not removed by a decoding algorithm, will degrade the performance of estimating the solution of a task.

In practice, the calibration of most microscopes does not approach the fundamental limit of noise-derived uncertainty. Models of imaging systems are often limited by computational considerations that preclude the use of exhaustive, more physically accurate calibration. For example, the common assumption of linear shift-invariance is usually not entirely true, but measuring and utilizing the point-spread function unique to each point in the field is impractical.

The power of random encoders

Considering an encoder as a random function $\mathbf{E}^*$ is necessary to model its inherent uncertainty, but we may also be interested in designing encoders to be random. One of the central results of information theory, the Noisy Channel Coding Theorem (Sec. A.4) proves that under certain asymptotic conditions, transmission of information through a noisy channel can be be maximized using a *random* encoder, and no specially-designed encoder can have better performance than this. Therefore, taking advantage of randomness in the encoder to maximize information transmission is a powerful concept. In order to analyze the randomness of the encoder, the distribution of possible encoders of a particular class must be considered, which is possible using the random function E*. Balancing the incorporation of randomness and the uncertainty introduced by doing so (because it makes decoding harder) is an important consideration when jointly designing optical encoders and algorithmic decoders.

Detector

Though we can theoretically analyze or simulate a noiseless image $\mathbf{X}^*$ formed on the detector, in experimental situations we will always work instead with a detected noisy image $\mathbf{X}$. The noise in the detected image may arise form three sources: 1) "Classical" fluctuations in the intensity of the light incident on a sample determined by the source of light. 2) "Quantum" noise that arises from detecting photons, also known as **shot noise**. 3) Electronic noise from the detector, also known as **read noise**.

Separation of encoding and detection The model rests on the assumption that image formation (encoding) and detection can be considered independently. Others have proposed similar versions of this separation [128]. This assumption arises directly from the "semiclassical" treatment of the statistical properties of electromagnetic radiation ([149] Ch. 9). This represents a hybrid of the most rigorous theoretical treatment of light-matter interactions, which is based on quantum electrodynamics, and the simplified "classical" approach, which ignores quantum effects. The semiclassical approach treats light classically until it interacts with atoms of a photosensitive material on the detector.

Within the semiclassical model, there are two potential sources of noise, classical and quantum. The classical noise arises from fluctuations in the intensity of light illuminating the object, which are characteristic of the physical process producing light. For example, thermal light, as would be produced from the filament of a bulb, will have different fluctuations in its intensity than laser light. Lasers in general produce light that is more stable, but nonetheless always have some degree of intensity fluctuation. Quantum noise arises from the fact the information is carried via photons whose behavior is governed by quantum mechanical effects. This inescapable randomness in detected photons is usually called "shot noise."

In the visible region of the electromagnetic spectrum, quantum noise dominates classical noise, such that the latter can be ignored without affecting results. As a result, light can be treated as deterministic up to the point that it interacts with a photosensitive material (i.e. the detector) and the stochastic effects of quantum noise are realized. This approximation is what enables encoding and detection to be considered in separate, independent steps.

Detection as a noisy channel

Detectors are directly analogous to noisy channels in information theory (Sec. A.10). A simple model for a noisy channel/detector is as a conditional probability distribution $p_{\mathbf{X}|\mathbf{X}^*}$, which describes the probability of each possible noisy image as a function of a noiseless input image.

The presence of noise reduces the amount of information that can be transmitted through the channel. Intuitively, this occurs because the ability to discriminate between different images (information) lessens when observing noisy versions of those images, which might be confused for one another. Mathematically, the amount of successfully detected information is measured by $I(\mathbf{X}^*;\mathbf{X})$. This quantity can be decomposed as (Sec. A.2):

$$I(\mathbf{X}^*;\mathbf{X}) = H(\mathbf{X}) - H(\mathbf{X} \mid \mathbf{X}^*)$$

Where $H(\mathbf{X})$ quantifies both the information in the detected image and additional variation due to noise, and $H(\mathbf{X} \mid \mathbf{X}^*)$ quantifies the average noise added to a specific distribution of noiseless images.

Alternatively, it can be decomposed as:

$$I(\mathbf{X}^*;\mathbf{X}) = H(\mathbf{X}^*) - H(\mathbf{X}^* \mid \mathbf{X})$$

Where $H(\mathbf{X}^*)$ is the average information in the noiseless image, and $H(\mathbf{X}^* \mid \mathbf{X})$ is the information loss in the detected image due to noise.

Even with future detector technology that could eliminate read noise thanks to advances in magic, the presence of shot noise is an inescapable fact of Nature, and this noise will always lead to a loss of information. The presence of this noise means that $H(\mathbf{X}^* \mid \mathbf{X})$ will always be greater than 0. The mathematical structure of shot noise (i.e. a Poisson distribution) dictates that if a nonzero noiseless image $\mathbf{X}^* = \mathbf{x}$ is incident on the sensor, every possible detected image might occur with some probability (though most images will have very low probability). This means that the presence of shot noise will always make it harder to discriminate between which noiseless image was incident on the detector, or equivalently, it will lead to a loss of information. As a result, the **detector capacity** C_{detector}–the amount of information that can be transmitted through a noisy detector will always be less than the encoder capacity.

Taken together with the previous results, this implies that:

$$C_{\text{detector}} < C_{\text{encoder}} \leq C_{\text{sensor}}$$

Somewhat incredibly, the Noisy Channel Coding Theorem (Sec. A.4) demonstrates that under certain circumstances, it is possible to make the loss of information when transmitting through a noisy channel arbitrarily close to zero. This has important implications for the design of good optical encoders, though unlike the circumstances under which this theorem is proven, optical encoders have additional physical constraints that must be accounted for.

Optimal noiseless and noise-robust distributions The distribution of noiseless images that contains the most information given the physical constraints of an encoder (and considering that encoder as fixed) will not necessarily be the distribution that contains the most information after noisy detection. To maximize the entropy of noiseless images, a good encoder will try to make a distribution that is uniformly probable over all physically realizable images. In contrast, the encoded distribution that preserves the most information after noisy detection will instead create a distribution in which the most probable noiseless images will have the least noise added to them and also produce noisy images that are maximally discernible from one another.

Decoders

The final stage of a computational imaging system is to use an algorithm called a decoder, which is a function $d(\cdot)$ that takes in the detected image $\mathbf{X}$ and produces some type of **estimate** relevant to a particular task. The task could be to solve for the object itself, in which case it is known as an "inverse problem". For example, a deconvolution algorithm, which attempts to remove the blur introduced by the encoder and noise introduced by the detector, might be used for this purpose. Alternatively, the task might be to predict another quantity which shares some information with the object. This case includes an example that will be discussed repeatedly throughout this section, estimating the abundance of a particular protein given an image of the light scattered off of a cell. This example has the advantage that the probability space of the solution is a scalar, which allows for plotting the relevant distributions.

Decoders face a different set of challenges than encoders. A good encoder creates images that contain as much information as possible in such a way that it is robust to the noise of detection and approaches the information transmission limit of the detector capacity. In contrast, there is not an analogous limiting capacity once a noisy image $\mathbf{x}$ has been recorded on a computer, since no more information can be extracted from it than the number of (data) bits already used to store the image on a computer. Instead, for decoders the challenge is to transform that information into a useful form to perform some task, while discarding noise and other information irrelevant to that task.

A necessity for the analysis of decoders is that they produce outputs that can be compared to the true values. Mathematically, this means that their outputs must be defined on the same probability space as the task of interest. For example, when trying to predict the number of proteins on a cell, the probability space $\mathcal{T}$ is the set of non negative integers. The detected image, which is an element of the set of possible images $\boldsymbol{\mathcal{X}}$ is not interpretable until transformed onto the space $\mathcal{T}$.

Given that the decoder produces estimates on the correct space, the quality of these estimates should be evaluated based on three criteria:

Decoder optimality criteria

1. **Sufficiency**. The estimates should contain as much task-relevant information as possible.

2. **Minimality** Variations in the estimates produced by the decoder should be strictly related to variations in the true value being estimated. That is, the estimates should not have additional entropy that carries no information about the images. To achieve this, the decoder must discard irrelevant variations in the detected images that arise from detector noise, encoder uncertainty, irrelevant features of the object, etc.

3. **Predictivity**. The decoder should produce estimates that are consistent with the true values taken by instances of the task. When the task can be directly measured, this

simplifies to saying that the estimates predicted by the decoder should be similar to measurements.

Sufficiency

Estimates made from an input $\mathbf{x}$ about a task should contain as much task-relevant information as possible. This raises the questions: how much information is there to utilize? and will it be enough to decode the value of the task exactly? To answer these questions, we must introduce the concept of **aleatoric uncertainty**.

Aleatoric uncertainty Suppose we had access to a magical decoder that consistently made predictions that were interpretable and perfectly sufficient, minimal, and predictive. Is this equivalent to saying that the decoder would always produce a point estimate of the task equal to its true value? It is not: the problem may have inherent, irreducible uncertainty, known as **aleatoric uncertainty** [60] that even a magical decoder could not reduce.[2]

Aleatoric uncertainty is determined by the amount of information we don't have that is needed to estimate the true value of the task exactly. It can result from information lost in an earlier step of the imaging system (extraction, encoding, detection, etc.), in which case, changing the imaging system may change the level of aleatoric uncertainty. Alternatively, it could simply be that the task $\mathbf{T}^*$ has some inherently stochastic relationship with the object $\mathbf{O}^*$. For example, we want to measure the locations of proteins, but instead we measure the fluorescent molecules bound to them which are slightly offset in a random direction.

The level of aleatoric uncertainty could be different for different inputs. For example, if predicting protein abundance from images of cells, for some images of cells it may be possible to predict protein abundance almost exactly, and in other cases there might be little information in the image. In the latter case, the lack of information means that only highly uncertain estimates are possible–the aleatoric uncertainty is high.

Stochastic decoders

The simplest and most straightforward type of decoder is a **deterministic decoder**, which takes in an image and produces a point estimate $\hat{t}$ for the task. Mathematically, $d(\mathbf{x}) : \boldsymbol{\mathcal{X}} \to \mathcal{T}$. However, in the presence of aleatoric uncertainty, a deterministic decoder may not be a suitable choice.

An alternative is a **stochastic decoder**, which does not estimate a point $\hat{t}$ but instead a probability distribution $\hat{p}$. We will denote this distribution $\hat{p}(t \mid \mathbf{x})$, since it is conditional on the image $\mathbf{x}$ fed into the decoder. Using $\mathcal{P}_{\mathbf{T}}$ to denote the space of probability distributions over the space of the task, a stochastic decoder can be mathematically defined:

$$d(\mathbf{x}) : \boldsymbol{\mathcal{X}} \to \mathcal{P}_{\mathcal{T}}$$

[2]Magic is subject to the laws of information theory too

For example, given an image of cell $\mathbf{x}$, the decoder would produce a probability mass function describing the possible number of proteins on that cell (**Fig. 6.2a**).

Stochastic decoders are more general than deterministic decoders. It is always possible to derive a deterministic estimate from the output of a stochastic decoder, for example by taking the median or mean of $\hat{p}(t)$.

The generality of stochastic decoders is useful for two reasons. First, it enables a more complete theoretical description of the optimal characteristics of decoders in the presence of aleatoric uncertainty. Second, they can extract more task-relevant information than deterministic decoders in the presence of aleatoric uncertainty, because of their ability to adapt to the inherent randomness of the estimation problem.

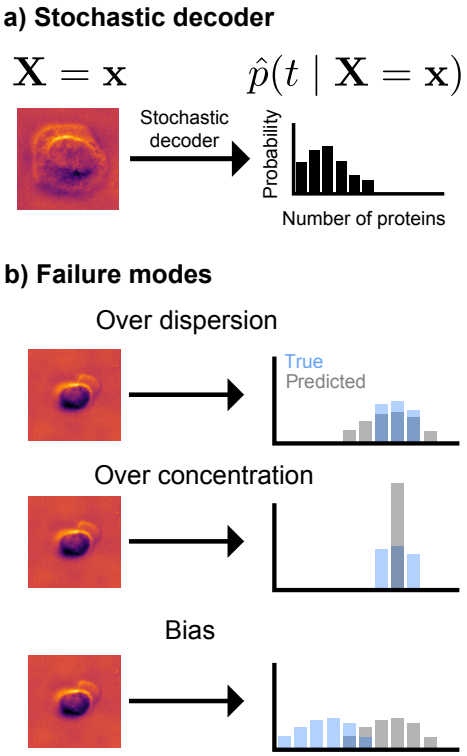

Figure 6.2: **A stochastic decoder a)** A stochastic decoder, which takes in an image of a cell and produces an estimate of the distribution of proteins on that cell. **b)** Stochastic decoders can produce incorrect estimates that are overly dispersed, overly concentrated, biased, or some combination thereof.

Defining the optimal stochastic decoder $d^*(\cdot)$ is straightforward: it produces an estimate $\hat{p}(t \mid \mathbf{x})$ that is equal to $p(t \mid \mathbf{x})$, the true distribution defined by aleatoric uncertainty for every possible input $\mathbf{x}$. In reality, this perfection is unattainable. One reason is because of

deficiencies in the second optimality criterion: Minimality, the principal that the estimates should not have additional entropy that carries no information about the images.

Minimality

For any reasonably complex problem, the estimates made by a decoder will usually contain additional variations unrelated to variations in the task itself. In other words, there will be additional entropy/uncertainty/noise. This can be broadly grouped under **epistemic uncertainty** [60], which describes not the inherent uncertainty of the problem but additional uncertainty caused by our own ignorance of the solution.

The resulting variations can be seen as errors that manifest in multiple ways: relative to the true distribution, the predicted distribution may be overly dispersed, overly concentrated, biased, or some combination thereof (**Fig. 6.2b**). This error results from the decoder that we employ in practice $d(\cdot)$ differing from the optimal decoder $d^*(\cdot)$[3].

There are two important subcategories of epistemic uncertainty. The first is philosophical in nature and harder to model and quantify. It relates to how to choose the decoder from a set of possible alternatives ("**decoder choice uncertainty**"). The second arises from the failure to suppress noise and irrelevant information, given the choice of a particular decoder. It has a concrete information-theoretic interpretation and mathematical formula. These two categories are intertwined: the errors made by a particular decoder are an important consideration in its selection as appropriate. After introducing the former in the next section, we will subsequently ignore it by assuming a fixed decoder. Thus, moving forward, "epistemic uncertainty" and "minimality" refer to the failure to suppress noise and irrelevant information.

Decoder choice uncertainty

The first subcategory arises from uncertainty about choosing the decoder itself, and requires reasoning about the broader set of possible decoders. Most commonly, decoders are designed based on physical principles or learned from data using machine learning approaches. Which data is used to learn these decoders and the learning procedures will be important factors affecting their performance. In the case of physical modeling, these approaches will involve choosing which assumptions and approximations to use. Making concrete choices about the appropriateness of decoders in either case is both philosophically and technically challenging.

Nonetheless, logically separating the choice of decoder from the probabilistic estimates of different decoders is important because it distinguishes two fundamentally different states of knowledge: knowing that something is uncertain (i.e. has high aleatoric uncertainty) versus ignorance about whether it is uncertain or not (i.e. unknown aleatoric uncertainty due to uncertainty over decoders). A single probability distribution cannot capture the difference

[3]The existence and significance of such an optimal decoder may be philosophically debatable[42], but this issue is beyond the scope of the current work

between these two concepts: a highly dispersed distribution could be interpreted to represent either one. Thus, this outer layer of uncertainty over the choice of decoder is warranted.

There is disagreement about the best way to mathematically represent this additional layer [60]. One approach is to use a separate probability distribution (e.g. over different decoders or different parameters of a single decoder). Alternatively, others argue for using a set of decoders to represent epistemic uncertainty, where a larger set indicates a higher degree of ignorance of the problem's uncertainty.

Adopting the former approach, we create a separate probability distribution over decoders. This is represented by the random function $\mathbf{D}$. With a fixed input image $\mathbf{x}$, $\mathbf{D}(\mathbf{x})$ is a random (decoder) function that deterministically produces an estimated task distribution. The estimated task distribution is thus itself random, since it depends on the random choice of decoder. Mathematically, this random estimated task distribution is denoted $\hat{P}(t \mid \mathbf{x})$ (where the capital $\hat{P}$ differentiates from a specific estimated distribution $\hat{p}$).

Different decoder functions $d_1(\mathbf{x}), d_2(\mathbf{x}), ...$ can each produce their own estimates $\hat{p}_1(t \mid \mathbf{x}), \hat{p}_2(t \mid \mathbf{x}), \ldots$. Depending on the distribution of $\mathbf{D}$ over its different possible values (decoder functions), there may be disagreement between the estimated distributions given a fixed input $\mathbf{x}$. That is, it is possible that $\hat{p}_1(t \mid \mathbf{x}) \neq \hat{p}_2(t \mid \mathbf{x}) \neq \ldots$. This disagreement represents the epistemic uncertainty.

Given the complexities involved, we will consider just a single decoder in later sections: $\mathbf{D} = d(\cdot)$. This can be done by simply choosing a single decoder from a set of possibilities, or by creating a composite decoder by from an ensemble average.

Ensemble averaging of decoders By averaging over many possible decoders, it may be possible to arrive at a single decoder with better performance than any individual. Mathematically, using $d_i(\mathbf{x})$ to represent the ith decoder in an ensemble:

$$d_{\text{ensemble}}(\mathbf{x}) = \sum_{\boldsymbol{\mathcal{D}}} d_i(\mathbf{x}) = \sum_{\boldsymbol{\mathcal{D}}} \hat{p}_i(t \mid \mathbf{x}) = \hat{p}_{\text{ensemble}}(t \mid \mathbf{x})$$

In many cases, ensemble averaging can improve performance [74]. Such a procedure can reduce epistemic uncertainty by averaging out independent errors from different types of encoders. The ensemble could be different classes of decoders (e.g. linear, polynomial, etc.) or different parameter settings within a single class of decoders (e.g. multiple neural networks with different weights).

Explicit modeling of uncertainty over decoders Given the extra complication it introduces to have to consider an ensemble of possible decoders, is useful to explicitly model it? One argument for doing so is based partially on the ensemble averaging approach discussed above: by creating multiple decoders, one can compute an average over them, which could get closer to the "optimal" decoder . However, it is quite possible that even with this averaging, one cannot never approach the true decoder because it lies outside the class of decoders

being considered. For example, if the ensemble of decoders are all linear functions, averaging many together will never be able to reach a true decoder that is a nonlinear function.

In practice, many advocate for explicitly representing uncertainty over the choice of decoder, because in addition to an ensemble average potentially improving performance, a large degree of disagreement between decoders can indicate that the process of constructing decoders needs to be improved. Importantly, this is distinct from the question of the level of aleatoric uncertainty in the estimate. An ensemble of decoders can agree or disagree that the estimation problem is uncertain or not.

Epistemic uncertainty

Having chosen to ignore decoder choice uncertainty and focus on a fixed choice of decoder, epistemic uncertainty reduces to task-irrelevant variations in image caused by the object or detection noise "leak" into the the decoder's predictions.

Predictivity

Having chosen to focus on a single decoder at a time, it is safe to assume that this decoder will deviate in some way from the optimal decoder $d^*(\cdot)$. However, some deviations will be worse than others. Furthermore, as will be shown subsequently, a perfectly sufficient, minimal decoder is not unique. We thus need another means of criticizing and comparing different decoders, in order to select the best one. The way to do this is to determine whether the decoder is **predictive** [10, 131, 36].

If a decoder is predictive, it means that its predicted distribution $\hat{p}(t \mid \mathbf{x})$ will place probability in such a way that it would match the distribution of future data drawn from the same distribution. In other words, the values it predicts are realistic ones. This can be considered in the per-input case or in aggregate over the whole distribution of predictions. However, the former can't actually be evaluated in practice, so we must settle for the latter.

Theoretical evaluation

Sufficiency, minimality, and productivity all in measure in some way discrepancies between the estimated and true distributions. There is a quite satisfying mathematical way of describing this, which seems quite impossible to calculate in practice. If it were possible, we wouldn't need three separate criteria–just this one would suffice. The three criteria can in some sense be though of as low-dimensional projections of a higher dimensional truth.

In the context of information theory, KL divergence can be used to measure the discrepancy between two probability distributions. Given two probability mass functions $p(y)$ and $q(y)$, the KL divergence is:

$$D_{KL}(p(y) \parallel q(y)) = \sum_{\mathcal{Y}} p(y) \frac{p(y)}{q(y)}$$

$D_{KL}(p(y) \parallel q(y))$ will be 0 when the distributions are identical, and > 0 otherwise.

For the specific case of assessing a single prediction of a stochastic decoder, the relevant KL divergence would be either:

$$D_{KL}(\hat{p}(t \mid \mathbf{X}) \parallel p(t \mid \mathbf{X}))$$
(Forward KL divergence)

or

$$D_{KL}(p(t \mid \mathbf{X}) \parallel \hat{p}(t \mid \mathbf{X}))$$
(Reverse KL divergence)

KL divergence is not symmetric in its arguments, so these two expressions might give different values. The forward KL divergence incentivizes the predicted distribution to avoid having low probability mass wherever there is probability mass on the true distribution, while the reverse KL-divergence preferences the predicted distribution to put high probability mass where ever the true distribution has high probability mass. It is also possible to balance these two incentives by averaging the forward and reverse KL divergences, which yields a quantity known as the Jensen-Shannon divergence.

The units of KL divergence (assuming base-2 logarithms) are bits and thus can be directly compared to entropy. It is thus meaningful (given a choice of forward KL, reverse KL, or Jensen-Shannon divergence) to state something like, "this prediction problem has an aleatoric uncertainty of 5.4 bits, and the decoder used has an epistemic uncertainty of 4.3 bits". Or rather, it would meaningful, if this could be computed in practice, which sadly it cannot, because the true distribution $p(t \mid \mathbf{X})$ is unknown (more on this shortly).

Throughout this discussion, uncertainty has been synonymous with additional variations that carry no useful information. Thus the fact that the discrepancy between the true and predicted distribution represents an uncertainty is easiest to understand in the case where the predicted distribution $\hat{p}(t \mid \mathbf{x})$ is over-dispersed compared to the the true distribution (**Fig. 6.2b, Top**). In the other cases, over-concentration and bias (**Fig. 6.2b, Middle, Bottom**), the predicted distribution actually has *fewer* variations. To make sense of this, one must consider the ensemble perspective: that one decoder is an element of a set of many possible decoders. If there is a discrepancy between the prediction of a decoder and the true distribution, it is because the optimal decoder $d^*(\cdot)$ is not being used. The variations typically associated with uncertainty are thus distributed over multiple possibilities of which decoder to use.

Thus far, we quantified epistemic uncertainty for just a single prediction based on the image $\mathbf{x}$. Taking a weighted average over all possible images gives a full characterization of the decoder's performance. Mathematically:

$$\mathbb{E}\left[D_{KL}(\hat{p}(t \mid \mathbf{X}) \parallel p(t \mid \mathbf{X}))\right]$$

If this expectation is 0, it means that our decoder is perfect: There is no epistemic uncertainty and it is making each prediction as precisely as the problem's aleatoric uncertainty allows. That is, all task-relevant information has been extracted, and there are no additional variations in the predictions.

Unfortunately, as mentioned previously, due to a lack of knowledge of the true distribution $p(t \mid \mathbf{X})$, this cannot be evaluated in practice. Thus we must employ other means of evaluating whether the optimality criteria for decoders are satisfied. In order to do this, we must move from considering the conditional distributions (i.e. $p(t \mid \mathbf{x})$), which describe the behavior of individual predictions, to marginal distributions (i.e. $p(t)$), which describe these predictions averaged over all images.

Practical evaluation

True marginal distribution

The first marginal distribution to consider is that of the ground truth values for the task, which is modeled by the random variable T^*. For example, if the task is to predict the abundance of particular proteins on each cell, T^* would be the number of proteins present for a cell drawn from the population at random.

The amount of information needed (on average) to predict T^* exactly is given by the entropy $H(\mathrm{T}^*)$. As introduced in Section 6.5, the detected image $\mathbf{X}$ may not contain the full amount of information to predict this value exactly. Mathematically, this can be seen by decomposing $H(\mathrm{T}^*)$ into the sum of the aleatoric/inherent uncertainty for a given encoder and the task-relevant information in the detected image:

$$H(\mathrm{T}^*) = H(\mathrm{T}^* \mid \mathbf{X}) + I(\mathrm{T}^*; \mathbf{X})$$

These two quantities govern the information theoretic limits of a decoder. A decoder can extract no more task-relevant information from the image than $I(\mathrm{T}^*; \mathbf{X})$, and it cannot reduce uncertainty below $H(\mathrm{T}^* \mid \mathbf{X})$.

Estimated marginal distribution

Analogous to the marginal distribution of true values, there is a marginal distribution of the estimates produced by our decoder. Given an image $\mathbf{x}$, a stochastic decoder will produce a distribution $\hat{p}(t \mid \mathbf{x})$. A probability-weighted average over the distributions produced by the

decoder for every possible image $\mathbf{x}$ will give another distribution that can be interpreted as the *average* prediction (**Fig. 6.3a**). Mathematically:

$$\hat{\mathrm{T}} \sim \mathbb{E}\left[\hat{p}(t \mid \mathbf{X})\right] \tag{6.3}$$

$\hat{\mathrm{T}}$ is analogous to T^*, but while the latter is distributed according to the true value of the task, the former is distributed according to the predictions made by the decoder.

Sufficiency

A decoder that produces estimates with high sufficiency has successfully extracted information relevant to the task from the image. Mathematically, this is quantified by $I(\hat{\mathrm{T}}; \mathrm{T}^*)$. A decoder cannot extract more information than is present in the image itself:

$$I(\hat{\mathrm{T}}; \mathrm{T}^*) \leq I(\mathrm{T}^*; \mathbf{X})$$

Visualizing the joint distribution of $p_{\hat{\mathrm{T}},\mathrm{T}^*}$ provides insight into what it means to have information-rich predictions. When $I(\mathrm{T}^*; \mathbf{X}) = H(\mathrm{T}^*)$, the image contains enough information such that aleatoric uncertainty is 0. In this circumstance, the joint distribution produced by an optimal decoder would have all its probability mass concentrated along the diagonal (**Fig. 6.3b, left**). Drawing a vertical line corresponding to a specific value of $\mathrm{T}^* = t$ would trace out a conditional distribution $p_{\hat{\mathrm{T}}|\mathrm{T}^*=t}$, which has all its probability mass on the same value $\hat{\mathrm{T}} = t$. This means that whenever $\mathrm{T}^* = t$, the decoder would have predicted the value t with probability 1. It makes the correct prediction each time with no uncertainty.

At the other end of the spectrum, predictions containing no information will produce a joint distribution of $\mathrm{T}^*, \hat{\mathrm{T}}$ that is the product of the marginal distributions T^* and $\hat{\mathrm{T}}$ (**Fig. 6.3b, right**). Using the same procedure as before of drawing a vertical line for a specific value of $\mathbf{T^*} = t$, we find the distribution of $p_{\hat{\mathrm{T}}|\mathrm{T}^*=t}$ is identical to $p_{\hat{\mathrm{T}}}$. This means that decoder's estimate does not change at all when T^* is t, and ample probability mass is placed on incorrect values.

Minimality

A decoder that produces estimates that are minimal has successfully discarded epistemic uncertainty. Such irrelevant information can arise from a variety of sources. For example, it could be variations in the object that carry no information about the task or detection noise. It can also be expressed in a variety of equivalent information-theoretic terms. For example: $I(\mathbf{X}; \hat{\mathrm{T}} \mid \mathrm{T}^*)$ or $H(\hat{\mathrm{T}} \mid \mathbf{X}) - H(\hat{\mathrm{T}})$.

Figure 6.5 shows an information-theoretic view of a sufficient, but not minimal encoder.

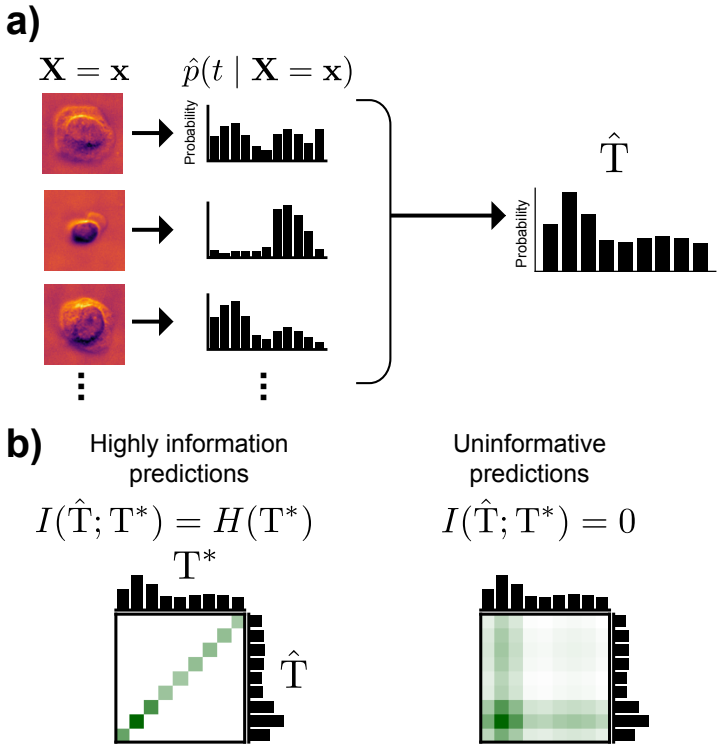

Figure 6.3: **a)** The marginal estimated distribution is found by taking a probability weighted average over many individual estimated distributions. **b)** Left, a perfectly sufficient marginal distribution (with no aleatoric uncertainty). Right, an estimated marginal distribution that carries no information about the true value.

The information bottleneck

The information bottleneck [158] is a well-known formalism for expressing the trade-off between minimality and sufficiency. The "bottleneck" portion refers to information being "squeezed" as it passes from $\mathbf{X}$ to $\hat{\mathrm{T}}$. Ideally, the information lost in the bottleneck is unrelated to T^*, but relevant information may be lost as well. This can be visualized by plotting $I(\hat{\mathrm{T}}; \mathrm{T}^*)$ vs. $I(\hat{\mathrm{T}}; \mathbf{X})$ (**Fig. 6.4**).

One shortcoming of the information bottleneck in the context of this model is that it does not require interpretability. The distribution of $\hat{\mathrm{T}}$ could take a variety of forms and still perform optimally under its criteria. This shortcoming is the reason that predictivity is needed as a third factor.

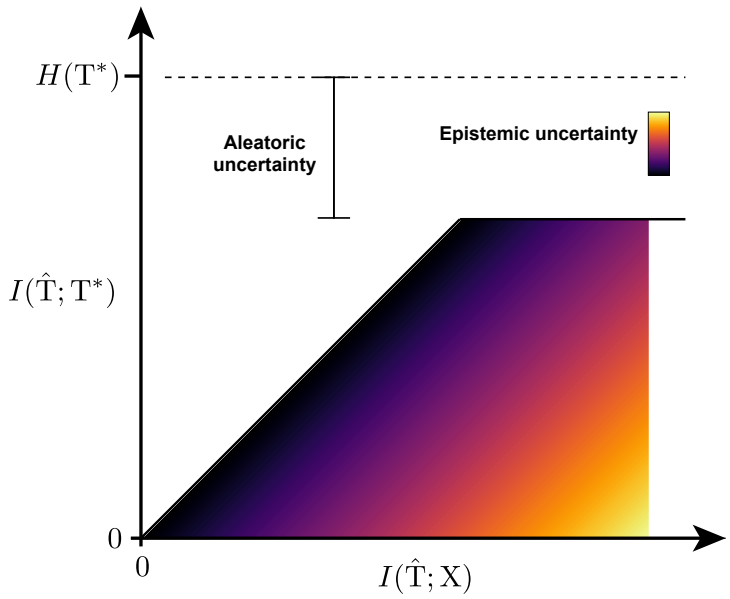

Figure 6.4: **The information bottleneck**

Predictivity

While sufficiency quantifies the ability to discriminate different values of the task, it says nothing about whether the individual predicted values are actually correct. For example, if the rows of the joint distributions shown in **Figure 6.3b** were randomly permuted, $I(\mathrm{T}^*;\mathbf{X})$ would remain unchanged.

If a decoder is optimally predictive, $H(\hat{\mathrm{T}})$ will be equal to $H(\mathrm{T}^*)$, though converse is not necessarily true.

Putting it all together

Sufficiency, Minimality, and Predictivity are all interrelated. Figure 6.5 shows the entropies of the various relevant quantities. Figure 6.6 shows true and estimated marginal distributions with differing levels of predictivity and minimality.

6.6 Concluding thoughts

In this chapter, a generic model of a computational imaging system has been presented through the lens of information theory. Like every model, it is undoubtedly wrong. However, it may prove quite useful. This is left to future work.

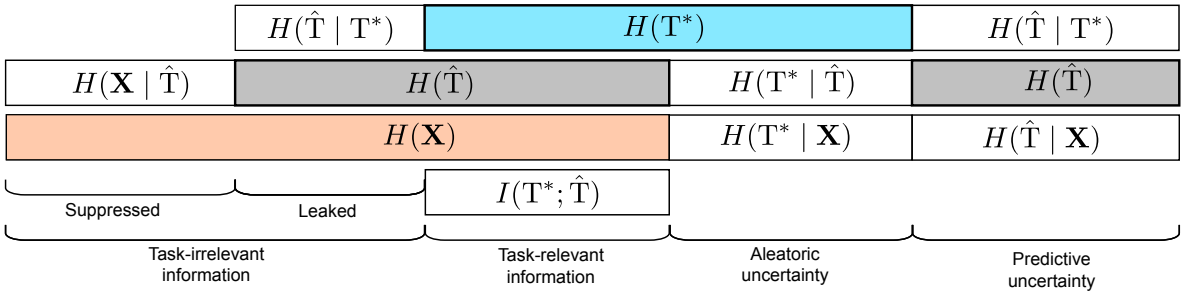

Figure 6.5: **Information-theoretic view of a sufficient, non-minimal decoder**. Horizontal bars represent entropy, with vertical overlap representing shared entropy.

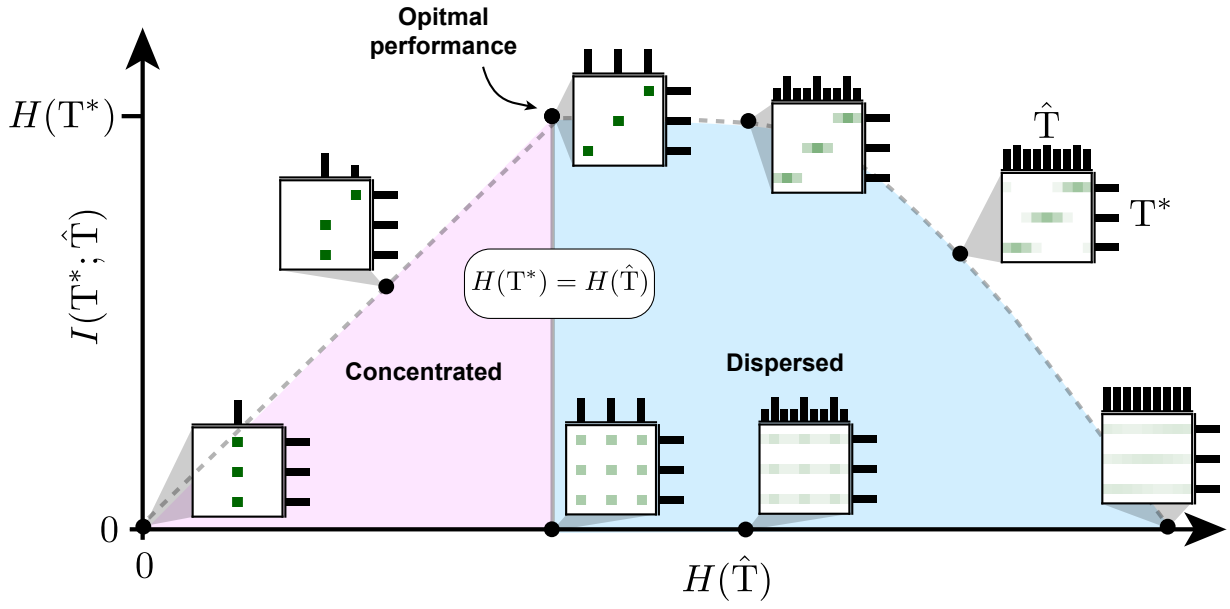

Figure 6.6: **The joint (marginal) true and estimated distributions** Sufficiency is shown on the y-axis of the outer plot. The x-axis shows the entropy of the marginal predicted distribution. In between concentrated and dispersed areas, decoders with perfect predictivity are found. Increasingly dispersed distributions have greater epistemic uncertainty. Either over-concentration or over-dispersion will eventually reduce sufficiency.

Chapter 7

Future work

The work presented in this dissertation, along with the concurrent work of others have opened many exciting new possibilities for future research in computational microscopy. Here, we briefly discuss several of these.

7.1 Adaptive biological microscopy

The techniques for autofocusing and adaptive illumination presented in chapters 2 and 3 are examples of adaptive microscopy, in which an algorithm interprets images and decides how to control the microscope. These techniques were built upon recent advances in computer vision and open source microscope control software, which have both advanced considerably in recent years, creating many new possibilities.

One area of application is fluorescence microscopy of biological samples. Often, and especially in super resolution fluorescence microscopy, biological samples are subjected to damaging, high-intensity illumination light. Selectively applying these techniques at their full capabilities at the right times and locations enables the capture of a larger number of rare events. The automation of this process enables the capture of events that manual human control would not be able to capture in real-time.

Furthermore, the ease of developing algorithms for adaptive microscopy experiments is poised to continue being supercharged by the power of deep neural networks. Neural networks possess an impressive ability to learn functions from training data. This means adaptive microscopy techniques no longer require the careful design of customized image processing algorithms that determine what and where to image; Instead, labelling some representative examples is sufficient to teach the microscope to start looking for more instances of a particular phenotype. In addition, neural networks can be executed quickly–on the order of milliseconds–such that real-time adaptation of imaging parameters is possible. The days of researchers "babysitting" experiments may be numbered.

But the possibilities don't stop there. In contrast to "supervised" approaches, where instances of the pattern of interest must be explicitly provided (**Fig. 7.1a**), deep neural net-

works can be used as "unsupervised" probabilistic models (called "deep generative modes") that learn the distribution of a particular type of images. This means that by feeding a neural network a large amount of unlabelled data, it can learn to model the inherent biological heterogeneity–which images are rare, which are common, and the shared characteristics over which they vary. This could be deployed to make adaptive microscopes that themselves discover which phenotypes are surprising and proceed to gather more detailed information on them (**Fig. 7.1b**). And because computers are often better at pattern-matching than people, they may be able to make new discoveries that were previously overlooked by their human counterparts.

Deep reinforcement learning algorithms are another promising neural network-based controller for adaptive microscopes. They learn not just patterns in images, but also strategies to take actions based on these patterns. They gained attention in recent years for their ability to beat expert human players at games like Chess and Go, and they have potentially powerful biological applications as well. For example, one can imagine agents that automatically learn to optimize cell culture conditions or employ existing optical tools that manipulate intracellular signalling [159] or transcription [132] to better understand biological processes or control them to create biological products (**Fig. 7.1c**).

Neural networks are only as good as the data fed to them. They are quite capable of picking up and even amplifying human biases in the training data provided to them. If they are entrusted to decide what data to capture and what experiments to do, these biases must be measured and appropriately mitigated.

7.2 Information-optimal microscopy

The application of the information-theoretic framework described in the previous chapter opens many new possibilities. While a substantial effort of previous research has focused on increasing the *maximum* amount of information that a computationally synthesized image might contain [105], to the best of my knowledge, none has directly focused on how to actually capture that information in practice. That is, we have many ways to create big bottles (data), but very little idea how to fill them with lots of water (information).

The reason this was not previously possible is the lack of probabilistic models for each stage of an imaging system. Shannon's information theory can only be applied when considering distributions over multiple possible samples on a microscope. Traditionally, most research has focused on benchmarking performance on just one standardized sample. The model presented in the previous chapter alleviates this limitation.

Information-optimal point-spread functions

Information theory provides a clear answer to the question of which point spread function is optimal. In the absence of a particular task that requires only specific features of the object, the optimal point spread function is the one that captures the most information about the

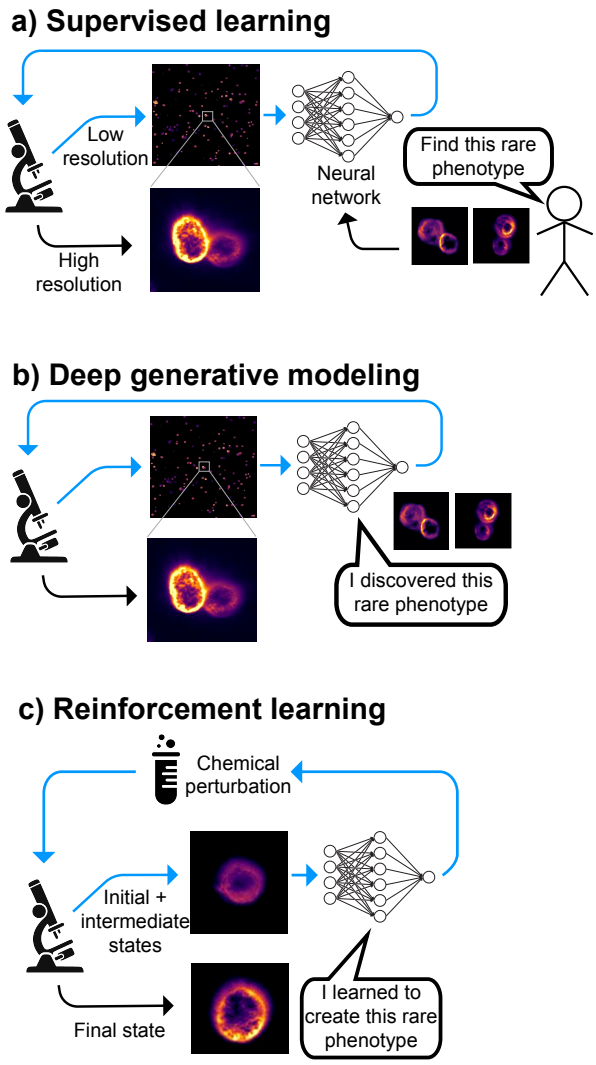

Figure 7.1: **a)** Using supervised deep learning, a human labels examples of a rare phenotype, and a neural network locates similar cells and controls the microscope to image them at higher resolution. **b)** Using deep generative modeling, the neural network can itself discover which phenotypes are rare and image them at higher resolution. **c)** Using deep reinforcement learning, the neural network learns how to chemically perturb cells to produce a particular phenotype.

object: $I(\mathbf{X};\mathbf{O}^*)$. Alternatively, when a specific task is defined, the relevant information is found in $I(\mathbf{X};\mathbf{T}^*)$.

It is worth noting that a similar line of research attempting to develop point spread functions that maximize the precision with which single point emitters can be localized [107, 142, 41, 126] has been described in similar terms. However, this work utilizes Fisher Information, which, despite sharing the same name, is an entirely different concept from the idea of information/entropy in information theory.

Noiseless images Computing $I(\mathbf{X};\mathbf{O}^*)$ can be considered with several layers of complexity. In the simplest case, one can assume noiseless detection. In this case, $\mathbf{X} = \mathbf{X}^*$. Under this assumption, the image can carry no more information about the object than it itself contains. That is, $I(\mathbf{X};\mathbf{O}^*) < H(\mathbf{X}^*)$. If we consider only a single, fixed system, whose image formation process is known, there will be no additional sources of uncertainty in $H(\mathbf{X}^*)$. Thus, $I(\mathbf{X};\mathbf{O}^*) = H(\mathbf{X}^*)$. This means that by measuring the amount of information in the image, we can know how much information we've ascertained about the object. By comparing different image formation models (e.g. systems with different point spread functions), we can ascertain which provides the most information about the object (i.e. the sample being imaged). This investigation is particularly well suited to simulations of optical systems, because noiseless images of a sample under different point spread functions can readily be generated.

The challenge lies in the fact that computing $H(\mathbf{X}^*)$ involves performing an integration over a number of dimensions equal to the number of pixels in the image. For any reasonably-sized image, this operation is computationally intractable.

Luckily, there are several ways we might approximate or bound this value. First, we can assume that every pixel in the image is independent. In this case, for an N-pixel image, we can compute N 1-dimensional integrals rather than 1 N-dimensional integral. This will provide an upper bound on the information in the images. It will tell us when we're doing badly, but not necessarily when we're doing well, because it doesn't consider what information is unique in each pixel and what information is redundant between pixels.

Alternatively, we can make the assumption that the images are distributed according to a multivariate Gaussian distribution, the entropy of which can be computed much more easily by first computing a covariance matrix over image pixels. This assumption is certainly not true in practice, but it remains unclear (to me) how much approximation error it would introduce.

Finally, and perhaps most promisingly, an upper bound on the the entropy of a distribution of images can be computed by fitting a generative model to the distribution of data. A model $p_\theta(x)$ can be fit to the true distribution $p(x)$ (as approximated through an empirical distribution of images). The value of a standard maximum likelihood loss function represents the cross entropy between the model distribution and the true distribution. Cross entropy $H(p_\theta, p)$ can be decomposed as:

$$H(p, p_\theta) = H(p) + D_{KL}(p \parallel p_\theta)$$

The second term, $D_{KL}(p \parallel p_\theta)$, measures the similarity between the true distribution and the model distribution. If a perfect model of the true distribution is learned, this term will be to zero and the cross entropy loss will equal the entropy of the data distribution. This can be used to measure the entropy of an arbitrary distribution of images generated with a particular point spread function.

Still, there remains the challenge of finding a model that can perfectly fit the true distribution. It is a widely held belief that by using an extremely flexible model (i.e. a neural network) with enough training data and enough computational power, that any arbitrary distribution can be fit. Empirical studies suggest that the most powerful neural networks continue to reach new state-of-the-art performance with increasing data and compute [52]. Thus, the tightness of the bound produced by this method can be expected to continue to improve into the future.

Even with the ability to perfectly estimate the entropy of a distribution of images with this procedure, there remains the problem of optimizing the optical systems based on this criterion. This is challenging because each measurement of entropy requires the complete generation of a dataset of images and the training of a deep generative model, a process which can take days-weeks with state of the art models. Moving this technique beyond the realm simply picking a few promising optical system designs and measuring the amount of information they encode into automated search over many classes of optical systems will require a performant way of performing this outer loop search.

There are many analogs of this type of problem in other fields, and thus borrowing from these techniques could be fruitful. For example, the problem of neural architecture search, in which the optimal architecture for a neural network is selected based on training networks with different architectures on the same data has an analogous outer loop over a time intensive inner loop of training an entire network from scratch. Nonetheless, state-of-the-art results have been achieved using a reinforcement learning agent that can intelligently search over the parameter space architectures [184].

Noise-robust information gathering In practice, detected images will always contain noise. That is $\mathbf{X}^* \neq \mathbf{X}$. Thus, it will be necessary to account for its effects when designing information optimal systems. Under some relatively mild assumptions, a computationally tractable mathematical model of this process can be formulated.

The task-relevant information in a noisy image can be written as:

$$I(\mathbf{T}^*, \mathbf{X}) = H(\mathbf{X}) - H(\mathbf{X} \mid \mathbf{T}^*)$$

$H(\mathbf{X})$ can be estimated in the same manner as previously described. Next, a low dimensional task and a deterministic relationship between task and object can be assumed.

For example, the task could simply be an index into a finite set of objects. This allows the decomposition:

$$\begin{aligned} H(\mathbf{X} \mid \mathrm{T}^*) &= \sum_{\mathcal{T}} \sum_{\boldsymbol{X}} p(\mathbf{x}, t^*) \log p(\mathbf{x} \mid t^*) \\ &= \sum_{\mathcal{T}} \sum_{\boldsymbol{X}} p(\mathbf{x} \mid t^*) p(t^*) \log p(\mathbf{x} \mid t^*) \end{aligned}$$

If there is a one-to-one relationship between noiseless images and task value indices (a fact that is nearly assured if $|\mathcal{T}|$ is not too large and $\mathbf{X}$ is much higher dimensional), then $p(\mathbf{x}^* \mid t^*)$ is one when the function mapping t to a noiseless image is equal to the particular noiseless image $\mathbf{x}^*$ and zero otherwise. Furthermore, since the noise at every pixel is independent, $p(\mathbf{x}^* \mid \mathbf{x})$ can be rewritten as:

$$\begin{aligned} &p(\mathbf{x}^* \mid \mathbf{x}) \\ &\qquad = \prod_i p(x_i^* \mid x_i) \\ &\qquad = \prod_i p(x^* \mid x) \end{aligned}$$

Where $p(x^* \mid x)$ is the noise model for a single pixel. Plugging this in to the previous expression:

$$\begin{aligned} &\sum_{\mathcal{T}} \sum_{\boldsymbol{\mathcal{X}}} p(\mathbf{x} \mid t^*) p(t^*) \log p(\mathbf{x} \mid t^*) \\ &= \sum_{\mathcal{T}} \sum_{\mathcal{X}^*} \sum_{\boldsymbol{X}} p(\mathbf{x} \mid \mathbf{x}^*) p(\mathbf{x}^* \mid t^*) p(t^*) \log p(\mathbf{x} \mid \mathbf{x}^*) p(\mathbf{x}^* \mid t^*) \end{aligned}$$

Plugging in the above assumptions to this expression should yield a computationally tractable estimator for the amount of task-relevant information present in a noisy image.

Though the two ideas in the previous two sections have strong theoretical motivations, it will be important also to ground their findings in empirical analysis. For example, does gathering images with more information practically improve performance on classification tasks? Related to this, there is much discussion in the information theory literature about the difficulty of creating estimators for information-theoretic quantities with desirable theoretical qualities. How much might this theoretical difficulty impact the practical application of such techniques in computational microscopy.

Information-optimal decoders The above two sections describe how to calculate the amount of information in noisy and noiseless images, respectively. There is another important question of how to decode this information in a useful way. Developing performant

decoders, especially for problems with aleatoric uncertainty, requires the use stochastic decoders. While the optimality criteria for such decoders were outlined in the previous chapter, practical strategies for developing such decoders are also needed. This will be easiest on low-dimensional tasks where these optimality criteria can be explicitly computed by brute force methods.

The dataset described in chapter 5 provides the necessary data, and a practically relevant problem (protein prediction from label-free images) to perform this type of research. Developing robust learning algorithms for stochastic decoders may lead to powerful, uncertainty-aware predictors that will be essential for challenging, real-world problems.

Bibliography

[1] Martin Abadi et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems*. Tech. rep. Google Research, 2015. URL: http://download.tensorflow.org/paper/whitepaper2015.pdf.

[2] Patrick Autissier et al. "Evaluation of a 12-color flow cytometry panel to study lymphocyte, monocyte, and dendritic cell subsets in humans". In: *Cytometry Part A* 77.5 (2010), pp. 410–419. ISSN: 15524922. DOI: 10.1002/cyto.a.20859.

[3] Boris Babenko et al. "Poverty Mapping Using Convolutional Neural Networks Trained on High and Medium Resolution Satellite Images, With an Application in Mexico". In: *arXiv* (Nov. 2017). URL: http://arxiv.org/abs/1711.06323.

[4] Vladimir P. Badovinac, Jodie S. Haring, and John T. Harty. "Initial T Cell Receptor Transgenic Cell Precursor Frequency Dictates Critical Aspects of the CD8+ T Cell Response to Infection". In: *Immunity* 26.6 (2007), pp. 827–841. ISSN: 10747613. DOI: 10.1016/j.immuni.2007.04.013.

[5] M. Bathe-Peters, P. Annibale, and M. J. Lohse. "All-optical microscope autofocus based on an electrically tunable lens and a totally internally reflected IR laser". In: *Optics Express* 26.3 (2018), p. 2359. ISSN: 1094-4087. DOI: 10.1364/OE.26.002359. URL: https://www.osapublishing.org/abstract.cfm?URI=oe-26-3-2359.

[6] A. J. Bell and T. J. Sejnowski. "An information-maximization approach to blind separation and blind deconvolution." In: *Neural computation* 7.6 (1995), pp. 1129–1159. ISSN: 08997667. DOI: 10.1162/neco.1995.7.6.1129. URL: http://www.ncbi.nlm.nih.gov/pubmed/7584893.

[7] Neil J. Bershad. "Resolution, Optical-Channel Capacity and Information Theory". In: *Journal of the Optical Society of America* 59.2 (Feb. 1969), p. 157. ISSN: 0030-3941. DOI: 10.1364/JOSA.59.000157. URL: https://www.osapublishing.org/abstract.cfm?URI=josa-59-2-157.

[8] Eric Betzig et al. "Imaging intracellular fluorescent proteins at nanometer resolution." In: *Science (New York, N.Y.)* 313.5793 (2006), pp. 1642–1645. ISSN: 0036-8075. DOI: 10.1126/science.1127344.

[9] Joseph N Blattman et al. "Estimating the Precursor Frequency of Naive Antigen-specific CD8 T Cells". In: *The Journal of experimental medicine* 195.5 (2002).

[10] David M Blei. "Posterior Predictive Checks". In: *Cos597a* (2011), pp. 1–8.

[11] Vivek Boominathan et al. "Lensless Imaging: A computational renaissance". In: *IEEE Signal Processing Magazine* 33.5 (2016), pp. 23–35. ISSN: 10535888. DOI: 10.1109/MSP.2016.2581921.

[12] James Bradbury et al. *{JAX}: composable transformations of {P}ython+{N}um{P}y programs*. 2018. URL: http://github.com/google/jax.

[13] Anna Brewitz et al. "CD8+ T Cells Orchestrate pDC-XCR1+ Dendritic Cell Spatial and Functional Cooperativity to Optimize Priming". In: *Immunity* 46.2 (2017), pp. 205–219. ISSN: 10974180. DOI: 10.1016/j.immuni.2017.01.003.

[14] Mauro Buttarello and Mario Plebani. "Automated blood cell counts: State of the art". In: *American Journal of Clinical Pathology* 130.1 (2008), pp. 104–116. ISSN: 00029173. DOI: 10.1309/EK3C7CTDKNVPXVTN.

[15] E.J. Candes and M.B. Wakin. "An Introduction To Compressive Sampling". In: *IEEE Signal Processing Magazine* 25.2 (2008), pp. 21–30. ISSN: 1053-5888. DOI: 10.1109/MSP.2007.914731.

[16] Anne E Carpenter et al. "CellProfiler: image analysis software for identifying and quantifying cell phenotypes." In: *Genome biology* 7.10 (2006), R100. ISSN: 1465-6914. DOI: 10.1186/gb-2006-7-10-r100.

[17] Claire Lifan Chen et al. "Deep Learning in Label-free Cell Classification". In: *Scientific Reports* 6.August 2015 (2016), p. 21471. ISSN: 2045-2322. DOI: 10.1038/srep21471. URL: http://www.nature.com/articles/srep21471.

[18] Michael Chen, Zachary F. Phillips, and Laura Waller. "Quantitative differential phase contrast (DPC) microscopy with computational aberration correction". In: *Optics Express* 26.25 (2018), p. 32888. ISSN: 1094-4087. DOI: 10.1364/oe.26.032888.

[19] Tong-Sheng Chen et al. "High-order photobleaching of green fluorescent protein inside live cells in two-photon excitation microscopy." In: *Biochemical and biophysical research communications* 291.5 (2002), pp. 1272–1275. ISSN: 0006-291X. DOI: 10.1006/bbrc.2002.6587.

[20] François Chollet et al. *Keras*. \url{https://github.com/fchollet/keras}. 2015.

[21] Kengyeh K. Chu, Daryl Lim, and Jerome Mertz. "Two-photon microscopy with adaptive illumination power". In: *Biomedical Optics, BIOMED 2008* 1 (2008), pp. 1–3. DOI: 10.1364/biomed.2008.bmd55.

[22] William S. Cleveland. "Robust locally weighted regression and smoothing scatterplots". In: *Journal of the American Statistical Association* 74.368 (1979), pp. 829–836. ISSN: 1537274X. DOI: 10.1080/01621459.1979.10481038.

[23] Thomas M Cover and Joy A. Thomas. *Elements of Information Theory*. 2nd. Wiley Series in Telecommunications. New York, USA: Wiley, Sept. 2005. ISBN: 9780471241959. DOI: 10.1002/047174882X. URL: http://staff.ustc.edu.cn/~mfy/InfoTheory/Complements/Elements%20of%20Information%20Theory%202nd.pdf%20http://doi.wiley.com/10.1002/0471200611%20https://onlinelibrary.wiley.com/doi/book/10.1002/047174882X.

[24] I. J. Cox and C. J. R. Sheppard. "Information capacity and resolution in an optical system". In: *Journal of the Optical Society of America A* 3.8 (Aug. 1986), p. 1152. ISSN: 1084-7529. DOI: 10.1364/JOSAA.3.001152. URL: https://www.osapublishing.org/abstract.cfm?URI=josaa-3-8-1152.

[25] Warren N. D'Souza and Stephen M. Hedrick. "Cutting Edge: Latecomer CD8 T Cells Are Imprinted with a Unique Differentiation Program". In: *The Journal of Immunology* 177.2 (2006), pp. 777–781. ISSN: 0022-1767. DOI: 10.4049/jimmunol.177.2.777.

[26] Jia Deng et al. "ImageNet: A large-scale hierarchical image database." In: *Cvpr* (2009), pp. 248–255. ISSN: 1063-6919. DOI: 10.1109/CVPRW.2009.5206848. URL: http://ieeexplore.ieee.org/iel5/5191365/5206488/05206848.pdf?arnumber=5206848%5Cnhttp://ieeexplore.ieee.org/articleDetails.jsp?arnumber=5206848%5Cnhttp://dx.doi.org/10.1109/CVPRW.2009.5206848%5Cnhttp://www.dblp.org/rec/bibtex/conf/cvpr/DengDSLL009.

[27] Regina Eckert, Zachary F. Phillips, and Laura Waller. "Efficient illumination angle self-calibration in Fourier ptychography". In: *Applied Optics* 57.19 (2018), p. 5434. ISSN: 1559-128X. DOI: 10.1364/ao.57.005434.

[28] Arthur Edelstein et al. "Computer Control of Microscopes Using μManager". In: *Current Protocols in Molecular Biology* 92.1 (Oct. 2010), pp. 1–17. ISSN: 1934-3639. DOI: 10.1002/0471142727.mb1420s92. URL: https://onlinelibrary.wiley.com/doi/10.1002/0471142727.mb1420s92.

[29] Arthur D Edelstein et al. "Advanced methods of microscope control using MicroManager software". In: *Journal of Biological Methods* 1.2 (Nov. 2014), p. 10. ISSN: 2326-9901. DOI: 10.14440/jbm.2014.36. URL: http://www.jbmethods.org/jbm/article/view/36.

[30] Michael Eisenstein. "Smart solutions for automated imaging". In: *Nature Methods* 17.November (2020), pp. 1–5. ISSN: 1548-7091. DOI: 10.1038/s41592-020-00988-2. URL: http://www.nature.com/articles/s41592-020-00988-2.

[31] Andre Esteva et al. "Dermatologist-level classification of skin cancer with deep neural networks". In: *Nature* 542.7639 (2017), pp. 115–118. ISSN: 0028-0836. DOI: 10.1038/nature21056. URL: http://dx.doi.org/10.1038/nature21056.

[32] Philipp Eulenberg et al. "Deep Learning for Imaging Flow Cytometry: Cell Cycle Analysis of Jurkat Cells". In: *bioRxiv* (2016), pp. 1–13. URL: http://biorxiv.org/content/early/2016/10/17/081364.abstract.

[33] Jean Luc Faucher et al. ""6 Markers/5 colors" extended white blood cell differential by flow cytometry". In: *Cytometry Part A* 71.11 (2007), pp. 934–944. ISSN: 15524922. DOI: 10.1002/cyto.a.20457.

[34] P.B. Fellgett and E.H. Linfoot. "On the assessment of optical images". In: *Philosophical Transactions of the Royal Society of London. Series A, Mathematical and Physical Sciences* 247.931 (Feb. 1955), pp. 369–407. ISSN: 0080-4614. DOI: 10.1098/rsta.1955.0001. URL: https://www.jstor.org/stable/10.2307/j.ctt211qv60.7%20https://royalsocietypublishing.org/doi/10.1098/rsta.1955.0001.

[35] I. M. Gel'fand and A.M. Yaglom. "Calculation of the amount of information about a random function contained in another such function". In: Dec. 1959, pp. 199–246. DOI: 10.1090/trans2/012/09. URL: http://www.ams.org/trans2/012.

[36] Andrew Gelman and Cosma Rohilla Shalizi. "Philosophy and the practice of Bayesian statistics". In: *British Journal of Mathematical and Statistical Psychology* 66.1 (Feb. 2013), pp. 8–38. ISSN: 00071102. DOI: 10.1111/j.2044-8317.2011.02037.x.

[37] Audrey Gérard et al. "Detection of rare antigen-presenting cells through T cell-intrinsic meandering motility, mediated by Myo1g". In: *Cell* 158.3 (2014), pp. 492–505. ISSN: 10974172. DOI: 10.1016/j.cell.2014.05.044.

[38] Audrey Gérard et al. "Secondary T cell-T cell synaptic interactions drive the differentiation of protective CD8+ T cells." In: *Nature immunology* 14.4 (2013), pp. 356–63. ISSN: 1529-2916. DOI: 10.1038/ni.2547. URL: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3962671&tool=pmcentrez&rendertype=abstract.

[39] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. "Explaining and Harnessing Adversarial Examples". In: (Dec. 2014). URL: http://arxiv.org/abs/1412.6572.

[40] Logan Grosenick, James H. Marshel, and Karl Deisseroth. "Closed-loop and activity-guided optogenetic control". In: *Neuron* 86.1 (2015), pp. 106–139. ISSN: 10974199. DOI: 10.1016/j.neuron.2015.03.034. URL: http://dx.doi.org/10.1016/j.neuron.2015.03.034.

[41] Ginni Grover, Sri Rama Prasanna Pavani, and Rafael Piestun. "Performance limits on three-dimensional particle localization in photon-limited microscopy". In: *Optics Letters* 35.19 (Oct. 2010), p. 3306. ISSN: 0146-9592. DOI: 10.1364/OL.35.003306. URL: https://opg.optica.org/abstract.cfm?URI=ol-35-19-3306.

[42] Peter Grunwald. *A tutorial introduction to the minimum description length principle.* 2004. ISBN: 0100000010. DOI: arXiv:math/0406077. URL: http://arxiv.org/abs/math/0406077.

[43] Kaikai Guo et al. "Microscopy illumination engineering using a low-cost liquid crystal display." In: *Biomedical optics express* 6.2 (2015), pp. 574–9. ISSN: 2156-7085. DOI: 10.1364/BOE.6.000574. URL: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=4354584&tool=pmcentrez&rendertype=abstract.

[44] T. E. Gureyev et al. "Complementary aspects of spatial resolution and signal-to-noise ratio in computational imaging". In: *Physical Review A* 97.5 (2018), pp. 1–14. ISSN: 24699934. DOI: 10.1103/PhysRevA.97.053819.

[45] Timur Gureyev, Yakov Nesterets, and Frank de Hoog. "Spatial resolution, signal-to-noise and information capacity of linear imaging systems". In: *Optics Express* 24.15 (2016), p. 17168. ISSN: 1094-4087. DOI: 10.1364/oe.24.017168.

[46] Timur E. Gureyev et al. "Signal-to-noise, spatial resolution and information capacity of coherent diffraction imaging". In: *IUCrJ* 5 (2018), pp. 716–726. ISSN: 20522525. DOI: 10.1107/S2052252518010941.

[47] M. G L Gustafsson. "Surpassing the lateral resolution limit by a factor of two using structured illumination microscopy". In: *Journal of Microscopy* 198.2 (2000), pp. 82–87. ISSN: 00222720. DOI: 10.1046/j.1365-2818.2000.00710.x.

[48] Julien Guy et al. "A 5-color flow cytometric method for extended 8-part leukocyte differential". In: *Cytometry Part B - Clinical Cytometry* 92.6 (2017), pp. 498–507. ISSN: 15524957. DOI: 10.1002/cyto.b.21524.

[49] Charles R. Harris et al. "Array programming with NumPy". In: *Nature* 585.7825 (2020), pp. 357–362. ISSN: 14764687. DOI: 10.1038/s41586-020-2649-2. URL: http://dx.doi.org/10.1038/s41586-020-2649-2.

[50] Jason Hataye et al. "Naive and Memory CD4+ T Cell Survival Controlled by Clonal Abundance". In: *Science* 312.April (2006), pp. 114–116.

[51] Fritjof Helmchen and Winfried Denk. "Deep tissue two-photon microscopy". In: *Nature methods* 2.12 (2005). DOI: 10.1038/NMETH818. URL: http://www.nature.com/nmeth/journal/v2/n12/abs/nmeth818.html.

[52] Tom Henighan et al. "Scaling Laws for Autoregressive Generative Modeling". In: (Oct. 2020). URL: http://arxiv.org/abs/2010.14701.

[53] Tiffany R Hensley et al. "Enumeration of Major Peripheral Blood Leukocyte Populations for Multicenter Clinical Trials Using a Whole Blood Phenotyping Assay". In: *Journal of Visualized Experiments* 67 (Sept. 2012), e4302. ISSN: 1940-087X. DOI: 10.3791/4302. URL: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3490252&tool=pmcentrez&rendertype=abstract%20http://www.jove.com/video/4302/enumeration-major-peripheral-blood-leukocyte-populations-for.

[54] R. A. Hoebe et al. "Controlled light-exposure microscopy reduces photobleaching and phototoxicity in fluorescence live-cell imaging". In: *Nature Biotechnology* 25.2 (2007), pp. 249–253. ISSN: 10870156. DOI: 10.1038/nbt1278.

[55] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. "Multilayer feedforward networks are universal approximators". In: *Neural Networks* 2.5 (Jan. 1989), pp. 359–366. ISSN: 08936080. DOI: 10.1016/0893-6080(89)90020-8. URL: https://linkinghub.elsevier.com/retrieve/pii/0893608089900208.

[56] Roarke Horstmeyer and Changhuei Yang. "Diffraction tomography with Fourier ptychography". In: *Optica* 3.8 (2015), pp. 1–22. ISSN: 2334-2536. DOI: 10.1364/OPTICA.3.000827. URL: http://arxiv.org/abs/1510.08756.

[57] Roarke Horstmeyer et al. "Convolutional neural networks that teach microscopes how to image". In: *arXiv* (Sept. 2017). URL: http://arxiv.org/abs/1709.07223.

[58] Roarke Horstmeyer et al. "Standardizing the resolution claims for coherent microscopy". In: *Nature Photonics* 10.2 (2016), pp. 68–71. ISSN: 1749-4885. DOI: 10.1038/nphoton.2015.279. URL: http://www.nature.com/doifinder/10.1038/nphoton.2015.279.

[59] Gao Huang et al. "Densely connected convolutional networks". In: *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017* 2017-Janua (2017), pp. 2261–2269. ISSN: 0022-4790. DOI: 10.1109/CVPR.2017.243.

[60] Eyke Hüllermeier and Willem Waegeman. *Aleatoric and epistemic uncertainty in machine learning: an introduction to concepts and methods*. Vol. 110. 3. Springer US, 2021, pp. 457–506. ISBN: 0123456789. DOI: 10.1007/s10994-021-05946-3. URL: https://doi.org/10.1007/s10994-021-05946-3.

[61] Na Ji. "Adaptive optical fluorescence microscopy". In: *Nature Methods* 14.4 (2017), pp. 374–380. ISSN: 15487105. DOI: 10.1038/nmeth.4218.

[62] Zhong Jingshan et al. "Transport of Intensity phase imaging by intensity spectrum fitting of exponentially spaced defocus planes". In: *Optics Express* 22.9 (2014), p. 10661. ISSN: 1094-4087. DOI: 10.1364/oe.22.010661.

[63] John Jumper et al. "Highly accurate protein structure prediction with AlphaFold". In: *Nature* 596.7873 (Aug. 2021), pp. 583–589. ISSN: 14764687. DOI: 10.1038/s41586-021-03819-2.

[64] Michael R. Kellman et al. "Physics-Based Learned Design: Optimized Coded-Illumination for Quantitative Phase Imaging". In: *IEEE Transactions on Computational Imaging* 5.3 (2019), pp. 344–353. ISSN: 2573-0436. DOI: 10.1109/tci.2019.2905434.

[65] Ryan T. Kelly. "Single-cell Proteomics: Progress and Prospects". In: *Molecular and Cellular Proteomics* 19.11 (2020), pp. 1739–1748. ISSN: 15359484. DOI: 10.1074/mcp.R120.002234. URL: http://dx.doi.org/10.1074/mcp.R120.002234.

[66] Peter V. Kharchenko. "The triumphs and limitations of computational methods for scRNA-seq". In: *Nature Methods* 18.7 (2021), pp. 723–732. ISSN: 15487105. DOI: 10.1038/s41592-021-01171-x. URL: http://dx.doi.org/10.1038/s41592-021-01171-x.

[67] Diederik P. Kingma and Jimmy Lei Ba. "Adam: A method for stochastic optimization". In: *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings* (2015), pp. 1–15.

[68] Thomas Kluyver et al. "Jupyter Notebooks—a publishing format for reproducible computational workflows". In: *Positioning and Power in Academic Publishing: Players, Agents and Agendas - Proceedings of the 20th International Conference on Electronic Publishing, ELPUB 2016* (2016), pp. 87–90. DOI: 10.3233/978-1-61499-649-1-87.

[69] Arash Komaee. "Mutual information rate between stationary Gaussian processes". In: *Results in Applied Mathematics* 7 (2020), p. 100107. ISSN: 25900374. DOI: 10.1016/j.rinam.2020.100107. URL: https://doi.org/10.1016/j.rinam.2020.100107.

[70] Marko Kreft, Matjaž Stenovec, and Robert Zorec. "Focus-drift correction in time-lapse confocal imaging". In: *Annals of the New York Academy of Sciences* 1048 (2005), pp. 321–330. ISSN: 00778923. DOI: 10.1196/annals.1342.029.

[71] Jan Kremer, Kim Steenstrup Pedersen, and Christian Igel. "Active learning with support vector machines". In: *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 4.4 (2014), pp. 313–326. ISSN: 19424795. DOI: 10.1002/widm.1132.

[72] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: *Advances In Neural Information Processing Systems* (Feb. 2011), pp. 1–9. ISSN: 10495258. DOI: http://dx.doi.org/10.1016/j.protcy.2014.09.007. URL: http://arxiv.org/abs/1102.0183.

[73] Alexander Krull, Tim Oliver Buchholz, and Florian Jug. "Noise2void-Learning denoising from single noisy images". In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 2019-June (2019), pp. 2124–2132. ISSN: 10636919. DOI: 10.1109/CVPR.2019.00223.

[74] Mark J van der Laan, Eric C Polley, and Alan E Hubbard. "Super Learner". In: *Statistical Applications in Genetics and Molecular Biology* 6.1 (2007), Article25. ISSN: 1544-6115. DOI: 10.2202/1544-6115.1309. URL: http://www.ncbi.nlm.nih.gov/pubmed/17910531%5Cnhttp://www.degruyter.com/view/j/sagmb.2007.6.issue-1/sagmb.2007.6.1.1309/sagmb.2007.6.1.1309.xml.

[75] Yann Lecun, Yoshua Bengio, and Geoffrey Hinton. *Deep learning*. 2015. DOI: 10.1038/nature14539. URL: http://www.scopus.com/inward/record.url?eid=2-s2.0-84930630277&partnerID=40&md5=befeefa64ddca265c713cf81f4e2fc54.

[76] Yann LeCun et al. "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2323. ISSN: 00189219. DOI: 10.1109/5.726791.

[77] Anat Levin et al. "Understanding blind deconvolution algorithms". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33.12 (2011), pp. 2354–2367. ISSN: 01628828. DOI: 10.1109/TPAMI.2011.148.

[78] David D. Lewis and Jason Catlett. "Heterogeneous Uncertainty Sampling for Supervised Learning". In: *Machine Learning Proceedings 1994* (1994), pp. 148–156. DOI: 10.1016/b978-1-55860-335-6.50026-x.

[79] J P Lewis. *Fast Normalized Cross-Correlation*. Tech. rep. 2010.

[80] Bin Li and Kevin W. Eliceiri. "Dual-stream Maximum Self-attention Multi-instance Learning". In: *arXiv preprint* (June 2020). URL: http://arxiv.org/abs/2006.05538.

[81] Bo Li et al. "An adaptive excitation source for high-speed multiphoton microscopy". In: *Nature Methods* 17.2 (2020), pp. 163–166. ISSN: 15487105. DOI: 10.1038/s41592-019-0663-9.

[82] J Liao et al. "Single-frame rapid autofocusing for brightfield and fluorescence whole slide imaging". In: *Biomedical Optics Express* 7.11 (2016), pp. 4763–4768. ISSN: 21567085. DOI: 10.1364/BOE.7.004763. URL: https://www.ncbi.nlm.nih.gov/pubmed/27896014.

[83] Jun Liao et al. "Rapid focus map surveying for whole slide imaging with continuous sample motion". In: *Opt. Lett.* 42.17 (2017), pp. 3379–3382. ISSN: 15394794. DOI: 10.1364/OL.42.003379. URL: http://ol.osa.org/abstract.cfm?URI=ol-42-17-3379.

[84] Jia Ren Lin, Mohammad Fallahi-Sichani, and Peter K. Sorger. "Highly multiplexed imaging of single cells using a high-throughput cyclic immunofluorescence method". In: *Nature Communications* 6 (2015), pp. 1–7. ISSN: 20411723. DOI: 10.1038/ncomms9390.

[85] Ruilong Ling et al. "High-throughput intensity diffraction tomography with a computational microscope". In: *Biomedical Optics Express* 9.5 (2018), p. 2130. ISSN: 2156-7085. DOI: 10.1364/boe.9.002130.

[86] Ziji Liu et al. "Real-time brightfield, darkfield, and phase contrast imaging in a light-emitting diode array microscope". In: *Journal of Biomedical Optics* 19.10 (2014), p. 106002. ISSN: 1083-3668. DOI: 10.1117/1.jbo.19.10.106002.

[87] David G Low. "Distinctive image features from scale-invariant keypoints". In: *International Journal of Computer Vision* (2004), pp. 91–110. URL: https://www.cs.ubc.ca/~lowe/papers/ijcv04.pdf.

[88] W. Lukosz. "Optical Systems with Resolving Powers Exceeding the Classical Limit*". In: *Journal of the Optical Society of America* 56.11 (Nov. 1966), p. 1463. ISSN: 0030-3941. DOI: 10.1364/josa.56.001463. URL: https://www.osapublishing.org/abstract.cfm?URI=josa-56-11-1463.

[89] David J C MacKay. *Information Theory, Inference, and Learning Algorithms David J.C. MacKay*. 1st. Cambridge University Press, 2003. ISBN: 9780521642989. DOI: 10.5555/971143.

[90] Amanda L Marzo et al. "Initial T cell frequency dictates memory CD8+ T cell lineage commitment." In: *Nature immunology* 6.8 (2005), pp. 793–799. ISSN: 1529-2908. DOI: 10.1038/ni1227.

[91] Naoya Matsumoto et al. "Aberration correction considering curved sample surface shape for non-contact two-photon excitation microscopy with spatial light modulator". In: *Scientific Reports* 8.1 (2018), pp. 1–13. ISSN: 20452322. DOI: 10.1038/s41598-018-27693-7.

[92] Shalin B Mehta and Colin J R Sheppard. "Quantitative phase-gradient imaging at high resolution with asymmetric illumination-based differential phase contrast". In: *Optics Letters* 34.13 (2009), p. 1924. ISSN: 0146-9592. DOI: 10.1364/ol.34.001924.

[93] Thorsten R Mempel, Sarah E Henrickson, and Ulrich H Von Andrian. "T-cell priming by dendritic cells in lymph nodes occurs in three distinct phases." In: *Nature* 427.6970 (2004), pp. 154–9. ISSN: 1476-4687. DOI: 10.1038/nature02238. URL: http://www.ncbi.nlm.nih.gov/pubmed/14712275.

[94] Alistair Miles et al. *zarr-developers/zarr-python*. 2021. DOI: 10.5281/ZENODO.3773449. URL: https://zenodo.org/record/3773449.

[95] Mustafa Mir et al. "Optical measurement of cycle-dependent cell growth". In: *Proceedings of the National Academy of Sciences of the United States of America* 108.32 (2011), pp. 13124–9. ISSN: 1091-6490. DOI: 10.1073/pnas.1100506108. URL: http://www.pnas.org/content/108/32/13124.short.

[96] Allard P. Mosk et al. "Controlling waves in space and time for imaging and focusing in complex media". In: *Nature Photonics* 6.5 (2012), pp. 283–292. ISSN: 17494885. DOI: 10.1038/nphoton.2012.88.

[97] Evgenii Narimanov. "Resolution limit of label-free far-field microscopy". In: *Advanced Photonics* 1.05 (2019), p. 1. ISSN: 2577-5421. DOI: 10.1117/1.ap.1.5.056003.

[98] Mark A. Neifeld. "Information, resolution, and space–bandwidth product". In: *Optics Letters* 23.18 (1998), p. 1477. ISSN: 0146-9592. DOI: 10.1364/ol.23.001477.

[99] Andrew Y. Ng, Michael I. Jordan, and Yair Weiss. "On Spectral Clustering: Analysis and an Algorithm". In: *Advances in Neural Information Processing Systems 14* (2002), pp. 849–856. DOI: 10.1.1.19.8100. URL: http://papers.nips.cc/paper/2092-on-spectral-clustering-analysis-and-an-algorithm.pdf.

[100] *Nikon Perfect Focus*. URL: https://www.microscopyu.com/applications/live-cell-imaging/nikon-perfect-focus-system.

[101] Tom O'Malley et al. *Keras Tuner*. \url{https://github.com/keras-team/keras-tuner}. 2019.

[102] Rudolf Oldenbourg. "Polarization microscopy with the LC-PolScope". In: *Live cell imaging: a laboratory manual* (2005), pp. 205–237.

[103] Xiaoze Ou et al. "Quantitative phase imaging via Fourier ptychographic microscopy." In: *Optics letters* 38.22 (2013), pp. 4845–8. ISSN: 1539-4794. DOI: 10.1364/OL.38.004845. URL: http://www.ncbi.nlm.nih.gov/pubmed/24322147.

[104] Wei Ouyang et al. "ImJoy: an open-source computational platform for the deep learning era". In: *Nature Methods* 16.12 (2019), pp. 1199–1200. ISSN: 15487105. DOI: 10.1038/s41592-019-0627-0. URL: http://dx.doi.org/10.1038/s41592-019-0627-0.

[105] Jongchan Park et al. "Review of bio-optical imaging systems with a high space-bandwidth product". In: *Advanced Photonics* 3.04 (June 2021), pp. 369–407. ISSN: 2577-5421. DOI: 10.1117/1.AP.3.4.044001. URL: https://royalsocietypublishing.org/doi/10.1098/rsta.1955.0001%20https://www.spiedigitallibrary.org/journals/advanced-photonics/volume-3/issue-04/044001/Review-of-bio-optical-imaging-systems-with-a-high-space/10.1117/1.AP.3.4.044001.full.

[106] George H. Patterson and David W. Piston. "Photobleaching in two-photon excitation microscopy". In: *Biophysical Journal* 78.4 (2000), pp. 2159–2162. ISSN: 00063495. DOI: 10.1016/S0006-3495(00)76762-2. URL: http://dx.doi.org/10.1016/S0006-3495(00)76762-2.

[107] Sri Rama Prasanna Pavani and Rafael Piestun. "High-efficiency rotating point spread functions". In: *Optics Express* 16.5 (Mar. 2008), p. 3484. ISSN: 1094-4087. DOI: 10.1364/OE.16.003484. URL: https://opg.optica.org/abstract.cfm?URI=oe-16-5-3484.

[108] Sri Rama Prasanna Pavani and Rafael Piestun. "Three dimensional tracking of fluorescent microparticles using a photon-limited double-helix response system". In: *Optics Express* 16.26 (2008), p. 22048. ISSN: 1094-4087. DOI: 10.1364/oe.16.022048.

[109] Tingying Peng et al. "A BaSiC tool for background and shading correction of optical microscopy images". In: *Nature Communications* 8 (2017), pp. 1–7. ISSN: 20411723. DOI: 10.1038/ncomms14836. URL: http://dx.doi.org/10.1038/ncomms14836.

[110] Saul Perlmutter. "Supernovae, Dark Energy, and the Accelerating Universe". In: *Physics Today* 56.4 (Apr. 2003), pp. 53–60. ISSN: 0031-9228. DOI: 10.1063/1.1580050. URL: http://supernova.lbl.gov/PhysicsTodayArticle.pdf%20http://physicstoday.scitation.org/doi/10.1063/1.1580050.

[111] Zachary F. Phillips, Regina Eckert, and Laura Waller. "Quasi-Dome: A Self-Calibrated High-NA LED Illuminator for Fourier Ptychography". In: *Imaging and Applied Optics 2017 (3D, AIO, COSI, IS, MATH, pcAOP)*. Vol. IW4E.5. 2017. DOI: 10.1364/isa.2017.iw4e.5.

[112] Zachary F. Phillips et al. "Multi-Contrast Imaging and Digital Refocusing on a Mobile Microscope with a Domed LED Array". In: *Plos One* 10.5 (2015), e0124938. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0124938. URL: http://dx.plos.org/10.1371/journal.pone.0124938.

[113] Henry Pinkard. *Code for reproducing analysis of data and figures in "Learned adaptive multiphoton illumination microscopy for large-scale immune response imaging"*. 2020. DOI: 10.5281/ZENODO.4315851. URL: https://zenodo.org/record/4315851.

[114] Henry Pinkard. *Data for "Learned adaptive multiphoton illumination microscopy"*. 2020. DOI: 10.6084/M9.FIGSHARE.12841781.V1. URL: https://figshare.com/articles/dataset/Data_for_Learned_adaptive_multiphoton_illumination_microscopy_/12841781/1.

[115] Henry Pinkard. *Learned Adaptive Multiphoton illumination Microscopy tutorial Jupyter notebook*. 2020. DOI: 10.5281/ZENODO.4314107. URL: https://zenodo.org/record/4314107.

[116] Henry Pinkard, Kaitlin Corbin, and Matthew F. Krummel. "Spatiotemporal Rank Filtering Improves Image Quality Compared to Frame Averaging in 2-Photon Laser Scanning Microscopy". In: *PLOS ONE* 11.3 (Mar. 2016). Ed. by Jonathan A Coles, e0150430. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0150430. URL: http://dx.plos.org/10.1371/journal.pone.0150430.

[117] Henry Pinkard et al. "Deep learning for single-shot autofocus microscopy". In: *Optica* 6.6 (2019), p. 794. ISSN: 2334-2536. DOI: 10.1364/optica.6.000794.

[118] Henry Pinkard et al. "Learned adaptive multiphoton illumination microscopy". In: *bioRxiv* (2020), p. 2020.08.14.251314. URL: https://doi.org/10.1101/2020.08.14.251314.

[119] Henry Pinkard et al. "Micro-Magellan: open-source, sample-adaptive, acquisition software for optical microscopy". In: *Nature Methods* 13.10 (Sept. 2016), pp. 807–809. ISSN: 1548-7091. DOI: 10.1038/nmeth.3991. URL: http://www.nature.com/doifinder/10.1038/nmeth.3991.

[120] Henry Pinkard et al. "Pycro-Manager: open-source software for customized and reproducible microscope control". In: *Nature Methods* 18.3 (2021), pp. 226–228. ISSN: 15487105. DOI: 10.1038/s41592-021-01087-6. URL: http://dx.doi.org/10.1038/s41592-021-01087-6.

[121] Russell A. Poldrack, Krzysztof J. Gorgolewski, and Gaël Varoquaux. "Computational and Informatic Advances for Reproducible Data Analysis in Neuroimaging". In: *Annual Review of Biomedical Data Science* 2.1 (July 2019), pp. 119–138. ISSN: 2574-3414. DOI: 10.1146/annurev-biodatasci-072018-021237. URL: https://www.annualreviews.org/doi/10.1146/annurev-biodatasci-072018-021237.

[122] Yury Polyanskiy, H. Vincent Poor, and Sergio Verdú. "Channel coding rate in the finite blocklength regime". In: *IEEE Transactions on Information Theory* 56.5 (2010), pp. 2307–2359. ISSN: 00189448. DOI: 10.1109/TIT.2010.2043769.

[123] Ben Poolel et al. "On variational bounds of mutual information". In: *36th International Conference on Machine Learning, ICML 2019*. Vol. 2019-June. 2019, pp. 9036–9049. ISBN: 9781510886988.

[124] Gabriel Popescu and YongKeun Park. "Special Section Guest Editorial:Quantitative Phase Imaging in Biomedicine". In: *Journal of Biomedical Optics* 20.11 (2015), p. 111201. ISSN: 1083-3668. DOI: 10.1117/1.jbo.20.11.111201. URL: http://www.ncbi.nlm.nih.gov/pubmed/26648557.

[125] Bali Pulendran and Rafi Ahmed. "Translating innate immunity into immunological memory: implications for vaccine development." In: *Cell* 124.4 (Feb. 2006), pp. 849–63. ISSN: 0092-8674. DOI: 10.1016/j.cell.2006.02.019. URL: http://www.ncbi.nlm.nih.gov/pubmed/16497593.

[126] Sean Quirin, Sri Rama Prasanna Pavani, and Rafael Piestun. "Optimal 3D single-molecule localization for superresolution microscopy with aberrations and engineered point spread functions". In: *Proceedings of the National Academy of Sciences* 109.3 (Jan. 2012), pp. 675–679. ISSN: 0027-8424. DOI: 10.1073/pnas.1109011108. URL: https://pnas.org/doi/full/10.1073/pnas.1109011108.

[127] Zhenbo Ren, Zhimin Xu, and Edmund Y. Lam. "Learning-based nonparametric autofocusing for digital holography". In: *Optica* 5.4 (2018), p. 337. DOI: 10.1364/optica.5.000337.

[128] Vasco Ronchi. "Resolving Power of Calculated and Detected Images". In: *Journal of the Optical Society of America* 51.4 (Apr. 1961), pp. 458–460. ISSN: 0030-3941. DOI: 10.1364/JOSA.51.0458{_}1. URL: https://www.osapublishing.org/abstract.cfm?URI=josa-51-4-458_1%20https://opg.optica.org/abstract.cfm?URI=josa-51-4-458_1.

[129] F. Rosenblatt. "The perceptron: A probabilistic model for information storage and organization in the brain." In: *Psychological Review* 65.6 (1958), pp. 386–408. ISSN: 1939-1471. DOI: 10.1037/h0042519. URL: http://doi.apa.org/getdoi.cfm?doi=10.1037/h0042519.

[130] Brian C. Ross. "Mutual information between discrete and continuous data sets". In: *PLoS ONE* 9.2 (2014). ISSN: 19326203. DOI: 10.1371/journal.pone.0087357.

[131] Donald B. Rubin. "Bayesianly Justifiable and Relevant Frequency Calculations for the Applied Statistician". In: *The Annals of Statistics* 12.4 (Dec. 1984), pp. 1151–1172. ISSN: 0090-5364. DOI: 10.1214/aos/1176346785. URL: https://projecteuclid.org/journals/annals-of-statistics/volume-12/issue-4/Bayesianly-Justifiable-and-Relevant-Frequency-Calculations-for-the-Applied-Statistician/10.1214/aos/1176346785.full.

[132] Marc Rullan et al. "An Optogenetic Platform for Real-Time, Single-Cell Interrogation of Stochastic Transcriptional Regulation". In: *Molecular Cell* 70.4 (2018), pp. 745–756. ISSN: 10974164. DOI: 10.1016/j.molcel.2018.04.012. URL: https://doi.org/10.1016/j.molcel.2018.04.012.

[133] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. "Learning representations by back-propagating errors". In: *Nature* 323.6088 (Oct. 1986), pp. 533–536. ISSN: 0028-0836. DOI: 10.1038/323533a0. URL: http://www.nature.com/articles/323533a0.

[134] Duncan P. Ryan et al. "Automatic and adaptive heterogeneous refractive index compensation for light-sheet microscopy". In: *Nature Communications* 8.1 (2017), pp. 1–9. ISSN: 20411723. DOI: 10.1038/s41467-017-00514-7. URL: http://dx.doi.org/10.1038/s41467-017-00514-7.

[135] Bahaa Saleh. "A priori information and the degrees of freedom of noisy images". In: *Journal of the Optical Society of America* 67.1 (1977), p. 71. ISSN: 0030-3941. DOI: 10.1364/josa.67.000071.

[136] Etai Sapoznik et al. "A Versatile Oblique Plane Microscope for Large-Scale and High-Resolution Imaging of Subcellular Dynamics." In: *bioRxiv* (2020). DOI: 10.1101/2020.04.07.030569.

[137] Nico Scherf and Jan Huisken. "The smart and gentle microscope". In: *Nature Biotechnology* 33.8 (Aug. 2015), pp. 815–818. ISSN: 1087-0156. DOI: 10.1038/nbt.3310. URL: http://www.nature.com/doifinder/10.1038/nbt.3310.

[138] Johannes Schindelin et al. "Fiji: an open-source platform for biological-image analysis". In: *Nature Methods* 9.7 (June 2012), pp. 676–682. ISSN: 1548-7091. DOI: 10.1038/nmeth.2019. URL: http://www.nature.com/doifinder/10.1038/nmeth.2019.

[139] Caroline A. Schneider, Wayne S. Rasband, and Kevin W. Eliceiri. "NIH Image to ImageJ: 25 years of image analysis". In: *Nature Methods* 9.7 (2012), pp. 671–675. ISSN: 15487091. DOI: 10.1038/nmeth.2089.

[140] Burr Settles. *Active Learning Literature Survey*. Tech. rep. Mar. 2010. DOI: 10.1016/j.matlet.2010.11.072. URL: https://linkinghub.elsevier.com/retrieve/pii/S0167577X10010438.

[141] Claude E Shannon. "A mathematical theory of communication". In: *The Bell System Technical Journal* 5.1 (Jan. 1948), p. 3. ISSN: 15591662. DOI: 10.1145/584091.584093. URL: http://cm.bell-labs.com/cm/ms/what/shannonday/shannon1948.pdf%20http://portal.acm.org/citation.cfm?doid=584091.584093.

[142] Yoav Shechtman et al. "Optimal point spread function design for 3D imaging". In: *Physical Review Letters* 113.3 (Sept. 2014). ISSN: 10797114. DOI: 10.1103/PhysRevLett.113.133902.

[143] Feimo Shen, Louis Hodgson, and Klaus Hahn. "Digital Autofocus Methods for Automated Microscopy". In: *Methods in Enzymology*. Vol. 414. Academic Press, 2006, pp. 620–632. ISBN: 0121828190. DOI: 10.1016/S0076-6879(06)14032-X.

[144] J Shi and J Malik. "Normalized Cuts and Image Segmentation". In: *Ieee Transactions on Pattern Analysis and Machine Intelligence* 22.8 (2000), pp. 888–905. ISSN: 0162-8828. DOI: 10.1109/34.868688. URL: http://www.computer.org/portal/web/csdl/doi?doc=abs/proceedings/cvpr/1997/7822/00/78220731abs.htm%5Cnpapers3://publication/uuid/268FC197-AF47-4C7C-887F-BEDB94A81320.

[145] David Silver et al. "Mastering the game of Go with deep neural networks and tree search". In: *Nature* 529.7587 (Jan. 2016), pp. 484–489. ISSN: 14764687. DOI: 10.1038/nature16961.

[146] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. "Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps". In: *arXiv preprint* 1312.6034 (Dec. 2013). ISSN: 17088240. DOI: 10.1111/jerd.12389. URL: http://arxiv.org/abs/1312.6034.

[147] Kevin Smith et al. "CIDRE: An illumination-correction method for optical microscopy". In: *Nature Methods* 12.5 (2015), pp. 404–406. ISSN: 15487105. DOI: 10.1038/nmeth.3323.

[148] Nitish Srivastava et al. "Dropout: A Simple Way to Prevent Neural Networks from Overfitting". In: *Journal of Machine Learning Research* 15 (2014), pp. 1929–1958. ISSN: 15337928. DOI: 10.1214/12-AOS1000.

[149] *Statistical optics*. ISBN: 9781119009450.

[150] James V Stone. "Information Theory : A Tutorial Introduction". PhD thesis. 2019.

[151] Christian Szegedy et al. "Intriguing properties of neural networks". In: *arXiv preprint arXiv: ...* (Dec. 2013), pp. 1–10. ISSN: 15499618. DOI: 10.1021/ct2009208. URL: http://arxiv.org/abs/1312.6199.

[152] Jianyong Tang, Ronald N. Germain, and Meng Cui. "Superpenetration optical microscopy by iterative multiphoton adaptive compensation technique". In: *Proceedings of the National Academy of Sciences of the United States of America* 109.22 (2012), pp. 8434–8439. ISSN: 00278424. DOI: 10.1073/pnas.1119590109.

[153] Lucas Theis and Eirikur Agustsson. "On the advantages of stochastic encoders". In: *arXiv* (Feb. 2021), pp. 1–8. DOI: 10.48550/arXiv.2102.09270. URL: http://arxiv.org/abs/2102.09270.

[154] Lei Tian and Laura Waller. "3D intensity and phase imaging from light field measurements in an LED array microscope". In: *Optica* 2.2 (2015), pp. 104–111. ISSN: 2334-2536. DOI: 10.1364/OPTICA.2.000104.

[155] Lei Tian and Laura Waller. "Quantitative differential phase contrast imaging in an LED array microscope". In: *Optics Express* 23.9 (2015), p. 11394. ISSN: 1094-4087. DOI: 10.1364/OE.23.011394. URL: https://www.osapublishing.org/oe/abstract.cfm?uri=oe-23-9-11394.

[156] Lei Tian, Jingyan Wang, and Laura Waller. "3D differential phase-contrast microscopy with computational illumination using an LED array". In: *Optics Letters* 39.5 (2014), p. 1326. ISSN: 0146-9592. DOI: 10.1364/ol.39.001326.

[157] Lei Tian et al. "Computational illumination for high-speed in vitro Fourier ptychographic microscopy". In: *Optica* 2.10 (2015), p. 904. ISSN: 2334-2536. DOI: 10.1364/optica.2.000904.

[158] Naftali Tishby et al. "The information bottleneck method". In: (1999), pp. 1–16. arXiv: 0004057 [physics].

[159] Jared E. Toettcher et al. "Light-based feedback for controlling intracellular signaling dynamics". In: *Nature Methods* 8.10 (2011), pp. 837–839. ISSN: 15487091. DOI: 10.1038/nmeth.1700.

[160] Stéfan Van Der Walt, S. Chris Colbert, and Gaël Varoquaux. "The NumPy array: A structure for efficient numerical computation". In: *Computing in Science and Engineering* 13.2 (2011), pp. 22–30. ISSN: 15219615. DOI: 10.1109/MCSE.2011.37.

[161] Stéfan Van Der Walt et al. "Scikit-image: Image processing in python". In: *PeerJ* 2014.1 (2014), pp. 1–18. ISSN: 21678359. DOI: 10.7717/peerj.453.

[162] Pascal Vincent and Hugo Larochelle. "Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion Pierre-Antoine Manzagol". In: *Journal of Machine Learning Research* 11 (2010), pp. 3371–3408. ISSN: 15324435. DOI: 10.1111/1467-8535.00290.

[163] Pauli Virtanen et al. "SciPy 1.0: fundamental algorithms for scientific computing in Python". In: *Nature Methods* 17.3 (2020), pp. 261–272. ISSN: 15487105. DOI: 10.1038/s41592-019-0686-2.

[164] Dushan N. Wadduwage et al. "De-scattering with Excitation Patterning (DEEP) Enables Rapid Wide-field Imaging Through Scattering Media". In: *ArXiv* (Feb. 2019). URL: http://arxiv.org/abs/1902.10737.

[165] Aaron B. Wagner and Johannes Balle. "Neural Networks Optimally Compress the Sawbridge". In: *Data Compression Conference Proceedings* 2021-March (2021), pp. 143–152. ISSN: 10680314. DOI: 10.1109/DCC50243.2021.00022.

[166] Ru Wang et al. "Dispersion-relation phase spectroscopy of intracellular transport". In: *Optics Express* 19.21 (2011), p. 20571. ISSN: 1094-4087. DOI: 10.1364/OE.19.020571.

[167] Weiran Wang and Miguel Á. Carreira-Perpiñán. "Projection onto the probability simplex: An efficient algorithm with a simple proof, and an application". In: *arXiv* (2013). URL: http://arxiv.org/abs/1309.1541.

[168] Martin Weigert et al. "Content-aware image restoration: pushing the limits of fluorescence microscopy". In: *Nature Methods* 15.12 (2018), pp. 1090–1097. ISSN: 15487105. DOI: 10.1038/s41592-018-0216-7. URL: http://dx.doi.org/10.1038/s41592-018-0216-7.

[169] Gordon Wetzstein et al. "Inference in artificial intelligence with deep optics and photonics". In: *Nature* 588.7836 (2020), pp. 39–47. ISSN: 14764687. DOI: 10.1038/s41586-020-2973-6.

[170] Benjamin K Wilson and Genevieve D Vigil. "Automated bacterial identification by angle resolved dark-field imaging." In: *Biomedical optics express* 4.9 (2013), pp. 1692–701. ISSN: 2156-7085. DOI: 10.1364/BOE.4.001692. URL: http://www.opticsinfobase.org/boe/abstract.cfm?URI=boe-4-9-1692.

[171] Yichen Wu et al. "Extended depth-of-field in holographic imaging using deep-learning-based autofocusing and phase recovery". In: *Optica* 5.6 (June 2018), p. 704. ISSN: 2334-2536. DOI: 10.1364/OPTICA.5.000704. URL: http://arxiv.org/abs/1803.08138%0Ahttp://dx.doi.org/10.1364/OPTICA.5.000704%20https://www.osapublishing.org/abstract.cfm?URI=optica-5-6-704.

[172] Chihiro Yamazaki et al. "Critical roles of a dendritic cell subset expressing a chemokine receptor, XCR1." In: *The Journal of Immunology* 190.12 (2013), pp. 6071–6082. ISSN: 1550-6606. DOI: 10.4049/jimmunol.1202798. URL: http://www.jimmunol.org/cgi/doi/10.4049/jimmunol.1202798%5Cnpapers3://publication/doi/10.4049/jimmunol.1202798.

[173] Samuel J. Yang et al. "Assessing microscope image focus quality with deep learning". In: *BMC Bioinformatics* 19.1 (2018), p. 28. ISSN: 14712105. DOI: 10.1186/s12859-018-2087-4.

[174] Siavash Yazdanfar et al. "Simple and robust image-based autofocusing for digital microscopy." In: *Optics express* 16.12 (2008), pp. 8670–8677. ISSN: 1094-4087. DOI: 10.1364/OE.16.008670.

[175] Li-Hao Yeh et al. "uPTI: uniaxial permittivity tensor imaging of intrinsic density and anisotropy". In: *Biophotonics Congress 2021*. Washington, D.C.: Optica Publishing Group, 2021, NM3C.4. ISBN: 978-1-943580-85-9. DOI: 10.1364/NTM.2021.NM3C.4. URL: https://opg.optica.org/abstract.cfm?URI=NTM-2021-NM3C.4.

[176] Jonghee Yoon et al. "Identification of non-activated lymphocytes using three-dimensional refractive index tomography and machine learning". In: *Scientific Reports* 7.1 (2017). ISSN: 20452322. DOI: 10.1038/s41598-017-06311-y.

[177] Mark D. Zarella et al. "A practical guide to whole slide imaging a white paper from the digital pathology association". In: *Archives of Pathology and Laboratory Medicine* 143.2 (2019), pp. 222–234. ISSN: 15432165. DOI: 10.5858/arpa.2018-0343-RA.

[178] *Zeiss Definite Focus*. URL: https://www.zeiss.com/microscopy/us/products/light-microscopes/axio-observer-for-biology/definite-focus.html.

[179] F Zernike. "How I discovered phase contrast". In: *Science* 121.3141 (1955), pp. 345–349. ISSN: 00368075. DOI: 10.1126/science.121.3141.345.

[180] Xin Zhang et al. "Improvement in focusing accuracy of DNA sequencing microscope with multi-position laser differential confocal autofocus method". In: *Optics Express* 26.2 (2018), p. 887. ISSN: 1094-4087. DOI: 10.1364/OE.26.000887. URL: https://www.osapublishing.org/abstract.cfm?URI=oe-26-2-887.

[181] Guoan Zheng, Christopher Kolner, and Changhuei Yang. "Microscopy refocusing and dark-field imaging by using a simple LED array". In: *Optics Letters* 36.20 (2011), p. 3987. ISSN: 0146-9592. DOI: 10.1364/OL.36.003987. URL: https://www.osapublishing.org/abstract.cfm?URI=ol-36-20-3987.

[182] Guoan Zheng et al. "Wide-field, high-resolution Fourier ptychographic microscopy." In: *Nature photonics* 7.9 (2013), pp. 739–745. ISSN: 1749-4885. DOI: 10.1038/nphoton.2013.187. URL: http://eutils.ncbi.nlm.nih.gov/entrez/eutils/elink.fcgi?dbfrom=pubmed&id=25243016&retmode=ref&cmd=prlinks%5Cnpapers3://publication/doi/10.1038/nphoton.2013.187%5Cnhttp://dx.doi.org/10.1038/nphoton.2013.187.

[183] Michał Ziemczonok et al. "3D-printed biological cell phantom for testing 3D quantitative phase imaging systems". In: *Scientific Reports* 9.1 (2019), pp. 1–9. ISSN: 20452322. DOI: 10.1038/s41598-019-55330-4.

[184] Barret Zoph and Quoc V. Le. "Neural Architecture Search with Reinforcement Learning". In: (2016), pp. 1–16. ISSN: 1938-7228. DOI: 10.1016/j.knosys.2015.01.010. URL: http://arxiv.org/abs/1611.01578.

Appendix A

Visual information theory

Though originally developed for communications engineering, information theory [141] contains mathematical tools with myriad potential applications in microscopy. In this chapter its key ideas are introduced, focusing on intuitions and providing visual explanations wherever possible.

A.1 Introduction

In the 1940s, Claude Shannon [141] showed that a precise mathematical definition of information arises directly from the axioms of probability. Since then, information theory has served as a foundation for the digital world we live in today by defining the limits of data compression and the reliable transmission of data across noisy networks. In addition, the mathematical tools it provides have found numerous applications in diverse areas such as statistics, machine learning, cryptography, quantum computing, biology, and many others.

Here we introduce the foundations of information theory, with an emphasis on intuition. This is meant to serve as a minimal introduction to key ideas. A more comprehensive treatment can be found in the excellent textbooks [89, 23] as well as Shannon's original formulation [141].

Main ideas Outside of the context of information theory, most of us already have a vague intuitive notion of what "information" is, which is usually something close to "knowledge obtained about something unknown." One of the main advances of information theory was to formalize this intuition into something mathematically precise. Shannon himself once said that "information can be treated very much like a physical quantity such as mass or energy."

Shannon was able to make the idea of information mathematically concrete by defining it in terms of probability. If a sender is transmitting information about something to a receiver, that information is describing something that would otherwise be unpredictable. In other words, from the receiver's perspective, it is random. Information theory shows that the randomness of the message, as characterized by the probability distribution over potential

messages, is in fact the *only* thing that matters for figuring out how to transmit information. The actual content of the messages is irrelevant.

This fact enables us to consider the fundamental concepts in information theory using a generic source of random events: drawing colored marbles from an urn, with each marble representing a possible message from the set of all messages. The mathematics developed on this example can be directly applied to other application areas of information theory. For example, the way we describe the randomness of a sequence of colored marbles is the same as how we would describe randomness of a string of letters in English text, or a sequence of pixels in images taken on the Hubble telescope.

The fundamental unit of information content is the **bit**. It is worth noting that the word "bit" also describes a container that can hold information: a digit that can be either `0` or `1`. The actual information within the container must be defined with respect to some probability distribution, and it could hold less than 1 bit of information in the same way as a 1 liter bottle can be filled with only $\frac{1}{2}$ liter of water. Throughout this paper, we try to avoid this ambiguity by choosing examples where where each bit container is "full" with one bit of information. So it is safe to assume that "bit" here refers to the units of information, unless explicitly stated otherwise.

There are two key problems in information theory that will come up in the sections below. The first is **source coding**, which concerns data compression: Given a source of random events, what are the limits on how succinctly we can record the outcome of a random sequence, on average? This can be further subdivided into the cases where the original sequence can be perfectly reconstructed upon decompression (**lossless compression**) or the case where some distortion from the original sequence can be tolerated to achieve even smaller data size (**lossy compression**).

The second problem is **channel coding**, which concerns data transmission: Given a source of random events, and an imperfect system for transmitting information (a **noisy channel**), how can we encode a sequence of such events such that they will be robust to errors introduced by the channel? Again, we can subdivide into perfect data transmission, and data transmission with errors.

Notation

X	a random variable, which will also be called a "random event"
x	a particular outcome of X
$\mathcal{X}$	the probability space/set of possible outcomes for X. $x \in \mathcal{X}$
$\lvert\mathcal{X}\rvert$	the number of possible outcomes in the probability space
$p_{\mathrm{X}}(x)$, $p(\mathrm{X} = x)$ or $p(x)$	the probability that the random event X has outcome x
$H(\mathrm{X})$	the entropy of X
$H_{\max}(\mathcal{X})$	the maximum entropy for the state space $\mathcal{X}$
$W(\mathrm{X})$	the redundancy of X
$\mathbf{X} = \mathrm{X}_1, \mathrm{X}_2, ...$	a stochastic process: an ordered sequence of random variables
$\mathcal{X}^{\mathbf{2}} = \mathcal{X} \times \mathcal{X}$	the state space for (x_1, x_2) tuples where $x_1 \in \mathcal{X}, x_2 \in \mathcal{X}$
$\mathcal{P}_{\mathcal{X}}$	The set of probability distributions on the space $\mathcal{X}$
$\mathbf{p}_{\mathrm{X}}$	A vector representation of the probabilities of the distribution of X
$\mathbf{P}_{\mathrm{X,Y}}$	A matrix representation of the joint distribution of X and Y
$\mathbf{P}_{\mathrm{Y\lvert X}}$	A matrix representation of conditional distribution Y given X
C	The channel capacity: the maximum information that can be transmitted by a channel

A.2 Information, uncertainty, entropy, and mutual information

What is information?

Uncertainty, randomness, and a lack of information are different ways of viewing the same underlying idea. If we cannot predict the outcome of an event, we are uncertain, and it is, from our perspective, random. However, we may be able to acquire information about the event that reduces its apparent randomness, making us more certain about it.

What will the weather be like in 3 hours? It's uncertain. But from now until then, we can continuously acquire information by observing the current weather, and 3 hours from now our uncertainty will have collapsed to zero. It no longer appears random to us.

Weather forecasting is not the only area that fits this paradigm. In fact, information theory shows that the semantic context of the random event (i.e. *what* is happening) is unrelated to its information content, which derives solely from the probability distribution of possible outcomes (i.e. *how often* it happens). Thus, we can consider information theory in the context of a generic random event: repeatedly drawing a marble at random from an urn containing `blue`, `green`, `yellow`, and `gray` marbles. (**Fig. A.1a**).

Formally, we have a finite, discrete set of independent and identically distributed (IID) events and a random variable that assigns a probability to each element of the set. The

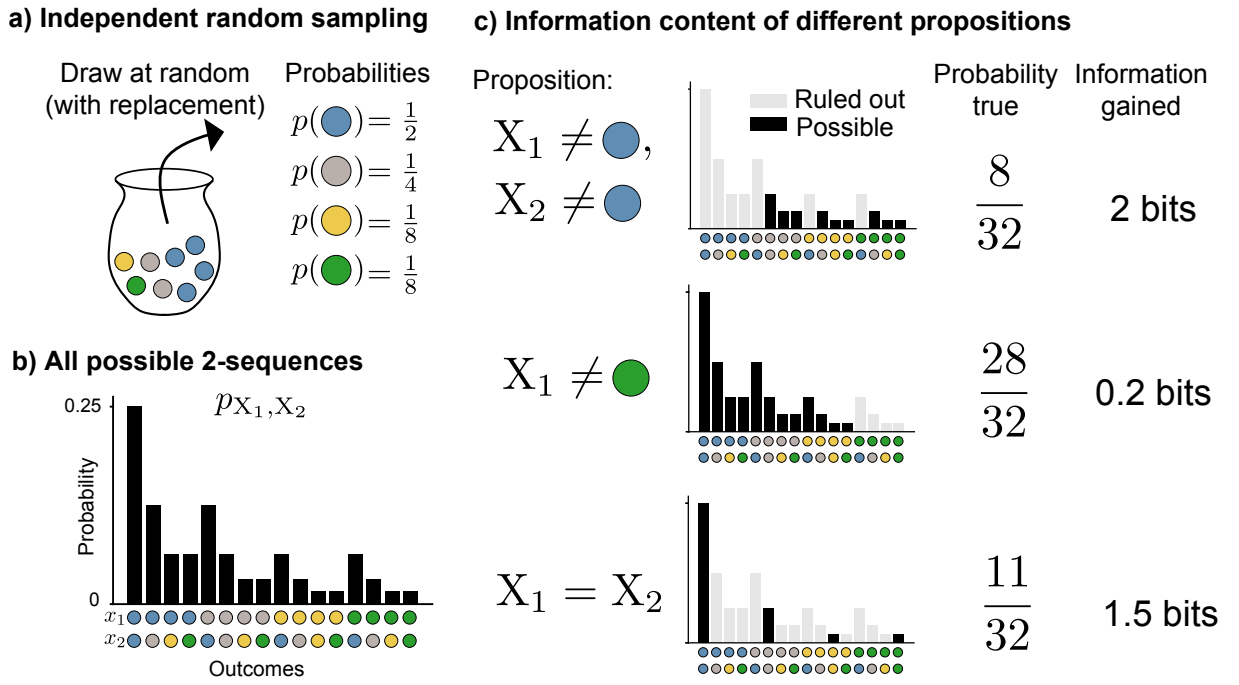

Figure A.1: **Equivalence of probability and information a)** A sequence of two marbles is drawn at random (with replacement) from an urn, giving rise to **b)** a probability distribution over the 16 possible two-color sequences. **c)** Learning that a proposition about the two colors drawn is true enables the elimination of certain outcomes. For example, learning neither marble is blue eliminates $\frac{7}{16}$ possibilities containing $\frac{3}{4}$ of the the probability mass. Eliminating probability mass, reducing uncertainty about the outcome, and gaining information are all mathematically equivalent. Reduction of 50% of the probability mass corresponds to 1 bit of information.

intuitions developed on this simplified scenario will later be generalized to the non IID case (Sec. A.2) and continuous variables and probability density functions (Sec. A.2).

More common outcomes provide us with less information, and rarer outcomes provide us with more. To understand why, suppose someone has drawn two colored marbles from an urn, but we don't know what they've draw. We'll denote these two random events with X_1 and X_2. X_1 and X_2 have a joint probability distribution p_{X_1,X_2}, which tells us the probability of each of the 16 possible two-color sequences (**Fig. A.1b**). Suppose we then learn some facts about what was drawn (but not the exact outcome). For example, we might learn that neither marble is `blue`, or the first marble isn't `green`, or the two marbles are different colors (**Fig. A.1c**).

Learning each fact allows us to rule out possibilities for the outcome of X_1, X_2. The less

often the fact is true, the more possible outcomes we can rule out. For example, knowing the two marbles are the same color rules out 12 of the 16 possible outcomes that correspond to $\frac{21}{32}$ of probability mass. The more possibilities we can rule out, the more information we've gained. Thus, rarer outcomes contain more information. The more surprised we are by the outcome, the more information it has provided us.

Furthermore, we can calculate exactly how much information that outcome has provided based on its probability. Each time we can rule out half of the probability mass, we have gained exactly 1 bit of information. In the case where we learn that neither marble is `blue`, we're left with only $\frac{1}{4}$ of the probability mass, and we have gained 2 bits of information, since we have halved the probability mass twice. The amount of information can be calculated from the probability of the outcome by:

$$\log_2 \frac{1}{p(x)}$$

The conventional choice of a base-2 logarithm means that the units will be **bits** (The 2 will often be omitted in subsequent sections).

Since we can calculate the information provided by each outcome, and we know each outcome's probability, we can compute the probability-weighted average amount of information provided by a random event, otherwise known as the **entropy**. Entropy is denoted $H(\mathrm{X})$ and is defined mathematically as:

$$H(\mathrm{X}) = \sum_{x \in \mathcal{X}} p(x) \log \frac{1}{p(x)}$$

.

Entropy can also be interpreted as how surprising the random event is on average. Events that are more random will tend to be more surprising on average, and observing their outcome will yield more information, and reduce our uncertainty more. On the flip side, higher entropy also means that there is more uncertainty in the random event to begin with, so we must acquire more information to be certain about its outcome compared to an event with lower entropy.

It might seem that, since lower probability outcomes contain more information, that random events containing many extremely rare outcomes will have the highest entropy. In fact, the opposite is true: random events with probability spread as evenly as possible over their possible outcomes will have the highest entropy. Mathematically, this happens because the gains in information that come from an event becoming increasingly rare are outweighed by the decrease in the probability that the highly informative outcome will occur, due to the logarithm in the formula for information. Section A.2 will discuss maximum entropy distributions in more detail.

Entropy and data compression

A random variable's entropy is connected to many other interesting properties. For example, it quantifies the limit of how much (lossless) compression of data from a given source can be achieved, a problem in information theory known as **source coding**.

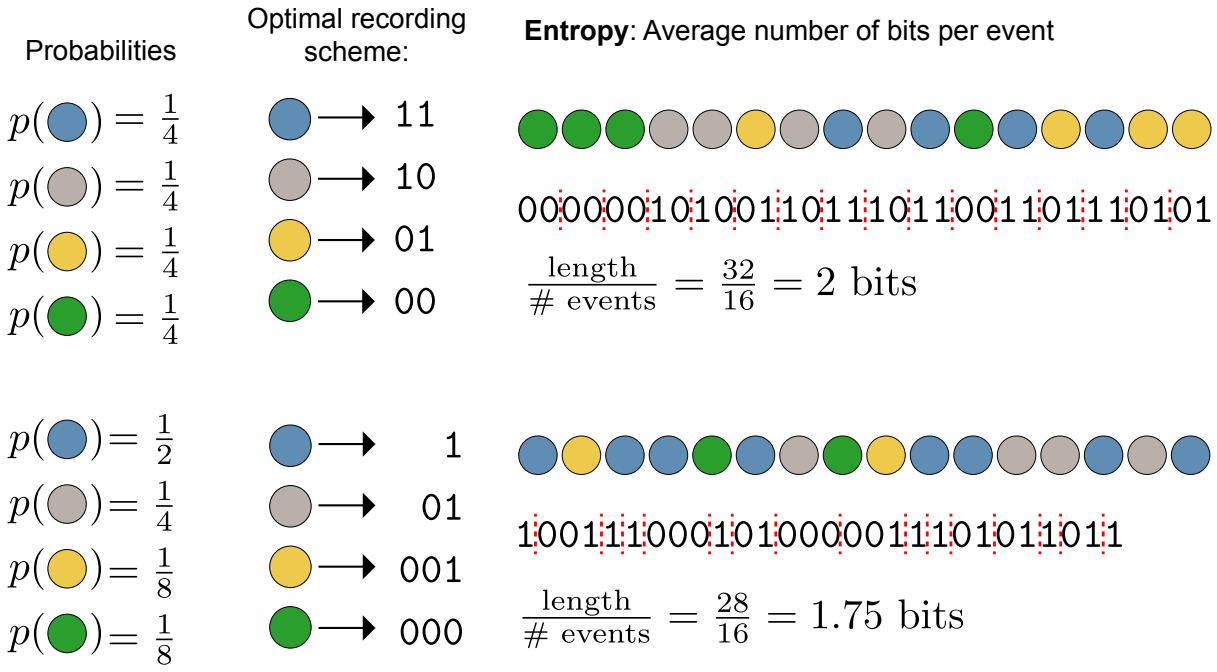

Figure A.2: **Entropy** can be interpreted as the average length of the shortest encoding of a sequence of random events, which here are repeated draws (with replacement) of colored marbles from an urn. **(Top)** With equal probability of each color, the shortest binary recording assigns a two-digit binary string code to each event. The entropy is the average number of bits per event of a typical sequence: 2 bits. **(Bottom)** When some colors are more likely than others, the more probable ones can be recorded as shorter binary strings to save space. This gives a shorter entropy: 1.75 bits.

To demonstrate this, we return to the problem of drawing colored marbles from an urn. Now we're going to draw a sequence of marbles, and record the outcome of each draw using binary **codewords**, such as `1`, `10`, etc. Our goal is to choose an encoding scheme that will (on average) yield the shortest possible binary description of our sequence without introducing any ambiguity as to what the outcomes of the draws were. This is a **lossless compression** problem, and the shortest possible length is determined by the entropy the random draw.

We want to pick an encoding scheme that matches outcomes (e.g. a `green` marble drawn) to bit strings (e.g. `01`), such that the length of the string for a series of random draws is as short as possible. How short we can make this encoding, while still being able

to correctly decode our binary encoding into the original sequence, is closely related to the fundamental **information** content of the sequence: sequences that require longer encodings (on average) contain more information. Thus, the length of the optimal binary recording scheme is determined by the random variable's **entropy**–the average information present in a single outcome.

The amount of information and the optimal recording scheme both depend on the probabilities of each outcome. For example, consider a random variable X on the **outcome space** $\mathcal{X}$ with associated probabilities p_{X} (**Fig. A.2, top**):

$$\mathcal{X} = \{\texttt{blue}, \texttt{gray}, \texttt{yellow}, \texttt{green}\}$$
$$p_{\mathrm{X}} = \{\tfrac{1}{4}, \tfrac{1}{4}, \tfrac{1}{4}, \tfrac{1}{4}\}.n$$

The shortest lossless encoding scheme given these probabilities is a unique 2 digit binary codeword for each outcome: `00`, `01`, `10`, `11`, because the average information in each random event (its **entropy**) is 2 bits.

Alternatively, consider the case where the outcomes are not equally probable (**Fig. A.2, bottom**), but instead:

$$\mathcal{X} = \{\texttt{blue}, \texttt{gray}, \texttt{yellow}, \texttt{green}\}$$
$$p_{\mathrm{X}} = \{\tfrac{1}{2}, \tfrac{1}{4}, \tfrac{1}{8}, \tfrac{1}{8}\}.$$

In this case, even though the outcome still has some randomness, we can more reliably predict the next element of the sequence compared to the case where all probabilities are equal: it is more likely to be `blue` and less likely to be `yellow` or `green`. Thus, we are more certain and and shouldn't need as much information (on average) to describe the outcomes compared to the previous case.

This is reflected in the fact that we can achieve a shorter average binary string length by encoding more probable outcomes with shorter strings and less probable outcomes with longer strings. The shortest, lossless encoding scheme will be to use the codewords `1`, `01`, `001`, `000`. (This encoding is not unique–we could always swap `0`s and `1`s). This is an example of a **prefix code**, which can be uniquely decoded even if the the codewords are concatenated one after another, since no codeword is a prefix of another. Dividing the length of the encoding of a series of events by the number of events, we compute that the average information (entropy) in each random event is 1.75 bits.

If we instead used the 2 digit binary code, the amount of information would be unchanged, but average encoding would be longer than necessary. In other words, this code would introduce **redundancy**.

In addition to defining the limits of data compression about a random variable, entropy describes the uncertainty about what the outcome of a random variable will be. If $p(\texttt{blue}) = 1$ and all other probabilities equal 0, the entropy is 0. In this case there is no uncertainty because we are sure of the outcome even before seeing it.

Redundancy

The **redundancy** of a random event depends on how uncertain its outcome is, compared to how uncertain a variable with the same set of possible outcomes *could* be. The latter is defined by the **maximum entropy** distribution $H_{\max}(\mathcal{X})$ on the set of possible outcomes $\mathcal{X}$. The maximum entropy distribution is always the one in which the probability of all events is equal (**Fig. A.2, top**). This makes sense because the outcome is the most uncertain, or equivalently the optimal binary recording of the events has the longest possible length.

By setting the probability of all events equal in a particular probability space, we find that maximum entropy is equal to the log number of possible states:

$$H_{\max} = \log_2 |\mathcal{X}|$$

,
where $|\mathcal{X}|$ is the number of elements in the set $\mathcal{X}$. The larger the number of elements in $\mathcal{X}$, the more uncertain we can potentially be about what value it will take.

The **redundancy** W is the difference between this maximum entropy and the actual entropy of the random variable:

$$W(\mathrm{X}) = H_{\max}(\mathcal{X}) - H(\mathrm{X}).$$

General data compression limits

In order to make the example in **Figure A.2** illustrate the concept of entropy, we've made two convenient choices that won't necessarily be true in general.

First, we've chosen the probabilities to be all of the form $\frac{1}{2^k}$ where k is a positive integer. This makes the information in each event equal to an integer number of bits, which means we can fully compress the sequence by encoding each event individually. If we hadn't done this, the information content of a single event might equal something like $\frac{3}{4}$ bits, and encoding in a one digit binary string would waste $\frac{1}{4}$ bits of space and yield a redundant binary representation. This technicality is why the entropy is in general less than or equal to the expected length of the shortest binary encoding, rather than always exactly equal. We've specifically picked the case where it is equal for illustrative purposes. It is possible to get closer to this bound in the more general case by encoding sequences of multiple events together into a single binary string, otherwise known as **block encoding**.

The second simplification we've made is that the random sequences of colors we've chosen are made up of exactly the expected number of each color. For example, the bottom sequence is half ($\frac{8}{16}$ events) `blue`, a quarter ($\frac{4}{16}$ events) `gray`, etc. In reality, different random sequences might yield binary strings with very different lengths. For example, a sequence of all `blue` (which happens to be the most probable sequence) would have a 16 bit binary encoding, while a string of all greens (one of the least probable sequences) would yield a 48 bit binary encoding. These sequences in **Figure A.2** are called **typical sequences**.

Typical sequences

A **typical sequence** is an outcome which has information close to the entropy of random event to which it belongs. The **typical set** is the set of all such typical sequences, which is usually a small subset of all possible sequences. For a sequence of N independent and identically distributed random variables $\mathrm{X}_1, \mathrm{X}_2, ..., \mathrm{X}_N$, the average information content is $H(\mathrm{X}_1) + H(\mathrm{X}_2) + ... + H(\mathrm{X}_N) = NH(\mathrm{X})$. For a small positive number ϵ, a typical sequence is one that satisfies:

$$H(\mathrm{X}) - \epsilon \leq -\frac{1}{N} \log p(x_1, x_2, ..., x_N) \leq H(\mathrm{X}) + \epsilon.$$

Does the choice of ϵ matter? For the sequences in **Figure A.2**, we've conveniently avoided this question since the sequences shown have information content exactly equal to $NH(\mathrm{X})$. However, as $N \to \infty$, we would see essentially the same behavior, because all of the probability mass of the distribution over sequences concentrates on the typical set defined with *any positive value of* ϵ (**Fig. A.3**). Furthermore, as $N \to \infty$, all typical sequences will occur with approximately the same probability: $\approx 2^{-NH(\mathrm{X})}$. Since the total probability is 1, and each typical sequence occurs with probability $\approx 2^{-NH(\mathrm{X})}$, there must be $\approx 2^{NH(\mathrm{X})}$ typical sequences. This result is known as the **asymptotic equipartition property (AEP)**. It is a direct consequence of the Law of Large Numbers, and it is used to prove many important theorems in information theory.

To summarize, the AEP says that with infinite sequence length, we can ignore the vanishingly small probability that falls on non-typical sequences. As a result, we can focus on the typical set of $\approx 2^{NH(\mathrm{X})}$ sequences, which each occur with probability $\approx 2^{-NH(\mathrm{X})}$.

A data compression scheme that encoded only sequences in the typical set and discarded all other sequences would require $\log_2 2^{NH(\mathrm{X})} = NH(\mathrm{X})$ bits to assign a unique binary string to each element of the typical set. Since there is negligible probability mass outside the typical set, the scheme would be lossless in the limit as $N \to \infty$. Thus, lossless compression of a length N sequence requires $NH(\mathrm{X})$ bits. Using fewer bits would result in different typical sequences mapping to the same binary string. Using more would waste bits on non-typical sequences that occur with nearly zero probability.

Interestingly, the typical set does *not* include the most probable individual sequences. For example, in **Figure A.3b**, the sequence for each N that consists of N `blue`s in a row is the most probable sequence, and would be omitted from the typical set for a small value of ϵ. For any finite N you could improve the compression algorithm described above by taking the binary string used for the least probable typical sequence and instead using it for the sequence of all `blue`. The demonstrates an important point about the typical set. It is useful for theoretical work, because the number of sequences it contains can be counted. However, in practical situations (i.e. $N < \infty$), better compression can be achieved by designing for the set of *most probable* sequences. As $N \to \infty$, this becomes less and less true, because the most probable sequences contain vanishingly small total probability mass.

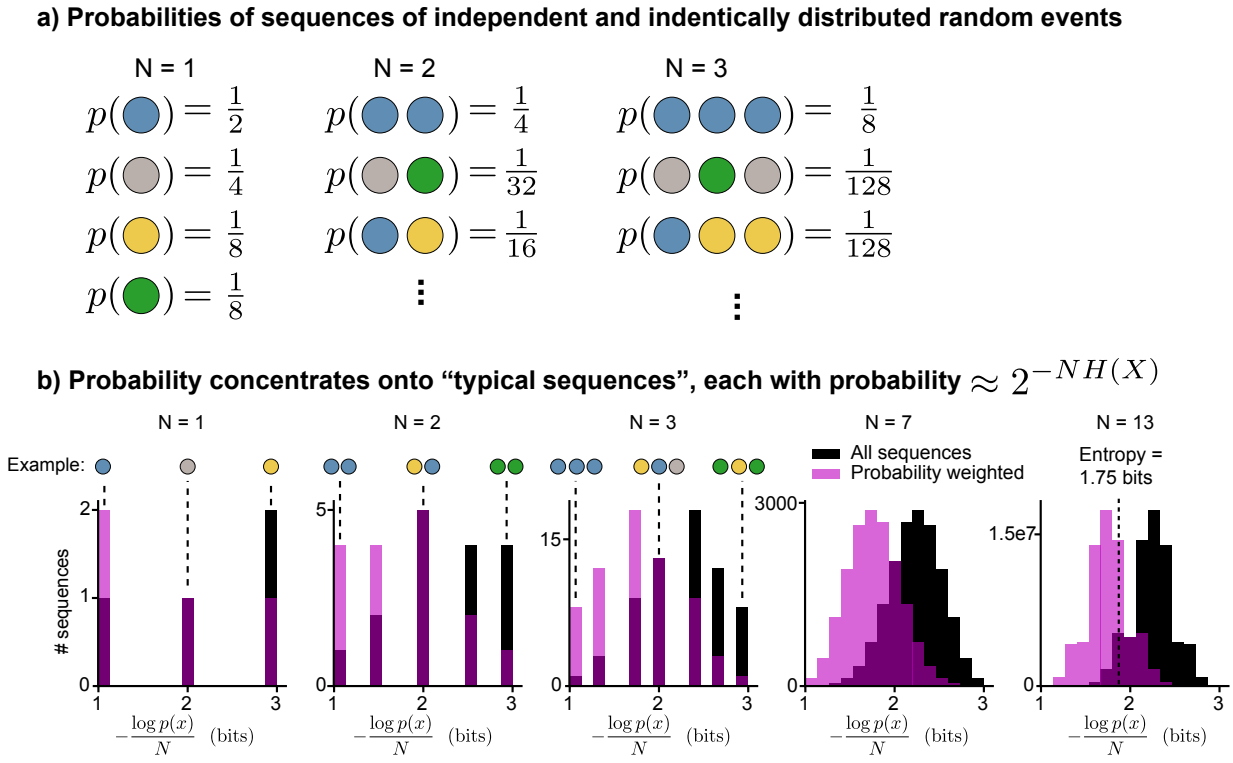

Figure A.3: **Typical sequences a)** Example sequences of independent and identically distributed events with increasing increasing length (N). **b)** Histograms of the information (i.e. $-\frac{\log p(x)}{N}$) of each possible sequence with length N. Black shows the histogram of every possible sequence. Magenta shows the distribution of probability-weighted sequences (i.e. the expected distribution one would get by taking a random sample). As N increases, nearly all of the probability mass concentrates on a tiny subset of the total number of sequences: typical sequences. There are $\approx 2^{NH(\mathrm{X})}$ typical sequences each with probability $\approx 2^{-NH(\mathrm{X})}$.

The relationship between entropy and redundancy, probability distributions, and typical sequences are shown in **Figure A.4**.

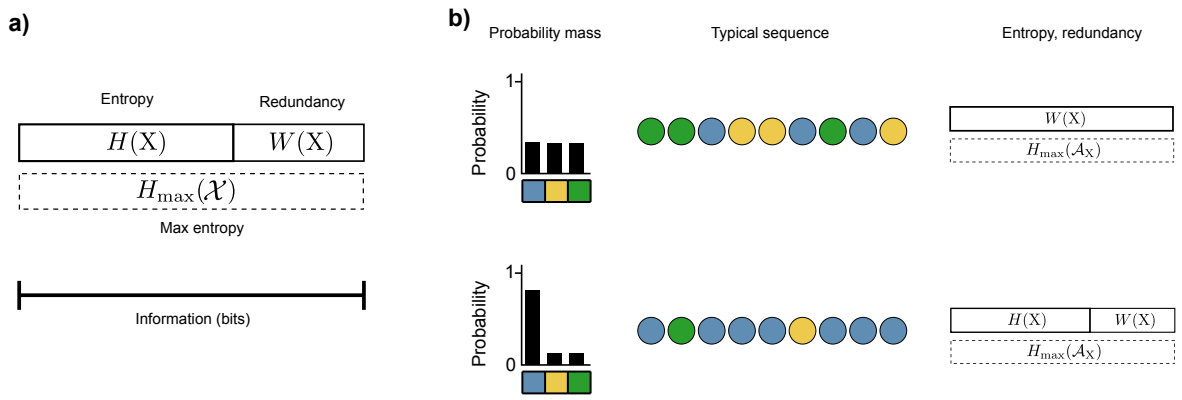

Figure A.4: **Probability, redundancy, and typicality**. **a)** The redundancy of a random variable X is equal to the difference between its entropy $H(\mathrm{X})$ and the maximum possible entropy on its probability space $H_{\max}(\mathcal{X})$. **b)** Distributions with more concentrated probability mass have higher redundancy. (Top) The equal probability case, (Bottom) the concentrated probability case. (Left) Probability distribution over a single event of an independent and identically distributed sequence, (Middle) a typical sequence of events from this distribution. (Right) The entropy, redundancy and maximum entropy.

Mutual information

Having described how to quantify the information content of a random event, we now consider the more general case of how information can be shared among multiple random events–that is, how how much uncertainty will be reduced about a random event by knowing the outcome of a different random event. This provides the basis for reasoning about and calculation the transmission of information.

Consider the scenario of drawing not colored marbles, but colored objects from an urn. We have the same four possible colors, `blue`, `green`, `yellow`, and `gray`, and now there are also four possible object shapes: ♦, ▲, ★, and ●. Our random draws will now be shape-color combinations like ▲, ♦, ♦, etc. X will still represent the represent the object's color, and now we'll also use Y to represent its shape. The random event of drawing a particular shape with a particular color is X, Y, and its value is governed by the joint probability distribution over $\mathcal{X} \times \mathcal{Y}$, the outcome space of all shape/color combinations.

Suppose that we do not observe which object was drawn, but only a black and white photograph of the object in which all colors look the same. What can we say about the object's color (X), having only observed its shape Y? In other words, how much information

does its shape convey about its color? The answer to this question is determined by the **mutual information** between X and Y.

Mutual information, denoted $I(\mathrm{X};\mathrm{Y})$, quantifies how well you can discriminate alternative possibilities of an unknown random event X based on observing some other random event Y. In other words, it tells you how much of the information gained from observing Y = ▲ will reduce your uncertainty about the object's color, X. The minimum possible value is $I(\mathrm{X};\mathrm{Y}) = 0$, which means that X and Y are independent–observing shape will not reduce your uncertainty about color at all. The maximum possible value is the lesser of $H(\mathrm{X})$ and $H(\mathrm{Y})$, because a random event can neither convey more information about another event than it itself contains, nor can it convey more information about the other event than the other event contains. Mutual information is symmetric: $I(\mathrm{X};\mathrm{Y}) = I(\mathrm{X};\mathrm{Y})$.

Another common way of quantifying the dependence between two variables is through correlation. In some sense, mutual information can be thought of as a more general version of correlation, because unlike correlation, which usually captures only linear relationships between the variables, mutual information captures any sort of statistical dependency between them.

To compute mutual information, we need to know the joint probability distribution of the two variables X and Y. That is, we need to know which shapes come in which colors. For a particular draw, say ★, we can calculate the amount of information that the color and shape convey about one another using the **point-wise mutual information**:

$$\log\left(\frac{p(x,y)}{p(x)p(y)}\right).$$

Point-wise mutual information is analogous to the information of a single outcome. It can be rewritten in a few different ways, which allow for different interpretations. For example:

$$\begin{aligned}\log\left(\frac{p(x,y)}{p(x)p(y)}\right) &= \log\left(\frac{p(x \mid y)}{p(x)}\right)\\ &= \log p(x \mid y) - \log p(x)\\ &= \log\frac{1}{p(x)} - \log\frac{1}{p(x \mid y)}\end{aligned}$$

The final line can be interpreted as the surprise of the outcome x minus the surprise of x when y is already known. If knowledge that $\mathrm{Y} = y$ makes $\mathrm{X} = x$ less surprising, then $\frac{1}{p(x|y)}$ will decrease, reflecting the fact that y tells us something about the value of x. For example, if there are many different colors of objects, but all ★'s are `blue`, then learning the object is a ★ will eliminate all uncertainty about its color: $p(\text{★} \mid \text{★}) = 1$. This means that in this case all of the information provided by the outcome X = `blue` is shared by Y = ★.

By taking a probability-weighted average of the point-wise mutual information over all possibilities, we arrive at the **mutual information**, which represents the average amount

of information provided from the outcome of one random event that is relevant to another. Mathematically:

$$I(\mathrm{X};\mathrm{Y}) = \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(x,y)\log\left(\frac{p(x,y)}{p(x)p(y)}\right) \tag{A.1}$$

Just as in **Figure A.1**, when one bit of information meant we could rule out half of the probability mass, for each bit of mutual information between X and Y, we can eliminate half of the probability mass of one random event after observing the outcome of the other. This is most easily seen if all outcomes are equally probable, because in this case eliminating half of the probability mass corresponds to eliminating half of the outcomes.

Figure A.5 shows three different scenarios with different amounts of mutual information, and four different ways of visualizing the relationships between the two variables. When $I(\mathrm{X};\mathrm{Y}) = 0$ bits, observing the object's shape does not allow us to eliminate any possibilities of the object's color. A mutual information of 0 between two random variables is equivalent to those two random variables being independent. Alternatively, when $I(\mathrm{X};\mathrm{Y}) = 1$ bit, after observing the shape only, we can eliminate two of the four possible colors. Finally, when $I(\mathrm{X};\mathrm{Y}) = 2$, we can eliminate 3 of the four possible colors and know exactly what color it is. In this case, we have gained all 2 bits of information that were present in X.

Since we can at most gain 2 bits from observing 1 of 4 possible shapes (i.e. $H_{\max}(\mathcal{Y}) = 2$), if there were more than 2 bits of information in X, we wouldn't be able to determine color exactly. For example, if there were 8 possible colors that were all equally probable, we could not uniquely identify all possibilities based on 4 possible shapes.

Joint and conditional entropy

Using entropy and mutual information, we can now describe two other important quantities in information theory: **conditional entropy** and **joint entropy**.

Conditional entropy

Mutual information quantifies how much knowing the outcome of one random event reduces uncertainty about a second random event, while conditional entropy quantifies how much uncertainty remains about the second random event. Adding the two together gives the total uncertainty of the second random event on its own.

Once again, we start with an example in the point-wise case, where the information gained by learning the outcome of a random event X after already knowing the outcome of a second random event $\mathrm{Y} = y$ is:

$$\log\frac{1}{p(x \mid y)}$$

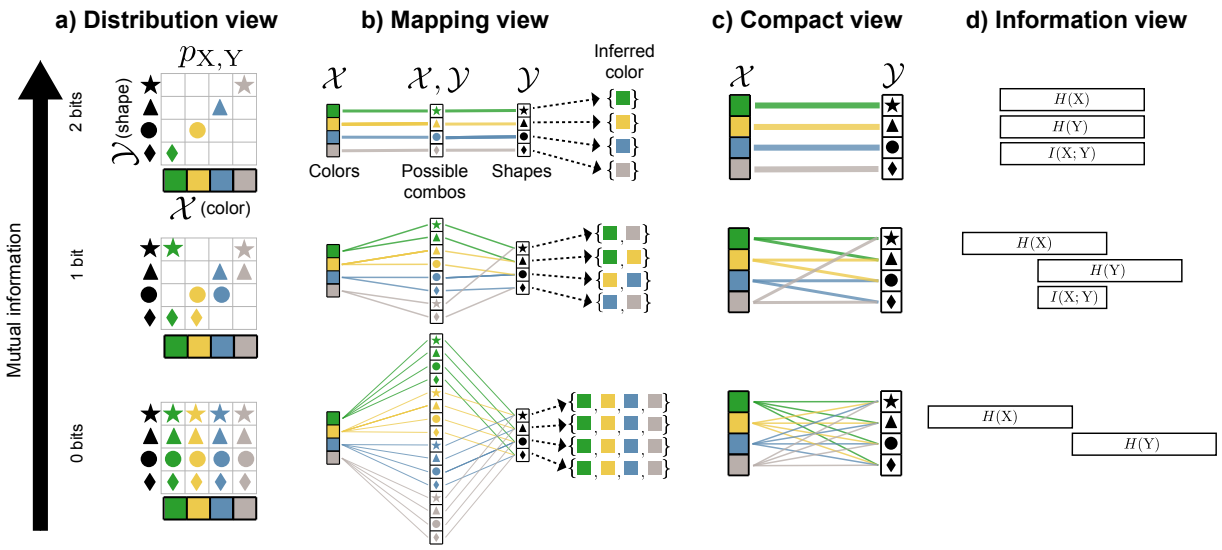

Figure A.5: **Mutual information** describes the relationship between two random variables. Here those random variables are the shape and color of an object drawn at random. The joint distribution of shape and color determines the amount of mutual information. (**Top row**) 2 bits of mutual information, (**middle row**) 1 bit of mutual information, (**bottom**) 0 bits of mutual information. **a)** The joint distribution of shape and color, with uniform probability over all possible shape/color combinations shown. **b)** Mapping view showing the colors, possible shape color combinations, possible shapes, and the possibilities for colors that can be inferred from shape alone. Line thickness shows strength of the relationship. **c)** Compact view that omits the joint distribution and color inference. **d)** More of the entropy of the two events is shared with greater mutual information.

Or, in the specific case of learning an object is green after knowing it is a ●:

$$\log \frac{1}{p(\texttt{green} \mid \bullet)}$$

The surprise of finding a green ● depends on how many different colors of ● are in our set of objects. In the same way we calculated entropy as a probability-weighted average of all possible outcomes of a random event, we can similarly compute an average uncertainty of the object's color, given that the object is a ● (**Fig. A.6a**). This yields the point-wise conditional entropy:

$$H(\mathrm{X} \mid \bullet) = \sum_{\substack{x \in \{\texttt{green},\texttt{blue},\\ \texttt{yellow},\texttt{gray}\}}} p(x \mid \bullet) \log \frac{1}{p(x \mid \bullet)} \tag{A.2}$$

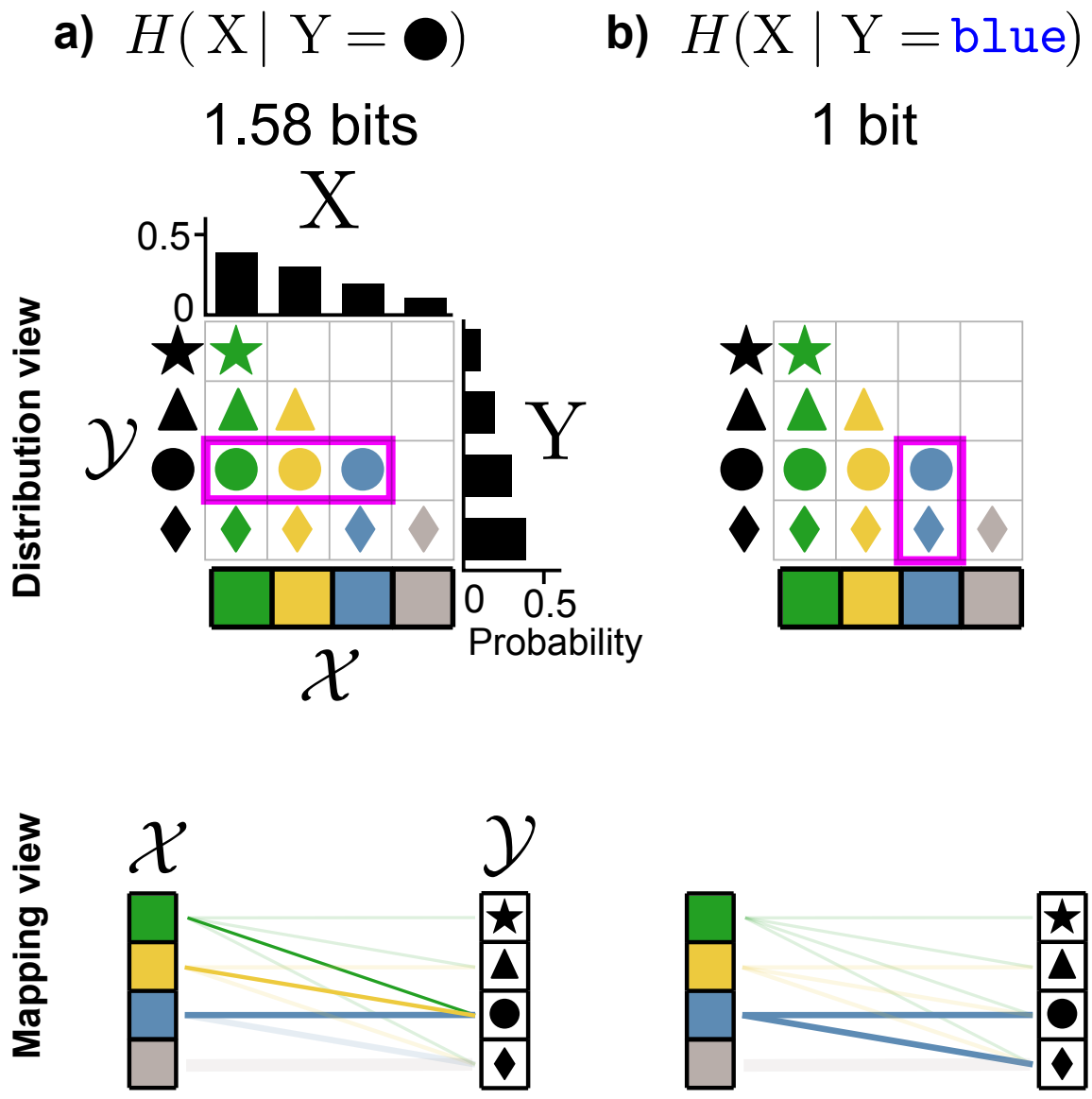

Figure A.6: **Point-wise conditional entropy** (**Top**) The joint and marginal distributions of shape and color. (**Bottom**) The compact mapping view between shape and color. **a)** The conditional entropy of color given the shape is ● is $\log_2 3 \approx 1.58$ bits since there are three equally probably possibilities (in magenta box on distribution view) **b)** The conditional entropy of shape given a `blue` object is 1 bit since there are two equally probable shapes.

Which can also be written in the generic form as:

$$H(\mathrm{X} \mid y) = \sum_{x \in \mathcal{X}} p(x \mid y) \log \frac{1}{p(x \mid y)}$$

Looking at the joint distribution in the middle row of **Figure A.5a**, knowing that the object is a ● leaves three possible colors: `green`, `yellow`, or `blue`. Since we've specified that these occur with equal probability, the point-wise conditional entropy will be $\log_2 3 \approx 1.58$ bits.

Alternatively, if we looked at the conditional entropy of color given that the object is a ★, the only possible color would be `green`. This means the point-wise conditional entropy is 0–we know the value of X exactly given Y.

Next, we might want to consider the average uncertainty of color, conditional not specifically on ♦, but averaged over all possible shapes. To do so, we'll sum equation (A.2) over all possible states in $\mathcal{Y}$ (e.g. ★,♦,▲,●), weighting by the probability of each shape. The generic form of this is:

$$H(\mathrm{X} \mid \mathrm{Y}) = \sum_{y \in \mathcal{Y}} p(y) \sum_{x \in \mathcal{X}} p(x \mid y) \log \frac{1}{p(x \mid y)}$$

With some slight algebraic manipulation, we get to the traditional equation for **conditional entropy**:

$$H(\mathrm{X} \mid \mathrm{Y}) = \sum_{y \in \mathcal{Y}} \sum_{x \in \mathcal{X}} p(x, y) \log \frac{1}{p(x \mid y)}$$

When $H(\mathrm{X} \mid \mathrm{Y}) = 0$, after observing the outcome of Y we know exactly what the outcome of X is – there is no remaining uncertainty. Alternatively, if $H(\mathrm{X} \mid \mathrm{Y}) > 0$ then the outcome of X is not fully known having observed Y, at least for some outcomes of Y.

Joint entropy

Joint entropy is an extension of entropy to multiple random variables, which takes into account the dependence between them (i.e. the mutual information):

$$H(\mathrm{X}, \mathrm{Y}) = \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x, y) \log \frac{1}{p(x, y)}$$

For example, $p(x, y)$ in the example shown in **Figure A.5** would represent the probability of a particular shape-color combination.

In the case that X and Y are independent, observing the value of Y provides no information about X. Thus, $H(\mathrm{X},\mathrm{Y}) = H(\mathrm{X}) + H(\mathrm{Y})$. This can be proven algebraically by substituting $p(x,y) = p(x)p(y)$ into the joint entropy formula, as shown in Section A.7. This can be equivalently expressed as $p(y) = p(y \mid x)$ or $p(x,y) = p(x)p(y)$.

When X and Y are dependent, the joint entropy will be less than the sum of the individual entropies, since there will be nonzero mutual information. For example, in the top row of **Figure A.5**, $H(\mathrm{X}) = 2$, $H(\mathrm{Y}) = 2$, but $H(\mathrm{X},\mathrm{Y}) = 2$ also, since the 2 bits are mutual to both random variables.

Relationships between entropy and mutual information

Entropy, joint entropy, conditional entropy, and mutual information can be described in terms of each other. Their relationships are summarized in **Figure A.7**. Entropy is the average uncertainty in a single random event, mutual information is the amount of that uncertainty that is in common with another random event through some statistical relationship. Conditional entropy is amount of remaining uncertainty in one random event after subtracting the mutual information shared with another random event. Joint entropy is the average uncertainty when describing both random events together.

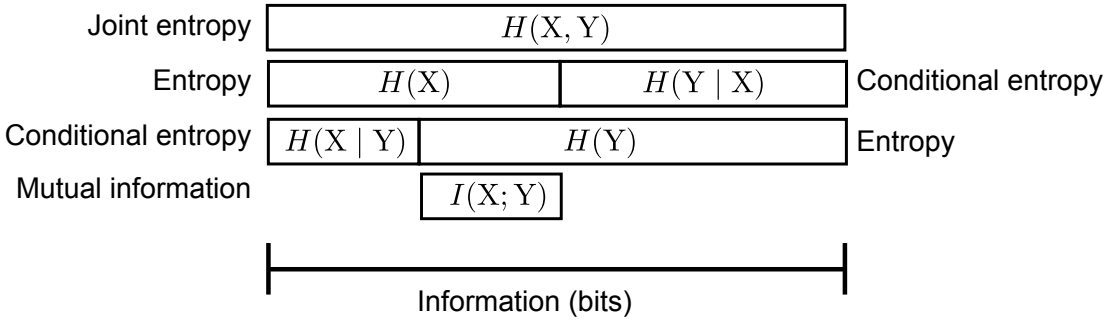

Figure A.7: **The relationships between entropy, joint entropy, conditional entropy and mutual information** The width of each bar represents its size (in bits). Adapted from [89], p140

Mutual information can be written in three different ways in terms of the other entropies:

$$\begin{aligned} I(\mathrm{X};\mathrm{Y}) &= H(\mathrm{Y}) - H(\mathrm{Y} \mid \mathrm{X}) \\ &= H(\mathrm{X}) - H(\mathrm{X} \mid \mathrm{Y}) \\ &= H(\mathrm{X}) + H(\mathrm{Y}) - H(\mathrm{X},\mathrm{Y}) \end{aligned}$$

An example of how $I(\mathrm{X};\mathrm{Y})$ can be manipulated into these other forms can be found in Section A.7.

The first two can be interpreted as the average uncertainty in one random event after subtracting the uncertainty that remains having observed a different random event. The third can be interpreted as the total uncertainty of the both random events in isolation minus the uncertainty of both considered together.

Entropy rate of stochastic processes

Thus far, we've discussed only the case in which the source of information is an independent and identically distributed (IID) random variable, but the information theoretic concepts we've introduced generalize to non-independent ordered sequences of random variables, also known as **stochastic processes**.

In the IID case, we have a random variable X, which has some probability density/-mass function $p_{\mathrm{X}}(x)$. A sequence of events can be represented by the random vector $\mathbf{X} = (\mathrm{X}_1, \mathrm{X}_2, ...)$, which is an ordered sequence of random variables. Its joint probability density/mass function is denoted $p_{\mathbf{X}}(\mathbf{x}) = p_{\mathrm{X}_1,\mathrm{X}_2,...}(x_1, x_2, ...)$. The IID assumption allows us to simplify this expression by factoring the joint distribution and writing each term as the shared probability density/mass function of X:

$$\begin{aligned} p_{\mathbf{X}}(\mathbf{x}) &= p_{\mathrm{X}_1,\mathrm{X}_2,...}(x_1, x_2, ...) \\ &= p_{\mathrm{X}}(x_1)p_{\mathrm{X}}(x_2)... \end{aligned}$$

Entropy is additive for independent random variables, so the entropy of $\mathbf{X} = (\mathrm{X}_1, \mathrm{X}_2, ...\mathrm{X}_N)$ will be equal to N times the entropy of X. Thus, $H(\mathrm{X})$, the entropy per event, is called the **entropy rate** of the process. This quantity is important because if we are trying to transmit information through sequential uses of a communication channel, our channel will need to be able to keep up with this rate (as discussed in greater depth in section A.4).

Alternatively, suppose that the this IID assumption does not hold. What is the entropy rate of a stochastic process with an arbitrary joint probability distribution? There are two ways we might do this, and under certain assumptions, they are equivalent ([23] Sec. 4.2).

Returning to the example of drawing marbles, suppose we are drawing from a magical urn that, the first time we draw from it, will give all four colors with equal probability. After the first draw, the magic kicks in, and the urn makes it more likely that we will again draw the same color as our previous draw. For example, the conditional probability $p_{\mathrm{X}_{N+1}|\mathrm{X}_N}(\texttt{blue} \mid \texttt{blue}) = \frac{5}{8}$, while $p_{\mathrm{X}_{N+1}|\mathrm{X}_N}(\texttt{green} \mid \texttt{blue}) = \frac{1}{8}$ (**Fig. A.8a**). If we write out all the possible conditional probabilities, we will get a matrix where each column represents the conditional distribution $p_{\mathrm{X}_{N+1}|\mathrm{X}_N}$. A typical sequence from this distribution will tend to give repeats of the same color (**Fig. A.8b**).

What can we say about uncertainties in this process? We are more uncertain about the color of the first draw X_1 (all colors with $\frac{1}{4}$ probability) than we are about the second color

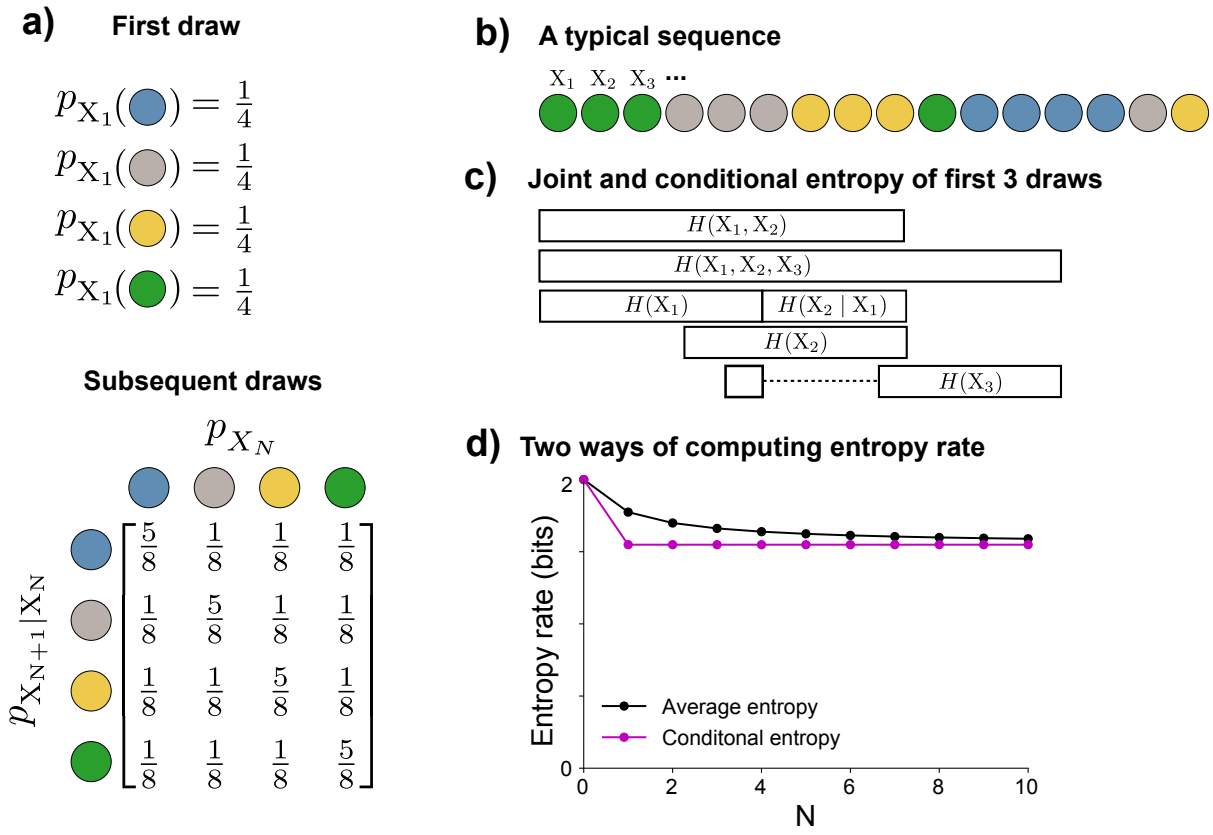

Figure A.8: **Entropy rate of a stochastic process a)** The stochastic process, which consists of **(top)** an initial draw of a colored marble with each color having a probability of $\frac{1}{4}$ and **(bottom)** subsequent draws where the probability of repeating the same color as the previous draw is $\frac{5}{8}$ and the probability of all other colors $\frac{1}{8}$. **b)** A typical sequence from this stochastic process where a random variable X_k represents the color selected at each position. **c)** Entropy, conditional entropy, and joint entropy of the first three draws. Knowledge of past outcomes reduces uncertainty of future outcomes (or vice versa). **d)** The two ways of computing entropy rate: the average of the joint entropy and the conditional entropy of the next draw given the previous. For a stationary stochastic process, these converge to the same value as $N \to \infty$.

after observing the first ($\frac{5}{8}$ chance of the same color again). This means that $H(\mathrm{X}_1) > H(\mathrm{X}_2 \mid \mathrm{X}_1)$ (**Fig. A.8c**). Knowing X_1 reduces our uncertainty about the value of X_2, and similarly knowing X_2 would reduce our uncertainty about the value of X_1. This means that the joint entropy is less than the sum of the individual entropies $H(\mathrm{X}_1, \mathrm{X}_2) < H(\mathrm{X}_1) + H(\mathrm{X}_2)$.

What about the **entropy rate** of the process? There are two ways we could define this. First, we could simply say that the entropy rate is our average uncertainty over all draws. This can be quantified by dividing the joint entropy by the number of draws to get an average of the information in each draw:

$$\frac{1}{N} H(\mathrm{X}_1, \mathrm{X}_2, ...\mathrm{X}_N) \tag{A.3}$$

Alternatively, we could say that the entropy rate is our uncertainty about the next draw, given the previous ones. This is quantified with the conditional entropy:

$$H(\mathrm{X}_{N+1} \mid \mathrm{X}_N, \mathrm{X}_{N-1}, ...\mathrm{X}_1) \tag{A.4}$$

Under our example, (A.3) will gradually decrease as N increases, as our high uncertainty about the first draw has a proportionally smaller and smaller effect compared to the subsequent, more predictable draws.

For the second definition, since our model specifies that draw $N+1$ only depends on the outcome of draw N (a type of model called a **Markov chain**), (A.4) will simplify to:

$$H(\mathrm{X}_{N+1} \mid \mathrm{X}_N) \tag{A.5}$$

Thus, it will have a large value for X_1 and then be slightly lower and stay constant for all subsequent draws.

As $N \to \infty$, both measures will converge to the same value for the entropy rate because the high uncertainty of the first draw is ultimately neglected for both, so the two definitions are equivalent in the limit. This will be true whenever the stochastic process is **stationary**, which means that its joint probability distribution is shift-invariant. Mathematically, a stationary process has the property:

$$p_{\mathrm{X}_{N+1}|\mathrm{X}_N,\mathrm{X}_{N-1},...} = p_{\mathrm{X}_{N+1+k}|\mathrm{X}_{N+k},\mathrm{X}_{N-1+k},...}$$

Where k is some arbitrary integer.

It is often useful to consider stationary processes in an asymptotic sense. For example, the process in **Figure A.8** is not stationary in that $p_{\mathrm{X}_2|\mathrm{X}_1} \neq p_{\mathrm{X}_{102}|\mathrm{X}_{101}}$. However, as $N \to \infty$ these differences disappear.

Redundancy and stochastic processes

Moving from a sequence of independent and identically distributed random variables to a stochastic process requires revisiting the definition of **redundancy** provided in section A.2. We can no longer define redundancy with respect to only the entropy of a single random variable in the sequence (e.g. $H(\mathrm{X}_k)$). The more general definition of redundancy must account for the joint entropy:

$$W(\mathbf{X}) = H_{\max}(\mathcal{X} \times \mathcal{X} \times ...) - H(\mathrm{X}_1, \mathrm{X}_2, ...)$$

Essentially, this means that redundancy is now a function of higher order interactions between variables rather than just of the distribution of each random variable individually. In the example above $H(\mathrm{X}_k) = H_{\max}(\mathcal{X})$, all colors are equally likely for every draw in the absence of any information about the outcomes of other draws. But the joint distribution in our example is not the maximum entropy distribution, because the information present in the distribution of one random variable is shared by others. This manifests in the fact that we can do much better than random chance in guessing the next color in a sequence given knowledge of the previous ones. The maximum entropy joint distribution would be independent and identically distributed random events, each of which has equal probability over all of its states.

Probability densities and differential entropy

The examples above all considered probability mass functions over finite, discrete sets of events. How do the formulae and interpretations generalize to cases other than this, such as probability density functions defined over the infinite real line?

For probability densities that are defined over the continuous real line (e.g. the normal distribution, the exponential distribution), we can define an analog to entropy called the **differential entropy**, replacing the discrete sum with an integral:

$$H(\mathrm{X}) = -\int_{\mathcal{X}} p(x) \log p(x) dx \tag{A.6}$$

where $p(x)$ is the probability density function of X.

The differential entropy does not have the same clear interpretations as the discrete entropy. In particular, it cannot be interpreted as the number of bits needed to describe the outcome of a random variable with that probability density. In fact, it would take an infinite number of bits to describe a real number with arbitrary precision. The differential entropy can also take negative values unlike its discrete counterpart. Finally, as is pointed out in ([89] p180), equation (A.6) is actually improper in the sense that it takes the logarithm of a dimensional quantity, the probability density (which has units $\frac{1}{x}$). For this reason, the value of differential entropy will change with a change of units.

One way to resolve these difficulties is to convert the continuous probability density into a discrete probability mass function by dividing the real line into equally spaced intervals with width Δ and integrating the probability density within each interval to get a probability mass function ([23] section 8.3). One can then compute the discrete entropy of this distribution, which will be equal to the differential entropy minus $\log \Delta$. In other words, the more fine grained our discretization of the probability density, the more bits are needed to describe it. In many cases we may work directly with such discretized probability densities on a computer, and it is important to remember that the absolute value of the entropy depends on an additive factor based on our arbitrary choice of discretization interval.

The maximum entropy distribution for a given state space also works differently in the case of continuous probability densities. If our state space $\mathcal{X}$ is the set of real numbers $\mathbb{R}$, then the entropy can be driven infinitely high by making an infinitely wide distribution. However, we can still describe a maximum entropy distribution under certain constraints. For example, the maximum entropy distribution on $\mathbb{R}$ with a variance equal to σ^2 is a normal distribution with that variance. Many commonly used distributions are optimal under similar constraints, and maximum entropy distributions have many important applications in statistical inference and statistical physics that are beyond our scope here.

Luckily, the mutual information of two continuous probability densities does not suffer from the same mathematical difficulties and lack of clear interpretability. Even though it may take an infinite number of bits to specify an arbitrary real number, the interpretation that 1 bit of information allows you to rule out half of the probability mass remains valid. Mathematically, mutual information in the case of probability densities is equivalent to the discrete case, with the sums replaced by integrals:

$$I(\mathrm{X};\mathrm{Y}) = \int_{\mathcal{X}} \int_{\mathcal{Y}} p(x,y) \log \left(\frac{p(x,y)}{p(x)p(y)} \right) dxdy$$

The mutual information also avoids the problem of the improper logarithm of the differential entropy, because for mutual information, the log is taken on the ratio of densities, in which the units cancel out. It is also possible to compute mutual information between discrete and continuous random variables [130].

In summary, in spite of the difficulties associated with the differential entropy over probability densities, mutual information retains its usefulness and a clear interpretation as the amount of information one random variable provides about another.

Rate distortion theory

What happens if we try to compress data to a size smaller than its information content, or infer the outcome of a random event about which we have incomplete information? These questions can be answered by **rate-distortion theory**.

One application area of rate distortion theory is in lossy compression, where we're trying to compress some source of events to the smallest size possible, but we're willing to tolerate

some level of error upon decompression. That is, the decompressed message will be "close to" but not exactly the same as the compressed source. Because we can tolerate some error, we need not preserve all of the information about the source in the message.

Figure A.9 provides an example of this. We have some source of information, like the equally probable (maximum entropy) four-color marble scenario of **Figure A.2**. We'll take a series of random colors from this source, and pass it into a deterministic compressor function, which will map it to a binary string. Next, we pass that binary string into a deterministic decompressor function, which tries to reproduce the original string. We want our compressor to map the sequence to the shortest binary string possible (on average). In **lossless compression** (**Fig. A.9a**), we impose the requirement that the decompressed binary string must match exactly the original sequence. In **lossy compression** (**Fig. A.9b**), we're willing to tolerate some errors upon decompression, which will allow us to discard some of the source information and achieve an even shorter binary encoding.

There is a trade-off between how much information we discard and how many errors will be present after decompression: the more information we throw away, the shorter a binary encoding we can make, but the more errors we will make upon decompression. The **rate-distortion curve** quantifies this trade-off. The rate quantifies the amount of information we have about the source, and distortion quantifies the number and severity of the errors we make. The rate-distortion curve applies not just to data compression problems, but to any situation in which we have incomplete information about a random event, such as the imperfect transmission of data.

Mathematically, a **distortion** function $d(x, \hat{x})$ takes in an original event on the space of $\mathcal{X}$ (e.g. a color) and a decompressed (and possibly distorted) event on the space $\hat{\mathcal{X}}$, and it computes a non-negative real number that represents the distortion or difference between them:

$$d : \mathcal{X} \times \hat{\mathcal{X}} \to \mathbb{R}^+$$

In the example shown in **Figure A.9**, $\mathcal{X}$ and $\hat{\mathcal{X}}$ are identical (i.e. the set of the four possible colors), but this may not be true in every situation. For example, we might consider the distortion associated with storing an arbitrary real number as a 32-bit floating point number on a computer. In this case, $\mathcal{X}$ would be the set of real numbers and $\hat{\mathcal{X}}$ would be the set of possible 32-bit floating point numbers.

The source produces information at some rate, which we'll call the "source rate", whereas the **rate** describes the how much of it we're capturing it (over, for example, time, space, etc.). Higher rates are good, because we're not losing source information. For a discrete source, the maximum achievable rate is the entropy rate of the source $H(\mathrm{X})$.

A useful scalar descriptor of the distortion for a particular source and encoding scheme is the average value of a distortion function applied to a source and some distorted version of that source. For example, this could be the average number of differences (the distortion function) between a binary string from a source and the string found by passing the original string through a lossy encoder and then decoder functions (as shown in (**Fig. A.9b**)):

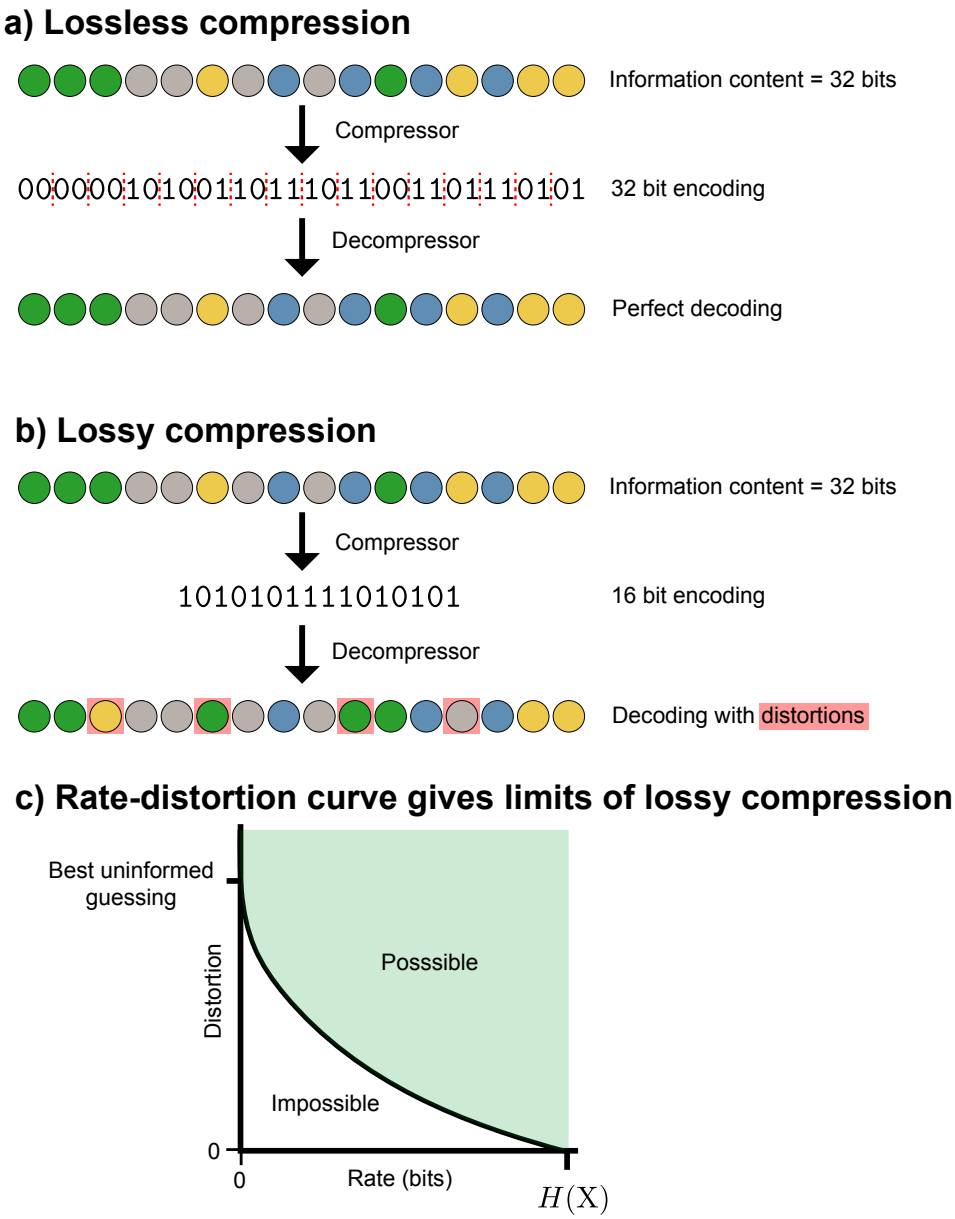

Figure A.9: **Lossy compression and rate distortion a**) In lossless compression, a typical sequence is mapped to a binary codeword with length equal to its information content and can be decompressed without error. **b**) In lossy compression, a sequence is mapped to a sequence shorter than its information content, and errors are present upon decompression. **c**) The black curve shows the minimum number of bits needed to achieve a given average distortion.

$$\mathrm{D} = \mathbb{E}\left[d(\mathrm{X}, \hat{\mathrm{X}})\right]$$

The limiting cases of complete information (where distortion may be zero) and zero information (where nonzero distortion is impossible except in the trivial case of a constant source) suggest that more information allows for lower distortion. Rate-distortion theory takes this intuition even further, proving that if we want to achieve distortion below some threshold (i.e. $\mathrm{D} < \mathrm{D}_{\max}$), we must have at least $R(\mathrm{D})$ bits of information about X. $R(\mathrm{D})$ is called the **rate-distortion function**. This means that if we're already achieving the best average distortion for the amount of information we have, $\mathrm{I}(\mathrm{X}; \hat{\mathrm{X}})$, we cannot do better unless we acquire more information. This is true regardless of the chosen distortion function, provided that the distortion function $d(x, \hat{x})$ is finite for every possible x and $\hat{x}$. This finding is proven in the asymptotic case of infinitely long block lengths.

This doesn't necessarily mean that having N bits about a source will give us an $\hat{\mathrm{X}}$ that achieves the N-bit best performance for a given distortion function. For example, if we know X perfectly, and we applied an invertible operation like mapping each color to a new, distinct color, we would not have lost any information. However, if our distortion function was the number of incorrect colors, the average distortion would increase.

An example rate-distortion function for a discrete source in shown in **Figure A.9c**. The green area represents possible combinations of rates and distortions. For sufficiently high values of average distortion, no information about the source is necessary, so the curve intersects the horizontal axis. For example, consider simply guessing the average outcome each time. For discrete sources, the curve intersects the vertical axis at the value of the source's entropy, at this point we have zero remaining uncertainty. In contrast, for a continuous source, the curve would asymptotically approach the vertical axis, since an infinite number of bits would be needed to describe a continuous source exactly (Sec. A.2).

In general, $R(\mathrm{D})$ will be a non-increasing convex function of D, which means that 1) achieving lower and lower distortion will always require acquiring more information (assuming we we already making good use of the information we had), and 2) there will be diminishing returns on lowering distortion as we acquire more and more information.

A.3 Channels

Thus far we've considered the mathematical foundations of information theory in a general sense, without specifying what gives rise to the statistical relationship between X and Y. Now we move to a more a specific scenario that models the transmission of information through a **channel**. Often, channels model an actual physical medium for information transmission, like a band of frequencies of radio waves or transistors on a computer.

Mathematically, a channel is some mapping from inputs in $\mathcal{X}$ to outputs in $\mathcal{Y}$. Rather than using the term "outcome" to describe the value taken by a random variable as in the

previous section, we now use the terms "input" and "output". Which inputs map to which outputs is determined by the conditional probability distribution $p_{\mathrm{Y|X}}$. In the special case that $p_{\mathrm{Y|X}}$ can be described by a deterministic function, then we have a **noiseless channel**; otherwise, we have the more general case of a **noisy channel**.[1]

Channels as matrices

A noisy channel is equivalent to a conditional probability distribution $p_{\mathrm{Y|X}}$, and in the case of a discrete outcome space, this can be visualized either as a mapping or a matrix (**Fig. A.10a**). In the matrix form, which we denote $\mathbf{P}_{\mathrm{Y|X}}$, each column is the conditional distribution $p(y \mid \mathrm{X} = x)$, which describes the probability that the input x will map to each possible output in $\mathcal{Y}$, This describes the noise distribution of that input. Since each column is itself a probability distribution, it sums to 1.

Using the channel matrix, two quantities of interest can be computed. The output distribution can be described by a vector of probabilities $\mathbf{p}_{\mathrm{Y}}$, and it can be computed from the input distribution $\mathbf{p}_{\mathrm{X}}$ by multiplying it by the channel matrix (**Fig. A.10b**):

$$\mathbf{p}_{\mathrm{Y}} = \mathbf{P}_{\mathrm{Y|X}}\mathbf{p}_{\mathrm{X}}$$

The joint distribution $\mathbf{P}_{\mathrm{X,Y}}$ is also useful because it is needed in the computation of conditional entropy and mutual information. It can be found by putting in the input probability vector $\mathbf{p}_{\mathrm{X}}$ along the diagonal of a matrix and multiplying by the channel matrix (**Fig. A.10c**):

$$\mathbf{P}_{\mathrm{X,Y}} = \mathbf{P}_{\mathrm{Y|X}}\mathrm{diag}(\mathbf{p}_{\mathrm{X}})$$

Unlike $\mathbf{P}_{\mathrm{Y|X}}$, for which each column is a probability distribution and sums to 1, all entries of $\mathbf{P}_{\mathrm{X,Y}}$ together sum to 1 since the whole matrix is a joint probability distribution.

Entropy and mutual information in channels

Entropy, conditional entropy, and mutual information have more specific interpretations for channels than general probability distributions, and they provide intuitions for how one might design a system for optimally transmitting information. Our goal is to recover as much of $H(\mathrm{X})$, the average information in at the channel inputs, as possible. The mutual information between X and Y measures the information shared between a transmitted and received message. The higher it is, the more information has successfully been transmitted to the receiver.

[1] This is true of a "memoryless" channel, which means that multiple channel uses are independent and identically distributed. All channels discussed here assume memorylessness

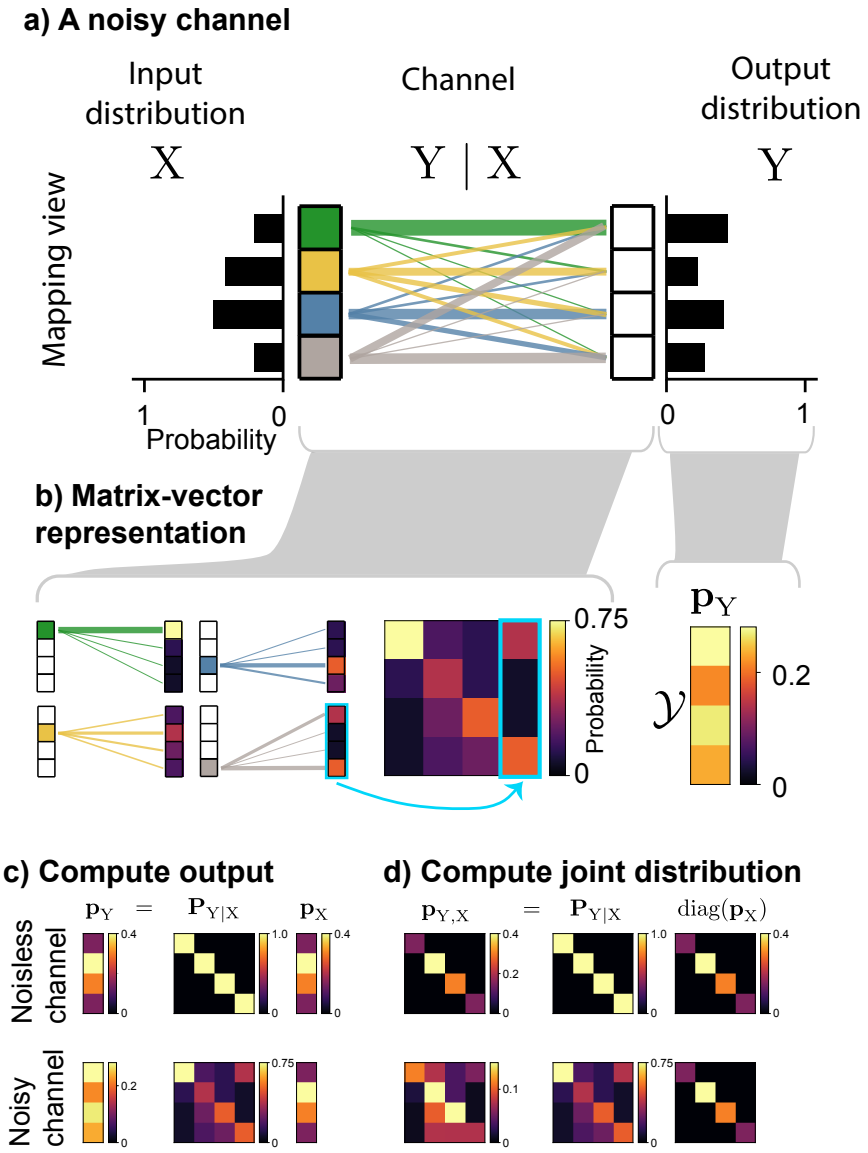

Figure A.10: **Channels a)** The mapping view of a noisy channel. The channel is mathematically represented by the conditional random variable Y | X **b)** Channels and inputs/outputs can also be represented by matrices and vectors of probabilities, respectively. The vectors sum to one since they represent probability distributions, and each column of the matrix sums to one since it represents the conditional distribution Y | X = x. The matrix/vector representations can be used to compute **c)** the output distribution and **d)** the joint distribution through matrix-vector and matrix-matrix multiplications, respectively. **c)** The joint distribution is computed from the matrix-matrix product of the channel and a matrix with the input matrix along the diagonal.

When considering information propagation through noisy channels, it is important to recognize that both signal and noise can be described in terms of entropy. In general two dependent random variables might have interesting or useful information on their own (like shapes and colors in Section A.2). In the case of a noisy channel, p_{X} contains the information we're interested in, and $p_{\mathrm{Y|X}}$ is the noise imparted by the channel. Since the definitions of entropy is based on probabilities alone, both $H(\mathrm{X})$ (the input) and $H(\mathrm{Y} \mid \mathrm{X})$ (the noise) are measured in bits and can technically be described mathematically as information. It is thus important to keep in mind this distinction between information that helps us infer the state of the source, and randomness that is irrelevant. The entropy of the channel output $H(\mathrm{Y})$ will usually contain both informaiton about the input and noise.

The information transmitted through a channel depends on both the channel itself ($p_{\mathrm{Y|X}}$) and the distribution over the channel's inputs (p_{X}). Two factors govern how much information is transmitted through the channel: the level of noise in the individual channel inputs, and the how much the outputs of different inputs overlap.

The noise level of different inputs can be computed from knowledge of the channel alone without knowledge of the input distribution. For input x, the noise at the output is $H(\mathrm{Y} \mid x)$. Assuming a particular input distribution allows us to compute the average noise imparted by the channel $H(\mathrm{Y} \mid \mathrm{X})$. Input distributions that put more probability on channel inputs that are less noisy will tend to transmit more information.

The other important factor is how much different inputs overlap at the channel output. The output overlap of a particular input with other inputs, averaged over all outputs is $H(x \mid \mathrm{Y})$. Unlike the noise level in a particular input, this overlap can only be calculated with respect to a particular input distribution, because it requires knowledge of the joint distribution of inputs and outputs. By averaging over all inputs, we can calculate the average uncertainty about which channel input gave rise to a certain output, $H(\mathrm{X} \mid y)$. Averaging over both inputs and outputs gives $H(\mathrm{X} \mid \mathrm{Y})$. Channels that have less overlap at the output will tend to transmit more information.

Examples of noisy and non-noisy channels We now give specific examples of channels that are either noisy or noiseless and either lose information about the inputs or preserve all of it. For simplicity, here we assume distribution over inputs is uniform. The ideal information transmission system would lose no information and be able to recover X exactly given a output Y, which would mean that $I(\mathrm{X};\mathrm{Y}) = H(\mathrm{X})$. One way this can occur is in a noiseless channel in which the channel maps every input to a unique output (**Fig. A.11a**). In this situation, $H(\mathrm{X}) = H(\mathrm{Y}) = I(\mathrm{X};\mathrm{Y})$ and $H(\mathrm{Y} \mid \mathrm{X}) = H(\mathrm{X} \mid \mathrm{Y}) = 0$, meaning that we could perfectly predict X given Y (or Y given X).

However, a channel does not need to be noiseless to completely preserve input information. All input information can be recovered through a noisy channel, so long as each input maps to outputs that don't overlap with those of other inputs (**Fig. A.11b**). In this scenario, $H(\mathrm{Y} \mid \mathrm{X})$ is no longer equal to zero, because even knowing what the input was, we cannot predict exactly how the noise of the channel manifests at the output. In contrast,

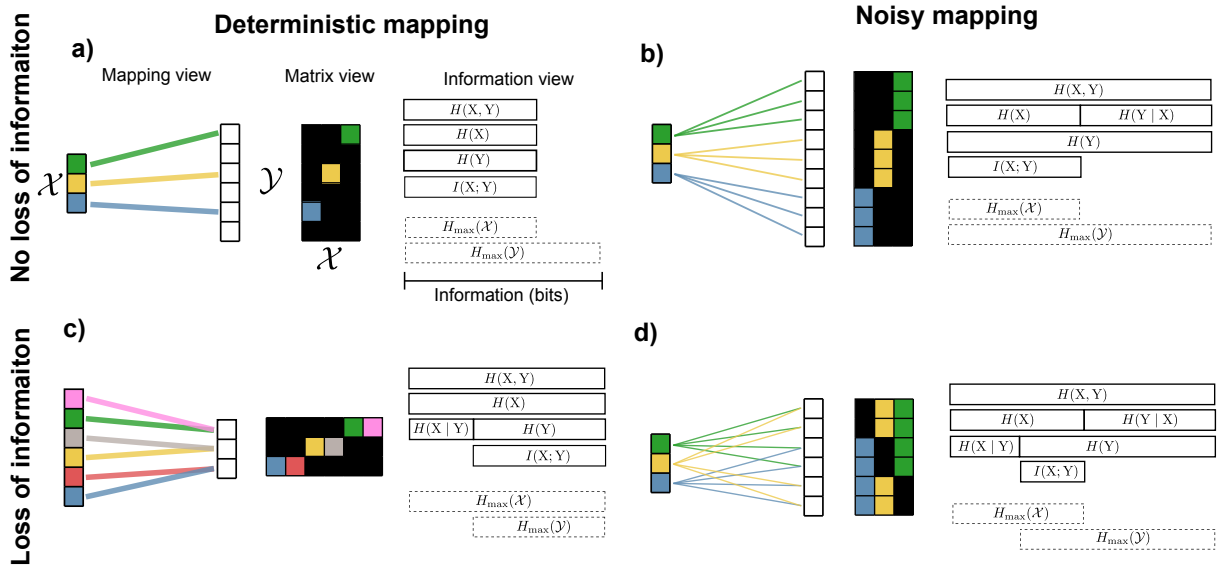

Figure A.11: **Examples of noisy and noiseless channels.** A random variable X with equal probability over all of its states is mapped to a random variable Y. These mappings can be either deterministic or noisy and information-preserving or information-destroying. Bars to the right of each mapping show entropy, conditional entropy, joint entropy, mutual information, and the maximum entropy over the state space of each random variable. Line thickness denotes magnitude of $p(y \mid x)$ **a)** A deterministic, information-preserving mapping with more outputs than inputs. Each input maps uniquely to exactly one output. The outputs with no line connected have zero probability. **b)** An information-preserving, noisy mapping. Each input maps into multiple outputs, but the outputs mapped to by a particular input are all disjoint. **c)** A deterministic, information-destroying mapping. Multiple inputs collide on the same output, meaning the mapping cannot be inverted. **d)** A noisy, information-destroying mapping. Each input maps to multiple outputs, and the outputs mapped to by each input are not disjoint.

$H(\mathrm{X} \mid \mathrm{Y}) = 0$ because given the output, we can know exactly what the input was. $H(\mathrm{Y})$ is greater than $H(\mathrm{X})$ because it includes all of the entropy $H(\mathrm{X})$ along with some additional entropy caused by noise.

Alternatively, even if the channel imparts no noise, information can be lost. This happens if multiple inputs map to the same output (**Fig. A.11c**). It thus becomes impossible to tell with certainty from the output Y which input was used, or equivalently, $H(\mathrm{X} \mid \mathrm{Y}) > 0$.

Finally, the most general situation, which will be encountered most in practice, is the one in which there is both a loss of information ($H(\mathrm{X} \mid \mathrm{Y}) > 0$ or equivalently $H(\mathrm{X}) > I(\mathrm{X};\mathrm{Y})$) and noise present in the output ($H(\mathrm{Y} \mid \mathrm{X}) > 0$ or $H(\mathrm{Y}) > I(\mathrm{X};\mathrm{Y})$) (**Fig. A.11d**). Although **Figure A.11d** shows a scenario where $H(\mathrm{Y}) > H(\mathrm{X})$, this can also occur even if $H(\mathrm{X}) = H(\mathrm{Y})$: some of the input entropy is replaced by noise.

To summarize: $I(\mathrm{X};\mathrm{Y})$ is the average amount of information successfully transmitted through the channel. $H(\mathrm{Y} \mid \mathrm{X})$ is the average noise at the output–uncertainty about what the output outcome will be knowing the specific input used. $H(\mathrm{X} \mid \mathrm{Y})$ is uncertainty in which input was used knowing the output outcome, which can arise from deterministic many-to-one mappings in the channel or inputs whose noise overlaps at the output.

The different ways of decomposing mutual information presented in Section A.2 have more specific interpretations in the context of a noisy channel:

$$I(\mathrm{X};\mathrm{Y}) = H(\mathrm{X}) - H(\mathrm{X} \mid \mathrm{Y})$$

Can now be interpreted as the average input information, minus the uncertainty at the output about which message was sent, which is determined by how much different messages overlap at the channels output.

$$I(\mathrm{X};\mathrm{Y}) = H(\mathrm{Y}) - H(\mathrm{Y} \mid \mathrm{X})$$

Can be interpreted as the total uncertainty at the output (the sum of input uncertainty and noise), minus the noise in the received messages.

Data processing inequality

The **Data Processing Inequality** is an important theorem which states that the information about a signal cannot be increased by any physical or computational operation. In the context of channels, this means that if we had a series of channels, each of which mapped from one random variable to another, information about the first random variable can only be preserved or lost at each step, and thus we can only be equally or more uncertain after each channel.

Mathematically, if we have $\mathrm{A} \to \mathrm{B} \to \mathrm{C}$, where each arrow represents a channel, we say that these three variables form a **Markov chain**. The theorem states that:

$$I(\mathrm{A};\mathrm{B}) \geq I(\mathrm{A};\mathrm{C})$$

and this statement can be generalizes to greater than three random variables.

Maximizing information throughput

If all channel inputs are 1) equally noisy and 2) have outputs that overlap with the outputs from other inputs equally, then a uniform distribution over the inputs will maximize the amount of information the channel transmits. There may also be other input distributions which transmit the same amount of information, but redistributing probability cannot lower noise (since all inputs are equally noisy), and any information transmission gains had by decreasing the overlap at certain outputs will be offset by increased overlap at other outputs.

In the more general case, both of these conditions will not be met, which leads to the questions: 1) What are the optimal input distribution(s) $\mathbf{p}_{\mathrm{X}}^*$ for maximizing the channel's information throughput? 2) How much information will the channel be able to transmit if this optimal input distribution is used? The answer to the latter question is called the **channel capacity** and is denoted by C.

The quantities can be found by solving an optimization problem:

$$\mathbf{p}_{\mathrm{X}}^* = \arg\max_{\mathcal{P}_{\mathcal{X}}} I(\mathrm{X};\mathrm{Y}) \tag{A.7}$$

$$C = \max_{\mathcal{P}_{\mathcal{X}}} I(\mathrm{X};\mathrm{Y}) \tag{A.8}$$

$$\tag{A.9}$$

Where $\mathcal{P}_{\mathcal{X}}$ is the set of probability distribution on the space $\mathcal{X}$. In the case where $\mathcal{X}$ is discrete, this will be the set of nonnegative vectors that whose sum is 1 (formally known as the **probability simplex**). The computational details of how to solve this optimization problem can be found in Section $A.6$. Here, we focus on the intuition behind its objective function.

Mutual information can be decomposed as $I(\mathrm{X},\mathrm{Y}) = H(\mathrm{Y}) - H(\mathrm{Y} \mid \mathrm{X})$, and this decomposition is useful in analyzing the competing goals of this optimization problem. Consider the noisy channel shown in **Figure A.12a**, which has two inputs that are noisy but map to distinct outputs, and one input that is less noisy but has outputs that overlap more with the outputs of the other inputs.

If we were to not maximize mutual information, but instead only maximize the first term in the decomposition, $H(\mathrm{Y})$, probability would be placed on inputs in such a way that the output probability was as uniformly distributed as possible. This is beneficial because it means more of the channels outputs are utilized, which means different channel inputs have more space to avoid overlapping on the same outputs. In the case of this channel, this would

result in putting half the probability mass on each of the two noisy but non-overlapping inputs, which would result in 1 bit of information gained with each use of the channel (**Fig. A.12b**).

Alternatively, if we were to maximize only the second term, $H(\mathrm{Y} \mid \mathrm{X})$ would be minimized, leading to probability being placed on in the inputs in such a way as to select for the least noisy inputs. Since the middle input is less noisy than the other two, this would result in all of the probability mass being placed on it, which would lead to no information being transmitted.

By optimizing the full objective, these goals are balanced: some probability is placed on the least noisy middle input, while some probability is placed on the noisier inputs, which makes use of the full output space. This leads to an information transmission of 1.13 bits, which is the channel capacity C.

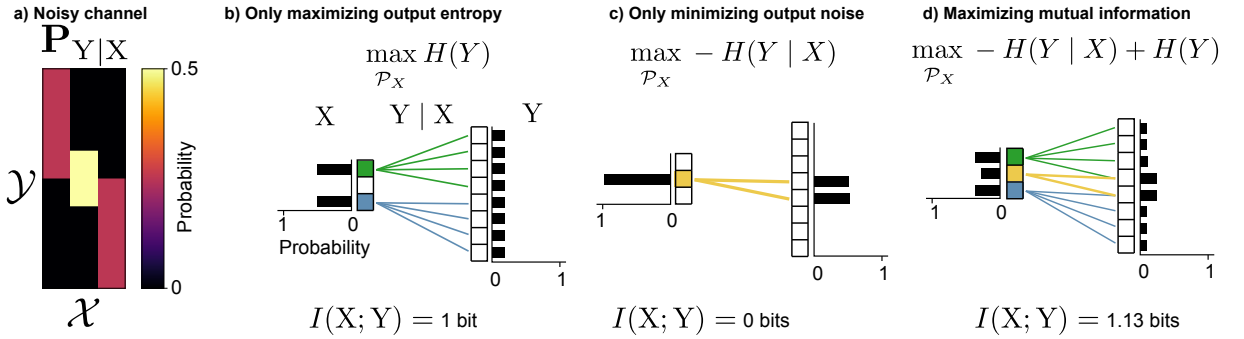

Figure A.12: **Optimizing mutual information for a fixed channel** gives both the optimal input distribution and the channel capacity. **a)** Matrix representation of a noisy channel. **b)** Only maximizing output entropy disregards putting probability on the least noisy input. **c)** Only minimizing the the average input noise fails to utilize the full output space. **d)** The full objective function balances these competing goals.

A.4 Channel coding

In information transmission problems, there will usually be a source of random events S. We will refer to the outcomes of S as **messages**. We don't have any control over the messages or the channel, but we do have the ability to design a function called an **encoder**, which will map certain messages to certain channel inputs.

In this section we discuss the problem of **channel coding**, which addresses the question of how we should design encoders to maximize information transmission, and minimize the probability that we incorrectly infer the source message.

Encoders

An **encoder** is a deterministic function that maps messages from the source to the inputs of a channel. We'll make the assumption, unless otherwise stated, that every message is mapped to a unique input, so no infomration is lost from S to X. A good encoder will maximize the mutual information between X and Y, so its goal is to map S to X in such a way that the distribution of X will balance: 1) having probability mass on non-noisy inputs, and 2) spreading probability of the channel outputs as uniformly as possible to avoid different inputs overlapping at the output.

Figure A.13a shows a noisy channel which maps each input X = x to two possible outputs with equal probability. Our goal is to transmit messages from a non-redundant source (**Fig. A.13b**), in which three colors occur with equal probability through a noisy channel, with minimal information loss.

Figure A.13c shows a bad way of doing this: using three states in $\mathcal{X}$ that have overlapping outputs in Y, which leads to a loss of information that creates ambiguity as to what the original source message was.

Alternatively, we could design an encoder which uses only a subset of the states in $\mathcal{X}$ that produce disjoint outputs in $\mathcal{Y}$ (**Fig. A.13d**). In this case, there is no loss of information.

This encoder is not unique, since we can randomly permute which state of S mapped to which state of X and achieve the same mutual information.

Compatibility of channels and sources

Suppose we are given some arbitrary source, we have multiple channels to choose from, and we can design any type of encoder we want, provided that it only encodes 1 source message at a time. What channel will allow us to transmit the most information? How much information will we be able to transmit? Will this channel also be the best for any other different source?

The answer to the last question is no. Even with freedom to design the encoder as we see fit, some sources are channels are inherently better "matched" to one another. Furthermore, there is no guarantee that for a given source and a given channel that we'll be able to achieve the channel capacity.

Intuitively, how well a source and channel are "matched" is determined by how well our best encoder maps more probable source outcomes to 1) inputs with less noisy outputs and 2) inputs whose outputs don't overlap with those of other inputs. The process of creating an encoder to perform this matching is also known as **joint source-channel coding**.

Figure A.14 shows an example of the differences in information transmission that occur when matching different source distributions to different channels. There are two noisy channels (**Fig. A.14a**): The first channel (the "symmetric channel") has inputs that are all equally noisy and all have outputs that overlap equally. The second channel (the "asymmetric channel") also has equally noisy inputs, but their outputs are not equally overlapping with

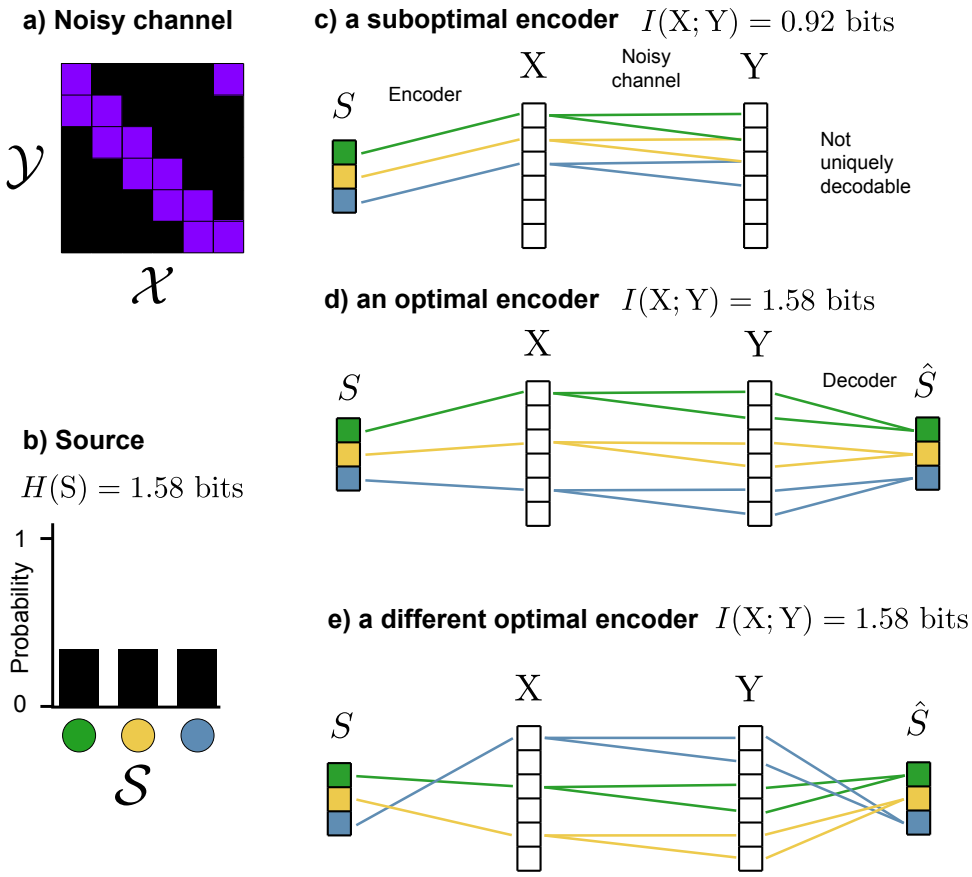

Figure A.13: **Optimal encoders for a noisy channel**. **a)** a noisy channel that maps each input to two possible outputs **b)** A maximum entropy source of random messages. **c)** A encoder that maps messages in $\mathcal{S}$ to inputs in $\mathcal{X}$ that produce overlapping outputs that are not uniquely decodable, thus transmitting less information than the maximum possible. **d)** An encoder that transmits the maximum amount of information by mapping messages to inputs that produce disjoint, and thus decodable, outputs. **e)** Another encoder/decoder pair that transmits the maximum amount of information, showing that the optimal encoder is not always unique.

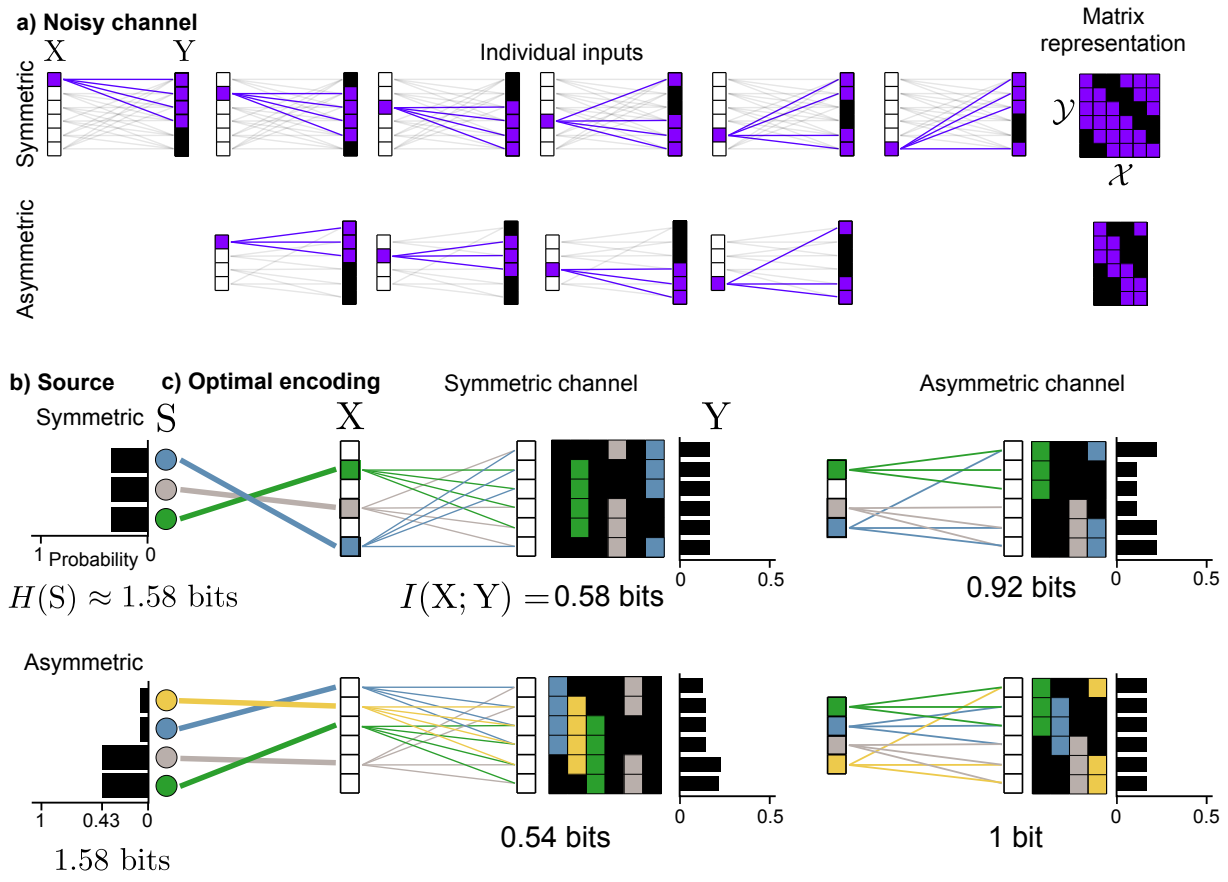

Figure A.14: **Matching sources and channels for maximum information transmission. a)** A symmetric noisy channel, in which all inputs are equally noisy and equally overlapping (for a uniform input distribution) and an asymmetric noisy channel, in which all inputs are equally noisy, but two pairs of inputs overlap more at the output with each other than the other pair (for a uniform input distribution). **b)** Symmetric and asymmetric sources with the same entropy. **c)** The symmetric source can be encoded to transmit more information in the symmetric channel, because it is able to choose inputs such that overlap equally at the output, whereas the asymmetric source is not. For the asymmetric channel, this is reversed, and the more probable messages from the asymmetric source can be encoded such that they are less overlapping at the output.

one another: it has two pairs of inputs whose outputs overlap less with each other than do the outputs within each pair.

There are also two different sources with the same entropy (**Fig. A.14b**). The first source (the "symmetric source") has three equally probably outcomes, while the second source (the "asymmetric source") has two more probable and two less probable outcomes. For channels and sources with a small number of states such as the ones shown here, the optimal encoder can be found by exhaustively searching all possible ways of mapping messages to channel inputs.

For the symmetric channel, the optimal encoder for the symmetric source is able to transmit more information than the optimal encoder for the asymmetric source, but for the asymmetric channel, the reverse is true (**Fig. A.14c**). In the latter case, this occurs because the encoder is able to map each of source's two most highly probable messages (`green` and `gray`) to distinct subsets of channel inputs that overlap less at the output. This is not possible for the symmetric source, because all of its messages are equally probable.

In contrast, for the asymmetric channel, more information can be transmitted about the asymmetric source than the symmetric source. This is because the outputs for all inputs are equally overlapping, which forces the asymmetric source to map highly probable messages channel outputs with more overlap.

Maximal information transmission Having seen that the channel capacity C can be measured by finding the optimal input distribution, and that encoders for different sources that map one message at a time to channel inputs can achieve different maximal amounts of information transmission, it can be concluded that the amount of information transmitted is less than the channel capacity under these circumstances. However, a central result in information theory, the **Noisy channel coding theorem**, proves that it is in fact possible to transmit information up to the channel capacity for *any* source/channel pair. The key to making this possible is not encoding one message at a time, but rather many messages at once. This both changes the distribution of the source messages and the noise per each channel input so that they are more uniform and can be more easily matched by an encoder.

The noisy channel coding theorem

Noisy channel coding is one of the central problems in information theory. The **noisy channel coding theorem** is the foundation of the digital world we live in today, because it defined the limits of reliable transmission and storage of information using inherently noisy physical components.

The general setup is: There is a source of random messages, which could be human-chosen or the result of some natural random process. The goal is to transmit these messages to a receiver with minimal or no error, but to do so they must pass through a **noisy channel**.

The noisy channel coding theorem shows that any source/channel pair is able to transmit information at a **rate** (i.e. information per channel use) up to the channel capacity and in doing so, enable a decoder to infer the true message with arbitrarily small probability of

error. It also shows that above the channel capacity, achieving arbitrarily small probability of error is impossible without driving the rate to 0.

The analysis and proof of the noisy channel coding theorem break apart noisy channel coding into separate source coding and channel coding steps. This yields five steps: (**Fig. A.15**): 1) A compressor takes in the random events, and removes any redundancy, making it a maximum entropy source. 2) The compressed events pass into an encoder, which adds redundancy to try to prevent the loss of information when 3) passing through a noisy channel. 4) The channel output is then fed into a decoder, which attempts to remove any errors introduced by the noisy channel and recover the original message. 5) The compressed messages are decompressed back onto the original probability space.

Steps 2-4 correspond to the random variables:

- S: The (compressed) source message
- X: The input to the channel (the encoded message)
- Y: The output of the channel (the received message)
- $\hat{S}$: The estimate of the (compressed) message

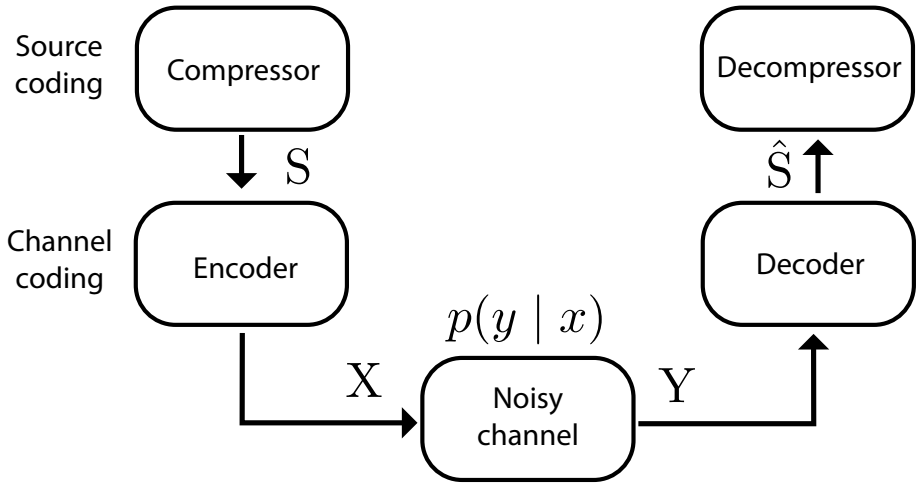

Figure A.15: **Noisy channel coding: the big picture** A redundant source passes into a compressor, which yields a compressed (i.e. maximum entropy) random message S. S then passes into an encoder, which adds redundancy to create robustness to passage over a noisy channel. The encoded message X passes through the channel, yielding the received message Y, which possibly contains errors. Y then goes into a decoder, yielding an estimate of S, $\hat{S}$, and finally a decompressor to give an estimate of the source. Adapted from [89] p146.

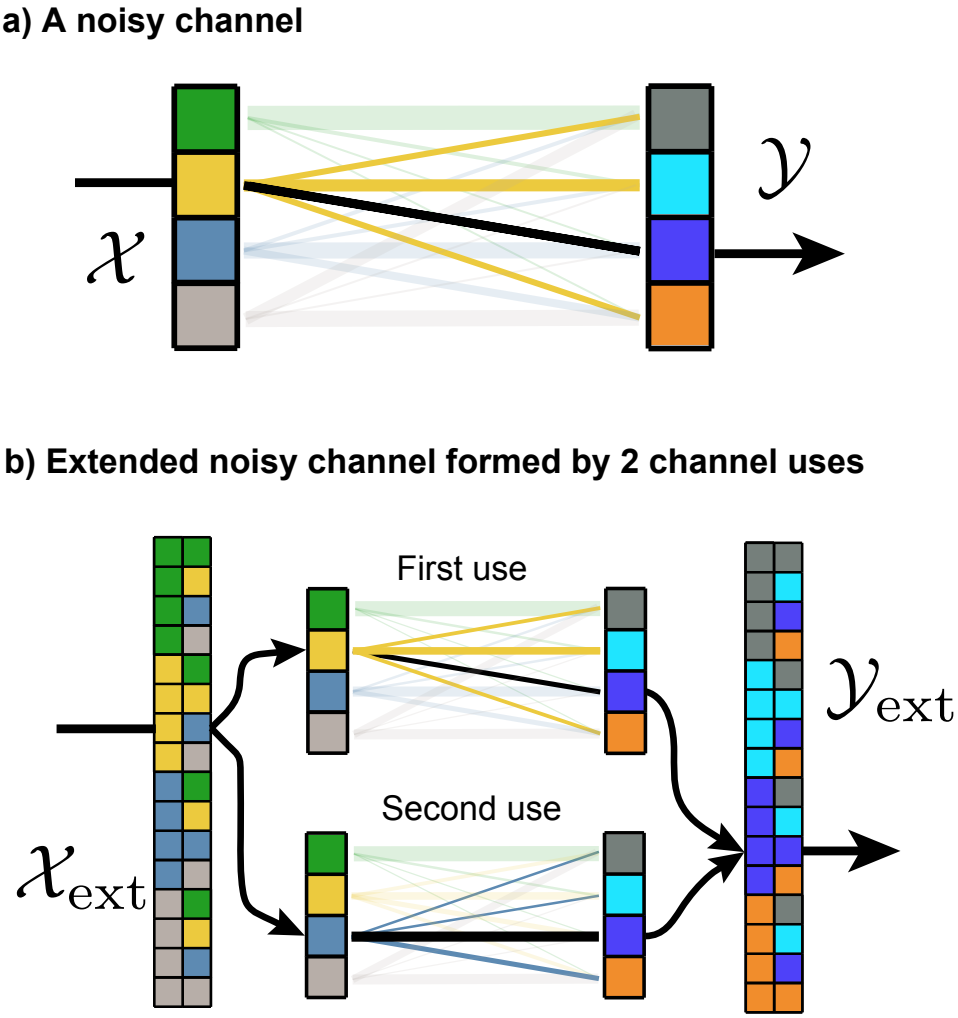

Figure A.16: **An extended noisy channel** is formed by treating multiple uses of a single channel as a channel itself. **a)** One use of the original channel. **b)** one use of the extended channel.

Ignoring compression with block encoders

The result of the noisy channel coding theorem is achieved by using an encoder that doesn't encode 1 message at a time to channel inputs, but rather one that encodes multiple messages at a time. The number of messages is called **block length**. In practice, an infinitely long block length cannot be used, because the message would never actually transmit. As a result, a noisy channel might not be able to transmit information up to its full capacity for certain sources. There may also be possible performance gains in practice by using joint source-channel coding, as will be discussed in Section A.4.

Compression and encoding (and decompression and decoding) are separated into distinct steps not because they must be carried out separately, but to allow for easier analysis of the problem. The **source-channel separation theorem** states that there is no cost to this division–compressors and encoders can be designed just as well separately as they can jointly (in the limit of an infinitely long block length).

Without loss of generality, we can treat S as a maximum entropy source and ignore the compression/decompression step. This is because if we have a redundant source, it is possible when using a long block length to perform lossless compression, which yields a maximum entropy source over a much larger state space.

The larger state space of the source means that we'll also need a channel with a larger state space, which we can form by using an **extended noisy channel**. This is simply using the channel N times, and considering a meta-channel whose inputs and outputs are all possible combinations of of the original channel's inputs and outputs (**Fig. A.16**).

As a result of this separation and the theorem justifying it, we can disregard the source coding problem (i.e. the compressor and decompressor) and focus only on the channel coding problem (**encoder** $\rightarrow$ **noisy channel** $\rightarrow$ **decoder**). Since any source of random events can be compressed into a maximum entropy source of (Sec. A.2), we can simply assume this has already taken place.

Theoretical limits of perfect transmission

The **noisy channel coding theorem** was a revolutionary step forward because it proved something that was previously thought to be impossible: that the probability of error when transmitting a message over a noisy channel and trying to infer its value on the other side could be made arbitrarily small without the rate of information transfer dropping to zero.

To understand this, we start with an **encoder** $\rightarrow$ **noisy channel** $\rightarrow$ **decoder**. As shown in **Figure A.17a**, the messages we are trying to send are the outcomes of a series of maximum entropy/non-redundant random events (like colored marbles drawn from an urn with equal probability). The noisy channel transmits binary data, so we will then convert the outcome of each random event to a binary string using an optimal encoder that minimizes the number of bits used. Using binary channels simplifies the analysis, but is not strictly needed. Noisy channels can be over any probability space, continuous or discrete.

These binary strings will be transmitted over a noisy channel, which will, with some probability, flip `0`s to `1`s and vice versa. If we sent the raw binary messages over this channel, errors will occur and the message on the other side will be interpreted incorrectly as a result of noise displacing and thus destroying some of the usable information in each string. To combat this, before transmitting over the noisy channel, we will add some type of redundancy so that the messages are more robust to the channel's noise. Adding this redundancy means that the binary strings transmitted will become longer, but the information that contain will be unchanged. Thus, the number of (information) bits will be smaller than the number transmitted (data) bits.

The **rate** R at which we transmit data depends on the amount of redundancy added:

$$R = \frac{\#\text{ source bits}}{\#\text{ transmitted bits}}$$

(**Note:** This rate is not exactly the same as the rate used in rate-distortion theory (Sec. A.2). Rate as used here is defined as the compression of a transmitted message, whereas for rate-distortion theory it describes the absolute amount of information about particular source.)

Next, the redundant message passes through a noisy channel. The channel shown in **Figure A.17a** shows a **binary symmetric channel** in which `0`s and `1`s transmit correctly with probability $\frac{4}{5}$ and flip with probability $\frac{1}{5}$.

Finally, a decoder interprets the received message and attempts to correct any errors and reconstruct the original message.

A naive approach to this problem is through repetition coding (**Fig. A.17b**). Here, the message is repeated N times before being sent through the channel, and the decoder chooses the bit at each position that occurred the greatest number of times in the repetition-coded message. The wrong message can be received if more than half of the bits in a given position flip. To lower the probability of this happening, we can increase the number of repetitions. This reduces the probability of error, but in doing so lengthens the message needed for each event and thus lowers the rate at which information can be transmitted. As we specify a lower and lower maximal acceptable probability of message error, the rate at which information is sent approaches 0 (**Fig. A.17c**).

The noisy channel coding theorem proved that we can do better than this. To do so, we must send multiple messages at once, also known as **block coding**. In the case of this binary channel, this is done by taking multiple binary messages, and passing them through an encoder that maps to a single binary string. The encoder in **Figure A.17d** is called a Hamming code, which is a good choice for this problem, but not specifically needed for the noisy channel coding theorem. Similar to the previous case, those bits are then transmitted and the decoder attempts to correct errors and reconstruct the original message.

As the bottom panel of **Figure A.17d** shows, by transmitting a sequence of random events as a single outcome, we can increase the block length of our transmission system

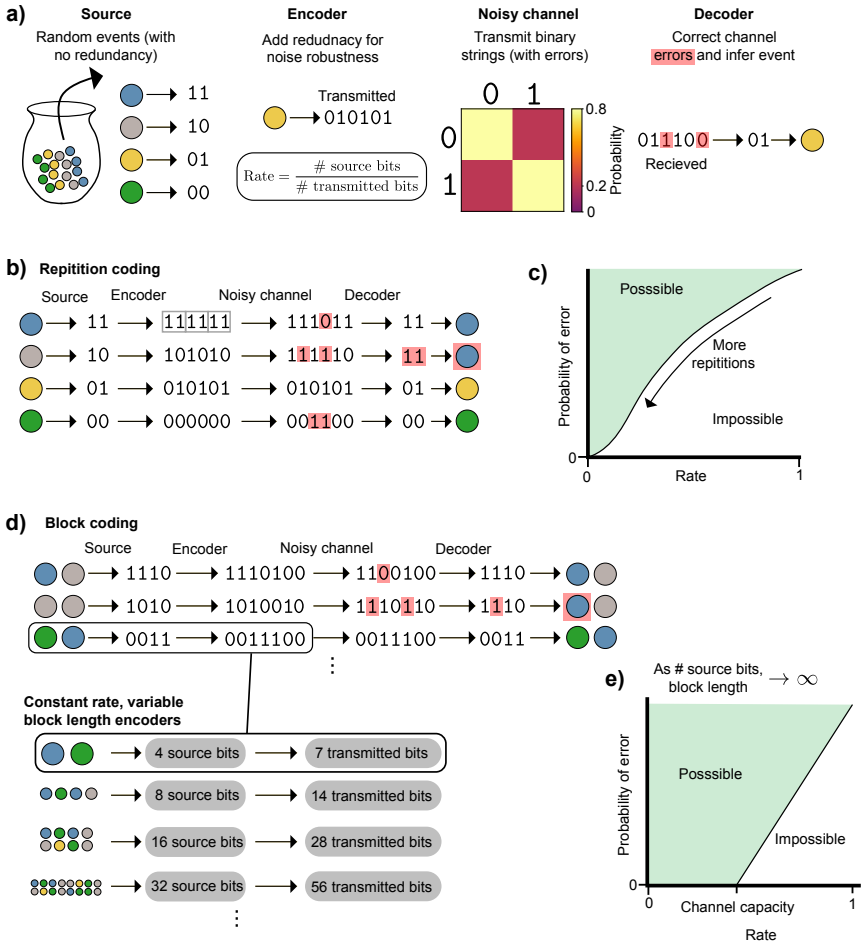

Figure A.17: **The noisy channel coding theorem a)** An overview of the problem setup. A source produces non-redundant (i.e. maximum entropy) messages, here a colored marble which is transformed into a two-digit binary encoding. An encoder adds redundancy to prepare a transmitted message (e.g. shown here, repeating the message 3 times). The rate is defined as the ratio of source bits to transmitted bits. The message passes through a noisy channel (e.g. shown here one that flips each bit with probability $\frac{1}{5}$). A decoder attempts to correct errors and recover the original message. **b)** A repetition coding scheme (same as part a) repeats the source bits a fixed number of times and **c)** can achieve arbitrarily small probability of error (for the noisy channel in (a)), but needs to send information at a rate approaching 0 to do so. **d)** A block coding scheme can do better, by encoding multiple events together. **e)** As the block length goes to ∞, such a scheme can achieve communication with arbitrarily small probability of error at rates up to the channel capacity. At rates above the channel capacity, infinite block lengths will have nonzero probability of error.

while leaving the rate unchanged by keeping the ratio of source bits and transmitted bits constant.

The noisy channel coding theorem considers the performance of channels as the block length goes to infinity, and it has two important implications (**Fig. A.17e**):

1. Encoder/decoder pairs exist which can transmit messages with arbitrarily small probability of error at nonzero rates up to the **channel capacity**

2. At rates above the channel capacity, no encoder/decoder pairs can transmit messages with arbitrarily small error probability

The theorem does not state how one can find the appropriate encoders/decoders to achieve this performance, only that they exist. Specifically, it shows that the *average* performance of randomly constructed encoders achieves this, which implies that at least one of the individual randomly constructed encoders can achieve it. Finding good performing codes that can also be decoded is not straightforward based on this theorem alone, in part because extending the block length makes naive decoding algorithms exponential more complex. It took several decades of research to discover codes for binary channels that approach the performance the theorem implies.

The benefits of long block length

The noisy channel coding theorem shows that information can be transmitted at a rate up to the channel capacity in the asymptotic setting of infinitely long block lengths. This leads to the question, what is it about long block lengths that makes this achievable, even for channel/source pairs (like the ones in **Figure A.14**) that cannot reach the channel capacity in the short block length setting?

The answer is that long block lengths make both channels and sources more uniform in such a way that the problem of finding an encoder that optimally maps source messages to channel inputs becomes trivial: it can accomplished by simply picking a fixed random mapping. As discussed in Section A.4, a good encoder will map the most probable source messages to channel inputs that are the least noisy (minimize $H(\mathrm{Y} \mid \mathrm{X})$) as well as choosing inputs that minimize the overlap of messages at the channel output (minimize $H(\mathrm{Y} \mid \mathrm{X})$).

By extending a source to a long block length (i.e. encoding a sequence of messages rather than individual messages), all sources will start to resemble maximum entropy sources, which have uniform probability over all messages. This happens either because: 1) individual messages are already coming from a maximum entropy source, and this will remain true at any block length. 2) Redundant sources will produce equally probable typical sequences (discussed in Section A.2) onto which nearly all of the probability mass will concentrate, and non-typical sequences can be ignored because they contain vanishingly small probability. Thus, a redundant source, when extended, behaves like a maximum entropy source over a smaller number of possible messages (the typical set).

A similarly uniformity arises when a noisy channel is extended: both the noisiness of each channel input and the overlap of its outputs with other channel inputs (assuming a uniform input distribution) become the same for all inputs. The noisiness of a channel input x is measured by its point-wise conditional entropy $H(\mathrm{Y} \mid x)$. A noiseless input would have a value of 0, and noisy inputs have positive values. By looking at the distribution of point-wise conditional entropies for each channel input, we can assess how heterogeneous the channel inputs are in terms of their noisiness (**Fig. A.18**). The absolute noise level will always increase as we make extended noisy channels with longer block lengths, but the relative amount of noise in each input becomes increasingly close to the block length N times the average noisiness of the non-extended channel, $H(\mathrm{Y} \mid \mathrm{X})$. This is a consequence of the Central Limit Theorem, as shown in Section A.7.

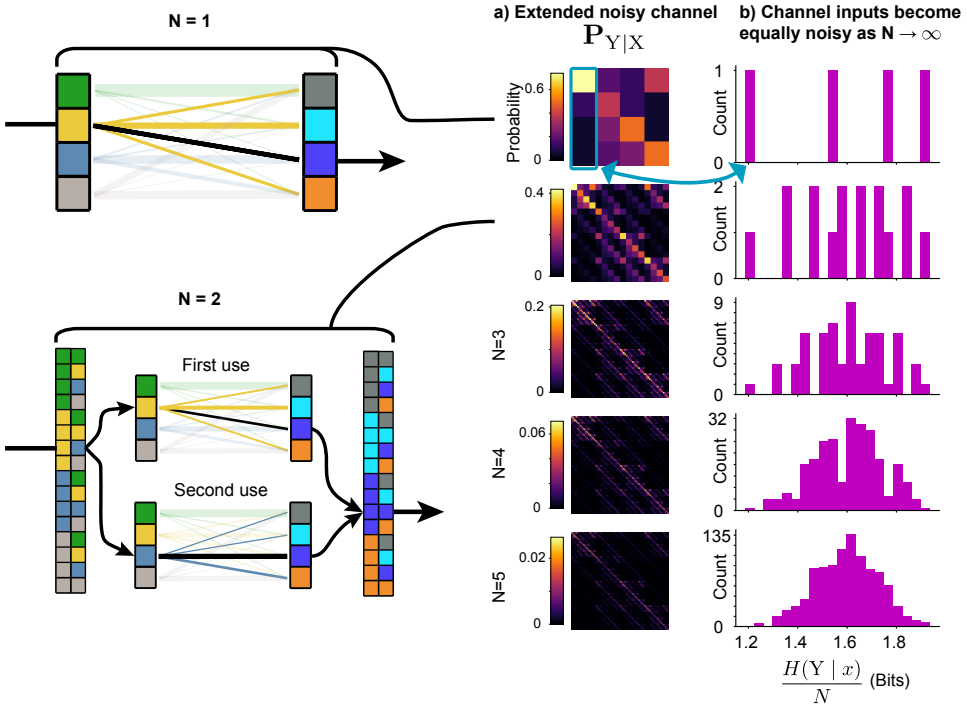

Figure A.18: **Conditional entropy in an extended noisy channel. a)** Matrix representation of extended noisy channels with increasing block lengths. **b)** Histogram of conditional entropy for each channel input. With increasing block lengths, channel inputs tend towards having the same conditional entropy, which means they are equally noisy.

In addition to this uniformity in the noise level of different channel inputs, the overlap of different channel inputs becomes uniform as block length increases. Mathematically, this means that $H(x \mid \mathrm{Y})$ tends toward the same value for all choices of x. Like the increasing

uniformity of noise for each channel input in the previous paragraph, this phenomenon is a consequence of the Law of Large Numbers and can be shown mathematically with a proof similar to that in Section A.7. The only difference is that unlike the noise of different inputs, the overlap of different inputs requires knowledge of the joint distribution $p_{\mathrm{X,Y}}$. It thus depends on a particular choice of input distribution, since $p_{\mathrm{X,Y}} = p_{\mathrm{Y|X}} p_{\mathrm{X}}$. However, since we know that all sources will produce equally probable messages as $N \to \infty$, we can simply assume that p_{X} is uniform. This means that the matrix $\mathbf{P}_{\mathrm{X,Y}}$ is equal to $\mathbf{P}_{\mathrm{Y|X}}$ multiplied by a scalar constant.

It is worth noting that extended noisy channels still have some inputs that are especially noisy and especially noiseless, as well as those the are more or less overlapping. For example, the extended noisy channel input that corresponds to using the most noisy input of the non-extended channel for each transmission still may be much noisier than the average extended channel input. Outliers like this don't go away as $N \to \infty$, they simply cease to matter because they are outnumbered. However, since we can never actually have an infinite block length in practice, in many situations information transmission can do better than a fixed, random encoder, by shifting more probable messages onto less noisy, less-overlapping inputs. This is possible because when N is not infinite, any redundant source will not produce equally probable messages. How to do this in practice will be explored in the next section.

Noisy channel coding in practice

In Section A.4 we demonstrated that for a given source/channel pair, designing encoders that encode one message at a time may not always be able to transmit information at a rate up to the channel capacity. In Section A.4, we described how the Noisy Channel Coding Theorem proves that by extending the block length to infinity, there will always be an encoder that can transmit information at a rate up to the channel capacity. In this section, we turn to the more practical intermediate setting, where we want to design an encoder for maximal information transmission and have some, but not an infinite ability to extend the block length.

Longer, but not infinite, block length

The first important question is, how do the asymptotic changes that improve our ability to match channels and sources manifest as N increases? These changes, the transformation of a source into an extended source with uniform probability over its messages and a channel into an extended noisy channel with equally noisy inputs that overlap equally at the output (for a uniform input distribution), are not only possible as $N \to \infty$. Some source/channel pairs, like the one shown in **Figure A.13**, already have this property with a block length of 1. However, this represents a special case that will usually not be true.

Generally, a redundant source, when extended, will produce messages that are more uniform in probability as N increases (as shown in **Figure A.3**). Similarly, the inputs of a

noisy channel will become closer to each other in their noise level and overlap at the output (for equally probably inputs) as N increases.

There will initially be large performance gains for increases in block length beyond $N = 1$ that rapidly diminish as block length continues to increase. Different theoretical results provide performance bounds that demonstrate this type of performance. For example, it has been proven that there are block codes in which the probability of decoding error is bounded by an exponentially decreasing function ([89] p171). Similarly, for a fixed probability of error, the minimum discrepancy between the channel capacity and the actual amount of information transmitted decreases proportional to $\frac{1}{\sqrt{N}}$ [122].

Optimizing encoders

Since sources will not produce equally probable messages and not all inputs of a channel will be equivalent, in many cases we will be able to increase information throughput by designing encoders that map more probable messages to less noisy inputs that overlap less with the outputs from other inputs [122]. This is also known as **joint source-channel coding**, to differentiate it from the setting where an encoder is designed based on the assumption that it will be encoding a maximum entropy source.

We can do this by solving the following optimization problem:

$$\arg\max_{f} I(f(\mathrm{S}); \mathrm{Y}) \tag{A.10}$$

$$\tag{A.11}$$

Where $f(\cdot)$ is the encoder function that maps from the space of possible messages $\mathcal{S}$ to the the space of channel inputs $\mathcal{X}$.

When $\mathcal{S}$ and $\mathcal{X}$ are discrete spaces, this can be a challenging optimization problem, because it is not amenable to gradient-based optimization. If we consider this problem in matrix form, $\mathbf{p}_\mathrm{S}$ is a vector of probabilities of different messages, and $\mathbf{P}_\mathrm{f}$ is noiseless channel that maps $\mathbf{p}_\mathrm{S}$ to $\mathbf{p}_\mathrm{X}$. Since the channel is noiseless, each column has exactly one entry with probability 1, and all other entries are 0, like the channels shown in **Figure A.11a,c**.

A naive approach to solving this optimization problem is to use a brute force search, in which the mutual information is calculated for every possible encoder matrix. However, this quickly becomes computationally intractable, since the number of possible encoder matrices grows with with the factorial of the number of channel inputs.

Another possibility is to instead use **stochastic encoders**, which consist of replacing the deterministic encoder function $f(\cdot)$ with a conditional probability distribution $p(x \mid y)$. In the matrix view, this corresponds to matrices which can have an arbitrary number of nonzero elements along each column, so long as they are all positive and each column sums to 1, like the matrices in **Figure A.11b,d**.

Since every deterministic encoder represents a specific case of a larger class of stochastic encoders, stochastic encoders should be able to do as good or better, in theory. In practice

that often do worse, though there are certain conditions under which they're superior [153, 165]. This remains an area of active research.

The major advantage that stochastic encoders do have is that they are much more amenable to optimization. Since each column of a stochastic encoder matrix (or any channel) is a conditional probability distribution it can be optimized with respect individual probabilities using the same numerical optimization tools used to find the optimal input distribution for a fixed channel (Section A.6).

Continuous encoders It is also possible to optimize encoders in the case of continuous sources/channels. That is, a source is represented by a probability density function, and encoder is a continuous function (if it is deterministic) or a conditional probability density function (if it is stochastic). In this setting, the entropies become differential entropies, which have a less clear intuitive interpretation and can in some cases take on infinite values (Section A.2). However, the gradients of these entropies are well defined and do not suffer from such mathematical difficulties, making gradient-based optimization possible [6]. Still, there remain many difficulties to optimizing mutual information in both continuous and discrete settings, particularly in high-dimensional settings, and this remains an active area of research [123].

Variable channels and the cliff effect

Throughout the discussion of channels and encoders, we've made the implicit assumption that the channel's noise characteristics are known and fixed, and that any encoder we've designed will be deployed on source channel combinations exactly like those for which it was designed. In practice, this may not always be true, and we may encounter a phenomenon known as the **cliff effect**, which is a rapid degradation in performance when the noise characteristics of the channel deviate from the ones for which the encoder was designed [122]. The results of the noisy channel coding theorem do not provide any guidance about how to design encoders that are robust to this type of situation, and this too remains an active area of research.

Decoders

Without a proper encoder, we may incur an unnecessary loss of information in the noisy channel. However, even with a an optimal encoder, there remains the challenge of inferring the original message S from the channel output Y. This is the job of the **decoder**, a function or conditional probability distribution which forms an estimate of the original source message $\hat{\mathrm{S}}$. Encoders and decoders must be designed together, and the noise characteristics of the channel must be considered to achieve good performance.

The channel output will contain both source information $I(\mathrm{X};\mathrm{Y})$ and noise added by the channel $H(\mathrm{Y} \mid \mathrm{X})$. A optimal decoder will get rid of the latter, mapping all noisy versions of a particular message $p(y \mid x)$ back to a correct estimate of the original message $s = \hat{s}$. In

practice, our decoder may not be able to eliminate all of the noise $H(\mathrm{Y} \mid \mathrm{S})$ or preserve all the signal $I(\hat{\mathrm{S}}; \mathrm{S})$: There may be a trade-off between these competing goals.

Preserving the information contained in the received message Y is necessary, but not sufficient, for estimating the correct message. To understand why it is necessary, consider the rate distortion curve in **Figure A.9c**. The probability that the estimated source message is not equal to the actual source message is a valid distortion function, and the rate-distortion curve tells us that to improve the best possible average performance on this distortion function, we will need more information. To understand what it is not sufficient, consider that if we had a perfect decoder, we could always permute all of its predictions such that it would be wrong every time. This would give the worst possible distortion, but all the information would remain, because we could always perform the inverse permutation.

As discussed in the previous section, using block encoders and an extended noisy channel can increase the information throughput of a channel (up to the channel capacity). However, this strategy comes at the cost of increasing the complexity of designing a corresponding decoder. Specifically, as we increase the block length, the number of different messages we have to decode increases exponentially: $|\mathcal{Y}|^N$. A naive decoder would simply be a lookup table that maps a received message Y to an estimate of the source event $\hat{\mathrm{S}}$, and this lookup table becomes exponentially large with increasing block length. The best encoder-decoder pairs are thus semi-random, giving them the advantages of random encoder construction with the potential for cleverly designed decoding algorithms that can do better than the exponential complexity of the naive approach. There is not a known formula for designing such encoder-decoder pairs that works across different types of channels.

A.5 Code availability

The code used to produce the figures and hi res versions of the figures can be found at `https://doi.org/10.5281/zenodo.6647779`

A.6 Numerical optimization of channel input distribution

Computing the optimal input distribution for a noisy channel can be used not only to design encoders that provide maximal information throughput, but also to figure out what the channel's capacity is. In some cases we may be able to use analytical properties of the channel (e.g. Gaussian noise with zero mean known variance) to perform this computation. In other cases, we can find this distribution by maximizing the channel's mutual information with respect to the input probabilities ([89] p169). Mathematically:

$$\mathbf{p}_{\mathrm{X}}^{*} = \arg\max_{\mathcal{P}_{\mathcal{X}}} I(\mathrm{X};\mathrm{Y}) \tag{A.12}$$

$$C = \max_{\mathcal{P}_{\mathcal{X}}} I(\mathrm{X};\mathrm{Y}) \tag{A.13}$$

$$\tag{A.14}$$

This optimization problem can be solved, and the optimal distribution/channel capacity computed, using numerical optimization techniques such as gradient ascent. Mutual information has the nice property that it is a convex function with respect to the input probabilities, so optimizing the input probabilities directly using gradient ascent will be guaranteed to find a global maximum eventually with a proper learning rate.

We solve this problem using projected ascent. which consists of applying the following updates:

$$\mathbf{p}_{\mathrm{X}}^{k+1} = \mathrm{proj}(\mathbf{p}_{\mathrm{X}}^{k} + \lambda \nabla_{\mathbf{p}_{\mathrm{X}}} I(\mathrm{X},\mathrm{Y}))$$

Where $\mathbf{p}_{\mathrm{X}}$ is a vector of probabilities for each state in $\mathcal{X}$.

The projection operator $\mathrm{proj}(\cdot)$ enforces the constraints that the probabilities need to be positive and sum to 1. This is done by taking the element-wise maximum of $\mathbf{p}_{\mathrm{X}}$ and $\mathbf{0}$ (a vector of all 0s), and then adding enough to each element such that they sum to one. Mathematically:

$$\mathbf{p}_{\mathrm{X}} = \max(\mathbf{p}_{\mathrm{X}}, \mathbf{0})$$
$$\mathbf{p}_{\mathrm{X}} = \mathbf{p}_{\mathrm{X}} + \frac{1 - \sum_{i=1}^{N} p_i}{N}$$

Where N is the number of elements in $\mathbf{p}_{\mathrm{X}}$ and p_i is the ith entry of $\mathbf{p}_{\mathrm{X}}$.

While this heuristic projection step appears to work in practice, we note that there may be better and more theoretically motivated alternatives [167].

A.7 Proofs

Proof that joint entropy of two random variables equals sum of individual entropies

$$
\begin{aligned}
H(\mathrm{X},\mathrm{Y}) &= -\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(x,y)\log\frac{1}{p(x,y)} \\
&= -\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(x)p(y)\log\frac{1}{p(x)p(y)} \\
&= -\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(x)p(y)\left(\log\frac{1}{p(x)}+\log\frac{1}{p(y)}\right) \\
&= -\left(\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(x)p(y)\log\frac{1}{p(x)}\right)-\left(\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(x)p(y)\log\frac{1}{p(y)}\right) \\
&= -\left(\sum_{x\in\mathcal{X}} p(x)\log\frac{1}{p(x)}\sum_{y\in\mathcal{Y}} p(y)\right)-\left(\sum_{y\in\mathcal{Y}} p(y)\log\frac{1}{p(y)}\sum_{x\in\mathcal{X}} p(x)\right)
\end{aligned}
$$

Taking advantage of the fact that the sum over any probability distribution is one, this reduces to:

$$
\begin{aligned}
&= -\left(\sum_{x\in\mathcal{X}} p(x)\log\frac{1}{p(x)}\right)-\left(\sum_{y\in\mathcal{Y}} p(y)\log\frac{1}{p(y)}\right) \\
&= H(\mathrm{X})+H(\mathrm{Y})
\end{aligned}
$$

Decomposition of mutual information

$$
\begin{aligned}
I(\mathrm{X};\mathrm{Y}) &= \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(x,y)\log\left(\frac{p(x,y)}{p(x)p(y)}\right) \\
&= \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(x,y)\log\left(\frac{p(y\mid x)}{p(y)}\right) \\
&= \sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(x,y)(-\log p(y)+\log p(y\mid x)) \\
&= -\left(\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(x,y)\log p(y)\right)+\left(\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(x,y)\log p(y\mid x)\right) \\
&= -\left(\sum_{y\in\mathcal{Y}}\left(\sum_{x\in\mathcal{X}} p(x,y)\right)\log p(y)\right)+\left(\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(x,y)\log p(y\mid x)\right) \\
&= -\left(\sum_{y\in\mathcal{Y}} p(y)\log p(y)\right)+\left(\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(x,y)\log p(y\mid x)\right) \\
&= H(\mathrm{Y})-H(\mathrm{Y}\mid\mathrm{X})
\end{aligned}
$$

Proof that block length-normalized extended noisy channel inputs become equally noisy fo infinte block length

Since each channel use within the extended noisy channel is independent, the point-wise conditional entropies will add. For a sequence of channel uses where $x^{(k)}$ represent the input used on the kth transmission of the non-extended channel:

$$
\begin{aligned}
H(\mathrm{Y}\mid\mathbf{x}) &= && \text{(A.15)}\\
&= H(\mathrm{Y}\mid x^{(1)},x^{(2)},...,x^{(N)}) && \text{(A.16)}\\
&= H(\mathrm{Y}\mid x^{(1)})+H(\mathrm{Y}\mid x^{(2)})+...+H(\mathrm{Y}\mid x^{(N)}) && \text{(A.17)}
\end{aligned}
$$

Where $\mathbf{x}=x^{(1)},x^{(2)},...$ represents the input used on each channel use. These too are independent random variables, since the input chosen on one channel use is independent of subsequent channel uses. Thus, we can take a probability-weighted average over the choice of channel input, yielding:

$$\sum_{\mathbf{x}\in\mathcal{X}^{\mathbf{N}}} p(\mathbf{x})H(\mathrm{Y}\mid\mathbf{x}) = \left(\sum_{\mathbf{x}\in\mathcal{X}^{\mathbf{N}}} p(\mathbf{x})H(\mathrm{Y}\mid x^{(1)})\right) + \left(\sum_{\mathbf{x}\in\mathcal{X}^{\mathbf{N}}} p(\mathbf{x})H(\mathrm{Y}\mid x^{(2)})\right) + ...$$
$$= \left(\sum_{x^{(1)}\in\mathcal{X}} p(x^{(1)})H(\mathrm{Y}\mid x^{(1)}) \left[\prod_{i=2}^{N}\sum_{x^{(i)}\in\mathcal{X}} p(x^{(i)})\right]\right) +$$
$$\left(\sum_{x^{(2)}\in\mathcal{X}} p(x^{(2)})H(\mathrm{Y}\mid x^{(2)}) \left[\sum_{x^{(1)}\in\mathcal{X}} p(x^{(1)})\prod_{i=3}^{N}\sum_{x^{(i)}\in\mathcal{X}} p(x^{(i)})\right]\right) ...$$

By using the fact the sum of a probability distribution is equal to one, this reduces to:

$$\left(\sum_{x^{(1)}\in\mathcal{X}} p(x^{(1)})H(\mathrm{Y}\mid x^{(1)})\right) + \left(\sum_{x^{(2)}\in\mathcal{X}} p(x^{(2)})H(\mathrm{Y}\mid x^{(1)})\right) + ...$$
$$= H(\mathrm{Y}\mid \mathrm{X}^{(1)}) + H(\mathrm{Y}\mid \mathrm{X}^{(2)}) + ...$$
$$= \sum_{k=1}^{N} H(\mathrm{Y}\mid \mathrm{X})$$

By dividing by N and invoking the Law of Large Numbers, we can see that as $N \to \infty$, the block length-normalized noise level of an average extended channel input will become increasingly close to average noise of the non-extended channel