A computational tour on random matrices : imaging through complex media and optical computing

Jonathan Dong

PSL
UNIVERSITÉ PARIS

THÈSE DE DOCTORAT
DE L'UNIVERSITÉ PSL

Préparée à École Normale Supérieure
(Laboratoire de Physique de l'École Normale Supérieure
et Laboratoire Kastler-Brossel)

A computational tour on random matrices: imaging through complex media and optical computing

Soutenue par
Jonathan DONG
Le 17 novembre 2020

École doctorale n°564
EDPIF

Spécialité
Physique

Composition du jury :

Prof. Marc MÉZARD Ecole Normale Supérieure	*Président*
Prof. Romain COUILLET GIPSA-lab, Université Grenoble-Alpes	*Rapporteur*
Prof. Vivek GOYAL Boston University, USA	*Rapporteur*
Prof. Laura WALLER UC Berkeley, USA	*Membre*
Prof. Lenka ZDEBOROVA IPhT, Paris-Saclay	*Membre*
Prof. Ulugbek KAMILOV Washington University, USA	*Membre*
Prof. Sylvain GIGAN Laboratoire Kastler Brossel	*Directeur de thèse*
Prof. Florent KRZAKALA Laboratoire de Physique de l'ENS	*Directeur de thèse*

ENS | PSL
Laboratoire Kastler Brossel — Physique quantique et applications
LPENS — LABORATOIRE DE PHYSIQUE DE L'ÉCOLE NORMALE SUPÉRIEURE

Abstract

The topic of this thesis is the development of large-scale optimization algorithms, popularized by machine learning, for various applications around light propagation through complex media. Machine learning, already inevitable in our life, is a remarkably successful approach to retrieve meaningful patterns from very large datasets. Recommendation systems, speech recognition, and computer vision are all based on this powerful paradigm. However, there is still a long way to go before we reach a good theoretical understanding of how these algorithms really work. This calls for rigorous and well-defined studies of the non-linear optimization problems underneath machine learning.

In this thesis, we study these questions through the prism of optical scattering, a physical phenomenon describing the propagation of light in complex materials. This unconventional point of view sheds a new light on the optimization algorithms that arise for two different purposes: imaging and optical computing.

Imaging an object through a scattering medium is often a two-step process requiring first a characterization of the system, followed by the proper image reconstruction. It will be the opportunity to apply the most recent advances in Phase Retrieval, arguably the most important non-linear equation to solve in computational imaging, and introduce low-rank matrix factorization as another very valuable tool to analyze large datasets. These computational advances push the limits of non-invasive imaging through scattering, that would open up many possibilities such as the observation of individual neurons deep inside the brain of alive animals.

Optical computing is uniquely suited to perform parallel computation efficiently for machine learning. Scattering will be the opportunity to realize randomly-wired optical neural networks of very large dimension. A particular emphasis will be put on the optical realization of a recurrent architecture called Reservoir Computing, particularly useful for the prediction of chaotic time series. In the end, we will show how these ideas from optics inspired us to draw a rigorous link with recurrent kernel methods, enabling us to considerably speed up Reservoir Computing even without the use of optics.

The tight link between these two lines of study is symbolized in the random matrix introduced to model the effect of scattering. In the first case, we need to *invert* the operation performed by the scattering medium, whereas in the second, our goal is to *use* this complex operation to process information in the optical domain. We will show how the regularity properties of high-dimensional random matrices allow us to obtain rigorous results and robust reconstruction guarantees.

Acknowledgments

To my supervisors, Sylvain Gigan and Florent Krzakala, I would like to express my sincere gratitude. Thanks for all your support and guidance, it was a pleasure working with you both. A PhD is an important step as it concludes the academic studies, and I'm glad I could do it with you.

From the bachelor internship to a PhD, I owe a lot to Sylvain. Thanks for accompanying me through most of my university degrees, showing how to conduct research projects and navigate the academic world. If many of your alumni continued in scientific research, it would certainly have to do with the unique atmosphere in the lab and the role model you represent. I will fondly remember the great moments and good food we have shared too!

I could also learn a lot from Florent, from his intuition and knowledge of the machine learning field. Thanks for having the patience to listen to all the different ideas we explored but did not converge on, making studious moments enjoyable with humour and detachment. The different summer schools were great opportunities to meet very bright researchers and enjoy being in beautiful places at the same time. We did not have time to play some music together, but I hope we can make up for it in Lausanne!

Special thanks as well to Lenka Zdeborova, who was instrumental in creating this stimulating research environment we all benefitted from. Your passion for science and your drive to make a positive impact around you are an inspiration for all of us.

To the members of the COMEDIA and SPHINX groups, I was privileged to be part of these two awesome teams. PhDs are relatively short (especially when they end), but friendships last for much longer. Thank you for all the moments we could share!

These three years were also the opportunity to meet other researchers. First, I would like to thank Laura Waller who has played an important part in my academic journey, from my first research project in 2015 to the frequent visits in her group the last few years. I was also lucky that I could discuss at length about mathematical questions with Romain Couillet and Yue Lu, where their kindness, expertise, and pedagogy helped a lot.

I would also like to thank LightOn, for the joint work we have carried out. To Igor Carron, Laurent Daudet, and the entire team, all the best for the future!

This PhD experience would not have been possible without all the administrative staff of LKB. I'm not the best at these tasks and their help was much needed to make it as smooth as possible.

Of course there would be plenty of other people to acknowledge here. I try not to be too long, focusing on professional relationships built over the years, so many thanks to family and friends! I'm blessed to have you close to me.

Contents

CHAPTER 1

General introduction

In this thesis, we investigate the interplay between complex media optics and non-linear optimization algorithms. This work is divided in two halves of equal importance: computational imaging and optical computing. In this introduction, we give an overview of the general background in both domains and emphasize the particular role of random matrix multiplication in both cases. This generic operation shuffles information linearly in a neural network and is intimately linked with multiple scattering in optics.

1.1 Random matrices in machine learning

1.1.1 Context

We live today in the information age. Technology has brought us a powerful tool to communicate with others, get informed about the world, learn about new concepts, or entertain ourselves. All these new possibilities only a few clicks away, available with unprecedented ease. The societal impact of digital technologies has been demonstrated during the on-going public health crisis (as of 2020), when they offered a precious way to stay connected with the world from the comfort of our homes. On the other hand, these tools are also very powerful at analyzing large crowds of people for governments and private companies. As we get more and more interconnected, privacy and control of our digital lives become increasingly hard to protect. For better or for worse, the momentum of this technological progress will only carry forward, and it is important to study how it works to understand its potential and limits. Of course, this limited research dissertation only studies a small facet of these complex questions and paints a small part of this big picture.

It is important to introduce how computations are performed nowadays. At the heart of this technological revolution lie small bits of information, a large number of 0s and 1s, manipulated and stored in our computers and smartphones. Our devices contain very efficient electronic cards, with many small transistors performing logical operations on bits. The speed of the technological progress in the past 50 years has been powered by the increasing capabilities of the computing hardware at our disposal, embodied by the Moore's law: the number of transistors on integrated circuits doubles about every two years [1]. Depicted in Fig. 1.1, this exponential growth enabled the development of more and more sophisticated algorithms, and it also contributed to the increase of memory capacity and the finer resolution of digital sensors.

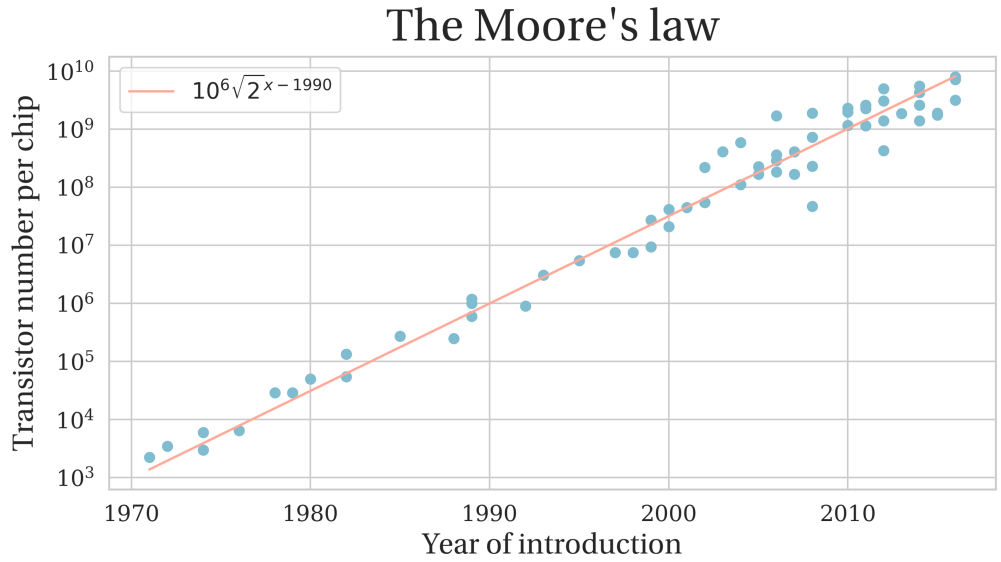


Figure 1.1: An illustration of the Moore's law. The number of transistors per electronic board (or chip) has been doubling every two years since 1970. This exponential increase in computing power has driven the progress of numerical technologies. Data source: Wikipedia (Transistor count).

1.1.2 A brief history of Machine Learning

These very rapid technological advances are at the core of the Machine Learning revolution. Already in the 50s, artificial neural networks have been proposed, with the very first denominated the perceptron [2]. A lot of hopes were built around this machine, The New York Times describing it as "the embryo of an electronic computer that [the Navy] expects will be able to walk, talk, see, write, reproduce itself and be conscious of its existence." [3] Despite all these promises, its performance was lackluster at the time. Indeed, the computation capabilities were too limited back then for the neural network to learn useful tasks. Research on artificial neural networks remained confidential for several decades, even though some researchers still pursued this idea. In 1989, Yann LeCun et al. developed a convolutional neural network to recognize handwritten ZIP codes on mail letters [4], this preliminary study built the foundations on which neural networks are built today.

An artificial neural network is composed of artificial neurons, connected together to perform a certain task. Each neuron is a unit, that receives information from other neurons, processes it with a non-linear function, and sends some information to other neurons (see Fig. 1.2). They can be organized in a succession of layers (and we call them in this case Deep Neural Networks), or interconnected with less structure (Recurrent Neural Networks) as shown in Fig. 1.3.

In 2012, the Machine Learning revolution began [5]. For the first time, a Deep Neural Network outperformed by a large margin all the other computer vision algorithms on the ImageNet competition (Fig. 1.4), challenging researchers to propose algorithms for object recognition based on images [6]. This convolutional neural network called AlexNet (Fig. 1.5) with 61 million parameters achieved an unprecedented error rate, beating consistently the performance of traditional computer vision al-

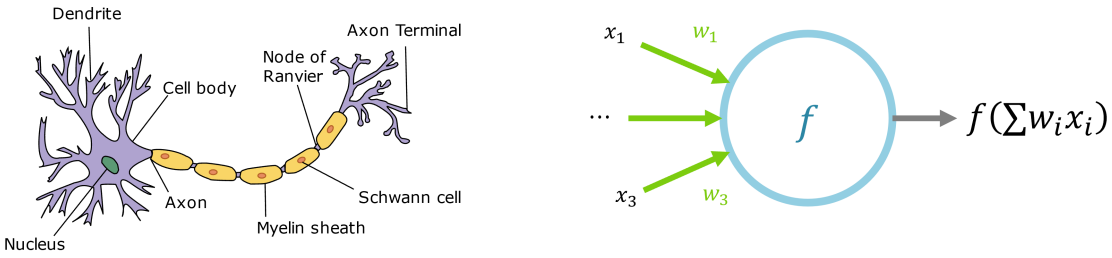


Figure 1.2: A biological and an artificial neuron. A biological neuron receives inputs from other neurons in its dendrites and sends information to other neurons at the end of its axon. The interconnection strengths between neurons are slowly modified for learning. An artificial neuron is a simplified model that receives an input x with weights w and applies a non-linear function f to this linear combination. The weights w are tunable to perform certain tasks.

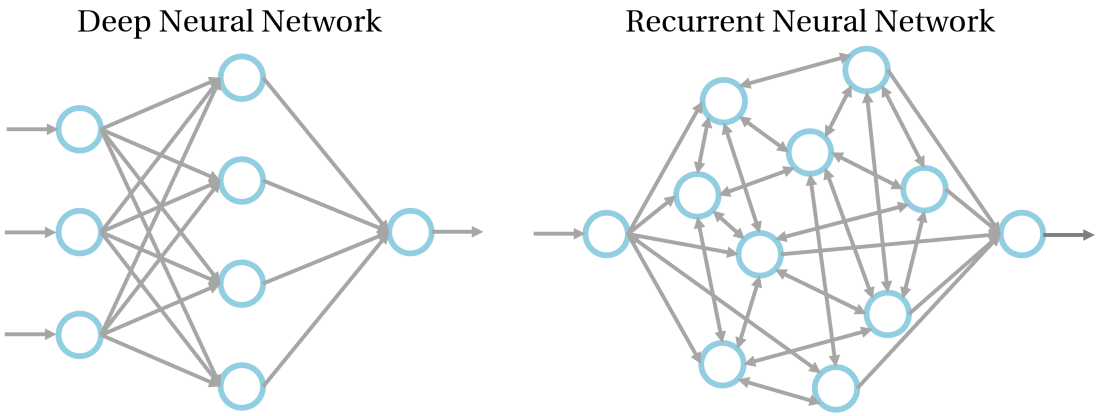


Figure 1.3: Deep and Recurrent Neural Networks. Artificial neurons are used together to form these neural networks. In Deep Neural Networks, the information is flowing from layer to layer in a forward sequence. They are very popular today and are used in most Machine Learning applications as they are well-suited for training with iterative methods. With Recurrent Neural Networks, neurons are densely interconnected without this feedforward structure. They are better suited for time-dependent datasets, but training them remains a difficult challenge.

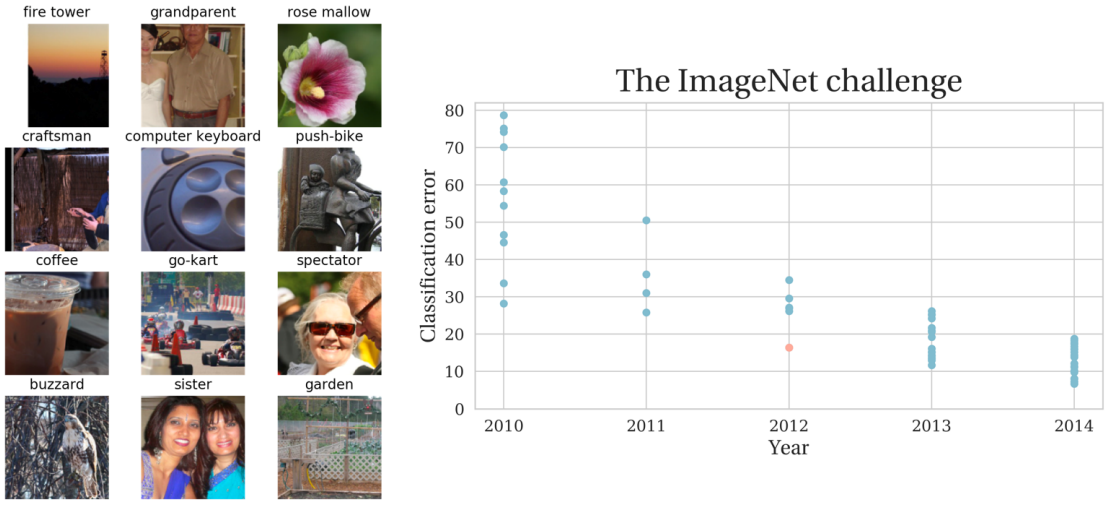


Figure 1.4: Results from the ImageNet competition and the architecture of the Convolutional Neural Network AlexNet that won in 2012. The ImageNet dataset is a large-scale image categorization challenge. We can see how the performance had reached a plateau with conventional computer vision techniques and in orange how much AlexNet improved the state-of-the-art performance in 2012.

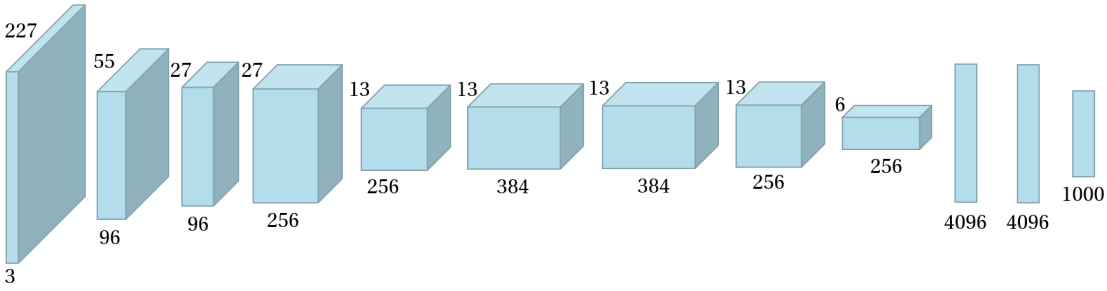


Figure 1.5: Scheme of the Alexnet architecture. It is made of several convolutional layers to extract features from the images, followed by fully-connected layers to perform the classification.

gorithms that were plateauing. Since then, there has been an exponential increase in the numbers of research articles on this field called Deep Learning, and its applications have permeated the whole society [7]. Today, social networks use neural networks to analyze images automatically, with human-like performance. It has for example enabled the development of autonomous cars, that are able to make sense of their surroundings [8].

1.1.3 Reasons for this success

The recent success of Machine Learning lies on three pillars that explain its surprising performance.

An avalanche of data: Artificial neural networks require large datasets to train and learn how to perform complex tasks. For example, the ImageNet 2012 challenge

contained more than 14 million images of more than 20,000 categories, labeled using a crowd-sourcing platform. With the availability of larger and better sensors, data has become more and more abundant. For example, a connected car generates 25 GB of data per hour of driving and an airplane 10 TB per 30 minutes of flight [9]. One then needs algorithms to make senses of this tremendous amount of data.

Non-linear optimization: Neural networks have a large number of parameters that need to be tuned in order to perform a particular task. These parameters are trained using iterative algorithms [10]. Based on pairs of examples x with known labels y, a loss function $l_w(x, y)$ parametrized by weights w is constructed and minimized using gradient-descent (Fig. 1.6): we compute the gradient (i.e. how slightly changing each weights affect the loss function) and use it to refine the weights w. This process is performed iteratively, and it hopefully converges towards an appropriate set of weights. It is important to note that this non-linear optimization is complex and not fully understood. For example, it is difficult to tell if at the end of the optimization process, the algorithm has found the optimal set of parameters (global minimum of the loss function), as it can also be stuck in poor local minima or saddle points. A lot of theoretical efforts have recently been gathered to analyze and study this important problem. Understanding better the dynamics of gradient descent in neural networks would help accelerate them and understand their limits.

Heavy computation: Neural networks are rather heavy models that operate on large datasets. Hence, they need a high amount of computational power. Actually, the algorithm developed in 2012 [5] was based on the same ideas as the CNN by [4]. The novelty came from the efficient Graphics Processing Unit (GPU) optimization they implemented based on the CUDA library. GPUs are specialized computing hardware optimized for matrix operations (Fig. 1.6), that have initially been developed for video rendering, decoding, and encoding. They have found themselves very useful for machine learning in the past decade. GPU clusters were used to train larger and larger models, for example [11] used 800 GPUs at the same time. This raises important questions about the energy-efficiency of all these techniques, as these large computational capabilities are out-of-reach of most academic institutions.

1.1.4 Introducing random matrices

Matrix multiplications play an important role in these models. To go from one layer to the next in a neural network, we have to multiply by the weight matrix between these layers. Large-scale matrix multiplications are common, and machine learning is at the interplay between well-understood linear models and simple element-wise non-linearities in each neuron. For instance the last fully-connected layers of AlexNet involve multiplications by matrices of size 9216×4096, 4096×4096, and 4096×1000. Each of these matrices contains millions of weights to be tuned during training. As stated previously, this iterative training process is not well understood and still the subject of intense study by the research community.

An alternative consists in fixing the weights randomly. This alternative present interesting advantages because it makes the network faster to train, easier to analyze, and the network is still performing a generic non-linear operation on the input data. There are many examples in machine learning where this choice has been

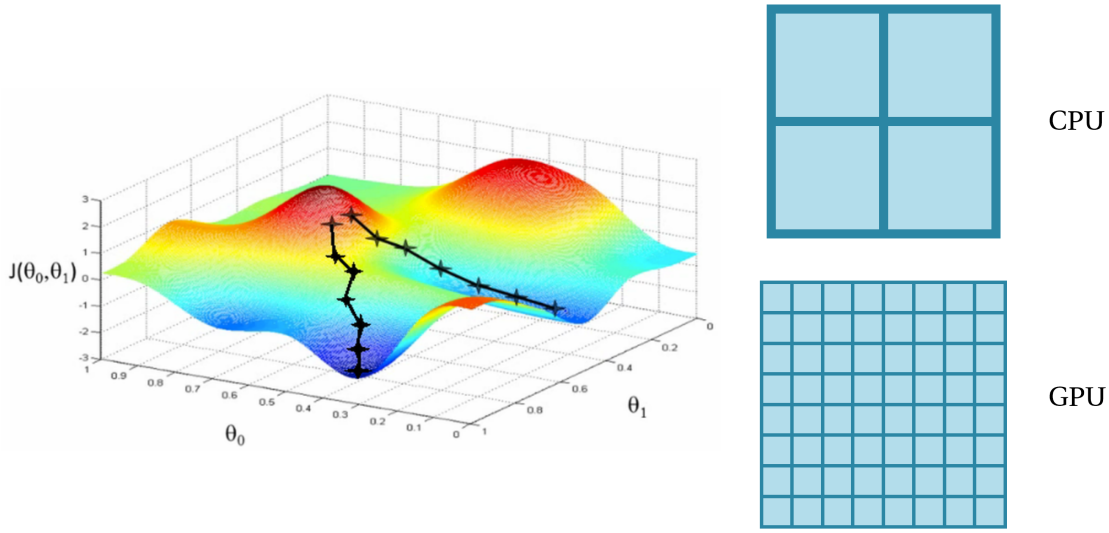


Figure 1.6: Left: Gradient descent. To solve non-linear optimization problem, gradient descent is an iterative technique using the gradient of the loss function to obtain a descent direction. In complex non-linear optimization problems, several local minima may be present and gradient descent may be stuck in one of them, depending on the initial starting point. Right: Comparison between a Central Processing Unit (CPU) and a Graphics Processing Unit (GPU). A CPU is made of 2 to 16 versatile cores. The GPU uses an array of smaller cores to enable more efficient parallel computations.

made (Fig. 1.7). Already in 2001, H. Jaeger proposed to fix all the weights of a Recurrent Neural Network [12], as this class of network is known to be very difficult to train [13], [14]. This proposal has inspired a lot of subsequent works and created a whole field called Reservoir Computing that we will investigate more in detail in the following [15]. These randomly-wired neural networks have been studied recently and they approach state-of-the-art performance for certain tasks even compared to Recurrent Neural Networks where all the weights are trained [16]–[18]. Interestingly, randomly-wired Convolutional Neural Networks with very little training have also shown surprisingly-good performance for denoising, super-resolution and inpainting [19], an approach which has been applied to computational imaging [20], [21].

Random Features were proposed by [22]. In this feedforward architecture with two-layers that we will study in depth later, the first layer is fixed while the second is trained. This specific class of algorithms has attracted quite a lot of attention with [23] and [24]. As Random Features can be seen as a linear model trained on a non-linear transformation of the input data, they have profound links with kernel methods [25]. They provide evidence that high-dimensionality and non-linearities may be more important than the exact weight values.

Interestingly, it is possible to perform large scale multiplications by random matrices in optics, obtained whenever light propagates in a complex material. We leverage this potential to implement optically such neural networks with random weights [26]–[28]. This optical computing strategy is fast and energy-efficient, but at the cost of noise and a preprocessing step that is mandatory for the input data. This line

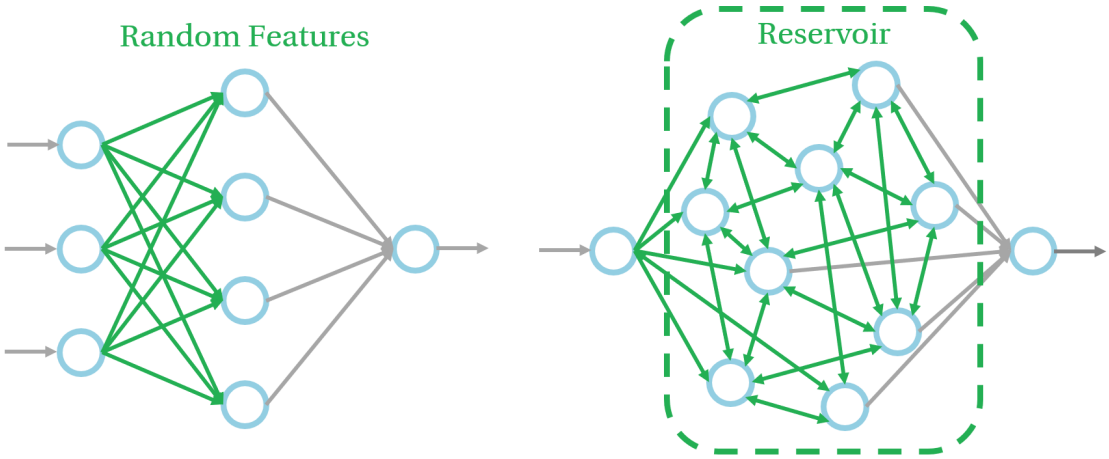


Figure 1.7: Randomized Neural Networks. Left: Random Features is an example of a feedforward architure with fixed random weights, only the second layer is trained. Right: Reservoir Computing represents a simplified model for Recurrent Neural Networks. All the internal weights are fixed and only the output weights are trained with a simple linear regression.

of development is pushed by LightOn, a company cofounded by my two PhD co-supervisors.

Using random weight matrices also helps us understand what happens in the large size limit [29], [30]. Whereas there are not many mathematical tools available to tackle general non-linear optimization problems, there exists a whole field called Random Matrix Theory describing the properties of these random matrices when their size grows large. Powerful theorems characterize how fast properties of random matrices converge towards their expectation value, and it will be a valuable tool to derive rigorous results on non-linear optimization models.

Hence, I will show in this thesis how this link between random matrices and machine learning can bear fruitful insights on the behavior of large neural networks.

1.2 Random matrices with optical scattering

1.2.1 Linear optics

Random matrices can also play an important role in optics. To introduce how one may encounter a random matrix, let us introduce first the basics of linear optics. In classical optics, light propagation is fundamentally described the Maxwell's equations, from which can be derived how electric and magnetic fields propagate. Neglecting for simplicity the magnetic component (coupled with the electric field), the broadband nature of light (assuming that we have a monochromatic laser) and polarization (that adds another degree of freedom), we focus here on the electric field propagation.

In any given plane, be it the one of a camera, or a biological slide to look under a microscope, it is described by a complex-valued field. This field propagates, and any

propagation can be modeled by a linear operator, i.e. a linear matrix multiplication applied on the electric field [31]. This linear model is valid for free-space propagation, through lenses, or even more complex optical elements, as long as there is no non-linear phenomenon.

For example, a lens performs a Fourier transform: a plane wave, after passing through a lens, converges to a point in the focal plane of the lens (Fig. 1.8). With two lenses, one can image objects on a different plane, the principle behind microscopes and telescopes. Free-space propagation may even be described as a linear transform using the Rayleigh-Sommerfeld convolution integral [31]. Non-linearities might happen with events such as inelastic scattering, specific non-linear elements or when the optical power is high [32], but in this study we will stay in the regime of linear optics.

1.2.2 Propagation through complex scattering media

In real life, one may encounter many optical media more complex than a lens: fog, milk, or white paint on a wall. In all cases, a thermodynamic number (i.e. of the order of the Avogadro constant $N_A \sim 10^{23}$) of refractive index inhomogeneities are present at apparently random positions and they scatter (i.e. deviate) light (Fig. 1.9). A coherent beam of light will therefore propagate through this kind of media along many different optical paths, resulting in an intricate figure of random interferences, also called a speckle pattern [33] (Fig. 1.10). Other more exotic but still important cases are opaque biological tissues (skin, bones, and brain tissue for example) and multimode fibers (fibers with large cores, where input modes are mixed as they propagate). Propagation through such complex media is impossible to describe completely, but it can still be modeled by a linear operator that connects the ingoing and outgoing waves [34], [35].

This so-called Transmission Matrix (TM) can be experimentally measured to characterize light propagation in such a complex medium. A proof-of-concept experiment was performed in 2010 [36], and it opened up new possibilities for imaging through complex media. This Transmission Matrix characterizes light propagation through a particular realization of disorder. As long as the medium is static, it will perform this deterministic transform that links the electric field at an input plane to the one at an output plane.

A Transmission Matrix can be considered random, as it depends on the positions of the different inhomogeneities in a complex medium. To discuss in more detail this essential point, the complete scattering matrix of the medium (describing both forward and backward scattering) is unitary thanks to energy conservation, with many non-trivial properties (open modes, correlations) described by mesoscopic physics [35]. However, we only consider a subsampled version of the Scattering Matrix, the Transmission Matrix, which does not account for backward scattering. Moreover, the camera has a finite area and does not collect all the transmitted modes. As such, a complex scattering medium performs a multiplication by a random matrix where each element is drawn according to a complex Gaussian distribution. This is reminiscent of the random matrices encountered in machine learning.

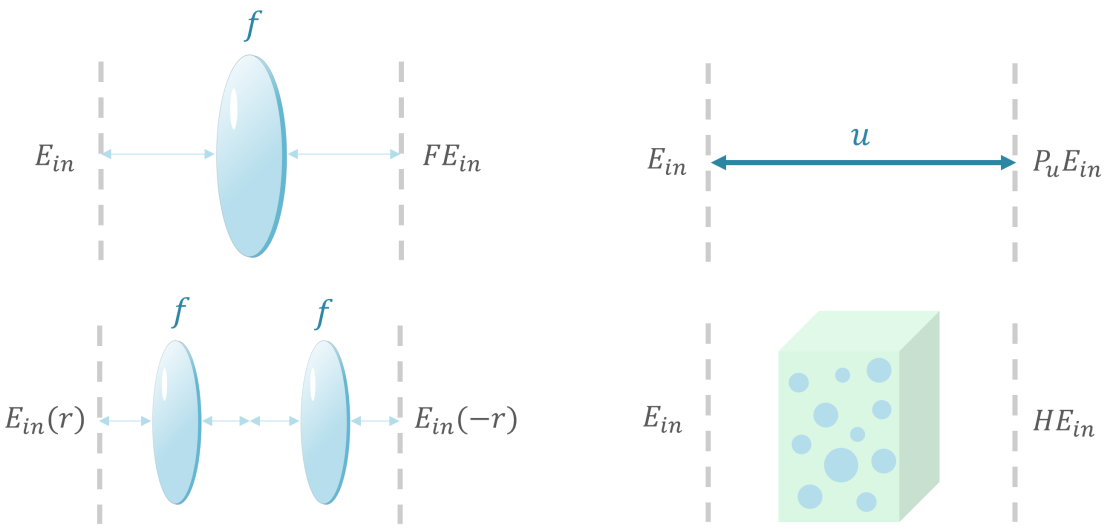


Figure 1.8: Linearity of electric wave propagation. Top left: Light propagation through a lens can be modeled by a Fourier transform. F denotes the Discrete Fourier Transform matrix. Bottom left: Propagation through two conjugated lenses is often used for imaging. Top right: Free-space propagation is also a linear operation, the explicit matricial formulation Fresnel diffraction. Bottom right: Propagation through complex media remains linear, and this time, we will model it by a dense random Transmission Matrix H.

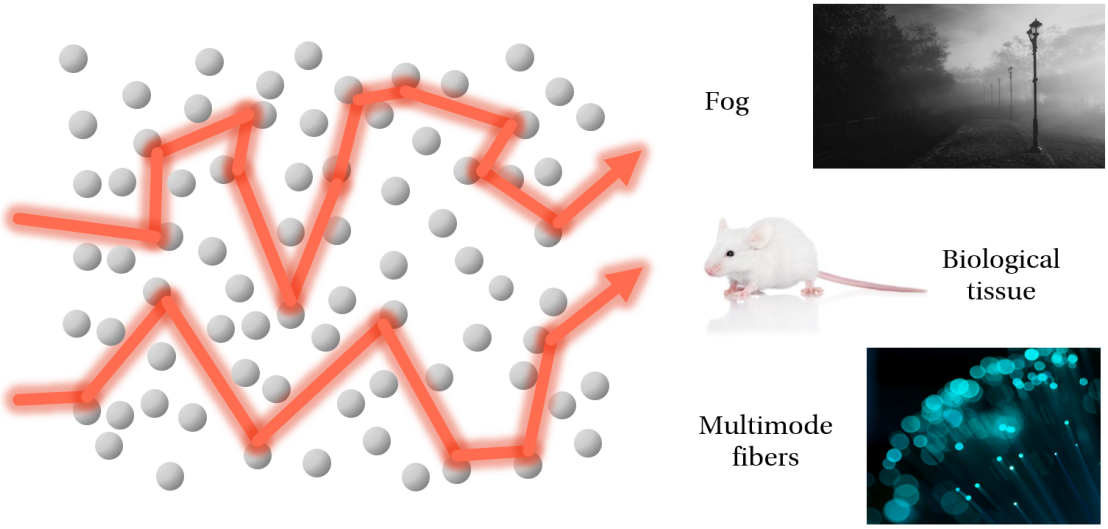


Figure 1.9: Multiple light scattering. Left: In complex heterogeneous media, light does not propagate in a straight line but is scattered by refractive index inhomogeneities. Right: There are many examples of this multiple light scattering in our daily life, for example fog, biological tissue, or multimode fibers.

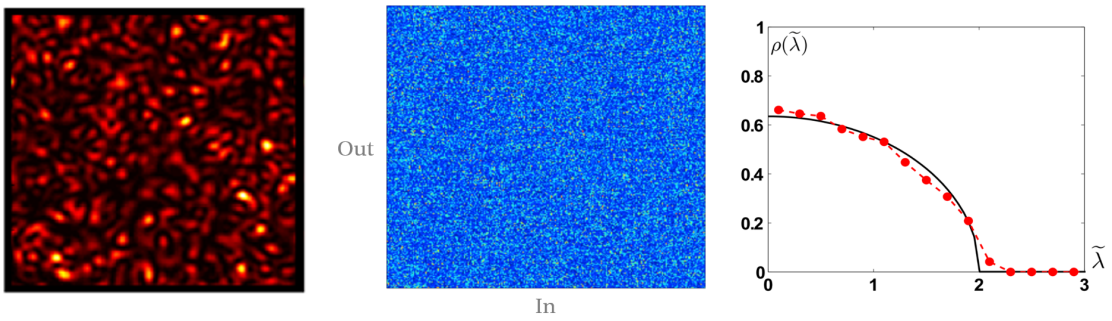


Figure 1.10: The Transmission Matrix. Left: A speckle pattern image resulting from the complex interference pattern generated after multiple light scattering. Middle: An example of a Transmission Matrix, a large-dimensional random matrix. Right: The singular value distribution of the Transmission Matrix follows the quarter-circle law, which is a classical result with random matrices.

1.2.3 Efficient large-dimensional hardware in optics

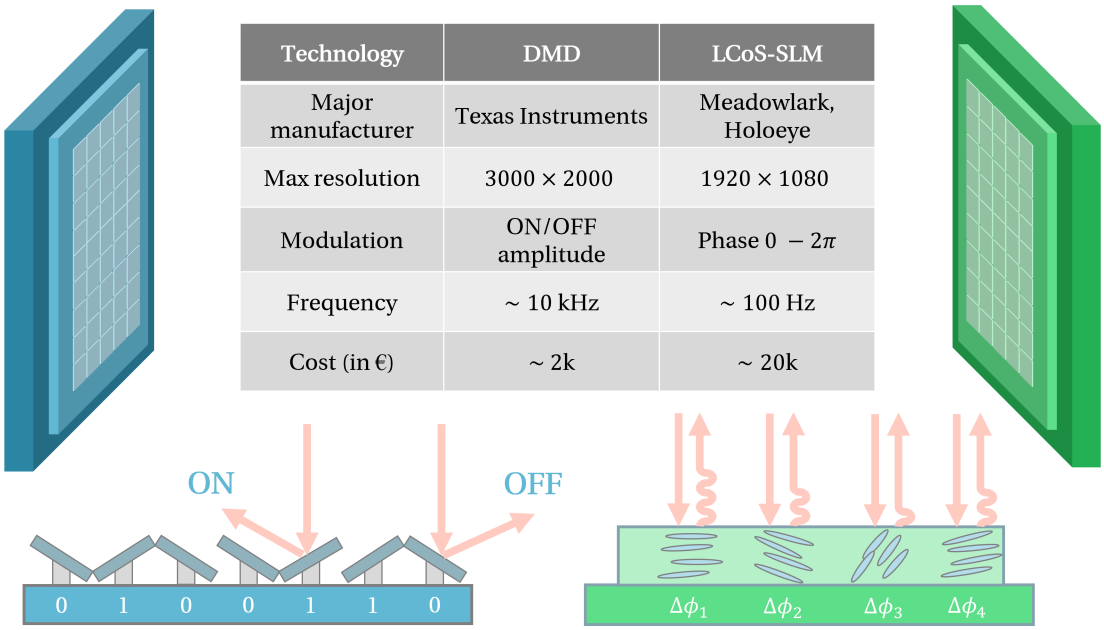
To continue the parallel with machine learning that has been enabled by the efficiency of GPU for vectorized computations, recent advances in optics were enabled by the development of more and more powerful hardware in optics. Most experimental designs that will be presented are quite simple, but they rely on two important components: a Spatial Light Modulator and a camera. The Spatial Light Modulator imprints a digital image on an electric field, and the camera measures the intensity of the electric field in a different plane, typically after propagation through a complex medium. They are the interface between the optical domain and a computer operating in the electronic domain.

Spatial Light Modulators (SLMs) can typically be associated with display devices in our everyday life (Fig. 1.11):

- Computer Screens are based on the Liquid Crystal technology. With an appropriate set of polarizers, Liquid Crystals can modulate the amplitude or the phase of the electric field, depending on the applied voltage. Liquid Crystals on Silicon (LCoS) SLMs enables a precise high-resolution modulation of any electric field, but they are relatively slow for our demanding applications, as they can display on the order of 100 images per second.

- Videoprojectors use a Digital Micromirror Device (DMD). It is made of an array of micromirrors with two different orientations, that can be flipped on or off. They are very fast as their operating frequency exceeds 20 kHz, but only allow a binary amplitude modulation.

These two technologies are quite inexpensive nowadays, especially DMDs, as they have been pushed forward by the consumer market. For the same reason, their resolution is remarkable as they typically have millions of pixels. There also exists optical microelectromechanical systems (MEMs) with deformable mirrors, that were developed for adaptive optics. These enable phase modulation at high speed, but are quite expensive and have a limited number of pixels ($\sim 10^3$).



Technology	DMD	LCoS-SLM
Major manufacturer	Texas Instruments	Meadowlark, Holoeye
Max resolution	3000 × 2000	1920 × 1080
Modulation	ON/OFF amplitude	Phase $0 - 2\pi$
Frequency	~ 10 kHz	~ 100 Hz
Cost (in €)	~ 2k	~ 20k

Figure 1.11: Comparison of different SLM technologies. A Liquid-Crystal SLM is able to modulate the phase of the electric field with high resolution but limited speed. On the other hand, a Digital Micromirror Device is quite fast and still has a high resolution, but it can only generate binary images. This is due to the underlying technology based on micromirrors, that can be mechanically turned on or off depending on their orientations.

In a similar vein, cameras also offer high-quality high-resolution sensors. To convert photons into electric charges, two strategies exist with CCD or CMOS cameras. They produce Megapixel images that contain a considerable amount of information. In recent years, cameras have been considerably developed improving both resolution, miniaturization, and photon efficiency (being able to capture images in low-light settings). It is also common nowadays to pair these sensors with algorithms, and we will show how to design them in the challenging context of multiple light scattering.

In the end, the Transmission Matrix links the electric field displayed at the SLM plane (the input plane), to the electric field at the camera plane (the output plane). It connects a high-dimensional ($\sim 10^6$) input to a high-dimensional output, which makes it an attractive topic of research in this information age. In the presence of a complex scattering medium, modes are mixed and the overall TM can be considered random.

1.2.4 Applications

The study of light propagation through complex media has found a number of applications:

- **Biological imaging:** Performing deep in-vivo imaging would allow the observation of cells deep inside a living animal. For example, it makes possible

to look at neurons from the inner regions of the brain of mice [37]. In these cases, scattering prevents conventional microscopy techniques and needs to be countered [38], [39].

- **Optical computing:** As any scattering medium realizes a multiplication by a fixed random matrix, it can be leveraged to perform large-scale computations for randomly-wired neural networks [26]–[28].

- **Fiber communication:** Optical fibers are commonly used to transmit data over long distances. Characterizing how modes are mixed in a multimode fiber could increase the bandwidth of these optical fibers by multiplexing information, important for communication [40]–[42] and imaging [43].

- **Quantum information:** One may send entangled photons through multimode fibers and take advantage of their complexity to perform an arbitrary linear operator [44], [45].

- **Non-line-of-sight imaging:** This recent field of research studies how to spy a room occluded from direct sight, looking at the light scattered on a wall reaching our detector [46], [47].

In this work, we will focus on the first two points.

1.3 At the intersection of these two fields

Our goal here is to show how the interplay between complex media optics and large-scale computations in machine learning provides interesting insights in both domains. The leitmotiv throughout this thesis will be the presence of random matrices, that we will experimentally measure, characterize their singular value distribution, design machine learning algorithms around them, or compute asymptotic limits of some quantities based on their probabilistic nature.

In computational imaging, one wants to see through or inside complex media. SLMs and cameras are powerful tools to do so, and we design algorithms to retrieve images. We want to *invert the random matrix multiplication* performed by the complex medium. This operation needs to be performed non-invasively, which is an important constraint for in-vivo applications. We will discuss how to solve a generic phase retrieval problem, measure transmission matrices non-invasively, and distinguish targets by looking at the singular value distribution of a random matrix.

In optical computing, one wants to accelerate machine learning computation using optics. Random weight matrices are present in many neural networks, and are typically associated with expensive large-scale computations. We want to *accelerate the random matrix multiplication* using a complex medium. We will discuss how to accelerate both feedforward and recurrent neural networks with optics, make the link with kernel methods, and compare this optical computing strategy with current CPUs and GPUs. As optics allow us to scale to large dimensions, we will also provide in the end an interesting theoretical result for recurrent neural networks, where

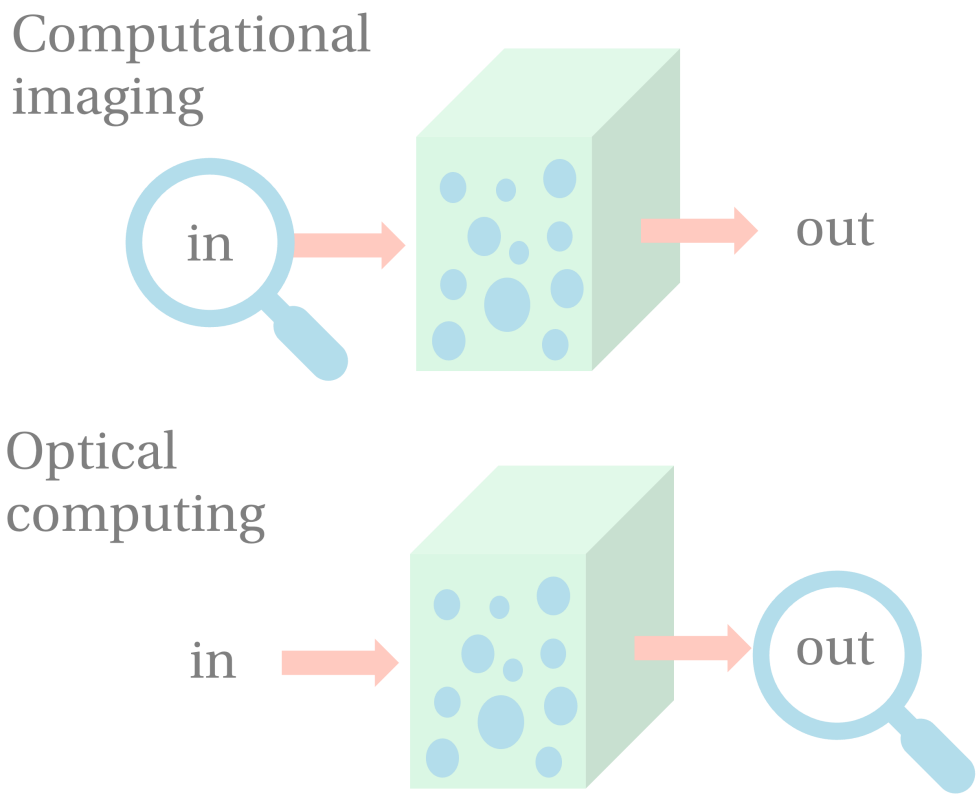


Figure 1.12: Computational imaging vs optical computing. In computational imaging, we measure the result of the complex light propagation and would like to image an object deep inside the scattering medium. With optical computing, we accelerate the random matrix multiplication using optics, using the measured output in machine learning algorithms.

we directly perform computation on the asymptotic limit of an infinite-size neural network, without optics.

In a nutshell, we investigate how information is transformed by a multiplication by a large random matrix (Fig. 1.12). Even though they share many common ideas, both parts are independent and can be read separately. An emphasis will be put on the theoretical and computational insights brought by these studies, rather than the experimental procedure of the optical implementations.

1.4 Personal contributions

As presented in this introduction, the research performed during this PhD lies at the intersection of two fields, between two groups studying complex media optics and statistical physics for machine learning. It has resulted in the production of 12 peer-reviewed journal articles and proceedings.

1.4.1 Optical Machine learning

The main project of the PhD was initially the exploration of our optical computing implementation for machine learning, and in particular the framework of Reservoir Computing. Optical computing to accelerate machine learning is a blooming field: convolutional neural networks have been implemented optically on integrated nanophotonic circuits [48] or in free-space [49], building on early pioneering works in the 90s [50], [51].

Our work takes the original approach of exploring how to use multiple light scattering for optical computing. Compared to the previous approaches and to other physical implementations of Reservoir Computing [52], [53], our approach scales better to very large network sizes, up to 10^5 [27]. Thus, we obtain a system that becomes competitive with electronics at these very large sizes.

However, the promises of this scientific endeavor need to be balanced by also presenting the challenges encountered in analog computing. Encoding digital information in the analog domain, communication bandwidth between devices, and noise robustness remain unavoidable questions to tackle in optical computing. Moreover, our approach that makes it possible to generate very large networks has inspired a surprisingly successful theoretical work. We show that similar gains can be obtained without optics, by applying Random Features acceleration schemes to Reservoir Computing.

In the end, the presented work exhibits both an effort towards high-performance optical computing and an acceleration strategy without optics to nuance the previous claim. While promising, optical computing still needs further improvements before it overtakes electronics on specific operations. **All this line of work is presented in Chapter 6 with more technical details in Appendix D.**

- [27] **J. Dong**, S. Gigan, F. Krzakala, G. Wainrib (2018). ***Scaling up Echo-State Networks with multiple light scattering***. In 2018 IEEE Statistical Signal Processing Workshop (SSP) (pp. 448-452). IEEE.
  This work demonstrated the proof-of-concept of using multiple light scattering to perform large-dimensional Reservoir Computing. Personal contributions: design, code, data production, writing.

- [28] **J. Dong**, M. Rafayelyan, F. Krzakala, S. Gigan (2019). ***Optical Reservoir Computing using multiple light scattering for chaotic systems prediction***. IEEE Journal of Selected Topics in Quantum Electronics, 26(1), 1-12.
  This work compared different modulation technologies and proposed a model for the encoding step on the SLM to improve performance. Personal contributions: idea, theory, code, data production, writing.

- [54] M. Rafayelyan, **J. Dong**, Y. Tan, F. Krzakala, S. Gigan (2020). ***Large-Scale Optical Reservoir Computing for Spatiotemporal Chaotic Systems Prediction***. arXiv preprint arXiv:2001.09131.
  This work pushed the limit of this optical implementation Reservoir Computing to reproduce the latest Reservoir Computing achievements. Personal contributions: initial experiment, code, review.

Two more theoretical works study the asymptotic limit of Optical Random Features and Reservoir Computing :

- [55] R. Ohana, J. Wacker, **J. Dong**, S. Marmin, F. Krzakala, M. Filippone, L. Daudet (2019). ***Kernel computations from large-scale random features obtained by Optical Processing Units***. IEEE ICASSP 2020.
  This work studies the kernel limit of the random features obtained optically and applies it to Transfer Learning. Personal contributions: idea, theory, writing, review. **Presented in Chapter 5.**

- [56] **J. Dong***, R. Ohana*, M. Rafayelyan, F. Krzakala (2020) (* = equal contribution). ***Reservoir Computing meets Recurrent Kernels and Structured Transforms***. arXiv preprint arXiv:2006.07310.
  This work studies the recurrent kernel limit of Reservoir Computing. Personal contributions: idea, theory, code, data production, writing. **Presented in Chapter 7.**

1.4.2 Computational imaging

As part of the daily discussions in a research group, we have also proposed new algorithms and experimental designs for the field of computational imaging in general, and imaging through complex media in particular. Most of the research presented here result from collaborations with other researchers, where I contributed mostly in the mathematical model and the algorithmic reconstruction.

We show how the recent developments in non-linear optimization can help us reconstruct objects in challenging settings, either because they are hidden inside a scattering medium or because measurements are few. Phase Retrieval and low-rank matrix factorization are the two main algorithmic tools in these studies, to provide a well-understood and robust reconstruction. To put it in perspective, there is no deep neural network here, even though they represent a very trendy and promising direction for computational imaging.

The exploration in complex media imaging includes the following work:

- [57] T. Wu, **J. Dong**, X. Shao, S. Gigan (2017). ***Imaging through a thin scattering layer and jointly retrieving the point-spread-function using phase-diversity***. Optics Express, 25(22), 27182-27194.
  This work jointly retrieves a hidden object and the optical system Point-Spread Function (PSF), based on a phase retrieval algorithm applied on a stack of images retrieved by translating the camera. Personal contributions: design, review. **More details in Appendix A.**

- [58] **J. Dong**, F. Krzakala, S. Gigan (2019). ***Spectral Method for Multiplexed Phase Retrieval and Application in Optical Imaging in Complex Media***. In ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 4963-4967). IEEE.
  This work proposes a spectral method (based on the eigenvalue decomposition of a particular matrix) to retrieve information in deep fluorescence microscopy

with a single-pixel camera. Personal contributions: design, experiment, code, data production, writing. **Presented in Chapter 3.**

- [59] A. Boniface, B. Blochet, **J. Dong**, S. Gigan (2019). ***Non-invasive light focusing in scattering media using speckle variance optimization***. Optica Vol. 6, Issue 11, pp. 1381-1385.
  This work introduces a simple optimization procedure to focus light in deep fluorescence microscopy. Personal contributions: theory, review. **More details in Appendix A.**

- [60] A. Boniface, **J. Dong**, S. Gigan (2020). ***Non-invasive focusing and imaging in scattering media with a fluorescence-based transmission matrix.*** Nature Communications (accepted, in press).
  This work introduces a double Transmission Matrix recovery involving matrix factorization and phase retrieval, to image complex fluorescent objects in scattering media. Personal contributions: idea, code, review. **Presented in Chapter 4.**

- [61] T. Wu, **J. Dong**, S. Gigan (2020). ***Non-invasive single-shot recovery of point-spread function of a memory effect based scattering imaging system***. Optics Letters (accepted, in press).
  This work shows how to recover the PSF of a complex optical system in a single-shot, based on the autocorrelation imaging technique and the statistical properties of the speckle. Personal contributions: idea, discussion, review. **More details in Appendix A.**

Other directions not involving any complex medium were also explored, to improve the algorithmic reconstruction process in Raman imaging and ptychography:

- [62] F. Soldevila, **J. Dong**, E. Tajahuerce, S. Gigan, H. B. de Aguiar (2019). ***Fast compressive Raman bio-imaging via matrix completion***. Optica, 6(3), 341-346.
  This work uses a matrix factorization algorithm for a Raman microscope, to compress the amount of collected data and reduce acquisition time. Personal contributions: idea, discussion, review. **More details in Appendix A. Featured in** `https://www.sciencedaily.com/releases/2019/03/190314101323.htm`

- [63] L. Valzania*, **J. Dong***, S. Gigan (2020) (* = equal contribution). ***Accelerating ptychographic reconstructions using spectral initializations***. arXiv preprint arXiv:2007.14139.
  This work applies the spectral method to accelerate the phase retrieval reconstruction of ptychography, with a demonstration on THz ptychography. Personal contributions: idea, code, writing, review. **More details in Appendix A.**

1.4.3 Outline

For a balanced and coherent presentation of these works, a particular emphasis will be put on projects involving random matrices, both in computational imaging and

optical computing. It will naturally be split in two halfs that can be read separately. A roadmap of the thesis is given in Figure 1.13.

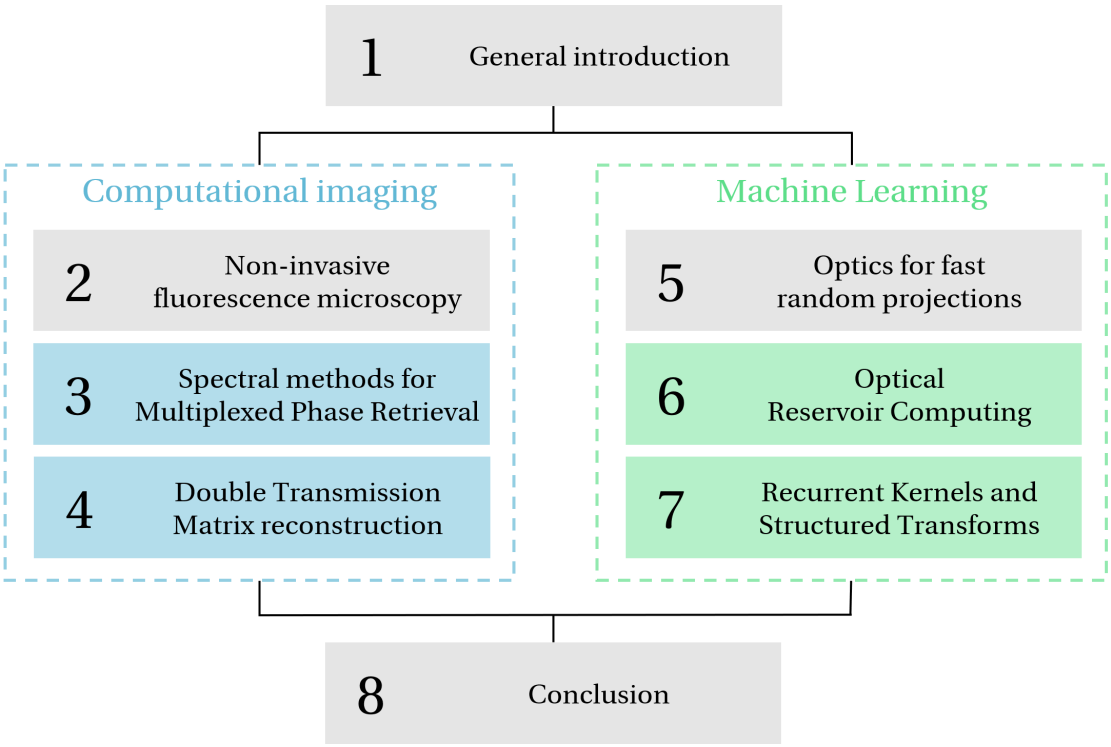


Figure 1.13: Outline of this thesis. Introductory chapters are in grey, computational imaging contributions in green, and machine learning contributions in blue.

In the computational imaging half, we will see how to model, analyze, and solve complex optical systems for deep fluorescence microscopy. After a specific introduction on non-invasive deep imaging in Chapter 2, we will present first a variant of Phase Retrieval, a canonical problem in computational imaging, and how to solve it looking at a particular eigenvalue-eigenvector decomposition. Then we will introduce a double Transmission Matrix approach to completely characterize the scattering process in a thick sample to image, in and out of the sample. The two matrices will be recovered with phase retrieval and low-rank factorization algorithms. Other contributions in computational imaging and a more detailed description of the very diverse techniques proposed to solve the Phase Retrieval problem are provided in the appendix.

In the machine learning half, we will show how to leverage the random matrix multiplication in optics and accelerate the computation of randomly-wired neural networks. In Chapter 5, we will introduce the general principle and its application in feedforward architectures, while the two following chapters are focused on Recurrent Neural Networks and a subclass with fixed internal weights called Reservoir Computing. Chapter 6 presents our optical implementation of Reservoir Computing and Chapter 7 how to accelerate Reservoir Computing even without optics. In the appendix are detailed a few technical proofs of theorems mentioned in the main text and how to successfully train a Optical Reservoir Computing implementation.

CHAPTER 2

Wavefront shaping for fluorescence microscopy

2.1 A quick overview of imaging

2.1.1 From measurements to images

In physics, waves carry information as they propagate. One may measure them in order to sense distant objects. Actually, two of our senses are based on this simple principle: our eyes measure electromagnetic waves in the optical domain, while our ears measure acoustic waves (pressure oscillations) that propagate in the air. Many other kinds of waves exist and are used to sense our world: x-rays are typically used in hospital scanners to observe bone structure, electrons (which are particles and waves at the same time thanks to quantum mechanics) are used nowadays to probe the shape of viruses or proteins [64], and even gravitational waves have been detected coming from the spectacular event of two black holes merging into one [65]. Simply put, all these waves can be described with the same physical equations, the main difference being their frequency.

Sometimes, the raw measurements of a sensor may be difficult to interpret. Imaging corresponds to the process that forms an image from these unpolished measurements. It is very valuable, as our eyes are very sharp tools to analyze these images and understand the world, but are somehow limited to a narrow spectral domain with a given frame rate and resolution. In a conventional camera, each pixel detects light from a different region of space and, put together, their responses form an image. Other imaging modalities are also possible, for example radars reconstruct an image from the echoes of short pulses they send. For a more recent example, the first image of a black hole, reconstructed from observations of telescopes all around the globe [66], gathered a lot of public attention and allowed researchers to verify their cosmological models. We will exhibit other exotic ways to recombine the raw measurements of the sensors in order to retrieve an image, to be used in challenging imaging situations.

2.1.2 The trade-off between resolution and penetration depth

In this work, we focus on imaging biological systems. For such applications, two notions are going to be useful:

- **Resolution** describes the smallest size of a detail that can be imaged with a given imaging system. The finer the resolution the more powerful the corresponding imaging system as it can resolve very small details of an object.

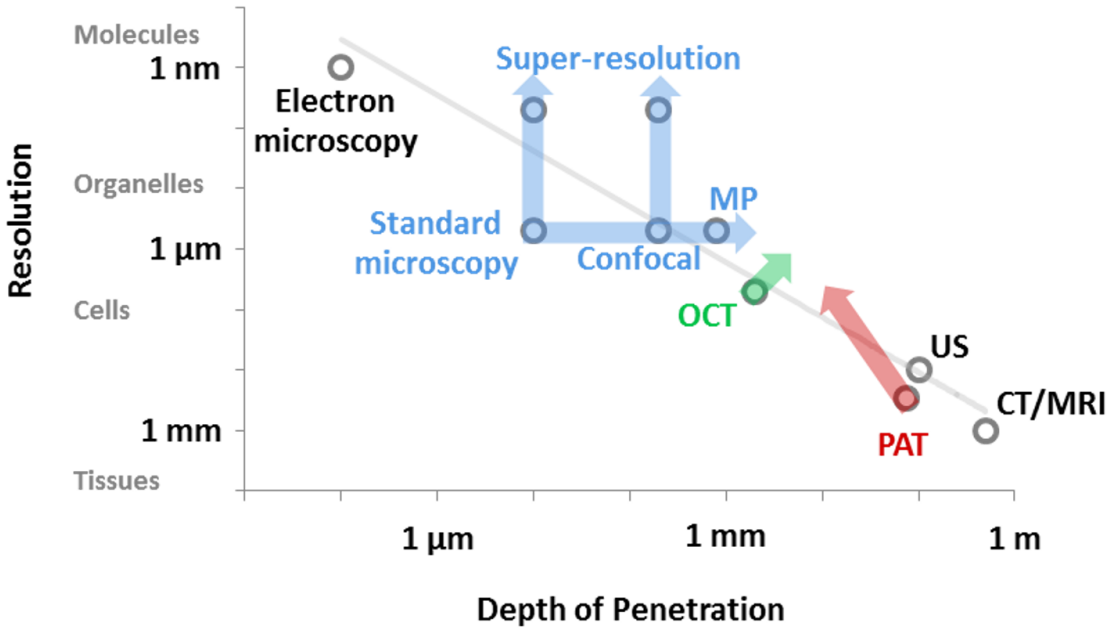


Figure 2.1: The trade-off between resolution and penetration depth. MP: multiphoton microscopy, OCT: Optical Coherence Tomography, PAT: Photoacoustic tomography, CT: Computed Tomography, MRI: Magnetic Resonance Imaging, US: ultrasound imaging. (Image from [68])

- **Penetration depth** describes how deep imaging remains possible. As light propagates through biological tissue, it gets scattered on inhomogeneities, its direction of propagation changes randomly, and imaging is not possible beyond a certain depth. Of course, this penetration depth depends on the imaging modality and the type of sample to study.

Several other quantities are required to describe an imaging modality, namely speed (important to image very fast objects or phenomena) or contrast mechanism (the physical phenomenon that generates or modifies wave propagation, that can then be detected for imaging, like absorption, phase, or fluorescence).

Magnetic Resonance Imaging (MRI) and X-ray Coherent Tomography (x-ray CT) are commonplace in hospitals to look inside the human body. They have the advantage of a very long penetration depth ($\sim$ 1 m), at the expense of a limited resolution, of the order of 1 millimeter, preventing the observation of single cells. Optical microscopes offer a much better resolution, resolving details below 1 micron, but their penetration depth is limited. For example, in brain tissue, penetration depth is of the order of one-tenth of a millimeter or 100 microns. Other techniques exist like Optical Coherence Tomography (OCT, to increase penetration depth) [67] or electron microscopy (to resolve very fine molecular structure) [64], but with current techniques there is a universal trade-off between resolution and penetration depth [68].

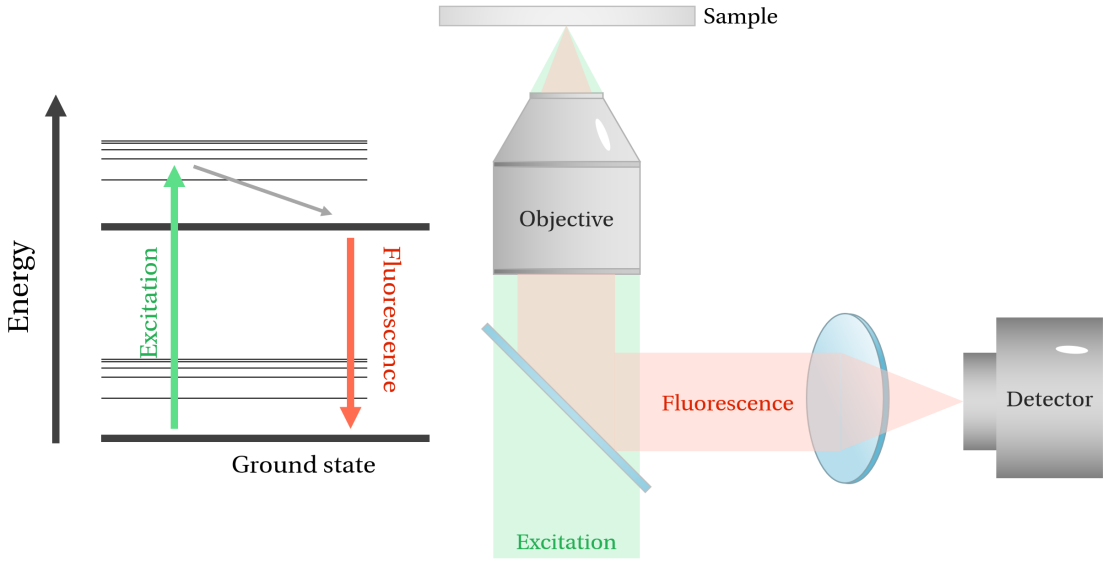


Figure 2.2: Fluorescence microscopy. Left: Scheme of the energy bands of a fluorescent molecule. An excitation beam excites the particle that fluoresces at a different wavelength. Right: The epifluorescence configuration. An excitation beam from a laser is sent on a sample to image, the fluorescence is collected from the same side of the sample using a dichroic mirror (DM).

2.1.3 Fluorescence microscopy

We will focus here on fluorescence microscopy [69] which is now widely used to probe the realms of biological systems. This technique uses the fluorescent light emitted by specific proteins, to form images of biological cells. Thanks to genetic engineering, these proteins can be only expressed in targeted types of cells, which makes fluorescence microscopy a very valuable tool to image with a specific contrast mechanism. For example, it has been used extensively to produce "brainbow" images [70], images of the brain in which individual neurons can be distinguished from their neighbors. Fluorescent proteins need to be excited with a laser at a given wavelength, and they reemit light at other wavelengths.

To image them, we typically use a conventional microscope in combination with a few specific filters. To increase their penetration depth beyond the limit of standard microscopy would enable new possibilities that are out of reach of other modalities, like functional imaging of deep regions inside the brain of alive animal. In a nutshell, fluorescence microscopy thanks to its contrast, speed, resolution and specificity is an inescapable tool in life science. Nevertheless, as any optical technique, it is limited to image transparent sample or only at shallow depth, because of light scattering.

2.2 Wavefront shaping to counteract scattering

2.2.1 Focusing with optimization

For deep fluorescence imaging, we thus need to overcome the problem of optical scattering. Several strategies have already been proposed to push the penetration depth limit. For example, combining optics with acoustics represents a promising road, as acoustic waves offer a much higher penetration depth, via acousto-optics [71] or photo-acoustics [72] (techniques that send or detect acoustic waves). However, their resolution is limited to about a millimeter due to the long wavelength of these acoustic waves. One may also use non-linear fluorescence [73], and this technique has actually already been used for deep brain imaging with mice, but it requires an expensive apparatus to send laser pulses on the sample.

In this work, we restrict ourselves to linear fluorescence and make the deliberate choice to focus on computational techniques to reconstruct images. The complex propagation from the SLM plane to the camera plane is modeled by a Transmission Matrix T and the camera measures the intensity of the electric field:

$$I^{\text{out}} = |TE^{\text{in}}|^2 \tag{2.1}$$

We will begin our discussion of optical scattering with a fundamental experiment that demonstrates the basic concepts and tools that will be developed later. Following the landmark experiment of [74], we place a Spatial Light Modulator (SLM) before the complex medium and a camera after. To counteract scattering, one may want to make the complex medium act like a lens; when we shine a plane wave on a lens, it focuses light onto a single point. Here, it is possible to shape the phase profile of the incoming wavefront to focus light. This experiment is the very first proof that wavefront shaping with a large number of modes is able to counteract scattering.

Let us choose an arbitrary camera pixel with index k to focus on. The intensity on this pixel is:

$$I_k^{\text{out}} = |T^k E^{\text{in}}|^2 \tag{2.2}$$

where T^k is the k-th row of T. Initially, the wavefront on the SLM is also arbitrary and the camera collects a random speckle image. We then use a simple optimization procedure. We choose one mode (or SLM pixel) indexed by j, and modulate its phase ϕ_j from 0 to 2π. As this is an interference with a static reference, it will modulate sinusoidally the intensity at the pixel of interest k, as can be proven after isolating the contribution of this mode:

$$I_k^{\text{out}}(\phi_j) = |E_{/j}^{\text{out}} + T_j^k e^{i\phi_j}|^2 \tag{2.3}$$

$$= |E_{/j}^{\text{out}}|^2 + |T_j^k|^2 + 2|E_{/j}^{\text{out}} T_j^k| \cos(\phi_j - \alpha) \tag{2.4}$$

where $E_{/j}^{\text{out}}$ corresponds to the contribution to the output field of all the other modes with phase $\arg(E_{/j}^{\text{out}}) = \alpha$, and T_j^k is the component of T^k corresponding to the particular mode j.

We fix the phase ϕ_i to the value α maximizing $I_k^{\text{out}}(\phi_i)$, before moving to the next mode $i+1$. The intensity at the chosen pixel increases iteration after iteration, and

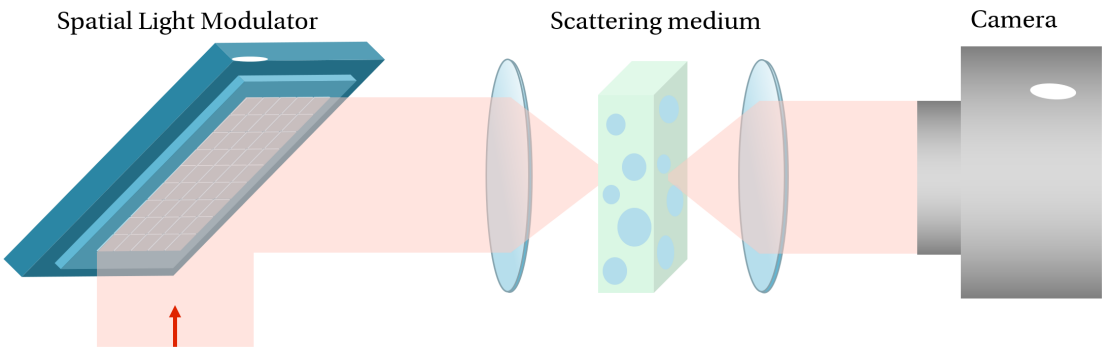


Figure 2.3: Canonical setup for wavefront shaping in complex media. Light from a laser is sent on a Spatial Light Modulator onto a scattering medium, a camera detects the output intensity. Playing with the incident wavefront with the SLM, it is possible to reverse the effect of scattering.

after a while, we obtain a sharp focal spot where intensity is much higher than the background speckle. This technique is an example of wavefront shaping, where the wavefront is *shaped* to counteract the effect of a complex or imperfect imaging system. Historically, wavefront shaping first appeared in adaptive optics methods for astronomy, where a deformable mirror is used to compensate the aberrrations introduced by the atmosphere.

A metric to quantify the quality of this focusing experiment is the Signal-to-Background Ratio (SBR). To compute it, we divide the intensity at the focal spot by the initial speckle intensity. This Signal-to-Background Ratio is proportional to the number of modes that are controlled with the SLM, assuming the complex material is mixing a much higher number of modes.

Recently, an experiment pushed this optimization strategy to millions of modes, achieving enhancement ratios in the hundred thousands [75]. This demonstrates the large number of modes in a typical complex scattering medium and the very large dimensionality of the operation performed optically. Furthermore, this optimization procedure is also possible for fluorescence microscopy, when a single fluorescent target is present inside the scattering medium. By maximizing the total recorded fluorescence intensity, we will also maximize the amount of excitation light onto the target and focus on it.

2.2.2 Imaging with the memory effect

Focusing light does not form an image but at least it constrains the signal to come from a single spatial position. To recover an image, one needs to retrieve information on different spatial positions. An interesting property towards imaging is the memory effect. Even though scattering is complex, some correlations may still be present. For instance, after the previous optimization for focusing, tilting slightly the obtained wavefront (i.e. adding a phase ramp to the field, the phase of an off-axis plane wave) shifts the position of the focus away from the medium. More generally, the same operation on any incident wavefront shifts the speckle observed on the camera. This happens only in a limited range, that is the memory effect, and washes out beyond it

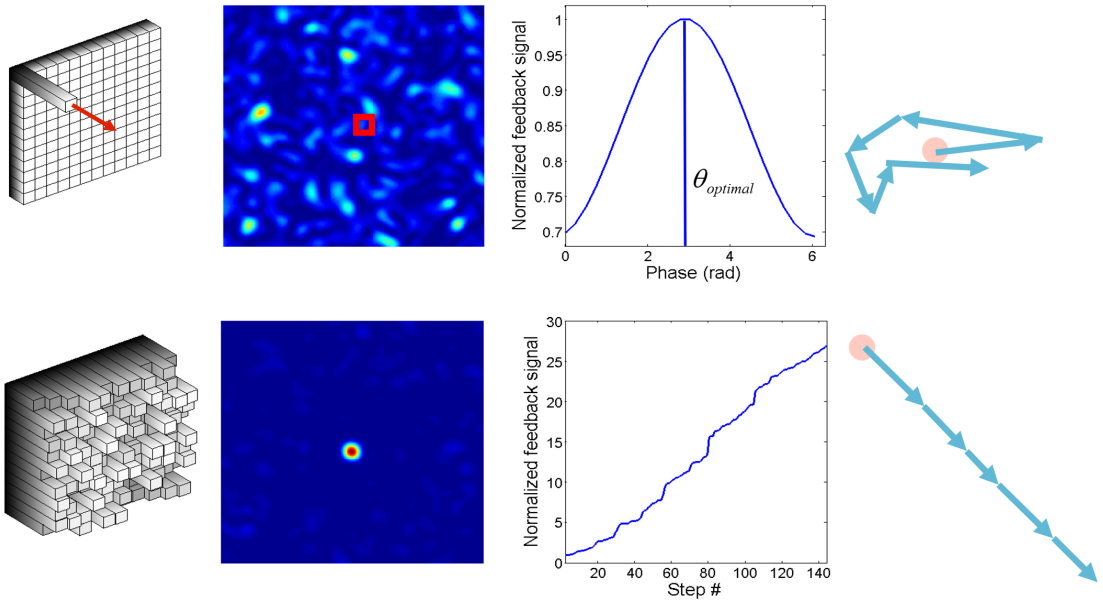
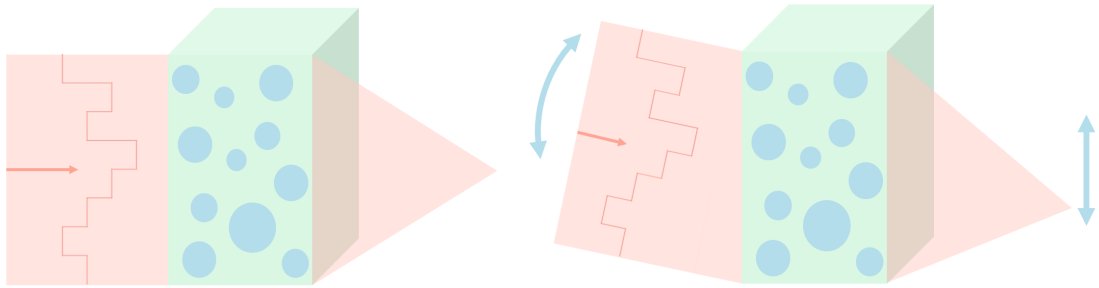


Figure 2.4: Focusing on a single point with an optimization strategy.
Top row: Optimization of a single mode. The intensity in one pixel (in red) varies with a sinusoidal law with the phase modulation. We observe a speckle because it results from a random interference pattern, contributions from each mode (in blue) have a random phase.
Bottom row: Focusing light after optimization. From left to right: The wavefront on the SLM after optimization, the corresponding camera image, the increase in intensity during optimization at the pixel of interest, optimization creates a constructive interference at the focus point.



1. Focus light 2. Scan the focus

Figure 2.5: Retrieving images with the memory effect. One first needs to focus light to a single point, then this focus can be scanned around the focus position by tilting the incident wavefront using the SLM. This is possible in a limited region around the focus position, the memory effect range.

[76], [77]. As the thickness of the complex medium increases, these correlations are reduced, and the memory effect range diminishes which limits the imaging field-of-view.

Nevertheless, this residual correlation is still present in practice and it has already been used for deep brain imaging. For example, one can focus excitation light inside a complex medium and raster scan a region around this focus by tilting the wavefront [78]. The measured fluorescence intensity will thus provide us information about the density of fluorescent molecules at each position of the focus, enabling the reconstruction of an image. However, as we lose the focus when we go out of the memory effect range, the achievable field-of-view is limited, and this is an important limitation in practice to overcome.

2.2.3 Transmission Matrix measurement

The previous technique is limited to the neighborhood of one camera pixel. To extend wavefront shaping to multiple output pixels, we can rather measure the Transmission Matrix (TM) of the complex optical system. Already introduced in the introductory chapter, the TM defines the linear operator that describes the electric field propagation from the SLM plane to the camera plane [36]. To measure it, one can send a basis of input vectors on the SLM and measure the corresponding camera images.

However, it is important to note that cameras only record intensities, i.e. the amplitude square of a complex-valued electric field (see Eq. (2.1)). Since a complex number is typically described by its amplitude and phase, phase information is lost although it is important in certain imaging settings. To solve this problem, one may use a reference wave (external or internal) and send for each input mode multiple images with different phase shifts. Using the interferences with the reference, one can derive the expressions to retrieve the original phase and reconstruct the Transmission Matrix.

Let us describe how to find the Transmission Matrix column $T_j \in \mathbb{C}^d$ corresponding to one mode j. Assuming the reference field (internal or external) is $E^{\text{ref}} \in \mathbb{R}_+^D$ is

real positive, we send K images with regular phase shifts (generalizing the formulas for $K = 4$ from [36]):

$$I^k = |E^{\text{ref}} + T_j e^{2ik\pi/K}|^2 \tag{2.5}$$

$$= \left(E^{\text{ref}}\right)^2 + |T_j|^2 + 2E^{\text{ref}}|T_j| \cos\left(\alpha_j + \frac{2ik\pi}{K}\right) \tag{2.6}$$

The modulus $|T_j|$ and phase $\alpha_j = \arg(T_j)$ are retrieved with:

$$\begin{cases} |T_j|^2 = \frac{1}{2K(E^{\text{ref}})^2} \sum_k \left(I_k - \frac{1}{K}\sum_l I_l\right)^2 \\ \cos\alpha_j = \frac{1}{KE^{\text{ref}}|T_j|} \sum_k I^k \cos\left(\frac{2ik\pi}{K}\right) \\ \sin\alpha_j = \frac{1}{KE^{\text{ref}}|T_j|} \sum_k I^k \sin\left(\frac{2ik\pi}{K}\right) \end{cases} \tag{2.7}$$

The previous equations are obtained by developing the right-hand side with trigonometric identities, using in particular the following one, valid for all $\alpha_j \in [0, 2\pi]$:

$$\frac{1}{K}\sum_k \cos^2\left(\alpha_j + \frac{2ik\pi}{K}\right) = \frac{1}{2} \tag{2.8}$$

2.2.4 Focusing light using phase conjugation

The knowledge of the TM enables us to focus light on every camera pixel very easily. Thanks to multiple scattering, each component of the TM is drawn following a complex gaussian distribution. The intensity at a camera pixel k with a flat wavefront $E^{\text{in}} = 1$ is:

$$I^k = \left|\sum_j T_j^k\right|^2 \tag{2.9}$$

Thus a speckle corresponds to a random interference between many phasors T_j^k with random amplitude and phase.

Focusing light on this pixel is very simple: the phase of each SLM pixel ϕ_j is set to $-\arg T_j^k$. This method called phase conjugation aligns all the phasors on the real axis to obtain a maximally constructive interference, as shown in Fig. 2.2.1:

$$I_{\max}^k = \left(\sum_j \left|T_j^k\right|\right)^2 \tag{2.10}$$

Compared to the previous optimization technique, phase conjugation allows to focus on all the camera pixel one by one, whereas we would need to reoptimize for each new focus position. The Transmission Matrix measurement is a powerful and comprehensive tool to characterize the scattering process in a complex optical material.

We would like to mention that phase conjugation is an example of phase-constrained optimizations: how to optimize a certain metric (I^k here) knowing that E^{in} is a phase-only vector with constant amplitude. In wavefront shaping, other optimization metrics have been studied such as the total intensity in a region [79], the distribution of

intensities [80], or how to obtain a given intensity image on the camera [36]. Constrained optimization may exhibit interesting and non-trivial results: for example with a ± 1 constraint, we know that minimizing the energy of a spin-glass configuration is an NP hard problem.

2.3 Towards non-invasive fluorescence imaging

Even though these historical experiments demonstrate important concepts about how to understand and work with complex scattering media, they are not suited for non-invasive imaging. In both the optimization and TM measurement experiments, the SLM and camera are at both sides of the scattering medium. However, in real-life settings, we would like to image objects hidden behind or inside a scattering medium. As such, we only have access to one side of the complex medium. This makes the problem more challenging and for instance, no transmission matrices have been measured beforehand in a non-invasive setting.

2.3.1 The epi-fluorescence configuration

We will thus stick with the well-established epi-fluorescence configuration. In this scheme, the excitation beam coming from a laser arrives on the thick biological sample to image, it excites some fluorescent molecules that reemit fluorescent light at other wavelengths, imaged on a sensor using a dichroic mirror (reflecting one wavelength and transmitting another). The sensor can be a bucket detector, or single-pixel camera which provides very sensitive measurements for low-intensity signals, or a conventional high-resolution camera. On top of this conventional setting, we introduce an LCoS SLM to perform wavefront shaping on the excitation beam. Hence, in this epi-fluorescence configuration, both the SLM and the sensor are on the same side of the scattering medium.

This defines an interesting computational problem. We know what phase pattern is displayed on the SLM, as we control it, we know what is the camera image, as we measure it. These are the information to work with, and from them, we would like to characterize the scattering medium in order to image a fluorescent object hidden behind. Hence we know the input and the output of our optical system, and would like to recover information about what happens in the black box in the middle. As we will show later, this inverse problem corresponds to a large-scale non-linear optimization.

2.3.2 Other techniques for deep imaging

We have introduced in this chapter wavefront shaping and the epi-fluorescence configuration that will be explored in the next chapters for linear fluorescence. Before moving to these works, let us present other approaches for deep imaging.

First, many different strategies have been proposed to focus light inside complex scattering media (a focus that may then be scanned for imaging):

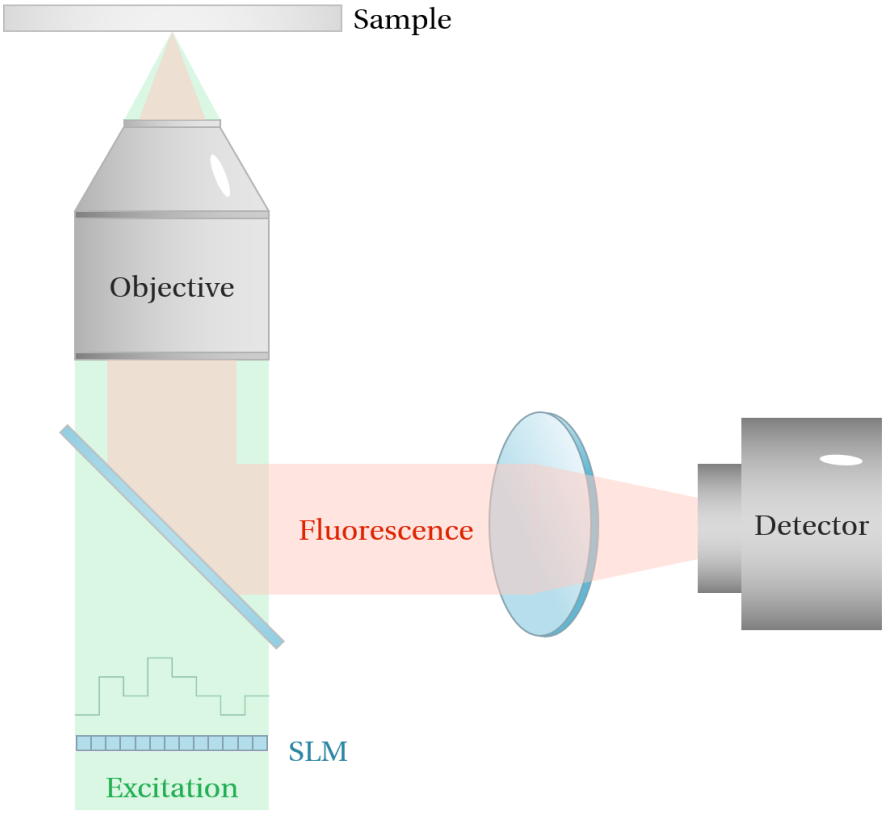


Figure 2.6: Epi-fluorescence configuration with wavefront shaping. We introduce a Spatial Light Modulator for the excitation beam to push wavefront shaping to this non-invasive configuration. Both the SLM and the camera are on the same side of the sample, corresponding to a non-invasive setting which can be replicated for real-life experiments.

- **Non-linear fluorescence:** When the emitted fluorescence is proportional to the square of the excitation intensity, in the same epi-fluorescence configuration, one simply has to optimize the total collected fluorescence intensity [76], [81]. Because the total excitation power is constant, this metric is optimized when all the excitation intensity is focused on a single target. Such an approach has then been optimized to obtain a focus in less than a second [78], opening up new applications in biological imaging. However, this approach requires a pulsed laser, a very fast Spatial Light Modulator, and dedicated electronics.

- **Guide-star:** A particle is implanted at a position of interest to focus on. It may be a nanoparticle using a process called Second Harmonic Generation [82] or a magnetic particle [83]. From these implanted point sources, focusing light back on them is obtained with a method called Digital Optical Phase Conjugation (DOPC) [84], similar to time-reversal for optical waves.

- **With acoustics:** As acoustics penetrates deeper without scattering into biological tissues, it may be combined with optics either with photo-acoustics [85], [86] or acousto-optics [87]–[90].

Finally, a surprising computational technique is also to retrieve objects from their autocorrelation [91]. It requires a spatially-incoherent object inside a memory effect range, and reconstructs it using a phase retrieval algorithm. This technique is very fast as it can be performed using a single-shot, but only works for thin scattering media because of the memory effect requirement.

All these innovative techniques to push the limit of the penetration depth require complex experimental setups. We choose a different point of view here, where we keep a very simple epi-fluorescence configuration with an SLM, and propose to study the computational problem that arises. Two different cases will be investigated depending on the detector we use: a single-pixel detector in Chapter 3 and a high-resolution camera in Chapter 4. In both cases, we design specific algorithms to characterize the scattering medium and focus light, paving the way towards fluorescence imaging at depth.

CHAPTER 3

Spectral methods for deep microscopy

3.1 Definitions

3.1.1 Experimental background

[1] We stick to the previously-introduced epi-fluorescence configuration in this chapter, but replace the camera with a bucket detector which registers the spatially integrated intensity. This single pixel detector is a very sensitive device to capture intensities. It is often used in low-luminosity setting, i.e. when the number of photons to be detected is low. Thanks to its very high photon efficiency, low noise, and high speed, such a detector is commonplace in optical experiments, from wavefront shaping experiments [92] to single-pixel cameras [93].

However, a single pixel camera provides a limited amount of information, since we only have access to one scalar number per measurement, the total fluorescence intensity in our case. We will show here how advanced computational tools enable us to still recover information from these measurements. In the next chapter, we will demonstrate how to leverage the additional information that one can get by placing a camera, when the number of captured photons is higher.

As stated previously, it is possible to focus light on a single fluorescent target inside a scattering medium [92], by optimizing the total collected fluorescence intensity. However, this proof-of-concept experiment may be too limited to be used in real-life settings due to the assumption of a single fluorescent target. In a real life setting, we may not know the number of targets present in the sample. The previous optimization strategy will not send light on a single target but usually on several of them [76]. Since light focusing is not possible in the presence of multiple targets, imaging at depth is not possible.

We would like to push this approach to the case of multiple fluorescent targets (Fig. 3.1). In constrast with previous optimization techniques, we choose another strategy: send random SLM patterns, capture the total fluorescence intensities on the bucket detector, and analyze this data computationally. Thus, instead of optimizing, we send random input patterns as a generic way to probe a complex system. This will allow us to count the number of targets and focus light on each of them.

[1] This project was originally an idea with preliminary numerical results by Prof. Sylvain Gigan, and I worked on it during the first year of my PhD.

I would like to thank Prof. Romain Couillet (GIPSA Grenoble) who hosted me during one week where I learned many concepts about Random Matrix Theory and Leonie Muggenthaler (ETH Zürich) who participated in many of the early discussions on this project.

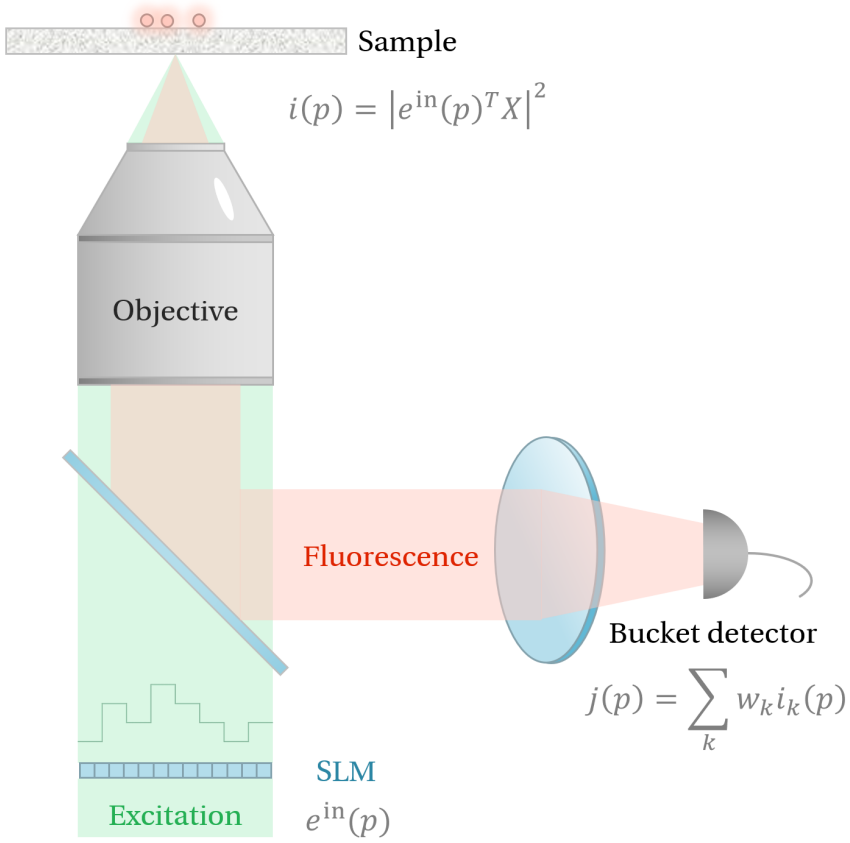


Figure 3.1: Non-invasive imaging configuration with an SLM and a bucket detector. This corresponds to the previously-introduced epifluorescence configuration. From the SLM patterns $e^{\text{in}}(p)$ and the total fluorescence intensity $j(p)$, one would like to retrieve the transmission matrix X.

3.1.2 Forward model

In this subsection, we introduce the mathematical formalism to describe this experimental setting. We display n random input patterns on the SLM. These patterns modulate the excitation wavefront, that propagates through the scattering medium to the plane of the K fluorescent targets and gets scattered along the way. Each fluorescent target is thus activated with varying excitation intensity, and will respond accordingly. This fluorescence intensity then propagates out of the medium and is detected on the bucket detector.

Let $e^{\text{in}}(p) \in \mathbb{C}^d$ be the input excitation electric fields of dimension d, defined by:

$$e_k^{\text{in}}(p) = e^{i\phi_k(p)} \tag{3.1}$$

with $\phi_k \sim \mathcal{U}(0, 2\pi)$ for $k = 1, \dots, d$ and $p = 1, \dots, n$.

We introduce a Transmission Matrix $X \in \mathbb{C}^{d \times K}$ to describe the complex propagation of the excitation light. The excitation intensity on each target is given by:

$$i(p) = \left|e^{\text{out}}(p)\right|^2 = \left|e^{\text{in}}(p)^\top X\right|^2 \tag{3.2}$$

for every pattern we send indexed by $p = 1, \dots, n$. For notation conciseness later, we introduce $x_k \in \mathbb{C}^d$ the k-th column of X, for $k = 1, \dots, K$.

The response of each fluorescent target is proportional to the excitation intensity with weights w_k for $k = 1, \dots, K$. These coefficient quantify the amount of fluorescence emitted by each target, that may vary due to size, concentration of fluorophores, bleaching, or other experimental parameters. We can write the total collected fluorescence intensity as:

$$j(p) = \sum_k w_k i_k(p) \tag{3.3}$$

for $p = 1, \dots, n$. We would like to stress once more that this total fluorescence intensity only corresponds to a scalar value, the information that we measure is quite limited.

Thus, we have the following forward model:

$$j(p) = \left|e^{\text{in}}(p)^\top X\right|^2 w = \sum_k w_k \left|e^{\text{in}}(p)^\top x_k\right|^2 \tag{3.4}$$

From these measured intensities $j(p)$ and the knowledge of the input field $e^{\text{in}}(p)$ for $p = 1, \dots, n$, we would like to retrieve the Transmission Matrix X to characterize scattering in our system. It would for example enable light focusing on each target individually with phase conjugation, enabling raster scan near each target for imaging in the memory effect range.

Before moving to the reconstruction algorithms, there are two simplifying hypothesis important to mention. First, fluorescence light also gets scattered on its way to the sensor. However, since we collect the total intensity, we have neglected this scattering process. It amounts to a decrease of the total detected fluorescence intensity by a factor, that should be approximately constant for all the fluorescence targets. The transmissivity of a multiple scattering medium decreases linearly with the thickness of the medium.

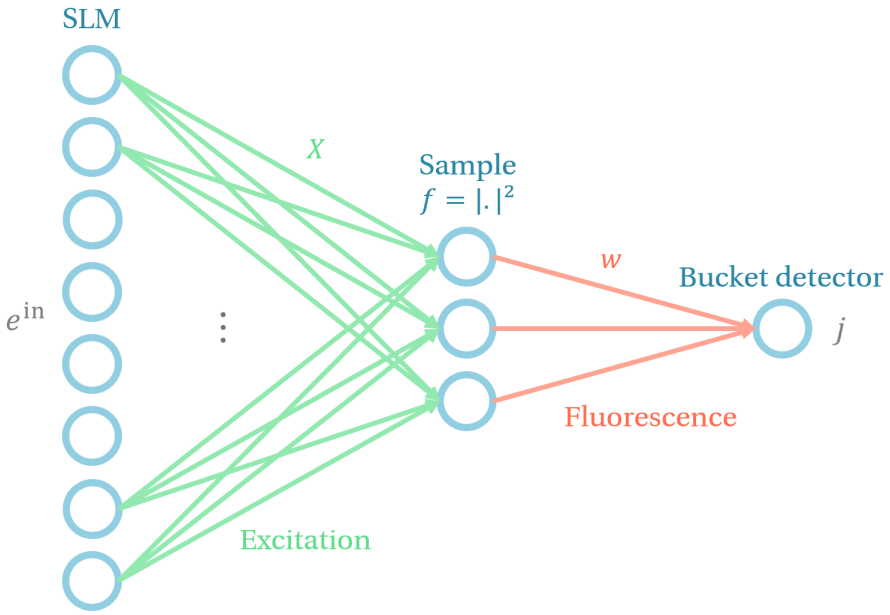


Figure 3.2: Two-layer neural network corresponding to our computational problem. From a set of input-output data ($E^{in}(p)$ and $j(p)$), one would like to retrieve the weights in this network.

Moreover, we have set the output TM dimension to K, to match the number of fluorescent targets. Indeed, the output dimension of X corresponds to the number of spatial positions of interest in the object plane. By doing so, we do not describe the electric field propagation outside the positions of the targets, which is natural since we do not collect any information about it in the backreflected fluorescence intensity.

This inverse problem corresponds to Multiplexed Phase Retrieval, a challenging variant of Phase Retrieval that we introduce in the following subsections. It is also interesting to relate this problem with a two-layer neural network as shown in Fig. 3.2. As the techniques presented are very far from the gradient descent approaches to train neural networks, they show the large variety of possible strategies to solve this non-linear optimization problem. Interpreting reconstruction problems in optics as neural networks to train has been a fruitful approach recently [94], with applications in tomography [95] and phase microscopy [96].

3.2 Reconstruction algorithm

3.2.1 Definitions

We first define the famous computational problem that is Phase Retrieval (PR):

Definition 1 (Phase Retrieval)**.** Let $y \in \mathbb{R}_+^n$ and $A \in \mathbb{C}^{n\times d}$. Phase Retrieval aims at solving for $x \in \mathbb{C}^d$ in the following equation:

$$y = |Ax|^2 \tag{3.5}$$

Due to the modulus operator, the phase of Ax is lost, which makes this equation much harder to solve than a classical linear equation $y = Ax$. This equation arises in many settings in physics because only energies or intensities can be measured, defined as the modulus square of the field. For example, PR is ubiquitous in computational imaging and arises in various applications such as coherent diffraction imaging, X-ray crystallography, ptychography, and Fourier ptychography. More insights about the history of phase retrieval and the vast array of computational methods to solve it are provided in Appendix B.

We now define Multiplexed Phase Retrieval (MPR), the problem to solve in our particular setting:

Definition 2 (Multiplexed Phase Retrieval)**.** Let $y \in \mathbb{R}_+^n$, $A \in \mathbb{C}^{n\times d}$, and $w \in \mathbb{R}_+^K$. Multiplexed Phase Retrieval aims at solving for $X = [x_k]_k \in \mathbb{C}^{d\times K}$ in the following equation:

$$y = |AX|^2 w = \sum_k w_k |Ax_k|^2 \tag{3.6}$$

If we knew each intensity value in the sum, we would have K parallel Phase Retrieval problems to solve. However, they are multiplexed here and we only measure a linear combination of them.

MPR corresponds to Eq. (3.4) with $y = j$ and $A = \left[e^{\text{in}}(p)^\top\right]_p$. More generally, it may arise whenever intensities from separate sources do not interfere (incoherent process) but mix in a single scalar value, for example in multispectral imaging or photoacoustic imaging.

3.2.2 Spectral method for Phase Retrieval

Phase Retrieval as a non-linear equation represents a challenging problem to solve. Thanks to its physical relevance, many strategies have been proposed, from gradient-based techniques, alternating projections, to convex relaxations. They are described in more details in Appendix B.

We will focus here on a specific family of techniques called spectral methods. They provide an elegant way to obtain quickly an approximate solution, based on a Principal Component Analysis (PCA) of a particular matrix we are going to construct. PCA decomposes any Hermitian matrix $M \in \mathbb{C}^{m\times m}$ into a set of pairs of eigenvalues and eigenvectors $\{\lambda_j, u_j\} \in \mathbb{R} \times \mathbb{C}^m$ such that:

$$M = \sum_j \lambda_j u_j u_j^\dagger \tag{3.7}$$

sometimes written as:

$$M = USU^\dagger \tag{3.8}$$

with S a diagonal matrix defined by $s_{jj} = \lambda_j$ for $j = 1,\dots,m$, and $U = [u_j] \in \mathbb{C}^{m\times m}$ an orthogonal matrix. The PCA of a matrix is a very important tool to characterize this linear operator, for example the rank of the matrix M corresponds to the number of non-zero eigenvalues. We typically rank the eigenvalues from the largest to the smallest, to compare the contribution of each eigenvector in M.

In the first spectral method by [97] (a similar idea was proposed by [98]), they introduced the following weighted covariance matrix (denoting by $a_k^\dagger$ the k-th row of A):

$$Z = \frac{1}{n}\sum_k y_k a_k a_k^\dagger \tag{3.9}$$

and compute its leading eigenvector (the one corresponding to the largest eigenvalue). Since the distribution of eigenvalues of a matrix is often called its spectrum, this technique and variants have been called spectral methods.

The intuition behind this choice is that when x is correlated with a_k, then $y_k = \left|a_k^\dagger x\right|^2$ is larger. Thus, the sum in Z gives more importance to sampling vectors a_k aligned with x, and the leading eigenvector of Z correlates with x. This intuition can be formalized with the following theorem:

Theorem 1. *For Phase Retrieval, assuming A is a random i.i.d. matrix with $\mathbb{E}(A_{ij}) = 0$, $\mathbb{E}(|A_{ij}|^2) = 1$, and $\mathbb{E}(|A_{ij}|^4) = 2$, the expectation value of Z defined in Eq. 3.9 is:*

$$\mathbb{E}(Z) = I_d + xx^\dagger \tag{3.10}$$

Proof. We develop the sum in $Z = [Z_{ij}]_{ij} \in \mathbb{C}^{d\times d}$ to express it as a function of $x = [x_i]_i \mathbb{C}^d$ and $A = [A_{ij}] \in \mathbb{C}^{n\times d}$ only:

$$Z_{ij} = \frac{1}{n}\sum_k \left|\sum_l A_{kl}x_l\right|^2 A_{ki}^* A_{kj} \tag{3.11}$$

$$= \frac{1}{n}\sum_k \sum_l \sum_m A_{kl}x_l A_{km}^* x_m^* A_{ki}^* A_{kj} \tag{3.12}$$

$$= \sum_l \sum_m x_l x_m^* \frac{1}{n}\sum_k A_{kl}A_{km}^* A_{ki}^* A_{kj} \tag{3.13}$$

We then take the expectation value on the i.i.d. random variables A_{ij}. Since they have zero mean, there are many cases for which $\mathbb{E}(A_{kl}A_{km}^* A_{ki}^* A_{kj}) = 0$. If $i \neq j$, this happens when $l \neq i$ and $m \neq j$. We obtain in this case:

$$\mathbb{E}(Z_{ij}) = x_i x_j^* \mathbb{E}(|A_{ij}|^2)^2 \tag{3.14}$$

If $i = j$, $\mathbb{E}(A_{kl}A_{km}^* A_{ki}^* A_{kj}) \neq 0$ when $l = m$. The diagonal terms are thus:

$$\mathbb{E}(Z_{ii}) = |x_i|^2 \mathbb{E}(|A_{ij}|^4) + \sum_{l\neq i} |x_l|^2 \mathbb{E}(|A_{ij}|^2)^2 \tag{3.15}$$

Using the assumptions on the moments (which are valid for a complex gaussian random variable), we obtain the desired formula. □

As a result, when the number of samples $n \to \infty$, the leading eigenvector of Z will converge towards x. This comes from the concentration in Eq. 3.13 of each component of Z_{ij} towards its expected value. Note that this is only an asymptotic analysis and not a result for a finite n. In [97], they prove convergence with a sample complexity $n \geq c_0 d \log d$ for some constant $c_0 > 0$. This relatively poor sample complexity

comes from the weights y in Eq. 3.13 which are not bounded, thus they sometimes produce large outliers that affect the leading eigenvector of Z.

Interestingly, other preprocessing functions (all bounded above) have been proposed since this seminal work [99]–[102]. They improve the sample efficiency of the spectral method: recovery is obtained with a smaller number of samples $n \sim O(d)$. We will not investigate them in this chapter as their approach cannot be easily transposed to Multiplexed Phase Retrieval.

3.2.3 Spectral method for Multiplexed Phase Retrieval

The previous spectral method can be adapted to Multiplexed Phase Retrieval. Using the definition of the matrix Z as in Eq. 3.9, we obtain a similar result. The proof of this theorem is based on the same expansion as in the non-multiplexed case.

Theorem 2. *For Multiplexed Phase Retrieval, assuming the same conditions on the moments of A, the expectation value of Z is:*

$$\mathbb{E}(Z) = I_d + \sum_k w_k x_k x_k^\dagger \tag{3.16}$$

For the leading K eigenvectors of Z will converge towards the different x_k as $n \to \infty$, two assumptions are necessary. First, all the w_k needs to be positive and pairwise different. If for two indices i and j, the corresponding weights w_i and w_j are equal, it is not possible to distinguish x_i and x_j and the algorithm will only return linear combinations of them. Second, X needs to be an orthogonal matrix, which is approximately true for a random X when d is large.

This spectral method is useful for its flexibility. We usually do not know in advance the number of hidden targets and the relative weights w_k. Looking at the distribution of eigenvalues of Z for very large values of n, it is possible to count the number of fluorescent targets K by looking at how many eigenvalues are significantly larger than 1. Then, it is also possible to evaluate the coefficients w_k from the corresponding eigenvalues. Hence, this spectral method requires less a priori information on the system to solve MPR as gradient descent for example.

It is important not to forget that the previous derivation is based on the expectation, i.e. what happens when n goes to infinity. Since the problem is more challenging, one could expect that a higher number of samples n is required. We will describe empirically the finite size effects we observe in the experiments.

3.3 Results

3.3.1 Experimental setting

We prove that such an approach works experimentally with a very simplified proof-of-concept experiment (Fig. 3.3): excitation light from a laser (Coherent Sapphire 532-50 CW) is modulated by a Spatial Light Modulator (SLM, Holoeye Pluto-2 NIR), then propagates through a layer of white paint where it gets scattered multiple times.

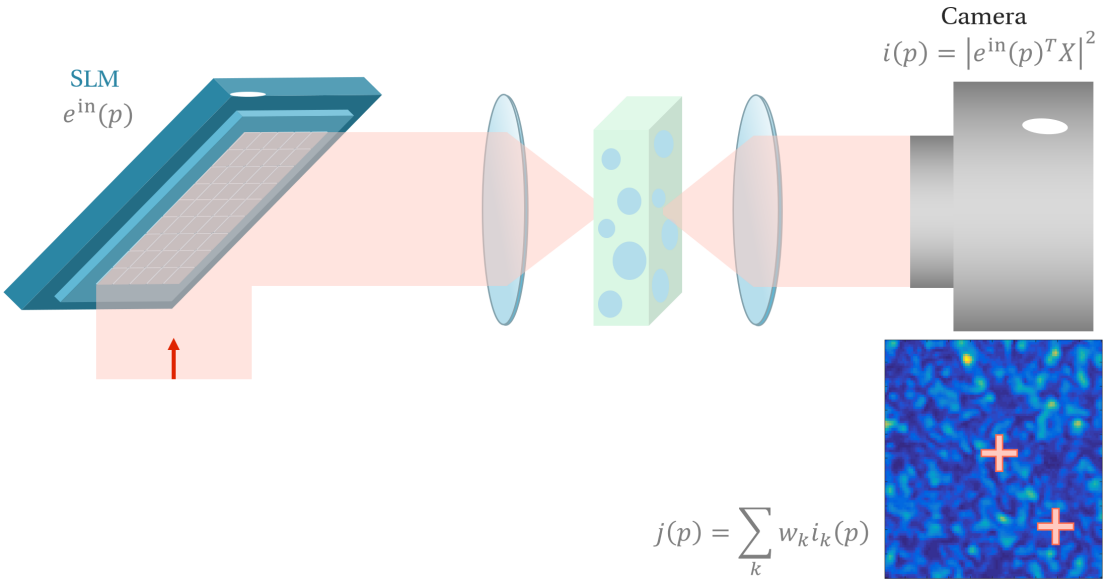


Figure 3.3: Proof-of-concept experimental setup. We reuse the canonical wavefront shaping experiment of Chapter 1, as a simplified experiment without fluorescence which still obeys the same equation. The total fluorescence intensity is emulated by summing intensities at two different positions in the camera image.

We place a camera (Basler acA1920 - 40um) after the scattering medium and observe a speckle pattern. We choose two points in the camera image and sum their intensities with coefficients $w_1 = 1$ and $w_2 = 0.7$, mimicking two beads with different quantum efficiencies. Thus, it corresponds exactly to the formalism above and we use only this multiplexed information to recover the TM to these two points. Note that in this artificial experiment, there is no real fluorescence target, equations are the same, but the experimental setting is simplified.

3.3.2 Reconstruction and focusing

We send n random SLM patterns and record the corresponding total intensities. From this information, we construct the matrix Z and diagonalize it to recover its two leading eigenvectors. They will provide estimates of x_1 and x_2, the corresponding lines of the TM for the two output spatial positions, and we use these to try to focus light on each target, using phase conjugation. If light has successfully been focused, it means that reconstruction of x_1 and x_2 has been successful. This technique allows us to verify the quality of the reconstruction experimentally, without the knowledge of the target values of x_1 and x_2.

We present the experimental results in Fig. 3.4. From the initial excitation speckle, with $n = 10{,}000$ patterns, we are able to focus on the first and the second target. This proves that with sufficiently many samples, both x_1 and x_2 can be successfully reconstructed.

Fig. 3.4 also presents the Signal-To-Background ratio as the number of patterns sent vary from 100 to 10,000. We observe that the quality of the reconstruction increases with the number of patterns sent. When the number of patterns is small,

not enough information has been acquired and reconstruction is impossible. There seems to be a minimal number of patterns after which reconstruction gets better and better.

This necessary number of measurements increases with the dimension of the SLM d. This parameter is chosen beforehand and corresponds to the resolution of the random patterns displayed on the SLM. The higher the dimension, the larger the quality of the focus, as we control more input modes, but at the cost of a larger characterization time. Here, $d = 256$ which is a typical order of magnitude for wavefront shaping in complex media [60], [78], and the required number of patterns is of the order of $n = 5,000$, which is about 20 times the SLM dimension d. Compared to an oversampling ratio usually around 4 for Phase Retrieval, this shows the complexity of the Multiplexed Phase Retrieval problem. This oversampling ratio is only obtained through a numerical study at finite dimensions and not from an explicit theory detailing the phase transition behavior, which is sufficient in practice since $d \sim 100$ already provides a sufficiently good focus of the excitation light for further applications.

Code on this application of spectral methods for Multiplexed Phase Retrieval is available at `https://github.com/jon-dong/multiplexed_phase_retrieval`.

3.3.3 Eigenvalue distribution and sample complexity

We now present a numerical investigation of the eigenvalue distribution of Z in Fig. 3.5, for a few different values of n. The theoretical results presented before are only asymptotic, due to the finite number of measurements n.

For a finite number of samples n, Z is not equal to its expected value $\mathbb{E}(Z)$ but fluctuates around it. This produces a continuous distribution of eigenvalues, a common observation with covariance matrices in Random Matrix Theory.

We see that as the number of samples increases, a first then a second eigenvalue come out of a bulk distribution. As long as these spikes are not appearing, the leading eigenvectors are not informative. This may suggest that a phase transition is happening here: for a small number of measurements, there is a phase where the leading eigenvectors do not carry any information about the parameters to estimate, and with more measurements, another phase where the spectral method performs well. Further studies would be required to relate this behavior to other results in Phase Retrieval, detailed in Appendix B.

This sample complexity is roughly proportional to the number of unknowns to recover, which corresponds to the SLM dimension $d = 1,024$ in the numerical results of this subsection. This means that if we want faster experiments requiring less patterns, one could decrease the number of pixels of an SLM image. This change comes at the cost of a decreased SBR of each focus, as we control less modes for wavefront shaping.

As a last remark, this spectral method also allows us to count the number of fluorescent targets K, by looking at the eigenvalue distribution of Z for a very large number of random patterns n. The number of spikes out of the bulk distribution corresponds to the number of detected targets.

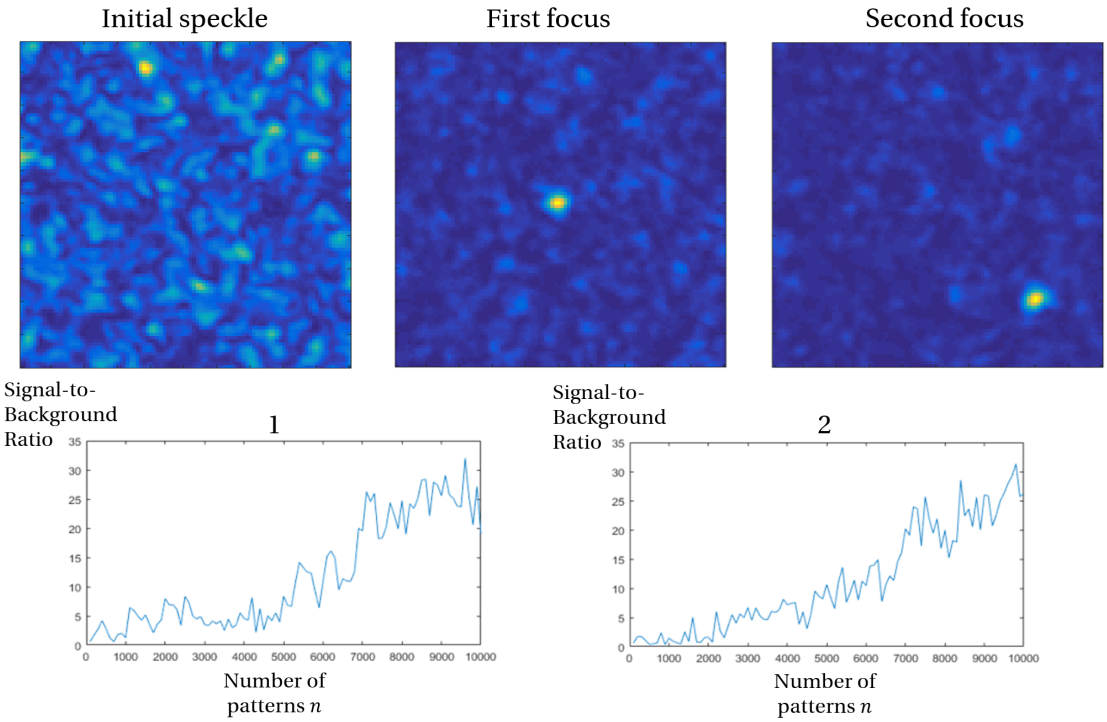


Figure 3.4: Experimentally focusing light using spectral methods to solve Multiplexed Phase Retrieval. (Top row) From the initial speckle pattern on the camera (left) and after sending $n = 10,000$ random SLM patterns, we are able to retrieve the Transmission Matrix and focus on the first (middle) and second (right) fluorescent targets. (Bottom row) We observe that the quality of the focii increases with the number of patterns n by evaluating the Signal-to-Background Ratio (defined as the peak intensity over mean intensity of the initial speckle), for the first focus (left) and the second (right).

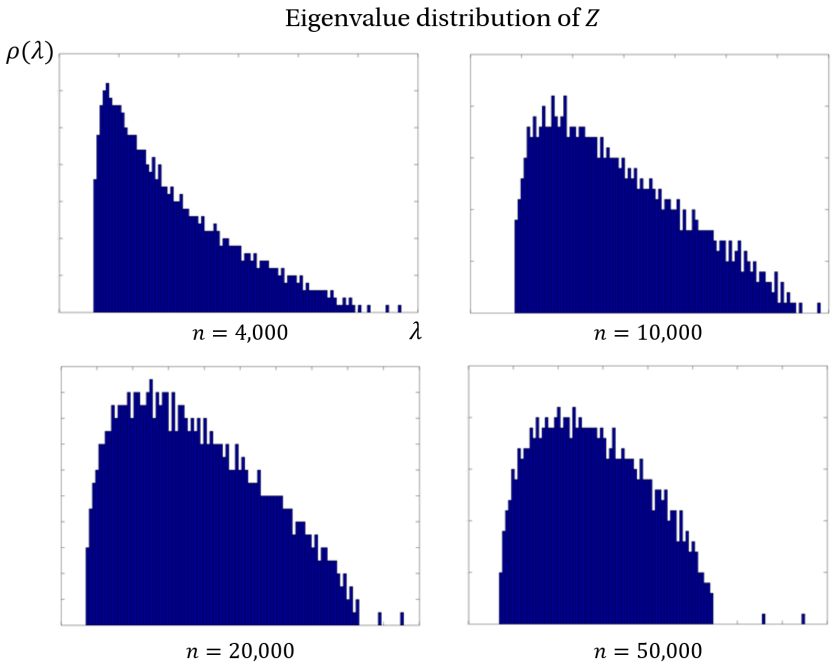


Figure 3.5: Spectra as a function of the number of SLM patterns (numerical results). The eigenvalue distribution of Z forms a bulk distribution with two eigenvalues that appear when the n is large enough. Each of these eigenvalues correspond to one fluorescent target.

3.4 To go further

3.4.1 High-speed implementation

The speed of the wavefront shaping experiment is crucial in real-life settings due to the fast decorrelation time of biological tissue [103], [104]. Due to the limited speed of our high-resolution camera, this proof-of-concept experiment is definitely not suited for real-time imaging. It would require focusing in less than a second to be able to compensate for the dynamic changes of the biological tissue, on timescales that range between 1 ms and 1 s [104].

Capturing $n = 10,000$ images at 60 Hz took a little more than two hours, which is why we resorted to a static medium in this proof-of-concept. Through the synchronization of the SLM with the single-pixel detector, we can potentially send patterns at a few kHz to perform this experiment in a few seconds. For example, a previous work in the group [78] implemented an optimization scheme in transmission with a frame rate of 16 kHz. Such a high acquisition rate would not be possible with a high-resolution camera due to a communication bandwidth bottleneck, which motivates the development of algorithms with a single-pixel detector such as this one. Moreover, at high speed the number of fluorescent photons to detect in a single frame may be low which further motivates the use of a bucket detector.

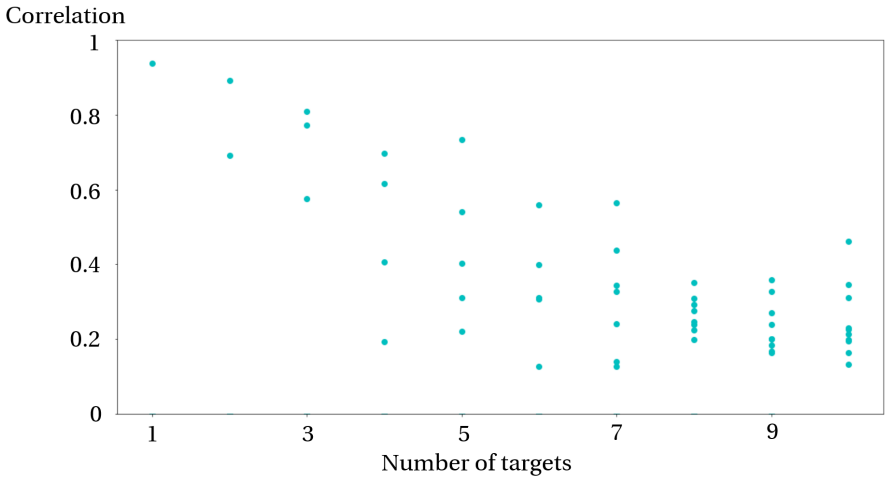


Figure 3.6: Correlations of the leading eigenvectors for different number of targets K. The experiment is repeated numerically for k from 1 to 10 and we observe that as the number of targets increases, the reconstruction becomes more difficult. Parameters: SLM dimension $d = 256$, number of samples $n = 10,000$, weights w linearly spaced between 0.5 and 1.

3.4.2 Non-invasive experimental validation

In a non-invasive experiment, there would be no-control to validate the quality of the focus. Other strategies need to be tested in order to validate the quality of the reconstruction in challenging conditions.

First, if it is possible to probe the medium with a large number of random SLM patterns, then the spikes coming out of the bulk distribution may be observed experimentally. Each spike would correspond to a focus position and would be distinct from the non-informative eigenvalues of the bulk distribution. This method would require the medium to be static while the random patterns are sent. Moreover, it would only work for sparse objects and would not be applicable for continuous objects.

On the other hand, we can use phase conjugation based on the reconstructed TM X, and the excitation light should be focused on the fluorescent targets, increasing the total fluorescence collected on the bucket detector. If that is not the case, we know that this row of the TM is not informative. This validation step would be useful to distinguish the frontier between particular eigenvectors and the ones which come from the bulk distribution of Z. On the other hand, this non-invasive validation does not verify if excitation is focused on a single target or several ones.

3.4.3 Study for larger number of targets

To investigate how this method scales with the number of targets, we have performed a numerical study with realistic experimental parameters. As we set the SLM dimension $d = 256$ and the total number of samples $n = 10,000$, we vary the number of targets k from 1 to 10 and compute the correlations between k leading eigenvectors

of Z and the target vectors in X. Results after correcting for shuffling are presented in Fig. 3.6.

We see that our method is quite successful for $k < 4$, but its performance quickly degrades as we increase the number of hidden targets. Indeed, adding more targets adds more random fluctuations in Z which prevents the spectral method from solving the MPR problem. Our method is thus only suited for very sparsely-labeled samples with very few number of fluorescent targets, showing the difficulty of Multiplexed Phase Retrieval. To increase the number of targets to be disentangled, two directions may be followed. To increase the oversampling ratio n/d, it is possible to increase the total number of patterns n or decrease the dimension d. While the value of n depends on the frame rate and the stability of the medium, d is a hyperparameter that we can arbitrarily set. A minimal value of d to obtain a minimal quality of the focii is around 100. Using two-photon fluorescence may increase the sparsity of the fluorescence excitation and allow reconstructions with more targets but a theory of spectral methods for this case still remains to be developed.

3.4.4 Other algorithms

We have presented a spectral method to disentangle mixed signals coming from different targets. Other algorithms may be investigated, for example the 2-layer neural network may also be solved using a gradient-descent approach. Another possibility is the use of Bayesian algorithms such as Approximate Message Passing [105].

All these other algorithms would be interesting to compare to, but the first would require to guess the number of hidden nodes k, while the second would even require to know the second layer of weights w, which would be irrealistic in an experimental setting.

3.5 Discussion

3.5.1 In imaging

In this chapter, we have shown how to use spectral methods to study Multiplexed Phase Retrieval, a computational problem that arises in fluorescence microscopy with a single-pixel camera. This technique enables light focusing on linear fluorescent targets after a scattering medium, even when several of them are present.

With a limited proof-of-concept experiment, this study remains far from practical implementations and has rather been developed with a deliberate focus on the computational side. The main limitations of this study for future applications are the high number of random patterns to send and the sparsity assumption that would require very specific fluorescence tagging.

We cannot get information about the positions of the focii since we only use a bucket detector and have no information to localize the focal points. This strategy may be useful for cases in which positions are not necessary, like optogenetics (using laser pulses to activate tagged channels and induce electric activity in specific neurons). More generally, this technique may be useful whenever signal from several

sources is multiplexed. For instance in photo-acoustics, the acoustic resolution limits our sensivity and it may be useful to disentangle the signal from several sources using such a technique.

3.5.2 In non-linear optimization

To improve the presented spectral method, other preprocessing functions may be introduced to increase the sample efficiency, inspired by the similar trend that happened with conventional phase retrieval. Since this work, results from a recent study [106] may help us reduce the number of samples to send for future implementations.

This work is a first example to show how the interplay between computational imaging and non-linear optimization is fruitful beyond the use of gradient-descent techniques. We have first modeled the optical system, characterized by a few unknown parameters, that we recover computationally benefitting from recent tools developed for Phase Retrieval. In the next section, we will show another example of this approach for deep linear fluorescence imaging.

CHAPTER 4

Double Transmission Matrix reconstruction

4.1 Definitions

4.1.1 Experimental background

[1]In this chapter, we continue our journey on computational approaches to retrieve information in the presence of scattering. More precisely, our main goal is to manipulate the illumination light such that it reveals the shape of a fluorescent object buried deep inside a scattering medium. We have seen how to use spectral methods to leverage the information on a bucket detector, that we now replace with a conventional camera, in the same epi-fluorescence configuration (Fig 4.1).

With this change, we have access to a wealth of additional information, from the single scalar value from a single-pixel camera to a high-resolution image. Instead of the total fluorescence intensity, we now have access to the fluorescent speckle emitted by the object. This change comes at a cost: acquisition speed must be reduced as we need more photons to reach the camera to form an image. However, this framework will enable more powerful reconstruction algorithms to tackle more complex objects.

Like previously, we still send random phase patterns on the SLM and would like to characterize our system. From this characterization, we focus light using phase conjugation and use the memory effect to recover images. Whereas we previously recovered a single Transmission Matrix T, here we are going to introduce two Transmission Matrices to describe the propagation of light to and from the sample.

4.1.2 Forward model

Similar to the previous project, we send n random phase patterns on the SLM $E^{\text{in}}(p) \in \mathbb{C}^d$ for $p = 1, \dots, n$. They propagate through the complex medium and the excitation at the positions of the fluorescent targets is

$$H(p) = |E^{\text{out}}(p)|^2 = |TE^{\text{in}}(p)|^2 \tag{4.1}$$

We suppose that the fluorescent object is made of N separate fluorescent targets, so that $T \in \mathbb{C}^{N \times d}$.

[1]This project was mainly carried out by Antoine Boniface. I contributed in the design of the approach and the algorithmic codes for reconstruction.

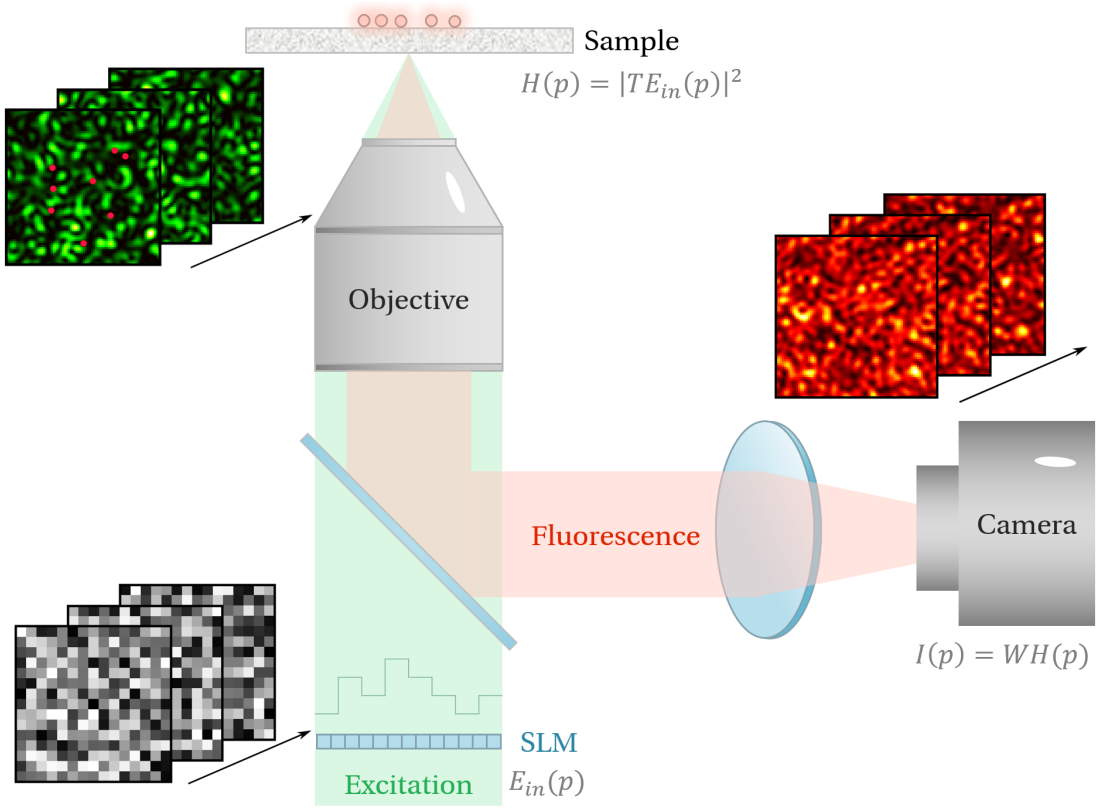


Figure 4.1: The experimental configuration for our double-transmission matrix model. We use an epi-fluorescence microscope with wavefront shaping, with a camera instead of a single-pixel detector compared to the Chapter 3. From the knowledge of the input wavefronts $E_{in}(p)$ and the fluorescence images $I(p)$, we would like to retrieve both T and W, the Transmission Matrices in and out of the scattering medium.

Each target emits fluorescence proportional to its excitation, and this fluorescent light gets scattered on its way out to the camera. This second scattering process is characterized by another Transmission Matrix W:

$$I(p) = W H(p) \tag{4.2}$$

with $W \in \mathbb{R}_+^{D\times N}$. This TM is quite different from the previously-introduced coherent TM T. Its components are not complex-valued, but all real and positive numbers. Indeed, each column of W describes the speckle intensity image generated by individual sources. Since the fluorescence signals emitted by all the targets are uncorrelated with a random time-varying phase, they sum up incoherently which produces a low contrast speckle pattern $I(p)$. W is thus an intensity Transmission Matrix, where a modulus square operator is applied element-wise on the coherent Transmission Matrix on the way back.

In the end, we measure camera images

$$I(p) = W|TE^{\text{in}}(p)|^2 \tag{4.3}$$

and would like to recover the two TMs $T \in \mathbb{C}^{N\times d}$ and $W \in \mathbb{R}^{D\times N}$ from the n pairs $\{I(p), E^{\text{in}}(p)\}$ for $p = 1,\dots,n$.

This computational problem can also be framed in a neural network approach: we want to find the weight matrices in a two-layer neural network. Like the previous chapter, we will see how to design an algorithmic pipeline to reconstruct these weights. For two reasons, we did not resort to plain gradient descent. First, this link was not explicit at first (this project originated from a casual discussion whether Non-negative Matrix Factorization could be useful in our setting). Moreover, we will show how to use the specialized Phase Retrieval algorithms to achieve convergence with a fewer number of captured images.

4.2 Reconstruction algorithms

4.2.1 A two-step reconstruction

The reconstruction of the 2 Transmission Matrices will be performed in 2 steps. First, the incoherent Transmission Matrix W is recovered using a matrix factorization approach, then H will be reconstructed by solving a Phase Retrieval problem. By decomposing the algorithmic pipeline in two distinct steps, we are able to interpret and validate the intermediate results, an important asset in any experimental approach.

4.2.2 Non-negative Matrix Factorization

Similar to the previous case with a bucket detector, we will make a sparsity assumption. The number of fluorescent targets N is supposed smaller than both the dimension of the camera D and the number of patterns we send n. This assumption is important and limits the complexity of the objects to be imaged, but we will see that this technique is actually quite robust and works with tens of fluorescent targets.

This sparsity assumption is reflected in the rank of the matrix formed by the camera images $I = [I(p)]_{p\in 1,\dots,n} \in \mathbb{R}_+^{D\times n}$. The signal on the camera comes from the fluorescence light emitted by the targets; each one of them generates a fluorescent speckle pattern, more or less intense depending on the excitation intensity each target receives. Thus, all the camera images are weighted sums of N individual speckle patterns. This means the matrix I is a rank-N matrix.

Hence, we leverage well-studied matrix factorization algorithms to decompose $I = WH$, a low-rank factorization with $W \in \mathbb{R}_+^{D\times N}$ and $H \in \mathbb{R}_+^{N\times n}$. W corresponds to the stack of individual speckle patterns for each target, while H describes how much each target is excited, for each random SLM pattern we send.

In particular, we use here a framework called Non-negative Matrix Factorization (NMF). It is based on the observation that both matrices W and H are non-negative, which makes the reconstruction more robust compared to other techniques, based for instance on Singular Value Decomposition. In practice, reconstruction is very robust and can be accurate even when the number of recorded images n is relatively small.

This principle of low-rank factorization has already been applied to imaging, to disentangle independent sources in videos. For example, neuron activations have been retrieved with this technique [107]. A recent work from our group [37] uses NMF to retrieve neuron activities from their speckle fingerprint.

Hence, in our case, we are able to obtain W and $H = |E^{\text{out}}|^2$. With this first step, we have already reconstructed one Transmission Matrix out of two.

4.2.3 Phase Retrieval

After the NMF, we still have the matrix T to reconstruct. The second TM resulting from the NMF, the factor matrix H, tells us how much excitation intensity each fluorescent target received. Hence, it is like a camera pixel was placed at the position of the hidden target. This link is made explicit in the forward model in Eq. (4.1).

Since we also know the SLM patterns $E^{\text{in}} \in \mathbb{C}^{d\times n}$, this boils down to a phase retrieval problem. As discussed in the previous chapter, we are in the well-characterized random setting as E^{in} are random phase patterns we send on the SLM. This well-understood phase retrieval problem is solved in parallel for each fluorescent target.

Many approaches have been proposed to solve this Phase Retrieval problem. For example, [108], [109] proposed Bayesian approaches to measure Transmission Matrices. We use here the spectral initialization from [101] followed by gradient descent steps. A more in-depth description of algorithms for Phase Retrieval is provided in Appendix B.

4.3 Results

4.3.1 Experimental setting

A continuous-wave laser (λ = 532 nm, Coherent Sapphire) is expanded on a phase-only MEMS SLM (Kilo-DM segmented, Boston Micromachines) with N_{SLM} = 1024

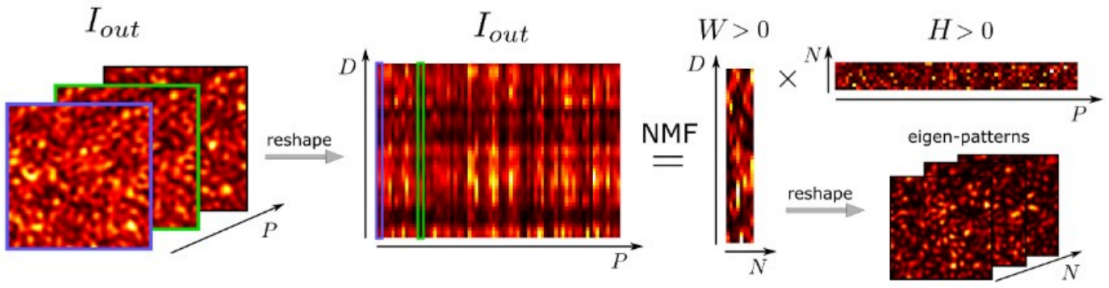


Figure 4.2: Applying Non-negative Matrix Factorization for signal demixing. This first step of the algorithmic pipeline exploits the low-rank and positivity assumptions in the measurements I_{out}, the matrix containing all the captured images. We obtain W the fluorescent fingerprints of each target, and H, how much excitation intensity each target receives. A reshaping operation is necessary to transform a 2D image into a vector of size D.

pixels. Once modulated, the beam is directed through the illumination objective (Zeiss W "Plan-Apochromat" 20×, NA 1.0) onto the sample containing the scattering medium and the fluorescent sample. We used ground glass (Thorlabs, DG10) or holographic diffusers (Newport 1 + Newport 10) with fluorescent beads (540/560 nm, Invitrogen FluoSpheres, size 1.0 μm), or one holographic diffuser (Newport 1) with fluorescent pollen seeds (Carolina, Mixed Pollen Grains Slide, w.m.).

The backscattered fluorescence is captured on a camera (sCMOS, Hamamatsu ORCA Flash), thanks to a dichroic mirror shortpass 550 nm (Thorlabs) and two additional filters (a 532 nm longpass from Semrock and a 533 notch from Thorlabs).

For control only, a second camera is placed behind the sample (Allied Vision, Manta) with a second microscope objective (Olympus "MPlan N" 20×, NA 0.4). This part of the setup is only used for passive monitoring of the excitation speckle.

4.3.2 Double TM reconstruction

We first show how the Non-negative Matrix Factorization is able to disentangle the fluorescent speckles coming from each target. The stack of images $I(p) \in \mathbb{R}^D$ (reshaping images into vectors) captured by the camera is concatenated form a matrix I. In Fig. 4.2, we show what is the output of the NMF on a simulated dataset. A reshaping operation is necessary to recover meaningful speckle images. In the end, the individual speckle patterns in W are more contrasted than the captured images I, as a sum of different uncorrelated speckle patterns reduces the contrast of the image. When applying it on experimental data, a preprocessing operation was needed to remove the envelope of the speckle.

In the algorithmic pipeline, there is only a single hyperparameter to tune, the rank r of the NMF. In principle, r should correspond to the number of targets N in the system, but this number is unknown in our non-invasive configuration. To estimate it, an order of magnitude is obtained by performing NMF for different ranks and looking at $\|I - WH\|^2$. As a rule of thumb, it is usually better to overestimate the number of targets.

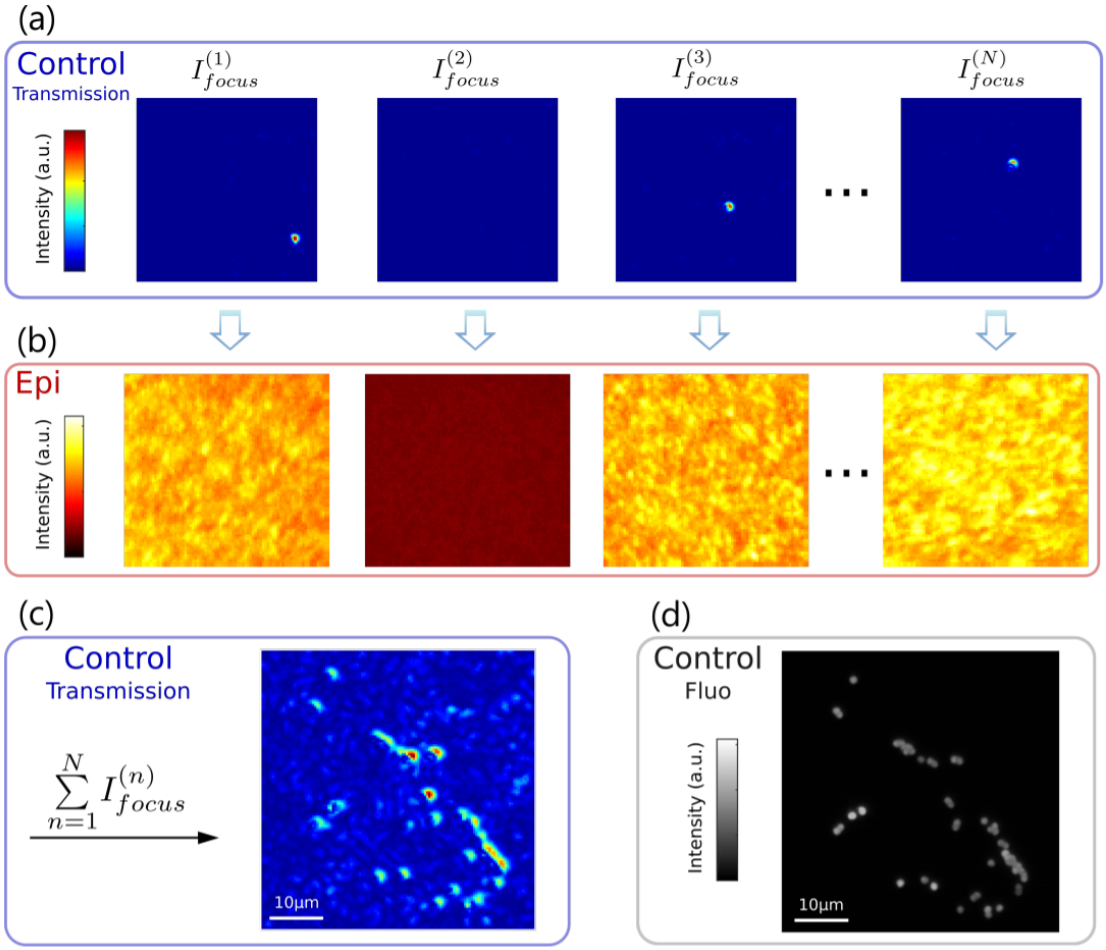


Figure 4.3: Focusing after the double TM reconstruction. Once the coherent TM has been recovered with phase retrieval, we focus light back on each target using phase conjugation. When the reconstruction is successful, we focus light back on a single target (a) and the variance of the collected fluorescence in epi is large (b). Sometimes the reconstruction is not successful and we can detect and remove these points non-invasively. For control, we check that the positions of the focii (c) match a control image after removing the scatterer (d).

From the recovered matrix H after the NMF, we can reconstruct the matrix T with phase retrieval, and focus light on each target using phase conjugation. This operation is repeated for each mode returned by the NMF. In Fig. 4.3, we show that focusing is successful in most cases. When it is not, it is possible to detect it non-invasively, as the epi-fluorescent image has a much lower variance. Thus, a general strategy consists in overestimating the rank and remove the superfluous eigenvectors in a post-processing step after trying to focus excitation back on the targets. For control, we also verify that most of the sample can be focused on.

In the end, our algorithmic pipeline is able to fully solve this computational problem. We have managed to reconstruct both matrices T and W, characterizing the scattering process affecting both excitation and fluorescent light. It is important not to forget that a model is often imperfect and it is important to design algorithms robust against noise and experimental imperfections.

Code to simulate and reconstruct this double-TM model is available at `https://github.com/laboGigan/NMF_PR`.

4.3.3 Imaging with memory effect

Now that the forward has been fully characterized, let us demonstrate the imaging capabilities enabled by this technique. Imaging corresponds to the reconstruction of the relative positions of the particles. If the medium is very thick and the Transmission Matrix completely random, there is no positional information and no imaging is possible. We can focus on the beads but cannot know where they are.

However, in deep biological imaging, there is often a little residual memory effect. This is demonstrated by the imaging technique of [78] where a focus is scanned in the memory effect range. However, this memory effect range usually limits the achievable field-of-view, which is an important limitation to image multiple cells for example.

In Fig. 4.4, we show that it is possible to exploit the memory effect for imaging. Thanks to the previous characterization of the optical system, we focus on each target and thus measure the fluorescent speckle coming from each target (that should correspond to the W matrix in the NMF, but measurement after focusing is much less noisy). It is then possible to cross-correlate all the pairs of focus positions, to find the ones which are close enough, within the memory effect range. Using all the pairs, it is possible to reconstruct an image with a field-of-view larger than the memory effect, iteratively from neighbor to neighbor or using Multidimensional Scaling algorithms. All the beads are reconstructed as long as their memory effect patches have some overlap.

4.3.4 Towards more complex objects

Finally, to test the limits of this approach, imaging of pollen seeds is reported in Fig. 4.5. This task is particularly challenging as our method requires specific assumptions which are not valid in this case. These objects are not sparse, they are continuous and even volumetric, which may be detrimental since we use a low-rank factorization and a 2D reconstruction.

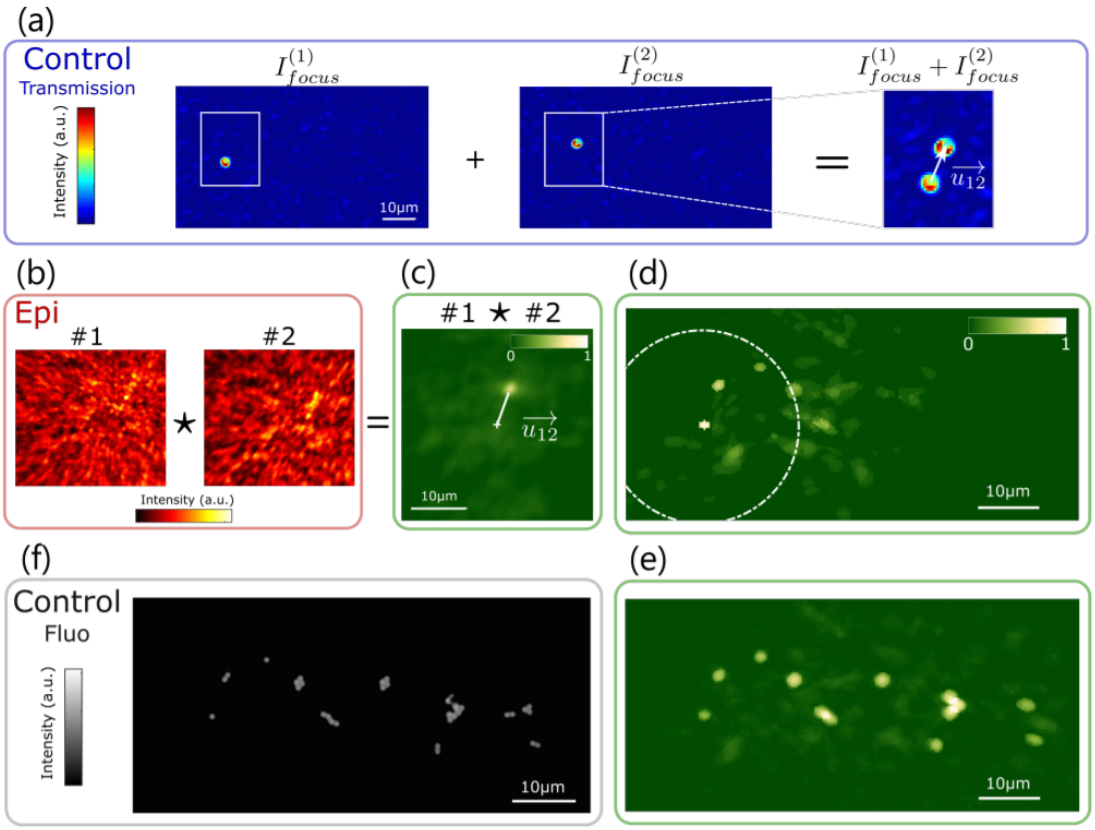


Figure 4.4: Imaging with the memory effect. If we focus on targets at 2 different positions (a), the corresponding fluorescent speckles are shifted (b-c) thanks to the memory effect. It is thus possible to reconstruct the positions around a focus (d), up to the memory effect range (dashed circle). We can then reconstruct iteratively an object (e), which matches well the control fluorescence image without scatterer (f). This imaging technique increases the field-of-view and goes beyond the memory effect range.

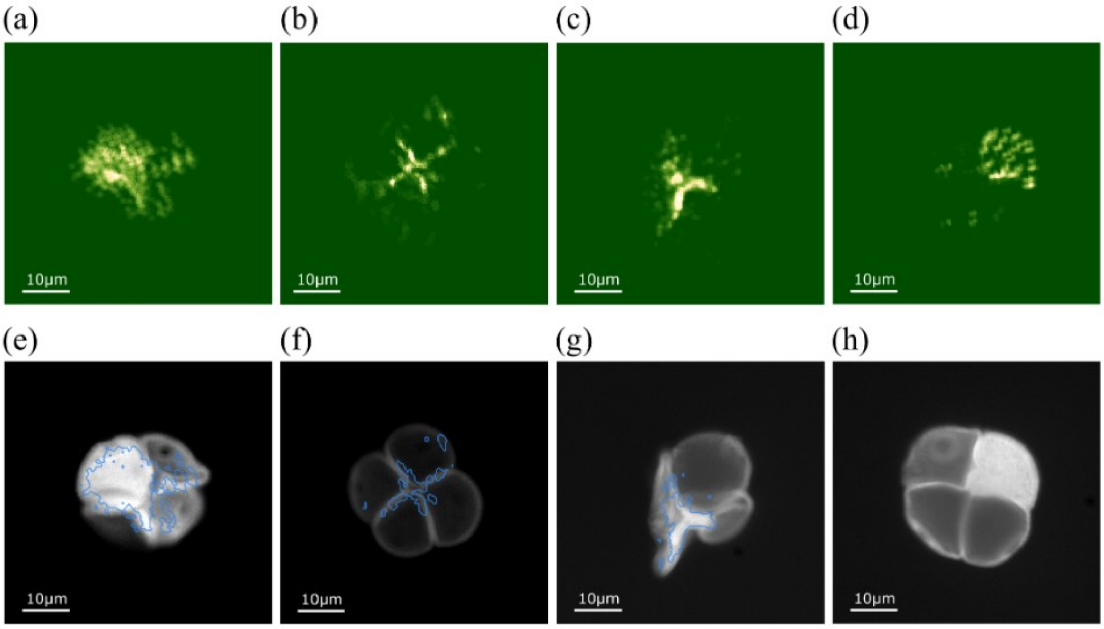


Figure 4.5: Imaging of fluorescent pollen seeds. After applying the computational pipeline, it is also possible to reconstruct volumetric objects hidden behind a scattering medium.

This thus tests the robustness of our technique, an important characteristic of any experimental investigation. The algorithmic pipeline is still able to reconstruct the two transmission matrices, allowing us to focus and reconstruct relative positions thanks to the memory effect. The final images are presented in Fig. 4.5, where we observe that we can locate where the density of fluorophores is higher.

An important step that makes this complex reconstruction possible is the focusing step that allows us to overestimate the number of focus positions in the NMF (up to 150 here) and clean the matrix afterwards. (a), (b), and (c) were obtained with $d = 256$, (d) with $d = 1024$ and we used an oversampling ratio $n/d = 20$ to ensure the phase retrieval problem is solved.

This proves that the developed algorithms are quite robust regarding the assumptions of the model, and can be applied to more complex objects. Note that we obtain a 2D projection of 3D object, so the presented reconstruction should only be considered as approximate projections.

4.4 Discussion

4.4.1 In non-linear optimization

We have already shown how this computational problem could be formulated as a two-layer neural network. Interestingly, whereas a gradient-descent algorithm would be the go-to method with a neural network, we have solved here this non-linear optimization problem with a two-step process involving an NMF first and Phase Retrieval

later. Both techniques are possible and are able to reconstruct the 2 TMs when the computational problem is simple (i.e. a lot of SLM patterns n are sent).

Nevertheless, our two-step approach is more sample efficient than gradient descent for finite n. This shows how drawing links between computational problems allows us to leverage precious algorithmic improvements. Vanilla gradient descent is not the optimal technique for this 2-layer network. Indeed, the limiting operation in the network training is the first layer obtained with phase retrieval. Gradient descent alone is quite inefficient at solving the phase retrieval problem [101], [110], and many specialized techniques detailed in the appendix have a better sample efficiency. All this discussion only comes from the numerical studies presented in this Chapter, we are not aware at this time of rigorous works to describe the maximal rank k for which reconstruction is still possible, for given number of samples n and input and output dimensions.

Gradient descent optimization would still be useful for later developments of this non-invasive scheme thanks to its flexibility. For example, other non-linearities can be introduced, in order to generalize to other optical contrast mechanisms like two-photon fluorescence or Second Harmonic Generation. It would also be interesting to modify the loss function, adding regularization terms or an L1 metric to remove outliers in certain adversarial settings.

4.4.2 In imaging

In practice, this completely non-invasive reconstruction strategy uses linear fluorescence to characterize light propagation in and out of a scattering medium. It allows both focusing at depth and, provided some memory effect is present, imaging of an extended object beyond the memory effect, an important achievement compared to previous techniques. The method is robust and provides another approach to push the limits of deep fluorescence imaging beyond the ballistic regime.

There are two main challenges to bypass or overcome for future applications in biological imaging. First, the scattering process in an alive animal is dynamic due to blood flow, breathing, etc. The two transmission matrices fluctuates in time accordingly. The two TMs measurement needs to be performed before decorrelation of the scattering medium, which limits the application of this technique. Still, it may prove useful for brain imaging through the skull, a very scattering but relatively static material.

Second, the contrast of the captured camera images needs to be large enough to be detected above noise. As it is an incoherent sum of individual speckles, the more targets are present, the more the different terms will average out. To increase the contrast of the final image, one can use filters to decrease the bandwidth of the fluorescence emission. It is also possible to tag selectively a very small region with fluorescence. This will reduce the number of excited targets and increase the overall contrast.

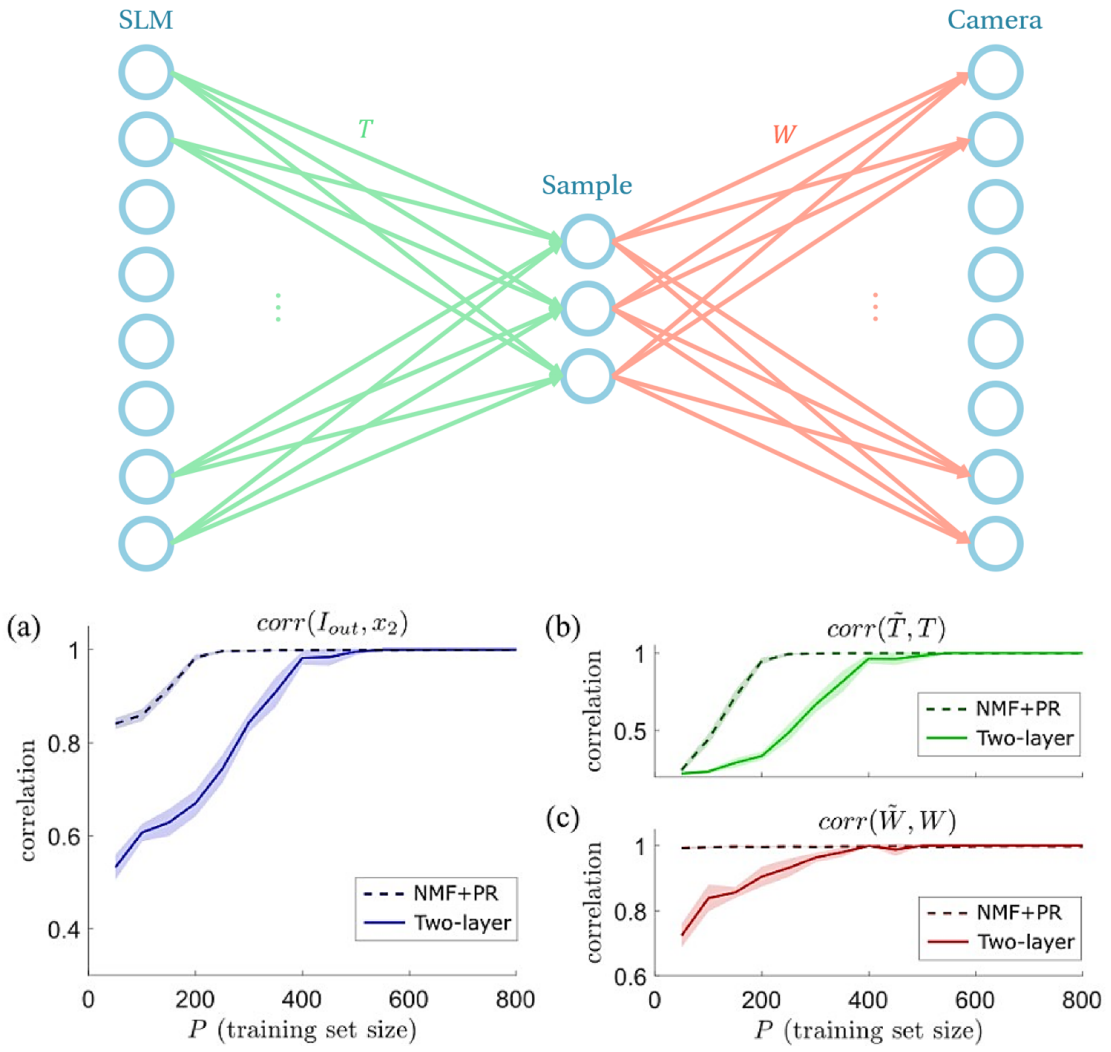


Figure 4.6: Comparison between the NMF+PR pipeline and a 2-layer neural network. Top: The computational problem of this chapter may be rephrased as finding the weights of a 2-layer neural network. Bottom: We show the gradient descent approach in a 2-layer neural network requires more examples to converge and is thus less efficient. (a) Correlations on a test set, (b) correlations of the coherent TM T, (c) correlations of the incoherent TM W. We observe that the bottleneck is the phase retrieval step to retrieve T.

CHAPTER 5

Optical random projections

5.1 A brief history of optical computing

[1] Optical computing describes a vast field describing how to process information in the optical domain [112]. After the first ideas emerged in the 50s [113], [114] followed a period filled with enthusiasm when researchers proposed several designs for an optical processor [115], [116]. The first successful optical processor was built for Synthetic Aperture Radar data, and was used until the seventies when the digital computer started to outperform the optical system [117]. In the end, we have inherited many strategies to implement correlators and Fourier transforms optically, these operations are well-suited to be performed in the optical domain.

However, optical computing never managed to replace electronic processors. Several reasons explain the relative lack of success of optical computing when compared to its electronics-based counterpart [112]. First, optical computing is suited to perform specific operations and does not possess the flexibility of manipulating bits directly with transistors. The Moore's law [1] also means that the efficiency of electronic chips grows exponentially, challenging optical computing to follow its pace. And on the other hand, several technological challenges, like efficient light sources or optical non-linearities, had to be overcome. In particular, there was for a long time no high-resolution, high-speed, and affordable Spatial Light Modulator available [118].

Many different technologies to modulate light were proposed. The two most promising technologies today were already introduced in Chapter 1: Liquid-Crystal SLMs and Digital Micromirror Devices. The first Liquid-Crystal SLMs appeared in the late 60s [119] with for example a 128×128 SLM in 1975 [120]. The characteristics of these modulation devices improved steadily, following the pace of our display devices. Today, one can find relatively affordable (~ 10 k€), high-resolution (1920×1080) and relatively fast (~ 100 Hz) Spatial Light Modulators to modulate the phase [121] (a variant of the LCD displays), or even cheaper and faster DMDs for binary amplitude modulation (also used in most videoprojectors).

Despite this technological progress, optical computing has not been able to compete with electronics for now. While a general-purpose optical computer is generally considered unrealizable nowadays, using optics may still prove useful for specific operations, as a co-processor with a digital computer.

Today, hopes in optical computing have revived thanks to the machine learning revolution [26], [48], [122], [123]. This formalism is on the one hand ubiquitous as it

[1] I am grateful to the team of LightOn [111] with whom I learned most of the notions presented in this introductory chapter. LightOn is a startup company co-founded by my two PhD supervisors developing optical computing solutions for large-scale machine learning.

can be applied to a large variety of topics and applications, and on the other gives a lot of importance to large-scale linear transforms, an operation of choice in optics.

Here, we investigate a particular subfield of optical computing, where one uses optics to perform random projections. For this reason we will first take a step back and introduce why random projections are useful to process high-dimensional data.

5.2 Randomness for dimensionality reduction

5.2.1 Dimensionality reduction

This chapter introduces the usefulness of a random matrix multiplication, to deal with high-dimensional data. To start this discussion, we focus on a particular application: dimensionality reduction. All the technical proofs of the theorems are provided in Appendix C.

We are often confronted with high-dimensional data, for example images, videos, texts, or genetic data are gathered in ever-increasing numbers today. These data points are usually represented as vectors $x_i \in \mathbb{R}^d$ for $i = 1, \dots, n$. Here d is the dimension of the data and n the number of data points. This data can then be processed by algorithms for various tasks, we will consider in this section the particular case of dimensionality reduction: how to send $x_i \in \mathbb{R}^d$ in a lower-dimensional representation $f(x) \in \mathbb{R}^k$ (with $k \ll d$) while still preserving the important features of the original dataset.

Dimensionality reduction is useful for many different reasons:

- **Visualization:** to visualize high-dimensional data, one usually projects them on a 2 or 3-dimensional space to represent them.
- **Generalization:** decreasing the dimensionality of a problem reduces the number of parameters and the risk of overfitting.
- **Efficiency:** algorithms working on compressed datasets are faster.
- **Modeling:** the low-dimensional representation captures the distribution of the original dataset.

In general the study of dimensionality reduction is a good starting point to understand what information is present in a dataset and how algorithms exploit it.

We will only discuss here Linear Dimensionality Reduction [124]. It consists in finding a set of projection vectors $U = [u_i]_{i=1,\dots,k} \in \mathbb{R}^{d\times k}$ such that:

$$f(x) = U^\top x \tag{5.1}$$

If we denote $f_i(x)$ the i-th component of $f(x) \in \mathbb{R}^k$, each of them corresponds to the projection of x on the direction of u_i: $f_i(x) = u_i^\top x$. These vectors u_i also form the basis to reconstruct the initial data from their low-dimensional representations:

$$\tilde{x} = Uf(x) = \sum_i f_i(x)\, u_i \tag{5.2}$$

where we assumed for simplicity the u_i to be normalized. This projection matrix may be defined in several different ways, which we are going to discuss next.

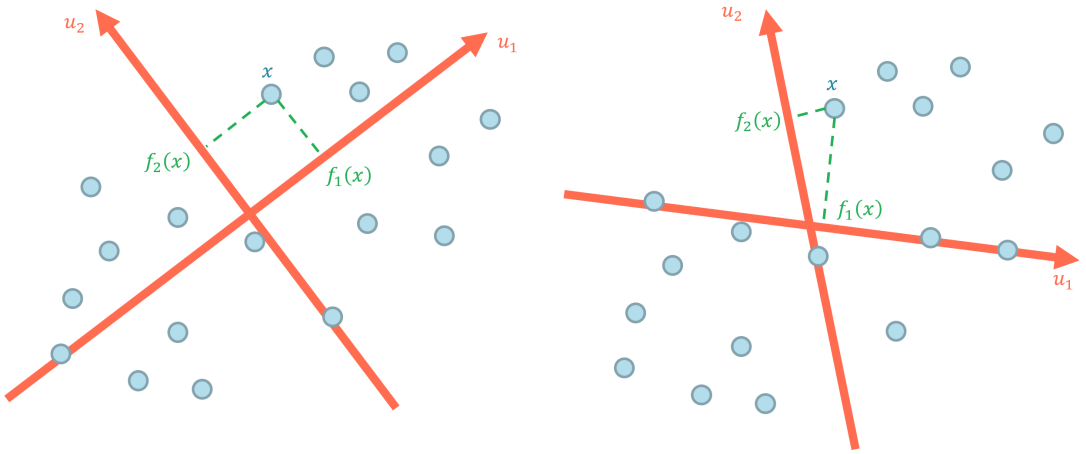


Figure 5.1: Dimensionality reduction using Principal Component Analysis or random projections. (Left) Data points x (in blue) are represented in 2D after PCA, the first eigenvectors encode the directions of highest variance, forming an orthogonal set of vectors. (Right) The same dataset may be projected using random vectors. In high-dimension, this operation approximately preserves pairwise distances and is an efficient method to compress information.

5.2.2 Principal Component Analysis

The most famous technique for dimensionality reduction is Principal Component Analysis (PCA) [125]. PCA was first proposed in 1901 by Karl Pearson, and has been a canonical method applied in very diverse topics since then. To introduce PCA, we first concatenate all the input data into a matrix $X = [x_i]_{i=1,\dots,n}$, and look at the eigenvalue decomposition of the covariance matrix $Z = XX^\top \in \mathbb{R}^{d\times d}$:

$$Z = XX^\top = U_{\text{full}}\Sigma U_{\text{full}}^\top \tag{5.3}$$

where U_{full} and Σ are two $d \times d$ matrices that are respectively orthogonal and diagonal. Σ describes the eigenvalues of Z that we usually sort in descending order, while U_{full} corresponds to the basis of eigenvectors, one for each eigenvalue.

The projection matrix $U_{\text{PCA}} \in \mathbb{R}^{d\times k}$ is obtained by taking the first k columns of U_{full}. They correspond to the k leading eigenvectors of $XX^\top$ and this minimizes the distance between X and its reconstruction $\tilde{X} = UU^\top X$ as shown in the following theorem [2]:

Theorem 3 (Principal Component Analysis)**.** *The matrix U_{PCA} defined as the k leading eigenvectors of $XX^\top$ minimizes the following error metric on the set of orthogonal matrices:*

$$U_{\text{PCA}} = \underset{U}{\arg\min}\, \|X - UU^T X\|^2 \tag{5.4}$$

As such, PCA may be seen as an optimal linear dimensionality reduction technique. However, optimality is always defined according to a particular metric, and

[2] Proofs of the theorems in this chapter are provided in Appendix C

thus is not universal. For example, PCA needs to compute the eigenvalue decomposition of $XX^\top$. This operation may not be practical when the input data X is very large. One would then prefer in this situation a dimensionality reduction technique which would not require an expensive analysis of X, and this is where random projections come into play.

5.2.3 Random projections

With random projections, one uses a random matrix $T \in \mathbb{R}^{k\times d}$, where each component T_{ij} follows an i.i.d. distribution (for $i = 1,\dots,k$ and $j = 1,\dots,d$). The random matrix multiplication outputs $f(x) = Tx \in \mathbb{R}^k$, in which the components of x have been shuffled and mixed together. Although the output $f(x)$ does indeed look random and may be hard to understand or visually interpret, we will show the regularity properties of this generic operation for it to be used in dimensionality reduction and more generally in machine learning. Thus, "random" should be considered more a synonym of "generic" rather than "suboptimal".

Instead of finding the directions of maximal variance in the dataset, this operation approximately preserves the distances between pairs of data points. This property is proven by the Johnson-Lindenstrauss lemma [126]:

Theorem 4 (Johnson-Lindenstrauss)**.** *Given* $0 < \epsilon < 1$ *and an output dimension* $k > 20\log(n)/\epsilon^2$*, then assuming the entries of* U *are sampled independently from* $\mathcal{N}\left(0, \frac{1}{k}\right)$*, we have with probability at least* $1 - 2e^{-(\epsilon^2-\epsilon^3)k/4}$*, for all* i, j*:*

$$(1-\epsilon)\|x_i - x_j\|^2 \leq \|U^\top(x_i - x_j)\|^2 \leq (1+\epsilon)\|x_i - x_j\|^2 \tag{5.5}$$

An important point to notice in this theorem is the size of the output dimension k. It is *logarithmic* in the number of samples n and it does not depend on the initial dimension of the data d. This surprising result means that random projections preserve distances very efficiently. Another application in imaging, called compressive sensing, also exploits this property of random projections, that mixes information to compress it.

Thus, random projections are:

- **Dense:** all the components of x are mixed together. There is no locality information like convolutional layers in Deep Learning.
- **Regular:** pairwise distances are preserved as long as the projection dimension is high enough.
- **Data-agnostic:** this projection can be applied to any data X. This is very different from PCA that requires an expensive diagonalization step.

Nonlinear dimensionality reduction techniques are also possible [127]. For example, autoencoders in Deep Learning have proven to be very powerful to compress texts or images [128]. Non-linear random projections have also been proposed: for example Locality-Sensitive Hashing [129], [130] introduces a non-linearity after the random matrix multiplication to preserve distances only when they are small, a feature which is interesting because it enables even higher compression than linear random projection.

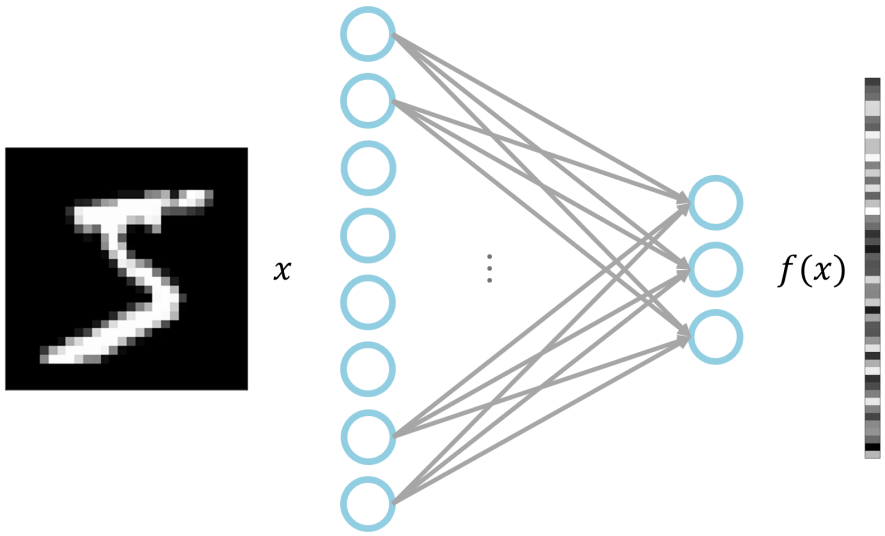


Figure 5.2: Dimensionality reduction as a neural network. Linear dimensionality reduction corresponds to a matrix multiplication, which may be viewed as a neural network. This network is fed high-dimensional inputs, for example images, and outputs a compressed representation of them. Using random weights is a generic and efficient way to compress information.

5.3 Random Features for inference

We have just discussed how random projections can be employed for dimensionality reduction, which may be roughly summarized to the case $k < d$. In this section, we investigate now the case $k > d$. High dimensionality with random projections also presents undeniable advantages. It is quite amenable to theoretical studies leading to algorithmic breakthroughs, thanks to the notions of measure concentration and the law of large numbers. For example, we are here going to introduce a non-linear inference algorithm based on random projections called Random Features [23].

5.3.1 Linear inference model

Inference problems consist in learning a function f to predict outputs labels $y \in \mathbb{R}$ from input data $x \in \mathbb{R}^d$. They are typically trained in a supervised setting. We first fix a model f_w depending a set of trainable parameters w, that can be a linear model as described below or a neural network for example. There is first a training step, where this model is shown many pairs of input outputs (x, y), adjusting the parameters w to minimize a loss function. The trained model may then be used for prediction, using the weights at the end of the training process.

Linear models are the fundamental building block of inference models. In these models, the predicted output $\hat{y}$ is obtained with a linear model:

$$\hat{y} = w_{out}^\top x \tag{5.6}$$

where the output weights are trained using a regression depending on the task at hand (linear regression for regression tasks and logistic regression for classification).

However, some problems cannot be solved with a linear model. A common example is the XOR task, where no hyperplane separates the output 1s to the 0s. The concept of non-linearly separable problems is fundamental: no linear transformation of x (dimensionality reduction or expansion) will make such a problem linearly separable. One needs to resort to non-linear models to solve such a task. We say that non-linear models are more *expressive* and can solve a larger class of problems.

5.3.2 Random Features

Thus, we will introduce here a first non-linear method. We transform the input data x with a random linear transform followed by an element-wise non-linearity:

$$\varphi(x) = \frac{1}{\sqrt{k}} f(Wx) \tag{5.7}$$

where $W \in \mathbb{R}^{k \times d}$ is a random matrix and f a non-linear activation function. Then, the output weights are trained on these non-linear features:

$$\hat{y} = w_{out}^{\top} f(Wx) \tag{5.8}$$

This approach may be interpreted as a two-layer neural network. The weights from the input to the input layer are fixed randomly, whereas the weights connected to the output are trained. Since only this last linear layer is trained, the training stays very simple and boils down to a linear regression. Still, the non-linear embedding $\varphi(x)$ augments the expressivity of this model. This approach appeared first in Extreme Learning Machines [131] and ELMs have then developed with many variants and applied to a large variety of tasks [24], [132], [133].

On the other hand, a high-dimensional embedding followed by a linear model is reminiscent of kernel methods, a famous and well-studied machine learning technique [25]. Kernel methods were also proposed to circumvent the limits of linear models. They send the data $x \in \mathbb{R}^d$ to $\psi(x)$ in a higher dimensional space $\mathscr{H}$ (possibly infinite-dimensional), in which the problem becomes linearly separable. Then they train a linear model on $\psi(x)$, which makes it a very robust and well-understood model.

Computing in a very large dimensional space $\mathscr{H}$ may be cumbersome. Instead, it is possible to train linear models not based on the actual data points but by using the scalar products only, all the $k(x, y)$ between each pairs of data points x and y. The set of pairwise scalar products $k(x, y)$ for all x, y forms an object called a Gram matrix, and the most common kernel methods build this Gram matrix without computing any explicit embedding $\psi(x)$.

However, these kernel methods requires to build an $n \times n$ Gram matrix, that grows quadratically with n. While a polynomial scaling may not seem problematic in complexity theory (it remains a problem of class P), with very large datasets, it is often not practical to construct such a large matrix. Furthermore, after the training step, to predict a new output, kernel methods need to compute the scalar products of the new input data point with all the training vectors. They are thus notoriously slow for prediction.

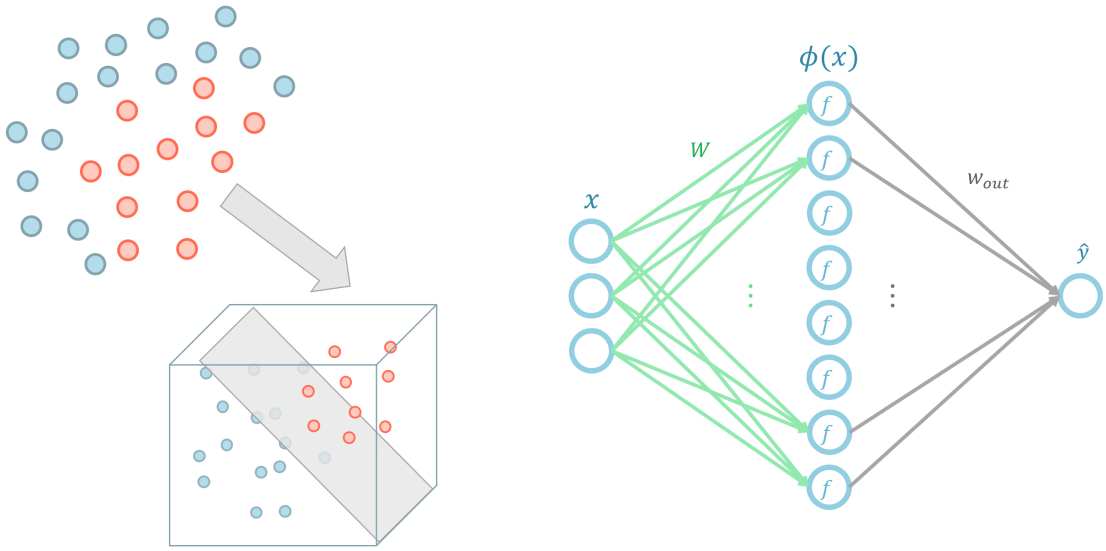


Figure 5.3: Kernel methods and Random Features. (Left) Kernel methods are based on non-linear embeddings in high-dimensional space, where problems become linearly separable. (Right) Random Features provide approximations of this non-linear embeding. They are more efficient than traditional kernel methods when the number of samples is large.

Rahimi and Recht proposed the celebrated Random Features in [22]. To approximate any translation-invariant kernel, they introduced a random projection followed by non-linearity to obtain a $\phi(x)$ similar to Eq. (5.7) that approximates the target kernel. More specifically, they define the random non-linear embedding as a concatenation, called Random Fourier Features:

$$\phi(x) = \frac{1}{k}[\cos(Wx), \sin(Wx)] \tag{5.9}$$

where $\phi(x) \in \mathbb{R}^{2k}$ and $W \in \mathbb{R}^{k \times d}$ a random matrix where each element W_{ij} is drawn from an i.i.d. distribution $p(w)$. Random Features are based on the following result:

Theorem 5. *For any translation-invariant positive definite kernel $k(x, y) = k(x - y)$, there exists a probability distribution $p(w)$ for the elements of W such that:*

$$\lim_{k \to \infty} \phi(x)^\top \phi(y) = k(x - y) \tag{5.10}$$

Thanks to this explicit embedding ϕ, Random Features reduce the computational complexity of kernel methods when the number of examples n is large. A linear model is trained on the non-linear features as described in Eq. (5.8). To perform a prediction after training, one only need to apply the Random Feature equation of Eq. (5.9) followed by the linear model of Eq. (5.8), in sharp contrast with conventional kernel methods which require the computation of scalar products with all the training data points. Next we will show how to accelerate further this Random Feature computation using optics to perform the matrix multiplication by W.

5.4 Optical Random Features

5.4.1 Principle

We have detailed in Chapter 1 how random matrix multiplications can readily be obtained in optics from multiple light scattering. It is thus natural to investigate whether one can perform this computation in the optical domain [26]. Since the operation is performed passively using light propagation alone, we will see the advantages of this implementation compared to the classical one on a conventional computer. This will provide an example of optical computing, where optics outperforms electronics for specific operations. In this section, we will introduce the principle and the results of the landmark paper [26], study the convergence towards the asymptotic limit and present a detailed benchmark [55].

In the first implementation of [26], they emulated in optics the Random Features algorithm described before for handwritten digits classification using the MNIST dataset [4], a classical benchmark in machine learning: the algorithm is presented images of handwritten digits between 0 and 9 and is asked to recognize them. The original purpose to build this dataset back in the 90s was to design an algorithm that would automatically recognize ZIP codes.

They used the simple experimental apparatus already described in Chapter 1, that we will later call an Optical Processing Unit (OPU). Light from a laser is shined on a Digital Micromirror Device (DMD), that acts as the Spatial Light Modulator here. The modulated wavefront is then sent in a complex material where it is scattered multiple times and is captured on a camera. The random matrix here corresponds to the Transmission Matrix between the SLM and the camera, which is apparently random due to the complexity of multiple light scattering. Note that it follows a complex gaussian distribution instead of a gaussian one, but Random Features can be used with different distributions.

More precisely, each input image x from the MNIST dataset is displayed on the DMD. Since the DMD is high-dimensional, the small resolution image (28×28 for MNIST) is expanded using macropixels on the Spatial Light Modulator. Thanks to multiple light scattering, one can collect on the camera the Random Feature projection $\phi(x)$ almost immediately (after a few picoseconds, the time for light to propagate to the camera):

$$\phi(x) = \frac{1}{\sqrt{k}}|Wx|^2 \tag{5.11}$$

This projection is also high-dimensional as cameras typically offer millions of pixels, it is subsampled for the following linear model. We would like to stress here the large dimensionality both for the input and output dimensions offered by the optics. Indeed, the number of modes that are mixed scales as $A/\lambda^2 \sim 10^6$ [35] with $A \sim 1$ mm^2 the illuminated area of the scattering medium and $\lambda \sim 1$ μm.

One limitation of the DMD is that this modulation device can only display binary 0/1 images. Thus, x needs to be binarized before being sent in the optical implementation. For this simple MNIST benchmark, images are already almost binary (pixels with ink versus pixels without ink) so Saade et al introduced a simple threshold-

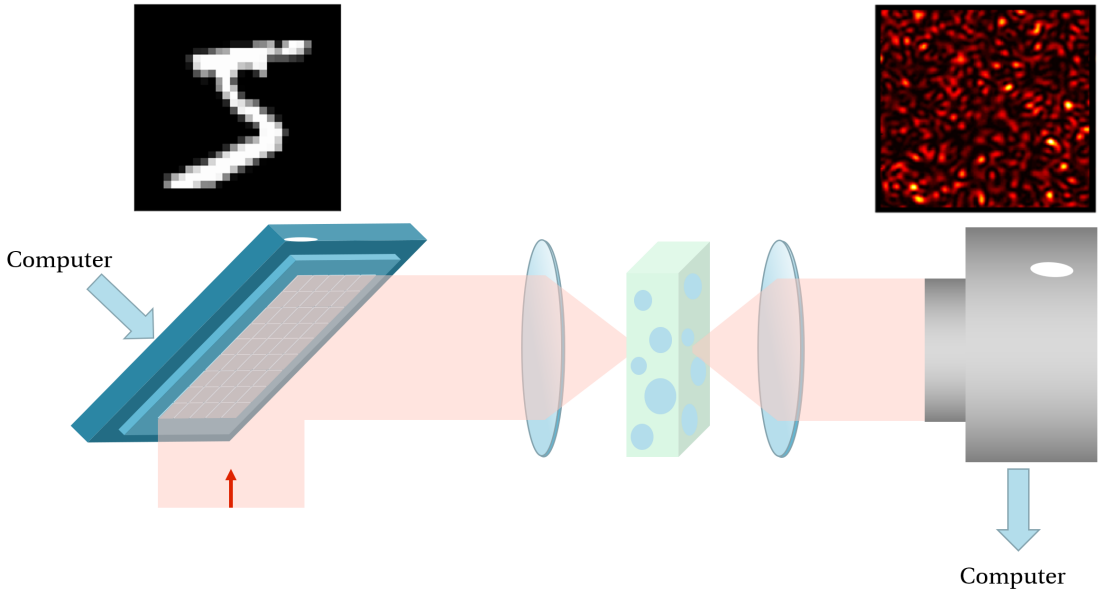


Figure 5.4: Scheme of the system to perform optical random projections. Light from a laser is sent on a Spatial Light Modulator displaying the vector x to project. Light then propagates through a complex scattering medium and a speckle pattern captured on a camera. Thanks to random interferences, the image on the camera corresponds to $\phi(x) = |Hx|^2$ with H a complex gaussian i.i.d. random matrix.

ing operation without losing too much information. This drawback could be solved by using Liquid-Crystal technology, but DMDs are less expensive and faster, making them very interesting for an optical computing implementation. We will compare these two technologies on a different task in the next chapter, as well as propose different binary encoding strategies.

5.4.2 Results

Fig. 5.5 presents the classification error on the MNIST dataset as the number of Random Features increases. We observe that the classification error decreases with the number of Random Features k, as we increase the dimension of the Random Feature expansion $\phi(x)$. The asymptotic limit for an infinite number of Random Features is a deterministic kernel with even better classification performance.

There is a discrepancy between experimental results and numerical simulation for a large number of features. [26] have checked that the Random Features algorithm is robust against additive noise (that may be interpreted as an L2-regularization term in the linear regression). Thus, they propose the residual correlations between output pixels as a possible explanation for this loss in performance, as they would reduce the effective dimension of the random feature map. Another important parameter to consider in experiments is the stability of the optical system: one wants the Transmission Matrix W to be constant from the beginning of the training to the end of the prediction steps. All these considerations are important when building an analog computing device, calling for the iterative refinement of optical computing systems.

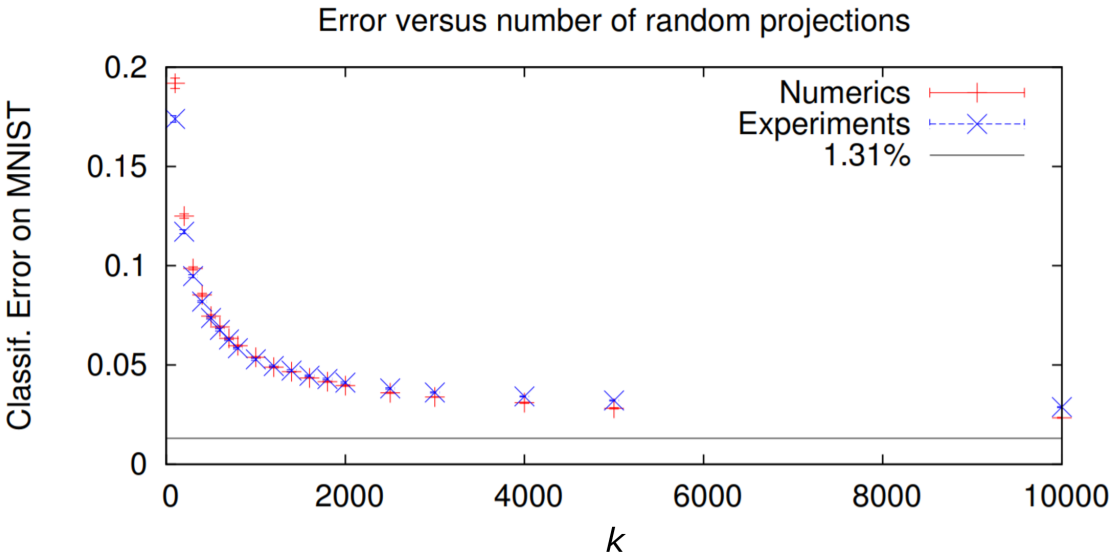


Figure 5.5: Classification error as the number of Random Features increases. The kernel limit at 1.31% is the asymptotic limit of the Random Features algorithms, and we can see that both numerical and optical Random Features seem to converge towards this limit. (Image from [26])

5.4.3 Convergence towards the kernel limit

Depending on the non-linearity f, different kernel functions can be approximated. In the first implementation, they implemented a modulus non-linearity, giving a complex elliptic kernel. We have proposed a different analysis in [55] that we detail here.

Theorem 6. *The kernel k approximated by the Optical Random Features defined in Eq.* (5.11) *is given by:*

$$k(x,y) = \|x\|\|y\|^2 + (x^\top y)^2 \tag{5.12}$$

Proof. The kernel limit of the Optical Random Features is defined by [22]:

$$k(x,y) = \int |w^\top x|^2 |w^\top y|^2 \mu(w)\,dw \tag{5.13}$$

with w a row of the random matrix W, drawn from an i.i.d. complex gaussian distribution.

Thanks to the rotational invariance of the random vector w, we can fix $x = \|x\|e_1$ and $y = \|y\|(\cos\theta e_1 + \sin\theta e_2)$. θ represents the angle between x and y and $\cos(\theta) = \frac{x^\top y}{\|x\|\|y\|}$. Let ${e_i}^\top w = w_i \sim \mathcal{CN}(0,1)$, $i = 1,2$.

Thus the kernel function becomes:

$$k_2(x,y) = \|x\|^2\|y\|^2 \int |w_1|^2 |w_1\cos\theta + w_2\sin\theta|^2 \mu(w)\,dw \tag{5.14}$$

We then expand the quadratic forms and compute the resulting gaussian integrals:

$$\begin{aligned}\frac{1}{\|x\|^2\|y\|^2}k_2(x,y) &= \int |w_1|^2|w_1\cos\theta + w_2\sin\theta|^2\frac{1}{\pi^d}e^{-\frac{\|w\|^2}{2}}dw \\ &= \int |w_1|^2|w_1\cos\theta + w_2\sin\theta|^2\frac{1}{\pi^2}e^{-\frac{(|w_1|^2+|w_2|^2)}{2}}dw \\ &= \int \left(|w_1|^4\cos^2\theta + |w_1|^2|w_2|^2\sin^2\theta + 2|w_1|^2\mathrm{Re}(w_1^* w_2)\cos\theta\sin\theta\right) \\ &\quad \frac{1}{\pi^2}e^{-\frac{|w_1|^2+|w_2|^2}{2}}dw\end{aligned}$$

The third term in the parenthesis is odd in w_2, so the integral of this term vanishes. Let's remark that if $w \sim \mathscr{CN}(0,\sigma^2)$ then $\mathrm{Im}(w), \mathrm{Re}(w) \sim \mathcal{N}(0, \sigma^{*2} = \frac{1}{2}\sigma^2)$. The two other terms can be computed easily using the moments of the gaussian distributions (the moment of order 2 of a complex gaussian random variable is $2\Gamma(2)\sigma^{*2} = 2\sigma^{*2}$, the moment of order 4 is $2^2\Gamma(3)\sigma^{*4} = 8\sigma^{*4}$ where σ^{*2} is the variance of the real and the imaginary part of $u_{1,2}$) .

$$\begin{aligned}k_2(x,y) &= 8\sigma^{*4}\cos^2\theta + 4\sigma^{*4}\sin^2\theta \\ &= 4\sigma^{*4}\|x\|^2\|y\|^2(2\cos^2\theta + \sin^2\theta) \\ &= 4\sigma^{*4}\|x\|^2\|y\|^2(1+\cos^2\theta) \\ &= 4\sigma^{*4}\|x\|^2\|y\|^2 + 4\sigma^{*4}\left(x^\top y\right)^2 \\ &= \|x\|^2\|y\|^2(1+\cos^2\theta) \quad \text{when} \quad \sigma^{*2} = \frac{1}{2}\end{aligned}$$

□

Numerically, one can change the exponent of the feature map to $m \in \mathbb{R}^+$, which becomes:

$$\phi(x) = \frac{1}{\sqrt{k}}|Wx|^m \tag{5.15}$$

Theorem 7. *When the exponent m is even, i.e. $m = 2s$, for all $s \in \mathbb{N}$, the dot product of feature maps of Eq. 5.15 tends to the kernel k_{2s} (for $k \to \infty$):*

$$k_{2s}(x,y) = \|x\|^m\|y\|^m\sum_{i=0}^{s}(s!)^2\binom{s}{i}^2\frac{(x^\top y)^{2i}}{\|x\|^{2i}\|y\|^{2i}} \tag{5.16}$$

Moreover, a generalization for all $m \in \mathbb{R}^+$ can be established.

Eq. 5.16 is connected to the polynomial kernel [134] defined as:

$$(\nu + x^\top y)^p = \sum_{i=0}^{p}\binom{p}{i}\nu^{p-i}(x^\top y)^i \tag{5.17}$$

with $\nu \geq 0$ and $p \in \mathbb{N}$ the order of the kernel. For $\nu = 0$ the kernel is called homogeneous. For $\nu > 0$ the polynomial kernel consists of a sum of lower order homogeneous polynomial kernels up to order p. It can be seen as having richer feature vectors including all lower-order kernel features.

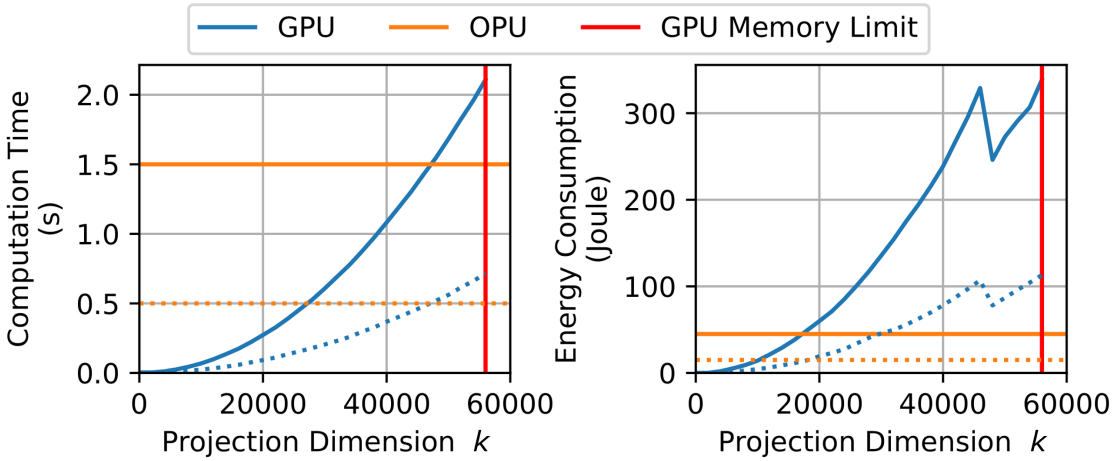


Figure 5.6: Time and energy spent for computing a matrix multiplication $(n,k) \times (k,k)$, with a batch size $n = 3,000$ (solid line) or $n = 1,000$ (dotted line). The curves cross each other at the same value of k, independent from n. Experiments were run on an NVIDIA P100.

5.4.4 Complexity and benchmark

Let us explain quickly why this operation is accelerated using optics. To recall the previous notation, if n describes the number of input data points, d their dimension, and k the Random Feature dimension, the complexity of the computation of $\phi(x)$ in Eq. (5.7) is $O(ndk)$ on a conventional computer, while the same computation has an $O(n(d+k))$ complexity in optics. One needs to display the n images one by one and the factor $d+k$ comes from the input-output bandwidth between the optical system and the computer, one needs to send and receive high-dimensional streams of data, an operation which is limited by the communication bandwidth between devices. It can sometimes be neglected as it dominates constant communication overheads only when $d+k \sim 10^5$. Note that only the random projection is accelerated optically, the linear model is still performed on the computer.

In [55], we have also benchmarked the so-called Optical Processing Unit (OPU), built and optimized by LightOn (a spin-off company co-founded by my two PhD supervisors), with a conventional CPU/GPU setup. The main advantage of the OPU is the speed at which the OPU can compute Random Features of large dimension. Moreover, its power consumption stays below 30 W independently of the workload. Fig. 5.6 shows the computation time and the energy consumption over time for GPU and OPU for different projection dimensions k. In both cases, the time and energy spending do not include the time to initialize the random matrix (a necessary step on the GPU). For the GPU, only the calls to the PyTorch function `torch.matmul` are measured and energy consumption is the integration over time of power values given by the command `nvidia-smi`.

For the OPU, the energy consumption is constant with respect to k and equal to 45 Joules (30 W multiplied by 1.5 seconds). The GPU computation time and energy consumption are monotonically increasing except for an irregular energy development between $k = 45,000$ and $k = 56,000$. This exact irregularity was observed

throughout all simulations we performed and can most likely be attributed to an optimization routine that the GPU carries out internally. The OPU is up to 7 times more energy efficient than the GPU for large random projections. The GPU starts to use more energy than the OPU from $k = 18,000$ and runs out of memory for $k = 58,000$. These exact crossover points are only indicative, as they depend on the GPU version and the optical implementation. The relevant point we make here is that the OPU has a better scalability in k with respect to computation time and energy consumption.

This benchmark is voluntarily focused on the Random Feature projection, which is usually followed by a linear regression. This second step is typically not the bottleneck for conventional computers, but becomes a challenge with the very high dimensionality achievable in optics. As such, we have not increased the size of the random projections to more than 100,000 in all the presented works, even though cameras offer millions of pixels, and implemented optimized linear solvers such as a Ridge regression method based on Cholesky factorization performed on GPU in [56], or a multi-GPU conjugate gradient iterative method in [55] or .

5.5 Where to use optical random projections

5.5.1 Advantages and challenges of optical random projections

Optical computing comes with a number of advantages compared to conventional computing:

- **Parallelism:** It is very large dimensional thanks to the large resolution of optical devices today. In particular, we do not need to store the random matrix in memory, which allows us to reach larger sizes than the memory limits of CPUs and GPUs.

- **Speed:** Information is processed as it propagates from the modulator to the camera. We are usually not limited by the propagation speed of information but rather by the operating speed and data bandwidth of the devices.

- **Energy-efficiency:** The computation is performed passively by optical propagation. As such, optical computing devices are much more efficient, and this key metric will prove very important in the years to come, as data centers become more and more power hungry.

Of course, challenges still need to be overcome before this technology reaches maturity. We may mention among others:

- **Noise and stability:** Analog computation are inherently corrupted by noise, with limited error correcting possibilities. Thus algorithms developed need to be robust against noise. Interestingly, this is what happens in the brain, and in general, high-dimensional problems are more robust against noise.

- **No backpropagation:** Since we do not know the matrix, backpropagation is not possible. Aligning the experiment pixel by pixel is possible, but challenging.

Architecture	ResNet34				VGG16		
Layer	L1	L2	L3	Final	MP2	MP4	Final
Dimension k	4 096	2 048	1 024	512	8 192	2 048	25 088
Sim. Opt. RFs	30.4	**24.7**	**28.9**	**11.6**	28.2	**20.7**	15.2 (**12.9**)
Optical RFs	31.1	25.7	29.7	12.3	30.9	21.5	16.4
RBF Four. RFs	**30.1**	25.2	30.0	12.3	28.0	**20.7**	14.8 (13.0)
No RFs	31.3	26.7	33.5	14.7	**27.1**	22.5	**13.3**

Figure 5.7: Test errors (in %) on CIFAR-10 using $k = 10^4$ RFs for each kernel (except linear). Values for the kernel limit are shown in parenthesis (last column). For the last column we added test errors for the kernel limit in parenthesis.

- **Non-linear readout:** The camera only detects intensities, imposing a non-linearity in the forward optical model.

- **Linearity only:** Conversely, the optical propagation is only linear. Having non-linear optical components would allow the implementation of deep neural networks optically, but for now no viable solution has been proposed (either requiring high power or not scalable to high dimensions).

5.5.2 Transfer Learning

Optical random projections may be used in any type of neural network with a fixed large-dimensional random layer. While an application will be studied in depth in the next chapter, let us mention here the case of Transfer Learning for image classification proposed in [55]. Transfer Learning is about reusing the trained layers of a deep neural for a different task, leveraging the preprocessing steps of the first layers of the network to extract meaningful information about images. This alleviates the need for a very long training process on the new task, by reusing the weights of the pre-trained network.

We extract a diverse set of features from the CIFAR-10 image classification dataset using two different convolutional neural networks (ResNet34 and VGG16). The networks were pretrained on the well-known ImageNet classification benchmark and we directly apply a classifier on the convolutional features of the data at hand. This requires much less computational resources than training a new deep neural network for this new dataset, while still producing considerable performance gains over the use of the original image pixels.

We compare Optical Random Features to a another type of Random Features called Fourier Random Features of [22] and a simple baseline that directly works with the provided convolutional features (with a linear model). Table 5.7 shows the test errors achieved on CIFAR-10. Each column corresponds to convolutional features extracted from a specific layer of one of the three networks.

Since the projection dimension $k = 10^4$ was left constant throughout the experiments, it can be observed that RFs perform particularly well compared to a linear kernel when $k \gg d$ where d is the input dimension. For the opposite case $k \ll d$ the lower dimensional projection leads to an increasing test error. This effect can be ob-

served in particular in the last column where the test error of the RF approximation is higher than without RFs. The contrary can be achieved with large enough k as indicated by the values for the true kernel in parenthesis. In general, the simulated as well as the physical optical RFs yield similar performances as the RBF Fourier RFs on the provided convolutional data.

5.5.3 Anomaly detection

There are other machine learning applications involving random projections. For example, NEWMA [135] (for "No-prior-knowledge" Exponentially-Weighted Moving Average) proposes an anomaly detection algorithm using Random Features, based on a simple principle.

Given a stream of input data x_t and a Random Feature embedding $\phi(x_t)$, one constructs the 2 following sketches of the input distribution:

$$\begin{cases} z_t = (1-\Lambda)z_{t-1} + \Lambda\phi(x_t) \\ z'_t = (1-\lambda)z'_{t-1} + \lambda\phi(x_t) \end{cases} \tag{5.18}$$

for two forgetting factors $0 < \lambda < \Lambda < 1$. The larger the forgetting factor, the more z_t changes, so z'_t evolves slower than z_t and encodes the long-term distribution of the input data, while z_t responds quickly to new changes.

As such, NEWMA detects a change-point when the following condition is observed:

$$C_{\text{NEWMA}} : \|z_t - z'_t\| \geq \tau \tag{5.19}$$

for some threshold $\tau > 0$. This condition corresponds to the short-term distribution encoded in z_t diverging from the long-term distribution z'_t. One characteristic of this technique is its flexibility, it requires no a priori information on the distribution of the input data.

Here, the random features $\phi(x_t)$ are employed to provide a regular and well-controlled sketch of x_t. NEWMA thus provides a simple online method to detect change-points in the distribution of input data streams, that can be accelerated using Optical Random Features. This work has been applied to change point detection in molecular dynamics.

5.5.4 Training networks without gradient descent

Intriguingly, we may also employ random projections to train deep neural networks [136]. Deep neural networks are typically trained using gradient descent and backpropagation, a method to compute the gradient with respect to each weight parameter by *backpropagating* the information backwards in the network. As backpropagation involves multiplying by the transpose of the weight matrices, one may wonder what would happen if these are replaced by random matrices instead.

Training is successful even with this "apparently-suboptimal gradient". This may be heuristically explained by the heavy overparametrization of deep neural networks: among all the solutions, FA and DFA find the set of weights for which the true gradient *aligns* with the random matrix.

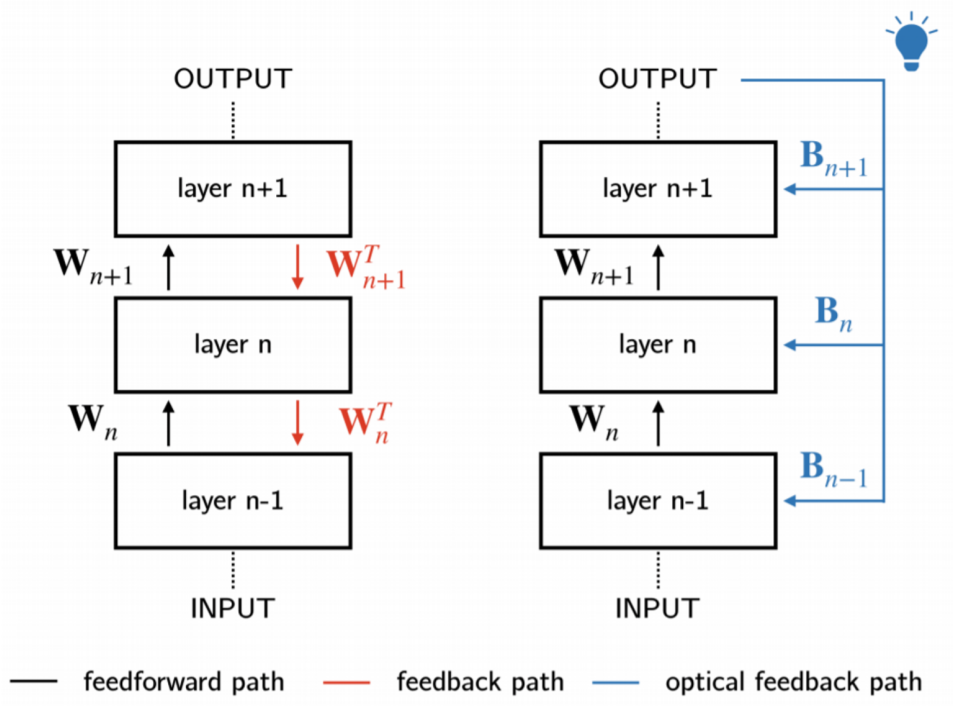


Figure 5.8: Principle of Direct Feedback Alignment. (Left) In backpropagation, the error signal is backpropagated through the network involving multiplications by the transpose of the weight matrices. (Not shown) In Feedback Alignment, these matrices are replaced by random matrices and training is still successful in certain settings. (Right) One may even backpropagate directly from the error signal to any layer using a random matrix. This approach called Direct Feedback Alignment has been accelerated optically (image from [137]).

This random matrix multiplication may be performed in optics as proven in [137], even for very large neural networks. Note that this operation requires a random matrix multiplication without the modulus square, made possible using an external reference arm in optics.

5.5.5 Towards Reservoir Computing

Finally, it is also possible to design Recurrent Neural Networks with randomly-fixed weights, also called Echo-State Networks [138]. As they gave rise to the more general framework of Reservoir Computing [139], this particular network architecture has been extensively studied, both theoretically and for physical accelerations.

In the next chapter, we will describe how to implement it using multiple light scattering, enabling very large reservoir sizes and pushing it to very high performance. Then a theoretical work will be presented in Chapter 7, accelerating Reservoir Computing considerably even without the use of optics.

CHAPTER 6

Optical Reservoir Computing

[1]As we have shown how optics represents a promising alternative to accelerate random matrix multiplications in machine learning, we will focus here on a particular Recurrent Neural Network (RNN) architecture called Reservoir Computing. RNNs are notoriously hard to train [13]. Recurrent connections are a challenge for error back-propagation, which is the method of choice to train neural networks with feed-forward connections such as Convolutional Neural Networks [4]. Back-propagation through time is possible [140], but this method faces the problem of local minima, as well as exploding and vanishing gradients [13]. As a possible solution to bypass this training issue, Echo-State Networks (ESNs) [12], [138] are Recurrent Neural Networks with randomly fixed internal weights, and they are part of the more general framework of Reservoir Computing [15]. Only the output weights are trained for a particular task, reducing the training to a simple linear regression. The number of tunable parameters is thus smaller, but this does not necessarily mean that this neural network model is less expressive than fully-tunable Recurrent Neural Networks. It is very easy to increase the number of neurons and it has been proven that large networks can universally approximate any fading memory input/output system [141].

In this chapter, we will show how optics, and in particular multiple scattering, can accelerate the random matrix multiplication in Reservoir Computing. We will first discuss how to operate a reservoir with time-dependent data, to then detail the optical implementation and how it performs against conventional computing solutions.

6.1 Reservoir Computing

6.1.1 Recurrent equation

To operate an Echo-State Network, a time-dependent input is first fed to the network with fixed weights. After the computation of all the ESN states, the output weights are either learned with a training dataset containing the desired outputs, or used to obtain a predicted output on another dataset for validation.

Let $\{i(t), t = 0, ..., T\} \in \left(\mathbb{R}^d\right)^T$ be an input time series of dimension d and of length T. The ESN will be initialized in a random state $x(0) \in \mathbb{R}^n$ and its state at time t will be denoted $x(t)$; n is the dimension of the network, i.e. the number of neurons or reservoir nodes. Let W_{res} be the internal weight matrix and W_{in} the weight matrix between the input and the network. Both weight matrices are random and fixed for

[1]This project was initially developed during my Master internship and first year of PhD. Then we were joined by Dr. Mushegh Rafayelyan (postdoctoral researcher with Pr. Sylvain Gigan) that pushed forward this research project, with the help of Yongqi Tan (Shanghai Jiao Tong University).

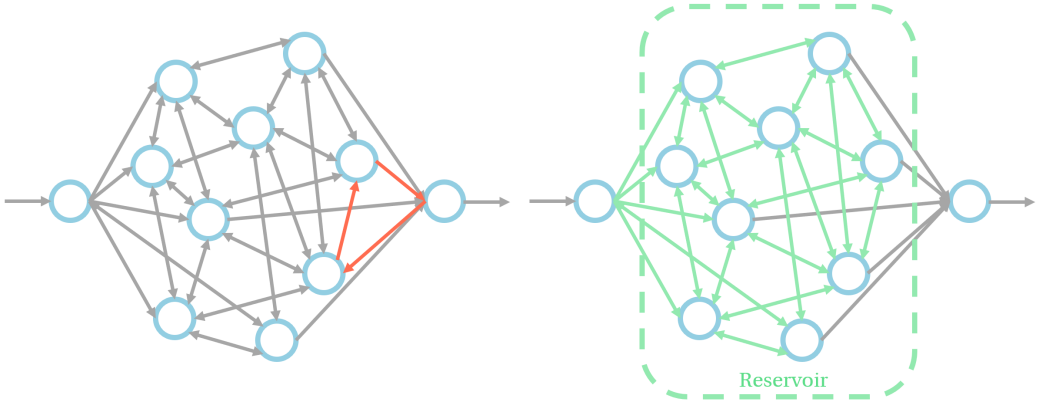


Figure 6.1: Principle of Reservoir Computing. (Left) A Recurrent Neural Network. A loop in the backpropagation algorithm is shown in red. (Right) Echo-State Networks where internal weights are randomly fixed are one of the first instance of Reservoir Computing.

Echo-State Networks, of variances σ_r^2 and σ_i^2 respectively. The nonlinear activation function of every neuron will be denoted f. The successive ESN states are computed using the following recursive equation:

$$x(t+1) = f(W_{\text{in}} i(t) + W_{\text{res}} x(t)) \tag{6.1}$$

In other words, an Echo-State Network is a large set of neurons randomly interconnected, that evolves dynamically driven by an external input. This leads to the more general framework of Reservoir Computing, where the neural network can be replaced by any non-linear dynamical system to which an input time series is fed. At any time, the current state of the reservoir depends on the previous values of input data, it encodes this information in its state.

This more general framework adds more flexibility to the model, that we will leverage for our optical computing implementation. In particular, we introduce a leak rate a, a random bias vector b, and an encoding function g into the ESN equation:

$$x(t+1) = (1-a)x(t) + af\big(W_{\text{in}} g(i(t)) + W_{\text{res}} g(x(t)) + b\big) \tag{6.2}$$

The leak rate a is an important parameter that controls the speed of the dynamics of the reservoir without changing its long-term stability, the bias parameter b controls the diversity of neurons inside the reservoir states. These two parameters are commonly used in conventional Reservoir Computing to improve the overall performance [16].

Finally, the encoding function will be important in the following as images on the Spatial Light Modulator need to be either binary or phase-only. We have first made a first implementation which did not use such encoding function, but binarized the reservoir states with a threshold activation function f [27]. With this simple change, it allows the networks states to be real, which increases their diversity and the final performance considerably.

6.1.2 Final linear layer

The output $o(t) \in \mathbb{R}^k$ is computed using a linear combination with weights $W_{\text{out}} \in \mathbb{R}^{k \times n}$, which can be written as:

$$\hat{o}(t) = W_{\text{out}} x(t) \tag{6.3}$$

The optimal set of output weights is obtained by solving a linear regression problem, which minimizes the following error metric:

$$E = \frac{1}{k} \sum_{t=0}^{T} \| o(t) - W_{\text{out}} x(t) \|^2 \tag{6.4}$$

where $o(t)$ is the target output at time t.

Linear regression is a well-studied problem and many libraries already provide an efficient solver. Ridge regularization is important to avoid overfitting when the number of parameters n, which is proportional to the number of neurons, is larger than the number of examples T. It corresponds to the addition of a term $\alpha \| W_{\text{out}} \|^2$ in Equation (6.4), yielding the following explicit formula:

$$W_{\text{out}} = O X^\top (X X^\top + \alpha I_n)^{-1} = O (X^\top X + \alpha I_T)^{-1} X^\top \tag{6.5}$$

where $X = [x(t)]_{t=1,\dots,T} \in \mathbb{R}^{n \times T}$ and $O = [o(t)]_{t=1,\dots,T} \in \mathbb{R}^{k \times T}$

The dimensionality of the linear regression is important for the choice of the algorithm to use. Here $n, T \sim 10^4$, which is still manageable on conventional CPUs and GPUs. Here, we use the Ridge solver of the scikit-learn library in Python [142], as we do not focus on optimizing this linear regression step in this proof-of-concept of Optical Reservoir Computing. If performance is an issue, one could use the multi-GPU conjugate gradient optimization implemented for the Transfer Learning work presented in the previous Chapter [55].

6.1.3 Physical implementations of Reservoir Computing

The flexibility of RC, that can use any generic high-dimensional reservoir for computation, makes it very promising for physical implementations [52], [53]. Many have been proposed originating from very different research areas, such as optical nano-circuits [143] or carbon nanotubes [144], one early RC implementation even observed the ripples at the surface of a bucket of water for pattern recognition [145]. The RC framework is robust against noise and changes in network topology (dense, sparse, and local connections can be used). In principle, various physical systems receive an external time-dependent excitation and follow non-linear dynamics, sending the sequential input into a high-dimensional feature space. To make a prediction, the output is obtained by a linear combination of the observed state of the reservoir.

Following the recent trend of specialized electronic processors for efficient machine learning [146], neuromorphic electronic circuits for RC have been developed, based on analog circuits, FPGAs [147] or memristive devices [148].

Optical node arrays have also been proposed, where semiconductor optical amplifiers are used to perform the non-linear activation function [143]. Increasing the

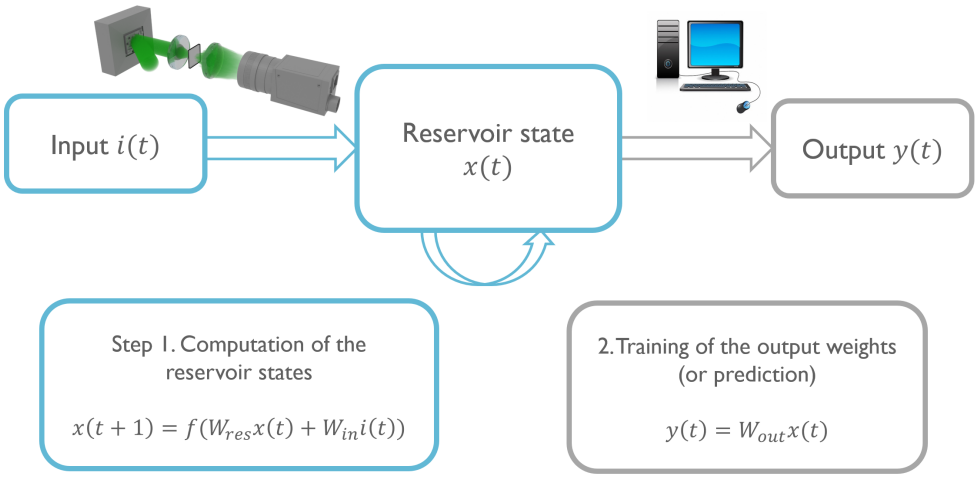


Figure 6.2: Reservoir Computing pipeline. We first compute the successive reservoir states using Eq. (6.2), then train or use a linear model to perform a prediction (6.3). We use optics to accelerate the first step, using multiple light scattering to perform the random matrix multiplication. The rest of the pipeline is performed on a conventional computer.

number of neurons means increasing the number and the density of optical nodes, which is a challenging manufacturing task to address.

On the other hand, some optical reservoir computers use a single physical node [149]–[151]. They use temporal multiplexing and are based on a delay-fiber system to generate interconnections between units. They can be operated very fast at the GHz frequency and there is a very active community pursuing this line of research. In this case, a large number of neurons decreases the effective frequency of this time-multiplexed scheme as more nodes need to be sent one by one in the delay system.

Our strategy to increase the dimension of the reservoir is to use free-space optics, combined with high-dimensional cameras and SLMs. The manufacturing challenge of producing cameras and SLMs with a large number of pixels has already been solved thanks to their use in the display industry. By exploiting this commodity, it is possible to implement large-scale Optical Reservoir Computing, interconnection between units being either provided by a Diffractive Optical Element (DOE) [122], [152] or a scattering medium [27]. In the first case, connections are typically local and they can be engineered when designing the DOE. In the second case that we study here, the weight matrix is dense and random, which is closer to the original ESN definition. With these devices, we can reach very large sizes but the operating frequency is limited by the SLM, which is typically in the tens or hundreds of Hertz, or more often by the camera. An implementation using an LCoS-SLM in phase modulation to generate the reservoir couplings has also been proposed [153].

6.1.4 Analysis of the reservoir dynamics

Before moving onto the practical optical implementation, let us discuss a particularity of Reservoir Computing, where a non-linear dynamical system is driven by an

external input. The reservoir in Reservoir Computing is a tunable dynamical system, that can be put in a stable or chaotic regime depending on a few parameters controlling the dynamics. This is linked with the Echo-State Property, introduced since the founding Echo-State Network paper [12]. It states that a reservoir needs to forget its initial state after a finite transition time. More precisely, for a given input time series, the network state $x^{(t)}$ needs to be uniquely determined by $\{i^{(t')} | t' \leq t\}$ (assuming the input time series is infinite on the left). In practice, we observe a fading memory of the past in a stable network, an important property as one does not want the current state to depend on the network initialization, since it is random and unrelated to the task at hand. The reservoir needs to operate in a stable dynamical regime, in contrast to a chaotic regime that would be very sensitive to initial conditions. As a consequence, a chaotic dynamical system cannot be used for Reservoir Computing.

Empirically, best performance in Reservoir Computing happens when the reservoir is "at the edge of chaos" [154], which corresponds to a stable regime close to a chaotic one that exhibits rich but stable dynamics. A sufficient condition for stability is $\sigma_r < 1/2$, assuming W_{res} to be i.i.d. random and f 1-Lipschitz continuous, and a necessary condition is $\sigma_r < 1$ [155], while the necessary condition is often sufficient in practice. Finally, to ensure that the reservoir state does not depend on the initial conditions, we typically throw away the first states following the initialization, both for training and prediction.

To describe the complex dynamics of the Reservoir, several properties and quantities of interest have been proposed. The first and most important one, called the Echo-State Property [12], describes the stability of the dynamical system. It states that a reservoir needs to forget its initial state after a finite time to be used for RC. In other words, for a given input time series, the observed state of a reservoir after a warm-up phase shall not depend on the method used to initialize the network, which is not related with the task to solve. The reservoir needs to operate in a stable dynamical regime, in contrast to a chaotic regime that would be very sensitive to initial conditions. As a consequence, a chaotic dynamical system cannot be used for Reservoir Computing. Additionally, to ensure that the reservoir state does not depend on the initial conditions, we typically throw away the first states following the initialization, both for training and prediction.

In practice, adding an external input and bias stabilizes the reservoir dynamics [156], this frontier between chaos and stability thus depends on the dataset at hand. We will choose an empirical approach: the dynamics is controlled by a few parameters and we use a hyperparameter search to find a particular dynamical system suited for the task at hand. For example in our last work [56], we have chosen $\sigma_r \approx 1.01$. Note that this procedure is common in Machine Learning as we often need to search for the best model to use for a particular problem by tuning a set of hyper-parameters.

To describe the complex dynamics of the Reservoir, other properties and quantities of interest have been proposed [157]. The separation property means that two different input time series need to lead to different final reservoir states. A complex reservoir with rich dynamics is able to encode more information and differentiate a larger number of inputs. The approximation property states that a single input time series perturbed with noise should consistently be mapped to the reservoir state. These properties are useful to characterize in a simple way the high-dimensional dy-

namics of a reservoir. For instance, it is possible to derive quantitative measures of these properties on experimental Reservoir Computers, and even display the observables introduced in this subsection in a 3D space to visualize the performance of a particular RC implementation [158].

In the end, there are four requirements for a dynamical system to be effectively used in RC [52]: high-dimensionality of the reservoir, non-linearity in the dynamics, the Echo-State Property, and a balance between the separation and approximation properties which depends on the task to solve. Note that this discussion is mainly heuristic, and it is difficult to compare different Reservoir Computing implementations, especially given the large variety of physical implementations. An approach towards a unifying framework is provided in [158].

6.2 Optical Reservoir Computing

6.2.1 General principle

The experimental setup for our implementation here is the same as detailed in the previous chapter [26]. To summarize it quickly here, it comprises three main devices: a Spatial Light Modulator to modulate the electric field, a complex scattering medium to perform the random matrix multiplication optically, and a camera to capture the resulting image called a speckle pattern.

Only the reservoir update of Eq. (6.2) is accelerated in the optical domain. More precisely, the random matrix multiplication is performed optically and the feedback from the camera image to the next SLM image is performed electronically. We are thus limited by the communication speed between devices and constant overheads.

This resulting speckle pattern determines the next reservoir state, that will be encoded and displayed back on the SLM with the next input. This feedback loop between the camera and the SLM corresponds to one Reservoir Computing iteration and it will be repeated as many times as there are reservoir states to compute. Compared to the Optical Random Features algorithm on the MNIST dataset [26], our configuration is more beneficial for our optical design, because the input dimension in their implementation is relatively low at 784. Here we leverage not only the high resolution of the camera (to get the next reservoir state) but also of the SLM (to display the current reservoir state), to enable large reservoirs.

On the other hand, the linear regression (the second step in Fig. 6.2) is performed on the computer. This prevents us from going to very large reservoir sizes $n \sim 10^6$, due to the time and memory costs. To circumvent this issue, an all-optical implementation has been proposed [122] but also creates additional challenges. For example, they do not measure $x(t)$, only $\hat{o}(t)$, and had to resort to a genetic algorithm to solve the large-scale linear regression.

6.2.2 Different optical implementations

This project has been carried out over the span of 4 years, with two different SLM technologies to compare. It resulted in a number of different types of optical imple-

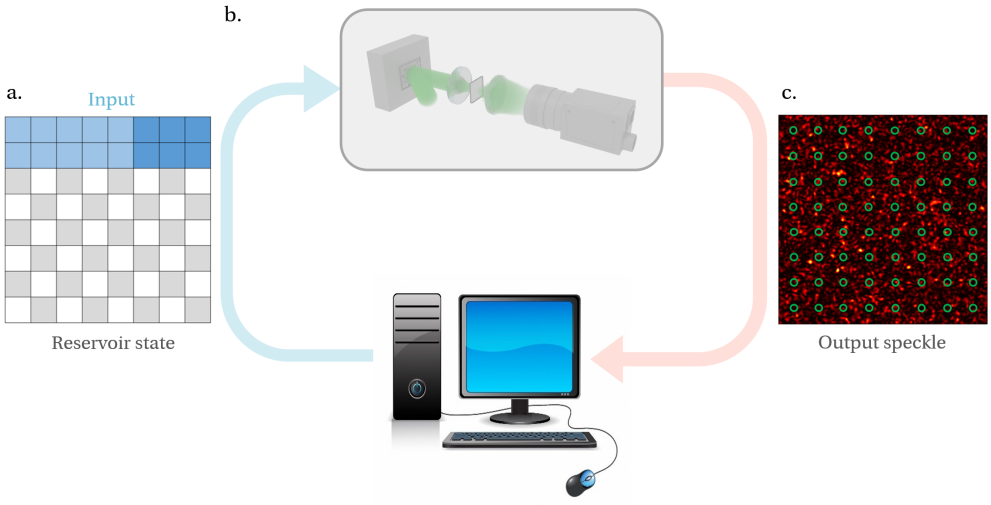


Figure 6.3: Loop between the computer and the optical system for Reservoir Computing. For every RC iteration, we send an image containing the current reservoir state and input to the SLM. The result of is collected on the camera, where we subsample to avoid the speckle grain size correlation.

mentations, that reflects the different steps of the project:

- **Optical system DMD1:** I inherited this setup from [26], with a Vialux DMD and a Basler ace camera. It was used for the first implementation of Optical Reservoir Computing in 2016 during my Master internship. As a non-optimized system, we were limited in speed to 10 Hz to wait for the synchronization between camera and DMD. Since it is based on a Digital Micromirror Device, we can only display binary images in the optical domain.

- **Optical system DMD2:** This DMD implementation has been developed by the startup LightOn and available for researchers as a cloud service. This system was accessed in February 2018 and has been greatly improved since then.

- **Optical system LCoS1:** An LCoS-SLM implementation (we recall that the Liquid Crystal technology enables phase modulation), initially built for another project [159], was used in our comparison between DMDs and LCoS-SLMs [28]. This optical system was built to study the propagation of optical pulses in scattering media. The wavelength of the laser we used is at 800 nm (Ti:Sapphire laser, MaiTai, Spectra Physics, operated in continuous-wave mode), we expand the beam to fill the SLM, a Meadowlark 512×512 LCoS Reflective SLM. The modulated electric field is then focused by a 20× objective with 0.4 numerical aperture on a 0.5 mm thick scattering material. The scattered field is collected by another similar objective and the resulting speckle is captured by a CCD camera (Allied Vision, Manta G-046).

- **Optical system LCoS2:** We have then built a dedicated system optimized for Optical Reservoir Computing. The laser has been replaced by a continuous 532 nm laser (Coherent), the SLM now has 1920 × 1152 pixels (Meadowlark Optics HSP192-532), and the camera is a Basler acA2040-55um with 2048×1536 pixels.

In the end, two systems for both DMD and LCoS technologies were implemented, the first one for a proof-of-concept and a second where stability, number of modes, and speed have been optimized.

6.2.3 Complexity and speed

Here we propose a dedicated benchmark of the optical acceleration for Reservoir Computing, for the Optical system LCoS2. Compared to the similar one presented in the last chapter, with Reservoir Computing, there is no batch size, images are displayed one by one in Reservoir Computing. Additionally, we take into account the communication bandwidth between devices. As any benchmark, it is important to keep in mind that the numbers presented here are dependent on the hardware at hand and the particular implementation. They serve to obtain overall trends and general information about regimes where one device is more performant than another.

We introduced our optical implementation to accelerate the random matrix multiplication of Eq. (6.2). This operation represents the bottleneck for large-scale Reservoir Computing, as it has a quadratic complexity in the reservoir size n. On the other hand, the optical implementation has a linear complexity scaling: while the time of the random matrix multiplication is usually negligible in optics, this linear complexity comes from the time to transfer data with and from the optical devices. In the inset with log-log axis, we observe two regimes for the optical device. For small sizes (up to $\sim 10^4$), the operating speed and the synchronization of the devices is the most demanding operation, with a maximum frequency less than 150 Hz. We then observe the linear scaling, with crossover points at $5,000$ and $25,000$ for the CPU and GPU respectively. Thus, optics enables us to perform the operation faster for large reservoir sizes.

Moreover, the CPU and GPU are restricted in size due to memory limitations. GPUs come with embedded memory, here we used an NVIDIA Tesla V100 with 16 GB of memory, one of the best available GPU as of 2020. On the other hand, CPUs use the memory of the computer, here the computer was equipped with an Intel Xeon Gold 5120 with 64 GB of RAM memory. In both cases, the largest possible size of the reservoir was on the order of 50'000, as the square internal weight matrix overflows the memory for very large sizes.

This benchmark is focused on the random matrix multiplication. Other operations necessary in the Reservoir Computing pipeline are the linear regression and the matrix-vector multiplication for a prediction. These are typically not the bottleneck, but become important for large-scale Optical Reservoir Computing. Many different algorithms have been proposed for linear regression, it would be interesting to perform a benchmark for Reservoir Computing. However, we focused here on the step that could be accelerated optically.

6.2.4 Encoding as a pre-processing kernel

Physical constraints are unavoidable with analog computing. For example here, due to the physical constraints of the SLM, the displayed image needs to be either binary

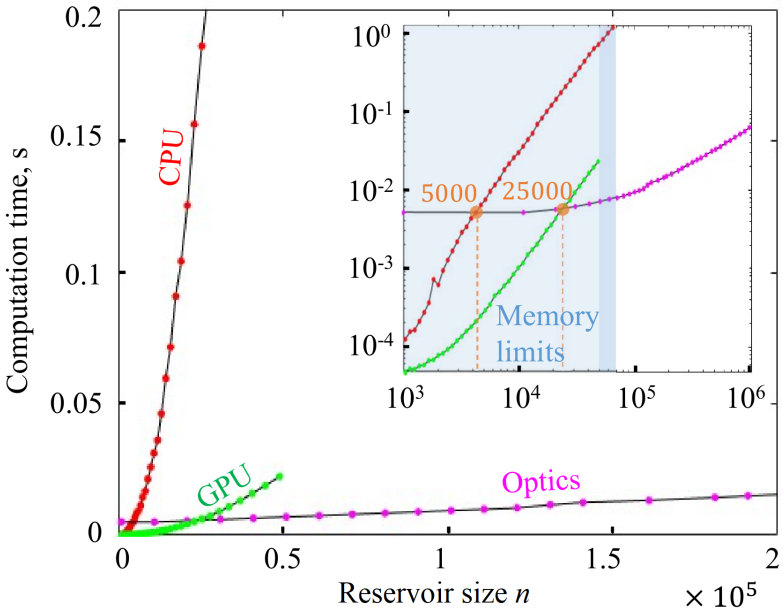


Figure 6.4: Speed benchmark of Reservoir Computing and memory limitations, on CPU, GPU and the optical implementation LCoS2. Inset shows the performance curves in a logarithmic scale extended for larger reservoir sizes. The turning points where the optical scheme starts to perform faster than the CPU and GPU correspond to $n \approx 5000$ and $n \approx 25000$, respectively.

amplitude or phase-only. This means that an encoding operation needs to be introduced on the reservoir and input data, it corresponds to the function g in (6.2). We assume that this g function operates componentwise to increase its speed of computation and simplicity.

In the case of phase encoding, g is simply defined by $g(x) = e^{i\pi x}$. x is between 0 and 1 after normalization and discretized in 8 bits, which correspond to 255 grey levels that are generally sufficient to avoid any loss of performance. In practice, there is no clear difference or drawback introduced by this encoding.

On the other hand, the binarization constraint introduced by the DMD is more critical as some information is irremediably lost. Thanks to the introduction of this encoding function, we can also change the dimensionality of the encoding and encode a reservoir activation on several DMD pixels. With this strategy, the function g encodes the given value of one camera pixel on n_{bin} binary DMD pixels. Higher values of n_{bin} increase the regularity of the encoding function, and make the RC dynamics smoother. Such an encoding also requires more DMD pixels to display the reservoir state. Hence, an efficient binarization scheme needs to balance between two constraints: a limited dimension expansion, to allow very large reservoir to be displayed on the DMD, and sufficient regularity and precision of the encoding.

There are several possibilities to define the encoding function g. As a separable function, i.e. operating componentwise, it is uniquely defined by the image of a single real number. After normalization of the number to encode, we assume that this number lies between 0 and 1. As there is a link between these binary embeddings and kernel, methods, we will characterize encodings based on their distance matri-

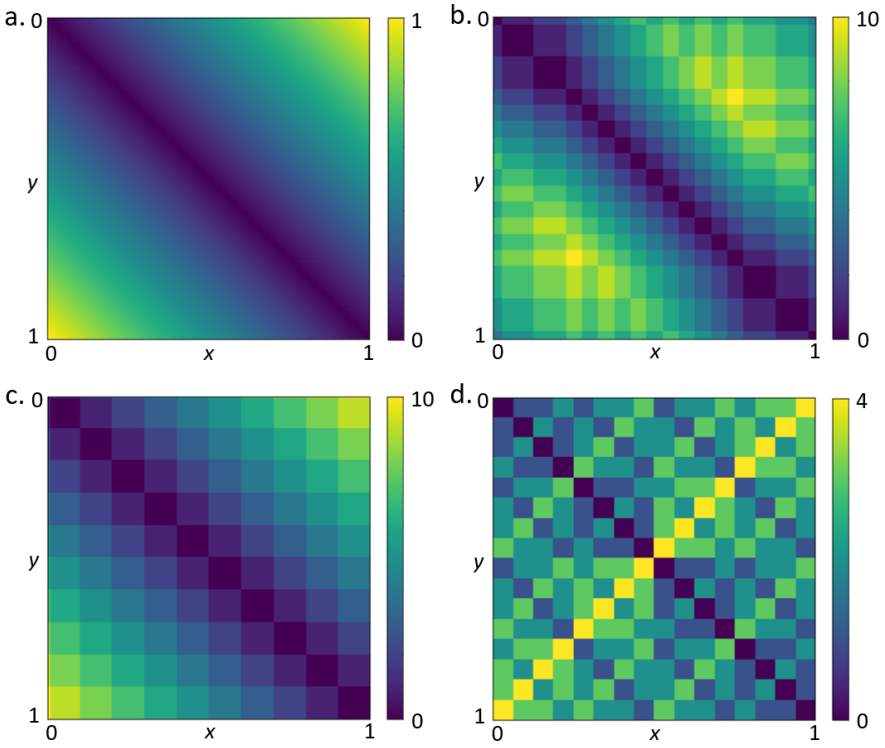


Figure 6.5: Distance matrices to compare encoding strategies. (a) Distance matrix $\|g(x) - g(y)\|$ for x and y between 0 and 1, with no encoding here $g(x) = x$ (for control). (b) Distance matrix for basket encoding (6.6) with $n_{\text{bin}} = 10$. (c) Distance matrix for threshold encoding (6.7) with $n_{\text{bin}} = 10$. (d) Distance matrix for base-2 encoding with $n_{\text{bin}} = 4$.

ces, defined by the set of $\|g(x) - g(y)\|$ for all pairs $(x, y) \in [0;1]^2$. These distances should be small close to the diagonal, where x and y are similar, and increase further away from the diagonal.

Inspired by the Random Binning Features developed by Rahimi and Recht [22], we propose here an encoding function g, called basket encoding, where each component $g_i(x)$ for $i = 1, \ldots, n_{\text{bin}}$ is defined by:

$$g_i(x) = \begin{cases} 1, & \text{if } x \in [c_i - s, c_i + s] \\ 0, & \text{else} \end{cases} \tag{6.6}$$

where the centers and size of the bins are defined by $c_i = \frac{2i-1}{2n_{\text{bin}}}$ and $s = \frac{2\lfloor n_{\text{bin}}/2 \rfloor - 1}{4n_{\text{bin}}}$ respectively. These values are chosen to obtain a larger number of different binary encodings while keeping a regular distance matrix.

Another binarization strategy, previously used in [27] and referred to as threshold encoding, is to encode the input by using a set of uniformly-spaced thresholds:

$$g'_i(x) = \begin{cases} 1, & \text{if } x > t_i \\ 0, & \text{else} \end{cases} \tag{6.7}$$

where the thresholds are defined by $t_i = \frac{i}{n_{\text{bin}}}$ for $i = 1, \ldots, n_{\text{bin}}$. This scheme is also regular and can be used for RC, but the number of different binary encodings is smaller

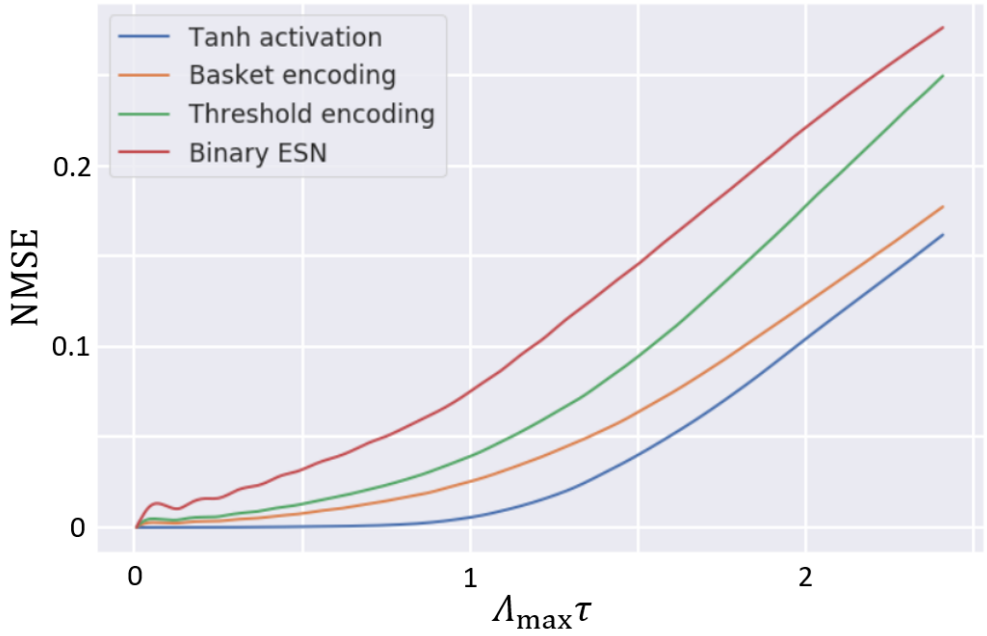


Figure 6.6: NMSE of Mackey-Glass prediction with RC using different encoding strategies: (a) tanh activation, (b) basket encoding defined in (6.6), (c) threshold encoding defined in (6.7), and (d) binary ESN of [27]. Each curve is an average of 5 realizations (numerical simulation), reservoir sizes is 512 for all cases with binary encoding dimension equal to 10 for (b) and (c).

in this case (10 possible binary encodings for threshold encoding compared to 15 for basket encoding). As a result, for a fixed binary encoding dimension n_{bin}, it will contain less information about the original real-value.

Fig. 6.6 presents the distance matrices for these three different binarization strategies. We observe that both basket encoding and threshold encoding are well-behaved, as distances close to the diagonal, which correspond to close original real values, are small while the distance increases smoothly as the original values get further apart. Additionally, the basket encoding provides more precision for a given bit depth, here $n_{\text{bin}} = 10$.

For comparison, we also present the distance matrix using the representation in base 2. This binary encoding is the one used in the memory of a computer, it is the most compact one as n_{bin} represent $2^{n_{\text{bin}}}$ different numbers. However, the computer implicitly differentiates bits according to their position, from the most-significant to the least-significant bit; in RC all bits should have the same importance. For example, 7 and 8 (or $\frac{7}{16}$ or $\frac{8}{16}$ after normalization) are very close but their binary encodings in base 2, 0111 and 1000, are very different.

Encoding represents an unavoidable step in analog computing, and we framed it here as a pre-processing kernel. This point of view should be applicable to most element-wise encodings performed on the original data. Moreover, it fits well in the framework of the next chapter with the kernel limit of Reservoir Computing, where the whole algorithm may be described as a compositional kernel.

6.3 Results

6.3.1 Chaotic time series prediction

Since Reservoir Computing is initially based on a Recurrent Neural Network, it is mainly used to process sequential data. One application is particularly promising in chaotic time series prediction, as Reservoir Computing has demonstrated state-of-the-art performance compared to other strategies [17], [18]. We will introduce the two chaotic systems that will be used for later benchmarks.

Mackey and Glass proposed in 1977 their famous equation to model physiological feedback systems [160]:

$$\frac{du(t)}{dt} = \beta \frac{u(t-\tau)}{1+u(t-\tau)^n} - \gamma u(t) \tag{6.8}$$

The first term corresponds to a delayed response of the system, which tends to 0 as u tends to either 0 or infinity to keep the model realistic, while the second term can be interpreted as a classical decay with rate γ. This time-delay differential equation, in appearance simple, displays chaotic behavior for certain ranges of parameters, for example $\beta = 0.2$, $\gamma = 0.1$, $\tau = 17$, $n = 10$ as in [138]. The maximal Lyapunov exponent in this case is $\Lambda_{\max} = 0.006$. In general, the Lyapunov exponent gives a measure for the total predictability of a system, it characterizes quantitatively the rate of separation of infinitesimally close trajectories in dynamical system, namely, the minimum amount of the time for which trajectories are diverging by a factor of e. This chaotic time series is one of the standard models used to test Reservoir Computing algorithms [12].

The Kuramoto-Sivashinsky (KS) equation is a model of nonlinear partial differential equation frequently encountered in the study of nonlinear chaotic systems with intrinsic instabilities, such as wave propagation in chemical reaction-diffusion systems, the velocity of laminar flame front instabilities, thin fluid film flow down inclined planes and hydrodynamic turbulence [161], [162]. The one-dimensional Kuramoto-Sivashinsky partial differential equation is:

$$u_t = -uu_x - u_{xx} - u_{xxxx} \tag{6.9}$$

where we assume the scalar field $u = u(x,t)$ is periodic with period L. This defines a spatio-temporal chaotic system, which is a much more complex object than the Mackey-Glass differential equation. When not explicitly defined, the period L will be set to 22, which corresponds to a Lyapunov exponent of $\Lambda = 0.043$.

6.3.2 First prediction results

We first focused on the simpler Mackey-Glass series, as it is only a scalar time series. Figure 6.7c shows one of the successful prediction results with the optical system LCoS1. The dashed line is the test time series fed to the reservoir until $t_0 = 0$. Afterwords, the algorithm has been switched into prediction mode (solid line). The temporal axis is normalized by the maximal Lyapunov exponent $\Lambda_{\max}$. To obtain such a

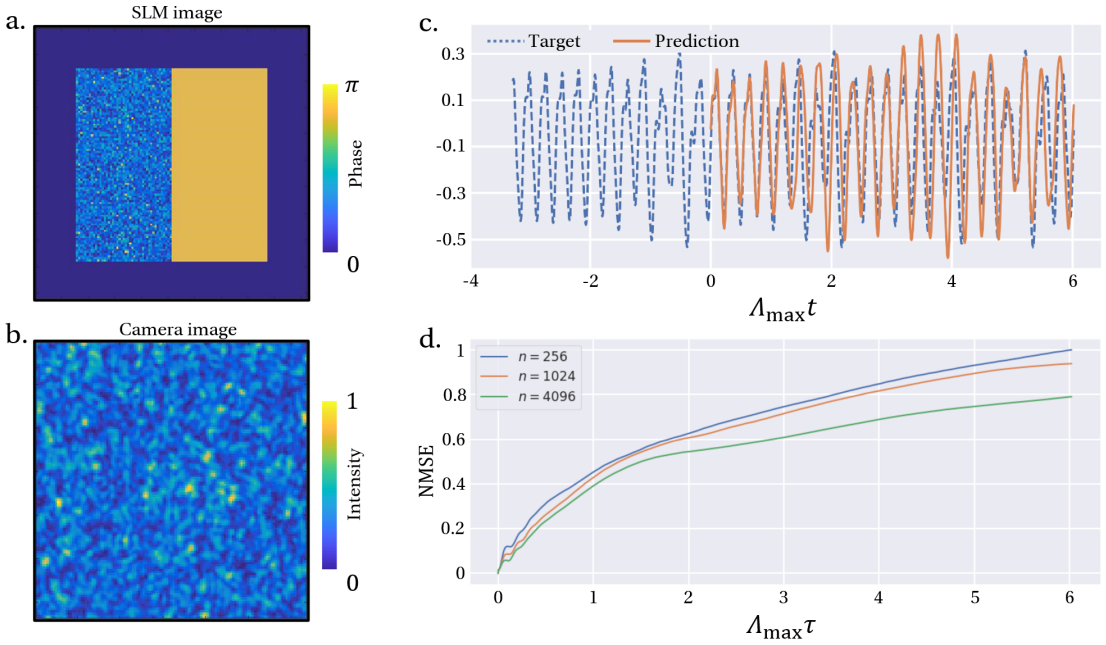


Figure 6.7: Optical Reservoir Computing for prediction of the Mackey-Glass system, using the optical system LCoS1 (with an LC-SLM). (a) Image displayed on the SLM. (b) Image recorded on the camera. (c) Chaotic system prediction: the value of the time series is fed to the reservoir, up to a point at which the reservoir is used to predict the future. Time on the x-axis is normalized by the Lyapunov exponent. (d) Normalized Mean Square Error (NMSE) as a function of the prediction time, for different reservoir sizes n.

result, various hyperparameters need to be taken into account to improve the prediction performance, like the relative weights between the input and the reservoir state. They are discussed at more at length in the appendix, along with how to readout the reservoir states from the camera images.

As a benchmark of the prediction performance we use the NMSE (Normalized Mean Square Error) for N values of t_0 defined by:

$$\text{NMSE} = \frac{E}{N\sigma^2}, \tag{6.10}$$

where E is the squared error on the prediction task presented in Equation (6.4), normalized by σ^2 the variance of the Mackey-Glass time series. Consequently, the smaller the NMSE value, the better the prediction performance is. In order to obtain smoother NMSE curves, we calculated its average over ten independent experiments with nine hundred test time series for each. The performance curves we obtain with the SLM optical implementation for different reservoir sizes $n = 256,\ 1024,\ 4096$ are collected in Figure 6.8d. As one can see, the larger the reservoir size, the better the algorithm performance. The small oscillations of the NMSE from short prediction times originate from the oscillations of the Mackey-Glass time series, which are faster than the Lyapunov time. For longer prediction times, the task becomes harder which explains why the performance reaches a kind of plateau. The task becomes exponentially harder since the Mackey-Glass time series is chaotic and the Lyapunov exponent defines the typical time scale for chaotic divergence. It should converge to 1 for much longer time series as the reservoir only outputs the mean value of Mackey-Glass.

6.3.3 Comparing DMD and LC-SLM implementations

Now that we have proven the feasibility of Reservoir Computing in optics, we investigate the difference in performance between the two SLM technologies. We present in Figure 6.8d the NMSE of binary Reservoir Computing, binary Echo-State Network and the phase Reservoir Computing obtained with an LCoS-SLM. Reservoir sizes are set at 512 for binary Reservoir Computing, and 5120 for binary ESN to obtain a fair comparison as the basket encoding expands the binary dimension 10 times.

We observe that thanks to the basket encoding, this new binarized version of Reservoir Computing is performing better than the previous binary ESN implementation. Additionally, we see that this experimental binary Reservoir Computing is performing better than the other experimental implementation of Reservoir Computing based on phase modulation. The gap in performance between DMD and LCoS-SLM implementations is probably due to a difference in stability and SNR of the optical devices, which is higher for the one developed by LightOn. On the other hand, binary Echo-State Networks do not perform as well due to the binarization operation.

This shows that designing custom optical setups with very few moving parts is beneficial, as it improves the stability of the system. Then, having an appropriate encoding strategy is improving the prediction performance considerably. It shows the information loss due to binarization is quite detrimental for our particular task, as chaotic systems are typically very sensitive to small perturbations.

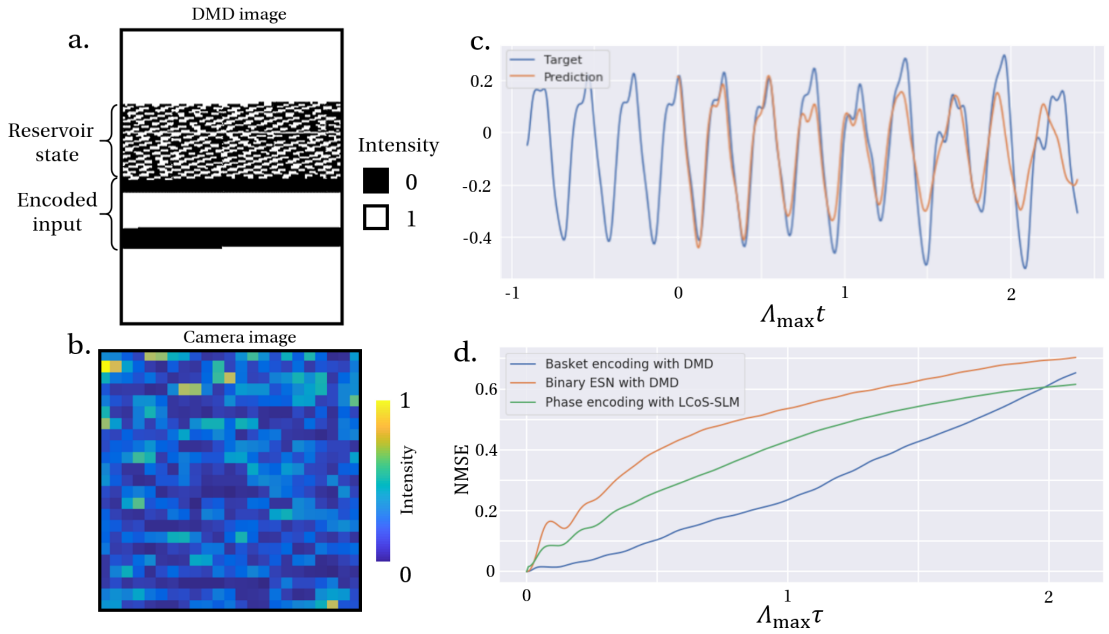


Figure 6.8: Optical Reservoir Computing for prediction of the Mackey-Glass system, using the optical system A (with a DMD). (a) Image displayed on the DMD. (b) Image recorded on the camera. (c) Chaotic system prediction: the value of the time series is fed to the reservoir, up to a point at which the reservoir is used to predict the future. Time on the x-axis is normalized by the Lyapunov exponent. (d) Normalized Mean Square Error (NMSE) as a function of the prediction time, for two different DMD encoding strategies and the LC-SLM of the optical system B.

6.3.4 Optimizing Optical Reservoir Computing

We now test Optical Reservoir Computing on the Kuramoto-Sivashinsky system. As a spatiotemporal chaotic system, this prediction task is more challenging. A recent study has reported the very promising performance of Reservoir Computing for this particular task [17]. We would like to stress the interest that this last work has raised on the community, as it has shown the intriguing effectiveness of Reservoir Computing for a challenging prediction task. This is the reason why we also used this benchmark for our Optical Reservoir Computing implementation.

Our work on this project has been reported in [54]. To increase the performance of the algorithm, we built a dedicated optical system for this optical computing strategy, following the principle as before but with improved stability and enabling us to increase further the reservoir dimension, up to $n = 50,000$. Still, stability was the limiting factor as decorrelation of the scattering medium prevented us from performing experiments for more than one hour.

We also used a trick referred to as autonomous dynamics of the reservoir to increase the prediction performance. To put it simply, we first train the reservoir to predict the next time-step only, and for prediction feed this next-time step prediction as input to iterate the reservoir. With the new reservoir state, another next-time step prediction is performed and this process is continued recursively. As such, the reservoir becomes an autonomous dynamical system that we hope mimics the dynamics of the initial chaotic system. Empirical evidence of the effectiveness of this method is presented in Appendix D.

We thus prove here that optical computing may also tackle state-of-the-art benchmarks, with a prediction up to 2 Lyapunov exponents on the Kuramoto-Sivashinsky system. This last step required to build an improved optical system with an LCoS-SLM and large reservoir sizes. To improve performance further and compare with [17], we would need to improve the optical stability further and study the electronic noise of our particular Spatial Light Modulator, which we believe is the limiting factor here.

Code to generate data and run the models is available at `https://github.com/jon-dong/reservoir-computing-python`.

6.4 Discussion

6.4.1 In optical computing

This research project provides an example of application of the optical computing strategy explained in the previous chapter: using optical scattering for random matrix multiplications. Optics speeds up this computation by at least an order of magnitude and allows to build larger and larger networks. It already shows how optics can complement electronics, accelerating specific operations to enable the best out of both worlds. Optics is perfectly suited for large-scale parallel operations, while electronics is more versatile.

However, implementing competitive optical computing systems is no easy task.

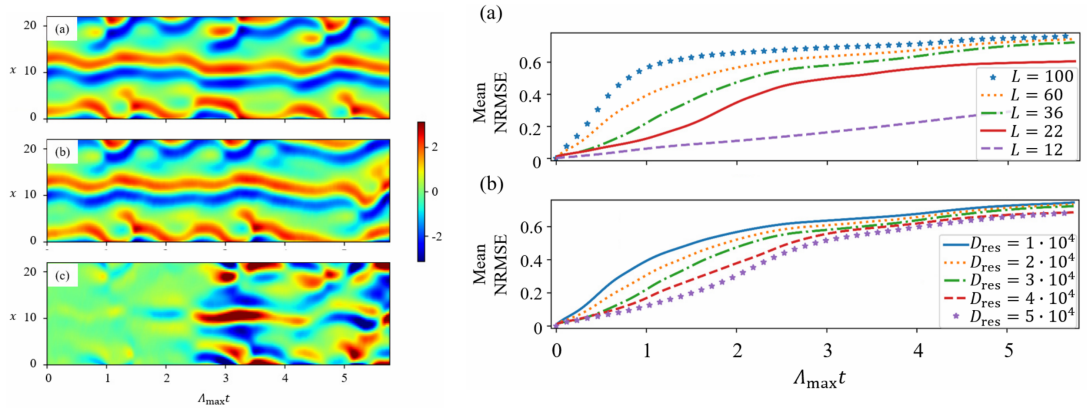


Figure 6.9: Optical Reservoir Computing for prediction of the Kuramoto-Sivashinsky system, using optical system C (with an LC-SLM). (Left top) True prediction (target). (Left middle) Prediction using Reservoir Computing. (Left bottom) Difference between the target and prediction. (a) Normalized Root Mean-Square Error (NMRSE) for different sizes of the Kuramoto-Sivashinsky system, for $D_{\text{res}} = 5 \cdot 10^4$. (b) NMRSE for different reservoir sizes, for $L = 22$.

We also had to face the challenges of analog computing with the detrimental impact of stability and noise. Thankfully, the presented algorithms work in high-dimension and therefore are relatively robust against noise, still enabling us to implement performant algorithms. On the other hand, the encoding step, i.e. how to send data in the optical domain, remains an important question to tackle, due to constraints introduced by the modulation devices. Our first experimental implementation was not competitive on the Mackey-Glass dataset if we compare quantitatively Fig. 6.6 with Fig. 6.8, while results on the Kuramoto-Sivashinsky model from Fig. 6.9 are competitive with reported state-of-the-art results.

To put in perspective this work, other optical computing applications are investigated in the startup LightOn. For our particular case of Reservoir Computing, one may wonder what benefits optical computing brings for real-world applications. As a counterexample, increasing the reservoir dimension is interesting as a research endeavor but merely a technical achievement. Nevertheless, working with very large reservoir sizes may improve the robustness of the prediction and the energy consumption of the optical device is much lower than a GPU. In the future, as energy efficiency and high computation power will be two important metrics to balance, one could envision devices where different operation are performed on specialized hardware, based on electronics or optics.

6.4.2 In Reservoir Computing

One particular feat of Reservoir Computing is the already-existing diversity of physical implementations, leveraging the flexibility of the framework to search for an efficient implementation. This work builds on this quest and proposes another approach using free-space propagation and multiple light scattering. Our approach is very scalable compared to other techniques, where there is usually a trade-off be-

tween dimension and speed. Moreover, our approach has shown for the first time that physical implementations can perform well on a state-of-the-art application, the prediction of the chaotic Kuramoto-Sivashinsky system.

Changing the computation paradigm dramamatically modifies the computational bottlenecks. Here the linear regression is the limiting factor to scale to even larger reservoir sizes for two different reasons. First, we need to store in memory the successive reservoir states for the linear regression. Second, the generic algorithm we used for linear regression becomes the bottleneck in terms of computational time. It would be interesting to investigate whether an online training algorithm would be effective when paired with Optical Reservoir Computing.

Finally, as optics enables very large reservoir sizes, one may wonder what this change brings and what happens when the reservoir size goes to infinity. We tackle this question in the next chapter, presenting a theoretical work with no application in optics. In scientific research, impact often does not correlate well with effort. All the results presented in this chapter may be put in perspective with the message of the following chapter: the quest towards very large size Reservoir Computing does not require optics, one may also replace the dense random matrix by a structured transform, a very light and efficient operation.

CHAPTER 7

Recurrent Kernels and Structured Transforms

[1]We have seen in the previous chapter how to accelerate Reservoir Computing in optics, enabling very large reservoir sizes. One may then wonder: how beneficial is it to increase the reservoir size? In particular, is there an asymptotic limit for very large reservoir sizes? We will show in this chapter that indeed, the asymptotic limit exists and corresponds to an explicit Recurrent Kernel.

This study will be the opportunity to make explicit the link between the Random Features presented in Chapter 5 [22], [23], [163]–[165] and Reservoir Computing, our topic of interest since Chapter 6. This natural association will result in the introduction of Structured Reservoir Computing, a software acceleration technique for Reservoir Computing.

To accelerate and scale-up the computation of Random Features, one can use structured transforms [166], [167], providing a very efficient method for kernel approximation: structured transforms such as the Fourier or Hadamard transforms can be computed in $O(n \log n)$ complexity. We show in this work that structured transform are equally well-suited to Reservoir Computing: by simply replacing the dense random matrix by a succession of structured transforms and random diagonal matrices, we accelerate by orders of magnitude the Reservoir Computing computation without any compromise in performance. The so-called Structured Reservoir Computing is equivalent to Reservoir Computing, as they both represent finite dimensional approximations of a Recurrent Kernel.

We believe this approach is fundamentally important for the study of Reservoir Computing, both in theory, to revisit many results in Reservoir Computing, and in pratice, thanks to faster and more efficient algorithms. No optical computing system is present in this chapter, but optical computing served as an inspiration to study the high-dimensional limit of Reservoir Computing.

7.1 The Recurrent Kernel limit

7.1.1 Definition

As RC embeds input data in a high-dimensional reservoir, it has already been linked with kernel methods [15], but this heuristic interpretation has not been pursued further at the time. In 2012, a standalone study derived the explicit formula of the cor-

[1]This project was an idea that gradually emerged during the PhD, and we worked together with Ruben Ohana (PhD student with Pr. Florent Krzakala) to make it concrete and propose a competitive implementation.

responding recurrent kernel associated with RC [168], this important result meaning the infinite-width limit of RC is a deterministic Recurrent Kernel (RK). Still, no theoretical study of convergence towards this limit has been conducted previously, which is what we are going to propose here. With both theoretical and numerical evidence, it will bear insights on the conditions for Reservoir Computing to converge towards its limit. In this chapter, d denotes the input dimension, N the reservoir size, n and m the number of samples in the training and testing set respectively.

For this particular study, we will distinguish two major classes of kernel functions: translation-invariant **(TI)** kernels and rotation-invariant **(RI)** kernels. We will consider TI kernels of the form:

$$k(u,v) = k(\|u-v\|_2^2) \tag{7.1}$$

and RI kernels of the form:

$$k(u,v) = k(\langle u,v\rangle) \tag{7.2}$$

We have already introduced in Chapter 5 the Random Fourier Features [22] to approximate any TI kernel and [169] proposes a method to construct Random Features to approach RI kernels. Note that the examples presented in the following are generally not rigorously TI or RI kernels as they depend on $\|u\|$, $\|v\|$, and $\langle u,v\rangle$. For more examples, a detailed taxonomy of Random Features can be found in [170]. TI and RI kernels represent the 2 canonical cases from which other Recurrent Kernel formulas may be derived.

To draw the link between Reservoir Computing and Random Features, let's recall the update equation (7.4) here:

$$x^{(t+1)} = \frac{1}{\sqrt{N}} f(W_r x^{(t)} + W_i i^{(t)}) \tag{7.3}$$

In this equation, $x^{(t+1)}$ can be interpreted as a Random Feature embedding of a vector $[x^{(t)}, i^{(t)}]$ (of dimension $q = N + d$) multiplied by an i.i.d. random matrix $W = [W_r, W_i]$. This means the inner product between two reservoirs $x^{(t)}, y^{(t)}$ driven respectively by two inputs $i^{(t)}$ and $j^{(t)}$ converges to a deterministic kernel as N tends to infinity:

$$\left\langle x^{(t+1)}, y^{(t+1)} \right\rangle \approx k\left(\left[x^{(t)}, i^{(t)}\right], \left[y^{(t)}, j^{(t)}\right]\right) \tag{7.4}$$

As explained previously, this kernel depends on the choice of f and the distribution of W_r and W_i. To simplify this general form with a concatenation, one can then derive the corresponding formulas for TI and RI kernels, by denoting $l^{(t)} = \sigma_i^2 \left\langle i^{(t)}, j^{(t)} \right\rangle$ and $\Delta^{(t)} = \sigma_i^2 \left\| i^{(t)} - j^{(t)} \right\|^2$:

$$k\left(\left[x^{(t)}, i^{(t)}\right], \left[y^{(t)}, j^{(t)}\right]\right) = k\left(\sigma_r^2 \left\langle x^{(t)}, y^{(t)} \right\rangle + l^{(t)}\right) \quad \text{(RI)} \tag{7.5}$$

$$= k\left(\sigma_r^2 \left\| x^{(t)} - y^{(t)} \right\|^2 + \Delta^{(t)}\right) \quad \text{(TI)} \tag{7.6}$$

Looking at Eq. (7.5) and (7.6), we notice the kernel at time t depends on approximations of kernels at previous times in a recursive manner. Here, we introduce Recurrent Kernels to remove the dependence in $x^{(t)}$ and $y^{(t)}$. We suppose for the sake of simplicity $x^{(0)} = y^{(0)} = 0$ and define RI recurrent kernels as:

$$\begin{cases} k_1\left(l^{(0)}\right) = k\left(l^{(0)}\right) \\ k_{t+1}\left(l^{(t)}, ..., l^{(0)}\right) = k\left(\sigma_r^2 k_t\left(l^{(t-1)}, ..., l^{(0)}\right) + l^{(t)}\right), \quad \text{for } t \in \mathbb{N}^* \end{cases} \tag{7.7}$$

Similarly for TI recurrent kernels with Random Fourier Features, exploiting the property that $\|x^{(t)}\|^2 = \|y^{(t)}\|^2 = 1$ with random Fourier Features:

$$\begin{cases} k_1\left(\Delta^{(0)}\right) = k\left(\Delta^{(0)}\right) \\ k_{t+1}\left(\Delta^{(t)}, ..., \Delta^{(0)}\right) = k\left(\sigma_r^2\left(2 - 2k_t\left(\Delta^{(t-1)}, ..., \Delta^{(0)}\right)\right) + \Delta^{(t)}\right), \quad \text{for } t \in \mathbb{N}^* \end{cases} \tag{7.8}$$

These Recurrent Kernel definitions describe hypothetical asymptotic limits of large-dimensional Reservoir Computing and we will study next the convergence towards this limit.

7.1.2 Convergence theorem

Our first main result is a convergence theorem of Reservoir Computing to its kernel limit. We want to bound the deviation between the kernel limit and an actual scalar product between two reservoir states. This result will be probabilistic due to the randomness of the weight matrix W. Although convergence is observed in a larger range of cases, this theorem represents a statement of what is achievable using common theoretical tools introduced for Random Features. Several assumptions will be necessary:

- The kernel function k is Lipschitz-continuous with constant L, i.e. $|k(a) - k(b)| \leq L|a - b|$. This assumption is classical in Reservoir Computing to study the algorithm stability, but here we assume the Lipschitz-continuity of the kernel function k and not the activation function f.
- The random matrices W_r and W_i are resampled for each t to obtain uncorrelated reservoir updates: $x^{(t+1)} = \frac{1}{\sqrt{N}} f\left(W_r^{(t)} x^{(t)} + W_i^{(t)} i^{(t)}\right)$. This assumption is necessary to remove correlations that are challenging to deal with in a probabilistic setting. Resampling the weight matrices at each iteration is possible in Reservoir Computing but would be painfully slow.
- The function f is bounded by a constant κ almost surely:

$$\left| f\left(W_r^{(t)} x^{(t)} + W_i^{(t)} i^{(t)}\right) \right| \leq \kappa \tag{7.9}$$

Theorem 8. *(Rotation-invariant kernels) For the RI recurrent kernel defined in Eq. (7.7), under the assumptions detailed above, and with $\Lambda = \sigma_r^2 L$. For all $t \in \mathbb{N}$, the following inequality is satisfied for any $\delta > 0$ with probability at least $1 - 2(t+1)\delta$:*

$$\left| \langle x^{(t+1)}, y^{(t+1)} \rangle - k_{t+1}\left(l^{(t)}, ..., l^{(0)}\right) \right| \leq \frac{1 - \Lambda^{t+1}}{1 - \Lambda} \Theta(N) \qquad \text{if} \quad \Lambda \neq 1 \tag{7.10}$$

$$\leq (t+1)\Theta(N) \qquad \text{if} \quad \Lambda = 1 \tag{7.11}$$

with $\Theta(N) = \frac{4\kappa^2 \log\frac{1}{\delta}}{3N} + 2\kappa^2\sqrt{\frac{2\log\frac{1}{\delta}}{N}}$.

Proof. We are going to prove this property by applying the Bernstein inequality recursively.

We use the following Proposition (Theorem 3 of [171] restated in Proposition 1 of [163]):

Lemma 1. *(Bernstein inequality for a sum of random variables). Let $X_1, ..., X_N$ be a sequence of i.i.d. random variables on $\mathbb{R}$ with zero mean. If there exist $R, S \in \mathbb{R}$ such that $|X_i| \leq R$ almost everywhere and $\mathbb{E}\left[X_i^2\right] \leq S$ for $i \in \{1, ..., N\}$, then for any $\delta > 0$ the following holds with probability at least $1 - 2\delta$:*

$$\left| \frac{1}{N} \sum_{i=1}^{N} X_i \right| \leq \frac{2R \log \frac{1}{\delta}}{3N} + \sqrt{\frac{2S \log \frac{1}{\delta}}{N}} \tag{7.12}$$

Under the assumptions, Lemma 1 yields with probability greater than $1 - 2\delta$:

$$\left| \langle x^{(t+1)}, y^{(t+1)} \rangle - k([x^{(t)}, i^{(t)}], [y^{(t)}, j^{(t)}]) \right| \leq \frac{4\kappa^2 \log \frac{1}{\delta}}{3N} + 2\kappa^2 \sqrt{\frac{2 \log \frac{1}{\delta}}{N}} = \Theta(N) \tag{7.13}$$

It means the larger the reservoir, the more Random Features N we sample, and the more the inner product of reservoir states concentrates towards its expectation value, at a rate $O(1/\sqrt{N})$. We now apply this inequality recursively to complete the proof, based on the observation that both Eq. (7.10) and (7.11) are equivalent to:

$$\left| \langle x^{(t+1)}, y^{(t+1)} \rangle - k_{t+1}(l^{(t)}, ..., l^{(0)}) \right| \leq (1 + \Lambda + \Lambda^2 + ... + \Lambda^t)\Theta(N) \tag{7.14}$$

For $t = 0$, provided $x^{(0)} = y^{(0)} = 0$, we have, according to Eq. 7.13, with probability at least $1 - 2\delta$:

$$\left| \langle x^{(1)}, y^{(1)} \rangle - k_1(l^{(0)}) \right| \leq \Theta(N) \tag{7.15}$$

For any time $t \in \mathbb{N}^*$, let us assume the following event A_t is true with probability $\mathbb{P}(A_t) \geq 1 - 2t\delta$:

$$\left| \langle x^{(t)}, y^{(t)} \rangle - k_t(l^{(t-1)}, ..., l^{(0)}) \right| \leq (1 + ... + \Lambda^{t-1})\Theta(N) \tag{7.16}$$

Using the Lipschitz-continuity of k, this inequality is equivalent to:

$$\left| k(\sigma_r^2 \langle x^{(t)}, y^{(t)} \rangle + l^{(t)}) - k(\sigma_r^2 k_t(l^{(t-1)}, ..., l^{(0)}) + l^{(t)}) \right| \leq (\Lambda + ... + \Lambda^t)\Theta(N) \tag{7.17}$$

With Eq. (7.13), the following event B_t is true with probability $\mathbb{P}(B_t) \geq 1 - 2\delta$:

$$\left| \langle x^{(t+1)}, y^{(t+1)} \rangle - k\left(\sigma_r^2 \langle x^{(t)}, y^{(t)} \rangle + l^{(t)}\right) \right| \leq \Theta(N) \tag{7.18}$$

Summing Eq. (7.17) and (7.18), with the triangular inequality and a union bound, the following event A_{t+1} is true with probability $\mathbb{P}(A_{t+1}) \geq \mathbb{P}(B_t \cap A_t) = \mathbb{P}(B_t) + \mathbb{P}(A_t) - \mathbb{P}(B_t \cup A_t) \geq 1 - 2\delta + 1 - 2t\delta - 1 \geq 1 - 2(t+1)\delta$:

$$\left| \langle x^{(t+1)}, y^{(t+1)} \rangle - k_{t+1}(l^{(t)}, ..., l^{(0)}) \right| \leq (1 + ... + \Lambda^t)\Theta(N) \tag{7.19}$$

□

This theorem controls the rate of convergence of Reservoir Computing towards the Recurrent Kernel limit in $1/\sqrt{N}$. Moreover, as it exploits Lipschitz-continuity, it describes three regimes similar to the transition in Reservoir Computing between stability and chaos. (1) A stable regime where Reservoir Computing is guaranteed to converge the RK limit. (2) A critical regime where the bound diverges linearly with

time. (3) An unstable regime where the bound diverges exponentially with time. On the other hand, the theorem does not imply divergence in the regimes (2) and (3), and we have observed that convergence is much more robust in pratice. The three regimes are only observed with unbounded activation functions such as the Rectified Linear Unit (ReLU).

A similar convergence bound for TI recurrent kernels can be obtained:

Theorem 9. *(Rotation-invariant kernels) For the RI recurrent kernel defined in Eq. (7.8), under the assumptions detailed above, and with* $\Lambda = 2\sigma_r^2 L$ *(note the factor* 2 *compared to Theorem 8). For all* $t \in \mathbb{N}$*, the following inequality is satisfied for any* $\delta > 0$ *with probability at least* $1 - 2(t+1)\delta$*:*

$$\left|\left\langle x^{(t+1)}, y^{(t+1)}\right\rangle - k_{t+1}\left(\Delta^{(t)}, ..., \Delta^{(0)}\right)\right| \leq \frac{1-\Lambda^{t+1}}{1-\Lambda}\Theta(N) \qquad \text{if} \quad \Lambda \neq 1 \tag{7.20}$$

$$\leq (t+1)\Theta(N) \qquad \text{if} \quad \Lambda = 1 \tag{7.21}$$

with $\Theta(N) = \frac{4\kappa^2 \log\frac{1}{\delta}}{3N} + 2\kappa^2\sqrt{\frac{2\log\frac{1}{\delta}}{N}}$.

7.2 Structured Reservoir Computing

7.2.1 Structured Random Features

In the Random Features literature, it is common to use structured transforms to speed-up computations of random matrix multiplications [166], [167]. They have also been introduced for trained architectures, with Deep [172] and Recurrent Neural Networks [173]. Thanks to their structure, they can be implemented using a divide-and-conquer strategy to reduce their computational complexity to $O(N \log N)$. They are thus a valuable tool for efficient Machine Learning. For example, the Hadamard transform is defined recursively as:

$$\begin{cases} H_0 = 1 \\ H_p = \frac{1}{\sqrt{2}}\begin{pmatrix} H_{p-1} & H_{p-1} \\ H_{p-1} & -H_{p-1} \end{pmatrix} \end{cases} \tag{7.22}$$

It is thus only defined as a square matrix and for dimensions $q = 2^p$ a power of 2.

Several structured transforms have been proposed to replace the dense random matrix. [166] proposes the Fastfood transform:

$$V = \frac{\sqrt{q}}{\sigma} SHG\Pi HB \tag{7.23}$$

where S is a diagonal scaling matrix, G a diagonal random gaussian matrix, B a diagonal random binary matrix, Π a random permutation, and H the Hadamard transform. Note that V is never computed explicitly but the Hadamard transform H is applied on vectors using the Fast Hadamard Transform, multiplications by diagonal

matrices are element-wise operations, and permutations are performed on the vectors directly.

Instead, [167] replaces the dense random weight matrix W by a succession of Hadamard matrices H and diagonal random matrices D_i for $i = 1,2,3$ sampled from an i.i.d. Rademacher distribution:

$$W = \frac{\sqrt{q}}{\sigma} HD_1HD_2HD_3 \tag{7.24}$$

Among many other possible structured transforms such as based on the Fourier transform, this particular structured transform is used here for its simplicity. In particular, high-performance libraries as in [174] are easily available online. This transform provides the two main properties of a dense random matrix: mixing the activation of the neurons (Hadamard transform) and randomness (diagonal matrices). However, the theory to explain their unreasonable effectiveness is still lacking today [175].

7.2.2 Principle

The principle of Structured Reservoir Computing is very simple: since Reservoir is to Recurrent Kernels as Random Features are to kernel methods, we replace the random weight matrix W by Eq. (7.24). This is similar to the optical acceleration of Chapter 6, we accelerate the most expensive operation and all the rest of the pipeline, and in particular the linear regression step, remains unchanged.

To satisfy the constraints imposed by the nature of the structured transform, we add a zero padding to $\left[x^{(t)}, i^{(t)}\right] \in \mathbb{R}^{N+d}$ to increase its dimension to the nearest power of 2 greater than $N+d$. The output then has the dimension of $x^{(t+1)} \in \mathbb{R}^N$, and is obtained by subsampling result. This process keeps the orthogonality of the rows of W [167]. If we had to increase the dimensionality as with Random Features [167], [175], one can concatenate the result of several structured transforms for different realizations of the random matrices D_i, $i = 1,\ldots,3$.

7.2.3 Computational complexity

We want to discuss here in which regimes Recurrent Kernels and Structured Reservoir Computing are computationally efficient. To understand which algorithm to use for chaotic system prediction, we need to focus on the limiting operation in the whole pipeline of Reservoir Computing, the recurrent iterations. They correspond to Eq. (7.3) for RC/SRC and Eq. (7.7, 7.8) for RK.

This discussion will be focused on chaotic time series prediction, a task in which Reservoir Computing has proven to be particularly proficient. We have a time series of dimension d with a train and test datasets of lengths n and m respectively. A fundamental assumption in this case is the stationarity of the equation underlying the observed time series. At each time step corresponds a reservoir state, which can be used to predict the future of the time series.

On the other hand, other datasets are not stationary, such as the spoken digit recognition task with Reservoir Computing presented in [176]. For each time series of length T (audio recordings for example in [176]) corresponds a single output (digit).

	RC	SRC	RK
Forward	$O(nN^2)$	$O(n\boldsymbol{N}\log\boldsymbol{N})$	$O(n^2\tau)$
Training	$O(nN^2+N^3)$	$O(nN^2+N^3)$	$O(n^3)$
Prediction	$O(mN^2)$	$O(m\boldsymbol{N}\log\boldsymbol{N})$	$O(mn\tau)$
Memory	$O(nN+N^2)$	$\boldsymbol{O(nN)}$	$O(n^2+mn)$
Forward (parallel)	$O(nTN^2)$	$O(nT\boldsymbol{N}\log\boldsymbol{N})$	$\boldsymbol{O(n^2T)}$
Prediction (parallel)	$O(mTN^2)$	$O(mT\boldsymbol{N}\log\boldsymbol{N})$	$\boldsymbol{O(mnT)}$

Table 7.1: Computational and memory complexity of the three algorithms. SRC accelerates the forward pass and decreases memory complexity compared to conventional RC. The complexity of RK depends on the number of training and testing points and would be advantageous when $n \ll N$.

Thus, there are more Reservoir Computing iterations per output, and we will see that this setting is favorable for Recurrent Kernels. Here, we will denote by n the number of training samples and m the number of samples in the test set.

The exact computational and memory complexities of each step are described in Table 7.1.

Forward: In both Reservoir Computing and Structured Reservoir Computing, Eq. (7.3) needs to be repeated as many times as the length of the time series. For Reservoir Computing, it requires a multiplication by a dense $N \times N$ matrix, the associated complexity scales as $O(N^2)$. On the other hand, Structured Reservoir Computing uses a succession of Hadamard and diagonal matrix multiplications, reducing the complexity per iteration to $O(N\log N)$.

Recurrent Kernels need to recurrently compute Eq. (7.7, 7.8) for all pairs of input points. For chaotic time series prediction, this corresponds to a $n \times n$ kernel matrix for training, and another kernel matrix of size $n \times m$ for testing. To keep computation manageable, we use a well-known property in Reservoir Computing, called the Echo-State Property: the reservoir state should not depend on the initialization of the network, i.e. the reservoir needs to have a finite memory τ. Transposed in the Recurrent Kernel setting, it means we can fix the number of iterations of Eq. (7.7, 7.8) to τ, by using a sliding window to construct shorter time series if necessary.

For non-stationary tasks, reservoir states need to be computed in parallel for each sample. It thus adds a factor in the complexity of RC and SRC, while the complexity of RK is constant since the Gram matrices remain the same size. Since we will see that Recurrent Kernels are competitive even for chaotic time series prediction, it would be interesting to use them on non-stationary datasets.

Training requires, after a forward pass on the training dataset, to solve an $n \times N$ linear system for RC/SRC and a $n \times n$ linear system for RK. It is important to note SRC and RK do not accelerate this linear training step. We will use Ridge Regression with regularization parameter α to learn W_o, computed in practice using the Cholesky factorization of the Pytorch library.

Prediction in Reservoir Computing and Structured Reservoir Computing only requires the computation of reservoir states and multiplication by the learned output weights. Recurrent Kernels need to compute a new kernel matrix for every pair (i_r, j_q)

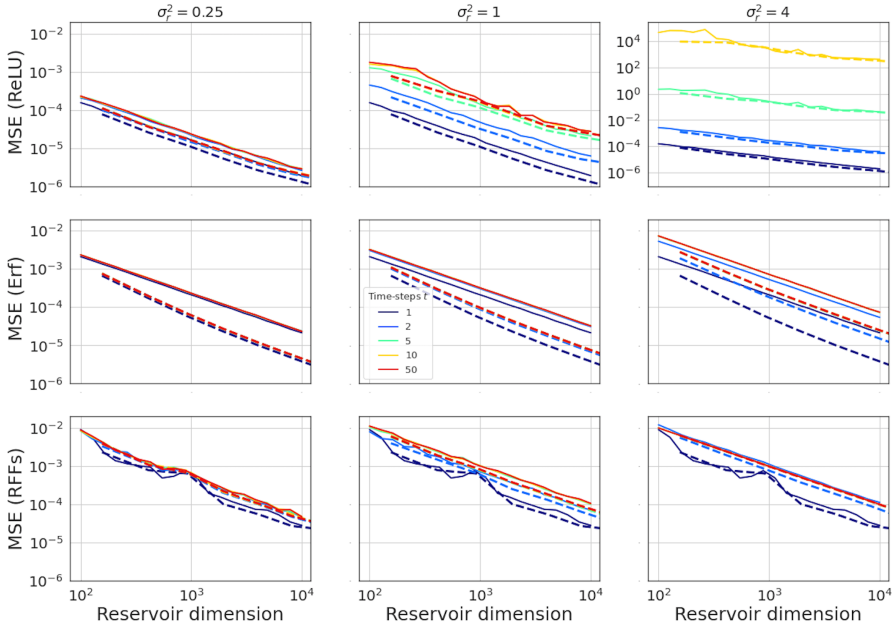


Figure 7.1: Convergence of Reservoir Computing towards its Recurrent Kernel limit for different variances of the reservoir weights σ_r^2 (columns), activation functions (lines: ReLU, Erf, RFFs) and times, for RC **(solid lines)** and SRC **(dashed lines)**. We observe that for the two bounded activation functions (Erf and RFFs), RC always converge towards the RK limit even at large times t. For ReLU, RC converges when $\sigma_r^2 = 0.25$ and 1, and diverges as t increases when $\sigma_r^2 = 4$. We also observe that SRC always yields equal or faster convergence than RC. The MSE decreases with an $O(1/N)$ scaling, which is consistent with the convergence rates derived in Theorem 8.

with i_r in the training set and j_q in the testing set. Note that the prediction step includes a forward pass on the test set, followed by a linear model.

7.3 Results

7.3.1 Numerical study of convergence

The previous theoretical study required three important assumptions that may not be valid for Reservoir Computing in practice. Moreover, there is still no rigorous proof on the convergence of Structured Random Features in the non-recurrent case due to the difficulty to deal with correlations between them. Thus, we numerically investigate whether convergence of RC and SRC towards the Recurrent Kernel limit is achieved in practice.

In Fig. 7.1, we numerically compute the Mean-Squared Error (MSE) between the inner products obtained with a Recurrent Kernel and RC/SRC for different number of neurons in the reservoir. We generate 50 i.i.d. gaussian input time series $i_k^{(t)}$ of length T, for $k = 1, \dots, 50$ and $t = 0, \dots, T-1$. Each time series is fed into 50 reservoirs that share the same random weights, for RC and SRC. We compute the MSE between inner products of pairs of final reservoir states $\langle x_k^{(T)}, x_l^{(T)} \rangle$ and the deterministic limit

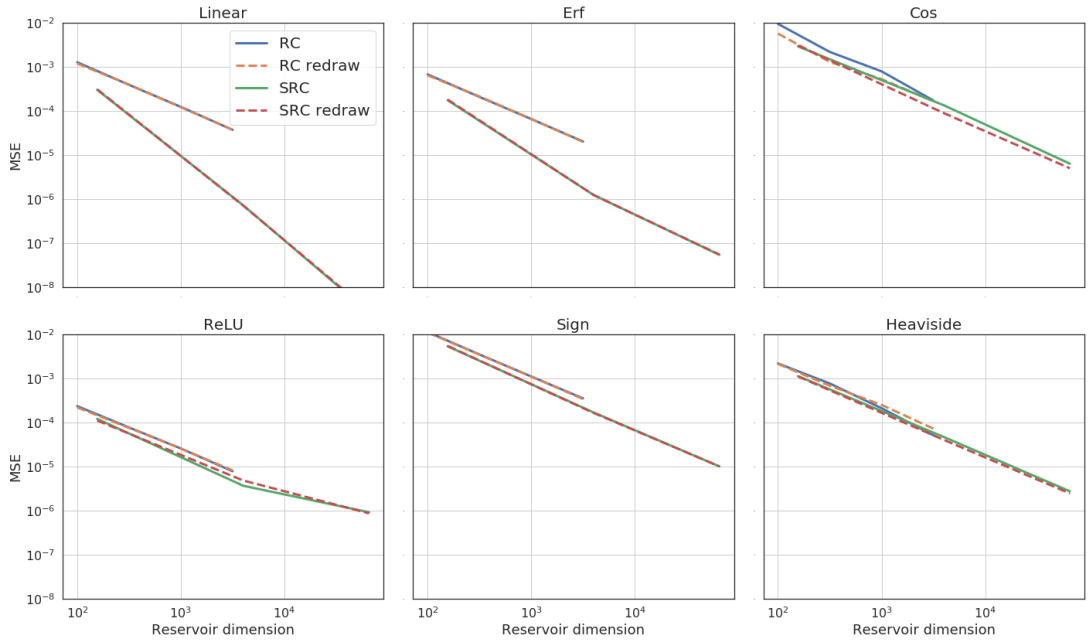


Figure 7.2: Mean-Squared error between the kernel matrix obtained with RC/SRC with the asymptotic kernel limit, with and without resampling the random matrices at each iteration, to test the independence hypothesis of the theorem for several activation functions and their corresponding recurrent kernels. We observe that the hypothesis does not seem to be necessary since RC and SRC without resampling also converge to the RK limit at sensibly the same speed.

obtained directly with $k_T(i_k^{(T-1)}, i_l^{(T-1)}, \dots, i_k^{(0)}, i_l^{(0)})$. The computation is vectorized to be efficiently implemented on a GPU. Three different activation functions, ReLU, the error function, and Random Fourier Features, have been tested with different variances of the reservoir weights. The larger the reservoir weights, the more unstable the reservoir dynamics becomes.

Nonetheless, convergence is achieved in a large variety of settings, even when the assumptions of the previous theorem are not satisfied. For example, the ReLU nonlinearity is not bounded and converges when $\sigma_r^2 \geq 1$. It is interesting to notice even for a large variance $\sigma_r^2 = 4$ do Reservoir Computing and Structured Reservoir Computing converge towards the RK limit for the second and third activation functions. This behavior has been consistently observed with any bounded f.

On the other hand, Structured Reservoir Computing seems to always converge faster than Reservoir Computing. We thus confirm in the recurrent case the intriguing effectiveness of Structured Random Features [175], that may originate from the orthogonality of the matrix W_r in SRC.

As a final remark, weight matrices in Fig. 7.1 were not redrawn as supposed in Section 7.1.2. This assumption was necessary as correlations are often difficult to take into account in a theoretical setting. This is important for Reservoir Computing as it would be unrealistically slow to draw new random matrices at each time step. In Fig. 7.2, we investigate the convergence with and without redrawing weights at each iteration, and this independence hypothesis does not seem to be necessary: conver-

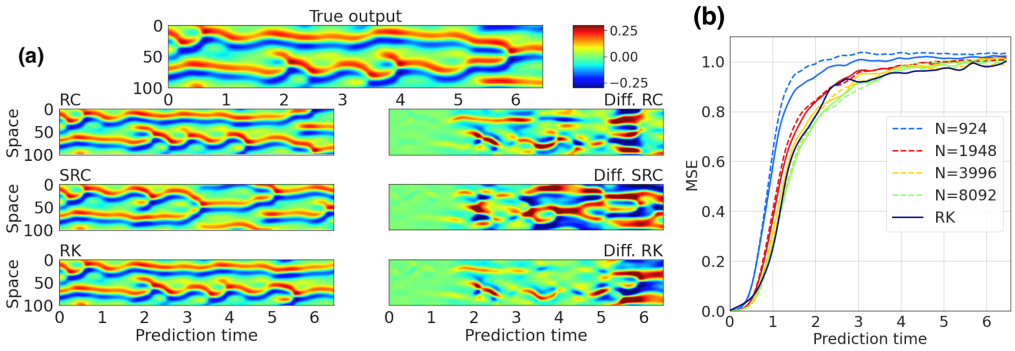


Figure 7.3: (a) Comparison of different algorithms for the prediction of the Kuramoto-Sivashinsky dataset. True output (top), predictions of RC/SRC/RK (left) and differences with the true output (right), with reservoirs in RC/SRC of size $N = 3{,}996$. We observe that each technique is able to predict up to a few characteristic times. (b) Mean-Squared Error as a function of the prediction time for RC **(full lines)**, SRC **(dashed lines)**, and RK **(black)**. For all the reservoir sizes considered, the performances of RC and SRC are very close and they converge for large dimensions to the RK limit.

gence is still achieved with fixed weight matrices. We show the Mean-Squared Error $\|K_1 - K_2\|_2^2/n^2$ between the kernel matrix K_1 from the explicit RK formula and $\hat{K}_2$ the one obtained with RC and SRC, with and without redrawing the random matrices at every timestep. Each kernel matrix is of size 50×50, as we use $n = 50$ random i.i.d gaussian input time series of dimension 50 and time length 10. Each curve is an average over 10 realizations and the reservoir scale is set to $\sigma_r^2 = 0.25$ to ensure stability. We observe that convergence is still achieved when resampling the weights at each iteration, and speed of convergence is not significantly different than for the fixed random matrix case. Thus convergence seems to be much more robust in practice, and this may call for further theoretical studies.

7.3.2 Chaotic system prediction

The same hyperparameters for RC, SRC, and RK were found with a grid search. To improve the performance of the final algorithm, we also add a random bias and use a concatenation of the reservoir state and the current input for prediction, replacing Eq. (6.3) by $\hat{o}^t = W_o[x^{(t)}, i^{(t)}]$.

Prediction performance is presented in Fig. 7.3. RC and SRC are trained on $n = 70{,}000$ training points and RK on a sub-sampling of 7,000 of these training points, due to memory constraints. The testing dataset length was set at 2,000. The sizes N in Reservoir Computing and Structured Reservoir Computing are chosen so the dimension $q = N + d$ in Eq. (7.24) is a power of two for the multiplication by Hadamard matrix. Linear regression is solved using Cholesky decomposition.

The predictions in Fig. 7.3 show that all three algorithms are able to predict up to a few characteristic times. Due to high variance in the predictions, we also display the Mean-Squared Error (MSE) of each algorithm, as a function of the prediction time and averaged over 10 realizations. We normalize each curve by the MSE between two

independent KS systems.

We observe a decrease in the MSE when the size of the reservoir increases, meaning a larger reservoir yields better predictions. Performances are equivalent between RC and SRC, and they converge towards the RK performance for large reservoir sizes. Hence, this means RC, SRC, and RK can seamlessly replace one another in practical applications.

Code to generate data and run these models is available at `https://github.com/rubenohana/Reservoir-computing-kernels`.

7.3.3 Timing benchmark

Several steps in the Reservoir Computing pipeline need to be assessed separately, as described previously. We present the timings on a training set of length $n = 10,000$ and testing length of $m = 2,000$ in Table 7.2 for all three algorithms.

The *forward pass*, i.e. computing the recurrent iterations of each algorithm, is considered separately from the linear regression for *training*, to emphasize the cost of this important step. In RC, the most expensive operation is the dense matrix multiplication; the GPU memory was not large enough to store the square weight matrix for the two largest reservoir sizes. With Structured Reservoir Computing, this forward pass becomes very efficient even at large sizes, and memory is not an issue anymore. On the other hand, Recurrent Kernels iterations are very fast, as we only need to compute element-wise operations in a kernel matrix.

Prediction requires a forward pass and then is performed with autonomous dynamics as presented on Fig. 7.3 where Eq. (6.3) is repeated 600 times. For Recurrent Kernels, prediction remains slow, and this drawback is exacerbated by the autonomous dynamics strategy in time series prediction, that requires many prediction steps. For non-stationary datasets, Recurrent Kernels would remain relatively efficient even compared to RC and SRC, because of the moderate size of the training set.

This shows that SRC is a very efficient way to scale-up Reservoir Computing to large sizes and reach the asymptotic limit of performance. On the other hand, the deterministic Recurrent Kernels are surprisingly fast to iterate, at the cost of a relatively slow prediction when $n \gg N$.

	$N = 1,948$	$N = 3,996$	$N = 8,092$	$N = 16,284$	$N = 32,668$
RC	**2.6**/0.02/1.9	**3.1**/0.05/4.6	10.4/0.16/15.4	Mem. Err.	Mem. Err.
SRC	3.3/0.02/**1.6**	3.4/0.05/**2.7**	**3.5**/0.16/**3.7**	**3.6**/0.57/**6.8**	**3.6**/2.57/**13.0**
RK	**0.7**/**0.09**/23.0				

Table 7.2: Timing (Forward/Train/Predict, in seconds) for a KS prediction task as a function of N. We observe that Recurrent Kernels are surprisingly fast, except for prediction. Structured Reservoir Computing reduces drastically the speed of the forward pass at large sizes and is more memory-efficient than Reservoir Computing. Experiments were run on an NVIDIA V100 16GB.

7.4 Stability study

The previous section presented results from [56] to prove the usefulness and efficiency of Recurrent Kernels and Structured Reservoir Computing. We showed convergence of RC towards its limit, timing benchmarks, and their applications in chaotic time series prediction. In this section, we dive deeper into the implications of this Recurrent Kernel limit and present preliminary results on their stability and how it bears insight for Reservoir Computing as well.

A first stability study using this Recurrent Kernel limit was already proposed in [168]. Based on this asymptotic limit, they were able to rederive the stability condition of Reservoir Computing without inputs, depending on the variance of the random internal weights. This preliminary study serves as an inspiration for our presented work, to extend their analysis to non-differentiable activation functions (the Heaviside function) and in the presence of an input. Other works have also studied this stability in the presence of an input [156], [177], but these works are based on the Lipschitz-continuity of the activation function. We do not require this assumption here, but only assume the convergence of RC towards the RK limit when $N \to \infty$, which has been demonstrated numerically in Fig. 7.1.

7.4.1 Definition in Reservoir Computing and Recurrent Kernels

Stability of Reservoir Computing. As discussed previously, the Echo-State Property states that the reservoir state after a large-enough time τ shall not depend on the reservoir initialization. It is intimately linked with stability: whatever the arbitrary initialization, trajectories of the reservoir states for a given input should all converge into one.

Here, we study this stability for a random i.i.d. input. We initialize using an i.i.d. gaussian distribution n different reservoirs of size N, $x_1^{(t)}, \dots, x_n^{(t)}$. They all share the same internal weights and receive the same input $i^{(t)}$ of length T. This input $i^{(t)} \in \mathbb{R}^d$ is drawn at each time step uniformly on the unit sphere. The point here is to investigate whether these n reservoirs will converge towards a common trajectory.

Two important parameters will tune the transition between stability and chaos: the variances of the internal and input weights, σ_r^2 and σ_i^2 respectively. It is well-known that large internal weights tend to increase initial perturbations leading to a chaotic behavior, while a large input may regularize the dynamics because it saturates the activation function. We propose to revisit quantitatively these claims using the Recurrent Kernel limit.

The metric we choose here will be very simple:

$$L(t) = \frac{2}{n(n-1)} \sum_{i<j} \left\| x_i^{(t)} - x_j^{(t)} \right\|^2 \tag{7.25}$$

It characterizes the squared Euclidean distance between two reservoir states at time t, averaged over all the possible pairs (i, j).

Note that this set of n randomly-initialized reservoirs may also be seen as a probabilistic distribution on the initial reservoir state, with $L(t)$ an empirical estimate of the expectation value of the square distance between two independent samples.

Stability of Recurrent Kernels. It is then natural to describe with Recurrent Kernels the corresponding limit of the previous approach. This setting with random i.i.d. inputs corresponds exactly to Fig. 7.1 showing numerically the convergence of RC towards its RK limit very robustly (regardless of the parameter σ_r^2). We thus assume in the following that RC indeed converges towards its RK limit.

The n reservoirs in parallel define an $n \times n$ Gram matrix:

$$G_N(t) = \left[\left(x_i^{(t)} \right)^\top x_j^{(t)} \right]_{i,j} \tag{7.26}$$

Since the reservoirs are initialized with each component independently drawn from $\mathcal{N}(0, 1/\sqrt{N})$, the limit when $N \to \infty$ is the identity matrix:

$$\lim_{N\to\infty} G_N(0) = G(0) = I_n \tag{7.27}$$

The recurrent kernel equations (7.7), (7.8) are then used to update this Gram matrix, corresponding to the infinite-size limit of Reservoir Computing. If the ESP is satisfied, all the reservoirs converge to the same state $x^{(t)}$ and $G(\tau)$ becomes a matrix with all elements equal to:

$$[G(\tau)]_{i,j} = \|x^{(t)}\|^2 \tag{7.28}$$

To quantify whether or not the Gram matrix converges to this elementwise-constant matrix, the metric previously-defined for Reservoir Computing has a Recurrent Kernel equivalent by expanding the squared norm:

$$L(t) = \frac{2}{n(n-1)} \sum_{i<j} [G(t)]_{i,i} + [G(t)]_{j,j} - 2[G(t)]_{i,j} \tag{7.29}$$

Thanks to permutation symmetry (both in the initialization (7.27) and the updates (7.7), (7.8)), we will only have to characterize two terms in $G(t)$: the diagonal and the off-diagonal components that we will denote by $G_=(t) = [G(t)]_{1,1}$ and $G_\neq(t) = [G(t)]_{1,2}$. Thus the previous metric can greatly be simplified into:

$$L(t) = 2\left(G_=(t) - G_\neq(t)\right) \tag{7.30}$$

This simplicity comes from the deterministic nature of Recurrent Kernels. We only have to compute how the two scalar quantities evolve with time.

7.4.2 Error function activation

Explicit equations. Let us start by describing the explicit update equations of the Recurrent Kernels, when the activation function f is the error function. The corresponding kernel k is defined by:

$$k(u,v) = \frac{2}{\pi} \arcsin\left(\frac{2u^\top v}{\sqrt{(1+2\|u\|^2)(1+2\|v\|^2)}} \right) \tag{7.31}$$

This kernel is iterated with $u = \left[\sigma_r^2 x^{(t)}, \sigma_i^2 i^{(t)}\right]$ and $v = \left[\sigma_r^2 y^{(t)}, \sigma_i^2 i^{(t)}\right]$ (with the same input since this is a stability study). The diagonal terms correspond to the case where $y^{(t)} = x^{(t)}$:

$$G_=(t+1) = \frac{2}{\pi}\arcsin\left(\frac{2\sigma_r^2 G_=(t) + 2\sigma_i^2}{1 + 2\sigma_r^2 G_=(t) + 2\sigma_i^2}\right) \tag{7.32}$$

We thus obtain a recursive equation which describes the evolution of $\|x^{(t)}\|^2$.

For the off-diagonal terms, we equivalently have:

$$G_{\neq}(t+1) = \frac{2}{\pi}\arcsin\left(\frac{2\sigma_r^2 G_{\neq}(t) + 2\sigma_i^2}{1 + 2\sigma_r^2 G_=(t) + 2\sigma_i^2}\right) \tag{7.33}$$

Due to the norm in the denominator, the next off-diagonal term depends both on the previous diagonal and off-diagonal components.

Asymptotic limit of $G_=(t)$. We now characterize the asymptotic limit of $G_=(t)$ and $G_{\neq}(t)$. In particular, we are interested whether they have the same limit, in which case the RK is considered stable according to the metric $L(t)$ in Eq. (7.30).

The evolution of $G_=(t)$ defined in Eq. (7.32) consists in the recursive application of the map:

$$h_1(g) = \frac{2}{\pi}\arcsin\left(1 - \frac{1}{1 + 2\sigma_r^2 g + 2\sigma_i^2}\right) \tag{7.34}$$

with the initialization $G_=(0) = 1$. We are thus looking to characterize the fixed points of h_1.

Case $\sigma_i^2 = 0$:

A similar case has been studied in [168], it boils down to the study of whether a reservoir converges to zero with no input. 0 is already a fixed point of h_1 and thanks to the strict concavity of h_1 (which can be proven using its second derivative), there is at most a second one.

We thus compute its derivative in 0: $h_1'(0) = 4\sigma_r^2/\pi$ and compare it to 1.

- **Stable case:** $\sigma_r < \sqrt{\pi}/2$ or $h_1'(0) < 1$. $G_=(t)$ converges to 0 with zero input.
- **Critical case:** $\sigma_r = \sqrt{\pi}/2$ or $h_1'(0) = 1$. $G_=(t)$ still converges to 0 with zero input, but the speed of convergence is not exponential.
- **Unstable case:** $\sigma_r > \sqrt{\pi}/2$ or $h_1'(0) > 1$. $G_=(t)$ converges to another fixed point $a_0(\sigma_r^2) > 0$.

This result is coherent with other approaches based on Lipschitz continuity, as the error function is $2/\sqrt{\pi}$-Lipschitz.

Case $\sigma_i^2 > 0$:

In this case, h_1 is an increasing function and for all $g \in [0;1]$:

$$h_1(g) \geq h_1(0) = \frac{2}{\pi}\arcsin\left(\frac{2\sigma_i^2}{1 + 2\sigma_i^2}\right) > 0 \tag{7.35}$$

and:

$$h_1(g) \leq h_1(1) < 1 \tag{7.36}$$

Thus, the sequence $G_=(t)$ stays strictly between 0 and 1. There is a fixed point in $(0;1)$ thanks to the intermediate value theorem and this fixed point is unique thanks to the strict concavity of h_1 (proven using the computation of its second derivative). $G_=(t)$ converges towards this unique fixed point of h_1, we will denote this quantity $a\left(\sigma_r^2,\sigma_i^2\right)$ to emphasize its dependence on the variances of the weights, σ_r^2 and σ_i^2.

Asymptotic limit of $G_{\neq}(t)$.

Case $\sigma_i^2 = 0$:

This case is quite simple as $G_{\neq}(t)$ is defined by $G_{\neq}(0) = 0$ and the recursive Eq. (7.33). With $\sigma_i^2 = 0$, we obtain $G_{\neq}(t) = 0$ for all t.

Case $\sigma_i^2 > 0$:

We now study the asymptotic behavior of $G_{\neq}(t)$, defined from Eq. (7.33) with $G_{\neq}(0) = 0$. We obtain the asymptotic limit by replacing $G_=(t)$ by $a(\sigma_r^2,\sigma_i^2)$,[2] obtaining the following recursive update map:

$$h_2(g) = \frac{2}{\pi}\arcsin\left(\frac{2\sigma_r^2 g + 2\sigma_i^2}{1 + 2\sigma_r^2 a\left(\sigma_r^2,\sigma_i^2\right) + 2\sigma_i^2}\right) \tag{7.37}$$

h_2 has at most two fixed points due to its strong convexity (as it is of the form $h_2(g) = A\arcsin(Bg + C)$ with $A, B, C > 0$ for $g \geq 0$), with $a\left(\sigma_r^2,\sigma_i^2\right)$ being one of them thanks to its definition. In any case, such a recursive sequence initialized at 0 converges to the lowest fixed point of h_2.

Numerical results. We present a numerical investigation supporting the theoretical derivation above in Fig. 7.4. It will show how this theoretical framework can make quantitative predictions about stability of Reservoir Computing and Recurrent Kernels in the presence of an input.

We start with the definition of stability: $n = 40$ reservoirs of size $N = 1{,}000$ initialized randomly are fed the same input $i^{(t)}$ and observe whether they converge to the same trajectory with the metric $L_N(t)$. Two cases are presented: without ($\sigma_i^2 = 0$) and with ($\sigma_i^2 = 1$) input. In both cases, we encounter the transition between stability and chaos as we increase the reservoir weight variance σ_r^2. The stability domain in the presence of an input is increased as the critical transition happen for $\sigma_r \approx 1.65$ instead of $\sigma_r = \sqrt{\pi}/2 \approx 0.89$.

We also present the asymptotic limit of $L(t)$ sweeping through different values of σ_r and σ_i, for Recurrent Kernels (and the graph would be similar for Reservoir Computing and using the theoretical derivation above). We clearly observe the regularization performed by the input, which comes from the saturation of the activation function f, we do not observe it with unbounded activation functions such as the Rectified Linear Unit.

Finally, we present how our method based on fixed points of h_1 and h_2 predicts these behaviors, for example in the case of $\sigma_i^2 = 1$. In all the cases, h_1 has a single fixed point which is then used to define h_2. When the reservoir weights are small ($\sigma_r^2 = 1.3$), h_2 has two fixed points, with the lowest one being equal to the fixed point of h_1. Thus, we are in the stable regime as diagonal and off-diagonal terms both converge to this fixed point. At the transition ($\sigma_r^2 \approx 1.65$), h_2 only has one fixed point and after

[2] A rigorous theoretical derivation without replacing $G_=(t)$ by its limit is more cumbersome but can be written exploiting the fact that functions and sequences here are monotonous.

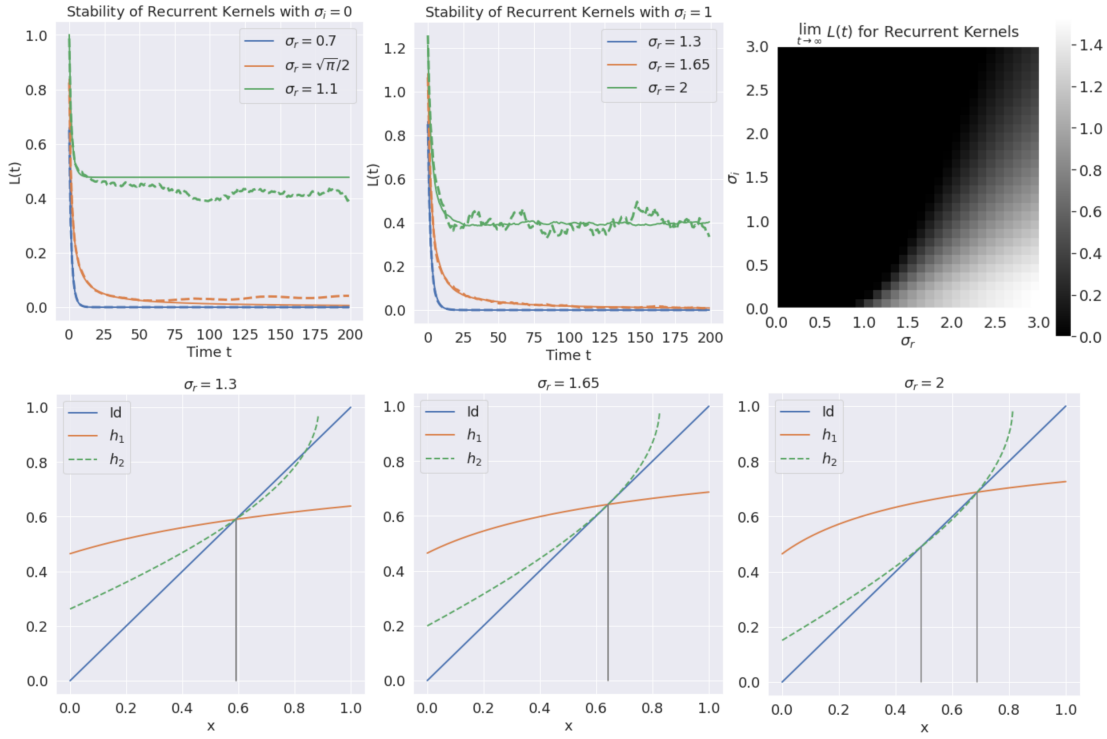


Figure 7.4: (Top left) $L(t)$ as a function of time for different values of internal weight variances σ_r^2, with no input. We see the three regimes, stable, critical, and chaotic as we increase the reservoir variance. (Top middle) $L(t)$ as a function of time, this time with an external input $\sigma_i = 1$. We observe that this input regularizes the dynamics and pushes the transition to higher values of σ_r. (Top right) The asymptotic value of L for many different values of σ_i and σ_r. We can distinguish a stable region on the left and a chaotic region on the right, with a frontier given by the study of the fixed points of h_1 and h_2. (Bottom) For the three different values of σ_r in the previous graph with $\sigma_i = 1$, depiction of the function h_1 (in orange) and h_2 (in green dashed, as its definition depends on the fixed point a of h_1). Fixed points are indicated with grey vertical lines.

the transition, h_2 has two fixed points again but the lowest does not correspond to the one of h_1. Thus, the diagonal term $G_=(t)$ converges towards the highest value (as it is the fixed point of h_1) and the off-diagonal term $G_{\neq}(t)$ converges towards the lowest fixed point of h_2, and the gap between the two means the Recurrent Kernel is operated in an unstable regime.

7.4.3 Heaviside activation

Explicit equations. We now move to the case where f is a Heaviside function. Similar to our study of the error function, we start with the corresponding kernel, which is given by:

$$k(u,v) = \frac{1}{2} - \frac{1}{2\pi}\arccos\left(\frac{u^\top v}{\|u\|\|v\|}\right) \tag{7.38}$$

The update equation for the diagonal coefficients of the Gram matrix are always equal to $G_=(t) = k(u,u) = 1/2$. The off-diagonal coefficents on the other hand obey the following recursive equation:

$$G_{\neq}(t+1) = \frac{1}{2} - \frac{1}{2\pi}\arccos\left(\frac{\sigma_r^2 G_{\neq}(t) + \sigma_i^2}{\frac{1}{2}\sigma_r^2 + \sigma_i^2}\right) \tag{7.39}$$

Once again, $G_{\neq}(t)$ converges to the lowest fixed-point of the following function:

$$h(g) = \frac{1}{2} - \frac{1}{2\pi}\arccos\left(\frac{\sigma_r^2 g + \sigma_i^2}{\frac{1}{2}\sigma_r^2 + \sigma_i^2}\right) \tag{7.40}$$

$1/2$ is already a fixed point of this equation. There is always a second fixed point strictly below, because $h(0) > 0$ and $\lim_{g\to 1/2} h'(g) = +\infty$. Thus, Reservoir Computing with a Heaviside activation or the corresponding Recurrent Kernels are always chaotic.

Numerical results. This conclusion is validated by numerical experiments on Recurrent Kernels, presented in Fig. 7.5. There is no stability region, where $\lim_{t\to\infty} L(t) = 0$, and no transition between stability and chaos. This means that two Reservoir Computing algorithms initialized differently will in general not converge to the same trajectory. This result is similar to other studies of stability using Lipschitz continuity, as the Heaviside activation is irregular at the origin. Our approach using its Recurrent Kernel limit still enables us to quantitatively predict the limit of the stability metric $L(t)$.

Additionally, we see that the input also somewhat regularizes the dynamics, which may help to dampen the chaotic behavior of reservoirs without Heaviside activation function. The degree of chaos in the dynamics can thus be tuned thanks to this parameter. Note that due to homogeneity in both the Reservoir Computing and Recurrent Kernel formulas, only the ratio σ_i/σ_r is relevant with Heaviside activation functions.

We were allowed to safely transcribe the results from Recurrent Kernels to Reservoir Computing and vice versa here thanks to the convergence of Reservoir Computing towards Recurrent Kernels, which has been proven numerically for the Heaviside activation (see Fig. 7.2). Thus, here we have an example of convergence (of

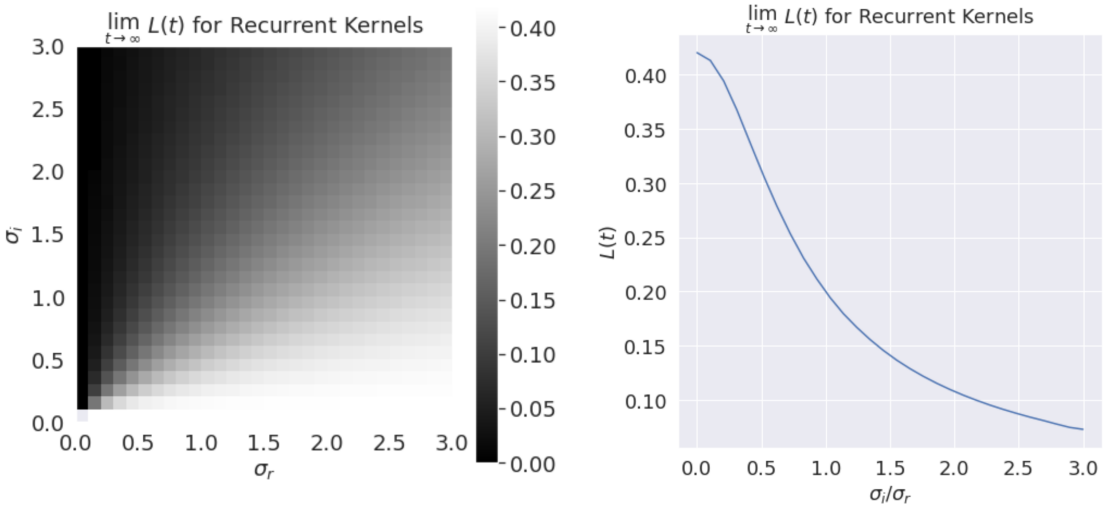


Figure 7.5: Unstability of Recurrent Kernels corresponding to a Heaviside activation. (Left) Limit of the metric $L(t)$ when $t \to \infty$ for Recurrent Kernels corresponding to the Heaviside activation function in Reservoir Computing, for many values of σ_r^2 and σ_i^2. We observe that there is no stable region, but this stability metric is close to zero for small reservoir weights variance σ_i^2. (Right) Due to the homogeneity of the Heaviside activation, the previous limit only depends on σ_i/σ_r. We thus display the previous limit of $L(t)$ depending on this parameter, to have a more quantitative representation.

RC towards RK), but no stability (of RC and RK), even though our theoretical proof of convergence relied on stability. In practice, convergence seems more robust than stability.

Chaotic time series prediction. We would like to take the opportunity here to mention two works using Heaviside activation functions in our previous works, both in [27] and [28]. Such an irregular activation function was introduced to the binary constraint of the Digital-Micromirror Device. We had already observed the regularization from the input, which was enabling some learning for chaotic time series prediction.

Hence, chaos does not prevent learning altogether. Thanks to the input regularization, one can also use a chaotic system and increase the input weights σ_i^2 to decrease $\lim_{t\to\infty} L(t)$. The system may still stay chaotic, i.e. $\lim_{t\to\infty} L(t) > 0$, but the linear regression may be able to distinguish the information depending on the input to the initialization noise. There is still a benefit from having no sensitivity to initial conditions, or even operating at the "edge of chaos" to increase the memory, as the performance of these binary Reservoir Computing in [27], [28] was limited.

7.4.4 Conclusion of this study

We have shown that the Recurrent Kernel framework is quite powerful to investigate stability in Reservoir Computing. Thanks to its deterministic nature, it boils down to the study of scalar functions and their fixed points. We were able to tackle two differ-

ent activation functions, one Lipschitz-continuous and the other not, giving rise to very different behaviors. Moreover, we were able to find the "edge-of-chaos" quantitatively in the presence of an input.

The results presented here are preliminary and a few points remain to be investigated further. For example, the transition between stability and chaos is observed in Fig. 7.4, but not completely characterized analytically. Interesting directions for future research include the link between this stability metric $L(t)$ and prediction error on a particular regression task, or the extension of this analysis for real-world datasets (Is the variance of the input the only quantity of interest? Maybe the covariance between different times also matters).

7.5 Discussion

7.5.1 In Reservoir Computing

This research project may be interpreted as a continuation of [168] to express the asymptotic limit of Reservoir Computing as a Recurrent Kernel. We made a conceptual link with Random Features and prove a first convergence theorem to make this connection rigorous. It would be interesting to extend this approach to other reservoir topology, for example with local connections [15] or Deep Reservoir Computing [178].

By leveraging recent works on kernel approximation, this theoretical study invited us to introduce Structured Transforms to accelerate conventional Reservoir Computing. While there is still a limited theoretical understanding on structured transforms, this simple change accelerates considerably Reservoir Computing while being numerically equivalent to it. We believe it has a great potential for future Reservoir Computing applications.

Finally, to put things in perspective, other asymptotic limits may be worth investigating. Here the reservoir dimension N goes to infinity while the number of samples n is fixed. A more challenging case would consider the number of samples also going to infinity, with a fixed ratio n/N for example to relate to other results using Random Matrix Theory [179].

7.5.2 In deep learning

Our approach may also have a deep learning interpretation. To see this connection, we perform the classical operation of unrolling the recurrent neural network into a multilayer perceptron. We then remove the input fed at each time step but only consider the one at $t = 0$. In this setting, our theorem proves the convergence rate of a multilayer perceptron towards its compositional kernel limit. Related studies have already been proposed [180]–[182], it would be interesting to investigate how the typical notions of Reservoir Computing such as stability, memory, or topology translate to the deep learning framework.

7.5.3 For physical implementations of Reservoir Computing

Reservoir Computing has already drawn a lot of attention towards fast and energy-efficient physical implementation [17], [27], [28], [53], [147], [150], [183]–[186]. One interesting direction of research would be the realization of Structured Reservoir Computing on dedicated hardware, as it simplifies the design in two blocks: a structured transform and element-wise multiplications. It can be efficiently implemented on dedicated electronics such as FPGAs, with very low memory costs. To reduce the footprint even further, training may be performed using an online method to avoid the storage of the reservoir states before the linear regression.

CHAPTER 8
General conclusion

The work presented in this thesis included two fairly distinct halves in computational imaging and optical computing, unified by the common theme of random matrices and multiple light scattering.

In computational imaging, we have shown how to leverage existing algorithmic tools, such as matrix factorization or phase retrieval, and adapt them for specific challenging imaging conditions. In particular, we have modeled a non-invasive deep fluorescence experiment and solved it for two different settings with different detectors: a single-pixel or a high-resolution camera. In both cases, reconstruction of the Transmission Matrix of the medium is successful, opening the way to new imaging applications, as for example the ones presented in Chapter 4. Important hurdles still need to be overcome before these techniques are applicable with alive animals, the most important one being probably the decorrelation speed of the scattering medium. Other experimental developments are also worth exploring in the future, for example with non-linear feedback and other contrast mechanisms, or combining these techniques with acoustics to have a more localized signal.

On the other hand, the computational tools introduced for our inverse problem are applicable to other imaging settings, not involving a scattering medium. In Appendix A, other contributions in optics are described. They show how the recent algorithmic advances to tackle large datasets and non-linear optimization may push forward other fields in computational imaging. For example, the spectral methods introduced for phase retrieval can be applied in many settings, accelerating reconstruction in ptychography for example. Moreover, matrix factorization can also prove useful: we show how to compress the measurements of a hyperspectral Raman microscope, to minimize the acquisition time with a limited photon budget.

For optical computing, we started from the initial observation that random matrix multiplications were easily obtained with multiple light scattering. To leverage this potential, we have implemented an optical system to accelerate various neural networks with randomly-fixed weights. A particular focus has been put in this thesis on Reservoir Computing, a recurrent architecture where optics enables very large reservoir sizes. We have shown how our optical computing strategy is able to obtain very competitive results on chaotic time series prediction, faster and more energy-efficient than its electronics counterpart.

These very large optical reservoirs have inspired a theoretical breakthrough, which does not involve optics, as we described the asymptotic Recurrent Kernel limit of Reservoir Computing and introduce Structured Reservoir Computing as a more-efficient variant. Recurrent Kernels are very efficient when the number of samples is not too large, and they provide a convenient angle to study properties of Reservoir Computing. In the future, we believe Structured Reservoir Computing represents a robust and easy-to-implement alternative to Reservoir Computing, possibly replacing it in

many different cases. This calls for further investigations on structured transforms, for which there is still limited theoretical understanding. While these last results reduce by their computational efficiency the impact of the proposed optical implementation, free-space optics still provides undeniable advantages in terms of parallelism and speed, to be investigated further.

APPENDIX A

Quick overview of other contributions

Other results are presented in this chapter with the main idea and a few figures. It shows different examples of the fruitful interplay between optimization and computational imaging, with or without complex optical media and random matrices.

This presentation also helps to describe the context around the projects of Chapter 3 and 4, to paint the bigger picture in which ideas slowly emerged. Such an exercise often brings interesting insights as it puts into perspective the presented results, helping us find blind spots in our computationally-oriented approach.

More information can be found in the original papers.

A.1 Variance optimization for deep imaging

This project [59] precedes the work presented in Chapter 4, to tackle the same problem of focusing with linear fluorescence in the epi-fluorescence configuration with wavefront shaping. The limits of the classical technique have already been discussed in Chapter 3: optimizing the total feedback intensity [92] does not focus excitation on a single target. Due to the absence of non-linearity, the system distributes light to all the fluorescent targets. Chapter 3 presents a computational method to focus light back on single targets.

Here we put a camera to gather more information than the total fluorescence feedback. We show that optimizing the variance of the measured camera image provides a simple technique to focus excitation light on individual targets. This variance metric comes from a simple observation: if many fluorescent targets are excited equally, they generate independent fluorescent speckle patterns that are summed together, averaging out the final image. Thus the variance is higher if all the excitation is focused on a single fluorescent target only. This was the first proof that there is meaningful information in the recorded fluorescent speckle, a concept which has been developed later with the double Transmission Matrix of Chapter 4.

If I_{fluo} denotes the captured fluorescence image and $I_{\text{exc}}^{(k)}$ the excitation intensity on the k-th target, we have:

$$\text{Var}(I_{\text{fluo}}) \propto \sum_k \left(I_{\text{exc}}^{(k)}\right)^2 \tag{A.1}$$

provided the excitation speckle are independent. Hence this variance optimization is formally equivalent to 2-photon fluorescence optimization, already demonstrated for focusing in complex media in [76].

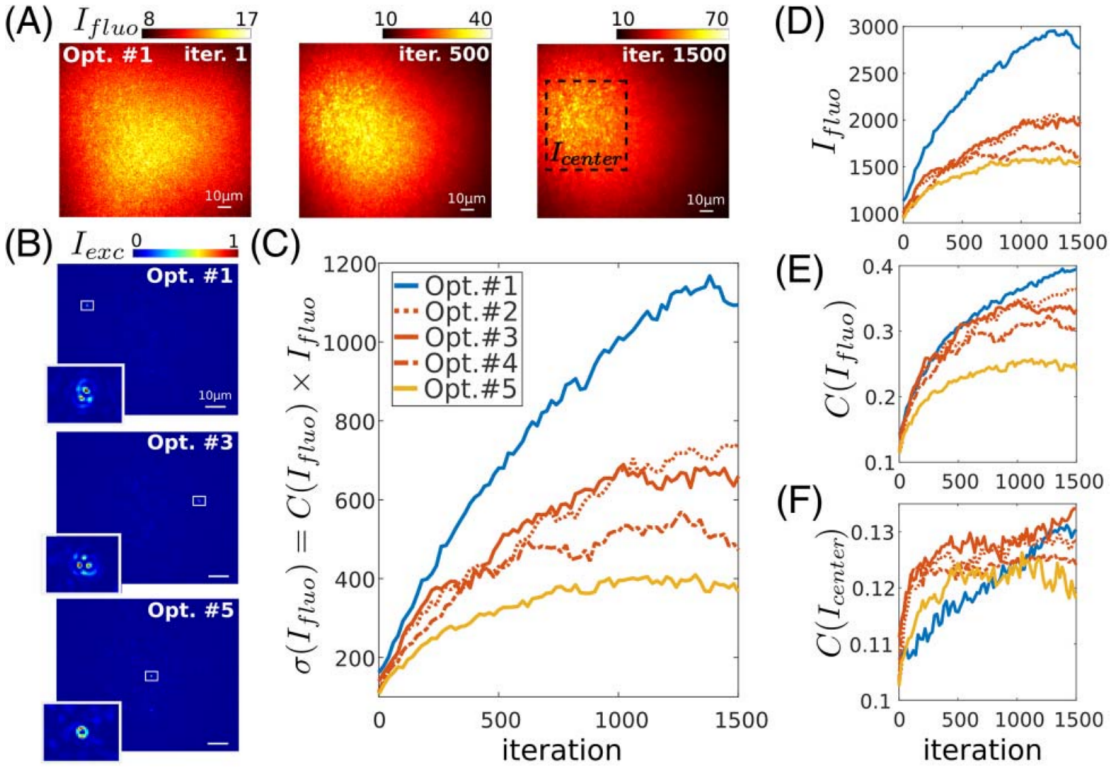


Figure A.1: Variance optimization to focus with linear fluorescence feedback. (A) Images on the camera of the fluorescent speckle during optimization. We observe that the contrast of the speckle increases as well as the total fluorescence intensity. (B) Images of the excitation wavelength using a control camera, this information is not used during the optimization. We observe that indeed a focus is obtained for the excitation. (C) Variance as a function of the iteration number. (D, E, F) Additional graphs of the total fluorescence intensity, contrast, and contrast in the central region.

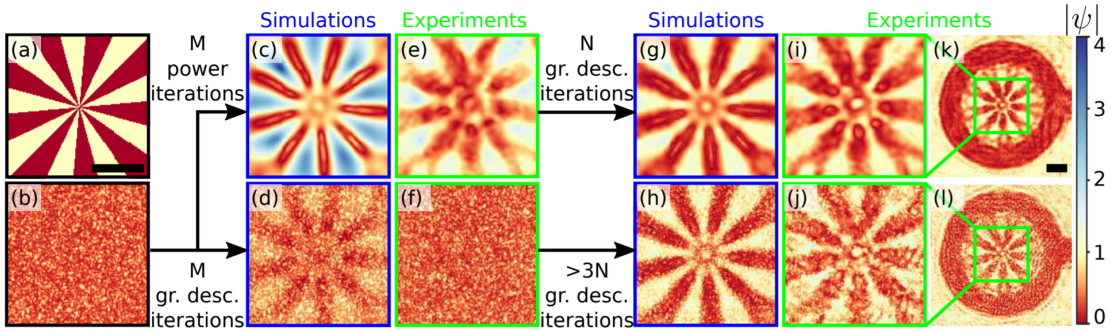


Figure A.2: Spectral method to provide an initialization in ptychographic reconstruction. (a) The target object, an amplitude-only object called a Siemens star. (b) Initial random guess. (c-f) Comparison between M power iterations of the spectral method and M gradient descent iterations, in simulation (c, d) and on experimental data (e, f). (g-l) Final results after N or $> 3N$ gradient descent iterations. We observe that the spectral method accelerates the convergence of gradient descent. Experimental data were taken from a THz ptychography experiment [187]

A.2 Spectral methods for ptychography

Spectral methods are a relatively new technique as they appeared in 2013 [97], [98], but they have already received extensive attention from the non-linear optimization community. These techniques are attractive for their speed and for the strong theoretical results obtained in the past years. For random sampling matrices, we know when that reconstruction is possible as soon as the minimal bound derived from information theory [101], and what is the optimal spectral method to use depending on the noise model [102].

However, one may wonder what happens when the sampling matrix is not random, a very common case in imaging problems. Here we show how spectral methods may be used for ptychography [63], a particular computational imaging technique where a coherent probe is scanned across the sample.

We want to solve for ψ in a phase retrieval problem:

$$y = |S\psi|^2 \tag{A.2}$$

where y are a stack of intensity images and S is a concatenation of the sampling matrices for each image. In this setting, spectral methods are used to provide an approximate initial estimate, which accelerates the subsequent iterative reconstruction. This enables a more robust reconstruction of complex objects in ptychography, without a priori knowledge of the sample.

A.3 Compressive Raman and matrix completion

Raman spectroscopy is a powerful technique to distinguish chemical species by looking at the molecular vibrational spectra. It does not require tagging, i.e. the introduction of specific particles to image in the sample, such as fluorescent proteins. However, the number of inelastically scattered photons to detect is low, which slows down the imaging process and calls for very efficient spectroscopic imaging systems. In this computational imaging age, a compressive spectrometer has been introduced in [188], placing a Digital Micromirror Device in the spectroscopic arm to multiplex frequencies to increase the photon efficiency. We typically want to undersample measurements and only take the minimal number of measurements to reconstruct an image.

In this work [62], we build a new algorithm to retrieve both images of different chemical species and their spectra from undersampled measurements. This is made possible thanks to the assumption that only a limited number of distinguishable chemical species are present, proteins and lipids for instance (such a property depends on the resolution of the spectrometer, the more it is precise, the more species it can distinguish). Here, we reduce the number of acquired patterns and fill in the incomplete measurement matrix to using a low-rank matrix completion algorithm.

If we assume that we have n spatial pixels, k spectral bands, and $s \ll n, k$ chemical species, the acquired patterns are denoted by H_{exp} which is a subsampled version of the total matrix $H \in \mathbb{R}^{n\times k}$. With the low-rank assumption, H can be factorized as a

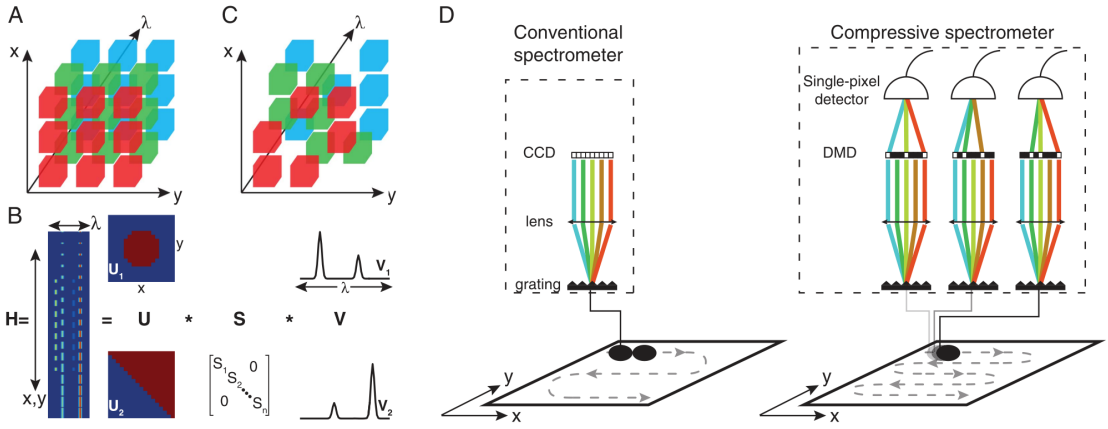


Figure A.3: Low-rank matrix completion for Raman microscopy. (A) The measurement tensor is a 3D object with two spatial dimensions x, y and one spectral dimension λ. (B) This tensor can be viewed as a 2D matrix by concatenating the spatial dimensions. This matrix is low-rank if the number of different chemical species is fixed. (C) To make the acquisition faster, we undersample the measurement tensor, subsampling the spectral domain randomly for each spatial position. (D) This subsampling is performed by replacing the camera in a conventional spectrometer by a single-pixel detector with a programmable Digital Micromirror Device.

rank-s matrix:

$$H = XY \tag{A.3}$$

with $X \in \mathbb{R}^{n\times s}$ the images for each chemical species, and $Y \in \mathbb{R}^{s\times k}$ the corresponding spectra.

This work has been featured in: `https://www.sciencedaily.com/releases/2019/03/190314101323.htm`.

A.4 Object and pupil recovery with phase-diversity

Phase diversity has been introduced in astronomy to obtain the phase of atmospheric aberrations [189]. It is based on the capture of several "phase-diverse" images to circumvent the missing-phase problem in a single image. For example, phase information is present in the propagation of light so one can translate a camera along the axial direction and take several images. By modeling this propagation and solving for the missing phase, this provides a framework to measure phases when experimentally, one only has access to the camera side.

Here, we apply phase diversity to image incoherent objects behind a surface diffuser [57]. Each captured image $I_n(r)$ is modeled by:

$$I_n(r) = O(r) * S_n(r) \tag{A.4}$$

where $O(r)$ denotes the object and $S_n(r)$ the Point-Spread Function (PSF) of the n-th image. All the $S_n(r)$ are linked to each other by a pupil function $H(f)$ in the Fourier space (since we have a surface diffuser) and an angular spectrum propagator $P_n(f)$:

$$S_n(r) = \left|\mathcal{F}\{H(f)P_n(f)\}\right|^2 \tag{A.5}$$

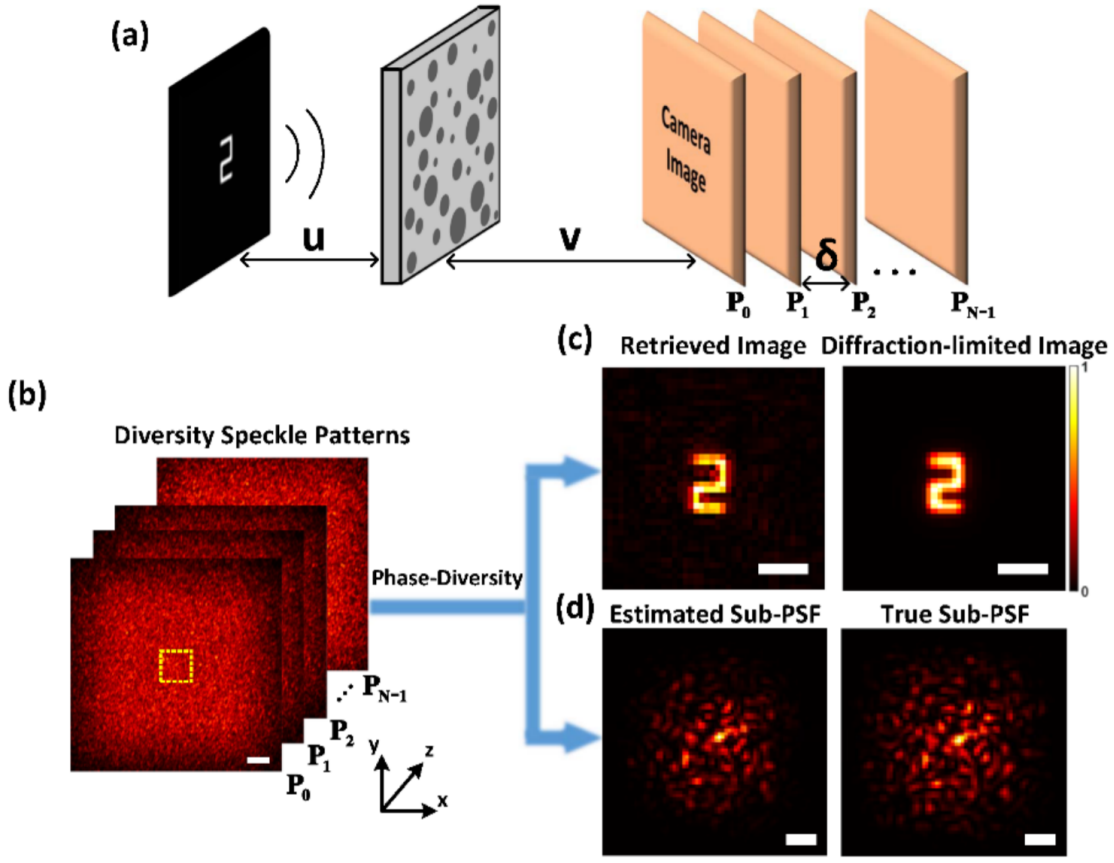


Figure A.4: Joint reconstruction of object and scatterer with phase diversity. (a) An incoherent object to image is place behind a thin diffuser. A stack of camera images is taken by translating the camera in the axial direction. (c) We obtain a stack of phase-diverse speckle patterns, from which we can reconstruct both the object (c) and a cropped region of the Point-Spread Function of the system (d).

From the knowledge that $H(f)$ is phase-only, it is possible to retrieve both $O(r)$ and $H(f)$, making it a non-invasive method to image objects behind a thin scattering medium.

A.5 Studying autocorrelation imaging

The autocorrelation imaging technique is an elegant technique for single-shot imaging through complex scattering media introduced in [91]. It can be applied in incoherent imaging settings when the object is not larger than the memory effect range. Light from an incoherent object $O(r)$ propagates through a scattering medium of Point-Spread Function (PSF) $S(r)$ (assuming the object is in a single memory effect range to have a constant PSF, this happens for thin diffusers), resulting in an image $I(r)$ on the camera given by:

$$I(r) = O(r)^* S(r) \tag{A.6}$$

where '$*$' is the convolution operator. The autocorrelation imaging technique is based on the identity:

$$I(r) \star I(r) = (O(r) \star O(r))^* (S(r) \star S(r)) \tag{A.7}$$

where '$\star$' is the autocorrelation operator. Since speckles are spatially random patterns, their autocorrelation is close to a delta-function, resulting in $I(r) \star I(r) \approx O(r) \star O(r)$. We thus obtain the autocorrelation of the object, from which one may retrieve the object $O(r)$ using a Phase Retrieval algorithm.

In this work [61], we precisely characterize how this technique is diffraction-limited, using the identity [33]:

$$S(r) \star S(r) \propto \left| \frac{J_1(\pi D r/\lambda v)}{\pi D r/\lambda v} \right|^2 \tag{A.8}$$

From this characterization, we show how to correct and obtain a more faithful object $O(r)$. We use in particular slightly-more refined model to reconstruct the PSF $S(r)$ as well, and hence we characterize the optical system for further imaging applications like imaging of moving objects behind a fixed scatterin medium.

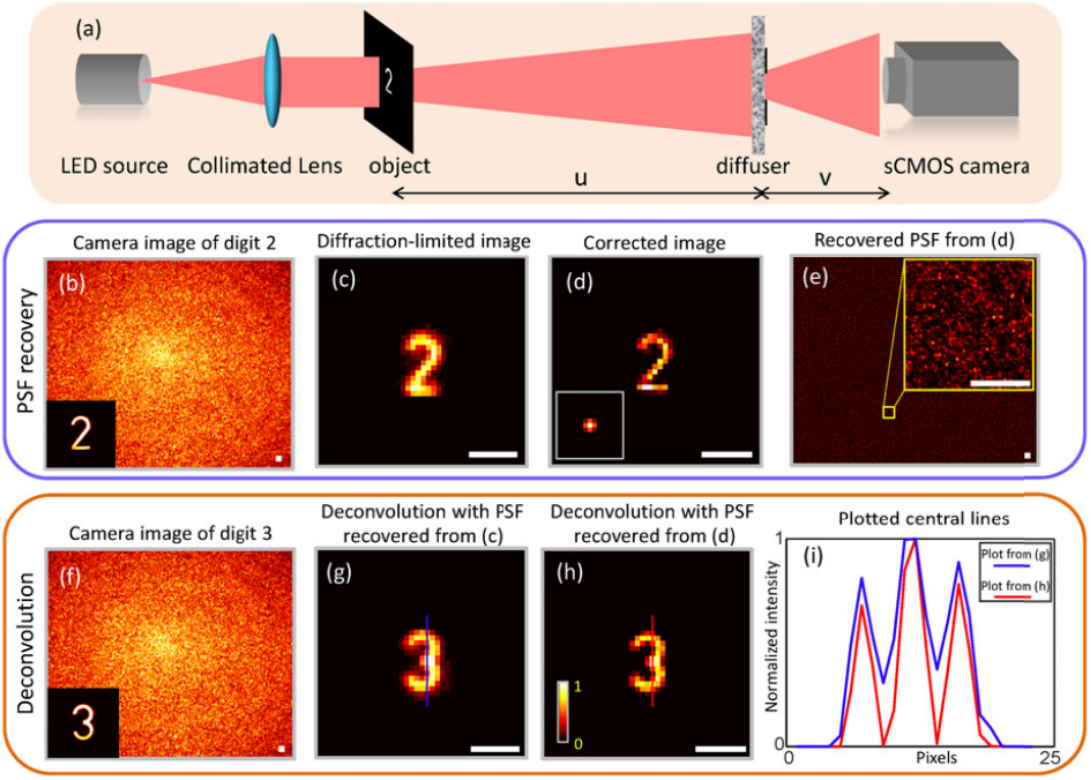


Figure A.5: Autocorrelation imaging, refined to obtain the PSF of the system and application. (a) Optical system, wherean incoherent object is placed before a diffuser and imaged on a camera. Here the autocorrelation imaging technique exploits the statistical properties of the speckle to reconstruct the object in a single-shot.
(b) Picture taken on the camera when the object is a "2" digit (inset). (c) Result of autocorrelation imaging without correction of the blur kernel. (d) Result after correcting the blur kernel. (e) Recovered PSF $S(r)$ by deconvolving (b) using (d).
(f) Another camera image for a different object, to image by deconvolution (because it is faster and simpler than the autocorrelation imaging technique). (g) Reconstruction using a first PSF, retrieved using the reconstructed object from (c). (h) Reconstruction using the PSF from (e), using the corrected object. (i) Intensities along the lines in (g) and (h), we observe that the corrected PSF enables a more accurate deconvolution. Thus, our refined model to take into account the blur kernel improves the reconstruction performance for other applications.

APPENDIX B

Phase Retrieval

Phase Retrieval is a longstanding computational problem that is ubiquitous in computational imaging, that is both simple to define and hard to solve. In this thesis, we have resorted to spectral methods to tackle this problem. To give more context about the choice of this particular method, we would like to discuss more about recent advances in phase retrieval, both in terms of applications and algorithms. Phase Retrieval has inspired researchers to propose a vast array of methods to solve this non-linear problem. This diversity comes from the large number of independent studies conducted in different communities, with slightly different assumptions and settings. Thus, Phase Retrieval is a non-linear equation with a very rich background to build on top of, even though it remains far from being solved.

Taking a step back, Phase Retrieval can also be interpreted as a one-layer neural network with a square activation function. Solving PR consists in finding the weights in this network, showing how even in a very simple network, training may be a very interesting theoretical challenge. It may provide an inspiration on how to solve more complex networks.

This broad overview is necessarily incomplete. We will sometimes refer to recent reviews [190], [191] for more details, while also complementing them with breakthroughs of the past five years.

B.1 Introduction

B.1.1 Definition

Phase Retrieval is about finding $x \in \mathbb{C}^d$ in:

$$y = |Ax|^2 \tag{B.1}$$

where $y \in \mathbb{R}^n$, $A \in \mathbb{C}^{n \times d}$, and $|.|^2$ is an element-wise operator. The simple fact that we miss the phase of Ax makes the problem quite difficult to solve as we will see in this chapter.

The equation without the modulus operator $y = Ax$ corresponds to linear regression, a canonical problem in statistical learning which has been extensively characterized. Explicit solutions can be derived, iterative algorithms are also possible and we are assured of their convergence since the optimization problem is convex. Moreover, numerical stability, robustness against noise, and performance with sparsity assumptions have all been precisely characterized.

In contrast, Phase Retrieval is a non-linear equation, yielding a non-convex optimization problem. The intense study of this equation has generated a lot of insights

in non-linear optimization, a very popular field nowadays with the advent of machine learning.

B.1.2 Motivation

Phase Retrieval has been mentioned in the physics literature since the 1950s in crystallography [192]–[194]. It arises whenever one wants to find a complex-valued object from intensity-only measurements. A possibility to avoid this missing-phase problem in practice is to introduce a well-characterized reference and use interferences to obtain the phase, but this is often infeasible, for example in X-ray crystallography [195] or electron microscopy [196]. A computational reconstruction involving Phase Retrieval is thus necessary in these challenging settings.

Additionally, a recent trend consists in designing simpler optical systems, leveraging modern computational capabilities to reconstruct an image [197], [198]. This enables new capabilities such as a very large field-of-view or 3D reconstruction, while keeping the simplicity of the system with few optical elements.

B.1.3 History and applications

To give a quick overview of how this equation was important in the history of physics, we can start with the first diffraction pattern of DNA by Rosalind Franklin in 1952, using X-ray coherent diffraction imaging. Based on this image, Crick and Watson were able to derive the composition and shape of the DNA molecule, and received the Nobel Prize in 1962. Phase Retrieval is still very important in this domain today: since optical components are challenging to design for these very energetic waves, we usually measure intensities after free-space propagation and reconstruct computationally an image of the object [195], [199], [200]. Another computational approach for this issue is ptychography, in which several images are taken for different object positions and this redundant information helps to solve the corresponding phase retrieval problem [201], [202].

Phase Retrieval has also played an important role in astronomy, to estimate phases of certain wavefronts from intensity measurements. For example, in 1993, after launching the Hubble Space Telescope, astronomers realized the main optical component of the telescope had a defect. The image of a distant star was not a point like expected, but aberrations in the optical system were degrading the resolution of the telescope. The solution required to solve a complex phase retrieval problem to quantify these aberrations from these intensity measurements [203], and send a physical module in space to compensate these aberrations on the telescope directly. The operation has been successful and the Hubble space telescope produced very famous images of cosmological objects [204].

More recently, microscopy and the field of computational imaging also benefitted from the flexibility of Phase Retrieval to design new imaging schemes. To extend the capabilities of the phase-contrast microscope proposed by Zernike in 1942 [205], computational reconstructions have been proposed to extend the field-of-view with Fourier Ptychographic Microscopy [197], obtain 3D tomographic measurements

Application	Sampling matrix A	Assumption on x
Fourier	Fourier matrix F	Positive / finite support
Coherent Diffraction Imaging, Ptychography, Fourier ptychography	$\begin{bmatrix} F & 0 & \ddots \\ 0 & \ddots & 0 \\ \ddots & 0 & F \end{bmatrix} \begin{bmatrix} D_1 \\ \vdots \\ D_k \end{bmatrix}$	None
Random setting	Random i.i.d.	None
Holography, phase-diversity	$\begin{bmatrix} P_1 \\ \vdots \\ P_k \end{bmatrix}$	Phase-only or none

Table B.1: Compacted view of the different settings for Phase Retrieval, from the random setting to different computational imaging applications.

[206], [207], or defocus the camera and retrieve phase information from this defocus [208], [209]. In all these situations, one needs to solve a phase retrieval problem.

B.2 Different models to solve

Since Phase Retrieval is associated to many different fields, it has also appeared in several different forms. We will focus here on the cases which corresponds to the matricial form of Eq. (B.1). More complicated variants may also be expressed, such as optical tomography, and they are typically framed as a non-linear optimization problem and solved using iterative algorithms.

In other cases, prior information about x is available: for example one often assumes in astronomy that a star is a positive real-valued object with finite support. These constraints help the algorithm converge, in particular to guarantee the unicity of the solution in ill-posed problems.

A taxonomy of diverse models involving Phase Retrieval is provided in Table B.1. In particular, we have grouped together applications with similar forward models, even though they traditionally are not presented together.

- Fourier transforms appear whenever we let a field propagate a long distance, with the very first example in X-ray crystallography [195].

- In Coherent Diffraction Imaging (CDI), ptychography [201], and Fourier ptychography [197], several pictures are taken in the far-field of an object that we modulate. For each picture, a different mask a_i is applied as an element-wise multiplication (it is a random mask in CDI, a translating probe in ptychography, and the pupil function in Fourier ptychography). The model is thus a concatenation of Fourier matrices multiplied by the $A_i = \mathrm{diag}(a_i)$, $i = 1, \dots, k$.

- The random setting is the most amenable to theoretical analysis and discovery of new algorithms [97], [101], [191].
- In holography [210]–[212] and phase diversity [57], [189], one wants to find to find the phase at a given plane to match the target or measured intensities at k different planes. Propagation to each plane is modeled by a propagation operator P_i, $i = 1, \ldots, k$.

There are two powerful regularization strategies that will not be discussed at length here: sparsity and deep learning based. If x only has a few non-zero components, one can dramatically reduce the number of required measurements as proven in compressed sensing [213]. Sparsity-based regularization has strong theoretical guarantees and different possible strategies to find a solution, we refer the reader to [190] for a more-detailed discussion. Deep learning based regularization is a very promising research direction for the future [214], with or without an extensive dataset to train the models on.

B.3 Algorithms for Phase Retrieval

B.3.1 Alternating projections

The first algorithm to solve Phase Retrieval was proposed by Gerchberg and Saxton in 1972 for astronomy [215]. It was formulated for Fourier measurements:

$$y = |Fx|^2 \tag{B.2}$$

where F is the matrix corresponding to a Fourier transform. Also called the Alternating Projection technique, it requires an intensity measurement y in the Fourier space, and some a priori information in the object space, such as an intensity measurement in this space as well [215], or a known support / positivity of the unknown object x [216].

It is an iterative technique, starting from an initial guess x_0. Its name comes from the two projections applied at each iteration. First, we go in the Fourier space and apply the intensity constraint:

$$\hat{y}_t = \sqrt{\frac{y}{|Fx_t|^2}} \cdot Fx_t \tag{B.3}$$

where $\cdot$ denotes an element-wise multiplication. Then, we go in the real space and project using P_ρ, the projector on the set of x satisfying the constraint (applying another intensity constraint for example as in the original paper [215] or enforcing positivity and a finite support):

$$x_{t+1} = P_\rho(F^{-1}\hat{y}_t) \tag{B.4}$$

One can show that the distance between the solution x and the estimate x_t decreases with time t, however there is no general proof of convergence towards the solution. A variant called the Hybrid Input-Output algorithm relaxes the projection,

moving towards its direction with a given step size [216]. This class of algorithms is generally quite robust, even though they are well-understood and convergence is not always achieved.

This approach is still used frequently for phase retrieval problems with a positivity constraint. For example, it has been used in the autocorrelation technique [91] to image small incoherent objects inside complex optical media.

This approach has also been extended to other computational imaging settings, whenever a stack of intensity images is captured. By considering each image as a constraint to satisfy, one can use successive projections to iteratively converge towards a solution satisfying all the constraints. For example, the Ptychographic Iterative Engine is still extensively used in ptychography [202], demonstrating the robustness of the approach.

B.3.2 Gradient-based techniques

Gradient descent is an ubiquitous technique to solve non-linear optimization problems, it has been used extensively to train neural networks for example. For Phase Retrieval, we first start by defining a certain loss function, for example the intensity loss function:

$$l(x) = \| y - |Ax|^2 \|^2 \tag{B.5}$$

$l(x)$ quantifies how good an estimate x is and we want to minimize this cost function. Other loss functions are possible, for example the amplitude loss function:

$$l'(x) = \| \sqrt{y} - |Ax| \|^2 \tag{B.6}$$

Depending on the problem, reconstruction will be more successful with one or the other. One may even resort to Maximum-Likelihood loss functions to include a statistical model of the measurement noise. Gradient descent is possible as soon as we know the gradient $\nabla l(x)$ of the loss function.

In this scheme, we start from an initial guess x_0, that is refined with small descent steps. At each iteration, one computes the gradient $\nabla l(x)$ using an explicit formula, usually called back-propagation for neural networks. The loss function in the neighborhood of x can be approximated by its first order expansion:

$$l(x+\delta) = l(x) + \delta^T \nabla l(x) \tag{B.7}$$

The direction of steepest descent (minimizing the loss for a given norm of δ) is $-\nabla l(x)$.

Gradient descent makes the following update at each iteration:

$$x_{t+1} = x_t - \alpha \nabla l(x_t) \tag{B.8}$$

where α is a step size, usually chosen small enough so that the first order expansion is valid. We may use a line search strategy like backtracking or based on the Wolfe conditions [217] to ensure the loss function decreases at each iteration.

Vanilla gradient descent is still quite slow for many non-linear optimization problems. To speed up gradient descent, it is possible to take into account curvature from the history of the previous points and gradients already evaluated. One may cite

conjugate gradient [218], Newton's second order method, quasi-Newton approaches such as L-BFGS [219], and other techniques proposed to train Deep Neural Networks like ADAM [220]. They accelerate considerable the convergence but are harder to understand and analyze.

B.3.3 Semidefinite relaxations

Relaxation consists in expanding the search space of x to make the optimization simpler. Based on an identity already mentioned in [221] (a consequence of $\mathrm{Tr}(AB) = \mathrm{Tr}(BA)$):

$$y_i = |a_i^\top x|^2 = \mathrm{Tr}(x^\top a_i a_i^\top x) = \mathrm{Tr}(a_i a_i^\top x x^\top) \tag{B.9}$$

$xx^\top$ form an $d \times d$ matrix over which we can perform the optimization. To enforce the result to be of rank 1, the following optimization is performed [222], [223]:

$$\begin{aligned} &\text{Minimize } \mathrm{Tr}(X) \\ &\text{such that } X \geq 0 \text{ and } \mathrm{Tr}(a_i a_i^\top X) = y_i,\ i = 1,\dots,n \end{aligned} \tag{B.10}$$

Exact recovery occurs when $X = xx^\top$. This defines a semidefinite program called Phaselift in [223]. The optimization is now performed on the d^2 parameters of X, instead of d in the original Phase Retrieval problem. This helps avoid local minima and convergence has been proven for a variety of settings [223], [224]. Another relaxation technique called PhaseCut has also been proposed in [225].

This technique has quickly been translated to real-world experiments. For example, [226] presents an application of this relaxation strategy in Fourier Ptychographic Microscopy. This technique is not used that much in practice due to its high computational and memory cost. One needs to compute and store the matrix X of size $d \times d$, which becomes an important bottleneck for high-resolution images.

B.3.4 Bayesian techniques

Bayesian techniques may also be applied to Phase Retrieval [227], and more generally to arbitrary inference problems (the task of estimating parameters from observations). This general framework has proven to be quite powerful as it often performs well with a minimal number of measurements [228].

In this probabilistic approach, each y_i, $i = 1,\dots,n$ is drawn from a distribution $\rho(y|a_i^\top x)$. In the noiseless case for example, $y_i = |a_i^\top x|^2$, but one may add noise to this model, like gaussian noise with fixed variance or poissonian noise for shot-noise limited measurements.

For many observations of (input a_i, output y_i) pairs, one would like to estimate the parameters x of the model. This is an inference problem, for which we use the famous Bayes formula in probability:

$$\mathbb{P}(x|y) \propto \mathbb{P}(y|x)\mathbb{P}(x) = \prod_{i=1}^{k} \rho(y_i|a_i^\top x)\mathbb{P}(x) \tag{B.11}$$

$\mathbb{P}(x)$ denotes the prior distribution of x. We typically assume each component of x to be drawn independently from a distribution $\mu(x)$. This distribution is usually

gaussian if no other assumption is specified, but it also makes Bayesian methods very appealing for sparse reconstructions, as this constraint can be included in the prior with a Bernoulli-Gauss model $\mu(x) = (1-\lambda)\delta(x) + \lambda\mathcal{N}(0,1)$.

From this general observation, iterative models to obtain a reliable estimate of x have been derived, starting with Belief Propagation for sparse matrices [229], AMP for the linear case [230], and GAMP for the generalized linear model case (including Phase Retrieval) [227], [231], with recent developments in prSAMP and prVAMP [109].

In this Bayesian setting, statistical physics provides valuable tools to describe when the problem is solvable. They describe the random case, when the matrix A is drawn from an i.i.d. distribution, and as such may not apply for other imaging settings. Still these results quantitatively tell us how many patterns n are required before reconstruction is possible. The larger n is, the more information we have and the problem becomes more easily solvable. There are three different phases [228] that are a function of the oversampling ratio n/d: an impossible phase where it is not possible to get any information about x, a hard phase where it is theoretically possible to recover some information on x but no algorithm is presently capable of doing so, and an easy phase where polynomial algorithms such as AMP are capable of finding a good solution.

B.4 Spectral methods and recovery thresholds

Spectral methods first appeared a little before 2015 in [98] and [97], and was used subsequently in many other works to refine this idea [99], [232]. Strong theoretical results soon followed with [101] and [102]. We would like to stress that a better understanding of these techniques only dates from the late 2010s, showing the novelty and the interest of the community in these approaches. On the side, it is worth noting that similar approaches were also proposed in parallel in [233] where they relied on a graph approach to solve the particular phase retrieval problem of ptychography.

Spectral methods are simple-to-code methods to find initial estimates, for the random setting or related, like Coded Diffraction Imaging. They are typically used in conjunction with gradient descent to refine this spectral initial estimate. The recent theoretical results have also improved this pipeline a lot, making their performance comparable to Bayesian algorithms.

In this thesis, we used them first as they adapted well to solve the Multiplexed Phase Retrieval problem. Then in the second part, they were used to solve a classical phase retrieval task, to find a Transmission Matrix. We will here spend more time to explain the background and advances with spectral methods, as they have been developed and used in the presented work. It will be the opportunity to introduce phase diagrams predicting when recovery is possible.

B.4.1 Intuition

The intuition of spectral methods is the following: for one intensity measurement $y_i = |a_i^\top x|^2$, when a_i is correlated with x (i.e. a little aligned with the direction of x),

the measured intensity y_i is increased. On the other hand, if a_i is orthogonal to x, y_i is equal to 0.

Hence, for example, one may try to use this simple observation to obtain a simple guess of the solution in the random setting. We can define for example

$$z_{\text{lin}} = \frac{1}{n}\sum_i y_i a_i \tag{B.12}$$

and hope it removes orthogonal vectors and points towards the direction of x. We would typically like that this estimate correlates with x when the number of measurements n gets very large. In this case, this empirical mean converges towards the expectation value of ya. However, since there is a symmetry between x and $-x$, the expectation value (over the i.i.d. gaussian distribution of a) is 0 and z only follows random walk centered in 0. The covariance matrix of this random walk is still biased in the direction of x, which calls for second order methods.

For second order methods, a good starting point is the empirical covariance matrix, a common object in Random Matrix Theory:

$$Z_0 = \frac{1}{n}\sum_i a_i a_i^T \tag{B.13}$$

Each component $[Z_0]_{kl} = \frac{1}{n}\sum_i [a_i]_k [a_i]_l$ corresponds to an empirical evaluation of the covariance between two components of a. Since the matrix A is drawn from i.i.d. random variables, the covariance matrix is identity and this empirical estimate converges towards it for large n.

To provide a bias, we then define a Weighted Covariance Matrix (WCM):

$$Z = \frac{1}{n}\sum_i y_i a_i a_i^T \tag{B.14}$$

Here, the weights make the expectation value of the matrix biased towards x (the proof is presented in Chapter 3):

$$\mathbb{E}(Z) = I_d + xx^T \tag{B.15}$$

Thus the leading eigenvector of Z for large n will correlate with x.

This is a first example of a spectral method, introduced in [97]. To use them, one only needs to define the WCM Z and use power iterations to retrieve its leading eigenvector (multiplying repeatedly by Z). This process is very efficient and robust, however it requires a relatively-high number of samples $n \sim O(d \log d)$.

B.4.2 Optimal spectral methods

To improve this initial spectral method and make it work in different settings, a preprocessing function may be introduced:

$$Z = \frac{1}{n}\sum_i \tau(y_i) a_i a_i^T \tag{B.16}$$

As long as the function τ is increasing, the WCM will remain biased towards the target solution x (as we give more weight to terms where a_i is correlated with x). For example, [99] introduced a threshold $T > 0$ to avoid very large values of y_i to dominate in the sum:

$$\tau_1(y) = \begin{cases} y & \text{if } y \leq T \\ 0 & \text{otherwise} \end{cases} \tag{B.17}$$

With this simple change, they were able to prove sample complexity in $O(d)$. This means that as soon as the oversampling ratio $\alpha = n/d$ is larger than a certain constant, the algorithm returns a solution close to x.

In both [97] and [99], this spectral method is usually followed by a gradient descent algorithm. Thus, they can be seen as a robust way to initialize an iterative algorithm: power iterations are guaranteed to converge as long as we control the largest and smallest eigenvalues of Z.

To recover some information about x with the least amount of samples n, Mondelli & Montanari have proposed this optimal preprocessing function:

$$\tau_2(y) = \frac{y-1}{y+\sqrt{\alpha}-1} \tag{B.18}$$

This preprocessing function is optimal in the sense that as soon as $\alpha > 1$, the leading eigenvector of the WCM is correlated with x, while an information-theoretic argument shows that no information about x may be recovered when $\alpha < 1$.

Optimality is always dependent on the metric to define it. In [102], they proposed a further improvement of the preprocessing function:

$$\tau_3(y) = 1 - \frac{1}{y} \tag{B.19}$$

This preprocessing is called optimal as for any given oversampling α, the leading eigenvector of the WCM is maximally correlated with x. Interestingly, its formula corresponds to τ_2 when $\alpha = 1$.

In practice, we use power iterations to compute efficiently the leading eigenvector of Z. It consists in repeatedly multiply by the matrix Z until we converge towards the leading eigenvector. Power iterations are fast as we converge geometrically towards this leading eigenvector with a ratio corresponding to the ratio between the two leading eigenvalues. With this technique, we need to make sure that there are no large negative eigenvalues, which is a problem for the two optimal preprocessing function described above. We introduce a lower threshold to avoid these issues and regularize the power iterations by adding a constant times identity to Z.

B.4.3 Recovery thresholds

To conclude this section, we would like to discuss recovery thresholds which have recently been demonstrated for the random setting. There are two thresholds to distinguish:

- Weak recovery (WR) threshold: the minimal value of the oversampling α for an algorithm to output a vector with a positive correlation with the target solution x.

- Full recovery (FR) threshold: the minimal value of the oversampling α for an algorithm to output the perfect solution x.

The study of these thresholds presents two steps. First, an information theoretic argument provides lower bounds of these thresholds. For example, for our complex gaussian setting of Phase Retrieval, [234] has proven that no information about x can be recovered for $\alpha \leq 1$, so that $\alpha_{\text{WR, IT}} = 1$ and [235] has derived the full-recovery threshold $\alpha_{\text{FR, IT}} = 2$.

However, these arguments do not guarantee the existence of an algorithm that recovers this solution in polynomial time for these oversampling values. We then need to define algorithmic recovery thresholds, i.e. minimal values for which there is a known algorithm to reconstruct a suitable solution. For example, spectral methods show that weak recovery is possible starting at $\alpha = 1$, which means that $\alpha_{\text{WR, Algo}} = \alpha_{\text{WR, IT}} = 1$ [101]. On the other hand, there is no algorithm that guarantees full recovery starting at $\alpha = 2$. Instead, Bayesian techniques are conjectured to be the most efficient ones, enabling full recovery starting at $\alpha_{\text{FR, Algo}} \approx 2.027$ [235]. Note that as of today, there is a gap between the information-theoretic bound and the actual full-recovery threshold in practice, as we do not have an algorithm to fully recover x with $\alpha > 2$.

B.5 Discussion

We have presented a broad overview of algorithms for Phase Retrieval, focusing particularly on recent results about spectral initializations. We have recently gained a better understanding of the non-linear optimization underlying Phase Retrieval, knowing when reconstruction is or is not possible in the random setting. These theoretical developments invite practitioners to apply this new class of algorithms to real-life problems, which we have done in Chapter 3 and 4.

Future directions of research include the application of these computational tools in other settings, such as our work on spectral initializations and ptychography [63]. Here the matrix A is not random any more but dictated by the experimental scheme; spectral methods can still provide valuable initializations for iterative algorithms. Additionally, an interesting approach has been to engineering the matrix A to accelerate reconstruction [236], [237]. More complex models, like the two-layer networks we have introduced, would also benefit from rigorous reconstruction guarantees. Recent works for related models include [105], [238].

APPENDIX C

Technical proofs

In this chapter of the Appendix, we would like to prove the technical results presented in Chapter 5. We will prove the presented theorems on linear dimensionality reduction, then move to Random Features and their kernel limits. Such proofs are useful to have a better understanding on the usefulness of random projections and their regularity properties.

Let $X = [x_i]_i \in \mathbb{R}^{d\times n}$ be a stack of n samples x_i of dimension d, for $i = 1,\dots,n$. In linear dimensionality reduction, we want to find a projection matrix $U \in \mathbb{R}^{d\times k}$ such that $U^\top X$ keeps as much information as possible on X even if $k \ll n$.

$\|.\|$ will denote the Euclidean norm for vectors and the Frobenius norm for matrices. Let $\mathscr{U} \in \mathbb{R}^{d\times k}$ be the set of orthogonal matrices.

C.1 Principal Component Analysis

We have seen that it is possible to reconstruct an estimate of X, $\tilde{X} = UU^\top X$, from the low-dimensional projection $U^\top x$.

Theorem 10 (Principal Component Analysis)**.** *The matrix U_{PCA} defined as the k leading eigenvectors of $XX^\top$ minimizes the following error metric on the set of orthogonal matrices:*

$$U_{\text{PCA}} = \underset{U\in\mathscr{U}}{\arg\min} \|X - UU^T X\|^2 \tag{C.1}$$

Proof. Let $U \in \mathscr{U}$ be an orthogonal matrix and L be the error metric defined by $L(U) = \|X - UU^T X\|^2$ for $U \in \mathscr{U}$. Thanks to the orthogonality of U, we can use the Pythagorean identity $\|X\|^2 = \|UU^\top X\|^2 + \|(I - UU^\top)X\|^2$:

$$L(U) = \|X\|^2 - \|UU^\top X\|^2 \tag{C.2}$$

We thus want to maximize $\|UU^\top X\|^2$. Since U is an orthogonal matrix, $\|UU^\top X\|^2 = \|U^\top X\|^2$. We can then reorganize this expression as:

$$\|U^\top X\|^2 = \underset{U}{\arg\max} \sum_i u_i^\top (XX^\top) u_i \tag{C.3}$$

This last quantity is maximal for U_{PCA}, a classical result from spectral analysis of matrices (that one may rederive by expressing both the $XX^\top$ and the u_i, $i = 1,\dots,n$ in the eigenvector basis of $XX^\top$). □

C.2 Random projections

We now move to random projections, for which the optimality result of the previous case is replaced by an inequality quantifying how a random dimensionality reduction preserves pairwise distances.

Theorem 11 (Johnson-Lindenstrauss)**.** *Given* $0 < \epsilon < 1$ *and an output dimension* $k > 20\log(n)/\epsilon^2$*, then assuming the entries of* U *are sampled independently from* $\mathcal{N}\left(0, \frac{1}{k}\right)$*, we have with probability at least* $1 - 2e^{-(\epsilon^2-\epsilon^3)k/4}$*:*

$$(1-\epsilon)\|x-y\|^2 \leq \|U^\top(x-y)\|^2 \leq (1+\epsilon)\|x-y\|^2 \tag{C.4}$$

Proof. Assuming $x \neq y$, we define for $j = 1, \ldots, k$:

$$\tilde{Z}_j = \sqrt{k}[U^\top(x-y)]_j / \|x-y\| \tag{C.5}$$

Each $\tilde{Z}_j$ is a random scalar variable distributed following a normal distribution $\mathcal{N}(0,1)$, independent from $\tilde{Z}_{i\neq j}$.

We bound the failure probability of one side:

$$\mathbb{P}\left(\|U^\top(x-y)\|^2 > (1+\epsilon)\|x-y\|^2\right) = \mathbb{P}\left(\sum_{i=1}^{k} \tilde{Z}_i^2 > (1+\epsilon)k\right) \tag{C.6}$$

$\sum_{i=1}^{k} \tilde{Z}_i^2$ follows the chi-squared distribution with k degrees of freedom, for which we can use concentration bounds. With Markov's inequality:

$$\mathbb{P}\left(\sum_{i=1}^{k} \tilde{Z}_i^2 > (1+\epsilon)k\right) = \mathbb{P}\left(e^{\lambda\sum_{i=1}^{k}\tilde{Z}_i^2} > e^{(1+\epsilon)k\lambda}\right) \tag{C.7}$$

$$\leq \frac{\mathbb{E}[e^{\lambda\sum_{i=1}^{k}\tilde{Z}_i^2}]}{e^{(1+\epsilon)k\lambda}} \tag{C.8}$$

$$= \frac{\mathbb{E}[e^{\lambda\tilde{Z}_1^2}]}{e^{(1+\epsilon)k\lambda}} \tag{C.9}$$

$$= e^{-(1+\epsilon)k\lambda}\left(\frac{1}{1-2\lambda}\right)^{k/2} \tag{C.10}$$

where the expectation of $e^{\lambda\tilde{Z}_1^2}$ has been evaluated explicitly. We then choose $\lambda = \frac{\epsilon}{2(1+\epsilon)}$ to minimize the expression, to obtain:

$$\mathbb{P}\left(\sum_{i=1}^{k} \tilde{Z}_i^2 > (1+\epsilon)k\right) = ((1+\epsilon)e^{-\epsilon})^{k/2} \tag{C.11}$$

$$\leq \exp\left(-\frac{k}{4}(\epsilon^2-\epsilon^3)\right) \tag{C.12}$$

using $1+\epsilon \leq \exp(\epsilon - (\epsilon^2-\epsilon^3)/2)$. □

Note that the same proof implies $\|U^\top X\| \approx \|X\|$, in contrast with U_{PCA} which maximizes $\|U^\top X\|$.

C.3 Random Features

We now move onto non-linear random projections with Random Features to approximate translation-invariant kernels. We define Random Features as [22]:

$$\phi(x) = \frac{1}{\sqrt{k}}[\cos(Wx), \sin(Wx)] \tag{C.13}$$

with $x \in \mathbb{R}^d$ an initial data point and $W \in \mathbb{R}^{k \times d}$ an i.i.d. random projection matrix.

Theorem 12. *For any translation-invariant positive definite kernel $k(x, y) = k(x - y)$, there exists a probability distribution $p(w)$ for the elements of W such that:*

$$\lim_{k \to \infty} \phi(x)^\top \phi(y) = k(x - y) \tag{C.14}$$

Proof. After properly rescaling the kernel function to have $k(0) = 1$, we use Bochner's theorem, stating that the positive definite kernel $k(x - y)$ is necessarily the Fourier transform of a probability measure $p(w)$:

$$k(x - y) = \int p(w) e^{iw^\top (x-y)} dw = \mathbb{E}_w[e^{iw^\top (x-y)}] \tag{C.15}$$

Since $k(x - y)$ is a real number, we can take the real part of both sides:

$$k(x - y) = \mathbb{E}_w[\cos(w^\top (x - y))] \tag{C.16}$$

$$= \mathbb{E}_w[\cos(w^\top x)\cos(w^\top y) + \sin(w^\top x)\sin(w^\top y)] \tag{C.17}$$

□

C.4 Optical Random Features

For Optical Random Features defined as:

$$\phi(x) = |Wx|^2 \tag{C.18}$$

with W an i.i.d. complex gaussian random matrix, we have the following result, proven in Chapter 5:

Theorem 13. *The kernel k approximated by the Optical Random Features is given by:*

$$k(x, y) = \|x\|\|y\|^2 + (x^\top y)^2 \tag{C.19}$$

APPENDIX D

Optical RC implementation details

This Appendix chapter contains practical information to implement Optical Reservoir Computing. Here we wrote down the experience and know-how accumulated during a few years of work in the group.

D.1 Sending an SLM image

At each iteration t, the image displayed on the Spatial Light Modulator (SLM) is composed of the current input $i^{(t)}$ and the reservoir state $x^{(t)}$. These vectors are concatenated and displayed on the high-resolution SLM using macropixels. The size of the macropixels is dependent on the dimension of the input d and the size of the reservoir n. In practice, an important parameter to control is the relative area of input and reservoir on the final SLM image. This ensures consistent results when varying the reservoir size.

In [28], we performed a hyper-parameter search on a noiseless simulation model for the time-series prediction task, and have obtained an optimal performance when the input area is around ten times larger than the reservoir area. However, in this case the contribution of the reservoir is very small: one-tenth of the signal after one iteration and approximately one-hundredth after two. Such a small perturbation would be quickly lost due to experimental noise, which would be detrimental for the capacity of the reservoir to remember the past. We thus chose to fix the reservoir area to be equal to the input area. In [54], we performed the hyperparameter search directly on the optical system, to obtain optimal parameters for the implementation at hand.

A portion of the SLM also remains constant and unmodulated, generating a random bias on the output field after propagation through the complex medium. The variance of this bias also represents another hyperparameter to tune, and we have observed in practice that it helps improve the performance of the algorithm, maybe by enriching the diversity of reservoir activations.

An encoding step is necessary to take into account physical constraints of the modulation devices, its impact and possible strategies are discussed in Chapter 6. Examples of images displayed on the DMD and LCoS-SLM are provided in Fig. 6.7 and 6.8.

D.2 Reading the camera image

After the propagation in the scattering medium, the camera returns a speckle pattern to the computer. This image represents a random projection of the input and reservoir state, as this data is displayed on the SLM. Such a random image presents small

speckle grains that are a few pixels wide, their size being determined by the diffraction of the finite numerical aperture of the optical system. We choose a sampling grid larger than the speckle grain size in order to remove this local range correlation and make sure that the interconnection matrix is fully random.

The intensity values of the speckle pattern follows an exponential distribution, as they are the absolute value square of a complex Gaussian random variable [33]. Its mean depends on the laser power, the thickness of the scattering medium, and the exposure time of the camera. We empirically observed better performance when using the square root of the intensity, i.e. the modulus of the electric field, instead of the raw intensity measured by the camera, probably because this operation regularizes the reservoir state distribution for the subsequent linear regression.

D.3 Training large-scale models

After computing the successive reservoir iterations in optics, we obtain a large stack of reservoir states and the corresponding prediction outputs. A linear regression is performed to find the set of output weights. Any algorithm for linear regression is possible as long as it enables an L2 regularization, and we have found success with the Ridge function of the scikit-learn library [142] and a custom model based on the Cholesky factorization of the Pytorch library [239]. A high value of the regularization parameter was necessary to compensate experimental noise, especially short-term fluctuations coming from detector noise, SLM flickering or mechanical vibrations. Long-term deviations are also present, coming from the decorrelation of the scattering medium, and they impose a maximal time for the experiment (typically an hour for [28] and a day in [54]).

We concatenate the reservoir state with the current input for the linear prediction step. This adds more parameters for the learning and in practice improves greatly the final results. One heuristic explanation is that with this small change, the reservoir is used to predict perturbations over the current input rather than the new input altogether.

D.4 Driving the reservoirs with autonomous dynamics

Following previous strategies developed for chaotic time series prediction [17], we train our algorithms in [54], [56] only to perform next-time-step prediction, from t to $t+1$. To predict further in the future, this prediction is then fed back into the algorithm to iterate further in time. This defines an autonomous dynamical system that should be synchronized with the chaotic time series if training is successful.

Another possible strategy would be to use a given reservoir state to predict T_{pred} time steps in the future, as done in [27], [28]. The output dimension $k = d\,T_{\text{pred}}$ is larger and the learning task becomes more difficult.

We show here the usefulness of this strategy based on autonomous dynamics. In Fig. D.1, we show the performance of Reservoir Computing prediction on the Kuramoto-Sivashinsky dataset, with and without recursive prediction. With recur-

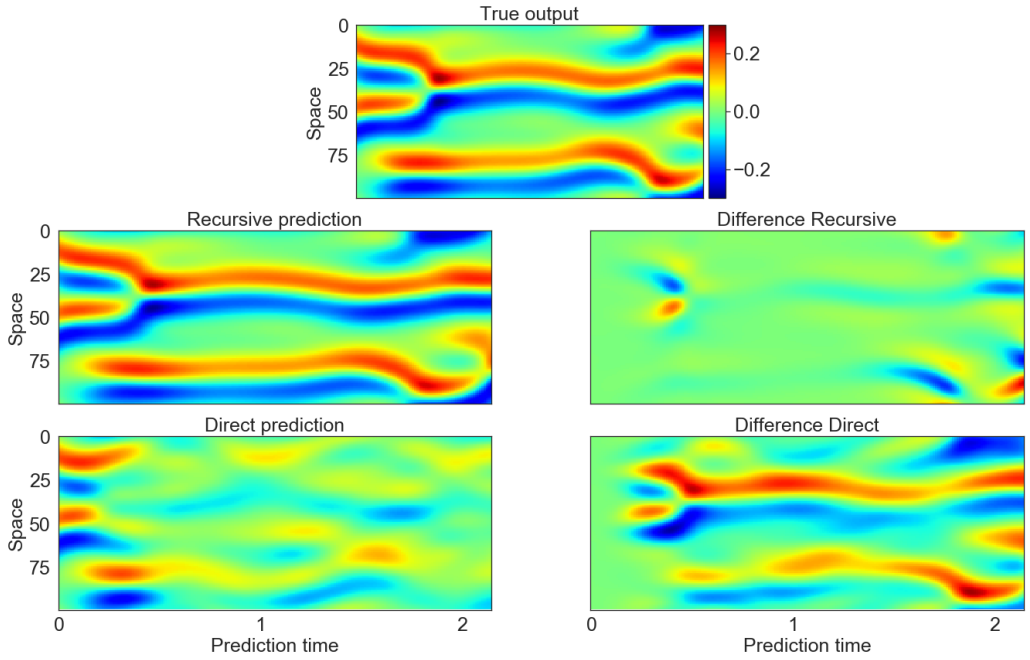


Figure D.1: Comparison of recursive and non-recursive prediction. We see that with recursive prediction (left), Reservoir Computing is able to predict quite precisely up to at least 2 characteristic times. On the other hand, without recursive prediction, Reservoir Computing quickly has a hard time to guess the future of the KS system and outputs its mean for long prediction times.

sive prediction (left), this corresponds to the strategy already presented in Fig. 6.9, and it is not surprising that prediction up to at least 2 Lyapunov exponents is possible. Without recursive prediction (right), the algorithm has a much harder time to predict the future of the chaotic system. Instead, after a short while, it only returns the average value of the time series.

Note that the same hyperparameters were used in both cases. While it may be possible to improve the performance of the direct prediction strategy, by increasing the size of the reservoir or playing with regularization parameter, but we show here the simplicity and effectiveness of the recursive prediction strategy.

D.5 Using batches

Finally, some optical devices are optimized to send images by batches to maximize the speed of data transfer and synchronization between SLM and camera. This is the case of the LightOn device accessed in the cloud as in 2018. In [28], we therefore compute batches of reservoirs in parallel driven by different time series, between 500 and 3,000. In the end, with a time series length of 800, we typically obtain tens of thousand of examples for training and testing.

To show the importance of this batch size, we achieve 640 Hz with a batch size of 3,000 and 285 Hz with a batch size of 500. Compared to our previous work [27], this represents a two-fold increase of operating frequency, thanks to hardware opti-

mization. An even faster implementation should be possible by sending directly the camera image to be displayed on the DMD with dedicated electronics, thus avoiding the computer in the RC loop to reduce latency.

This batch strategy was not necessary for the LCoS-SLM implementation, for which we were limited by the operating frequency of the Liquid Crystal device.

Bibliography

[1] R. R. Schaller, "Moore's law: Past, present and future," *IEEE spectrum*, vol. 34, no. 6, pp. 52–59, 1997.

[2] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain.," *Psychological review*, vol. 65, no. 6, p. 386, 1958.

[3] M. Olazaran, "A sociological study of the official history of the perceptrons controversy," *Social Studies of Science*, vol. 26, no. 3, pp. 611–659, 1996.

[4] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, *et al.*, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[5] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, 2012, pp. 1097–1105.

[6] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, *et al.*, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[7] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[8] Y. Tian, K. Pei, S. Jana, and B. Ray, "Deeptest: Automated testing of deep-neural-network-driven autonomous cars," in *Proceedings of the 40th international conference on software engineering*, 2018, pp. 303–314.

[9] ITU, "Measuring the information society," 2015. [Online]. Available: `https://www.itu.int/en/ITU-D/Statistics/Documents/publications/misr2015/MISR2015-w5.pdf` (visited on 07/09/2020).

[10] R. Hecht-Nielsen, "Theory of the backpropagation neural network," in *Neural networks for perception*, Elsevier, 1992, pp. 65–93.

[11] B. Zoph and Q. V. Le, "Neural architecture search with reinforcement learning," *arXiv preprint arXiv:1611.01578*, 2016.

[12] H. Jaeger, "The "echo state" approach to analysing and training recurrent neural networks-with an erratum note," *Bonn, Germany: German National Research Center for Information Technology GMD Technical Report*, vol. 148, no. 34, p. 13, 2001.

[13] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in *International Conference on Machine Learning*, 2013, pp. 1310–1318.

[14] H. Salehinejad, S. Sankar, J. Barfett, E. Colak, and S. Valaee, "Recent advances in recurrent neural networks," *arXiv preprint arXiv:1801.01078*, 2017.

[15] M. Lukoševičius and H. Jaeger, "Reservoir computing approaches to recurrent neural network training," *Computer Science Review*, vol. 3, no. 3, pp. 127–149, 2009.

[16] M. Lukoševičius, H. Jaeger, and B. Schrauwen, "Reservoir computing trends," *KI-Künstliche Intelligenz*, vol. 26, no. 4, pp. 365–371, 2012.

[17] J. Pathak, B. Hunt, M. Girvan, Z. Lu, and E. Ott, "Model-free prediction of large spatiotemporally chaotic systems from data: A reservoir computing approach," *Physical Review Letters*, vol. 120, no. 2, p. 024 102, 2018.

[18] P. R. Vlachas, J. Pathak, B. R. Hunt, T. P. Sapsis, M. Girvan, E. Ott, and P. Koumoutsakos, "Backpropagation algorithms and reservoir computing in recurrent neural networks for the forecasting of complex spatiotemporal dynamics," *Neural Networks*, 2020.

[19] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Deep image prior," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 9446–9454.

[20] K. C. Zhou and R. Horstmeyer, "Diffraction tomography with a deep image prior," *Optics Express*, vol. 28, no. 9, pp. 12 872–12 896, 2020.

[21] F. Wang, Y. Bian, H. Wang, M. Lyu, G. Pedrini, W. Osten, G. Barbastathis, and G. Situ, "Phase imaging with an untrained neural network," *Light: Science & Applications*, vol. 9, no. 1, pp. 1–7, 2020.

[22] A. Rahimi and B. Recht, "Random features for large-scale kernel machines," in *Advances in Neural Information Processing Systems*, 2008, pp. 1177–1184.

[23] ——, "Weighted sums of random kitchen sinks: Replacing minimization with randomization in learning," in *Advances in Neural Information Processing Systems*, 2009, pp. 1313–1320.

[24] G.-B. Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme learning machine for regression and multiclass classification," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 42, no. 2, pp. 513–529, 2011.

[25] T. Hofmann, B. Schölkopf, and A. J. Smola, "Kernel methods in machine learning," *The annals of statistics*, pp. 1171–1220, 2008.

[26] A. Saade, F. Caltagirone, I. Carron, L. Daudet, A. Drémeau, S. Gigan, and F. Krzakala, "Random projections through multiple optical scattering: Approximating kernels at the speed of light," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2016, pp. 6215–6219.

[27] J. Dong, S. Gigan, F. Krzakala, and G. Wainrib, "Scaling up echo-state networks with multiple light scattering," in *2018 IEEE Statistical Signal Processing Workshop (SSP)*, IEEE, 2018, pp. 448–452.

[28] J. Dong, M. Rafayelyan, F. Krzakala, and S. Gigan, "Optical reservoir computing using multiple light scattering for chaotic systems prediction," *IEEE Journal of Selected Topics in Quantum Electronics*, vol. 26, no. 1, pp. 1–12, 2019.

[29] A. Rosenfeld and J. K. Tsotsos, "Intriguing properties of randomly weighted networks: Generalizing while learning next to nothing," in *2019 16th Conference on Computer and Robot Vision (CRV)*, IEEE, 2019, pp. 9–16.

[30] C. Gallicchio and S. Scardapane, "Deep Randomized Neural Networks," in *Recent Trends in Learning From Data*, Springer, 2020, pp. 43–68.

[31] J. W. Goodman, *Introduction to Fourier optics*. Roberts and Company Publishers, 2005.

[32] R. W. Boyd, *Nonlinear optics*. Academic press, 2019.

[33] J. W. Goodman, *Speckle phenomena in optics: theory and applications*. Roberts and Company Publishers, 2007.

[34] A. P. Mosk, A. Lagendijk, G. Lerosey, and M. Fink, "Controlling waves in space and time for imaging and focusing in complex media," *Nature Photonics*, vol. 6, no. 5, p. 283, 2012.

[35] S. Rotter and S. Gigan, "Light fields in complex media: Mesoscopic scattering meets wave control," *Reviews of Modern Physics*, vol. 89, no. 1, p. 015 005, 2017.

[36] S. Popoff, G. Lerosey, R. Carminati, M. Fink, A. Boccara, and S. Gigan, "Measuring the transmission matrix in optics: An approach to the study and control of light propagation in disordered media," *Physical Review Letters*, vol. 104, no. 10, p. 100 601, 2010.

[37] C. Moretti and S. Gigan, "Readout of fluorescence functional signals through highly scattering tissue," *arXiv preprint arXiv:1906.02604*, 2019.

[38] Z. Yaqoob, D. Psaltis, M. S. Feld, and C. Yang, "Optical phase conjugation for turbidity suppression in biological samples," *Nature Photonics*, vol. 2, no. 2, p. 110, 2008.

[39] M. Jang, H. Ruan, I. M. Vellekoop, B. Judkewitz, E. Chung, and C. Yang, "Relation between speckle decorrelation and optical phase conjugation (opc)-based turbidity suppression through dynamic scattering media: A study on in vivo mouse skin," *Biomedical Optics Express*, vol. 6, no. 1, pp. 72–85, 2015.

[40] J. Carpenter, B. C. Thomsen, and T. D. Wilkinson, "Degenerate mode-group division multiplexing," *Journal of Lightwave Technology*, vol. 30, no. 24, pp. 3946–3952, 2012.

[41] N. Bozinovic, Y. Yue, Y. Ren, M. Tur, P. Kristensen, H. Huang, A. E. Willner, and S. Ramachandran, "Terabit-scale orbital angular momentum mode division multiplexing in fibers," *Science*, vol. 340, no. 6140, pp. 1545–1548, 2013.

[42] G. Labroille, B. Denolle, P. Jian, P. Genevaux, N. Treps, and J.-F. Morizur, "Efficient and mode selective spatial mode multiplexer based on multi-plane light conversion," *Optics Express*, vol. 22, no. 13, pp. 15 599–15 607, 2014.

[43] M. Plöschner, T. Tyc, and T. Čižmár, "Seeing through chaos in multimode fibres," *Nature Photonics*, vol. 9, no. 8, pp. 529–535, 2015.

[44] H. Defienne, M. Barbieri, I. A. Walmsley, B. J. Smith, and S. Gigan, "Two-photon quantum walk in a multimode fiber," *Science Advances*, vol. 2, no. 1, e1501054, 2016.

[45] S. Leedumrongwatthanakun, L. Innocenti, H. Defienne, T. Juffmann, A. Ferraro, M. Paternostro, and S. Gigan, "Programmable linear quantum networks with a multimode fibre," *Nature Photonics*, vol. 14, no. 3, pp. 139–142, 2020.

[46] M. Buttafava, J. Zeman, A. Tosi, K. Eliceiri, and A. Velten, "Non-line-of-sight imaging using a time-gated single photon avalanche diode," *Optics Express*, vol. 23, no. 16, pp. 20 997–21 011, 2015.

[47] M. O'Toole, D. B. Lindell, and G. Wetzstein, "Confocal non-line-of-sight imaging based on the light-cone transform," *Nature*, vol. 555, no. 7696, pp. 338–341, 2018.

[48] Y. Shen, N. C. Harris, S. Skirlo, M. Prabhu, T. Baehr-Jones, M. Hochberg, X. Sun, S. Zhao, H. Larochelle, D. Englund, *et al.*, "Deep learning with coherent nanophotonic circuits," *Nature Photonics*, vol. 11, no. 7, p. 441, 2017.

[49] J. Chang, V. Sitzmann, X. Dun, W. Heidrich, and G. Wetzstein, "Hybrid optical-electronic convolutional neural networks with optimized diffractive optics for image classification," *Scientific Reports*, vol. 8, no. 1, pp. 1–10, 2018.

[50] K. Wagner and D. Psaltis, "Multilayer optical learning networks," *Applied Optics*, vol. 26, no. 23, pp. 5061–5076, 1987.

[51] H.-Y. S. Li, Y. Qiao, and D. Psaltis, "Optical network for real-time face recognition," *Applied Optics*, vol. 32, no. 26, pp. 5026–5035, 1993.

[52] G. Tanaka, T. Yamane, J. B. Héroux, R. Nakane, N. Kanazawa, S. Takeda, H. Numata, D. Nakano, and A. Hirose, "Recent advances in physical reservoir computing: A review," *Neural Networks*, 2019.

[53] G. Van der Sande, D. Brunner, and M. C. Soriano, "Advances in photonic reservoir computing," *Nanophotonics*, vol. 6, no. 3, pp. 561–576,

[54] M. Rafayelyan, J. Dong, Y. Tan, F. Krzakala, and S. Gigan, "Large-scale optical reservoir computing for spatiotemporal chaotic systems prediction," *Physical Review X*, vol. 10, no. 4, p. 041 037, 2020.

[55] R. Ohana, J. Wacker, J. Dong, S. Marmin, F. Krzakala, M. Filippone, and L. Daudet, "Kernel computations from large-scale random features obtained by optical processing units," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2020, pp. 9294–9298.

[56] J. Dong, R. Ohana, M. Rafayelyan, and F. Krzakala, "Reservoir computing meets recurrent kernels and structured transforms," in *Advances in Neural Information Processing Systems*, 2020.

[57] T. Wu, J. Dong, X. Shao, and S. Gigan, "Imaging through a thin scattering layer and jointly retrieving the point-spread-function using phase-diversity," *Optics Express*, vol. 25, no. 22, pp. 27 182–27 194, 2017.

[58] J. Dong, F. Krzakala, and S. Gigan, "Spectral method for multiplexed phase retrieval and application in optical imaging in complex media," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2019, pp. 4963–4967.

[59] A. Boniface, B. Blochet, J. Dong, and S. Gigan, "Noninvasive light focusing in scattering media using speckle variance optimization," *Optica*, vol. 6, no. 11, pp. 1381–1385, Nov. 2019. DOI: 10.1364/OPTICA.6.001381. [Online]. Available: http://www.osapublishing.org/optica/abstract.cfm?URI=optica-6-11-1381.

[60] A. Boniface, J. Dong, and S. Gigan, "Non-invasive focusing and imaging in scattering media with a fluorescence-based transmission matrix," *Nature Communications*, vol. 11, no. 1, pp. 1–7, 2020.

[61] T. Wu, J. Dong, and S. Gigan, "Non-invasive single-shot recovery of a point-spread function of a memory effect based scattering imaging system," *Optics Letters*, vol. 45, no. 19, pp. 5397–5400, 2020.

[62] F. Soldevila, J. Dong, E. Tajahuerce, S. Gigan, and H. B. de Aguiar, "Fast compressive raman bio-imaging via matrix completion," *Optica*, vol. 6, no. 3, pp. 341–346, 2019.

[63] L. Valzania, J. Dong, and S. Gigan, "Accelerating ptychographic reconstructions using spectral initializations," *arXiv preprint arXiv:2007.14139*, 2020.

[64] M. Adrian, J. Dubochet, J. Lepault, and A. W. McDowall, "Cryo-electron microscopy of viruses," *Nature*, vol. 308, no. 5954, pp. 32–36, 1984.

[65] B. P. Abbott, R. Abbott, T. Abbott, M. Abernathy, F. Acernese, K. Ackley, C. Adams, T. Adams, P. Addesso, R. Adhikari, *et al.*, "Observation of gravitational waves from a binary black hole merger," *Physical Review Letters*, vol. 116, no. 6, p. 061 102, 2016.

[66] K. Akiyama, A. Alberdi, W. Alef, K. Asada, R. Azulay, A.-K. Baczko, D. Ball, M. Baloković, J. Barrett, D. Bintley, *et al.*, "First m87 event horizon telescope results. iv. imaging the central supermassive black hole," *The Astrophysical Journal Letters*, vol. 875, no. 1, p. L4, 2019.

[67] D. Huang, E. A. Swanson, C. P. Lin, J. S. Schuman, W. G. Stinson, W. Chang, M. R. Hee, T. Flotte, K. Gregory, C. A. Puliafito, *et al.*, "Optical coherence tomography," *Science*, vol. 254, no. 5035, pp. 1178–1181, 1991.

[68] K. Quinn. (2014). "Pushing the limits of imaging resolution and penetration depth," [Online]. Available: https://www.osa.org/en-us/the_optical_society_blog/2014/february_2014/pushing_the_limits_of_imaging_resolution_and_penet/ (visited on 07/09/2020).

[69] J. W. Lichtman and J.-A. Conchello, "Fluorescence microscopy," *Nature Methods*, vol. 2, no. 12, p. 910, 2005.

[70] T. A. Weissman, J. R. Sanes, J. W. Lichtman, and J. Livet, "Generating and imaging multicolor brainbow mice," *Cold Spring Harbor Protocols*, vol. 2011, no. 7, pdb–top114, 2011.

[71] D. A. Glenar, J. J. Hillman, B. Saif, and J. Bergstralh, "Acousto-optic imaging spectropolarimetry for remote sensing," *Applied Optics*, vol. 33, no. 31, pp. 7412–7424, 1994.

[72] M. Xu and L. V. Wang, "Photoacoustic imaging in biomedicine," *Review of scientific instruments*, vol. 77, no. 4, p. 041 101, 2006.

[73] C. Xu, W. Zipfel, J. B. Shear, R. M. Williams, and W. W. Webb, "Multiphoton fluorescence excitation: New spectral windows for biological nonlinear microscopy," *Proceedings of the National Academy of Sciences*, vol. 93, no. 20, pp. 10 763–10 768, 1996.

[74] I. M. Vellekoop and A. Mosk, "Focusing coherent light through opaque strongly scattering media," *Optics Letters*, vol. 32, no. 16, pp. 2309–2311, 2007.

[75] H. Yu, K. Lee, and Y. Park, "Ultrahigh enhancement of light focusing through disordered media controlled by mega-pixel modes," *Optics Express*, vol. 25, no. 7, pp. 8036–8047, 2017.

[76] O. Katz, E. Small, Y. Guan, and Y. Silberberg, "Noninvasive nonlinear focusing and imaging through strongly scattering turbid layers," *Optica*, vol. 1, no. 3, pp. 170–174, 2014.

[77] M. Hofer, C. Soeller, S. Brasselet, and J. Bertolotti, "Wide field fluorescence epi-microscopy behind a scattering medium enabled by speckle correlations," *Optics Express*, vol. 26, no. 8, pp. 9866–9881, 2018.

[78] B. Blochet, L. Bourdieu, and S. Gigan, "Focusing light through dynamical samples using fast continuous wavefront optimization," *Optics Letters*, vol. 42, no. 23, pp. 4994–4997, 2017.

[79] S. Yang, E. Papagiakoumou, M. Guillon, V. De Sars, C.-M. Tang, and V. Emiliani, "Three-dimensional holographic photostimulation of the dendritic arbor," *Journal of neural engineering*, vol. 8, no. 4, p. 046 002, 2011.

[80] N. Bender, H. Yılmaz, Y. Bromberg, and H. Cao, "Customizing speckle intensity statistics," *Optica*, vol. 5, no. 5, pp. 595–600, 2018.

[81] O. Katz, E. Small, Y. Bromberg, and Y. Silberberg, "Focusing and compression of ultrashort pulses through scattering media," *Nature Photonics*, vol. 5, no. 6, p. 372, 2011.

[82] C.-L. Hsieh, Y. Pu, R. Grange, G. Laporte, and D. Psaltis, "Imaging through turbid layers by scanning the phase conjugated second harmonic radiation from a nanoparticle," *Optics Express*, vol. 18, no. 20, pp. 20 723–20 731, Sep. 2010. DOI: 10.1364/OE.18.020723. [Online]. Available: http://www.opticsexpress.org/abstract.cfm?URI=oe-18-20-20723.

[83] H. Ruan, T. Haber, Y. Liu, J. Brake, J. Kim, J. M. Berlin, and C. Yang, "Focusing light inside scattering media with magnetic-particle-guided wavefront shaping," *Optica*, vol. 4, no. 11, pp. 1337–1343, 2017.

[84] M. Cui and C. Yang, "Implementation of a digital optical phase conjugation system and its application to study the robustness of turbidity suppression by phase conjugation," *Optics Express*, vol. 18, no. 4, pp. 3444–3455, 2010.

[85] A. M. Caravaca-Aguirre, D. B. Conkey, J. D. Dove, H. Ju, T. W. Murray, and R. Piestun, "High contrast three-dimensional photoacoustic imaging through scattering media by localized optical fluence enhancement," *Optics Express*, vol. 21, no. 22, pp. 26 671–26 676, 2013.

[86] T. Chaigne, O. Katz, A. C. Boccara, M. Fink, E. Bossy, and S. Gigan, "Controlling light in scattering media non-invasively using the photoacoustic transmission matrix," *Nature Photonics*, vol. 8, no. 1, p. 58, 2014.

[87] W. Leutz and G. Maret, "Ultrasonic modulation of multiply scattered light," *Physica B: Condensed Matter*, vol. 204, no. 1-4, pp. 14–19, 1995.

[88] X. Xu, H. Liu, and L. V. Wang, "Time-reversed ultrasonically encoded optical focusing into scattering media," *Nature Photonics*, vol. 5, no. 3, pp. 154–157, 2011.

[89] B. Judkewitz, Y. M. Wang, R. Horstmeyer, A. Mathy, and C. Yang, "Speckle-scale focusing in the diffusive regime with time reversal of variance-encoded light (trove)," *Nature Photonics*, vol. 7, no. 4, pp. 300–305, 2013.

[90] O. Katz, F. Ramaz, S. Gigan, and M. Fink, "Controlling light in complex media beyond the acoustic diffraction-limit using the acousto-optic transmission matrix," *Nature Communications*, vol. 10, no. 1, p. 717, 2019.

[91] O. Katz, P. Heidmann, M. Fink, and S. Gigan, "Non-invasive single-shot imaging through scattering layers and around corners via speckle correlations," *Nature Photonics*, vol. 8, no. 10, p. 784, 2014.

[92] I. M. Vellekoop and C. M. Aegerter, "Scattered light fluorescence microscopy: Imaging through turbid layers," *Optics Letters*, vol. 35, no. 8, pp. 1245–1247, 2010.

[93] M. F. Duarte, M. A. Davenport, D. Takhar, J. N. Laska, T. Sun, K. F. Kelly, and R. G. Baraniuk, "Single-pixel imaging via compressive sampling," *IEEE signal processing magazine*, vol. 25, no. 2, pp. 83–91, 2008.

[94] G. Ongie, A. Jalal, C. A. M. R. G. Baraniuk, A. G. Dimakis, and R. Willett, "Deep learning techniques for inverse problems in imaging," *IEEE Journal on Selected Areas in Information Theory*, 2020.

[95] U. S. Kamilov, I. N. Papadopoulos, M. H. Shoreh, A. Goy, C. Vonesch, M. Unser, and D. Psaltis, "Learning approach to optical tomography," *Optica*, vol. 2, no. 6, pp. 517–522, 2015.

[96] M. R. Kellman, E. Bostan, N. A. Repina, and L. Waller, "Physics-based learned design: Optimized coded-illumination for quantitative phase imaging," *IEEE Transactions on Computational Imaging*, vol. 5, no. 3, pp. 344–353, 2019.

[97] E. J. Candes, X. Li, and M. Soltanolkotabi, "Phase retrieval via Wirtinger flow: Theory and algorithms," *IEEE Transactions on Information Theory*, vol. 61, no. 4, pp. 1985–2007, 2015.

[98] P. Netrapalli, P. Jain, and S. Sanghavi, "Phase retrieval using alternating minimization," in *Advances in Neural Information Processing Systems*, 2013, pp. 2796–2804.

[99] Y. Chen and E. Candes, "Solving random quadratic systems of equations is nearly as easy as solving linear systems," in *Advances in Neural Information Processing Systems*, 2015, pp. 739–747.

[100] Y. M. Lu and G. Li, "Phase transitions of spectral initialization for high-dimensional nonconvex estimation," *arXiv preprint arXiv:1702.06435*, 2017.

[101] M. Mondelli and A. Montanari, "Fundamental limits of weak recovery with applications to phase retrieval," in *Conference On Learning Theory*, PMLR, 2018, pp. 1445–1450.

[102] W. Luo, W. Alghamdi, and Y. M. Lu, "Optimal spectral initialization for signal recovery with applications to phase retrieval," *IEEE Transactions on Signal Processing*, vol. 67, no. 9, pp. 2347–2356, 2019.

[103] D. Wang, E. H. Zhou, J. Brake, H. Ruan, M. Jang, and C. Yang, "Focusing through dynamic tissue with millisecond digital optical phase conjugation," *Optica*, vol. 2, no. 8, pp. 728–735, 2015.

[104] Y. Liu, P. Lai, C. Ma, X. Xu, A. A. Grabar, and L. V. Wang, "Optical focusing deep inside dynamic scattering media with near-infrared time-reversed ultrasonically encoded (true) light," *Nature Communications*, vol. 6, no. 1, pp. 1–9, 2015.

[105] B. Aubin, A. Maillard, F. Krzakala, N. Macris, L. Zdeborová, *et al.*, "The committee machine: Computational to statistical gaps in learning a two-layers neural network," in *Advances in Neural Information Processing Systems*, 2018, pp. 3223–3234.

[106] Y. Chi, Y. M. Lu, and Y. Chen, "Nonconvex optimization meets low-rank matrix factorization: An overview," *IEEE Transactions on Signal Processing*, vol. 67, no. 20, pp. 5239–5269, 2019.

[107] N. C. Pégard, H.-Y. Liu, N. Antipa, M. Gerlock, H. Adesnik, and L. Waller, "Compressive light-field microscopy for 3d neural activity recording," *Optica*, vol. 3, no. 5, pp. 517–524, 2016.

[108] A. Drémeau, A. Liutkus, D. Martina, O. Katz, C. Schülke, F. Krzakala, S. Gigan, and L. Daudet, "Reference-less measurement of the transmission matrix of a highly scattering material using a dmd and phase retrieval techniques," *Optics Express*, vol. 23, no. 9, pp. 11 898–11 911, May 2015. DOI: 10.1364/OE.23.011898. [Online]. Available: http://www.opticsexpress.org/abstract.cfm?URI=oe-23-9-11898.

[109] C. A. Metzler, M. K. Sharma, S. Nagesh, R. G. Baraniuk, O. Cossairt, and A. Veeraraghavan, "Coherent inverse scattering via transmission matrices: Efficient phase retrieval algorithms and a public dataset," in *2017 IEEE International Conference on Computational Photography (ICCP)*, IEEE, 2017, pp. 1–16.

[110] S. S. Mannelli, E. Vanden-Eijnden, and L. Zdeborová, "Optimization and generalization of shallow neural networks with quadratic activation functions," *arXiv preprint arXiv:2006.15459*, 2020.

[111] LightOn. (2020). "Lighton official website," [Online]. Available: http://www.lighton.ai (visited on 09/02/2020).

[112] P. Ambs, "Optical computing: A 60-year adventure," *Advances in Optical Technologies*, vol. 2010, 2010.

[113] E. O'Neill, "Spatial filtering in optics," *IRE Transactions on Information Theory*, vol. 2, no. 2, pp. 56–65, 1956.

[114] L. Cutrona, E. Leith, C. Palermo, and L. Porcello, "Optical data processing and filtering systems," *IRE Transactions on Information Theory*, vol. 6, no. 3, pp. 386–400, 1960.

[115] J. Armitage and A. Lohmann, "Character recognition by incoherent spatial filtering," *Applied optics*, vol. 4, no. 4, pp. 461–467, 1965.

[116] C. Weaver and J. W. Goodman, "A technique for optically convolving two functions," *Applied optics*, vol. 5, no. 7, pp. 1248–1249, 1966.

[117] E. N. Leith, "The evolution of information optics," *IEEE Journal of Selected Topics in Quantum Electronics*, vol. 6, no. 6, pp. 1297–1304, 2000.

[118] K. Preston, *Coherent optical computers*. McGraw-Hill, 1972.

[119] B. J. Lechner, F. J. Marlowe, E. O. Nester, and J. Tults, "Liquid crystal matrix displays," *Proceedings of the IEEE*, vol. 59, no. 11, pp. 1566–1579, 1971.

[120] G. Labrunie, J. Robert, and J. Borel, "A 128× 128 electro-optical interface for real time data processing," *Revue de Physique Appliquée*, vol. 10, no. 3, pp. 143–146, 1975.

[121] Hamamatsu. (2014). "Phase spatial light modulator lcos-slm," [Online]. Available: `https://www.hamamatsu.com/resources/pdf/ssd/e12_handbook_lcos_slm.pdf` (visited on 09/02/2020).

[122] J. Bueno, S. Maktoobi, L. Froehly, I. Fischer, M. Jacquot, L. Larger, and D. Brunner, "Reinforcement learning in a large-scale photonic recurrent neural network," *Optica*, vol. 5, no. 6, pp. 756–760, 2018.

[123] D. Pierangeli, G. Marcucci, and C. Conti, "Large-scale photonic ising machine by spatial light modulation," *Physical Review Letters*, vol. 122, no. 21, p. 213 902, 2019.

[124] J. P. Cunningham and Z. Ghahramani, "Linear dimensionality reduction: Survey, insights, and generalizations," *The Journal of Machine Learning Research*, vol. 16, no. 1, pp. 2859–2900, 2015.

[125] H. Abdi and L. J. Williams, "Principal component analysis," *Wiley interdisciplinary reviews: computational statistics*, vol. 2, no. 4, pp. 433–459, 2010.

[126] W. B. Johnson and J. Lindenstrauss, "Extensions of lipschitz mappings into a hilbert space," *Contemporary Mathematics*, vol. 26, no. 189-206, p. 1, 1984.

[127] L. Van Der Maaten, E. Postma, and J. Van den Herik, "Dimensionality reduction: A comparative," *Journal of Machine Learnining Research*, vol. 10, no. 66-71, p. 13, 2009.

[128] A. Ng *et al.*, "Sparse autoencoder," *CS294A Lecture notes*, vol. 72, no. 2011, pp. 1–19, 2011.

[129] M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni, "Locality-sensitive hashing scheme based on p-stable distributions," in *Proceedings of the twentieth annual symposium on Computational geometry*, 2004, pp. 253–262.

[130] A. Dasgupta, R. Kumar, and T. Sarlós, "Fast locality-sensitive hashing," in *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2011, pp. 1073–1081.

[131] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, vol. 70, no. 1-3, pp. 489–501, 2006.

[132] Z.-L. Sun, T.-M. Choi, K.-F. Au, and Y. Yu, "Sales forecasting using extreme learning machine with applications in fashion retailing," *Decision Support Systems*, vol. 46, no. 1, pp. 411–419, 2008.

[133] Y. Wang, F. Cao, and Y. Yuan, "A study on effectiveness of extreme learning machine," *Neurocomputing*, vol. 74, no. 16, pp. 2483–2490, 2011.

[134] B. Schölkopf, A. J. Smola, F. Bach, *et al.*, *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2002.

[135] N. Keriven, D. Garreau, and I. Poli, "Newma: A new method for scalable model-free online change-point detection," *IEEE Transactions on Signal Processing*, vol. 68, pp. 3515–3528, 2020.

[136] A. Nøkland, "Direct feedback alignment provides learning in deep neural networks," in *Advances in Neural Information Processing Systems*, 2016, pp. 1037–1045.

[137] J. Launay, I. Poli, K. Müller, I. Carron, L. Daudet, F. Krzakala, and S. Gigan, "Light-in-the-loop: Using a photonics co-processor for scalable training of neural networks," *arXiv preprint arXiv:2006.01475*, 2020.

[138] H. Jaeger and H. Haas, "Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication," *Science*, vol. 304, no. 5667, pp. 78–80, 2004.

[139] D. Verstraeten, B. Schrauwen, M. d'Haene, and D. Stroobandt, "An experimental unification of reservoir computing methods," *Neural networks*, vol. 20, no. 3, pp. 391–403, 2007.

[140] P. J. Werbos *et al.*, "Backpropagation through time: What it does and how to do it," *Proceedings of the IEEE*, vol. 78, no. 10, pp. 1550–1560, 1990.

[141] L. Grigoryeva and J.-P. Ortega, "Echo state networks are universal," *Neural Networks*, vol. 108, pp. 495–508, 2018.

[142] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, *et al.*, "Scikit-learn: Machine learning in python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[143] K. Vandoorne, P. Mechet, T. Van Vaerenbergh, M. Fiers, G. Morthier, D. Verstraeten, B. Schrauwen, J. Dambre, and P. Bienstman, "Experimental demonstration of reservoir computing on a silicon photonics chip," *Nature Communications*, vol. 5, p. 3541, 2014.

[144] M. Dale, J. F. Miller, S. Stepney, and M. A. Trefzer, "Evolving carbon nanotube reservoir computers," in *International Conference on Unconventional Computation and Natural Computation*, Springer, 2016, pp. 49–61.

[145] C. Fernando and S. Sojakka, "Pattern recognition in a bucket," in *European conference on artificial life*, Springer, 2003, pp. 588–597.

[146] V. Sze, Y.-H. Chen, J. Emer, A. Suleiman, and Z. Zhang, "Hardware for machine learning: Challenges and opportunities," in *2017 IEEE Custom Integrated Circuits Conference (CICC)*, IEEE, 2017, pp. 1–8.

[147] P. Antonik, A. Smerieri, F. Duport, M. Haelterman, and S. Massar, "FPGA implementation of reservoir computing with online learning," in *24th Belgian-Dutch Conference on Machine Learning*, 2015.

[148] C. Donahue, C. Merkel, Q. Saleh, L. Dolgovs, Y. K. Ooi, D. Kudithipudi, and B. Wysocki, "Design and analysis of neuromemristive echo state networks with limited-precision synapses," in *2015 IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA)*, IEEE, 2015, pp. 1–6.

[149] L. Appeltant, M. C. Soriano, G. Van der Sande, J. Danckaert, S. Massar, J. Dambre, B. Schrauwen, C. R. Mirasso, and I. Fischer, "Information processing using a single dynamical node as complex system," *Nature Communications*, vol. 2, p. 468, 2011.

[150] L. Larger, M. C. Soriano, D. Brunner, L. Appeltant, J. M. Gutiérrez, L. Pesquera, C. R. Mirasso, and I. Fischer, "Photonic information processing beyond Turing: an optoelectronic implementation of reservoir computing," *Optics Express*, vol. 20, no. 3, pp. 3241–3249, 2012.

[151] Y. Paquot, F. Duport, A. Smerieri, J. Dambre, B. Schrauwen, M. Haelterman, and S. Massar, "Optoelectronic reservoir computing," *Scientific reports*, vol. 2, p. 287, 2012.

[152] P. Antonik, N. Marsal, D. Brunner, and D. Rontani, "Performance analysis of a large-scale photonic reservoir computer on image classification," in *NOLTA*, 2018.

[153] J. Pauwels, G. Van der Sande, A. Bouwens, M. Haelterman, and S. Massar, "Towards high-performance spatially parallel optical reservoir computing," in *Neuro-inspired Photonic Computing*, International Society for Optics and Photonics, vol. 10689, 2018, p. 1 068 904.

[154] B. Schrauwen, L. Büsing, and R. A. Legenstein, "On computational power and the order-chaos phase transition in reservoir computing," in *Advances in Neural Information Processing Systems*, 2009, pp. 1425–1432.

[155] B. Zhang, D. J. Miller, and Y. Wang, "Nonlinear system modeling with random matrices: Echo state networks revisited," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 1, pp. 175–182, 2011.

[156] G. Wainrib and M. N. Galtier, "A local echo state property through the largest lyapunov exponent," *Neural Networks*, vol. 76, pp. 39–45, 2016.

[157] W. Maass, T. Natschläger, and H. Markram, "Real-time computing without stable states: A new framework for neural computation based on perturbations," *Neural computation*, vol. 14, no. 11, pp. 2531–2560, 2002.

[158] M. Dale, J. F. Miller, S. Stepney, and M. A. Trefzer, "A substrate-independent framework to characterise reservoir computers," *arXiv preprint arXiv:1810.07135*, 2018.

[159] M. Mounaix, D. M. Ta, and S. Gigan, "Transmission matrix approaches for nonlinear fluorescence excitation through multiple scattering media," *Optics Letters*, vol. 43, no. 12, pp. 2831–2834, 2018.

[160] M. C. Mackey and L. Glass, "Oscillation and chaos in physiological control systems," *Science*, vol. 197, no. 4300, pp. 287–289, 1977.

[161] Y. Kuramoto, "Diffusion-induced chaos in reaction systems," *Progress of Theoretical Physics Supplement*, vol. 64, pp. 346–367, 1978.

[162] G. Sivashinsky, "Nonlinear analysis of hydrodynamic instability in laminar flames—I. Derivation of basic equations," *Acta astronautica*, vol. 4, pp. 1177–1206, 1977.

[163] A. Rudi and L. Rosasco, "Generalization properties of learning with random features," in *Advances in Neural Information Processing Systems*, 2017, pp. 3215–3225.

[164] L. Carratino, A. Rudi, and L. Rosasco, "Learning with SGD and random features," in *Advances in Neural Information Processing Systems*, 2018, pp. 10 192–10 203.

[165] F. Liu, X. Huang, Y. Chen, and J. A. Suykens, "Random Features for Kernel Approximation: A Survey in Algorithms, Theory, and Beyond," *arXiv preprint arXiv:2004.11154*, 2020.

[166] Q. Le, T. Sarlós, and A. Smola, "Fastfood-computing Hilbert space expansions in loglinear time," in *International Conference on Machine Learning*, 2013, pp. 244–252.

[167] F. X. X. Yu, A. T. Suresh, K. M. Choromanski, D. N. Holtmann-Rice, and S. Kumar, "Orthogonal random features," in *Advances in Neural Information Processing Systems*, 2016, pp. 1975–1983.

[168] M. Hermans and B. Schrauwen, "Recurrent kernel machines: Computing with infinite echo state networks," *Neural Computation*, vol. 24, no. 1, pp. 104–133, 2012.

[169] P. Kar and H. Karnick, "Random feature maps for dot product kernels," in *Artificial Intelligence and Statistics*, 2012, pp. 583–591.

[170] Z. Liao and R. Couillet, "On the spectrum of random features maps of high dimensional data," in *International Conference on Machine Learning*, PMLR, 2018, pp. 3063–3071.

[171] S. Boucheron, G. Lugosi, and O. Bousquet, "Concentration inequalities," in *Summer School on Machine Learning*, Springer, 2003, pp. 208–240.

[172] M. Moczulski, M. Denil, J. Appleyard, and N. de Freitas, "ACDC: A structured efficient linear layer," *arXiv preprint arXiv:1511.05946*, 2015.

[173] M. Arjovsky, A. Shah, and Y. Bengio, "Unitary evolution recurrent neural networks," in *International Conference on Machine Learning*, 2016, pp. 1120–1128.

[174] A. Thomas, A. Gu, T. Dao, A. Rudra, and C. Ré, "Learning compressed transforms with low displacement rank," in *Advances in Neural Information Processing Systems*, `https://github.com/HazyResearch/structured-nets`, 2018, pp. 9052–9060.

[175] K. M. Choromanski, M. Rowland, and A. Weller, "The unreasonable effectiveness of structured random orthogonal embeddings," in *Advances in Neural Information Processing Systems*, 2017, pp. 219–228.

[176] L. Larger, A. Baylón-Fuentes, R. Martinenghi, V. S. Udaltsov, Y. K. Chembo, and M. Jacquot, "High-speed photonic reservoir computing using a time-delay-based architecture: Million words per second classification," *Physical Review X*, vol. 7, no. 1, p. 011 015, 2017.

[177] C. Gallicchio, A. Micheli, and L. Silvestri, "Local lyapunov exponents of deep echo state networks," *Neurocomputing*, vol. 298, pp. 34–45, 2018.

[178] C. Gallicchio, A. Micheli, and L. Pedrelli, "Deep reservoir computing: A critical experimental analysis," *Neurocomputing*, vol. 268, pp. 87–99, 2017.

[179] R. Couillet, G. Wainrib, H. Sevi, and H. T. Ali, "The asymptotic performance of linear echo state neural networks," *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 6171–6205, 2016.

[180] S. S. Schoenholz, J. Gilmer, S. Ganguli, and J. Sohl-Dickstein, "Deep information propagation," *arXiv preprint arXiv:1611.01232*, 2016.

[181] J. Lee, Y. Bahri, R. Novak, S. S. Schoenholz, J. Pennington, and J. Sohl-Dickstein, "Deep neural networks as gaussian processes," *arXiv preprint arXiv:1711.00165*, 2017.

[182] G. Yang, "Wide feedforward or recurrent neural networks of any architecture are gaussian processes," in *Advances in Neural Information Processing Systems*, 2019, pp. 9951–9960.

[183] Q. Wang, Y. Jin, and P. Li, "General-purpose LSM learning processor architecture and theoretically guided design space exploration," in *2015 IEEE Biomedical Circuits and Systems Conference (BioCAS)*, IEEE, 2015, pp. 1–4.

[184] Y. Zhang, P. Li, Y. Jin, and Y. Choe, "A digital liquid state machine with biologically inspired learning and its application to speech recognition," *IEEE transactions on neural networks and learning systems*, vol. 26, no. 11, pp. 2635–2649, 2015.

[185] Y. Jin and P. Li, "Performance and robustness of bio-inspired digital liquid state machines: A case study of speech recognition," *Neurocomputing*, vol. 226, pp. 145–160, 2017.

[186] F. Duport, B. Schneider, A. Smerieri, M. Haelterman, and S. Massar, "All-optical reservoir computing," *Optics Express*, vol. 20, no. 20, pp. 22 783–22 795, 2012.

[187] L. Valzania, T. Feurer, P. Zolliker, and E. Hack, "Terahertz ptychography," *Optics Letters*, vol. 43, no. 3, pp. 543–546, 2018.

[188] P. Berto, C. Scotté, F. Galland, H. Rigneault, and H. B. de Aguiar, "Programmable single-pixel-based broadband stimulated raman scattering," *Optics Letters*, vol. 42, no. 9, pp. 1696–1699, 2017.

[189] R. G. Paxman, T. J. Schulz, and J. R. Fienup, "Joint estimation of object and aberrations by using phase diversity," *JOSA A*, vol. 9, no. 7, pp. 1072–1085, 1992.

[190] Y. Shechtman, Y. C. Eldar, O. Cohen, H. N. Chapman, J. Miao, and M. Segev, "Phase retrieval with application to optical imaging: A contemporary overview," *IEEE signal processing magazine*, vol. 32, no. 3, pp. 87–109, 2015.

[191] F. Fogel, I. Waldspurger, and A. d'Aspremont, "Phase retrieval for imaging problems," *Mathematical programming computation*, vol. 8, no. 3, pp. 311–335, 2016.

[192] J. t. Karle and H. Hauptman, "The phases and magnitudes of the structure factors," *Acta Crystallographica*, vol. 3, no. 3, pp. 181–187, 1950.

[193] D. Sayre, "The squaring method: A new method for phase determination," *Acta Crystallographica*, vol. 5, no. 1, pp. 60–65, 1952.

[194] W. t. Cochran, "Relations between the phases of structure factors," *Acta Crystallographica*, vol. 8, no. 8, pp. 473–478, 1955.

[195] R. P. Millane, "Phase retrieval in crystallography and optics," *JOSA A*, vol. 7, no. 3, pp. 394–411, 1990.

[196] F. Hüe, J. Rodenburg, A. Maiden, F. Sweeney, and P. Midgley, "Wave-front phase retrieval in transmission electron microscopy via ptychography," *Physical Review B*, vol. 82, no. 12, p. 121 415, 2010.

[197] G. Zheng, R. Horstmeyer, and C. Yang, "Wide-field, high-resolution fourier ptychographic microscopy," *Nature Photonics*, vol. 7, no. 9, pp. 739–745, 2013.

[198] N. Antipa, G. Kuo, R. Heckel, B. Mildenhall, E. Bostan, R. Ng, and L. Waller, "Diffusercam: Lensless single-exposure 3d imaging," *Optica*, vol. 5, no. 1, pp. 1–9, 2018.

[199] J. Miao, T. Ishikawa, E. H. Anderson, and K. O. Hodgson, "Phase retrieval of diffraction patterns from noncrystalline samples using the oversampling method," *Physical Review B*, vol. 67, no. 17, p. 174 104, 2003.

[200] J. Miao, T. Ishikawa, Q. Shen, and T. Earnest, "Extending x-ray crystallography to allow the imaging of noncrystalline materials, cells, and single protein complexes," *Annu. Rev. Phys. Chem.*, vol. 59, pp. 387–410, 2008.

[201] J. M. Rodenburg and H. M. Faulkner, "A phase retrieval algorithm for shifting illumination," *Applied physics letters*, vol. 85, no. 20, pp. 4795–4797, 2004.

[202] A. M. Maiden and J. M. Rodenburg, "An improved ptychographical phase retrieval algorithm for diffractive imaging," *Ultramicroscopy*, vol. 109, no. 10, pp. 1256–1262, 2009.

[203] J. R. Fienup, J. C. Marron, T. J. Schulz, and J. H. Seldin, "Hubble space telescope characterized by using phase-retrieval algorithms," *Applied Optics*, vol. 32, no. 10, pp. 1747–1767, 1993.

[204] W. L. Freedman, B. F. Madore, B. K. Gibson, L. Ferrarese, D. D. Kelson, S. Sakai, J. R. Mould, R. C. Kennicutt Jr, H. C. Ford, J. A. Graham, *et al.*, "Final results from the hubble space telescope key project to measure the hubble constant," *The Astrophysical Journal*, vol. 553, no. 1, p. 47, 2001.

[205] F. Zernike, "Phase contrast, a new method for the microscopic observation of transparent objects," *Physica*, vol. 9, no. 7, pp. 686–698, Jul. 1942, ISSN: 00318914. DOI: 10.1016/S0031-8914(42)80035-X. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S003189144280035X.

[206] M. H. Maleki and A. J. Devaney, "Phase-retrieval and intensity-only reconstruction algorithms for optical diffraction tomography," *JOSA A*, vol. 10, no. 5, pp. 1086–1092, 1993.

[207] S. Chowdhury, M. Chen, R. Eckert, D. Ren, F. Wu, N. Repina, and L. Waller, "High-resolution 3d refractive index microscopy of multiple-scattering samples from intensity images," *Optica*, vol. 6, no. 9, pp. 1211–1219, 2019.

[208] T. E. Gureyev and K. A. Nugent, "Rapid quantitative phase imaging using the transport of intensity equation," *Optics Communications*, vol. 133, no. 1-6, pp. 339–346, 1997.

[209] E. Bostan, E. Froustey, B. Rappaz, E. Shaffer, D. Sage, and M. Unser, "Phase retrieval by using transport-of-intensity equation and differential interference contrast microscopy," in *2014 IEEE International Conference on Image Processing (ICIP)*, IEEE, 2014, pp. 3939–3943.

[210] D. Leseberg, "Computer-generated three-dimensional image holograms," *Applied optics*, vol. 31, no. 2, pp. 223–229, 1992.

[211] J. Zhang, N. Pégard, J. Zhong, H. Adesnik, and L. Waller, "3d computer-generated holography by non-convex optimization," *Optica*, vol. 4, no. 10, pp. 1306–1313, 2017.

[212] M. H. Eybposh, N. W. Caira, M. Atisa, P. Chakravarthula, and N. C. Pégard, "Deepcgh: 3d computer-generated holography using deep learning," *Optics Express*, vol. 28, no. 18, pp. 26 636–26 650, 2020.

[213] E. J. Candes, J. K. Romberg, and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, vol. 59, no. 8, pp. 1207–1223, 2006.

[214] G. Barbastathis, A. Ozcan, and G. Situ, "On the use of deep learning for computational imaging," *Optica*, vol. 6, no. 8, pp. 921–943, 2019.

[215] R. Gerchberg and W. Saxton, "A practical algorithm for the determination of phase from image and diffraction plane pictures," *Optik*, vol. 35, p. 237, 1972.

[216] J. R. Fienup, "Phase retrieval algorithms: A comparison," *Applied Optics*, vol. 21, no. 15, pp. 2758–2769, 1982.

[217] J. Nocedal and S. Wright, *Numerical optimization*. Springer Science & Business Media, 2006.

[218] B. T. Polyak, "The conjugate gradient method in extremal problems," *USSR Computational Mathematics and Mathematical Physics*, vol. 9, no. 4, pp. 94–112, 1969.

[219] J. Nocedal, "Updating quasi-newton matrices with limited storage," *Mathematics of computation*, vol. 35, no. 151, pp. 773–782, 1980.

[220] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[221] N. Z. Shor, "Quadratic optimization problems," *Soviet Journal of Computer and Systems Sciences*, vol. 25, pp. 1–11, 1987.

[222] A. Chai, M. Moscoso, and G. Papanicolaou, "Array imaging using intensity-only measurements," *Inverse Problems*, vol. 27, no. 1, p. 015 005, 2010.

[223] E. J. Candes, T. Strohmer, and V. Voroninski, "Phaselift: Exact and stable signal recovery from magnitude measurements via convex programming," *Communications on Pure and Applied Mathematics*, vol. 66, no. 8, pp. 1241–1274, 2013.

[224] E. J. Candes, Y. C. Eldar, T. Strohmer, and V. Voroninski, "Phase retrieval via matrix completion," *SIAM review*, vol. 57, no. 2, pp. 225–251, 2015.

[225] I. Waldspurger, A. d'Aspremont, and S. Mallat, "Phase recovery, Maxcut and complex semidefinite programming," *Mathematical Programming*, vol. 149, no. 1-2, pp. 47–81, 2015.

[226] R. Horstmeyer, R. Y. Chen, X. Ou, B. Ames, J. A. Tropp, and C. Yang, "Solving ptychography with a convex relaxation," *New journal of physics*, vol. 17, no. 5, p. 053 044, 2015.

[227] P. Schniter and S. Rangan, "Compressive phase retrieval via generalized approximate message passing," *IEEE Transactions on Signal Processing*, vol. 63, no. 4, pp. 1043–1055, 2014.

[228] J. Barbier, F. Krzakala, N. Macris, L. Miolane, and L. Zdeborová, "Optimal errors and phase transitions in high-dimensional generalized linear models," *Proceedings of the National Academy of Sciences*, vol. 116, no. 12, pp. 5451–5460, 2019.

[229] J. Pearl, *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Elsevier, 2014.

[230] D. L. Donoho, A. Maleki, and A. Montanari, "Message-passing algorithms for compressed sensing," *Proceedings of the National Academy of Sciences*, vol. 106, no. 45, pp. 18 914–18 919, 2009.

[231] S. Rangan, "Generalized approximate message passing for estimation with random linear mixing," in *2011 IEEE International Symposium on Information Theory Proceedings*, IEEE, 2011, pp. 2168–2172.

[232] G. Wang, L. Zhang, G. B. Giannakis, M. Akçakaya, and J. Chen, "Sparse phase retrieval via truncated amplitude flow," *IEEE Transactions on Signal Processing*, vol. 66, no. 2, pp. 479–491, 2017.

[233] S. Marchesini, Y.-C. Tu, and H.-t. Wu, "Alternating projection, ptychographic imaging and phase synchronization," *Applied and Computational Harmonic Analysis*, vol. 41, no. 3, pp. 815–851, 2016.

[234] E. J. Candes and T. Tao, "Near-optimal signal recovery from random projections: Universal encoding strategies?" *IEEE transactions on information theory*, vol. 52, no. 12, pp. 5406–5425, 2006.

[235] A. Maillard, B. Loureiro, F. Krzakala, and L. Zdeborová, "Phase retrieval in high dimensions: Statistical and computational phase transitions," *arXiv preprint arXiv:2006.05228*, 2020.

[236] S. Gupta, R. Gribonval, L. Daudet, and I. Dokmanić, "Don't take it lightly: Phasing optical random projections with unknown operators," in *Advances in Neural Information Processing Systems*, 2019, pp. 14 855–14 865.

[237] ——, "Fast optical system identification by numerical interferometry," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2020, pp. 1474–1478.

[238] S. Goldt, M. Advani, A. M. Saxe, F. Krzakala, and L. Zdeborová, "Dynamics of stochastic gradient descent for two-layer neural networks in the teacher-student setup," in *Advances in Neural Information Processing Systems*, 2019, pp. 6981–6991.

[239] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, *et al.*, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems*, 2019, pp. 8026–8037.

RÉSUMÉ

Le sujet de cette thèse est le développement d'algorithmes d'optimisation à grande échelle, popularisés par l'apprentissage automatique, pour diverses études autour de la propagation de la lumière dans des milieux complexes. Nous étudions ces questions à travers le prisme de la diffusion optique, un phénomène physique décrivant la propagation de la lumière dans des matériaux complexes. Ce point de vue non conventionnel jette un nouvel éclairage sur les algorithmes d'optimisation qui se présentent dans deux domaines différents : l'imagerie et l'informatique optique. Nous appliquerons les progrès les plus récents en matière d'extraction de phase et de factorisation matricielle de bas pour l'analyse de grands ensembles de données. Ces avancées algorithmiques repoussent les limites de l'imagerie non invasive par diffusion, ce qui ouvrirait de nombreuses possibilités telles que l'observation de neurones individuels au plus profond du cerveau d'animaux vivants. D'autre part, la diffusion sera l'occasion de réaliser des réseaux neuronaux optiques câblés de façon aléatoire et de très grande dimension. Un accent particulier sera mis sur la réalisation optique d'une architecture récurrente appelée calcul par réservoir, particulièrement utile pour la prédiction de séries temporelles chaotiques. Le lien étroit entre ces deux axes d'étude est symbolisé dans la matrice aléatoire introduite pour modéliser l'effet de la diffusion optique.

MOTS CLÉS

Matrices aléatoires, imagerie computationelle, diffusion optique, microscopie par fluorescence, extraction de phase, informatique optique, calcul par réservoir, noyaux récurrents

ABSTRACT

The topic of this thesis is the development of large-scale optimization algorithms, popularized by machine learning, for various applications around light propagation through complex media. We study these questions through the prism of optical scattering, a physical phenomenon describing the propagation of light in complex materials. This unconventional point of view sheds a new light on the optimization algorithms that arise for two different purposes: imaging and optical computing. We apply the most recent advances in Phase Retrieval and low-rank matrix factorization to analyze large datasets. These computational advances push the limits of non-invasive imaging through scattering, that would open up many possibilities such as the observation of individual neurons deep inside the brain of alive animals. On the other hand, scattering will be the opportunity to realize randomly-wired optical neural networks of very large dimension. A particular emphasis will be put on the optical realization of a recurrent architecture called Reservoir Computing, particularly useful for the prediction of chaotic time series. The tight link between these two lines of study is symbolized in the random matrix introduced to model the effect of scattering.

KEYWORDS

Random matrices, computational imaging, light scattering, fluorescence microscopy, phase retrieval, optical computing, reservoir computing, recurrent kernels