UCLouvain
Institute for Information
and Communication Technologies,
Electronics and Applied Mathematics

Random embeddings and efficient representations for signal processing and subspace methods

Rémi Delogne

Thesis submitted in partial fulfillment
of the requirements for the degree of
the *Ph.D. in Engineering Sciences*

August 26, 2026

ICTEAM-INMA
Ecole Polytechnique de Louvain
Université catholique de Louvain
Louvain-la-Neuve
Belgium

Thesis Committee:

Pr. Laurent **Jacques** (advisor)	UCLouvain
Pr. Pierre-Antoine **Absil** (chair)	UCLouvain
Pr. Christophe **De Vleeschouwer**	UCLouvain
Pr. Laurent **Daudet**	Université Paris Cité, France
Dr. Titouan **Vayer**	INRIA, France

Random embeddings and efficient representations for signal processing and subspace methods

by Rémi Delogne

ICTEAM
Université catholique de Louvain
Avenue Georges Lemaitre, 4
1348 Louvain-la-Neuve
Belgium

Abstract

This manuscript studies random quadratic embeddings as compact representations for signal processing and subspace learning tasks. We first investigate how information can be extracted directly from quadratic embeddings without reconstructing original signals. We then construct random features for subspace-valued data on the Grassmannian manifold. Binary, aggregated and periodic rank-one features are shown to approximate well-defined Grassmannian kernels, with uniform guarantees. Finally, we investigate efficient implementations through structured Hadamard-diagonal transformations and optical processing units. For the latter, we develop an encoder and decoder adapted to the binary-input constraints of the device and analyse the effect of physical noise on the reconstructed embeddings. Numerical experiments illustrate the approximation quality, computational gains and practical limitations of the proposed methods.

Acknowledgments

I would first like to thank the members of the jury for agreeing to serve on it. I would especially like to thank the two external members, Prof. Laurent Daudet and Dr Titouan Vayer, for joining us from abroad. I thank them for their insightful comments and suggestions for improving the manuscript, as well as for the fruitful discussions during both the private and public defences.

I also acknowledge the use of *ChatGPT* and other LLMs as auxiliary tools during the preparation of this manuscript, mainly as mathematical sparring partners, rephrasing aids, and coding assistants. The ideas and scientific content remain my own.

During the course of this adventure that is a PhD, I was lucky to be introduced to the world of research by several people with whom I had the pleasure of publishing my first works. Most notably, I would like to thank Vincent Schellekens, who completed his PhD with Laurent Jacques and whom I can really see as my PhD big bro, as well as Laurent Daudet from LightOn. I was also lucky to collaborate with Sylvain Gigan's team at the École normale supérieure in Paris, and particularly with Ziao Wang. I thank them for their support, for allowing me to use their machines, and for welcoming me to their lab in Paris, which enabled me to obtain the results presented in Chap. 5.

Next, I would like to say a big thank you to the administrative staff of the université catholique de Louvain, all those who make sure that the university keeps functioning properly, that we get paid on time, that we have a functioning IT infrastructure, that we are kept safe from cyberattacks, and that we are pestered with fake spam emails every now and then. Most importantly, I would like to thank the staff of the Euler building: Marie-Christine, Pascale, Etienne, Astrid, Dora and Marine, whose kind and caring support helped me book conference tickets on time, submit invoices correctly so that they would be accepted by the finance department, and get general life advice on all sorts of things. I always appreciated the time spent there, whether filling in boring conference forms or talking about upcoming holidays. I am also grateful to UCL for daring to employ me as a teaching assistant throughout my PhD, and to the F.R.S.-FNRS for supporting my participation in conferences and research activities.

I was lucky to spend most of my time in Louvain-la-Neuve in the Euler building. Although the building itself has few distinctive features (it is, in fact, rather representative of the bland functional architecture of LLN), it is made very special by the creatures inhabiting it. Inside, one finds a strange mix of recent graduates still figuring out what to do with the real world and insanely smart professors whose intellectual curiosity, enthusiasm and energy never seem to fade. Many of them remain interested in just about everything, are always ready for a discussion, a celebration or a new idea, and somehow still find the energy to work late into the evening or through the weekend. This blend of intellectual vigour, hard work and an impressive talent for procrastinationmaxxing through endless coffee breaks makes the place extremely well suited for great research. In hindsight, this extremely friendly environment is probably one of the reasons why the inhabitants of Euler do not completely lose their minds after spending several years haunting its corridors for most of their waking hours.

I had the pleasure of meeting many great colleagues and friends over the years, particularly Nicolas M. and Valentin de B., who had the difficult task of enduring my increasingly erratic schedule during the final months of writing. I am also very grateful to the many others with whom I had the honour of teaching, working, or simply sharing countless coffee breaks: Virginie, Sofiane, Guillaume O., Loïc V., Nizar, Simon M., Alexandre T., Philémon, Amir B., Clémence, Astrid D., Renato, Martina, Anne, Charles M., Adrien, Jonas, Guillaume VD., Antoine L., François L. and many others.

We also had the pleasure of sharing the building with colleagues from the MEMA department, who, as far as I can tell, spend their time doing mysterious things with bubbles, sand and other things beyond my mathematical understanding. Although our work rarely overlapped scientifically speaking, I have always been very fond of the interactions between our groups, and I would particularly like to thank Mattéo C., Simon Y., Antoine Q., Thomas, Insaf, Michel, Gatien, and many others for contributing to the atmosphere of the building.

Among the Eulerians I must give a particular mention to a few people. First, John and François W., with whom I spent many happy hours talking about different projects and ideas. Then, Olivier who started his thesis at the same time as me with Laurent and with whom I had many interesting discussions and not always about maths (thank God). Yassine, from the office across the corridor, was always up for moral support through lengthy sitreps on the latest happenings in and around Euler, or for a business lunch at the Café de Halles, often with Julien C. who was my original office mate for the first four years of this thesis.

This original office holds the record for the largest number of people gathered in a single office at once (video evidence shows 18 people[1]), and witnessed countless hours of discussion, often attracting passers-by who would join the conversation, start talking among themselves, and eventually allow Julien and (sometimes) me to quietly return to work while the discussion carried on around us. Although this was *not ideal* for work, we enjoyed each other's company enough to embark on an adventure that took us both on the iron ore train through the Mauritanian desert, a truly unforgettable experience.

Such a young and rowdy group of researchers of course needs taming and the occasional slap on the wrist by a truly adult authority to keep us in check and focused on our work. Fortunately, Euler is also home to a remarkable group of professors who, at least in theory, provide the necessary adult supervision to make the place a-cough cough-professional work environment. In practice, though, they are not there to police us. What I have always appreciated most is how naturally they treat new researchers as colleagues, taking our ideas seriously, involving us in discussions, and giving us the freedom to find things ourselves. That sense of trust makes the place intellectually stimulating and genuinely cool to work in. I would like to thank in particular the professors of Algebra, Discrete Maths, Data Sciences and Graph Theory[2]: Estelle Massart, Raphaël Jungers, Vincent Wertz (whom I thank for gifting me a new laptop charger), Michel Verleysen (with whom I had the great pleasure of talking very often in and around academic council[3] sessions), Julien Hendrickx (beloved head of institute with whom we had many business ideas during the Startech course that never quite came to fruition...), Olivier Pereira (whose kindness and availability are unrivalled), Vincent Blondel (with whom I also had the great pleasure of working (or sometimes against…) in and around the academic council, who very kindly allowed me to take on the graph theory course as unofficial co-titular, and gave me many other opportunities that I will always be thankful for) and, of course, Jean-Charles Delvenne Sensei (the GOAT of the nights, often working until the early hours of the morning and appearing the next day as if nothing had happened, the uncontested winner of EVERY cafeteria survey in Euler, the bright mind ready to take on any mathematical problem in any circumstances, and the kind professor who allowed me to take on the discrete maths course as unofficial co-titular). I also had great interac-

[1] Video available upon request

[2] LEPL1101, LEPL1108, LINMA2472 and LINMA1691

[3] Conseil académique, also known as CAc or CAca in French, the reader may decide the most appropriate version depending on the mood of the discussions

tions with professors from outside the engineering faculty during my time as Corsci president and representative in the academic council, most notably, Vincent Dujardin (whose lunchtime discussions I will miss dearly), René Rezsohazy (we still need to go for a pint...), Yves Horsmans (whose impeccable taste in culinary establishments deserves praises), and many others from around the university.

And on top of researchers, professors, and staff, a university would not be complete without students. We are very lucky at UCL to have such great students, somehow combining a strong partying culture with an equally impressive capacity for (very) hard work (I sometimes feel slightly guilty about the amount of work we ask of you guys). Teaching was always great fun, both in lectures and tutorials, not forgetting, of course, the joyful nights spent on Gradescope marking the same question 500 times in a row. I will genuinely miss this part of my job.

Outside of university, I must also mention the friends I bothered with my increasingly erratic moods over the past few years, always complaining about how I was never going to make it and always cancelling plans at the last minute because I suddenly had too much work to do on a Sunday evening after spending the entire weekend doing nothing. In particular, my housemates deserve special appreciation for their patience, especially those from the last two kots[4] who witnessed the harder parts of this adventure, and Fr. Simon Naveau for giving me a quiet place to stay in Louvain-la-neuve after late nights in the office. I am also grateful to the friends who have accompanied me at different points over the years and who, knowingly or not, helped me get here.

I also want to thank my family for their support. Starting with my grandparents who welcomed me so many times in the countryside, allowing me to escape the noise and distraction of the city. My siblings for their emotional support through animal imitations, and my parents for allowing me to set up camp in the living room of their house, free from distraction, during the last month of writing. Dad for your example of dedication and hard work, and mum for your unusual sense of humour and down-to-earth approach to life.

Finally, I come to my supervisor Laurent. It is difficult to know where to begin to thank him, as I could probably write another thesis about him. Above all, Laurent has been for me an example of what an academic should be. He is intellectually uncompromising while remaining humble, really passionate about research without making it about himself, and driven by a real desire to push knowledge for-

[4]Belgian name for shared flat

ward and help others do the same. He seems able to become interested in almost anything maths- or physics-related, and this curiosity does not stop when he leaves the office. Even during holidays, an interesting paper or idea discovered somewhere will inevitably find its way into my inbox. Somehow, despite this apparently unstoppable appetite for mathematics, he also finds time for a family life, for reading and thinking about things outside his own field, and for remaining very grounded.

What I also found impressive is the trust and freedom he gave me as a supervisor. Laurent never cared where or when I worked, never confused supervision with surveillance, and always showed enormous respect for the private lives of his students. Even more importantly, despite the obvious differences in experience and mathematical knowledge, he never made it feel like there was a hierarchy in a discussion. He would genuinely ask for my opinion, listen carefully to it, and give every idea a chance before dismissing it, even when it eventually turned out to be completely wrong. That ability to take younger researchers seriously and treat them as intellectual equals is something I have always deeply admired in him. At the same time, when a deadline approached, he could somehow answer an email at midnight and continue a mathematical discussion until the problem was solved. He has always put his students and collaborators before himself, generously sharing ideas and credit, encouraging curiosity rather than imposing directions, and treating young researchers as genuine colleagues. His integrity, humility and generosity have shaped not only this thesis, but also the image I will keep of what scientific research can and should be. Through him, I can even trace my academic ancestry all the way back to Erasmus, which is a rather fitting conclusion for someone who embodies so much of what I had hoped academia would be when I entered it.

Contents

List of Figures

List of Tables

1 Introduction

Over the last sixty years, computing technology has changed beyond recognition. If we only take a look back to 1969 with the Apollo 11 mission, which took humans to the moon for the first time in history, the spacecraft was guided by a computer called the *Apollo Guidance Computer*, whose memory and processing power were extremely limited by modern standards. It had a total of $4kB$ of *random-access memory* (RAM) and $72kB$ of *read-only memory* (ROM) [90]. Yet in spite of these seemingly drastic constraints on memory by today's standards, it played a vital role in taking the astronauts to the moon and safely back to Earth, making Apollo one of the most ambitious technological achievements of the twentieth century.

Several decades later, and especially over the last decade, we have seen a massive growth in the amount of data produced, stored and processed worldwide. The estimated amount of stored information increased from 2.6 exabytes (10^{18} bytes) in 1986 to 295 exabytes in 2007 [78]. More than ten years later, the same trend has not slowed down. The amount of data stored globally is now of the order of zettabytes (10^{21}) with data being produced and stored across scientific facilities (such as CERN), cloud services (such as Amazon Web Services), social networks (Meta platforms), video platforms (YouTube), among other sources [46].

This abundance of data is essential to modern machine learning methods. This is particularly visible in the development of large language models, which rely on increasingly large training datasets [109]. However, the growth of data is not solely due to an increase in the number of samples. The samples themselves can also have increasingly high dimension, which increases the storage capacity needed to save them. This appears in many areas, including genetics, hyperspectral satellite imaging, finance, medical imaging, biology, climatology, geol-

ogy, neurology, biomedical data analysis, and other modern scientific applications [61, 64, 92, 127]. Both the number of data points and the dimension of each point contribute to this growth. As a result, datasets become expensive to store, transmit, process and analyse.

A natural response is to keep scaling hardware and memory (and thus also energy consumption). This has driven many of the recent developments in IT and pushed us to massively increase computing capacity in order to allow the training of larger and larger learning systems [138]. However, scaling resources cannot be the only answer. Data centres already represent a significant and growing share of electricity consumption, with projections pointing to a sharp increase in the coming years [83, 139]. Hardware supply is also constrained by costly and geographically concentrated semiconductor production chains [141, 87]. These constraints make efficiency a scientific and strategic requirement, not just an obscure academic side quest. This has motivated calls for more sustainable artificial intelligence and for computationally efficient hardware and algorithms [160, 142]. Another approach is therefore to ask whether data must always be stored and processed in its raw form, and whether a different form may be more or less efficient. This leads to the idea of changing data representation, namely the form in which an object is described before it is stored, transmitted, compared, or processed. A signal can be described by its original samples, by Fourier coefficients, by wavelet coefficients, by coefficients in a learned dictionary, by random projections, by binary values, or in other forms. A collection of related images or signals can also sometimes be described through a low-dimensional subspace.

A representation can be more economical than another when it reduces one of the resources required by a task. This may mean fewer entries for vectors, fewer bits per entry, lower transmission cost, lower memory requirements, or faster comparisons between many objects. The challenge is then to change the representation while preserving the information needed for an intended task. This is the general problem we study in this manuscript.

1.1 The need for compact data representation

Compactness through sparsity. Replacing data with a more compact representation is not a new idea. A classical way to do this is to describe an object in a system of coordinates where most coefficients are negligible or equal to zero. For instance, many signals are well described by a small number of Fourier coefficients, while many images have compact representations in wavelet bases or in learned dictionar-

ies [48, 131]. This is the principle behind sparse representations and sparse coding. The original object may have ambient dimension n, but its useful information is then concentrated in only k significant coefficients, with k much smaller than n. In such a case, the relevant size of the representation is guided less by the ambient dimension than by the *intrinsic complexity* of the object, measured here by its sparsity level.

Compressed sensing gives a related example, but at the acquisition level. Classical sparse representations start from an already observed signal, then transform it into Fourier, wavelet, or dictionary coefficients. Compressed sensing instead bypasses this two-step route when the signal is known to have a sparse representation. Instead of first acquiring the full signal, we directly acquire a smaller number of linear projections that still contain enough information for reconstruction [60, 32]. For a k-sparse signal in dimension n, random constructions will use a number of dimensions of order $m = \mathcal{O}(k \log(n/k))$, up to constant factors. This again shows that the useful size of a representation can be governed by the intrinsic complexity of the signal rather than by its ambient dimension.

Compactness through dimension reduction. The previous examples exploit the fact that individual objects may have sparse descriptions. Another way to build compact representations is to map data points to a lower-dimensional space while preserving quantities that are useful at the level of a dataset. A classical example is principal component analysis (PCA), which projects data onto directions of largest empirical variance in order to obtain a lower-dimensional representation while retaining as much variability as possible [69]. This construction is data-dependent, since the projection directions are learned from the dataset itself.

A different route is to use random projections. The Johnson-Lindenstrauss lemma shows that a finite set of N points can be projected into a much lower-dimensional Euclidean space while approximately preserving all pairwise distances [91]. Under suitable conditions, for a target distortion $0 < \delta < 1$, the projected dimension scales like $\mathcal{O}(\delta^{-2} \log N)$, independently of the ambient dimension. In contrast with PCA, such random projections are data-agnostic. The projection is drawn independently of the dataset, and probabilistic guarantees ensure that distances or related geometric quantities are preserved up to a controlled distortion. Many constructions can be used for this purpose, including Gaussian matrices and more structured or sparse random matrices [1, 15].

Compactness through learned representations. Compact representations can also be learned from data. This approach, called *representation learning*, aims to construct features that make a later task easier to perform [19]. A classical example is neural networks. Their successive layers apply learned transformations (both linear and non-linear), to create representations suited to a particular training objective [101]. Such representations need not be lower-dimensional than the original data. Their compactness can instead come from keeping information relevant to a task while discarding less important parts.

Although learned representations may adapt more closely to a particular dataset or task, this manuscript does not go in that direction. We instead focus on data-agnostic representations, as explained above, whose behaviour can be described through explicit probabilistic guarantees.

Compactness for pairwise comparisons. The need for compact representations also appears when the task requires comparing many data points. In many learning problems, the relevant information is contained not only in each point separately, but also in the similarities or distances between pairs of points. Kernel methods replace direct manipulation of the original data by pairwise similarity (or kernel) values, allowing non-linear learning rules to be expressed through scalar products in an implicit feature space [136]. However, for a dataset with N samples, exact kernel methods will therefore require the construction and storage of an $N \times N$ Gram matrix. This creates a memory cost of order $\mathcal{O}(N^2)$, as well as a computational cost tied to the manipulation of that matrix of up to $\mathcal{O}(N^3)$ operations [2]. These costs can become prohibitive when N is large.

Random features address this difficulty by replacing exact kernel evaluations with scalar products between explicit random representations [125]. Each data point is mapped to a finite-dimensional feature vector, and the scalar product between two such vectors approximates the desired kernel value. Instead of storing all pairwise similarities, we store one feature vector per sample and then apply linear methods in the transformed space. The compactness is therefore not only tied to the dimension of a single data point, but also to the cost of comparing a large number of points.

Compactness through finite precision. Compactness can also be obtained without changing the number of dimensions. Instead, we can reduce the precision used to store or transmit each coordinate. Quantisation replaces real-valued entries by values in a *finite* alphabet, while

binarisation keeps only one bit per entry. This is a classical idea in signal processing [72], and it also appears in compressed sensing problems [85, 86]. Such representations can be much cheaper to store and manipulate, especially when the number of samples is large. The price is that the representation becomes coarser. The mathematical question is then to determine which quantities remain accessible with good precision after quantisation, such as distances or similarities.

Compactness through subspace models. A compact representation can also describe a collection of related data points rather than single objects. For example, several images of the same object, several frames of a video, or several signals from the same class may approximately come from a single low-dimensional subspace [18]. Instead of treating each sample independently, we can represent the collection with a low-dimensional subspace spanned by the dominant directions of variability. This is useful when the task is not to compare individual vectors, but to compare groups of observations. In this case, each group becomes an element of a space of subspaces, and the problem shifts towards defining suitable ways to represent, compare, and process such subspace-valued data.

The examples above show that compactness can take several forms. It may come from sparsity, dimension reduction, quantisation, cheaper pairwise comparisons, or structured models such as subspaces. In all these cases, we replace the original data by another description, often obtained through a map from the original space to a new representation space. We will refer to such maps as *embeddings*. In this manuscript, we are mostly interested in embeddings that preserve enough information to perform tasks directly in the transformed domain, without first reconstructing the original object. The next section introduces the non-linear, and in particular quadratic, embeddings that will play this role in the rest of this work.

1.2 Quadratic embeddings of data

Non-linear embeddings. The examples above mostly concern representations obtained by reducing dimension, reducing precision, or changing the geometry in which data is used. However, useful embeddings do not have to be linear. In many settings, the transformation applied to the data includes a non-linear operation chosen either because it is imposed by the sensing process, or because it makes a certain task easier. Quantised and one-bit embeddings are a first example, since they replace real-valued entries by coarse levels or signs while still

preserving enough information for estimation or comparison tasks [85, 86]. Random Fourier features give another example, where random linear embeddings are passed through non-linear periodic functions in order to approximate shift-invariant kernels by scalar products in an explicit feature space [125]. More generally, non-linear dimensionality reduction methods also show that changing representation may aim at preserving a geometry that might not be well captured by linear embeddings [102].

Quadratic embeddings and lifting. Among non-linear embeddings, quadratic embeddings play a special role in this manuscript. Given a vector $\boldsymbol{x} \in \mathbb{R}^n$ and random vectors $\boldsymbol{a}_i \in \mathbb{R}^n$, a basic quadratic embedding takes the form

$$\boldsymbol{x} \mapsto \left(\langle \boldsymbol{a}_i, \boldsymbol{x} \rangle^2\right)_{i=1}^m.$$

This map is non-linear in $\boldsymbol{x}$, but it becomes linear after *lifting*[1] the signal to the rank-one matrix $\boldsymbol{X} = \boldsymbol{x}\boldsymbol{x}^\top$. Indeed, each entry can be written as

$$\langle \boldsymbol{a}_i, \boldsymbol{x} \rangle^2 = \boldsymbol{a}_i^\top \boldsymbol{x}\boldsymbol{x}^\top \boldsymbol{a}_i = \langle \boldsymbol{a}_i \boldsymbol{a}_i^\top, \boldsymbol{x}\boldsymbol{x}^\top \rangle_{\mathrm{F}}.$$

Thus, a quadratic embedding of vectors can be interpreted as a linear embedding of lifted rank-one matrices. This interpretation is useful because the lifted representation naturally contains second-order information. For instance, if $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$, then

$$\langle \boldsymbol{x}\boldsymbol{x}^\top, \boldsymbol{y}\boldsymbol{y}^\top \rangle_{\mathrm{F}} = \langle \boldsymbol{x}, \boldsymbol{y} \rangle^2.$$

Quadratic embeddings are therefore naturally connected to squared similarities, quadratic kernels, and more generally to tasks where second-order relations between signals are the relevant quantities.

Quadratic embeddings appear in several settings. In phase retrieval, the measurements are intensities rather than linear observations, so the available information naturally takes the form of squared scalar products [13, 80, 36]. A related setting is ptychography, where diffraction intensities are measured for different illuminated regions of an object, also leading naturally to quadratic measurements [116, 110, 111]. Quadratic measurements also appear in radio-interferometric imaging, where the measurements are correlations between signals received by pairs of antennas and can be expressed as linear measurements of a lifted covariance matrix [100, 97, 98]. In covariance estimation, the object of interest is a covariance matrix, which describes

[1] As we will explain later, this procedure amounts to a deterministic embedding of $\boldsymbol{x}$ into the space of symmetric rank-one matrices.

products and correlations between coordinates [41]. Matrix recovery from rank-one projections[2] (ROP) follows a related idea, since embeddings of the form $\langle \boldsymbol{a}_i \boldsymbol{a}_i^\top, \boldsymbol{X} \rangle_{\mathrm{F}}$ give linear information about a matrix $\boldsymbol{X}$ through rank-one matrices [29]. Finally, optical sensing provides a physical motivation for such embeddings, since optical processing units can naturally produce large numbers of random quadratic features through light scattering [133]. These examples show that quadratic embeddings arise naturally when intensity or correlation measurements are observed, when matrices are the objects of interest, or when the hardware directly produces quadratic features.

Quadratic embeddings and subspaces. The same quadratic structure also appears when working with subspace-valued data. As mentioned above, a collection of related samples can sometimes be represented by low-dimensional subspaces. If such a subspace is represented by an orthonormal basis $\boldsymbol{U} \in \mathbb{R}^{n \times k}$, then its associated projector is

$$\boldsymbol{P} = \boldsymbol{U}\boldsymbol{U}^\top.$$

This projector is invariant to the choice of orthonormal basis and therefore gives a natural representation of the subspace itself. It is also a matrix-valued representation built from products of coordinates of $\boldsymbol{U}$. Rank-one projections of this projector then take the form

$$\langle \boldsymbol{a}\boldsymbol{a}^\top, \boldsymbol{P} \rangle_{\mathrm{F}} = \boldsymbol{a}^\top \boldsymbol{U}\boldsymbol{U}^\top \boldsymbol{a}.$$

Thus, random rank-one projections are also natural for subspaces. They provide finite-dimensional representations of them, and we will see later that their scalar products can be used to approximate kernels between subspaces [75].

Role of quadratic embeddings in this manuscript. The results presented in this manuscript use quadratic embeddings in two complementary ways. First, we study whether signal processing tasks can be performed directly from random quadratic embeddings, without reconstructing the original signal. In this setting, the question is whether quantities needed for detection or classification can still be approximated directly in the embedding domain. Second, we use quadratic embeddings to build random features for subspace-valued data. Their scalar products are then used to approximate kernels with the goal of

[2] Since quadratic embeddings are equivalent to linear projections of the lifted matrix $\boldsymbol{X}$ onto the lifted matrix $\boldsymbol{A}_i := \boldsymbol{a}_i \boldsymbol{a}_i^\top$ which has rank one, this operation is also called *rank-one projection* or ROP.

building explicit finite-dimensional representations. In both cases, the central question is the same. We want to understand which quantities are preserved by quadratic embeddings, how accurately they are preserved, and how this can reduce storage or computational costs.

1.3 Organisation of the manuscript

We describe here the organisation of this manuscript. To make it easier for the reader to distinguish between original results and background material, new theoretical results are shown with a shaded background, as illustrated here.

The chapters are all connected, but they do not all depend on each other in the same way. Chap. 2 introduces the main definitions and tools used throughout the manuscript. Chap. 3 and Chap. 4 both rely on this background, but can then be read fairly independently. Chap. 5 builds on the previous chapters, since it studies how the constructions introduced there can be computed more efficiently. This is summarised in Fig. 1.1.

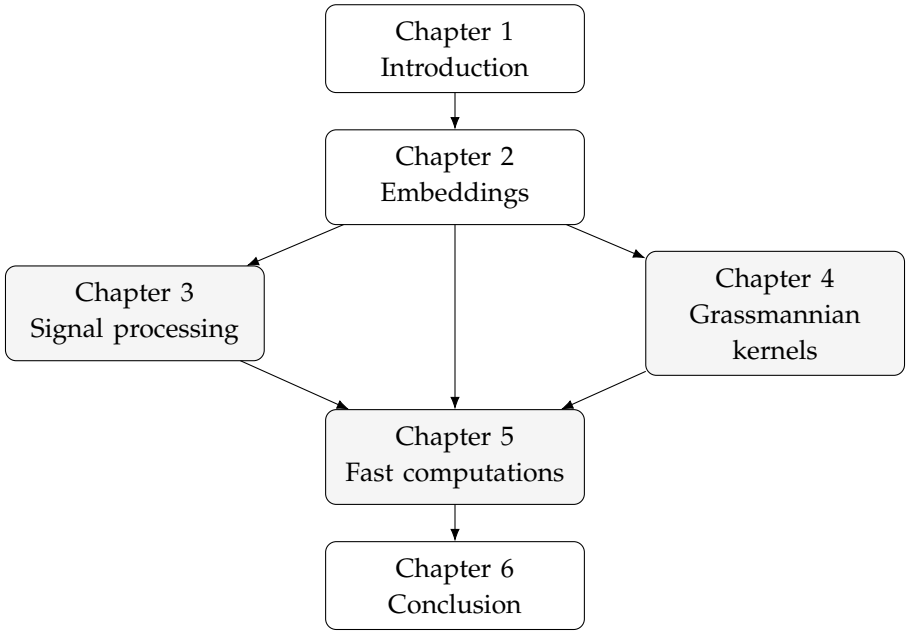


Fig. 1.1 Organisation of the manuscript and dependencies between the chapters.

Chapter 2 begins the technical part of the manuscript by turning the general idea of compact representations into more rigorous math-

ematical ideas. We first explain in more detail why high-dimensional data and large datasets create computational difficulties. We then introduce the main ways in which the representation of data can be changed while keeping useful information accessible. This leads naturally to embeddings, random transformations, subspace geometry, kernels, random features, quantisation, binarisation and concentration tools. The chapter is mostly expository, but it also sets the path followed in the rest of the manuscript. Each later chapter studies a particular way of changing the representation of data, and Chap. 2 introduces the language needed to understand them.

Chap. 3 studies signal processing from random quadratic transformations. The starting point is the debiased quadratic transformation obtained by taking differences of independent quadratic projections. We show that, after this transformation, certain quantities can be estimated directly without reconstructing the original signal. In particular, the chapter develops the sign product embedding property, which allows squared scalar products with fixed unit vectors to be estimated in the transformed domain. The same estimator is also interpreted as a finite-query approximation of the quadratic kernel. The results are illustrated with experiments.

Chap. 4 turns to subspace-valued data and Grassmannian kernels. The goal is to replace exact kernel evaluations between subspaces by scalar products between transformed representations. The chapter first discusses dense random constructions, and then introduces more compact rank-one transformations. Since raw rank-one quantities have poor theoretical concentration behaviour, the chapter studies bounded variants based on signs, aggregation, and periodic transformations. This leads to several random constructions for approximating Grassmannian kernels, together with theoretical guarantees and experiments on synthetic and real-data examples.

Chap. 5 studies how the previous constructions can be computed more efficiently. The chapter first considers structured Hadamard-diagonal transformations as cheaper alternatives to the random transformations from the previous two chapters. It then turns to optical computing and to the use of optical processing units for random quadratic transformations. Since the OPU imposes binary input constraints, the chapter introduces a quantisation and encoding procedure that allows quadratic embeddings of a quantised vector to be reconstructed from binary OPU calls.

Finally, Chap. 6 concludes the manuscript by summarising the ideas presented, discussing their limitations, and outlining perspectives for improvement and future work.

1.4 Related publications

- R. Delogne, V. Schellekens, L. Daudet, and L. Jacques. "ROP inception: Signal estimation with quadratic random sketching". In: *Proceedings of the 30th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)*. 2022, [54]
- R. Delogne, V. Schellekens, L. Daudet, and L. Jacques. "Signal processing with optical quadratic random sketches". In: *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2023, pp. 1–5, [56]
- R. Delogne, V. Schellekens, L. Daudet, and L. Jacques. "Signal processing after quadratic random sketching with optical units". In: *International Symposium on Computational Sensing (ISCS)*. 2023, [55]
- R. Delogne and L. Jacques. "Quadratic polynomial kernel approximation with asymmetric embeddings". In: *International Workshop on Deep Learning and Kernel Machines (DEEPK)*. 2024, [51]
- R. Delogne and L. Jacques. "Random Features for Grassmannian Kernels". In: *2025 IEEE Statistical Signal Processing Workshop (SSP)*. 2025, pp. 221–224, [53]
- R. Delogne and L. Jacques. "Random features for Grassmannian kernel approximation with bounded rank-one projections". In: *Under review for Transactions on Machine Learning Research (TMLR)* (2026). URL: `openreview.net/forum?id=wq18dZJ2pA`, [52]

Notations and useful formulae

Text abbreviations

w.h.p.	With high probability.
a.k.a.	Also known as.
i.e.	Id est, *a.k.a.* in other words.
e.g.	Exempli gratia, *a.k.a.* for example.
s.t.	Such that.

General mathematical symbols

$\square$	End of proof
$:=$	Is defined as
u	Scalar.
$\boldsymbol{u} = (u_i)_{i=1}^n$	Vector (bold lowercase).
$\boldsymbol{U}$	Matrix (bold uppercase).
$f(\boldsymbol{u})$	Componentwise application of a function. $(f(\boldsymbol{u}))_i := f(u_i).$

$\mathrm{dom}(f)$	domain of the function f
$\mathrm{im}(f)$	image of the function f
$\arg\max_x f(x)$	The value of x that maximises $f(x)$.
$\arg\min_x f(x)$	The value of x that minimises $f(x)$.

$\boldsymbol{e}_i$	Standard basis vector. $(\boldsymbol{e}_i)_j := \begin{cases} 1 & \text{if } j = i, \\ 0 & \text{otherwise.} \end{cases}$
$\mathbf{I}$	Identity matrix. $\mathbf{I}_n$ if dimension ambiguous
$\mathbb{1}\{x\}$	Indicator function. $\{x = 1\} := \begin{cases} 1 & \text{if } x = 1, \\ 0 & \text{otherwise.} \end{cases}$
$\lvert x \rvert$	Absolute value.
$\lfloor x \rceil$	Rounding of x to the nearest integer
$\lceil x \rceil$	Smallest integer greater or equal to x.
$\lfloor x \rfloor$	Largest integer smaller or equal to x.
i	Standard unit complex number. $\mathrm{i} := \sqrt{-1}$.
$\Re$	Real part of a number in $\mathbb{C}$. $\Re(a + b\mathrm{i}) := a$.
$\Im$	Imaginary part of a number in $\mathbb{C}$. $\Im(a + b\mathrm{i}) := b$.
$\mathcal{O}$	Order of complexity.

Sets

$\mathbb{N}$	The set of natural numbers.
$\mathbb{Z}$	The set of integers.
$\mathbb{R}$	The set of real numbers.
$\mathbb{C}$	The set of complex numbers.
$\mathcal{S}^n$	The set of n-dimensional vectors with entries in $\mathcal{S}$.
$\mathcal{S}^{m \times n}$	The set of $m \times n$ matrices with entries in $\mathcal{S}$.

Algebra

$\boldsymbol{U}^\top$	Transpose of $\boldsymbol{U}$.
$\boldsymbol{U}^*$	Conjugate transpose of $\boldsymbol{U}$.
$\mathrm{vec}(\boldsymbol{U})$	Vectorisation of the matrix by stacking its columns into a single vector.

$\mathrm{bdiag}(\boldsymbol{B}_{1,\dots,}\boldsymbol{B}_n)$	Block-diagonal matrix with entries given by the blocks $\boldsymbol{B}_1, \dots, \boldsymbol{B}_n$.
$\mathrm{tr}(\boldsymbol{U})$	Trace of $\boldsymbol{U}$.
$\det(\boldsymbol{U})$	Determinant of $\boldsymbol{U}$.
$\mathrm{pdet}(\boldsymbol{U})$	Pseudo-determinant of $\boldsymbol{U}$. If $\boldsymbol{U} \in \mathbb{R}^{n\times n}$ has rank r then $\mathrm{pdet}(\boldsymbol{U}) := \lim_{t\to 0} t^{r-n} \det(\boldsymbol{U} + t\boldsymbol{I})$.
$\mathrm{span}(\boldsymbol{U})$	Span of the columns of $\boldsymbol{U}$.
$\langle \boldsymbol{U}, \boldsymbol{V} \rangle_F$	Frobenius scalar product of $\boldsymbol{U}$ and $\boldsymbol{V}$.
	$\langle \boldsymbol{U}, \boldsymbol{V} \rangle_F = \mathrm{tr}(\boldsymbol{U}^\top \boldsymbol{V})$.
$\|\boldsymbol{U}\|_F$	Frobenius norm of $\boldsymbol{U}$. $\|\boldsymbol{U}\|_F := \sqrt{\mathrm{tr}(\boldsymbol{U}^* \boldsymbol{U})}$.
$\mathbf{0}$	Vector of zeros with dimension clear from context, otherwise $\mathbf{0}_n$.
$\langle \boldsymbol{u}, \boldsymbol{v} \rangle$	Scalar product between the vectors $\boldsymbol{u}$ and $\boldsymbol{v}$. $\langle \boldsymbol{u}, \boldsymbol{v} \rangle := \boldsymbol{u}^\top \boldsymbol{v}$.
$\|\boldsymbol{u}\|_p$	ℓ_p-norm of $\boldsymbol{u}$. $\|\boldsymbol{u}\|_p := \left(\sum_{i=1}^n |x_i|^p \right)^{1/p}$.
$\|\boldsymbol{u}\|$	$\|\boldsymbol{u}\| := \|\boldsymbol{u}\|_2$, the ℓ_2-norm of $\boldsymbol{u}$.
$\mathrm{supp}\,(\boldsymbol{u})$	Support of $\boldsymbol{u}$, *i.e.*, the set of indices where $\boldsymbol{u}$ is not zero.

Geometry

$\mathcal{G}(k, n)$	The Grassmannian manifold, containing k-dimensional subspaces of $\mathbb{R}^n$.
$\mathbb{V}(k, n)$	The Stiefel manifold, containing orthonormal matrices in $\mathbb{R}^{n\times k}$.
$\mathbb{G}(k, n)$	Projective Grassmann representation of subspaces, *i.e.*, representing the subspaces with the projection matrix associated with an orthonormal basis of that subspace.
$\mathsf{SO(n)}$	The special orthogonal group, containing matrices in $\mathbb{R}^{n\times n}$ such that $\forall \boldsymbol{U} \in \mathsf{SO(n)}, \det(\boldsymbol{U}) = 1$.
$\measuredangle(\boldsymbol{u}, \boldsymbol{v})$	Measured angle between $\boldsymbol{u}$ and $\boldsymbol{v}$.

Distances in $\mathcal{G}(k,n)$

d_{P} The projection distance. For $\boldsymbol{P}, \boldsymbol{Q} \in \mathbb{G}(k,n)$,

$$d_{\mathrm{P}}(\boldsymbol{P},\boldsymbol{Q}) = \left(\sum_{i=1}^{k} \sin^2\theta_i\right)^{1/2} = \frac{1}{\sqrt{2}}\|\boldsymbol{P}-\boldsymbol{Q}\|_F$$

d_{BC} The Binet-Cauchy distance. For $\boldsymbol{P}, \boldsymbol{Q} \in \mathbb{G}(k,n)$,

$$d_{\mathrm{BC}}(\boldsymbol{P},\boldsymbol{Q}) = \left(1-\prod_{i=1}^{k} \cos^2\theta_i\right)^{1/2} = \left(1-\mathrm{pdet}(\boldsymbol{P}\boldsymbol{Q})\right)^{1/2}$$

d_{G} The geodesic distance. For $\boldsymbol{P}, \boldsymbol{Q} \in \mathbb{G}(k,n)$,

$$d_{\mathrm{G}}(\boldsymbol{P},\boldsymbol{Q}) = \left(\sum_{i=1}^{k} \theta_i^2\right)^{1/2}.$$

Probability and Statistics

r.v.	Random variable.
$\mathbb{P}(X)$	Probability of X.
$\mathrm{Var}(X)$	Variance of X.
$\mathrm{Cov}(X,Y)$	Covariance between the r.v. X and Y.
i.i.d.	Independently and identically distributed.
$X \sim \mathcal{D}$	the r.v. X is distributed according to $\mathcal{D}$.
$X_i \underset{\text{i.i.d.}}{\sim} \mathcal{D}$	The random variables X_i are independently identically distributed according to a distribution $\mathcal{D}$.
$\mathcal{N}(0,1)$	The standard normal distribution.
$\mathcal{N}(\mathbf{0}, \mathbf{I}_n)$	The standard normal distribution in $\mathbb{R}^n$.

Embeddings

Vectors a_i, b_i are assumed random Gaussian, and $U \in \mathbb{V}(k,n)$.

This list is intended as a cheat sheet to quickly find the definition of an embedding

m Embedding dimension

ρ Oversampling ratio, $\rho := m/n$ with n the intrinsic dimension of the objects to embed.

ϕ General symbol for deterministic embeddings.

ψ General symbol for random embeddings.

ϕ^{p} Projection embedding of an orthonormal basis.

$$\phi^{\mathrm{p}}(U) = UU^\top \in \mathbb{R}^{n\times n}.$$

Ψ Dense random embedding operator. For $X \in \mathbb{R}^{n\times n}$,

$$\Psi(X) = \big(\langle \Psi_i, X\rangle\big)_{i=1}^m \in \mathbb{R}^m.$$

ψ^{s} Symmetric rank-one projection embedding for vectors. For $x \in \mathbb{R}^n$,

$$\psi^{\mathrm{s}}(x) = \big((a_i^\top x)^2\big)_{i=1}^m \in \mathbb{R}^m.$$

ψ^{-} Debiased rank-one projection embedding for vectors. For $x \in \mathbb{R}^n$,

$$\psi^{-}(x) = \big((a_i^\top x)^2 - (b_i^\top x)^2\big)_{i=1}^m \in \mathbb{R}^m.$$

ψ^{d} Dense random embedding on the projection embedding ϕ^{p}.

$$\psi^{\mathrm{d}}(U) = \Psi(\phi^{\mathrm{p}}(U)) \in \mathbb{R}^m.$$

ψ^{rop} Rank-one projection embedding.

$$\psi^{\mathrm{rop}}(\boldsymbol{U}) = \left(\boldsymbol{a}_i^\top \boldsymbol{U}\boldsymbol{U}^\top \boldsymbol{b}_i\right)_{i=1}^m \in \mathbb{R}^m.$$

$\psi^{\pm}$ Signed rank-one projection embedding.

$$\psi^{\pm}(\boldsymbol{U}) = \left(\mathrm{sign}(\boldsymbol{a}_i^\top \phi^{\mathrm{p}}(\boldsymbol{U})\boldsymbol{b}_i)\right)_{i=1}^m \in \{-1,1\}^m.$$

N_Σ Aggregation parameter used in ψ^Σ.

ψ^Σ Aggregated signed rank-one projection embedding.

$$\psi^\Sigma(\boldsymbol{U}) = \mathrm{sign}\left(\left(\textstyle\sum_{i=1}^{N_\Sigma} \boldsymbol{a}_{ij}^\top \boldsymbol{U}\boldsymbol{U}^\top \boldsymbol{b}_{ij}\right)_{j=1}^m\right) \in \{-1,1\}^m.$$

ψ_{opu} OPU version of ψ^{s}.

$$\psi_{\mathrm{opu}}(\boldsymbol{U}) = \left(\boldsymbol{a}_i^\top \boldsymbol{U}\boldsymbol{U}^\top \boldsymbol{a}_i\right)_{i=1}^m \in \mathbb{R}^m.$$

ψ_{opu}^- OPU version of ψ^-.

$$\psi_{\mathrm{opu}}^-(\boldsymbol{U}) = \left(\boldsymbol{r}_i^* \boldsymbol{U}\boldsymbol{U}^\top \boldsymbol{r}_i - \boldsymbol{s}_i^* \boldsymbol{U}\boldsymbol{U}^\top \boldsymbol{s}_i\right)_{i=1}^m \in \mathbb{R}^m.$$

ψ_{st} Structured rank-one projection embedding using fast structured random matrices.

$$\psi_{\mathrm{st}}(\boldsymbol{U}) = \left(\boldsymbol{a}_i^\top \boldsymbol{U}\boldsymbol{U}^\top \boldsymbol{b}_i\right)_{i=1}^m \in \mathbb{R}^m.$$

ψ° Periodic rank-one projection embedding for subspaces.

$$\psi^\circ(\boldsymbol{U}) = \left(\exp(\mathrm{i}\omega \boldsymbol{a}_i^\top \phi^{\mathrm{p}}(\boldsymbol{U})\boldsymbol{b}_i)\right)_{i=1}^m \in \mathbb{C}^m.$$

Kernels and approximations

This list is intended as a cheat sheet to quickly find the definition of a particular kernel

$\kappa(x, y)$ Arbitrary kernel applied to arbitrary objects x and y.

$\kappa^{\mathrm{S}}(\boldsymbol{x}, \boldsymbol{u})$ Sign product embedding kernel. For $\boldsymbol{x}, \boldsymbol{u} \in \mathbb{R}^n$,

$$\kappa^{\mathrm{S}}(\boldsymbol{x}, \boldsymbol{u}) := \mathbb{E}[\tfrac{\pi}{4m}\langle \psi^-(\boldsymbol{x}), \operatorname{sign}(\psi^-(\boldsymbol{u}))\rangle] = \langle \boldsymbol{u}, \boldsymbol{x}\rangle^2.$$

$\kappa^{\mathrm{d}}(\boldsymbol{U}, \boldsymbol{V})$ Kernel induced by the dense random embedding ψ^{d}.

$$\kappa^{\mathrm{d}}(\boldsymbol{U}, \boldsymbol{V}) := \mathbb{E}[\tfrac{1}{m}\langle \psi^{\mathrm{d}}(\boldsymbol{U}), \psi^{\mathrm{d}}(\boldsymbol{V})\rangle].$$

$\kappa^{\mathrm{rop}}(\boldsymbol{U}, \boldsymbol{V})$ Kernel induced by the rank-one projection embedding ψ^{rop}.

$$\kappa^{\mathrm{rop}}(\boldsymbol{U}, \boldsymbol{V}) := \mathbb{E}[\tfrac{1}{m}\langle \psi^{\mathrm{rop}}(\boldsymbol{U}), \psi^{\mathrm{rop}}(\boldsymbol{V})\rangle].$$

$\kappa^{\mathrm{P}}(\boldsymbol{U}, \boldsymbol{V})$ Projection kernel. For $\boldsymbol{P} = \boldsymbol{U}\boldsymbol{U}^\top$ and $\boldsymbol{Q} = \boldsymbol{V}\boldsymbol{V}^\top$,

$$\kappa^{\mathrm{P}}(\boldsymbol{U}, \boldsymbol{V}) := \|\boldsymbol{U}^\top \boldsymbol{V}\|_F^2 = k - d_{\mathrm{P}}(\boldsymbol{P}, \boldsymbol{Q})^2.$$

$\kappa^{\mathrm{BC}}(\boldsymbol{U}, \boldsymbol{V})$ Binet-Cauchy kernel. For $\boldsymbol{P} = \boldsymbol{U}\boldsymbol{U}^\top$ and $\boldsymbol{Q} = \boldsymbol{V}\boldsymbol{V}^\top$,

$$\kappa^{\mathrm{BC}}(\boldsymbol{U}, \boldsymbol{V}) := \det(\boldsymbol{U}^\top \boldsymbol{V})^2 = 1 - d_{\mathrm{BC}}(\boldsymbol{P}, \boldsymbol{Q})^2.$$

$\kappa^{\pm}(\boldsymbol{U}, \boldsymbol{V})$ Signed Grassmannian kernel.

$$\kappa^{\pm}(\boldsymbol{U}, \boldsymbol{V}) := \mathbb{E}[\tfrac{1}{m}\langle \psi^{\pm}(\boldsymbol{U}), \psi^{\pm}(\boldsymbol{V})\rangle].$$

$\kappa^{\circ}(\boldsymbol{U}, \boldsymbol{V})$ Periodic Grassmannian kernel.

$$\kappa^{\circ}(\boldsymbol{U}, \boldsymbol{V}) := \mathbb{E}[\tfrac{1}{m}\langle \psi^{\circ}(\boldsymbol{U}), \psi^{\circ}(\boldsymbol{V})\rangle].$$

$\kappa^{\Sigma}(\boldsymbol{U}, \boldsymbol{V})$ Aggregated signed Grassmannian kernel.

$$\kappa^{\Sigma}(\boldsymbol{U}, \boldsymbol{V}) := \mathbb{E}[\tfrac{1}{m}\langle \psi^{\Sigma}(\boldsymbol{U}), \psi^{\Sigma}(\boldsymbol{V})\rangle].$$

$\kappa^{\Sigma}_{\infty}(\boldsymbol{U}, \boldsymbol{V})$ Kernel associated with the aggregated signed embedding. If $\boldsymbol{U}$ and $\boldsymbol{V}$ have principal angles $\theta_1, \ldots, \theta_k$,

$$\kappa^{\Sigma}_{\infty}(\boldsymbol{U}, \boldsymbol{V}) := \tfrac{2}{\pi}\arcsin(\tfrac{1}{k}\textstyle\sum_{i=1}^{k}\cos^2\theta_i).$$

$\widehat{\kappa}(x, y)$ Approximation of an arbitrary kernel $\kappa(x, y)$ using random embeddings. For arbitrary objects x, y, and a random embedding ψ,

$$\widehat{\kappa}(x, y) = \tfrac{1}{m}\langle \psi(x), \psi(y)\rangle.$$

The approximated kernel depends on the choice of ψ.

Conventions

Throughout this manuscript, we use a small number of symbols consistently for dimensions and indexing. The exact meaning of each quantity remains clear from the context when the same symbol is used for related model parameters. Deterministic and random embeddings are denoted by ϕ and ψ, respectively, with specific variants introduced when needed. Their meaning is summarised in Tab. 1.2.

Symbol	Meaning
n	Ambient dimension of the objects under consideration.
N	Number of instances in a dataset.
m	Number of random measurements or dimension of an embedding.
k	Intrinsic model parameter. It denotes the sparsity level for sparse vectors and the subspace dimension for elements of $\mathcal{G}(k, n)$.
i, j, ℓ	Indices used for coordinates, measurements or elements of finite collections.
$\widehat{x}$	Approximation or estimate of a quantity x.
ϕ	General symbol for deterministic embeddings.
ψ	General symbol for random embeddings.

Table 1.2 General notation conventions used throughout the manuscript.

2
Representing and comparing data

The introduction motivated the use of compact representations for high-dimensional data and large datasets. We now turn this general idea into more rigorous mathematical concepts. The goal of this chapter is to introduce the objects, notations and tools used throughout the manuscript to describe how data can be transformed while preserving quantities of interest.

Datasets can be large in two different ways. The first difficulty comes from the dimension of each object. We denote by n the *ambient dimension* when the dataset is composed of vectors in $\mathbb{R}^n$. As n grows, even elementary operations such as storing a vector, computing scalar products or evaluating distances become costly. This already appears in standard image data. A greyscale image of resolution 1920×1080, after vectorisation, is represented by a vector in $\mathbb{R}^{2\,073\,600}$. A single everyday object can therefore correspond to more than two million entries when used in its vectorised form.

This issue is not limited to data represented as vectors. Later in this chapter, we will also consider data represented by subspaces. Such objects can be described through orthonormal bases or through projectors, and these representations may involve large matrices when the ambient dimension is high. The cost of storing and manipulating each object can therefore increase even before we take the size of the dataset into account.

The second difficulty comes from the number of objects that must be stored, compared or processed. We denote this number by N. This parameter becomes especially important in methods based on pairwise comparisons, such as kernel methods, where similarities between all

pairs of training samples may be needed. Kernel methods are therefore a natural place to begin, since they show how data can be compared through an implicit representation. This also prepares the transition to random features, which replace such implicit comparisons by scalar products between explicit finite-dimensional representations.

We then discuss embeddings more generally, including the linear and quadratic constructions used in later chapters. The chapter continues with the geometry of subspaces and the Grassmannian, before returning to kernels in this structured setting. Finally, we discuss quantisation, binarisation and concentration tools, which are used to describe compact representations and analyse approximation guarantees.

2.1 From kernels to random features

In this section we turn our attention to kernels. We first give a brief history of their development. We then define them formally and show how they are used in practice. Finally we show that despite their deterministic nature, random constructions are capable of being used to approximate those kernels and potentially improve their computational efficiency.

A bit of history

Kernel methods appeared well before their use in machine learning. The word *kernel* already appeared in functional analysis, integral equations, harmonic analysis, and statistics, where kernels were used to describe functions, integral operators, covariance structures, and inner products between functions [8, Intro.]. Classical results such as Mercer's theorem [114] (1909) played an important role in this development. Very roughly, Mercer's theorem gives conditions under which certain kernels can be decomposed into sums of products of functions. Such a result helped establish the idea that a kernel can encode an inner-product structure, even when the corresponding space of functions is not written down explicitly [136, Prop. 2.11].

In 1950 Aronszajn produced a paper on reproducing kernels [8] where he unified these ideas through the theory of reproducing kernel Hilbert spaces. A few years later in 1962, Parzen used this framework in statistics and signal processing contexts [120]. In that setting, using a kernel made it possible to express estimators and detectors through inner products in a function space.

The connection with modern machine learning came in 1964 in pattern recognition with the paper *Theoretical Foundations of the Potential Function Method in Pattern Recognition* by Aizerman, Braverman and Ro-

zonoer, where they used positive kernels to introduce the idea that nonlinear classification in the original space could be treated as linear classification after passing to a suitable "linearisation" space [6]. This idea was later taken up in the support vector machine framework of Boser, Guyon and Vapnik, where nonlinear classification was done through kernel evaluations rather than through an explicit construction of the feature space [25]. In this more recent setting, a kernel is therefore not only a similarity function between two objects but also a way of making linear methods available in a transformed space, while keeping that space implicit.

2.1.1 Positive-semidefinite kernels

Let us now define kernels. Let $\mathcal{X}$ be a set of objects. In standard Euclidean settings, we often compare two elements $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$ through their scalar product $\langle \boldsymbol{x}, \boldsymbol{y} \rangle$. Kernel methods allow us to generalise this idea by replacing the scalar product with a function

$$\kappa : \mathcal{X} \times \mathcal{X} \to \mathbb{R},$$

that measures similarity between two objects in $\mathcal{X}$. The set $\mathcal{X}$ may be a Euclidean space, a set of signals, a set of matrices, or other things such as a Grassmannian manifold.

A key property for a kernel to be valid is its symmetry and positive-semidefiniteness (PSD). This is not just a technical condition, it means that we will be able to interpret the kernel as an inner product that has to be symmetric and PSD after embedding the data into another space.

Definition 2.1.1 (PSD kernel)**.** *Let $\mathcal{X}$ be a non-empty set. A symmetric function $\kappa : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is called a PSD kernel if, for every $N \in \mathbb{N}$, every choice of points $x_1, \ldots, x_N \in \mathcal{X}$, and every choice of coefficients $c_1, \ldots, c_N \in \mathbb{R}$, we have*

$$\sum_{i=1}^{N} \sum_{j=1}^{N} c_i c_j \kappa(x_i, x_j) \geq 0.$$

Equivalently, for any finite collection of points $x_1, \ldots, x_N \in \mathcal{X}$, the matrix $\boldsymbol{K} \in \mathbb{R}^{N \times N}$ defined by

$$K_{ij} = \kappa(x_i, x_j),\ 1 \leq i, j \leq N$$

is symmetric PSD. This matrix is called the Gram matrix associated with the points $x_1, \ldots, x_N$ and the kernel κ.

The importance of this condition is that a PSD kernel behaves like an inner product. Recall that a Hilbert space is a vector space equipped

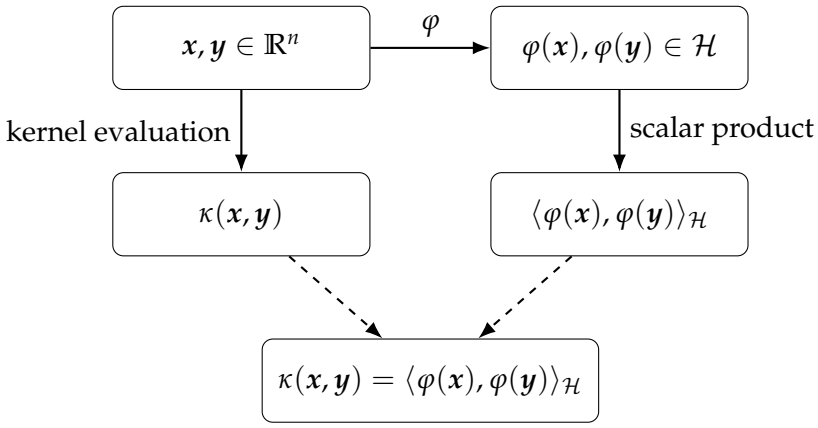


Fig. 2.1 The feature map φ sends the original vectors into a feature space. Computing the scalar product in this feature space can be done implicitly using the kernel function κ in $\mathbb{R}^n$.

with a scalar product, in which distances, angles, and orthogonal projections can be defined as in Euclidean space. The space may however be infinite-dimensional, which makes it flexible enough to represent very rich features. More precisely, there exists a Hilbert space $\mathcal{H}$ and a map

$$\varphi : \mathcal{X} \to \mathcal{H},$$

such that

$$\kappa(x, y) = \langle \varphi(x), \varphi(y) \rangle_{\mathcal{H}}, \tag{2.1}$$

for all $x, y \in \mathcal{X}$ and where $\langle x, y \rangle_{\mathcal{H}}$ is the scalar product defined in $\mathcal{H}$. The map φ is called a *feature map*. It represents each object x as a vector $\varphi(x)$ in a possibly high-dimensional or infinite-dimensional feature space, where the kernel is simply the scalar product.

This is referred to as the *kernel trick* [136]. The idea is that, whenever an algorithm only depends on scalar products between feature vectors, we do not need to construct the feature representation $\varphi(x)$ explicitly. We can instead replace each scalar product $\langle \varphi(x), \varphi(y) \rangle_{\mathcal{H}}$ by the kernel value $\kappa(x, y)$. The kernel therefore gives access to the geometry of the feature space through pairwise evaluations only.

Classical examples include the linear kernel

$$\kappa(\boldsymbol{x}, \boldsymbol{y}) = \langle \boldsymbol{x}, \boldsymbol{y} \rangle,$$

the polynomial of degree p kernel

$$\kappa(\boldsymbol{x}, \boldsymbol{y}) = \big(\langle \boldsymbol{x}, \boldsymbol{y} \rangle + c\big)^p,$$

or the Gaussian kernel

$$\kappa(\boldsymbol{x}, \boldsymbol{y}) = \exp(-\gamma \|\boldsymbol{x} - \boldsymbol{y}\|_2^2),$$

with $c \geq 0$, $p \in \mathbb{N}$, and $\gamma > 0$. In the linear case, the feature map is simply the identity. In the polynomial and Gaussian cases, the corresponding feature space is richer, and the kernel allows us to use this richer geometry without explicitly writing all the features.

We can also go in the other direction and design a feature map first, then define the corresponding kernel as the scalar product between the embedded points. For instance, if we take points in $\mathbb{R}^2$, with $\boldsymbol{x} = (x_1, x_2)$, we can define the embedding

$$\varphi(\boldsymbol{x}) = (x_1^2, \sqrt{2}x_1x_2, x_2^2).$$

Then, for $\boldsymbol{y} = (y_1, y_2)$,

$$\langle \varphi(\boldsymbol{x}), \varphi(\boldsymbol{y}) \rangle = x_1^2y_1^2 + 2x_1x_2y_1y_2 + x_2^2y_2^2.$$

This is exactly

$$(x_1y_1 + x_2y_2)^2 = \langle \boldsymbol{x}, \boldsymbol{y} \rangle^2.$$

Thus the feature map φ induces the kernel

$$\kappa(\boldsymbol{x}, \boldsymbol{y}) = \langle \boldsymbol{x}, \boldsymbol{y} \rangle^2.$$

Equivalently, this kernel can be viewed as computing a scalar product between quadratic features without explicitly treating those features as the original data representation.

This example is interesting because it shows explicitly how a nonlinear kernel can correspond to a scalar product after embedding. It also shows why explicit feature maps may become expensive. Indeed if we now work in dimension n, the full quadratic feature map has on the order of n^2 dimensions. Random features will later provide a way to approximate such kernels without constructing the full feature representation as well as accelerate the whole kernel method procedure by circumventing the need for a Gram matrix. We will explain this in detail in the next two sections.

In the next sections, we will show how these kernels can be used in learning algorithms, why they become computationally expensive for large datasets, and how random features can give explicit and finite-dimensional approximations of implicit feature maps.

2.1.2 Kernel machines and Gram matrices

Suppose that we are given a labelled dataset

$$(x_i, y_i)_{i=1}^N \subset \mathcal{X} \times \mathcal{Y}, \tag{2.2}$$

where $\mathcal{X}$ represents the input space and $\mathcal{Y}$ represents the labels assigned to every instance. If κ is a positive semi-definite (PSD) kernel with a feature map $\varphi : \mathcal{X} \to \mathcal{H}$, then learning with κ can be seen as learning a linear model in the Hilbert space $\mathcal{H}$.

Once a linear model with weight vector $\boldsymbol{w} \in \mathcal{H}$ has been learned, the prediction function $f : \mathcal{X} \to \mathcal{Y}$ for a new instance x is obtained by taking the inner product with its feature vector: $f(x) = \langle \boldsymbol{w}, \varphi(x) \rangle_{\mathcal{H}}$. The question is then how this can be evaluated without explicitly constructing the potentially high-dimensional feature vector with φ.

The *representer theorem* [134] provides the answer. It states that if $\boldsymbol{w}$ is obtained by minimising a regularised empirical risk functional, the optimal weight vector lies entirely in the span of the embedded training samples. Thus, it can be expressed as:

$$\boldsymbol{w} = \sum_{i=1}^N \alpha_i \varphi(x_i),$$

for some learned coefficients $\alpha_i \in \mathbb{R}$. Substituting this back into the prediction function yields:

$$f(x) = \langle \sum_{i=1}^N \alpha_i \varphi(x_i), \varphi(x) \rangle_{\mathcal{H}} = \sum_{i=1}^N \alpha_i \kappa(x_i, x). \tag{2.3}$$

Consequently, the model is linear in the feature space $\mathcal{H}$, even though it may be (highly) non-linear in the original input space $\mathcal{X}$. Most importantly, the learned model can be evaluated solely using kernel values between the new instance and the training samples, completely avoiding the explicit construction of the feature representation.

The advantage is that we do not need to construct the feature vectors $\varphi(x_i)$ explicitly. The algorithm can instead work only with pairwise kernel evaluations. For the training set, these evaluations are stored in the Gram matrix

$$\boldsymbol{K} \in \mathbb{R}^{N \times N}, \quad K_{ij} = \kappa(x_i, x_j).$$

This matrix stores the pairwise scalar products needed by algorithms that operate on the data after the implicit linearisation induced by the kernel trick. This is useful because linear models are easier to analyse with easily solvable optimisation problems [136, Chap. 7]. Kernels make it possible to use this linear machinery in a richer feature space without explicitly constructing that space [9].

The drawback is that the Gram matrix becomes expensive for large datasets. Storing the Gram matrix $\boldsymbol{K}$ requires $\mathcal{O}(N^2)$ in storage, and constructing it requires $\mathcal{O}(N^2)$ potentially costly kernel evaluations $\kappa(x_i, x_j)$. In addition, solving the resulting optimisation or linear algebra problem may introduce further costs of up to $\mathcal{O}(N^3)$ operations depending on the learning algorithm used [2].

Prediction can also be costly. Evaluating the prediction function from Eq. 2.3 on a new point may require comparing x with many, or in the worst case all N, training points. This situation appears naturally in support-vector machines. In the binary case, with labels $y_i \in \{-1, 1\}$, a kernel SVM learns a decision function of the form

$$f(x) = \sum_{i=1}^{N} \alpha_i y_i \kappa(x_i, x) + b, \tag{2.4}$$

where $b \in \mathbb{R}$ is a learned bias term. Only the indices with $\alpha_i \neq 0$ contribute to the sum. These training samples are called *support vectors*. Denoting their index set by $\mathcal{I}_{\text{sv}} := \{i : \alpha_i \neq 0\}$, the decision function can therefore be written as

$$f(x) = \sum_{s \in \mathcal{I}_{\text{sv}}} \alpha_s y_s \kappa(x_s, x) + b.$$

In the worst case, all training samples may be support vectors, so that $|\mathcal{I}_{\text{sv}}| = N$.

Taken together, these costs make exact kernel methods difficult to scale to datasets with a large number of instances N. This shows that problems in large dimensions and with large number of instances require simplification along two axes. The first is tied to the ambient dimension n, since representing each object, transforming it, or evaluating a single similarity may already be costly when the objects are high-dimensional. The second is tied to the dataset size N, since methods based on pairwise comparisons require a number of similarities that grows quadratically with the number of objects.

These two goals are often connected. In the kernel setting, the implicit feature map associated with κ avoids constructing a possibly very large feature representation, but it replaces this construction by the need to store and manipulate the Gram matrix. A natural objective is therefore to keep the benefit of working in a richer feature space while avoiding the explicit construction of all pairwise kernel values. In particular, we would like to replace the dependence on the full $N \times N$ Gram matrix by computations involving explicit representations of individual objects.

The next subsection explains how random features provide one way to do this.

2.1.3 Random features

Random features introduced in [125] provide a way to approximate kernel methods by replacing the implicit embedding associated with a kernel by an *explicit* finite-dimensional random embedding. Instead of working only with the kernel values $\kappa(x, y)$, we construct a random map

$$\psi : \mathcal{X} \to \mathbb{R}^m$$

such that inner products between embedded points approximate the desired kernel:

$$\widehat{\kappa}(x, y) := \tfrac{1}{m}\langle \psi(x), \psi(y) \rangle \approx \kappa(x, y).$$

The parameter m defines the number of random features and as we will see can be used to improve the approximation, at the cost of larger storage and computation.

The embedding ψ is chosen so that the empirical kernel $\widehat{\kappa}$ is unbiased, meaning that

$$\mathbb{E}\widehat{\kappa}(x, y) = \kappa(x, y)$$

for all $x, y \in \mathcal{X}$. This identity only states that the estimator is correct *on average*. To obtain an actual approximation result, we need to control the probability that the random error is large. For fixed $x, y \in \mathcal{X}$, this means proving bounds of the form

$$\mathbb{P}\{|\widehat{\kappa}(x, y) - \kappa(x, y)| > \delta\} \leq \eta(m, \delta), \tag{2.5}$$

for some probability $\eta(m, \delta)$ depending on m and δ, typically found using concentration inequality results. Equivalently, we want to determine how large m must be so that $|\widehat{\kappa}(x, y) - \kappa(x, y)| \leq \delta$ with probability at least $1 - \eta$. In stronger results, the same control is required uniformly over a set $\mathcal{S} \subset \mathcal{X}$:

$$\mathbb{P}\Big[\sup_{x,y \in \mathcal{S}} |\widehat{\kappa}(x, y) - \kappa(x, y)| > \delta\Big] \leq \eta(m, \delta), \tag{2.6}$$

for some probability $\eta(m, \delta)$ depending on m and δ. The concentration tools used to prove such results will be explained in Sec. 2.6.

More concretely, random features are built by drawing independent random parameters $\omega_1, \ldots, \omega_m$ and defining the features $z_{\omega_i}(x)$. The random embedding then takes the form

$$\psi(x) = \big(z_{\omega_1}(x), \ldots, z_{\omega_m}(x)\big),$$

so that

$$\widehat{\kappa}(x, y) = \tfrac{1}{m}\textstyle\sum_{i=1}^{m} z_{\omega_i}(x) z_{\omega_i}(y).$$

If the random features are chosen so that

$$\mathbb{E}_\omega\left[z_\omega(x)z_\omega(y)\right] = \kappa(x,y),$$

then $\widehat{\kappa}(x,y)$ is a Monte Carlo estimator of $\kappa(x,y)$.

This changes the structure of kernel methods described so far. Instead of forming and storing the full Gram matrix $\boldsymbol{K} \in \mathbb{R}^{N\times N}$, we compute the feature matrix

$$\boldsymbol{Z} = \begin{pmatrix} \psi(x_1)^\top \\ \vdots \\ \psi(x_N)^\top \end{pmatrix} \in \mathbb{R}^{N\times m}.$$

The cost of storing the data representation becomes $\mathcal{O}(Nm)$ instead of $\mathcal{O}(N^2)$. The approximate Gram matrix is formally given by

$$\widehat{\boldsymbol{K}} = \tfrac{1}{m}\boldsymbol{Z}\boldsymbol{Z}^\top,$$

but it does not need to be formed explicitly. Instead, we can train directly on the rows of $\boldsymbol{Z}$, using linear methods in $\mathbb{R}^m$. This avoids storing the full $N \times N$ Gram matrix and avoids computing all pairwise kernel evaluations. When $m \ll N$, this can substantially reduce both memory and computational cost.

The prediction stage is also more compact. A kernel predictor usually requires evaluations of $\kappa(x_i, x)$ against many (up to N) training points. With random features, the learned model can instead be represented by a vector $\boldsymbol{w} \in \mathbb{R}^m$, and prediction takes the form

$$f(x) = \langle \boldsymbol{w}, \psi(x)\rangle.$$

Thus prediction depends on the embedding dimension m, rather than directly on the number of training samples.

In the original work on random features by Rahimi and Recht [125], random feature approximations are developed for shift-invariant kernels such as the Gaussian kernel. The key tool is Bochner's theorem, which says that continuous shift-invariant PSD kernels are Fourier transforms of non-negative measures. This allows the kernel to be written as an expectation over random Fourier frequencies. Sampling a finite number of such frequencies gives an explicit feature map whose inner products approximate the kernel. Many other methods involving this idea were subsequently developed in all sorts of contexts but always with the same underlying idea of constructing explicit approximations of implicit embeddings [126, 107].

More recently, random feature approximations have also been used in modern neural architectures, for example in transformer models. Their attention mechanisms involve many pairwise comparisons between vectors, which can become computationally expensive for large inputs. Some of these comparisons rely on the softmax kernel $\kappa(\boldsymbol{q}, \boldsymbol{k}) = \exp(\langle \boldsymbol{q}, \boldsymbol{k} \rangle)$, up to a scaling of the vectors. Approximating this kernel by an explicit random feature map,

$$\kappa(\boldsymbol{q}, \boldsymbol{k}) \approx \langle \psi(\boldsymbol{q}), \psi(\boldsymbol{k}) \rangle,$$

can reduce this cost by avoiding the explicit computation of all pairwise similarities. This idea is used in the Performer architecture through the FAVOR+ construction [45]. Subsequent work has developed alternative positive and hybrid random feature constructions to improve such approximations [44, 106]. These developments suggest possible extensions of the structured or bounded random features developed later in this manuscript to attention mechanisms and other large-scale models.

2.2 Embeddings as changes of representation

The term *embedding* can have different meanings in mathematics and machine learning. Mathematically, in loose terms, an embedding is an *injective* map that places one object inside another in a way that preserves structure. Thus, different points must remain different, and the important relations between them must still be visible after the map. For example, a curve may be placed inside a plane, or a surface inside three-dimensional space, without tearing it or gluing distinct points together. Results such as Whitney's embedding theorem and Nash's embedding theorem show that very general abstract geometric spaces can be represented inside Euclidean spaces while preserving the appropriate geometric structure [156, 117]. In this classical sense, an embedding is therefore a faithful representation of one mathematical object as part of another.

In machine learning and data analysis, however, the word is used in a broader sense. An embedding is usually understood as a *change of representation*. Objects are mapped to vectors in a space where they are easier to compare. This is especially useful when the original objects are not naturally Euclidean, or when their comparison is not straightforward. For example, texts, words, graphs, or subspaces can be embedded into vector spaces so that standard geometric tools become available. Word embeddings such as Word2Vec are a typical example where words are represented by vectors whose relative positions encode semantic relations [115]. Unlike their strict mathematical counterparts,

machine learning embeddings are not necessarily injective. Depending on the context, this change of representation can have different purposes. Sometimes the goal is to represent non-Euclidean data in Euclidean spaces, sometimes it is to reduce the dimension of objects by replacing high-dimensional data with lower-dimensional vectors that preserve the most useful information, and in other cases the data points can be mapped to a richer feature space where comparisons become simpler or more meaningful. Kernel methods, introduced in the previous section, are an important example of this last situation, since a non-linear kernel can be interpreted as a scalar product after applying a possibly high-dimensional or infinite-dimensional feature map. This is also relevant for the Grassmannian kernels studied later, where random features will be used to approximate non-linear kernels beyond the projection kernel.

Random features are one example of a broader family of random embeddings. Instead of designing the representation deterministically, we design random strategies which guarantee, with high probability, that certain geometric quantities such as distances, scalar products, or kernel values are preserved up to a controlled error.

The rest of this section develops this language more precisely. We first introduce a general notation for embeddings, then discuss deterministic examples, and finally turn to random constructions.

2.2.1 Changing the representation space

We will use the word embedding in the broad sense introduced above. Formally, this means that we consider a map

$$\phi : \mathcal{X} \to \mathcal{Y},$$

where $\mathcal{X}$ is the original space containing the objects to be embedded, and $\mathcal{Y}$ is the representation space. An element $x \in \mathcal{X}$ is then replaced by $\phi(x) \in \mathcal{Y}$. A very simple example is vectorisation. If we look back to the image example, an image with resolution 1920×1080 can be represented as a matrix $\boldsymbol{X} \in \mathbb{R}^{1080\times 1920}$. The vectorisation embedding maps this matrix to a vector $\boldsymbol{x} := \mathrm{vec}(\boldsymbol{X})$ by stacking the columns of $\boldsymbol{X}$ on top of one another, with $\boldsymbol{x} \in \mathbb{R}^{2\,073\,600}$. Note that this specific example does not reduce the amount of information, but it illustrates that an embedding can simply change the form in which an object is represented.

The original space $\mathcal{X}$ depends on the nature of the data. In the simplest case, $\mathcal{X} = \mathbb{R}^n$ and the objects are vectors. In other situations, the objects may be matrices, signals, images, datasets, or structured objects

such as subspaces. The representation space $\mathcal{Y}$ may also take different forms. It may be a Euclidean space $\mathbb{R}^m$, a complex space $\mathbb{C}^m$, a matrix space, or a discrete space such as $\{-1, 1\}^m$.

The formulation of an embedding depends on the task we want to perform after changing representation. In some settings, the relevant quantity is a distance. In others, it is a scalar product or a kernel value. The choice of embedding is therefore guided by the quantity we want to preserve.

The change of space can make some quantities easier to access. For example, correlations or covariances are more naturally described through matrix-valued representations than through raw vectors alone. Similarly, a subspace is often better represented through a (unique) projector than through a particular choice of basis (of which there are many for one particular subspace). In such cases, the embedding is part of the representation on which the subsequent analysis is built.

The role of an embedding is therefore not limited to compression. When the representation space $\mathcal{Y}$ has smaller dimension than the original space $\mathcal{X}$, the embedding acts as a dimensionality reduction map. When $\mathcal{Y}$ is larger than $\mathcal{X}$, the embedding may instead be a form of feature enrichment which can make a task easier. This is common in learning problems, where data may become easier to separate after being mapped to a richer feature space. An embedding may also be task-adapted, when it is designed for a particular estimation or classification problem, or hardware-adapted, when its form is chosen so that it can be computed efficiently on a specific device.

It is also useful to distinguish explicit and implicit embeddings. In an *explicit* embedding, the vector $\phi(x)$ is actually computed and stored. Random features, introduced in the previous section, provide one example of such constructions. The linear and quadratic random embeddings introduced below will provide further examples. In contrast, kernel methods often rely on *implicit* embeddings. The embedded vectors are not necessarily computed directly, but their scalar products are evaluated through a kernel function. This was discussed in more detail in Sec. 2.1.

2.2.2 Deterministic embeddings

Before turning to random embeddings, we briefly discuss a few classical deterministic embedding methods. Here, deterministic should be understood in contrast with the random embeddings studied later. These methods may be learned from the data, and some implementations may involve random initialisation or approximate numerical

steps. However, the randomness is not part of the mathematical definition of the embedding in the same way as is the case for random ones.

A standard example is principal component analysis (PCA). Given a dataset, PCA defines a linear embedding by projecting the data onto directions of maximal variance, and is commonly used for dimensionality reduction [69]. Linear discriminant analysis (LDA) gives another example, but with a supervised objective. In this setting, the dataset is labelled, and the goal is then to build a representation in which these classes are easier to distinguish. LDA uses the labels to find directions along which the classes are well separated. The resulting embedding is therefore chosen for discrimination rather than for reconstruction or variance preservation. These examples show that an embedding may be chosen according to different criteria, such as variance preservation or class separation.

More broadly, learning methods can learn non-linear embeddings directly from data. Deep neural networks provide the most common example, with successive transformations producing features adapted to a supervised, reconstruction, or self-supervised objective [19, 101]. Although their training may involve random initialisation and stochastic optimisation, once their parameters are fixed, the resulting embedding is deterministic. They therefore belong to the present discussion rather than to the random embeddings studied later.

There exist other non-linear methods such as Isomap and Laplacian eigenmaps which aim to represent high-dimensional data in a lower-dimensional space while preserving geometric or neighbourhood information [102]. There are also visualisation-specific methods such as t-SNE or UMAP which follow a similar motivation, although their practical implementations often involve random initialisation or stochastic optimisation. They are therefore not deterministic in every computational detail, but they are also not random embeddings in the probabilistic sense considered later in this chapter.

Kernel methods, discussed in Sec. 2.1, give another important example of representation change. A kernel can be interpreted through a feature map into a space where the geometry is more suitable for a learning task. For instance, data that is not linearly separable in its original vector form may become easier to separate after being mapped to a richer feature space. In many kernel methods, this feature map is not computed explicitly. It only needs to exist, and may even take values in a high-dimensional or infinite-dimensional Hilbert space. The method can then be applied through kernel evaluations, which give scalar products between embedded objects without explicitly constructing the em-

bedded representation (see Eq. 2.1).

These examples show that embeddings may be designed for different purposes, including dimensionality reduction, visualisation, discrimination, or implicit feature enrichment. The rest of this section focuses on random embeddings, where the representation is defined through random objects and the quality of the approximation is controlled probabilistically.

Throughout this manuscript, we denote a general deterministic embedding by ϕ. Variants of this notation, such as ϕ^{p}, will be introduced later for specific deterministic embeddings.

2.2.3 Linear random embeddings

We now turn to *random embeddings*, which are the main mathematical object studied throughout this document. In contrast with the deterministic embeddings discussed above, randomness is here part of the definition of the embedding itself. We will use the notation ψ for random embeddings, with variants such as ψ^{rop}, $\psi^{\pm}$, ψ_{st} or ψ_{opu} for the specific constructions introduced later.

The most classical example of random embedding is the dense linear random embedding. Given a random matrix $\boldsymbol{\Psi} \in \mathbb{R}^{m \times n}$, we define

$$\Psi : \boldsymbol{x} \in \mathbb{R}^n \mapsto \Psi(\boldsymbol{x}) = \boldsymbol{\Psi}\boldsymbol{x} \in \mathbb{R}^m.$$

A standard choice is to take $\boldsymbol{\Psi}$ as a Gaussian matrix, with entries

$$(\boldsymbol{\Psi})_{ij} \underset{\text{i.i.d.}}{\sim} \mathcal{N}(0, 1/m)$$

for $1 \leq i \leq m$ and $1 \leq j \leq n$. With this normalisation, the embedding is *isotropic*, meaning that it preserves the squared norms in expectation. Mathematically, for every fixed $\boldsymbol{x} \in \mathbb{R}^n$,

$$\mathbb{E}\|\boldsymbol{\Psi}\boldsymbol{x}\|^2 = \|\boldsymbol{x}\|^2.$$

Equivalently, we can draw entries with distribution $\mathcal{N}(0, 1)$ and include a factor $1/\sqrt{m}$ in the definition of the embedding. The parameter m controls the dimension of the representation space. Increasing m gives a larger representation, while decreasing m gives a more compressed one. A larger m usually improves the quality of the approximation, but also increases storage and computational cost. A key part of the study of such embeddings is precisely to determine how large m has to be in order to guarantee that some quantities are preserved up to a controlled distortion, with failure probability decreasing as m increases.

A first fundamental result is the Johnson-Lindenstrauss lemma [91]. It states that a finite set of N points in $\mathbb{R}^n$ can be embedded into $\mathbb{R}^m$ while approximately preserving all pairwise distances, with m depending logarithmically on N.

Lemma 2.1 (Johnson-Lindenstrauss lemma for Gaussian embeddings)**.** *Let $\mathcal{S} \subset \mathbb{R}^n$ be a finite set with $|\mathcal{S}| = N$, and let $0 < \delta < 1$. Let $\Psi : \mathbb{R}^n \to \mathbb{R}^m$ be defined by $\Psi(\boldsymbol{x}) = \boldsymbol{\Psi}\boldsymbol{x}$, where the entries of $\boldsymbol{\Psi} \in \mathbb{R}^{m\times n}$ are independent Gaussian variables with distribution $\mathcal{N}(0, 1/m)$. Then there exist universal constants $C, c > 0$ such that, with probability at least*

$$1 - 2N^2 \exp(-c\delta^2 m),$$

we have

$$(1-\delta)\|\boldsymbol{x}-\boldsymbol{y}\|^2 \leq \|\Psi(\boldsymbol{x}) - \Psi(\boldsymbol{y})\|^2 \leq (1+\delta)\|\boldsymbol{x}-\boldsymbol{y}\|^2$$

for all $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{S}$. In particular, for a failure level $0 < \eta < 1$, it is enough to take

$$m \geq C\delta^{-2}\log(N/\eta)$$

up to a change of the universal constant C.

This result gives a first mathematical justification for dimensionality reduction through random linear maps. In the rest of this manuscript, we will use the expression *with high probability* (*w.h.p.*) for this type of guarantee, where the failure probability decreases exponentially in the number of measurements.

The previous statement applies to a fixed finite dataset. A stronger type of result is sometimes needed when the embedding must work not only for a set of observed samples, but for every possible object in a given model class. We call this a *uniform guarantee*. For example, rather than preserving distances only between the points already present in a dataset, we may want the same embedding to preserve the norm of every vector, or the distance between every pair of vectors in a given subspace.

Such guarantees require some control on the complexity of the model class. This smaller effective complexity is often called the *intrinsic dimension*, or the number of *degrees of freedom* of the model. It measures the number of parameters that are really needed to describe the relevant objects, as opposed to the ambient dimension n of the space in which they are represented.

A basic example is the set of k-sparse vectors, *i.e.*, vectors with at most k non-zero coordinates. Denoting by $\mathrm{supp}\,(\boldsymbol{x})$ the set of indices

where $\boldsymbol{x}$ is non-zero, this model is written as

$$\Sigma_k := \{\boldsymbol{x} \in \mathbb{R}^n : |\text{supp}\,(\boldsymbol{x})| \leq k\}.$$

This assumption is central in sparse representations and compressed sensing [68, 60, 33, 16] and is not limited to vectors that are sparse in their original representation. In many applications, signals are instead sparse after a change of representation. Images, for example, are often well represented by a small number of *wavelet* coefficients, which is one of the motivations behind the use of wavelet bases in signal and image processing [48, 112].

We can express this mathematically with what is called the *restricted isometry property* (RIP) [37, 34]. A linear map Ψ satisfies the RIP on Σ_k with distortion $0 < \delta < 1$ (*a.k.a.* RIP(δ, Σ_k)) if

$$(1-\delta)\|\boldsymbol{x}\|^2 \leq \|\Psi(\boldsymbol{x})\|^2 \leq (1+\delta)\|\boldsymbol{x}\|^2 \tag{2.7}$$

for all $\boldsymbol{x} \in \Sigma_k$. Equivalently, the embedding approximately preserves the Euclidean norm of every k-sparse vector. By applying this property to differences of sparse vectors, with the sparsity level replaced by $2k$, we can also obtain approximate preservation of distances within sparse sets. Random Gaussian matrices satisfy such restricted isometry properties with high probability when the number of measurements scales as

$$m = \mathcal{O}(k \log(n/k))$$

up to constants and distortion factors [15, 67].

The Gaussian construction is mathematically convenient, but it is not the only possible random linear embedding. Since a dense Gaussian matrix requires storing mn real numbers and applying it costs $\mathcal{O}(mn)$ operations, several works have proposed more compact constructions. Sparse and sign-based random projections can replace Gaussian entries by simpler discrete variables while retaining Johnson-Lindenstrauss like guarantees [1]. Other constructions combine random sign changes, fast orthogonal transforms such as the Hadamard transform, and subsampling to reduce the cost of applying the embedding [4]. These examples show that the random embedding principle is not tied to dense Gaussian matrices. The same geometric guarantees can be pursued with more structured maps that are cheaper to store and faster to apply. Related deterministic constructions are also of interest, although obtaining explicit deterministic matrices satisfying the RIP with guarantees comparable to random matrices remains a difficult problem [14].

The Johnson-Lindenstrauss and compressive sensing frameworks therefore show the same general principle, *i.e.*, that random linear embeddings can reduce dimension while preserving important geometric quantities, provided that the set of objects being embedded has controlled complexity.

The following sections move beyond the purely linear setting and introduce quadratic embeddings, which can be interpreted as linear measurements after a suitable lifted representation.

2.2.4 Quadratic embeddings

Linear random embeddings are the classical model for random dimensionality reduction. However, there are also important embedding models whose measurements are not linear with respect to the signal. A central case in this manuscript is that of *quadratic* embeddings, where the observed quantities depend on squared scalar products with measurement vectors. Such models appear in phase retrieval, where intensity measurements can be written as squared magnitudes of linear measurements [11, 12, 13, 93, 80, 36]. Related quadratic structures also appear in imaging problems such as radio interferometry [100, 97], in covariance estimation from quadratic samples [40, 41, 89], in matrix recovery from rank-one projections [29], and in optical random feature models [133].

The simplest quadratic random embedding considered here is the symmetric rank-one projection (ROP) embedding. Given independent Gaussian vectors $\boldsymbol{a}_i \underset{\text{i.i.d.}}{\sim} \mathcal{N}(\mathbf{0}, \mathbf{I}_n)$, for $i = 1, \dots, m$, we define

$$\psi^{\text{s}} : \boldsymbol{x} \in \mathbb{R}^n \mapsto \psi^{\text{s}}(\boldsymbol{x}) = \left(\langle \boldsymbol{a}_i, \boldsymbol{x} \rangle^2\right)_{i=1}^m \in \mathbb{R}^m. \tag{2.8}$$

Each entry of $\psi^{\text{s}}(\boldsymbol{x})$ measures the squared alignment of $\boldsymbol{x}$ with a random direction $\boldsymbol{a}_i$. The map is therefore nonlinear in $\boldsymbol{x}$. It also remains data-agnostic in the same sense as the linear random embeddings introduced above. We do not need to know what the data we are considering looks like or whether it is a finite dataset.

A key way to understand this construction is through *lifting*. We can indeed embed $\boldsymbol{x} \in \mathbb{R}^n$ into the set of rank-one matrices by defining the matrix $\boldsymbol{X} = \boldsymbol{x}\boldsymbol{x}^\top$. With this in mind, each quadratic measurement can be rewritten as

$$\langle \boldsymbol{a}_i, \boldsymbol{x} \rangle^2 = \boldsymbol{a}_i^\top \boldsymbol{x}\boldsymbol{x}^\top \boldsymbol{a}_i = \langle \boldsymbol{a}_i \boldsymbol{a}_i^\top, \boldsymbol{X} \rangle_{\text{F}}, \tag{2.9}$$

where $\langle \boldsymbol{A}, \boldsymbol{B} \rangle_{\text{F}} = \operatorname{tr}(\boldsymbol{A}^\top \boldsymbol{B})$ denotes the Frobenius inner product between matrices. Thus, although ψ^{s} is quadratic as a map of the vector

$\boldsymbol{x}$, it is linear as a map of the lifted matrix $\boldsymbol{X} = \boldsymbol{x}\boldsymbol{x}^\top$. In other words, quadratic random embeddings of vectors can be interpreted as linear random embeddings applied after the deterministic lifting $\boldsymbol{x} \mapsto \boldsymbol{x}\boldsymbol{x}^\top$. The measurement matrices $\boldsymbol{A}_i := \boldsymbol{a}_i\boldsymbol{a}_i^\top$ are rank-one, which explains the terminology of rank-one projections [29].

This identity also explains why this model appears in several settings. In phase retrieval, squared magnitudes of linear measurements can be interpreted as rank-one measurements of the lifted matrix $\boldsymbol{x}\boldsymbol{x}^\top$ [36]. Radio interferometry leads to related lifted formulations, where quadratic information about the observed signal is accessed through structured measurements [100, 97]. In covariance estimation, if $\boldsymbol{\Sigma}$ is a covariance matrix and $\boldsymbol{u}$ is a fixed direction, the variance of the projected data satisfies

$$\boldsymbol{u}^\top \boldsymbol{\Sigma} \boldsymbol{u} = \langle \boldsymbol{u}\boldsymbol{u}^\top, \boldsymbol{\Sigma} \rangle_{\mathrm{F}},$$

which is again a rank-one measurement [41]. Matrix recovery from rank-one projections studies the same measurement form directly on an unknown matrix [29]. Optical random feature models also produce squared random projections, leading to quadratic features of the input data [133].

Quadratic embeddings are therefore connected to the geometry of lifted rank-one matrices, and more generally to the study of low-rank matrices in high-dimensional spaces [35].

Of course, explicitly forming the lifted matrix $\boldsymbol{X} = \boldsymbol{x}\boldsymbol{x}^\top$ is expensive, since it belongs to $\mathbb{R}^{n \times n}$ thus introducing an unnecessary cost scaling like $\mathcal{O}(n^2)$. Quadratic embeddings give access to linear embeddings of this lifted object without needing to ever construct it explicitly. This explains why quadratic embeddings can be treated with tools taken from linear random embeddings, while still producing genuinely nonlinear representations of the original signal.

This point of view also shows a connection with randomised numerical linear algebra, where random embeddings are used to reduce the computational cost of large matrix operations by performing part of the computation on lower-dimensional representations [159, 77]. The methods developed in the later chapters of this manuscript can partly be interpreted from this perspective. Rather than explicitly manipulating the original high-dimensional objects, we construct compact random representations on which later computations can be performed. In particular, the rank-one measurements of projection matrices considered in Chap. 4 provide compressed representations of subspaces from which their similarities, and consequently Grassmannian kernels, can be approximated. This interpretation nevertheless differs from the

usual random numerical linear algebra setting, since the objective here is not only to accelerate standard linear algebra operations, and some of the proposed embeddings involve non-linear transformations designed to produce specific representations or kernels.

In the rest of this manuscript, we will use this construction in several ways. In Chap. 3, quadratic embeddings will be used as the domain in which signal processing tasks will be performed directly without returning to the original data, and in Chap. 4, related rank-one measurements will be applied to projection matrices representing subspaces, leading to random feature maps for Grassmannian kernels. The specific variants of the embeddings as well as guarantees needed for these applications will be introduced later when they become necessary.

2.3 Subspaces and Grassmannian geometry

The previous section introduced random embeddings of vectors, in both linear and quadratic forms. In Sec. 1.1 however we also mentioned subspaces as a way to represent data points. This means that the objects to compare are no longer individual vectors, but linear subspaces. This of course requires different geometric tools. Thus before introducing kernels or random embeddings for such objects, we will first need to explain how subspaces are represented and compared in this section.

As mentioned in the introduction, the study of subspaces is motivated by the fact that they provide an efficient way to represent data when important information is not contained in a single vector, but in the span of several observations. This appears in many settings, including signal processing, computer vision, machine learning, and subspace classification, where we compare collections of signals through the low-dimensional spaces they generate. More details on such applications will be found in Chap. 4.

A bit of history

The structure for collections of such objects is called the *Grassmannian manifold*. The idea of the Grassmannian manifold as a collection of subspaces came about during the nineteenth century. This era saw giant leaps in concepts and the study of geometry shifted from the study of individual figures (synthetic geometry) to the study of families of geometric objects and the transformations that preserve their relations. In this evolution, Julius Plücker (1801–1868) introduced algebraic coordinates to represent lines in three-dimensional space, showing that the set of all lines forms a geometric object in its own right, embedded in

a higher-dimensional projective space. This "line-geometry" was the first particular case of what would later be called a Grassmannian.

In fact, a few years earlier, Hermann Grassmann (1809–1877), who was then a secondary-school teacher in Stettin (Poland), working largely outside academic circles, had published his *Ausdehnungslehre* [71]. In it, he describes points, lines, and planes within a general framework, treating them as elements of a larger "space of subspaces" thus extending Plücker's construction from lines to subspaces of arbitrary dimension and laying out the principles that would later be used as the basis for all of linear algebra. Although his visionary work was ignored for decades, later geometers such as Cayley, Clifford and Klein recognised its importance and included it into the theory of projective geometry and invariants. For more details about this fascinating page of mathematical history the reader may refer to [118] for Grassmann's biography, to [58] for a summary of his work, or to [27] for a more general summary of the context in which these ideas came to be.

2.3.1 Subspaces and orthonormal bases

The Grassmannian manifold $\mathcal{G}(k,n)$ is the set of all k-dimensional linear subspaces of $\mathbb{R}^n$. For example, $\mathcal{G}(1,n)$ is the set of all lines through the origin in $\mathbb{R}^n$. More generally, an element of $\mathcal{G}(k,n)$ is not a vector or a matrix, but a subspace.

Recall that a linear subspace is defined as a subset of a vector space satisfying several conditions, among which the stability of its elements under linear combination [30]. One possible way of describing a finite dimensional subspace is through a finite set of linearly independent vectors of the subspace called a *basis*, such that every element of the subspace can be obtained by a linear combination of the elements of the basis. The number of elements in a basis describes the dimension of the subspace. There are infinitely many choices of different bases representing the same subspace. A subspace can therefore not be identified with a single basis.

In what follows, we will mostly use *orthonormal bases*, which are bases with the additional constraint of having elements of unit norm and orthogonal to each other. If $\mathcal{P} \in \mathcal{G}(k,n)$ is a k-dimensional subspace, we can represent it by a matrix $\boldsymbol{U} \in \mathbb{R}^{n\times k}$ whose columns form an orthonormal basis of $\mathcal{P}$. Equivalently, $\mathcal{P} = \operatorname{span}(\boldsymbol{U})$ and $\boldsymbol{U}^\top \boldsymbol{U} = \mathbf{I}_k$. The set of all such orthonormal matrices is the Stiefel manifold

$$\mathbb{V}(k,n) := \{\boldsymbol{U} \in \mathbb{R}^{n\times k} : \boldsymbol{U}^\top \boldsymbol{U} = \mathbf{I}_k\}.$$

If a subspace is obtained from a collection of vectors $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N \in \mathbb{R}^n$, we can form a data matrix $\boldsymbol{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N) \in \mathbb{R}^{n\times N}$. When

$\text{rank}(\boldsymbol{X}) \geq k$, we can form an orthonormal basis of dimension k of the subspace spanned by the columns of $\boldsymbol{X}$ using for instance a QR factorisation or a singular value decomposition (SVD).

It is important to distinguish the *subspace* from the *basis* used to represent it. A subspace $\mathcal{P}$ does not have a unique orthonormal basis. If $\boldsymbol{U} \in \mathbb{V}(k,n)$ is a basis of $\mathcal{P}$ and $\boldsymbol{R} \in \mathsf{SO}(k)$ is an orthogonal matrix, then $\boldsymbol{UR}$ is another orthonormal basis of the same subspace.

Therefore, any geometric quantity defined on $\mathcal{G}(k,n)$ must be invariant under this change of basis. If a formula is written in terms of a basis $\boldsymbol{U}$, it should give the same value when $\boldsymbol{U}$ is replaced by $\boldsymbol{UR}$.

2.3.2 Representing subspaces with projectors

Orthonormal basis are not unique for a particular subspace, we can instead use a different representation that is unique to its associated subspace. If $\boldsymbol{U} \in \mathbb{V}(k,n)$ is an orthonormal basis of a subspace $\mathcal{P} \in \mathcal{G}(k,n)$, we associate to it the *orthogonal projector*

$$\boldsymbol{P} = \boldsymbol{U}\boldsymbol{U}^\top.$$

This matrix is symmetric and idempotent, meaning that $\boldsymbol{P}^\top = \boldsymbol{P}$ and $\boldsymbol{P}^2 = \boldsymbol{P}$. It also has rank k. Conversely, any symmetric idempotent matrix of rank k is the orthogonal projector onto a unique k-dimensional subspace. This gives the projector representation of the Grassmannian,

$$\mathbb{G}(k,n) := \{\boldsymbol{P} \in \mathbb{R}^{n\times n} : \boldsymbol{P}^2 = \boldsymbol{P} = \boldsymbol{P}^\top, \text{rank}(\boldsymbol{P}) = k\}.$$

This representation is *invariant* to the choice of basis. Indeed, if $\boldsymbol{U}' = \boldsymbol{UR}$ for some orthogonal matrix $\boldsymbol{R} \in \mathsf{SO}(k)$, then

$$\boldsymbol{U}'\boldsymbol{U}'^\top = \boldsymbol{U}\boldsymbol{R}\boldsymbol{R}^\top\boldsymbol{U}^\top = \boldsymbol{U}\boldsymbol{U}^\top.$$

Thus, while the basis $\boldsymbol{U}$ is not unique, the projector $\boldsymbol{P} = \boldsymbol{U}\boldsymbol{U}^\top$ is unique. In the following sections and chapters, we will often identify a subspace $\mathcal{P} \in \mathcal{G}(k,n)$ with its projector $\boldsymbol{P} \in \mathbb{G}(k,n)$ or with an orthonormal basis $\boldsymbol{U} \in \mathbb{V}(k,n)$.

This representation will come in handy later because it turns a subspace into a matrix object that can be embedded, and compared using tools similar to those introduced in the previous section. In particular, the map

$$\phi^{\text{P}} : \boldsymbol{U} \in \mathbb{V}(k,n) \mapsto \phi^{\text{P}}(\boldsymbol{U}) = \boldsymbol{U}\boldsymbol{U}^\top \in \mathbb{G}(k,n)$$

can be viewed as a deterministic embedding of a basis into a matrix representation invariant under basis rotations. Random embeddings

of subspaces will then be built by applying random measurements to this projector representation.

Equipped with these representations, we can now define notions of distance and similarity between subspaces, which will serve as the basis for learning methods on $\mathcal{G}(k,n)$.

2.3.3 Principal angles

The only usable quantity to define the relative position of two subspaces is their *principal angles*. In the simplest case of $\mathcal{G}(1,n)$, each subspace is a line through the origin, and comparing two subspaces is equivalent to measuring the angle between the two lines. This case is illustrated in Fig. 2.2. For k-dimensional subspaces, this single angle is replaced by k *principal angles*.

These angles measure how well the two subspaces align in different directions. Small principal angles correspond to directions that are close to each other, while angles close to $\pi/2$ correspond to nearly orthogonal directions. When all principal angles are equal to zero, the two subspaces are identical and when all principal angles are equal to $\pi/2$, the two subspaces are perfectly orthogonal.

Let $\mathcal{P}, \mathcal{Q} \in \mathcal{G}(k,n)$ be two subspaces. We can find the principal angles by selecting pairs of vectors in the subspaces that are as aligned as possible. They are defined recursively as follows. First, choose unit vectors $\boldsymbol{p}_1 \in \mathcal{P}$ and $\boldsymbol{q}_1 \in \mathcal{Q}$ that maximise their scalar product, and define

$$\cos\theta_1 = \max_{\boldsymbol{p}_1 \in \mathcal{P}, \boldsymbol{q}_1 \in \mathcal{Q}} \langle \boldsymbol{p}_1, \boldsymbol{q}_1 \rangle,$$

where the maximum is taken over unit vectors $\|\boldsymbol{p}_1\| = \|\boldsymbol{q}_1\| = 1$. Then, for $i = 2, \dots, k$, one repeats the same procedure while forcing the new vectors to be orthogonal to the previously selected ones:

$$\cos\theta_i = \max_{\boldsymbol{p}_i \in \mathcal{P}, \boldsymbol{q}_i \in \mathcal{Q}} \langle \boldsymbol{p}_i, \boldsymbol{q}_i \rangle,$$

where the maximum is taken over unit vectors $\|\boldsymbol{p}_i\| = \|\boldsymbol{q}_i\| = 1$ satisfying $\langle \boldsymbol{p}_i, \boldsymbol{p}_j \rangle = 0$ and $\langle \boldsymbol{q}_i, \boldsymbol{q}_j \rangle = 0$ for every $j < i$. The resulting angles $\theta_i \in [0, \pi/2]$ are the principal angles between $\mathcal{P}$ and $\mathcal{Q}$ (see [70, chap. 12], [113] or [22]).

Alternatively, the same angles can be obtained directly from any orthonormal bases of the two subspaces. Let $\boldsymbol{U}, \boldsymbol{V} \in \mathbb{V}(k,n)$ be orthonormal bases of $\mathcal{P}$ and $\mathcal{Q}$ respectively. To find the principal angles, we compute an SVD of $\boldsymbol{U}^\top \boldsymbol{V}$:

$$\boldsymbol{U}^\top \boldsymbol{V} = \boldsymbol{W}_1 \operatorname{diag}(\sigma_1, \dots, \sigma_k) \boldsymbol{W}_2^\top.$$

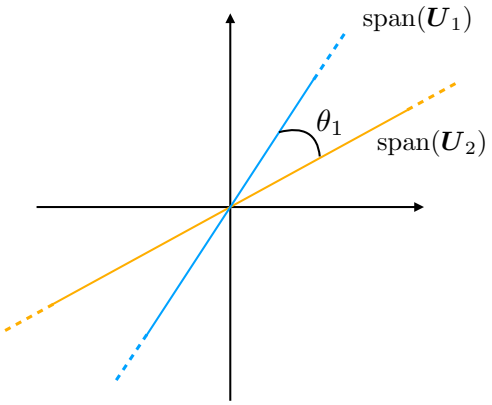


Fig. 2.2 Principal angle between two one-dimensional subspaces $\mathcal{P}_1, \mathcal{P}_2 \in \mathcal{G}(1,2)$. In this case, each subspace is a line through the origin, and the single principal angle coincides with the usual angle between the two lines.

The singular values σ_i are then the cosines of the principal angles, so that

$$\theta_i = \arccos(\sigma_i), \qquad i = 1, \ldots, k.$$

Equivalently, $\sigma_i = \cos\theta_i$ for each i. Note that if $\mathcal{P}$ and $\mathcal{Q}$ do not have the same dimension *i.e.*, $\dim(\mathcal{P}) = k \neq \dim(\mathcal{Q}) = k'$, then we can still use the recursive formula above with $\min(k, k')$ principal angles [113].

The recursive definition gives a more intuitive geometric interpretation of the principal angles, whereas the SVD formulation is usually the most convenient way to compute them. It also shows that the angles do not depend on the particular bases chosen for the subspaces. Indeed if we replace $\boldsymbol{U}$ and $\boldsymbol{V}$ by other orthonormal bases, it is equivalent to multiplying $\boldsymbol{U}^\top \boldsymbol{V}$ on the left and right by orthogonal matrices, which will not change its singular values. This shows the importance of principal angles when dealing with subspaces. We will see in the next section how these angles will allow us to define distances between subspaces.

2.3.4 Grassmannian distances

Armed with the definition of principal angles, we can put them to good use by defining *distances* on the Grassmannian of which there are many (see [124],[162, Tab. 2], or [63, Sec. 4.5]). We present here a few that will be useful in subsequent sections and chapters.

The first is the *projection distance*. Given two subspaces $\mathcal{P}$ and $\mathcal{Q}$ represented by their projectors $\boldsymbol{P} = \boldsymbol{U}\boldsymbol{U}^\top$ and $\boldsymbol{Q} = \boldsymbol{V}\boldsymbol{V}^\top$, it is defined

by

$$d_{\mathrm{P}}(\boldsymbol{P}, \boldsymbol{Q}) = \tfrac{1}{\sqrt{2}} \|\boldsymbol{P} - \boldsymbol{Q}\|_F. \tag{2.10}$$

Equivalently, in terms of the principal angles $\theta_1, \ldots, \theta_k$,

$$d_{\mathrm{P}}(\boldsymbol{P}, \boldsymbol{Q}) = \left(\textstyle\sum_{i=1}^{k} \sin^2 \theta_i \right)^{1/2}.$$

This distance is a very natural choice when working with the projector representation, since it is simply the Euclidean distance between the two projectors in Frobenius norm.

Another useful distance is the *Binet-Cauchy distance*, defined by

$$d_{\mathrm{BC}}(\boldsymbol{P}, \boldsymbol{Q}) = \left(1 - \textstyle\prod_{i=1}^{k} \cos^2 \theta_i \right)^{1/2}.$$

Since the values $\cos \theta_i$ are the singular values of $\boldsymbol{U}^\top \boldsymbol{V}$, we also have

$$\textstyle\prod_{i=1}^{k} \cos^2 \theta_i = \det(\boldsymbol{U}^\top \boldsymbol{V})^2.$$

The Binet-Cauchy distance therefore measures how much the squared volume of an orthonormal basis of one subspace shrinks when projected onto the other subspace. This squared volume is given by $\det(\boldsymbol{U}^\top \boldsymbol{V})^2 = \prod_{i=1}^{k} \cos^2 \theta_i$, and equals one when the two subspaces are identical. d_{BC} is thus equal to zero when the subspaces are identical, and increases as the subspaces become less aligned.

In projector form, this distance can also be written as

$$d_{\mathrm{BC}}(\boldsymbol{P}, \boldsymbol{Q}) = \left(1 - \operatorname{pdet}(\boldsymbol{P}\boldsymbol{Q})\right)^{1/2},$$

where the pseudo-determinant of a rank-r matrix $\boldsymbol{M}$ is defined as [66, p. 529]

$$\operatorname{pdet}(\boldsymbol{M}) = \lim_{t \to 0} t^{r-n} \det(\boldsymbol{M} + t\, \mathbf{I}_n). \tag{2.11}$$

For $\boldsymbol{P} = \boldsymbol{U}\boldsymbol{U}^\top$ and $\boldsymbol{Q} = \boldsymbol{V}\boldsymbol{V}^\top$, this gives $\operatorname{pdet}(\boldsymbol{P}\boldsymbol{Q}) = \det(\boldsymbol{U}^\top \boldsymbol{V})^2$.

Finally, the *geodesic distance* is the distance obtained by measuring the length of the shortest path on the Grassmannian manifold between two subspaces. It is given by

$$d_{\mathrm{G}}(\boldsymbol{P}, \boldsymbol{Q}) = \left(\textstyle\sum_{i=1}^{k} \theta_i^2 \right)^{1/2}.$$

It measures the length of the shortest path on $\mathcal{G}(k, n)$ connecting the two subspaces.

These distances provide different ways of comparing elements of $\mathcal{G}(k, n)$. The projection distance is directly tied to the projector representation, while the Binet-Cauchy distance is related to volume change under projection. Both of those will reappear later when we introduce kernels and random embeddings for subspaces.

2.4 Kernels and random features on the Grassmannian

In this section we demonstrate how the kernel theory we developed in Sec. 2.1 can be applied to subspaces in the Grassmannian manifold. The different geometry of course warrants slight practical adaptations but the philosophy behind it remains the same. Grassmannian kernels take subspaces of $\mathcal{G}(k,n)$ and embed them in a richer Hilbert space where inner products are possible. Access to those inner products is implicit, via formulae that work directly on representations of the subspaces, followed by learning in exactly the same way as described in Sec. 2.1, [75]. The process is illustrated in Fig. 2.3.

2.4.1 Kernels depending on principal angles

In order to be a valid Grassmannian kernel, the functions used as such must respect a few conditions. First, the kernel must be PSD to allow its use in kernel machines just as explained in Def. 2.1.1.

Second, as we target kernels defined over subspace bases, they must be independent of the orthonormal bases of $\mathbb{V}(k,n)$ used to represent the compared subspaces of $\mathcal{G}(k,n)$.

Definition 2.4.1 (Well-defined Grassmannian kernel)**.** *A well-defined* Grassmannian kernel *is a PSD function* $\kappa : \mathbb{V}(k,n) \times \mathbb{V}(k,n) \mapsto \mathbb{R}$ *that measures similarity between subspaces of* $\mathcal{G}(k,n)$ *independently of the bases chosen to represent them. Formally, given two orthonormal bases* $\boldsymbol{U}, \boldsymbol{V} \in \mathbb{V}(k,n)$*,* κ *satisfies*

$$\kappa(\boldsymbol{U},\boldsymbol{V}) = \kappa(\boldsymbol{U}\boldsymbol{R},\boldsymbol{V}\boldsymbol{R}'), \quad \forall \boldsymbol{R},\boldsymbol{R}' \in \mathsf{SO}(k),$$

where $\mathsf{SO}(k)$ *denotes the special orthogonal group in* $\mathbb{R}^k$*.*

Note that here we defined the functions for representations by orthonormal bases of elements of $\mathcal{G}(k,n)$. This could also be done via other representations such as the projectors.

Finally, the kernel must not change if both compared subspaces are similarly rotated in $\mathbb{R}^n$ [158].

Definition 2.4.2 (Rotational invariance)**.** *A* Grassmannian kernel $\kappa : \mathbb{V}(k,n) \times \mathbb{V}(k,n) \mapsto \mathbb{R}$ *is rotationally invariant if, given any two bases* $\boldsymbol{U}, \boldsymbol{V} \in \mathbb{V}(k,n)$ *of the subspaces* $\mathcal{P}, \mathcal{Q} \in \mathcal{G}(k,n)$*,*

$$\kappa(\boldsymbol{U},\boldsymbol{V}) = \kappa(\boldsymbol{R}\boldsymbol{U},\boldsymbol{R}\boldsymbol{V}),\ \forall \boldsymbol{R} \in \mathsf{SO}(n).$$

A direct consequence of the rotational invariance is the following important fact.

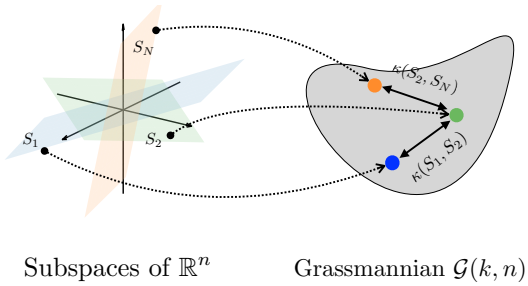


Fig. 2.3 A kernel function can be defined on the Grassmannian manifold to compute similarity scores between subspaces.

Theorem 2.1. *A rotationally invariant* Grassmannian kernel $\kappa : \mathbb{V}(k,n) \times \mathbb{V}(k,n) \mapsto \mathbb{R}$ *only depends on the principal angles of the compared subspaces,* i.e., *given two bases* $\boldsymbol{U}, \boldsymbol{V} \in \mathbb{V}(k,n)$ *of the subspaces* $\mathcal{P}, \mathcal{Q} \in \mathcal{G}(k,n)$ *and their k principal angles* $\{\theta_i\}_{i=1}^k$*, there exists a function* f *such that* $\kappa(\boldsymbol{U}, \boldsymbol{V}) = f(\theta_1, \ldots, \theta_k)$.

Proof. This theorem is a direct consequence of [158, Thm. 3] (see also [47]). □

2.4.2 Projection and Binet-Cauchy kernels

Two examples of well-defined, PSD, rotationally invariant kernels on $\mathcal{G}(k,n)$ are the projection kernel κ^{P} [74] and the Binet-Cauchy kernel κ^{BC} [157]. Given two subspaces $\mathcal{P}, \mathcal{Q} \in \mathcal{G}(k,n)$ related to projectors $\boldsymbol{P}, \boldsymbol{Q}$ and with bases $\boldsymbol{U}, \boldsymbol{V} \in \mathbb{V}(k,n)$, respectively, both kernels relate to their homonymous distances:

$$\kappa^{\mathrm{P}}(\boldsymbol{U}, \boldsymbol{V}) := \|\boldsymbol{U}^\top \boldsymbol{V}\|_F^2 = k - d_{\mathrm{P}}(\boldsymbol{P}, \boldsymbol{Q})^2, \tag{2.12}$$

$$\kappa^{\mathrm{BC}}(\boldsymbol{U}, \boldsymbol{V}) := \det(\boldsymbol{U}^\top \boldsymbol{V})^2 = 1 - d_{\mathrm{BC}}(\boldsymbol{P}, \boldsymbol{Q})^2. \tag{2.13}$$

Both kernels are linked to a classical embedding of the Grassmannian manifold allowing us to prove that they are *PSD*, *i.e.*, there exists an embedding $\phi : \mathbb{V}(k,n) \to \mathbb{R}^d$, with a possibly very large dimension d, such that the considered kernel κ can be recast as $\kappa(\boldsymbol{U}, \boldsymbol{V}) = \langle \phi(\boldsymbol{U}), \phi(\boldsymbol{V}) \rangle$. In particular, the projection kernel arises as the Frobenius inner product between orthogonal projectors, corresponding to the *projection embedding*

$$\phi^{\mathrm{P}} : \boldsymbol{U} \in \mathbb{V}(k,n) \mapsto \phi^{\mathrm{P}}(\boldsymbol{U}) = \boldsymbol{U}\boldsymbol{U}^\top \in \mathbb{G}(k,n), \tag{2.14}$$

with $d = n^2$. Similarly, the Binet-Cauchy kernel is given by the inner product of the *Plücker embedding* which represents each basis of $\mathbb{V}(k,n)$

by its $d = \binom{n}{k}$ possible $k \times k$ minors [75]. Such an embedding is never constructed explicitly in practice due to its high computational costs.

Let us also mention that beyond these two standard kernels, a larger family of PSD, rotationally invariant Grassmannian kernels can be constructed by composing classical Euclidean kernels with the projection or Plücker embeddings. In particular, [75] shows that applying radial basis function (RBF), Laplace, polynomial or binomial kernels to those embeddings yields well-defined Grassmannian kernels. A great example is the projection RBF kernel

$$\kappa_{\mathrm{RBF}}(\boldsymbol{U}, \boldsymbol{V}) = \exp\big(-\gamma \|\boldsymbol{P} - \boldsymbol{Q}\|_F^2\big), \text{ with } \gamma > 0, \tag{2.15}$$

which depends only on the Frobenius distance between projectors and is PSD on $\mathcal{G}(k, n)$. As we show in Sec. 4.5, the periodic kernel κ° introduced in this work recovers this projection RBF kernel in the small-frequency regime, thereby providing a random-feature approximation that naturally interpolates between linear and RBF-type Grassmannian similarities.

2.4.3 Random feature approximations on the Grassmannian

The random features method described earlier in Sec. 2.1.3 also applies to kernels defined on the Grassmannian. In this setting, the objects are subspaces $\mathcal{P} \in \mathcal{G}(k, n)$ represented by orthonormal bases $\boldsymbol{U} \in \mathbb{V}(k, n)$ or, equivalently, by projectors $\boldsymbol{P} = \boldsymbol{U}\boldsymbol{U}^\top$. Since the basis $\boldsymbol{U}$ is not unique, a random feature map for Grassmannian data must respect this representation issue.

A natural way to do this is to build the random features from the projector representation. Given random parameters $\omega_1, \ldots, \omega_m$, we may define features of the form

$$\psi(\boldsymbol{U}) = \big(z_{\omega_1}(\boldsymbol{U}\boldsymbol{U}^\top), \ldots, z_{\omega_m}(\boldsymbol{U}\boldsymbol{U}^\top)\big) \in \mathbb{R}^m.$$

Because the construction depends on $\boldsymbol{U}$ only through $\boldsymbol{U}\boldsymbol{U}^\top$, it is invariant to the choice of orthonormal basis. If $\boldsymbol{U}' = \boldsymbol{U}\boldsymbol{R}$ for some rotation $\boldsymbol{R} \in \mathsf{SO}(k)$, then $\boldsymbol{U}'\boldsymbol{U}'^\top = \boldsymbol{U}\boldsymbol{U}^\top$, and therefore $\psi(\boldsymbol{U}') = \psi(\boldsymbol{U})$. The resulting features are therefore well suited for $\mathcal{G}(k, n)$.

This invariance is especially natural for the random quadratic embeddings described in Sec. 2.2. Indeed, a rank-one quadratic measurement of the projector takes the form

$$z_{\boldsymbol{a}}(\boldsymbol{U}) = \boldsymbol{a}^\top \boldsymbol{U}\boldsymbol{U}^\top \boldsymbol{a}.$$

If the basis is replaced by $\boldsymbol{U}' = \boldsymbol{U}\boldsymbol{R}$, with $\boldsymbol{R} \in \mathsf{SO}(k)$, then

$$\begin{aligned} z_{\boldsymbol{a}}(\boldsymbol{U}') &= \boldsymbol{a}^\top \boldsymbol{U}\boldsymbol{R}\boldsymbol{R}^\top \boldsymbol{U}^\top \boldsymbol{a} \\ &= \boldsymbol{a}^\top \boldsymbol{U}\boldsymbol{U}^\top \boldsymbol{a} \\ &= z_{\boldsymbol{a}}(\boldsymbol{U}). \end{aligned}$$

Thus these measurements are not just compatible with the Grassmannian representation, they are measurements of the projector itself, and are therefore invariant to the choice of basis. This will be explained in more detail later.

The associated empirical kernel is

$$\widehat{\kappa}(\boldsymbol{U}, \boldsymbol{V}) = \tfrac{1}{m}\langle \psi(\boldsymbol{U}), \psi(\boldsymbol{V}) \rangle.$$

The goal is then to choose the random measurements z_{ω_i} so that this quantity approximates a well-defined Grassmannian kernel $\kappa(\boldsymbol{U}, \boldsymbol{V})$. As before, the desired guarantees can be pointwise (for a fixed pair of subspaces) or uniform (over a subset of $\mathcal{G}(k, n)$ or over the whole manifold).

This can be seen as a parallel construction of the deterministic embeddings introduced above. Instead of explicitly constructing a possibly large projector or Plücker representation, we only compute a finite number of random measurements of the subspace representation. This makes it possible to replace the full Gram matrix by explicit random feature vectors, while preserving the Grassmannian invariances needed for the resulting inner products to be useful.

The detailed constructions used later in Chap. 4 follow this principle. In particular, the Grassmannian chapter introduces random feature maps based on rank-one measurements of the projector $\boldsymbol{U}\boldsymbol{U}^\top$, together with non-linear transformations that yield bounded approximations of Grassmannian kernels [53].

2.5 Quantisation and binarisation

So far, we have discussed representation changes obtained by embedding objects into new spaces. This is only one part of the story since the cost of a representation does not only come from its dimension, but also from the numerical format used to store and manipulate it. A vector with real-valued entries may be accurate, but it can also be expensive to store, transmit, compare, or process on specialised hardware.

A second way to simplify data is therefore to reduce the precision of the representation itself. Quantisation replaces real-valued coordinates

by values from a finite alphabet, while binarisation pushes this idea to the extreme by keeping only one bit of information per entry. These operations incur loss of information, but they can still preserve enough for a given task. Again the goal is not necessarily to reconstruct the original object perfectly, but to keep the quantities needed for comparison, estimation, or classification.

This improvement will be useful in later chapters. Some of the embeddings considered in this manuscript will be explicitly binarised, while others must be adapted to quantised or binary inputs because of computational or hardware constraints. We will therefore present the basic ideas of quantisation, binarisation, and asymmetric comparisons, where two objects may be compared even though they are not represented in exactly the same way.

2.5.1 Quantised representations

A real-valued representation stores each coordinate with a certain numerical precision. In mathematical models, we write vectors as elements of $\mathbb{R}^n$, but in actual computations their entries must be represented with a finite number of bits. Quantisation formalises this step by replacing real-valued coordinates with values from a finite alphabet [72].

In the quantisation procedure we consider, each entry of a vector is quantised separately by a *scalar quantiser*. It is specified by a set of thresholds

$$t_1 < t_2 < \cdots < t_{L+1}$$

and a set of quantisation levels

$$q_1, \ldots, q_L.$$

The thresholds define the quantisation intervals

$$C_j = [t_j, t_{j+1})$$

for $j = 1, \ldots, L$. A scalar value x is then mapped to the level associated with the interval containing it:

$$Q(x) = q_j \quad \text{if} \quad x \in C_j.$$

Applied componentwise to a vector $\boldsymbol{x} \in \mathbb{R}^n$, this gives a quantised vector

$$Q(\boldsymbol{x}) = (Q(x_1), \ldots, Q(x_n)).$$

The number of levels L determines the number of bits needed to store each coordinate. If L levels are used, each quantised coordinate can be encoded using

$$B = \lceil \log_2 L \rceil$$

bits, where $\lceil x \rceil$ denotes the closest integer greater than or equal to x. Conversely, a quantiser using B bits per coordinate can represent at most 2^B levels. Standard floating-point formats typically use 32 or 64 bits per coordinate [81]. Although these formats are already finite-precision representations, they can still be expensive when either the ambient dimension n or the number of stored objects N is large. Quantising to a smaller number of levels reduces this cost since each coordinate is no longer stored as a floating-point number, but as the index of a level in a finite dictionary.

A particularly useful case is uniform scalar quantisation which is defined as follows.

Definition 2.5.1 (Uniform quantisation)**.** *Let $\delta > 0$ denote the quantisation step and let $L \in \mathbb{N}$ be an even number of levels. We define the uniform quantiser as*

$$q(t) = \delta(\lfloor t/\delta \rfloor + 1/2),$$

followed by clipping to the interval

$$[-(L/2 - 1/2)\delta, (L/2 - 1/2)\delta].$$

Here, clipping means that values above $(L/2 - 1/2)\delta$ are replaced by $(L/2 - 1/2)\delta$, while values below $-(L/2 - 1/2)\delta$ are replaced by $-(L/2 - 1/2)\delta$. The output therefore belongs to a finite alphabet of L equally spaced levels. The parameter δ controls the spacing between consecutive levels, while L controls how many levels are available before clipping occurs.

Quantisation creates a trade-off between precision and compactness of representation. The quantised vector $Q(\boldsymbol{x})$ is cheaper to store, transmit, and process, but it no longer contains all the information in $\boldsymbol{x}$. The important question is whether this loss of information affects the task we consider. In many situations, exact reconstruction of $\boldsymbol{x}$ is not required. It may be enough to preserve approximate similarities, classifications, or estimates.

In compressed sensing, for instance, the ideal model first forms linear measurements $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$, but these measurements cannot be stored or transmitted with infinite precision. They must eventually be coded with a finite number of bits. This leads to quantised CS, where the observation model includes both the measurement process and the quantisation step [26, 161, 165]. These works emphasise that quantisation is

a part of the sensing model itself. They study how the choice of quantiser, sensing matrix, and reconstruction method affects the information that can still be recovered from finite-precision measurements.

2.5.2 Binarised embeddings

Binarisation is the most extreme form of quantisation. Instead of representing each entry by one of several levels, we keep only a single bit. In the setting of measurements or embeddings, this often means replacing a real-valued vector by its signs. Given an embedding $\psi : \mathcal{X} \to \mathbb{R}^m$, its binarised version takes the form

$$\operatorname{sign}(\psi(x)) \in \{-1, 1\}^m,$$

where the sign is applied componentwise.

This operation has an immediate cost advantage. A real-valued embedding with m entries must store numerical values, while a binarised embedding stores only m bits. Binary vectors can also be compared efficiently. For example, their Hamming distance counts the number of disagreeing signs, and for vectors in $\{-1, 1\}^m$ this is equivalent to computing their scalar product. This makes binarised representations attractive when many embedded objects must be stored or compared.

The price is that binarisation discards all magnitude information. In particular, if we replace $\boldsymbol{A}\boldsymbol{x}$ by $\operatorname{sign}(\boldsymbol{A}\boldsymbol{x})$, then multiplying $\boldsymbol{x}$ by a positive scalar does not change the binary measurements. The scale of the signal is therefore lost. This is not necessarily fatal, because many tasks depend more on directions, angles, or similarities than on absolute magnitudes. The central question is then whether the binarised representation preserves the information needed for the task.

This question is central in one-bit compressed sensing. There, a signal is measured through

$$\boldsymbol{y} = \operatorname{sign}(\boldsymbol{A}\boldsymbol{x}),$$

so that each embedding records only whether a linear observation is positive or negative. In such a drastic context, it is still possible to extract stable information on signals despite the loss of information caused by binarisation [86, 122, 121, 3]. A key geometric idea is that the Hamming distance between two binary measurement vectors can reflect the angle between the original signals. Thus, even very coarse measurements can preserve useful geometry.

A related line of work is locality-sensitive hashing (LSH), where random projections are used to construct compact binary representations that preserve similarity between vectors [82, 38]. These binary

representations are obtained from the signs of random linear projections, and the probability of a sign disagreement depends directly on the angle between the vectors. Random Ferns provide another example of cheap binary representations, where several randomly chosen pairwise comparisons of local measurements, such as pixel intensities in an image patch, are grouped together and their outcomes are used to estimate probabilities for efficient classification [119]. These approaches are related to the embeddings considered in this manuscript through their use of compact binary representations, but the objective is different. The binary embeddings developed later are designed to approximate similarities or kernels with controlled accuracy, rather than for indexing or classification.

This geometric interpretation is closer to our needs than exact signal reconstruction. In later chapters, binarised embeddings are not used to recover the original object. They are used because their scalar products (or Hamming distances for binary cases) can approximate quantities of interest, such as similarities or kernel values. In that sense, binarisation is another way to change the representation while keeping enough information for the desired task.

Recent work also studies learning directly from binary measurements, both in statistical models such as one-bit linear regression [79] and in signal reconstruction from binary measurements [144]. Other applications include matrix completion when only 1-bit noisy observations are available [50]. This reinforces the message that binary representations are not just a storage trick. They can be the normal output of some measurement device, or the representation deliberately chosen because it is cheaper and still informative.

We will also use binarisation in asymmetric comparisons, where only one side of an estimator is binarised. For example, instead of comparing two objects through two identical embeddings, we may consider quantities of the form

$$\widehat{q}(x, y) = \tfrac{1}{m} \langle \operatorname{sign}(\psi(x)), \psi(y) \rangle.$$

Such expressions do not necessarily define symmetric kernels, but they can still provide useful estimators when the two objects play different roles. This idea will reappear in Chap. 3 when we estimate quantities directly from embedded or binarised measurements.

Finally we will also use binarisation to make real-valued vectors compatible with efficient computational devices whose inputs are restricted to binary representation, thus warranting a special algorithm to decompose quantised vectors into binary inputs. Increasing the number of levels L improves the accuracy of the representation, but also

increases the number of bits or elementary components needed to encode it. This is developed in Chap. 5.

2.6 From concentration to uniform guarantees

The previous sections introduced several random constructions whose purpose is to approximate deterministic quantities such as kernel functions or scalar products. In each case, the same question appears: how many random measurements or features are needed for the approximation to be reliable? We will answer this question using concentration inequalities and uniform approximation arguments.

In this section we thus introduce the probabilistic language used to answer this question *i.e.*, *concentration inequalities*. We do not aim to give a full exposition of this from head to toe. Instead, we present elements that will often return when we evaluate the quality of our estimates later. We first explain how to move from expectation to high-probability approximation for a fixed pair of objects. We then recall the role of tail behaviours of random variables, since bounded, sub-Gaussian, sub-exponential, and heavier-tailed random variables do not give the same result quality. Finally we explain how fixed pair estimates are upgraded to uniform guarantees through covering arguments and union bounds.

The resulting conditions on the number of random features should be understood as order-of-magnitude bounds rather than exact thresholds. This is enough for our purposes, because the main question is how the required number of measurements scales with the failure probability of the approximation. Throughout this manuscript, constants such as $C, c, C', \dots$ denote positive absolute constants that do not depend on the main parameters, and their value may change from line to line.

2.6.1 From expectation to concentration

The estimators considered in subsequent chapters are averages of independent random quantities. Given two objects x and y belonging to an arbitrary space, and for a deterministic target $q(x, y)$, we construct a random estimator of the form

$$\widehat{q}(x, y) = \frac{1}{m} \sum_{i=1}^{m} X_i(x, y),$$

where the variables $X_i(x, y)$ are independent copies of a random feature or random measurement depending on x and y. The first property we

will prove is its expected value

$$\mathbb{E}\widehat{q}(x,y) = q(x,y).$$

This identity simply says that the estimator will be correct *on average*. It does not say that a single draw of the random embedding gives a good approximation, or how many random features are necessary to make a good approximation.

If we now take a fixed pair of objects x, y, the next interesting statement is a high-probability bound of the form already introduced in Eq. 2.5 for kernel approximation:

$$\mathbb{P}\left\{|\widehat{q}(x,y) - q(x,y)| > \delta\right\} \leq \eta(m,\delta),$$

with $\eta(m,\delta)$ a probability depending on m and δ. Equivalently, with m trials of $X_i(x,y)$, the absolute error will be at most δ with probability at least $1-\eta(m,\delta)$. This statement is the first step to go from an identity about the expectation of our approximation to a guarantee over a larger continuous space. It says that a single random draw of the embedding is likely to produce an estimator close to the target quantity, rather than being correct only after averaging over infinitely many independent draws. This statement is also useful for seeing the role of m. Indeed by increasing the number of measurements, we will push the failure probability $\eta(m,\delta)$ down for a given target accuracy of δ.

A typical concentration bound for a fixed pair of objects (x,y) will take the form

$$\mathbb{P}\{|\widehat{q}(x,y) - q(x,y)| > \delta\} \leq C\exp(-cm\delta^2),$$

for $0 < \delta < 1$ and where $C, c > 0$ are absolute constants. Such a bound says that the failure probability decreases exponentially fast with $m\delta^2$. If we want to make the failure probability smaller than a pre-determined value $\eta > 0$, it is then enough to take m of the order of

$$\delta^{-2}\log(\eta^{-1}).$$

We repeat that here, and throughout the manuscript, statements of the form $m \geq C\delta^{-2}\log(\eta^{-1})$ give an order of magnitude for m rather than an exact threshold. The constants $C, c, C', \ldots$ do not depend on the main parameters such as m, δ, or η, and may change from line to line.

2.6.2 Tail behaviour and concentration inequalities

The strength of a concentration bound depends on the *tail behaviour* of the random variables being averaged. The tail of a random vari-

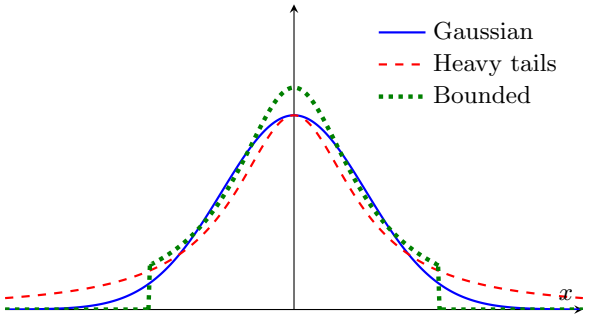


Fig. 2.4 Illustration of tail behaviour. Heavy-tailed random variables are more likely to produce large deviations, while bounded transformations cut off extreme values.

able describes how likely it is to take values far from its mean. Random variables with lighter tails lead to stronger concentration, while heavier-tailed variables give weaker concentration results.

A useful way to describe tail behaviour is through bounds on the probability of large deviations. We say that a random variable X has sub-Weibull tails of order $\alpha > 0$ if there exists a parameter $K > 0$ such that, for every $t \geq 0$,

$$\mathbb{P}\{|X| \geq t\} \leq 2\exp(-(t/K)^{\alpha}). \tag{2.16}$$

The parameter K controls the size of the variable, while α controls the speed of the tail decay. Larger values of α correspond to faster decay. This is illustrated in Fig. 2.4: heavy-tailed variables are more likely to produce large values, while bounded variables cannot exceed a fixed range.

The case $\alpha = 2$ corresponds to sub-Gaussian behaviour, where the tails decay like Gaussian tails. The case $\alpha = 1$ corresponds to sub-exponential behaviour, which allows larger deviations but still gives useful concentration results. These two regimes are especially important because they give access to standard concentration inequalities such as Hoeffding and Bernstein bounds. When $\alpha < 1$, the tails are heavier, and the resulting concentration estimates are weaker [24].

An equivalent way to describe the same behaviour is through ψ_α norms [148, Ch. 2]. For $\alpha > 0$, the ψ_α norm of a random variable X is defined as

$$\|X\|_{\psi_\alpha} := \inf\{t > 0 : \mathbb{E}\exp(|X|^{\alpha}/t^{\alpha}) \leq 2\}.$$

This norm measures the scale beyond which the tails of X decay like a sub-Weibull tail of order α. Finiteness of $\|X\|_{\psi_\alpha}$ is equivalent to the tail bound above. A random variable with finite ψ_α norm is therefore

called sub-Weibull of order α [24]. In the cases where $\alpha = 1$ the ψ_1-norm is referred to as the sub-exponential norm and when $\alpha = 2$ the ψ_2-norm is referred to as the sub-Gaussian norm.

Once the tail behaviour of the variables is known, we can choose the corresponding concentration inequality. For i.i.d. centred sub-Gaussian variables, concentration gives the basic quadratic regime

$$\mathbb{P}\{|\tfrac{1}{m}\sum_{i=1}^{m} Z_i| > \delta\} \leq 2\exp(-cm\delta^2/K^2),$$

where K depends on the ψ_2 (or sub-Gaussian) norm of the variables. This is the regime we obtain for all random variables satisfying the tail condition in Eq. 2.16 with $\alpha = 2$ which includes bounded variables.

For i.i.d. centred sub-exponential variables (*i.e.*, satisfying the tail condition in Eq. 2.16 with $\alpha = 1$) with K depending on their ψ_1 (sub-exponential) norms, we can use Bernstein's inequality which gives

$$\mathbb{P}\{|\tfrac{1}{m}\sum_{i=1}^{m} Z_i| > \delta\} \leq 2\exp(-cm\min(\delta^2/K^2, \delta/K)).$$

For small deviations, the first term dominates and the bound again behaves like $\exp(-cm\delta^2/K^2)$. For larger deviations, the second term shows the heavier sub-exponential tail.

2.6.3 Uniform approximation and covering arguments

So far the concentration inequality we derived controls the approximation error for a *fixed* pair of objects. However, since we want a result that holds for any pair of objects over a certain space, *i.e.*, a uniform approximation result, we must get to a stronger statement. If the objects of interest belong to a set $\mathcal{X}$, given a subset $\mathcal{S} \subseteq \mathcal{X}$, the desired estimate has the form introduced in Eq. 2.6,

$$\mathbb{P}\{\sup_{x,y\in\mathcal{S}} |\widehat{q}(x,y) - q(x,y)| > \delta\} \leq \eta(m,\delta),$$

with $\eta(m,\delta)$ a probability depending on m and δ. Equivalently, with probability at least $1-\eta(m,\delta)$, all pairs $x, y \in \mathcal{S}$ are approximated with error at most δ by the same draw of the random embedding.

This distinction is important whenever the embedding is used as a replacement for the original instances. The first concentration bound says that the embedding works for a fixed pair chosen in advance. A uniform bound on the other hand says that, after the random construction has been drawn, all comparisons inside the class can be made with controlled error. The set $\mathcal{S}$ may be a finite dataset, a structured signal model, or a manifold of subspaces. The required number of random

features then depends not only on the accuracy δ, but also on the size or complexity of $\mathcal{S}$.

For a finite class, going from a fixed pair to uniform control follows from the union bound. Suppose that $\mathcal{S}$ contains M elements and that, for every fixed pair $x, y \in \mathcal{S}$,

$$\mathbb{P}\{|\widehat{q}(x,y) - q(x,y)| > \delta\} \leq \eta_0.$$

Since there are at most M^2 pairs in $\mathcal{S} \times \mathcal{S}$, we obtain

$$\mathbb{P}\{\exists x, y \in \mathcal{S} : |\widehat{q}(x,y) - q(x,y)| > \delta\} \leq M^2 \eta_0.$$

If

$$\eta_0 \leq C \exp(-cm\delta^2),$$

then the uniform failure probability is bounded by

$$CM^2 \exp(-cm\delta^2).$$

To make this smaller than η, it is enough to take m of the order of

$$\delta^{-2}(\log M + \log(\eta^{-1})).$$

For infinite classes, we first reduce the problem to a finite one. Let d be a metric on $\mathcal{S}$. A finite set $\mathcal{N}_\epsilon \subset \mathcal{S}$ is called an ϵ-net if, for every $x \in \mathcal{S}$, there exists $x_0 \in \mathcal{N}_\epsilon$ such that

$$d(x, x_0) \leq \epsilon.$$

An illustration of an ϵ-net is provided in Fig. 2.5.

We then prove the approximation on all pairs of points in the finite net, apply a union bound over $\mathcal{N}_\epsilon \times \mathcal{N}_\epsilon$, and extend the approximation from the net to the full set $\mathcal{S}$. This last step uses some property of the quantities involved, such as Lipschitz continuity or a separate stability argument. Such arguments can be found in [125, 135].

The size of the net will influence the probability bound term in the final inequality. If the fixed pair failure probability behaves like $C \exp(-cm\delta^2)$, then the union bound over the net produces terms involving

$$|\mathcal{N}_\epsilon|^2 \exp(-cm\delta^2).$$

Thus m must be large enough to cancel out $\log |\mathcal{N}_\epsilon|$, up to the desired accuracy and failure probability.

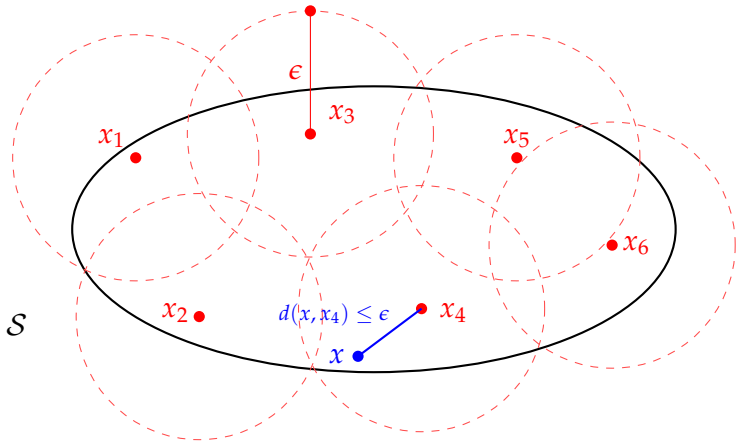


Fig. 2.5 Illustration of an ϵ-net $\mathcal{N}_\epsilon = \{x_1, \ldots, x_6\}$ for a set $\mathcal{S}$. Every point $x \in \mathcal{S}$ lies within distance at most ϵ of a point $x_i \in \mathcal{N}_\epsilon$.

2.7 Conclusion

This chapter presented the preliminary context for the rest of the text. We started from the general problem of representing high-dimensional data and large datasets in a way that preserves the quantities needed for estimation, or comparison, while reducing storage and computational costs. This led us to random embeddings, which replace the original objects by random lower-dimensional or feature-space representations with controlled approximation error.

We then introduced the geometric setting needed when data is represented as subspaces. Subspaces were represented through orthonormal bases and orthogonal projectors, and compared through principal angles and Grassmannian distances. This allowed us to present kernels on general spaces, then Grassmannian kernels in particular. Random features were also presented as explicit finite-dimensional embeddings whose inner products approximate kernel values, avoiding the need to form and store a full Gram matrix thus alleviating the computational and storage costs of kernel algorithms.

We also recalled how representations can be made more compact by reducing numerical precision. Quantisation replaces real-valued coordinates by finite-level representations, while binarisation keeps only one bit per entry.

Finally, we introduced the probabilistic language used to control random approximations. Concentration inequalities turn expectations into high-probability guarantees. Uniform approximation results widen fixed pair estimates by requiring the same random embedding

to work over an entire, possibly continuous set of objects using covering arguments and union bounds.

The next chapters build on these ideas. We first study signal processing directly from random quadratic embeddings, then random feature approximations of Grassmannian kernels, and finally efficient implementations based on structured transforms and optical computation.

3 Signal processing in the quadratic embedding domain

3.1 Introduction

The previous chapter introduced random embeddings as a way of changing the representation of data while preserving quantities needed for later processing. We now turn to the first concrete setting studied in this manuscript. The objects of interest are signals $\boldsymbol{x} \in \mathbb{R}^n$, and their representations are obtained through *random quadratic embeddings* as described in Sec. 2.2.4.

The central question of this chapter revolves around applying signal processing tasks *after* the embedding and without ever reconstructing the original signal representation. In many applications, the full signal is not the final object of interest. We may only need to estimate whether a given pattern is present, how strongly a signal overlaps with a region of interest, or whether an embedded observation belongs to a given class.

In this chapter, we show that a random quadratic embedding of any $k-$sparse vector $\boldsymbol{x} \in \mathbb{R}^n$ keeps enough information to estimate certain quantities depending on $\langle \boldsymbol{u}, \boldsymbol{x} \rangle$, where $\boldsymbol{u}$ is a fixed unit vector. After introducing the embedding model, we show that a *debiased* quadratic embedding satisfies a sign product embedding property. This allows us to approximate $\langle \boldsymbol{u}, \boldsymbol{x} \rangle^2$ directly from embedded data. We then interpret this mechanism as an asymmetric random feature approximation of a quadratic kernel and illustrate it through numerical experiments.

3.1.1 Motivation

The previous chapter introduced random embeddings as data-agnostic transformations used in signal processing, numerical linear algebra and machine learning [107, 77, 21]. Classical examples include linear random projections for dimensionality reduction and compressed sensing [91], as well as random feature constructions where random projections are combined with non-linear functions to induce kernel similarities [133, 126]. For such embeddings to be useful, the transformed representation must still preserve the quantities needed for a task at hand. We follow this philosophy here, but for the quadratic embeddings introduced in Sec. 2.2.4.

Quadratic embeddings arise naturally in several inverse and sensing problems. This is the case in phase retrieval, where the goal is to recover signals from intensity measurements [13, 12, 93, 80], in covariance estimation from quadratic samples [41, 40, 89], and in matrix recovery from rank-one projections [29]. Related lifted matrix-valued formulations also appear in low-rank matrix recovery [35], quantum state tomography [73], and compressive radio-interferometric imaging [100, 98]. These examples show that quadratic embeddings are used in concrete problems, where measurements of a vector $\boldsymbol{x}$ can often be interpreted as linear measurements of the lifted matrix $\boldsymbol{x}\boldsymbol{x}^\top$.

The question classically attached to such measurements is whether the original signal, or its lifted representation, can be reconstructed. This is not the question we pursue here. In this chapter, we study what signal processing tasks can be performed *directly in the embedding domain* generated by random quadratic embeddings, rather than focusing on reconstruction methods which can prove costly [67, 29, 36, 41]. More precisely, given a signal $\boldsymbol{x}$ and a fixed vector $\boldsymbol{u}$, we study how to estimate quantities depending on $\langle \boldsymbol{u}, \boldsymbol{x} \rangle$ directly from the embeddings of $\boldsymbol{x}$ and $\boldsymbol{u}$, by constructing a function $g(\boldsymbol{u}, \boldsymbol{y})$ that approximates a target function $f\big(\langle \boldsymbol{u}, \boldsymbol{x} \rangle\big)$. This setting is illustrated in Fig. 3.1.

The choice of the function f depends on the embedding under consideration. In the random linear case, the natural function is $f(t) = t$, since comparisons between the embeddings of $\boldsymbol{x}$ and $\boldsymbol{u}$ can be used to approximate their scalar product. This is, for instance, the setting behind the linear sign product embedding (SPE) property studied in [85]. For the quadratic embeddings considered here, however, $\boldsymbol{x}$ and $-\boldsymbol{x}$ have the same embedding. The sign of $\langle \boldsymbol{u}, \boldsymbol{x} \rangle$ is therefore lost, so that the linear function $f(t) = t$ cannot in general be recovered from the quadratic embedding alone. This naturally motivates the choice $f(t) = t^2$ studied in this chapter.

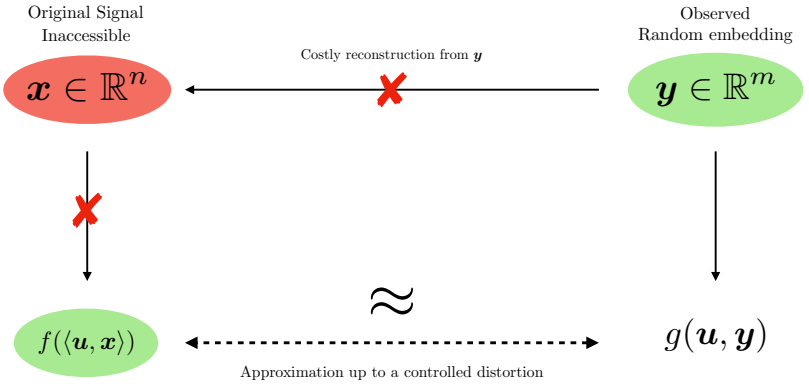


Fig. 3.1 Illustration of the framework: instead of reconstructing the signal x from its embedding y, we directly estimate a function $g(u, y)$ to approximate $f(\langle u, x \rangle)$, with a controlled error.

This will lead us to an *asymmetric* and partly binarised comparison between embedded objects and connect the chapter to the ideas of quantisation and binarisation discussed in Sec. 2.5.

3.1.2 Related works

Our work is closely related to signal processing with compressive measurements in the linear setting [49], where several tasks can be performed directly from embedded data without explicit reconstruction. We follow the same general philosophy, but in a quadratic framework. Similar work can also be found in [31] where the authors propose to perform learning tasks directly in the compressed domain, showing that classification can be carried out from random projections without explicit signal recovery.

Our main theoretical result is also closely related to the sign product embedding (SPE) property introduced for random linear embeddings in [85]. In the linear setting, the SPE uses an asymmetric comparison involving the sign of one linear embedding to approximate the scalar product $\langle u, x \rangle$. In this chapter, we extend this principle to the debiased quadratic embedding, for which the corresponding quantity is the squared scalar product $\langle u, x \rangle^2$.

In addition, the estimator that we consider induces a comparison between embedded signals that can be interpreted as a kernel-like similarity. This connects our approach to random feature constructions, where inner products in an embedding domain are used to approximate kernel evaluations, as in [125, 135, 107]. In particular, our construction can be seen as generating features whose inner products esti-

mate a function of $\langle \boldsymbol{u}, \boldsymbol{x} \rangle$, in a way that is reminiscent of random feature approximations, but arising from quadratic rather than linear embeddings.

3.1.3 Contributions

The main theoretical contribution of this chapter is a quadratic extension of the SPE property to the debiased quadratic embedding. The core result of this chapter shows that, after a simple debiasing step, the quadratic embedding of a signal can be combined with the sign of the quadratic embedding of a unit vector to approximate a function of their scalar product with controlled distortion. More precisely, this comparison gives access to the squared scalar product between the signal and the unit vector directly in the embedding domain. The resulting guarantee holds uniformly over the considered class of sparse signals, and can therefore be viewed as the quadratic equivalent of the existing linear SPE. This provides a direct way to perform basic estimation, detection and classification tasks from quadratic embeddings alone. Aside from this signal processing interpretation, the same mechanism also implies a kernel viewpoint since the resulting comparison behaves like an asymmetric random feature approximation of a quadratic polynomial kernel.

The rest of this chapter is organised as follows. We first introduce the random quadratic embedding model and its debiased version, and link them to rank-one projections. We then state and analyse the main estimation result in the embedding domain. After that, we discuss its interpretation, including its link with asymmetric random feature constructions. Finally, we illustrate these ideas on synthetic estimation problems, localised detection in images, and simple classification experiments.

3.1.4 Related publications

This chapter is related to the following publications, with contributions from Vincent Schellekens, Laurent Daudet, Laurent Jacques, and the author of the present document.

- R. Delogne, V. Schellekens, L. Daudet, and L. Jacques. “ROP inception: Signal estimation with quadratic random sketching”. In: *Proceedings of the 30th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)*. 2022, [54]

- R. Delogne, V. Schellekens, L. Daudet, and L. Jacques. "Signal processing with optical quadratic random sketches". In: *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2023, pp. 1–5, [56]
- R. Delogne, V. Schellekens, L. Daudet, and L. Jacques. "Signal processing after quadratic random sketching with optical units". In: *International Symposium on Computational Sensing (ISCS)*. 2023, [55]
- R. Delogne and L. Jacques. "Quadratic polynomial kernel approximation with asymmetric embeddings". In: *International Workshop on Deep Learning and Kernel Machines (DEEPK)*. 2024, [51]

3.2 Quadratic random embedding of signals

We now introduce the random quadratic embedding model used throughout this chapter. We start from the basic construction based on rank-one projections, and then introduce a debiased variant that will play a central role in the estimation results.

3.2.1 Symmetric rank-one projections

The use of rank-one projections as random measurements was introduced in [29]. In this section, we focus first on a symmetric version, which will be used as a building block for the embedding model studied in this chapter. Let us first remind the definition of ψ^{s} as presented in Eq. 2.8.

Definition 3.2.1 (Symmetric ROP embedding)**.** *Let $\boldsymbol{x} \in \mathbb{R}^n$ and let $(\boldsymbol{a}_i)_{i=1}^m$ be independent random vectors in $\mathbb{R}^n$ with $\boldsymbol{a}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_n), \forall i$. The symmetric rank-one projection embedding $\psi^{\mathrm{s}}(\boldsymbol{x}) \in \mathbb{R}^m$ is defined componentwise as a collection of measurements*

$$\psi^{\mathrm{s}}(\boldsymbol{x}) = \left(\langle \boldsymbol{a}_i, \boldsymbol{x}\rangle^2\right)_{i=1}^m, \qquad i = 1, \dots, m.$$

This construction corresponds to taking quadratic measurements of $\boldsymbol{x}$ of the form $(\langle \boldsymbol{a}_i, \boldsymbol{x}\rangle)^2$. Each measurement (or projection) reflects how well $\boldsymbol{x}$ aligns with a random vector $\boldsymbol{a}_i$, and the collection of measurements therefore captures how $\boldsymbol{x}$ interacts with many random directions. The embedding can thus be interpreted as a representation of $\boldsymbol{x}$ through its responses to a set of random probes.

Recall that this can also be interpreted as a linear mapping of *lifted* objects, as each quadratic projection can be rewritten as a linear projection of those lifted objects. Indeed Eq. 2.9 shows that

$$\psi^{\mathrm{s}}(\boldsymbol{x}) = \big(\langle \boldsymbol{A}_i, \boldsymbol{X} \rangle\big),$$

where $\boldsymbol{X} = \boldsymbol{x}\boldsymbol{x}^\top$ and where $\boldsymbol{A}_i = \boldsymbol{a}_i\boldsymbol{a}_i^\top$ are random rank-one matrices. The embedding is thus quadratic in $\boldsymbol{x}$ but linear in the lifted space, where it corresponds to projecting $\boldsymbol{X}$ onto random rank-one matrices. This viewpoint explains the terminology of *rank-one projections*.

A key property of this embedding is that it is *biased*. Indeed, for any $\boldsymbol{x} \in \mathbb{R}^n$, we have

$$\begin{aligned}
\mathbb{E}\big[\psi^{\mathrm{s}}(\boldsymbol{x})_i\big] &= \mathbb{E}\big[\langle \boldsymbol{a}_i, \boldsymbol{x}\rangle^2\big] \\
&= \mathbb{E}\big[\boldsymbol{x}^\top \boldsymbol{a}_i\boldsymbol{a}_i^\top \boldsymbol{x}\big] \\
&= \boldsymbol{x}^\top \mathbb{E}\big[\boldsymbol{a}_i\boldsymbol{a}_i^\top\big]\boldsymbol{x} \\
&= \boldsymbol{x}^\top \mathbf{I}_n \boldsymbol{x} \\
&= \|\boldsymbol{x}\|_2^2.
\end{aligned}$$

This shows that each term $\langle \boldsymbol{a}_i, \boldsymbol{x}\rangle^2$ has a non-zero mean equal to $\|\boldsymbol{x}\|_2^2$. This mean does not give us information about the alignment between $\boldsymbol{a}_i$ and $\boldsymbol{x}$, but only the norm of the signal. The directional information is therefore carried by the changes of $\langle \boldsymbol{a}_i, \boldsymbol{x}\rangle^2$ around this average value. This becomes problematic when comparing two embedded signals, since the raw scalar product between their quadratic measurements also contains a term depending only on their norms. Consequently, two orthogonal signals may still have a positive expected similarity after embedding, which hides the geometric information we want to preserve.

As a result, when using these measurements to compare signals or estimate quantities depending on $\langle \boldsymbol{u}, \boldsymbol{x}\rangle$, the dominant contribution comes from the norm of the signals (since it is going to accumulate when we combine many measurements), rather than from their orientations relative to the random vectors $\boldsymbol{a}_i$. This hides the directional information which is precisely the interesting part of the embedding that we want to preserve. This motivates the introduction of a *debiased* embedding.

3.2.2 Debiased rank-one projections

In order to remove the bias effect described just above, we consider a simple *debiasing* strategy based on taking differences of independent measurements, following the approach of [41] and [99].

Definition 3.2.2 (Debiased ROP embedding)**.** *Let $(\boldsymbol{a}_i)_{i=1}^{2m}$ be independent random vectors in $\mathbb{R}^n$ with $\boldsymbol{a}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_n)$. The debiased ROP embedding $\psi^-(\boldsymbol{x}) \in \mathbb{R}^m$ is defined as*

$$\psi^-(\boldsymbol{x})_i = \langle \boldsymbol{a}_i, \boldsymbol{x} \rangle^2 - \langle \boldsymbol{a}_{m+i}, \boldsymbol{x} \rangle^2, \qquad i = 1, \dots, m.$$

This construction can be seen as the difference of two single independent symmetric ROP embeddings. In particular, it can be implemented using the same basic symmetric quadratic embedding ψ^s, which will be relevant in settings where only symmetric projections are available such as in Optical Processing Units which will be presented in Chap. 5.

Remark 3.1. *The construction of the debiased embedding uses $2m$ independent probes instead of m. This effectively doubles the number of required probing vectors, but does not change the scaling of the embedding dimension. In particular, all results expressed in terms of m remain of the same order. As noted in [41], this modification does not affect the behaviour of the bounds derived for the embedding.*

By construction, the debiased embedding satisfies

$$\begin{aligned} \mathbb{E}\left[\psi^-(\boldsymbol{x})_i\right] &= \mathbb{E}\left[\langle \boldsymbol{a}_i, \boldsymbol{x} \rangle^2\right] - \mathbb{E}\left[\langle \boldsymbol{a}_{m+i}, \boldsymbol{x} \rangle^2\right] \\ &= \|\boldsymbol{x}\|_2^2 - \|\boldsymbol{x}\|_2^2 = 0, \end{aligned}$$

so that the constant contribution from $\|\boldsymbol{x}\|^2$ present in each measurement is now removed, meaning that, in contrast with the symmetric embedding, the debiased embedding is now centred, and therefore only shows variations due to the alignment between the random vectors and the signal. This will allow us to extract useful directional information from this embedding. This debiased construction satisfies an *isotropy* property, meaning that it preserves, in expectation, certain geometric quantities of the signal, as shown in the following proposition.

A more general version of this result is given in [41, Lem. 4], where the debiased embedding is extended to matrices of arbitrary rank and to sub-Gaussian probing vectors. For completeness, we provide here a short proof in the vector Gaussian case.

Proposition 3.1 (Isotropy of the debiased embedding)**.** *Given $\boldsymbol{x} \in \mathbb{R}^n$, we have*

$$\tfrac{1}{m}\mathbb{E}\|\psi^-(\boldsymbol{x})\|^2 = 4\|\boldsymbol{x}\|^4. \tag{3.1}$$

Proof. Let $\boldsymbol{a}, \boldsymbol{b} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_n)$ be independent. By definition,

$$\psi^-(\boldsymbol{x})_i = (\boldsymbol{a}^\top \boldsymbol{x})^2 - (\boldsymbol{b}^\top \boldsymbol{x})^2.$$

By rotational invariance of the Gaussian distribution, we can assume without loss of generality that $\boldsymbol{x} = \|\boldsymbol{x}\|\boldsymbol{e}_1$. Then

$$\psi^-(\boldsymbol{x})_i = \|\boldsymbol{x}\|^2(a_1^2 - b_1^2),$$

and therefore

$$\mathbb{E}\big[\psi^-(\boldsymbol{x})_i^2\big] = \|\boldsymbol{x}\|^4 \mathbb{E}\big[(a_1^2 - b_1^2)^2\big].$$

Expanding,

$$\mathbb{E}\big[(a_1^2 - b_1^2)^2\big] = \mathbb{E}[a_1^4] + \mathbb{E}[b_1^4] - 2\mathbb{E}[a_1^2 b_1^2].$$

Since a_1 and b_1 are independent standard Gaussian variables,

$$\mathbb{E}[a_1^4] = 3, \qquad \mathbb{E}[a_1^2 b_1^2] = 1,$$

which yields

$$\mathbb{E}\big[(a_1^2 - b_1^2)^2\big] = 4.$$

Hence,

$$\mathbb{E}\big[\psi^-(\boldsymbol{x})_i^2\big] = 4\|\boldsymbol{x}\|^4,$$

and summing over $i = 1, \ldots, m$ gives

$$\tfrac{1}{m}\mathbb{E}\|\psi^-(\boldsymbol{x})\|^2 = 4\|\boldsymbol{x}\|^4.$$

□

This result can be interpreted in the lifted space. Indeed, recalling that $\boldsymbol{X} = \boldsymbol{x}\boldsymbol{x}^\top$, we have $\|\boldsymbol{X}\|_F^2 = \|\boldsymbol{x}\|^4$. Prop. 3.1 therefore shows that the debiased embedding is correctly scaled with respect to the Frobenius norm of the lifted signal. In other words, it behaves, in expectation, like a well-calibrated linear embedding of $\boldsymbol{X}$.

Remark 3.2 (Connection with asymmetric projections)**.** *For completeness, we note that the debiased embedding can be rewritten as*

$$\psi^-(\boldsymbol{x})_i = \big(\langle \boldsymbol{d}_i, \boldsymbol{x}\rangle\big)\big(\langle \boldsymbol{s}_i, \boldsymbol{x}\rangle\big),$$

where $\boldsymbol{d}_i = \boldsymbol{a}_i - \boldsymbol{a}_{m+i}$ and $\boldsymbol{s}_i = \boldsymbol{a}_i + \boldsymbol{a}_{m+i}$. Assuming $\boldsymbol{a}_i$ follow a Gaussian distribution, $\boldsymbol{d}_i$ and $\boldsymbol{s}_i$ are uncorrelated and therefore independent Gaussian vectors for all i, so that ψ^- is equivalent to an asymmetric rank-one projection embedding ψ^{rop} up to a rescaling. This idea will be useful later, in particular in connection with implementations based on optical processing units (see Chap. 5) or asymmetric ROP (see Chap. 4).

3.3 Signal estimation in the embedding domain

We now turn to the main objective of this chapter. Given a signal $\boldsymbol{x} \in \mathbb{R}^n$ and a unit vector $\boldsymbol{u} \in \mathbb{R}^n$, we want to estimate a quantity of the form

$$f\big(\langle \boldsymbol{u}, \boldsymbol{x} \rangle\big)$$

directly from $\boldsymbol{y} = \psi^-(\boldsymbol{x})$, the embedding of $\boldsymbol{x}$, and a unit vector $\boldsymbol{u} \in \mathbb{R}^n$, without reconstructing $\boldsymbol{x}$. In the present setting, we will consider the square function

$$f : t \in \mathbb{R} \mapsto t^2.$$

Our goal is therefore to construct a function g acting on $\boldsymbol{y} := \psi^-(\boldsymbol{x})$, the embedding of $\boldsymbol{x}$ and $\boldsymbol{u}$ such that

$$g(\boldsymbol{u}, \boldsymbol{y}) \approx \langle \boldsymbol{u}, \boldsymbol{x} \rangle^2.$$

The key tool is a sign product embedding property satisfied by the debiased quadratic embedding with high probability. This result relies on the sign product embedding (SPE) framework introduced in [85]. The idea relies on the simple trick of applying the same embedding procedure ψ^- to $\boldsymbol{u}$, thus sending $\boldsymbol{u}$ to the same domain as $\boldsymbol{y}$, with the goal to perform the comparison between $\boldsymbol{y}$ and $\psi^-(\boldsymbol{u})$ in the embedding domain.

In order to obtain guarantees that hold simultaneously over a class of signals, rather than for a single fixed instance, it is necessary to restrict attention to structured signals. As presented in Sec. 2.2.3, such uniform guarantees are standard in compressive sensing and random embedding theory, where the complexity of the signal class controls the required number of measurements (See for example [135, 15, 35]).

We will apply this idea with the set of k-sparse vectors in the canonical basis,

$$\Sigma_k := \{\boldsymbol{v} \in \mathbb{R}^n : |\mathrm{supp}\,(\boldsymbol{v})| \leq k\}.$$

The role of Σ_k is to provide a simple low-complexity signal class on which uniform guarantees can be stated. It is useful because the embedding dimension in Prop. 3.2 scales proportionally to k, up to logarithmic factors. Thus, dimensionality reduction is meaningful when $k \ll n$, whereas in the extreme case $k = n$ the required number of measurements becomes comparable to the ambient dimension.

Remark 3.3. *More generally, and as already discussed in Sec. 2.2.3, many signals are not sparse in the canonical basis but admit sparse representations*

in an orthonormal basis, for instance images in wavelet bases. Let $\boldsymbol{V} \in \mathbb{R}^{n\times n}$ be orthonormal and consider signals of the form

$$\boldsymbol{x} = \boldsymbol{V}\boldsymbol{\alpha}, \qquad \boldsymbol{\alpha} \in \Sigma_k.$$

Since Gaussian distributions are rotationally invariant, for any orthonormal matrix $\boldsymbol{V}$, the random vectors $\boldsymbol{a}_i$ and $\boldsymbol{V}^\top \boldsymbol{a}_i$ have the same distribution. Therefore, applying the embedding to signals of the form $\boldsymbol{x} = \boldsymbol{V}\boldsymbol{\alpha}$ with $\boldsymbol{\alpha} \in \Sigma_k$ is statistically equivalent to applying it to k-sparse vectors in the canonical basis. As a result, all guarantees stated below extend directly to vectors that are sparse in an orthonormal basis.

We can now state the main result of this chapter.

Proposition 3.2 (SPE for the debiased embedding)**.** *Given a fixed unit vector $\boldsymbol{u} \in \mathbb{R}^n$ and a distortion $0 < \delta < 1$, provided that*

$$m \geq C\delta^{-2} k \log\left(\tfrac{n}{k\delta}\right), \tag{3.2}$$

then, with probability at least $1 - C\exp(-c\delta^2 m)$, for all $\boldsymbol{x} \in \Sigma_k$,

$$\left| \tfrac{\pi}{4m} \langle \operatorname{sign}(\psi^-(\boldsymbol{u})), \psi^-(\boldsymbol{x}) \rangle - \langle \boldsymbol{u}, \boldsymbol{x} \rangle^2 \right| \leq \delta \|\boldsymbol{x}\|^2. \tag{3.3}$$

Proof. The proof follows a three-step argument as described in Sec. 2.6. We first compute the expectation of the estimator and show that it matches the target quantity $\langle \boldsymbol{u}, \boldsymbol{x} \rangle^2$. We then establish concentration around this expectation for a fixed signal $\boldsymbol{x}$. Finally, we extend this concentration to a uniform guarantee over Σ_k using a covering argument combined with a union bound. It follows the same structure as presented in the supplementary material of [54] but has been rewritten slightly.

Expectation: We first compute the expectation of

$$\tfrac{\pi}{4m} \langle \operatorname{sign}(\psi^-(\boldsymbol{u})), \psi^-(\boldsymbol{x}) \rangle.$$

By rotational invariance of the Gaussian distribution and homogeneity in $\|\boldsymbol{x}\|^2$, it is sufficient to consider $\boldsymbol{u} = \boldsymbol{e}_1$ and $\boldsymbol{x} = c\boldsymbol{e}_1 + s\boldsymbol{e}_2$ with $c = \cos\theta$, $s = \sin\theta$.

Let $\boldsymbol{a}, \boldsymbol{b} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_n)$ be independent. Then

$$\begin{aligned}
&\tfrac{1}{m}\mathbb{E}\langle \operatorname{sign}(\psi^-(\boldsymbol{u})), \psi^-(\boldsymbol{x})\rangle \\
&\quad = \mathbb{E}\operatorname{sign}(a_1^2 - b_1^2)\left[(ca_1 + sa_2)^2 - (cb_1 + sb_2)^2\right] \\
&\quad = \mathbb{E}\operatorname{sign}(a_1^2 - b_1^2)\left[c^2(a_1^2 - b_1^2) + s^2(a_2^2 - b_2^2) + 2cs(a_1a_2 - b_1b_2)\right] \\
&\quad = c^2\mathbb{E}\operatorname{sign}(a_1^2 - b_1^2)(a_1^2 - b_1^2),
\end{aligned}$$

where the terms in s^2 and $2cs$ vanish since a_2, b_2 are independent of a_1, b_1 and centred. We then have

$$\tfrac{1}{m}\mathbb{E}\langle \operatorname{sign}(\psi^-(\boldsymbol{u})), \psi^-(\boldsymbol{x})\rangle = c^2\mathbb{E}|a_1^2 - b_1^2| = c^2\mathbb{E}|a_1 - b_1||a_1 + b_1|.$$

Since $a_1 - b_1 \sim \mathcal{N}(0, 2)$ and $a_1 + b_1 \sim \mathcal{N}(0, 2)$ are Gaussian and uncorrelated, they are independent. Writing

$$a_1 - b_1 = \sqrt{2}Z_1, \quad a_1 + b_1 = \sqrt{2}Z_2,$$

with $Z_1, Z_2 \sim \mathcal{N}(0, 1)$ independent, we obtain

$$\mathbb{E}|a_1 - b_1||a_1 + b_1| = 2\mathbb{E}|Z_1|\mathbb{E}|Z_2|.$$

Since Z_1 and Z_2 are identically distributed as $\mathcal{N}(0, 1)$, it remains to compute $\mathbb{E}|Z|$ for $Z \sim \mathcal{N}(0, 1)$. We have

$$\mathbb{E}|Z| = \textstyle\int_{\mathbb{R}} |z| \frac{1}{\sqrt{2\pi}} e^{-z^2/2} dz = 2\int_0^{+\infty} z \frac{1}{\sqrt{2\pi}} e^{-z^2/2} dz.$$

With the change of variable $u = z^2/2$, so that $du = zdz$, this gives

$$\mathbb{E}|Z| = \tfrac{2}{\sqrt{2\pi}} \textstyle\int_0^{+\infty} e^{-u} du = \frac{2}{\sqrt{2\pi}} = \sqrt{\frac{2}{\pi}}.$$

Therefore,

$$\mathbb{E}|a_1 - b_1||a_1 + b_1| = 2\mathbb{E}|Z_1|\mathbb{E}|Z_2| = 2\left(\sqrt{\tfrac{2}{\pi}}\right)^2 = \tfrac{4}{\pi}.$$

This allows us to conclude that

$$\tfrac{\pi}{4m}\mathbb{E}\langle \operatorname{sign}(\psi^-(\boldsymbol{u})), \psi^-(\boldsymbol{x})\rangle = c^2 = \langle \boldsymbol{u}, \boldsymbol{x}\rangle^2.$$

Concentration: We now study concentration for fixed $\boldsymbol{u}$ and, since Eq. 3.3 is homogeneous, we can consider $\boldsymbol{x}$ fixed with $\|\boldsymbol{x}\| = 1$ without loss of generality. Let us define

$$S := \tfrac{\pi}{4}\langle \operatorname{sign}(\psi^-(\boldsymbol{u})), \psi^-(\boldsymbol{x})\rangle - m\langle \boldsymbol{u}, \boldsymbol{x}\rangle^2 = \textstyle\sum_{i=1}^m Y_i,$$

with

$$Y_i := \tfrac{\pi}{4} Z_i - \langle \boldsymbol{u}, \boldsymbol{x} \rangle^2, \qquad Z_i := s_i(\boldsymbol{u}) \big[(\boldsymbol{a}_i^\top \boldsymbol{x})^2 - (\boldsymbol{b}_i^\top \boldsymbol{x})^2 \big],$$

and $s_i(\boldsymbol{u}) = \operatorname{sign}\big[(\boldsymbol{a}_i^\top \boldsymbol{u})^2 - (\boldsymbol{b}_i^\top \boldsymbol{u})^2 \big]$.

The variables Z_i are i.i.d. and sub-exponential, with

$$\|Z_i\|_{\psi_1} \leq 2 \|(\boldsymbol{a}_i^\top \boldsymbol{x})^2\|_{\psi_1} \leq 4 \|\boldsymbol{a}_i^\top \boldsymbol{x}\|_{\psi_2}^2 \leq C,$$

see [149, Def. 5.13 & Lemma 5.14]. Hence the variables Y_i are i.i.d. , centred, and sub-exponential, with bounded sub-exponential norm. Therefore, by Bernstein's inequality [149, Cor. 5.17], for any $\delta \geq 0$,

$$\mathbb{P}\big[|S| \geq \delta m \big] \leq 2 \exp(-c \min(\delta^2, \delta) m),$$

that is for $\boldsymbol{u}$ and $\boldsymbol{x}$ fixed,

$$\mathbb{P}\Big[\big| \tfrac{\pi}{4m} \langle \operatorname{sign}(\psi^-(\boldsymbol{u})), \psi^-(\boldsymbol{x}) \rangle - \langle \boldsymbol{u}, \boldsymbol{x} \rangle^2 \big| \geq \delta \Big] \leq 2 \exp(-c \min(\delta^2, \delta) m). \tag{3.4}$$

Finally since we assume that $0 < \delta < 1$, we can always choose $\min(\delta^2, \delta) = \delta^2$.

Covering: We extend this bound uniformly over vectors $\boldsymbol{x} \in \Sigma_k \cap \mathbb{S}^{n-1}$ and $\boldsymbol{u}$ fixed. Write

$$\psi^-(\boldsymbol{x}) = \big((\boldsymbol{a}_i^\top \boldsymbol{x})^2 - (\boldsymbol{b}_i^\top \boldsymbol{x})^2 \big)_{i=1}^m = \big(\boldsymbol{d}_i^\top \boldsymbol{X} \boldsymbol{s}_i \big)_{i=1}^m,$$

with $\boldsymbol{d}_i = \boldsymbol{a}_i - \boldsymbol{b}_i$, $\boldsymbol{s}_i = \boldsymbol{a}_i + \boldsymbol{b}_i$ and $\boldsymbol{X} = \boldsymbol{x}\boldsymbol{x}^\top$.

Let $\mathcal{G} = \{ \boldsymbol{x}\boldsymbol{x}^\top : \boldsymbol{x} \in \Sigma_k \cap \mathbb{S}^{n-1} \}$. Given $\epsilon > 0$, there exists an ϵ-net $\mathcal{G}_\epsilon \subset \mathcal{G}$ such that for every $\boldsymbol{X} \in \mathcal{G}$, there exists $\boldsymbol{X}' \in \mathcal{G}_\epsilon$ satisfying

$$\|\boldsymbol{X} - \boldsymbol{X}'\|_F \leq \epsilon,$$

and $\boldsymbol{X}'$ has the same row and column supports as $\boldsymbol{X}$. Moreover, the cardinality of this net satisfies

$$|\mathcal{G}_\epsilon| \leq \tbinom{n}{k} (9/\epsilon)^{2k+1} \leq \left(\tfrac{cn}{\epsilon k} \right)^{2k+1},$$

see [35, Lemma 3.1]. The binomial term comes from counting the possible supports of size k, and the net can be constructed separately on each support before taking the union over all supports.

Fix $\bar{\boldsymbol{u}} = \operatorname{sign}(\psi^-(\boldsymbol{u}))$. Note that

$$\tfrac{\pi}{4m} \langle \bar{\boldsymbol{u}}, \psi^-(\boldsymbol{x}) \rangle - \langle \boldsymbol{u}, \boldsymbol{x} \rangle^2 = \tfrac{\pi}{4m} \langle \bar{\boldsymbol{u}}, \big(\boldsymbol{d}_i^\top \boldsymbol{X} \boldsymbol{s}_i \big)_{i=1}^m \rangle - \boldsymbol{u}^\top \boldsymbol{X} \boldsymbol{u}.$$

By a union bound applied to Eq.3.4, with probability at least

$$1 - 2|\mathcal{G}_\epsilon| \exp(-c\delta^2 m),$$

the following holds for all $\boldsymbol{X}' \in \mathcal{G}_\epsilon$:

$$\left| \tfrac{\pi}{4m} \langle \bar{\boldsymbol{u}}, \left(\boldsymbol{d}_i^\top \boldsymbol{X}' \boldsymbol{s}_i \right)_{i=1}^m \rangle - \boldsymbol{u}^\top \boldsymbol{X}' \boldsymbol{u} \right| \leq \delta.$$

Using the bound $|\mathcal{G}_\epsilon| \leq \left(\frac{cn}{\epsilon k}\right)^{2k+1}$ and choosing $\epsilon = \delta/4$, the condition from Eq. 3.2 ensures that

$$2|\mathcal{G}_\epsilon| \exp(-c\delta^2 m) \leq C \exp(-c\delta^2 m).$$

The rest of the proof assumes that this event holds.

Define

$$\boldsymbol{Q} := \tfrac{\pi}{4m} \textstyle\sum_{i=1}^m \bar{u}_i \boldsymbol{d}_i \boldsymbol{s}_i^\top - \boldsymbol{u}\boldsymbol{u}^\top.$$

Then for all $\boldsymbol{X} \in \mathcal{G}$,

$$\tfrac{\pi}{4m} \langle \bar{\boldsymbol{u}}, \left(\boldsymbol{d}_i^\top \boldsymbol{X} \boldsymbol{s}_i \right)_{i=1}^m \rangle - \boldsymbol{u}^\top \boldsymbol{X} \boldsymbol{u} = \langle \boldsymbol{Q}, \boldsymbol{X} \rangle.$$

Let

$$\rho = \sup_{\boldsymbol{X} \in \mathcal{G}} |\langle \boldsymbol{Q}, \boldsymbol{X} \rangle|.$$

Fix $\boldsymbol{X} \in \mathcal{G}$ and let $\boldsymbol{X}' \in \mathcal{G}_\epsilon$ be such that $\|\boldsymbol{X} - \boldsymbol{X}'\|_F \leq \epsilon$, with $\boldsymbol{X}'$ having the same row and column supports as $\boldsymbol{X}$. Define

$$\boldsymbol{Y} = \frac{\boldsymbol{X} - \boldsymbol{X}'}{\|\boldsymbol{X} - \boldsymbol{X}'\|_F}.$$

Then $\|\boldsymbol{Y}\|_F = 1$ and

$$\boldsymbol{X} = \boldsymbol{X}' + \|\boldsymbol{X} - \boldsymbol{X}'\|_F \boldsymbol{Y},$$

so that

$$|\langle \boldsymbol{Q}, \boldsymbol{X} \rangle| \leq |\langle \boldsymbol{Q}, \boldsymbol{X}' \rangle| + |\langle \boldsymbol{Q}, \boldsymbol{Y} \rangle| \|\boldsymbol{X} - \boldsymbol{X}'\|_F \leq |\langle \boldsymbol{Q}, \boldsymbol{X}' \rangle| + |\langle \boldsymbol{Q}, \boldsymbol{Y} \rangle| \epsilon.$$

It remains to bound $|\langle \boldsymbol{Q}, \boldsymbol{Y} \rangle|$. Since both $\boldsymbol{X}$ and $\boldsymbol{X}'$ have rank 1, the matrix $\boldsymbol{Y}$ has rank at most 2. Moreover, since $\boldsymbol{X}$ and $\boldsymbol{X}'$ have the same row and column supports, there exists a set $S \subset [n]$ with $|S| \leq k$ such that both $\boldsymbol{X}$ and $\boldsymbol{X}'$ are supported on $S \times S$. Hence $\boldsymbol{Y}$ is also supported on $S \times S$.

Since $\boldsymbol{Y}$ is symmetric and has rank at most 2, it admits a spectral decomposition

$$\boldsymbol{Y} = \lambda_1 \boldsymbol{v}_1 \boldsymbol{v}_1^\top + \lambda_2 \boldsymbol{v}_2 \boldsymbol{v}_2^\top,$$

and since $\boldsymbol{Y}$ is supported on $S \times S$, each eigenvector $\boldsymbol{v}_i$ is supported on S, hence $\boldsymbol{v}_i \in \Sigma_k$. After normalisation, the vectors $\boldsymbol{v}_i$ are unit-norm eigenvectors, so that

$$\boldsymbol{v}_i \boldsymbol{v}_i^\top \in \mathcal{G}.$$

Therefore, by definition of ρ,

$$|\langle \boldsymbol{Q}, \boldsymbol{v}_i \boldsymbol{v}_i^\top \rangle| \leq \rho.$$

Using the spectral decomposition, we get

$$|\langle \boldsymbol{Q}, \boldsymbol{Y} \rangle| \leq |\lambda_1| \, |\langle \boldsymbol{Q}, \boldsymbol{v}_1 \boldsymbol{v}_1^\top \rangle| + |\lambda_2| \, |\langle \boldsymbol{Q}, \boldsymbol{v}_2 \boldsymbol{v}_2^\top \rangle| \leq (|\lambda_1| + |\lambda_2|)\rho.$$

Since $\boldsymbol{Y}$ has rank at most 2 and $\|\boldsymbol{Y}\|_F = 1$, its non-zero eigenvalues satisfy

$$\lambda_1^2 + \lambda_2^2 = \|\boldsymbol{Y}\|_F^2 = 1.$$

Hence, by Cauchy-Schwarz,

$$|\lambda_1| + |\lambda_2| \leq \sqrt{2}\sqrt{\lambda_1^2 + \lambda_2^2} = \sqrt{2}.$$

Thus

$$|\langle \boldsymbol{Q}, \boldsymbol{Y} \rangle| \leq \sqrt{2}\rho.$$

Combining previous estimates and using the fact that $|\langle \boldsymbol{Q}, \boldsymbol{X}' \rangle| \leq \delta$ for all $\boldsymbol{X}' \in \mathcal{G}_\epsilon$, we obtain

$$|\langle \boldsymbol{Q}, \boldsymbol{X} \rangle| \leq \delta + \sqrt{2}\rho\,\epsilon.$$

Taking the supremum over $\boldsymbol{X} \in \mathcal{G}$ yields

$$\rho \leq \delta + \sqrt{2}\rho\,\epsilon.$$

Choosing $\epsilon = \delta/4$ and using $\delta < 1$, we get

$$\rho \leq \frac{\delta}{1 - \sqrt{2}\epsilon} \leq 2\delta.$$

A final rescaling of δ concludes the proof.

□

Proposition 3.2 states that, provided that m is sufficiently large compared to the complexity of the signal model, the function

$$g\big(\boldsymbol{u}, \psi^-(\boldsymbol{x})\big) = \tfrac{\pi}{4m} \langle \operatorname{sign}(\psi^-(\boldsymbol{u})), \psi^-(\boldsymbol{x}) \rangle$$

gives an approximation with controlled quality of

$$\langle \boldsymbol{u}, \boldsymbol{x} \rangle^2.$$

In other words, the interaction between $\boldsymbol{x}$ and $\boldsymbol{u}$ can be estimated directly in the embedding domain by projecting the embedding of $\boldsymbol{x}$ onto the sign of the embedding of $\boldsymbol{u}$. This also has a nice geometric interpretation through lifting. If we define $\boldsymbol{U} = \boldsymbol{u}\boldsymbol{u}^\top$ and $\boldsymbol{X} = \boldsymbol{x}\boldsymbol{x}^\top$, we have

$$\langle \boldsymbol{U}, \boldsymbol{X} \rangle_{\mathrm{F}} = \langle \boldsymbol{u}, \boldsymbol{x} \rangle^2.$$

For unit vectors $\boldsymbol{u}$ and $\boldsymbol{x}$, this is equal to $\cos^2(\theta)$, where θ is the angle between them. The SPE can therefore be interpreted as estimating, directly from the quadratic embeddings, the alignment between the one-dimensional subspaces spanned by $\boldsymbol{u}$ and $\boldsymbol{x}$. Note that this interpretation is consistent with the fact that the quadratic embedding does not distinguish $\boldsymbol{x}$ from $-\boldsymbol{x}$.

This approximation is useful only when the quantity to be estimated is sufficiently large compared to the distortion term. Indeed, using the scaling on m found in Eq. 3.2, the error in Eq. 3.3 scales as

$$\delta \|\boldsymbol{x}\|^2 = \mathcal{O}\left(\sqrt{\tfrac{k}{m}}\right) \|\boldsymbol{x}\|^2$$

up to logarithmic factors. Therefore, the estimate is informative only when $\langle \boldsymbol{u}, \boldsymbol{x}/\|\boldsymbol{x}\| \rangle^2$ is not too small compared to δ. In particular, taking $\boldsymbol{u}$ orthogonal to $\boldsymbol{x}$ provides a way to empirically evaluate the distortion level as a function of m, since in that case the true value is zero, so the estimate directly reflects the distortion level.

This result is in the same spirit as signal processing from compressed measurements in the linear setting [49], where one seeks to estimate useful quantities directly from transformed data without reconstructing the original signal. Here, however, this principle is achieved in a genuinely quadratic setting.

The previous proposition extends directly to a finite family of unit vectors.

Corollary 3.1. *Given a set of S unit vectors $\{\boldsymbol{u}_s\}_{s=1}^S$ and a distortion $0 < \delta < 1$, provided that*

$$m \geq C\delta^{-2}\Big(k \log\left(\tfrac{n}{k\delta}\right) + \log S\Big), \tag{3.5}$$

then, with probability at least $1 - C\exp(-c\delta^2 m)$, for all $\boldsymbol{x} \in \Sigma_k$ and all $s \in [S]$,

$$\left|\tfrac{\pi}{4m}\langle \operatorname{sign}(\psi^-(\boldsymbol{u}_s)), \psi^-(\boldsymbol{x})\rangle - \langle \boldsymbol{u}_s, \boldsymbol{x}\rangle^2\right| \leq \delta\|\boldsymbol{x}\|^2. \tag{3.6}$$

Proof. For each fixed $s \in [S]$, Proposition 3.2 ensures that the desired inequality holds with probability at least $1 - 2\exp(-c\delta^2 m)$.

Applying a union bound over the S vectors $\{\boldsymbol{u}_s\}_{s=1}^S$, we obtain that the inequality holds simultaneously for all $s \in [S]$ with probability at least

$$1 - 2S\exp(-c\delta^2 m).$$

Using that $2S\exp(-c\delta^2 m) = 2\exp(\log S - c\delta^2 m)$ and the condition

$$m \geq C\delta^{-2}\Big(k \log\left(\tfrac{n}{k\delta}\right) + \log S\Big),$$

we obtain that this probability is bounded below by $1 - C\exp(-c\delta^2 m)$, which concludes the proof. □

This corollary shows that the same embedding can be used to estimate several quantities of the form $\langle \boldsymbol{u}_s, \boldsymbol{x}\rangle^2$ simultaneously, with only an additional logarithmic dependence on the number S of unit vectors. In particular, a single embedding of $\boldsymbol{x}$ can be reused to approximate several quantities with different unit vectors.

3.4 A kernel interpretation of the estimator

The estimation result of Proposition 3.2 has a natural interpretation in terms of kernel approximation. Recall that, for a fixed unit vector $\boldsymbol{u}$, the estimator

$$\tfrac{\pi}{4m}\langle \operatorname{sign}(\psi^-(\boldsymbol{u})), \psi^-(\boldsymbol{x})\rangle$$

provides an approximation of $\langle \boldsymbol{u}, \boldsymbol{x}\rangle^2$ with controlled distortion, uniformly over $\boldsymbol{x} \in \Sigma_k$.

Since the function $\kappa^S(\boldsymbol{x}, \boldsymbol{u}) := \langle \boldsymbol{x}, \boldsymbol{u}\rangle^2$ defines a quadratic polynomial kernel (more specifically a dot-product kernel of the form $f(\langle \boldsymbol{x}, \boldsymbol{y}\rangle)$ with $f(x) = x^2$, [123]), this result shows that the interaction between $\boldsymbol{x}$

and $\boldsymbol{u}$ can be approximated directly from their embeddings. In this sense, the estimator can be interpreted as a random feature approximation of the kernel κ^{S} in the same spirit as what can be found in [125] and [135], with features constructed from quadratic measurements.

More precisely, defining the mappings

$$\psi_1(\boldsymbol{x}) = \psi^-(\boldsymbol{x}), \qquad \psi_2(\boldsymbol{u}) = \operatorname{sign}(\psi^-(\boldsymbol{u})),$$

we can rewrite the estimator as

$$\tfrac{\pi}{4m}\langle \psi_2(\boldsymbol{u}), \psi_1(\boldsymbol{x})\rangle.$$

Prop. 3.2 then states that, for a fixed vector $\boldsymbol{u}$, this quantity approximates $\kappa^{\mathrm{S}}(\boldsymbol{x}, \boldsymbol{u})$ uniformly over Σ_k, provided that the number of measurements m is sufficiently large.

This construction differs from classical random feature methods found in [125], where a single feature map is used symmetrically for both arguments. Here, two distinct embeddings are used, leading to an *asymmetric* feature representation. This is in line with the framework of asymmetric random features introduced in [135], where kernel approximations are obtained from the inner product of two different embeddings, for instance when one side undergoes quantisation.

In our setting, this asymmetry arises naturally from the use of the sign function applied to the embedding of $\boldsymbol{u}$, while $\boldsymbol{x}$ remains unquantised. The resulting quantity is not, in general, a kernel in the strict sense, since it is neither symmetric nor guaranteed to be PSD. However, it provides a consistent approximation of the quadratic kernel and can therefore be used as an asymmetric similarity measure for tasks such as detection or classification. Note that the target kernel $\kappa^{\mathrm{S}}(\boldsymbol{x}, \boldsymbol{u}) = \langle \boldsymbol{x}, \boldsymbol{u}\rangle^2$ is symmetric and PSD, but its asymmetric approximation does not directly define a standard SVM kernel. For a standard SVM, the similarity matrix used during training should correspond to a symmetric PSD kernel. The asymmetric SPE approximation does not satisfy these properties. Although it can be symmetrised, this alone does not guarantee positive semidefiniteness. It should therefore be interpreted primarily as an approximation of the true quadratic kernel, rather than as a standard SVM kernel.

For comparison, we can also consider using the ψ^- for both sides without the sign and find the associated approximation

$$\tfrac{1}{m}\langle \psi^-(\boldsymbol{x}), \psi^-(\boldsymbol{u})\rangle.$$

This quantity is symmetric and PSD by construction. However, it is not the object controlled by Prop. 3.2. The fully real-valued symmetric

construction is closer in spirit to the unbounded rank-one embeddings studied later in Chap. 4. There, we will see that such unbounded random features have heavier-tailed behaviour, which motivates the use of bounded transformations when uniform approximation guarantees are desired.

Corollary 3.2 (Kernel approximation in the SPE domain)**.** *Let* $\{\boldsymbol{u}_s\}_{s=1}^{S} \subset \mathbb{R}^n$ *be a fixed set of unit vectors, and define the quadratic kernel*

$$\kappa^{\mathrm{S}}(\boldsymbol{x}, \boldsymbol{u}_s) := \langle \boldsymbol{x}, \boldsymbol{u}_s \rangle^2.$$

For $s \in [S]$*, define the asymmetric empirical similarity score*

$$\widehat{\kappa}^{\mathrm{S}}(\boldsymbol{x}, \boldsymbol{u}_s) := \tfrac{\pi}{4m} \langle \operatorname{sign}(\psi^-(\boldsymbol{u}_s)), \psi^-(\boldsymbol{x}) \rangle.$$

Under the assumptions of Corollary 3.1, with probability at least $1 - C\exp(-c\delta^2 m)$*,*

$$\sup_{\boldsymbol{x} \in \Sigma_k} \max_{s \in [S]} \left| \widehat{\kappa}^{\mathrm{S}}(\boldsymbol{x}, \boldsymbol{u}_s) - \kappa^{\mathrm{S}}(\boldsymbol{x}, \boldsymbol{u}_s) \right| \leq \delta \|\boldsymbol{x}\|^2.$$

Proof. This is exactly Corollary 3.1 after defining $\kappa^{\mathrm{S}}(\boldsymbol{x}, \boldsymbol{u}_s) = \langle \boldsymbol{x}, \boldsymbol{u}_s \rangle^2$ and $\widehat{\kappa}^{\mathrm{S}}(\boldsymbol{x}, \boldsymbol{u}_s) = \frac{\pi}{4m} \langle \operatorname{sign}(\psi^-(\boldsymbol{u}_s)), \psi^-(\boldsymbol{x}) \rangle$. □

It is important to stress that Cor. 3.2 does not provide a uniform approximation of $\kappa^{\mathrm{S}}(\boldsymbol{x}, \boldsymbol{u})$ over all possible pairs $(\boldsymbol{x}, \boldsymbol{u})$. The unit vectors $\boldsymbol{u}_s$ are fixed in advance, and the guarantee holds uniformly over $\boldsymbol{x} \in \Sigma_k$ and over this finite collection of references. Thus, the result is best understood as a finite-query kernel approximation statement. This is sufficient for settings where we compare embedded signals with a predetermined finite set of vectors, but it should not be read as a full random-feature approximation of κ^{S} over the whole ambient space.

3.5 Experiments

We now illustrate the behaviour of the SPE estimator introduced in Prop. 3.2. Throughout this section the reader should keep in mind that the goal of these experiments is not to optimise performance metrics of a given method, but rather to verify that the quantities appearing in Eq. 3.3 can be used in practice to perform simple signal processing tasks only using the data transform with the ψ^- embedding. We therefore focus on four questions. First, we check whether the distortion term δ decreases with m as predicted by the theory. Second, we

test whether the SPE estimate can replace direct scalar-product comparisons in a simple classification task on the MNIST dataset. We then show that event detection in the embedding domain is also possible by embedding frames of a video of a moving object and using those embeddings to detect when the object moves from one part of a frame to another. Finally we also show that the induced kernel interpretation creates an approximation that can be used for basic classification tasks.

3.5.1 Empirical distortion of the SPE estimator

We first study the distortion term δ from Eq. 3.3. Recall that, for a unit vector $\boldsymbol{u}$ and a signal $\boldsymbol{x}$, the SPE estimator is given by

$$g(\boldsymbol{u}, \boldsymbol{x}) = \tfrac{\pi}{4m} \langle \operatorname{sign}(\psi^-(\boldsymbol{u})), \psi^-(\boldsymbol{x}) \rangle,$$

with the goal to approximate $\langle \boldsymbol{u}, \boldsymbol{x} \rangle^2$. Since the error in Prop. 3.2 is controlled by $\delta \|\boldsymbol{x}\|^2$, we evaluate the normalised error

$$\tfrac{|g(\boldsymbol{u},\boldsymbol{x}) - \langle \boldsymbol{u},\boldsymbol{x} \rangle^2|}{\|\boldsymbol{x}\|^2}.$$

In particular, when $\boldsymbol{u}$ is orthogonal to $\boldsymbol{x}$, the target quantity $\langle \boldsymbol{u}, \boldsymbol{x} \rangle^2$ vanishes. In this case, the residual value of the estimator directly measures the distortion level:

$$\widehat{\delta} = \tfrac{1}{\|\boldsymbol{x}\|^2} \left| \tfrac{\pi}{4m} \langle \operatorname{sign}(\psi^-(\boldsymbol{u})), \psi^-(\boldsymbol{x}) \rangle \right|.$$

In Fig. 3.2, we show this quantity as a function of the *oversampling ratio* $\rho := m/n$, where m is the number of embedding dimensions and n is the ambient dimension of the original signal. This ratio measures the size of the embedding relative to the size of the input. Values $\rho < 1$ correspond to a compressed representation, values $\rho \simeq 1$ correspond to an embedding of comparable dimension, and values $m/n > 1$ correspond to an overcomplete representation. Plotting the error against ρ therefore indicates whether the estimator can be useful for dimensionality reduction purposes.

For each value of m, we repeat the experiment over 300 independent trials. In each trial, we draw a random unit vector $\boldsymbol{u}$ uniformly on the sphere $\mathbb{S}^{n-1}$. We then draw a second random vector $\boldsymbol{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_n)$ and set

$$\boldsymbol{x} = \boldsymbol{z} - \langle \boldsymbol{z}, \boldsymbol{u} \rangle \boldsymbol{u},$$

to ensure that $\boldsymbol{x}$ is orthogonal to $\boldsymbol{u}$. This matches the form of Prop. 3.2, where $\boldsymbol{u}$ is supposed to have unit norm and the error is proportional to

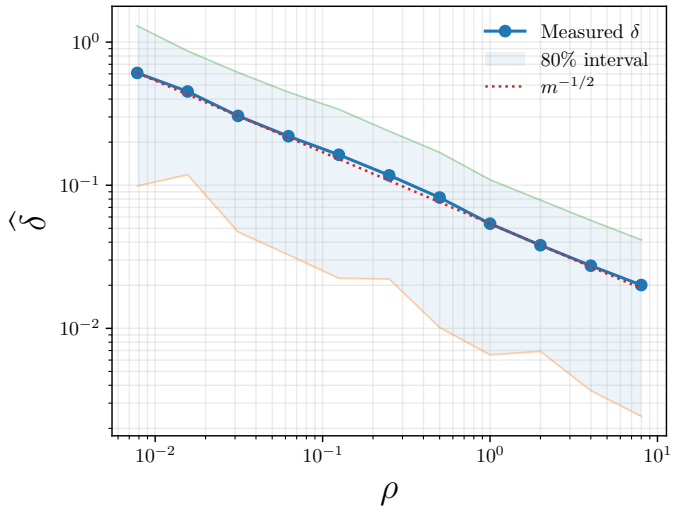


Fig. 3.2 Empirical distortion of the SPE estimator when $\boldsymbol{u}$ is orthogonal to $\boldsymbol{x}$. The curve is in logarithmic scale and shows the mean value of $\widehat{\delta}$ over several random trials, with the central percentile range displayed as a shaded region. The dotted line indicates an $m^{-1/2}$ slope.

$\|\boldsymbol{x}\|^2$. In the experiment, we fix the ambient dimension to $n = 512$ and test embedding dimensions

$$m \in \{2^2, 2^3, \ldots, 2^{12}\}.$$

For each trial, a new independent version of the debiased embedding ψ^- is generated, and we compute

$$\widehat{\delta} = \frac{1}{\|\boldsymbol{x}\|^2} \left| \frac{\pi}{4m} \langle \operatorname{sign}(\psi^-(\boldsymbol{u})), \psi^-(\boldsymbol{x}) \rangle \right|.$$

Since $\langle \boldsymbol{u}, \boldsymbol{x} \rangle = 0$ by construction, this quantity directly measures the relative SPE error. The main curve in Fig. 3.2 shows the average of $\widehat{\delta}$ over the independent trials, along with a shaded band including 80% of the results. The observed decay follows the expected $m^{-1/2}$ behaviour, in agreement with the scaling

$$\delta = \mathcal{O}\left(\sqrt{\frac{k}{m}}\right),$$

up to logarithmic factors. This confirms that the approximation function g becomes more accurate as the embedding dimension increases at the rate predicted by theory.

Fig. 3.3 Illustration of MNIST samples, hand-drawn digits from 0 to 9

3.5.2 MNIST classification in the embedding domain

We next test whether the SPE estimator can be used to perform a simple classification task directly from the instances embedded with ψ^-. We consider the MNIST dataset [57], which contains images of hand-drawn digits from 0 to 9 (see Fig. 3.3). Each image will be represented as a vector $\boldsymbol{x} \in \mathbb{R}^{784}$. The full dataset contains 70 000 images split into $N_{\text{tr}} = 60\,000$ training images and $N_{\text{te}} = 10\,000$ test images. We denote the training set by $\{(\boldsymbol{x}_i, t_i)\}_{i=1}^{N_{\text{tr}}}$, where $t_i \in \{0, \dots, 9\}$ is the label of $\boldsymbol{x}_i$. From the training set, we compute the centroid

$$\boldsymbol{c}_j := \tfrac{1}{|K_j|} \textstyle\sum_{i \in K_j} \boldsymbol{x}_i, \qquad K_j := \{i : \ t_i = j\},$$

of each digit class $j \in \{0, \dots, 9\}$, and define the corresponding query vector

$$\boldsymbol{u}_j = \tfrac{\boldsymbol{c}_j}{\|\boldsymbol{c}_j\|}.$$

In the direct domain (*i.e.*, without embedding), a test image $\boldsymbol{x}$ is classified according to

$$\hat{t}_{\text{d}}(\boldsymbol{x}) = \arg\max_j \langle \boldsymbol{u}_j, \boldsymbol{x} \rangle^2.$$

This simple direct centroid classifier provides the basic accuracy of 81.95% on the test instances.

The direct classifier still assumes that we have access to the test images $\boldsymbol{x}$ in the original domain. We now consider the setting where a new test instance is only available through its debiased quadratic embedding $\psi^-(\boldsymbol{x})$. This is the situation of interest in this chapter: the signal has already been transformed by ψ^-, and we want to classify it without reconstructing $\boldsymbol{x}$.

For each class centroid, we can compute the corresponding embedding $\psi^-(\boldsymbol{u}_j)$ once. Then, using Prop. 3.2, the direct score $\langle \boldsymbol{u}_j, \boldsymbol{x} \rangle^2$ can be replaced by its estimate in the embedding domain. This gives the SPE classifier

$$\hat{t}_{\text{SPE}}(\boldsymbol{x}) = \arg\max_j \tfrac{\pi}{4m} \langle \operatorname{sign}(\psi^-(\boldsymbol{u}_j)), \psi^-(\boldsymbol{x}) \rangle.$$

This classification process is summarised in Fig. 3.4. This classification method is directly aligned with Proposition 3.2, since the class centroids

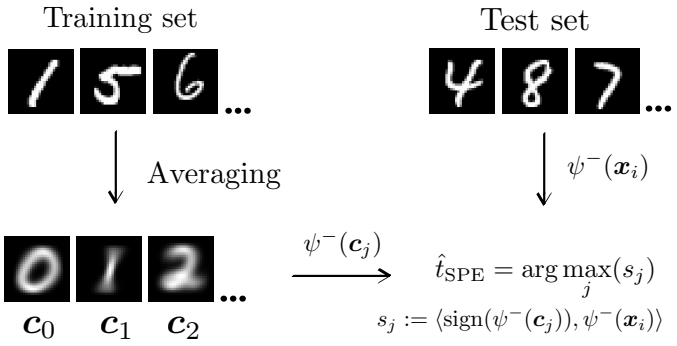


Fig. 3.4 Process of classification of embedded test data using the SPE

are still defined in the original signal domain, while each comparison with the test image is performed from $\psi^-(\boldsymbol{x})$ alone. In other words, once the class centroids have been embedded, an incoming embedded instance can be classified without accessing or reconstructing its original pixel representation. The resulting test accuracies are shown in Fig. 3.5, as a function of the oversampling ratio $\rho = m/n$. As expected, the classification accuracy improves with the embedding dimension and progressively approaches the accuracy of classification in the original domain.

We also test two classifiers trained directly in the embedding domain. For each class j, we compute an embedded centroid

$$\boldsymbol{c}_j^{\psi^-} = \tfrac{1}{N_j} \textstyle\sum_{i:t_i=j} \psi^-(\boldsymbol{x}_i),$$

where N_j is the number of training samples in class j. The first embedded-centroid classifier uses the rule

$$\hat{t}_{\psi^-}(\boldsymbol{x}) = \arg\max_j \langle \boldsymbol{c}_j^{\psi^-}, \psi^-(\boldsymbol{x})\rangle.$$

The second version keeps only the sign of the embedded centroids:

$$\hat{t}^{\mathrm{s}}_{\psi^-}(\boldsymbol{x}) = \arg\max_j \langle \mathrm{sign}(\boldsymbol{c}_j^{\psi^-}), \psi^-(\boldsymbol{x})\rangle.$$

This last rule is not a direct application of Prop. 3.2, since $\boldsymbol{c}_j^{\psi^-}$ is an empirical centroid in the embedding domain rather than the embedding of a known vector $\boldsymbol{u}_j$. However, it is motivated by the same stabilising role of the sign function. Since raw quadratic embeddings can have heavy-tailed coordinates (see Sec. 4.2.1 for more details), replacing an embedded centroid by its sign will reduce the influence of unusually large coordinates.

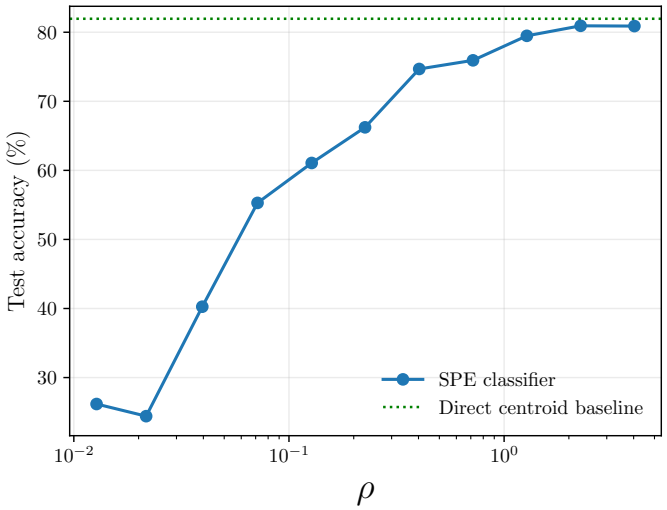


Fig. 3.5 Test accuracy of the SPE classifier on MNIST as a function of the oversampling ratio $\rho = m/n$. The dotted horizontal line indicates the direct centroid baseline in the original signal domain.

The results are shown in Fig. 3.6. The direct centroid classifier reaches an accuracy of 81.95% and is shown as a reference baseline. Both embedded-centroid classifiers are trained directly in the embedding domain, using class centroids computed from the embedded training samples $\psi^{-}(\boldsymbol{x})$. The unsigned version stays below the direct accuracy, suggesting that raw embedded centroids are not by themselves a stable replacement for the direct class centroids. By contrast, applying the sign operation to the embedded centroids leads to an improvement of up to 5% compared to the centroid classifier, at the cost of a higher ρ. This suggests that the sign operation acts as a useful stabilisation mechanism in the quadratic embedding domain.

Effect of sparse orthonormal representations

The previous MNIST experiment was performed directly in the pixel domain, without enforcing any explicit sparse model. However, Proposition 3.2 is stated for vectors in Σ_k, and the preceding discussion shows that the same result applies to vectors that are sparse in any orthonormal basis, by rotational invariance of Gaussian measurements. It is therefore natural to ask whether representing MNIST images in a sparse orthonormal basis improves the measurement-accuracy tradeoff.

To test this, we transform each image into an orthonormal discrete cosine basis and keep only the k largest coefficients in magnitude. No

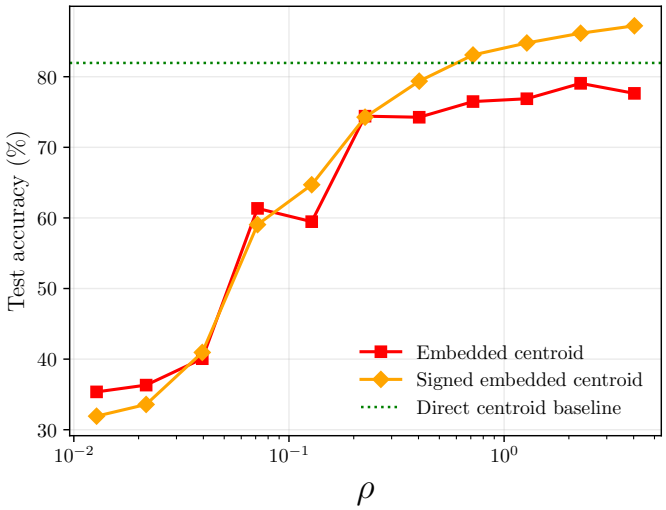


Fig. 3.6 MNIST classification accuracy of embedded-centroid classifiers as a function of the oversampling ratio $\rho = m/n$. The direct centroid classifier is shown as a horizontal reference line. The embedded-centroid curves correspond to classifiers trained directly from $\psi^-(\boldsymbol{x})$, with and without applying the sign operation to the embedded class centroids.

reconstruction is performed and the classifiers are applied directly to the truncated coefficient vectors. This gives exactly k-sparse coefficient vectors while keeping the same SPE methodology as before. However, in this experiment sparsity is imposed by truncation rather than being an intrinsic property of the original data.

The direct sparse-DCT centroid classifier remains accurate even for small values of k. For instance, using only $k = 50$ coefficients yields a direct accuracy of 81.19%, close to the full pixel-domain direct accuracy of 81.95%. With $k = 100$ coefficients, the direct sparse-DCT baseline reaches 82.06%. This confirms that the images are well represented by a small number of DCT coefficients for this particular centroid classification task.

The SPE classifier also benefits mildly from this representation. For example, in the pixel domain, an accuracy of 79.48% is obtained at $m = 1000$, corresponding to $\rho \simeq 1.28$. In the sparse-DCT representation with $k = 200$, the same accuracy is reached at $m = 800$, corresponding to $\rho \simeq 1.02$. The improvement is not dramatic, but it supports the expected trend that a representation closer to the sparse model can reduce the number of measurements required to reach a comparable accuracy.

To further study this experiment with data that actually satisfies the sparse model by construction, we consider another synthetic clas-

sification problem in the same ambient dimension $n = 784$. We generate ten classes of vectors of unit norm with support on a fixed set of k entries, with $k \in \{10, 25, 50, 100, 200\}$. For each sample of class $c \in \{1, \dots, 10\}$, one of the first ten active entries is associated with the class and receives a unit contribution, while the other entries are filled with Gaussian noise. More precisely, the c^{th} active entry contains this unit contribution in addition to a Gaussian perturbation, whereas the other nine contain only Gaussian perturbations. When $k > 10$, the remaining $k - 10$ active entries are also filled with independent Gaussian noise. The resulting vector is finally normalised to unit norm. We generate 500 training and 200 test samples per class and apply the same direct and SPE centroid classifiers as above. For the SPE classifier, the accuracies are averaged over five independent realisations of the random embeddings.

The direct classifier achieves between 96.55% and 97.15% accuracy across the different sparsity levels. In the sparsest case, $k = 10$, the SPE classifier reaches 95.91% accuracy with only $m = 400$ measurements, corresponding to $\rho \simeq 0.51$, and 96.17% with $m = 500$, corresponding to $\rho \simeq 0.64$, compared with a direct accuracy of 97.15%. Thus, in this genuinely sparse setting, the SPE classifier approaches the direct-domain performance while using fewer measurements than the ambient dimension. The effect is less pronounced as k increases. For instance, at $m = 400$, the accuracy decreases to approximately 91–92% for $k \geq 25$, despite the direct accuracy remaining close to 97%. This experiment therefore shows that it is possible to go below $\rho = 1$ when we are in a sparse enough regime.

Overall, these experiments support the main claims of this chapter. The orthogonal vector experiment validates the expected distortion scaling of the SPE estimator, while the MNIST experiment shows that this estimator can be used to transfer a simple classification rule to the embedding domain. The sparse DCT experiment shows a moderate benefit from imposing a sparse representation on MNIST. Finally, the synthetic experiment illustrates the regime described by Prop. 3.2 more directly. When the signals are genuinely sparse, accurate classification can be obtained with an embedding dimension smaller than the ambient dimension, although this advantage becomes less visible as the sparsity level increases.

3.5.3 Event detection in a synthetic image sequence

We next consider a simple event detection task. The goal is to illustrate how the SPE estimator can recover localised information from the

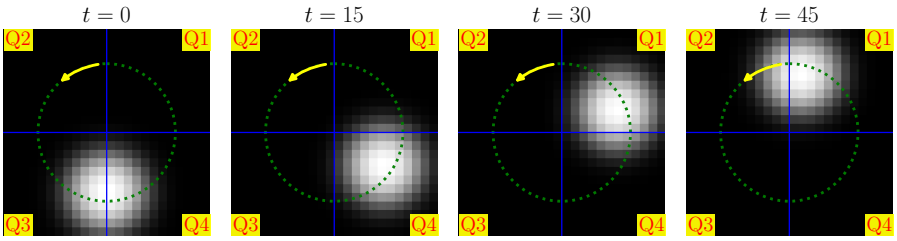


Fig. 3.7 Example frames from the revolving disk sequence used for the quadrant detection experiment. The blue lines show the four quadrant regions, while the dotted curve and arrow show the circular trajectory followed by the disk.

debiased quadratic embedding of an image sequence, without reconstructing the frames themselves.

We generate a synthetic sequence $\{\boldsymbol{x}_t\}_{t=0}^{T-1}$ of $T = 96$ images of size 24×24, hence with ambient dimension $n = 576$. Each frame contains a blurry white disk moving on a circular trajectory over a black background. An example frame is shown in Fig. 3.7.

We define four unit vectors $\{\boldsymbol{u}_j\}_{j=1}^{4}$, each corresponding to one quadrant of the image. More precisely, $\boldsymbol{u}_j$ is the normalised indicator of the j-th quadrant. For each frame $\boldsymbol{x}_t$, the true quadrant occupancy is defined as

$$q_j(t) := \langle \boldsymbol{u}_j, \boldsymbol{x}_t \rangle^2.$$

This quantity measures the squared intensity of the frame over quadrant j. According to Prop. 3.2, it can be estimated from the embedding $\psi^-(\boldsymbol{x}_t)$ through

$$g_j(t) = \tfrac{\pi}{4m} \langle \operatorname{sign}(\psi^-(\boldsymbol{u}_j)), \psi^-(\boldsymbol{x}_t) \rangle.$$

Thus, after computing and storing the signed embedded quadrant patterns $\operatorname{sign}(\psi^-(\boldsymbol{u}_j))$, every incoming frame can be processed directly in the embedding domain.

The experiment is shown in Fig. 3.8. We use $m = 170$ measurements, corresponding to $\rho \simeq 0.3$. The true occupancy curves $q_j(t)$ and their SPE estimates $g_j(t)$ are normalised before plotting. The estimated curves follow roughly the expected curves of the four quadrants and reproduce the passage of the disk through the image.

As a simple discrete detection criterion, we also assign each frame to the quadrant with largest estimated occupancy,

$$\widehat{o}(t) = \arg\max_j g_j(t),$$

and compare it with the true maximiser

$$o(t) = \arg\max_j q_j(t).$$

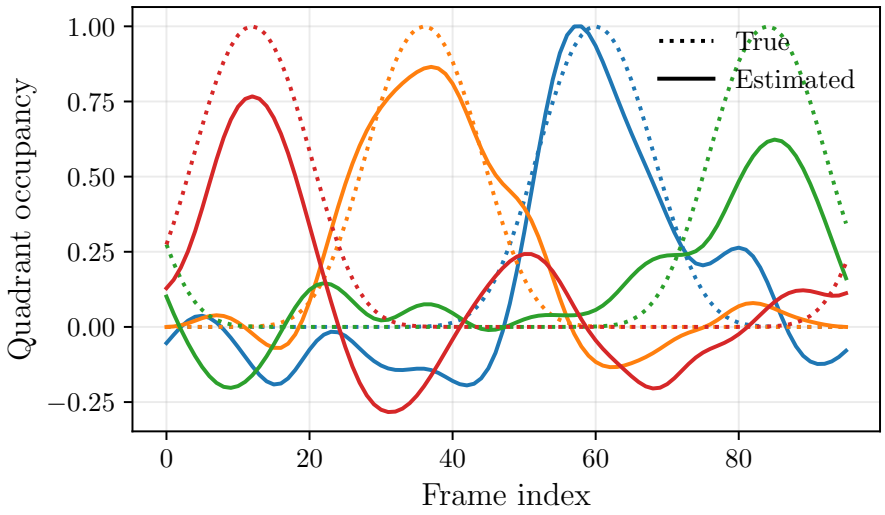


Fig. 3.8 Quadrant occupancy detection from debiased quadratic embeddings. Dashed curves show the true occupancies $q_j(t) = \langle \boldsymbol{u}_j, \boldsymbol{x}_t \rangle^2$, while solid curves show the SPE estimates $g_j(t) = \frac{\kappa}{m} \langle \operatorname{sign}(\psi^-(\boldsymbol{u}_j)), \psi^-(\boldsymbol{x}_t) \rangle$. The experiment uses 24×24 images, $T = 96$ frames, and $m = 170$ measurements.

This gives an argmax detection accuracy of 94.8%. A closer look shows that the incorrect detections occur mainly near transitions between nearby quadrants. In these frames, two true occupancy values are close, so a small perturbation of the estimated curves can change the value of $o(t)$. In addition, the predicted quadrant is typically the next quadrant along the trajectory, corresponding to a slight shift of the transition rather than a complete localisation failure.

This experiment should not be interpreted as an optimised compression scheme for such a synthetic video, which might have a lower intrinsic complexity. However, it provides an illustration of the mechanism behind Prop. 3.2, *i.e.*, localised quantities of the form $\langle \boldsymbol{u}_j, \boldsymbol{x}_t \rangle^2$ can be estimated directly from $\psi^-(\boldsymbol{x}_t)$, without reconstructing the image and without applying an inverse of the quadratic operator. Combined with the distortion experiment in Fig. 3.2, this confirms that the SPE estimate can be used as a practical signal processing tool directly in the embedding domain.

3.5.4 Kernel interpretation

We now test whether the kernel interpretation of Prop. 3.2 can be used for a simple classification task. The goal is to compare a classifier using the exact quadratic kernel $\kappa^{\mathrm{S}}(\boldsymbol{x}, \boldsymbol{y}) = \langle \boldsymbol{x}, \boldsymbol{y} \rangle^2$ with the same classifier

when its kernel evaluations are replaced by its approximation $\widehat{\kappa}^{\mathrm{S}}$.

We generate a synthetic binary classification dataset in $\mathbb{R}^{512}$. First, we draw two random unit vectors $\boldsymbol{c}_1, \boldsymbol{c}_2 \in \mathbb{R}^{512}$ and use them to define the centres of the two classes. To generate a new sample, we first choose a class label $t \in \{1, 2\}$ uniformly at random. We then choose a sign $\sigma \in \{-1, 1\}$ uniformly at random and start from the point $\sigma \boldsymbol{c}_t$.

A random perturbation is then added to create a cloud around this point. More precisely, we draw an independent Gaussian vector $\boldsymbol{\xi} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_n)$ for every sample. Thus, each sample is of the form

$$\boldsymbol{x} = \frac{\sigma \boldsymbol{c}_t + \boldsymbol{\xi}}{\|\sigma \boldsymbol{c}_t + \boldsymbol{\xi}\|}.$$

The random sign makes the problem invariant under the transformation $\boldsymbol{x} \mapsto -\boldsymbol{x}$. As a result, the quadratic kernel

$$\kappa^{\mathrm{S}}(\boldsymbol{x}, \boldsymbol{y}) = \langle \boldsymbol{x}, \boldsymbol{y} \rangle^2$$

is naturally well adapted to the task, since it will not see the difference between two vectors that differ only by their sign.

We create 400 training samples and 200 test samples. As a reference method, we train a binary SVM using the exact quadratic kernel Gram matrix

$$K_{ij} = \kappa^{\mathrm{S}}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle^2$$

for $0 \leq i \leq j < 400$, on the training set. A test sample $\boldsymbol{x}$ is then classified from the support-vector decision function (from Eq. 2.3)

$$f(\boldsymbol{x}) = \textstyle\sum_{s \in \mathcal{I}_{\mathrm{sv}}} \alpha_s y_s \kappa^{\mathrm{S}}(\boldsymbol{x}, \boldsymbol{x}_s) + b,$$

where $\mathcal{I}_{\mathrm{sv}}$ is the set of support-vector indices selected by the exact quadratic-kernel SVM.

We then keep the same trained coefficients α_s, labels y_s, and $\mathcal{I}_{\mathrm{sv}}$, but replace every exact kernel evaluation $\kappa^{\mathrm{S}}(\boldsymbol{x}, \boldsymbol{x}_s)$ by

$$\widehat{\kappa}^{\mathrm{S}}(\boldsymbol{x}, \boldsymbol{x}_s) = \tfrac{\pi}{4m} \langle \psi^-(\boldsymbol{x}), \operatorname{sign}(\psi^-(\boldsymbol{x}_s)) \rangle.$$

This gives the approximate decision function

$$\hat{f}_m(\boldsymbol{x}) = \textstyle\sum_{s \in \mathcal{I}_{\mathrm{sv}}} \alpha_s y_s \widehat{\kappa}^{\mathrm{S}}(\boldsymbol{x}, \boldsymbol{x}_s) + b.$$

In this second classifier, the support vectors are stored through the binary vectors $\operatorname{sign}(\psi^-(\boldsymbol{x}_s))$, while each incoming test sample is represented by its embedding $\psi^-(\boldsymbol{x})$.

We repeated this experiment for values of m varying from 1 to 100 on a logarithmic scale, with 20 trials for each value of m. The results

are shown in Fig. 3.9. The solid curve shows the test accuracy obtained when κ^{S} is replaced by its approximation $\widehat{\kappa}^{\mathrm{S}}$. The dashed horizontal line gives the test accuracy of the exact SVM. The curve shows the mean value over the trials, with the shaded region corresponding to one standard deviation. As m increases, the approximate classifier improves and eventually reaches 100% test accuracy.

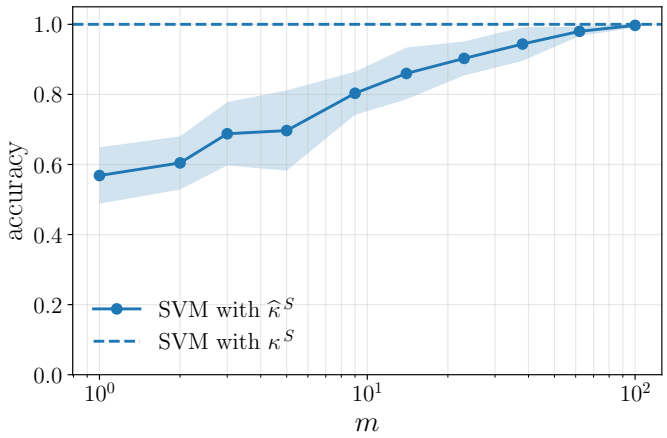


Fig. 3.9 Approximate support-vector prediction with $\widehat{\kappa}^{\mathrm{S}}$. The solid curve shows the test accuracy obtained by replacing κ^{S} in the support-vector decision function by its approximation $\widehat{\kappa}^{\mathrm{S}}$. The dashed line shows the test accuracy of the exact quadratic-kernel SVM.

Compared with linear embeddings, the quadratic setting considered in this chapter preserves a different notion of similarity. In the linear case, the corresponding SPE estimates the signed scalar product $\langle \boldsymbol{u}, \boldsymbol{x} \rangle$, whereas the quadratic embedding considered here naturally approximates $\langle \boldsymbol{u}, \boldsymbol{x} \rangle^2$. Linear embeddings thus keep directional information that is necessarily lost by quadratic measurements, since $\boldsymbol{x}$ and $-\boldsymbol{x}$ have the same quadratic representation. They are also generally simpler to analyse and use when linear measurements are available and the task only depends on linear geometry. The interest of the present approach is instead to show that, when the available measurements are intrinsically quadratic, useful estimation and classification tasks can still be carried out directly in the embedding domain without reconstructing the original signal.

3.6 Conclusion

In this chapter, we studied signal processing tasks performed directly from random quadratic embeddings. Starting from rank-one projec-

tions, we introduced the debiased embedding ψ^- in order to remove the constant bias present in the symmetric quadratic measurements. We then showed that this debiased construction can be used to estimate useful quantities without reconstructing the original signal.

The main result of the chapter is given in Prop. 3.2. It states that given a unit vector $\boldsymbol{u}$, the quantity obtained by comparing $\psi^-(\boldsymbol{x})$ with $\operatorname{sign}(\psi^-(\boldsymbol{u}))$ approximates $\langle \boldsymbol{u}, \boldsymbol{x} \rangle^2$ uniformly over sparse signals, provided that the number of measurements is large enough compared to the sparsity level. This gives a direct way to process embedded signals by precomputing signed embeddings of unit vectors and then applying simple scalar products in the embedding domain. The finite-query version given in Cor. 3.1 further shows that several such quantities can be estimated simultaneously, at the price of only a logarithmic dependence on the number of unit vectors.

Finally, we interpreted the estimator of Prop. 3.2 as an asymmetric random feature approximation of the quadratic polynomial kernel $\kappa^{\mathrm{S}}(\boldsymbol{x}, \boldsymbol{u}) = \langle \boldsymbol{x}, \boldsymbol{u} \rangle^2$. This interpretation shows that the same construction can be used not only for signal estimation, but also for simple kernel-based prediction.

We backed up our claims with experiments both evaluating the quality of the theoretical guarantees presented, and the usability of our approximation in experimental settings. We performed classification and detection as well as a kernel-based experiment, every time showing that the higher the embedding dimension, the better accuracy was achieved.

Overall this chapter demonstrated that randomly quadratically embedded data could be used directly in the new representation in order to perform simple signal processing tasks. The results also show that this idea is not limited to a single change of representation since both the real-valued embedding and its signed variant can preserve enough information to support useful computations after embedding.

The next chapter moves from vector-valued data to subspaces, where random embeddings will be used to approximate kernels on the Grassmannian manifold.

4 Approximating Grassmannian kernels with random features

4.1 Introduction

In the previous chapter we studied random quadratic embeddings of *vectors* and showed that useful quantities can be estimated directly in the embedding domain. We now move from individual signals to *subspaces*. Instead of representing a data point by a single vector $\boldsymbol{x} \in \mathbb{R}^n$, we represent it by a k-dimensional linear subspace of $\mathbb{R}^n$ as presented in Sec. 2.3. This setting occurs when the information is not contained in a single observation, but in the span generated by several observations of the same object.

The goal of this chapter is to develop random feature approximations of Grassmannian kernels. More precisely, we construct random embeddings of elements of the *Grassmannian* manifold such that Euclidean inner products between their embeddings approximate these kernels.

4.1.1 Motivation

There exist many supervised and unsupervised learning tasks where data classes or clusters are better represented by low-dimensional *subspaces* spanned by the instances of a specific class of data. This assumption arises naturally in a range of contexts such as image and video processing [154, 18, 129, 108, 84], wireless communications [137], and dynamic subspace modelling in signal processing [140, 132, 88]. For instance, a collection of images of the same object under different view-

points or illumination conditions can be represented by the subspace spanned by these images. Similarly, the frames of a video, or features extracted from them, can span a low-dimensional subspace representing the complete sequence, allowing videos to be compared or classified through their corresponding subspaces. In this setting, data is naturally modelled as elements of the Grassmannian manifold $\mathcal{G}(k, n)$, the set of all k-dimensional subspaces of $\mathbb{R}^n$.

As discussed in Secs. 2.1 and 2.4, kernel machines such as SVMs and kernel regression use pairwise similarities to learn non-linear classification boundaries [9]. For subspace-valued data, these similarities can be defined using Grassmannian kernels [75, 74, 157]. Their main limitation is the usual cost of kernel methods. For a dataset containing N subspaces, computing and storing the $N \times N$ Gram matrix has a cost that grows quadratically with N.

Methods such as the Nyström approximation reduce the cost of handling this matrix by constructing a lower-rank approximation from a subset of the data [62, 10]. This reduces the storage costs, however these methods still require computing a large number of pairwise similarities and remain data-dependent. In particular, the approximation is constructed from a subset of the dataset itself, so that the resulting representation depends on the choice of these samples and still requires kernel evaluations when processing new data. Random features instead replace each subspace with a finite-dimensional random vector whose scalar products approximate the chosen kernel [125]. This avoids constructing the full Gram matrix and allows learning methods to operate directly on these random representations. The goal is thus to develop such approximations for Grassmannian kernels.

A direct strategy consists in vectorising the projection matrices and applying standard random projections, as suggested by the compressive sensing literature [67, 35]. While this yields valid random feature constructions, it comes at a high computational and memory cost due to the ambient dimension n^2 of the projector representation.

An alternative is to rely on more structured measurements, such as rank-one projections of the form $\boldsymbol{a}^\top \boldsymbol{P}\boldsymbol{b}$, which significantly reduce the computational burden. These constructions provide unbiased estimators of classical Grassmannian kernels, but they suffer from poor concentration properties due to their heavy-tailed nature. As a result, obtaining accurate approximations requires a large number of features, limiting their practical usefulness.

The goal of this chapter is to design random feature maps for Grassmannian data that remain computationally efficient while providing accurate approximations of meaningful kernels. This requires construc-

tions that both respect the geometry of subspaces and have sufficiently strong concentration behaviour.

4.1.2 Related works

Early supervised classification methods for subspace-valued data relied on nearest-subspace algorithms, notably the CLAFIC (ClAss-Featuring Information Compression) approach introduced in 1967 by S. Watanabe and others in [153, 154], which represents each class by a reference subspace and assigns labels by projecting new instances into the reference subspaces. More recent works embed subspaces into Euclidean spaces allowing efficient nearest-neighbour search. In particular, [17] and [84] propose deterministic and binary embeddings based on vectorised projection matrices, approximately preserving the projection distance and enabling scalable classification.

Another related line of work is about Euclidean distance geometry and the recovery of Gram matrices from partial distance measurements [59, 145]. In this setting, a low-rank positive-semidefinite matrix such as $\boldsymbol{P} = \boldsymbol{U}\boldsymbol{U}^\top$ can be interpreted as a Gram matrix, and measurements of the form $(\boldsymbol{e}_i - \boldsymbol{e}_j)^\top \boldsymbol{P}(\boldsymbol{e}_i - \boldsymbol{e}_j)$ correspond to squared distances between embedded points. This is related to our use of measurements of projection matrices, but with a different objective. They aim to recover the matrix $\boldsymbol{P}$ itself from limited measurements, whereas we aim to build random feature maps whose inner products approximate a Grassmannian kernel. Hence, bounded non-linear functions are not introduced for projector recovery, but to obtain kernel estimators with controlled concentration.

Beyond distance-based methods, Grassmannian kernel methods provide a good alternative for learning from subspace data. Classical kernels defined through principal angles have been extensively studied in [75]. While effective, these kernels suffer from poor scalability, as kernel-based learning requires forming and storing an $N \times N$ Gram matrix, which quickly becomes prohibitive for large datasets [125]. The reader can refer to Sec. 2.4 where we recall essential information on Grassmannian kernels.

Several works have addressed this limitation through randomised representations. A first line of work applies random projections to vectorised projectors, relying on the JL lemma to provide guarantees for controlling distortion [17, 84]. A complementary approach, analysed in [105], directly projects subspaces using dense Gaussian operators to embed $\mathcal{G}(k, n)$ into a lower-dimensional Grassmannian $\mathcal{G}(k, m)$ while approximately preserving pairwise distances. Although theoretically

appealing, these approaches rely on dense random matrices and incur $\mathcal{O}(n^2)$ storage or computation costs, limiting their applicability in high dimensions. This will be developed further in Sec. 4.2.1.

Binary and quantised measurements have also been explored for subspace estimation and search. In [151], the authors propose a Grassmannian locality-sensitive hashing (LSH) scheme based on random one-dimensional subspaces, resulting in binary codes that preserve neighbourhood structure for approximate nearest-subspace search. A more recent work introduces a simple sensing framework for recovering a principal subspace from binary measurements, demonstrating that accurate subspace estimation is possible without transmitting full data vectors [42]. Quantisation has also been considered in random feature methods for kernel approximation, where low-precision random Fourier features can reduce memory requirements while retaining good learning performance [164]. These works highlight the effectiveness of random angular comparisons and quantisation, though their focus lies on retrieval, subspace recovery or Euclidean kernel approximation rather than on random feature approximations of Grassmannian kernels.

Closest in spirit to the present work is [150] which proposes random projection operators for fast nearest-subspace search, probing subspaces using random directions to construct compact summaries. While conceptually related to our use of random rank-one projections, their goal is efficient retrieval rather than the construction of random features whose inner products approximate Grassmannian kernels with theoretical guarantees.

In contrast to existing approaches, we combine kernel methods with carefully designed rank-one random projections to construct scalable random feature maps for Grassmannian data. We develop bounded random feature constructions that yield well-defined, positive-definite Grassmannian kernels depending only on principal angles, and establish uniform approximation guarantees. These properties allow kernel-based learning on subspaces at a computational and memory cost that scales linearly with the ambient dimension, enabling practical large-scale applications.

4.1.3 Contributions

In this chapter, we develop random feature methods that are suitable for Grassmannian data. More precisely, each subspace $\mathcal{P} \in \mathcal{G}(k, n)$ is represented by its orthogonal projector $\boldsymbol{P} = \boldsymbol{U}\boldsymbol{U}^\top$, and is probed through m independent asymmetric rank-one quadratic measurements

of the form $\boldsymbol{a}_i^\top \boldsymbol{P}\boldsymbol{b}_i$, where $\boldsymbol{a}_i, \boldsymbol{b}_i \sim_{\text{i.i.d.}} \mathcal{N}(\boldsymbol{0}, \mathbf{I}_n)$. Stacking these measurements defines the rank-one projection (ROP) feature map

$$\psi^{\text{rop}}(\boldsymbol{U}) = (\boldsymbol{a}_i^\top \boldsymbol{U}\boldsymbol{U}^\top \boldsymbol{b}_i)_{i=1}^m,$$

whose empirical inner product is an estimator of the projection kernel but exhibits *heavy-tailed* concentration.

To control this behaviour, we compose ψ^{rop} with bounded nonlinear functions. Using the sign function yields binary random features $\psi^{\pm} = \text{sign} \circ \psi^{\text{rop}}$, from which we can build an empirical kernel

$$\widehat{\kappa}^{\pm}(\boldsymbol{U}, \boldsymbol{V}) = \tfrac{1}{m}\langle \psi^{\pm}(\boldsymbol{U}), \psi^{\pm}(\boldsymbol{V})\rangle$$

that converges uniformly, with high probability, to a well-defined, positive-definite and rotationally invariant Grassmannian kernel $\kappa^{\pm}$ that depends only on the principal angles between two subspaces [75]. Although no explicit closed form is known for $\kappa^{\pm}$, we show that it can be expressed as a function of the associated projectors $\boldsymbol{P} = \boldsymbol{U}\boldsymbol{U}^\top$ and $\boldsymbol{Q} = \boldsymbol{V}\boldsymbol{V}^\top$ with

$$\kappa^{\pm}(\boldsymbol{U}, \boldsymbol{V}) = 1 - \tfrac{2}{\pi}\mathbb{E}\measuredangle(\boldsymbol{P}\boldsymbol{g}, \boldsymbol{Q}\boldsymbol{g}),$$

where $\measuredangle(\boldsymbol{u}, \boldsymbol{v})$ denotes the angle between two vectors $\boldsymbol{u}$ and $\boldsymbol{v}$ and $\boldsymbol{g} \sim \mathcal{N}(\boldsymbol{0}, \mathbf{I}_n)$. This characterisation provides insight into the behaviour of $\kappa^{\pm}$ with respect to the principal angles between subspaces.

Since no closed-form expression is known for $\kappa^{\pm}$, we also introduce the aggregated binary embedding ψ^{Σ}, a variant of $\psi^{\pm}$ in which each coordinate averages N_Σ independent rank-one measurements before applying the sign function. The expected kernel associated with this construction is denoted by κ^{Σ}. We show that, as $N_\Sigma \to \infty$, κ^{Σ} converges to the explicit Grassmannian kernel

$$\kappa_\infty^{\Sigma}(\boldsymbol{U}, \boldsymbol{V}) = \tfrac{2}{\pi}\arcsin\left(\tfrac{1}{k}\textstyle\sum_{i=1}^k \cos^2\theta_i\right),$$

with an approximation error bounded by $C/\sqrt{N_\Sigma}$. This construction therefore keeps binary random features while providing access to an explicit limiting kernel, at the cost of computing N_Σ rank-one measurements for each coordinate.

Using instead the complex exponential defines periodic random features $\psi^{\circ} = \exp(\mathrm{i}\omega\psi^{\text{rop}})$, for which the expected kernel admits the closed form

$$\kappa^{\circ}(\boldsymbol{U}, \boldsymbol{V}) = \mathbb{E}\langle \psi^{\circ}(\boldsymbol{U}), \psi^{\circ}(\boldsymbol{V})\rangle = \textstyle\prod_{j=1}^k \left(1 + \omega^2 \sin^2\theta_j\right)^{-1},$$

where $\theta_1, \dots, \theta_k$ are the principal angles between $\boldsymbol{U}$ and $\boldsymbol{V}$. For $\psi^{\pm}$ and ψ° inner products of the random features provide unbiased estimators of the corresponding kernels and satisfy uniform approximation bounds over $\mathbb{V}(k,n)$ if $m = \mathcal{O}(kn)$ (up to logarithmic factors), yielding scalable Grassmannian kernel approximations with controlled error.

From a computational perspective, these constructions significantly reduce the cost of kernel-based learning. While classical Grassmannian kernels require $\mathcal{O}(N^2)$ evaluations and storage for an N-sample Gram matrix, and dense random embeddings of projectors incur $\mathcal{O}(n^2)$ cost per feature, the proposed bounded ROP-based features can be computed in $\mathcal{O}(kmn)$ operations and stored using $\mathcal{O}(mn)$ memory. Combined with the fact that $m = \mathcal{O}(kn)$ features suffice to control the approximation error, this yields overall complexities that scale linearly with the ambient dimension.

The proposed approach is illustrated in Fig. 4.1.

The rest of the chapter is structured as follows. We will first introduce linear random features and show that they do approximate Grassmannian kernels, but at a high cost. We then show that we can alleviate this cost by using asymmetric rank-one projections, but these show poor concentration behaviour and thus do not allow us to have a concentration result for any pair of subspaces over $\mathcal{G}(k,n)$. To solve this, we introduce the three aforementioned kernels based on a bounded function applied to the random features. For each of them we study their theoretical properties and derive uniform bounds over the entire Grassmannian manifold. We finally study their computational and memory requirements before concluding with supporting experiments, on both synthetic and real datasets.

4.1.4 Related publications

This chapter is related to the following publications, with contributions from Laurent Jacques and the author of the present document.

- R. Delogne and L. Jacques. "Random Features for Grassmannian Kernels". In: *2025 IEEE Statistical Signal Processing Workshop (SSP)*. 2025, pp. 221–224, [53]
- R. Delogne and L. Jacques. "Random features for Grassmannian kernel approximation with bounded rank-one projections". In: *Under review for Transactions on Machine Learning Research (TMLR)* (2026). URL: `openreview.net/forum?id=wq18dZJ2pA`, [52]

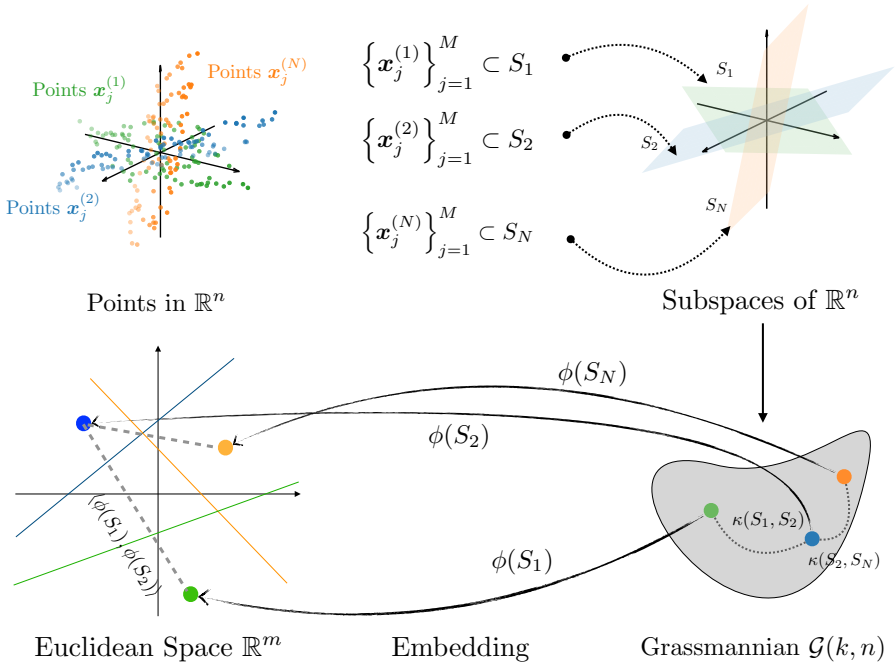


Fig. 4.1 Schematic representation of the proposed method. Data points are assumed to belong to low-dimensional subspaces of $\mathbb{R}^n$. Subspaces are represented by their orthonormal bases, which are then embedded via $\phi : \mathcal{G}(k, n) \to \mathbb{R}^m$ into a Euclidean space. Inner products in this space approximate well-defined Grassmannian kernels allowing for learning tasks to be performed efficiently.

4.2 From dense embeddings to rank-one random features

In this section, we present two linear random feature maps over subspaces of $\mathcal{G}(k, n)$, that enable us to approximate the projection kernel κ^{P} between two subspaces (see Sec. 2.4). These two maps are inspired by the compressive sensing of structured matrices [35, 67] and the context of compressive covariance estimation from quadratic sampling [41]. As explained below, while each of them has clear limitations, they define a benchmark from which we develop new random features in Sec. 4.3, 4.4 and 4.5.

4.2.1 Dense random projections of projectors

We can first embed each k-dimensional subspace $\mathcal{P} \in \mathcal{G}(k, n)$ by projecting its projector $\boldsymbol{P} \in \mathbb{G}(k, n)$ onto *dense* $n \times n$ random matrices. If $\mathcal{P}$ is known through a basis $\boldsymbol{U} \in \mathbb{V}(k, n)$, then projecting the well-defined $\boldsymbol{P} = \phi^{\mathrm{P}}(\boldsymbol{U}) = \boldsymbol{U}\boldsymbol{U}^\top$ (as well as any other function of $\boldsymbol{P}$) respects the

Grassmannian geometry. Formally, this random projection is

$$\Psi : \quad \boldsymbol{X} \in \mathbb{R}^{n\times n} \quad \mapsto \quad \Psi(\boldsymbol{X}) = \big(\langle \boldsymbol{\Psi}_i, \boldsymbol{X}\rangle\big)_{i=1}^{m} \in \mathbb{R}^m,$$

where the $n \times n$ probing matrices $\boldsymbol{\Psi}_i$ have Gaussian entries i.i.d. as $\mathcal{N}(0,1)$. Embedding elements of $\mathbb{V}(k,n)$ can then be done by composing Ψ with the projection embedding ϕ^{p}, *i.e.*, defining the *dense* random feature map

$$\psi^{\mathrm{d}} = \Psi \circ \phi^{\mathrm{p}} : \boldsymbol{U} \in \mathbb{V}(k,n) \mapsto \Psi(\phi^{\mathrm{p}}(\boldsymbol{U})) \in \mathbb{R}^m. \tag{4.1}$$

Interestingly, when we consider two bases of two k-dimensional subspaces $\mathcal{P}$ and $\mathcal{Q}$, inner products in the space mapped by ψ^{d} are unbiased estimators of the projection kernel κ^{p}. Indeed, if $\boldsymbol{U}$ and $\boldsymbol{V}$ are their respective $n \times k$ orthonormal bases, then, defining the empirical kernel

$$\widehat{\kappa}^{\mathrm{d}}(\boldsymbol{U},\boldsymbol{V}) := \tfrac{1}{m}\langle \psi^{\mathrm{d}}(\boldsymbol{U}), \psi^{\mathrm{d}}(\boldsymbol{V})\rangle,$$

we find

$$\begin{aligned}\mathbb{E}\,\widehat{\kappa}^{\mathrm{d}}(\boldsymbol{U},\boldsymbol{V}) &= \tfrac{1}{m}\textstyle\sum_{i=1}^{m} \mathbb{E}\langle \phi^{\mathrm{p}}(\boldsymbol{U}), \boldsymbol{\Psi}_i\rangle\langle \boldsymbol{\Psi}_i, \phi^{\mathrm{p}}(\boldsymbol{V})\rangle \\ &= \langle \phi^{\mathrm{p}}(\boldsymbol{U}), \phi^{\mathrm{p}}(\boldsymbol{V})\rangle \\ &= \kappa^{\mathrm{p}}(\boldsymbol{U},\boldsymbol{V}),\end{aligned}$$

since $\mathbb{E}\,\mathrm{vec}(\boldsymbol{\Psi}_i)\,\mathrm{vec}(\boldsymbol{\Psi}_i)^\top = \mathbf{I}_{n^2}$ by design.

We can also study how these inner products deviate from the projection kernel by invoking the compressive sensing literature [67]. The mapping Ψ is indeed known to embed low-rank matrices into $\mathbb{R}^m$ with a controlled distortion on their Frobenius norms [35]. Given some distortion level $0 < \delta < 1$ and provided $m = \mathcal{O}(\delta^{-2}kn)$ (up to log factors), the mapping Ψ respects, with high probability, the RIP (see Eq. 2.7 in Sec. 2.2.3), or $\mathrm{RIP}(\delta, \mathcal{R}(k,n))$, over the set $\mathcal{R}(k,n)$ of $n \times n$ rank-k matrices:

$$(1-\delta)\|\boldsymbol{X}\|_F^2 \;\le\; \tfrac{1}{m}\|\Psi(\boldsymbol{X})\|_2^2 \;\le\; (1+\delta)\|\boldsymbol{X}\|_F^2, \quad \forall \boldsymbol{X} \in \mathcal{R}(k,n). \tag{RIP}$$

The RIP allows us to reach a uniform approximation of the projection kernel κ^{p} through $\widehat{\kappa}^{\mathrm{d}}$ with distortion δk, as expressed in the following proposition.

Proposition 4.1 (RIP $\Rightarrow$ uniform kernel approximation on $\mathcal{G}(k,n)$)**.** *If the mapping $\Psi : \mathbb{R}^{n\times n} \to \mathbb{R}^m$ respects the $\mathrm{RIP}(\delta, \mathcal{R}(2k,n))$ for some distortion $0 < \delta < 1$, then,*

$$\big|\widehat{\kappa}^{\mathrm{d}}(\boldsymbol{U},\boldsymbol{V}) - \kappa^{\mathrm{p}}(\boldsymbol{U},\boldsymbol{V})\big| \le \delta k, \quad \forall \boldsymbol{U},\boldsymbol{V} \in \mathbb{V}(k,n).$$

Proof. Writing $\boldsymbol{P} = \boldsymbol{U}\boldsymbol{U}^\top$ and $\boldsymbol{Q} = \boldsymbol{V}\boldsymbol{V}^\top$, by the polarisation identity,

$$\kappa^{\mathrm{p}}(\boldsymbol{U}, \boldsymbol{V}) = \langle \boldsymbol{P}, \boldsymbol{Q} \rangle = \tfrac{1}{4}\big(\|\boldsymbol{P} + \boldsymbol{Q}\|_F^2 - \|\boldsymbol{P} - \boldsymbol{Q}\|_F^2\big)$$

and

$$\widehat{\kappa}^{\mathrm{d}}(\boldsymbol{U}, \boldsymbol{V}) = \tfrac{1}{4m}(\|\Psi(\boldsymbol{P} + \boldsymbol{Q})\|_2^2 - \|\Psi(\boldsymbol{P} - \boldsymbol{Q})\|_2^2.)$$

Since Ψ respects the RIP over $\mathcal{R}(2k, n)$, a set which includes the matrices $\boldsymbol{P} \pm \boldsymbol{Q}$, we get

$$\widehat{\kappa}^{\mathrm{d}}(\boldsymbol{U}, \boldsymbol{V}) \leq \langle \boldsymbol{P}, \boldsymbol{Q} \rangle + \tfrac{1}{4}(\|\boldsymbol{P} + \boldsymbol{Q}\|^2 + \|\boldsymbol{P} - \boldsymbol{Q}\|^2) \leq \langle \boldsymbol{P}, \boldsymbol{Q} \rangle + k\delta,$$

since $\|\boldsymbol{P}\|_F^2 = \|\boldsymbol{Q}\|_F^2 = k$ and $\|\boldsymbol{P} + \boldsymbol{Q}\|_F^2 + \|\boldsymbol{P} - \boldsymbol{Q}\|_F^2 = 4k$. We obtain the lower bound similarly. □

Having an approximation error bounded by δk instead of δ in Prop. 4.1 is a severe limitation. From a rescaling argument, if we rather target, with high probability, the uniform error

$$|\widehat{\kappa}^{\mathrm{d}}(\boldsymbol{U}, \boldsymbol{V}) - \kappa^{\mathrm{p}}(\boldsymbol{U}, \boldsymbol{V})| < \delta, \quad \forall \boldsymbol{U}, \boldsymbol{V} \in \mathbb{V}(k, n),$$

then Ψ must respect the RIP$(\delta/k, \mathcal{R}(2k, n))$, which thus happens with high probability provided $m = \mathcal{O}(\delta^{-2}k^3 n)$. Consequently, the resulting feature map ψ^{d} is not scalable in practice. First, each matrix $\boldsymbol{\Psi}_i \in \mathbb{R}^{n \times n}$ contains n^2 entries, globally requiring the storage of $\mathcal{O}(mn^2)$ real numbers. Second, evaluating each feature $\langle \boldsymbol{\Psi}_i, \phi^{\mathrm{p}}(\boldsymbol{U}) \rangle$ in Eq. 4.1 costs $\mathcal{O}(n^2)$ operations, so also a total of $\mathcal{O}(mn^2)$ operations for the m projections.

Consequently, evaluating ψ^{d} with $m = \mathcal{O}(\delta^{-2}k^3 n)$ components to estimate κ^{p} through $\hat{\kappa}$ has memory and computational complexities of $\mathcal{O}(\delta^{-2}k^3 n^3)$, which can be worse than a single evaluation of both κ^{p} and κ^{BC} for small values of $k < n$.

4.2.2 Rank-one projections for lighter storage and computation

An alternative random feature map with reduced computational and storage costs compared with dense random projections can be built using an asymmetric version of the rank-one projection embedding introduced in Eq. (2.8). Given a subspace $\mathcal{P}$ of $\mathcal{G}(k, n)$ with basis $\boldsymbol{U} \in \mathbb{V}(k, n)$, we define the *rank-one projection* embedding ψ^{rop} [41, 29, 56, 55] from m pairs of independent Gaussian random vectors $\boldsymbol{a}_i, \boldsymbol{b}_i \sim \mathcal{N}(\boldsymbol{0}, \mathbf{I}_n)$ and the associated rank-one matrices $\boldsymbol{A}_i = \boldsymbol{a}_i \boldsymbol{b}_i^\top$ as

$$\psi^{\mathrm{rop}} : \boldsymbol{U} \in \mathbb{R}^{n \times k} \mapsto \psi^{\mathrm{rop}}(\boldsymbol{U}) = \big(\langle \boldsymbol{A}_i, \phi^{\mathrm{p}}(\boldsymbol{U}) \rangle = \boldsymbol{a}_i^\top \boldsymbol{U}\boldsymbol{U}^\top \boldsymbol{b}_i\big)_{i=1}^m. \quad (4.2)$$

Crucially, as for Ψ, the feature map ψ^{rop}, which is defined over ϕ^{p}, is invariant to the choice of basis $\boldsymbol{U}$ meaning that it only depends on the

subspace $\mathcal{P}$. Moreover, we can compute the components of $\psi^{\mathrm{rop}}(\boldsymbol{U})$ with complexity of $\mathcal{O}(kmn)$ by computing all the k-dimensional vectors $\boldsymbol{\alpha}_i := \boldsymbol{U}^\top \boldsymbol{a}_i$ and $\boldsymbol{\beta}_i := \boldsymbol{U}^\top \boldsymbol{b}_i$, $1 \leq i \leq m$, with storage and computational complexities $\mathcal{O}(kmn)$, and then evaluating all the inner products $\psi_i^{\mathrm{rop}}(\boldsymbol{U}) = \boldsymbol{\alpha}_i^\top \boldsymbol{\beta}_i$ in $\mathcal{O}(km)$ computations. This makes ROP both geometrically sound and computationally practical.

Interestingly, like for Ψ again, given two k-dimensional subspaces $\mathcal{P}, \mathcal{Q} \in \mathcal{G}(k,n)$ with bases $\boldsymbol{U}, \boldsymbol{V} \in \mathbb{V}(k,n)$, the empirical kernel

$$\widehat{\kappa}^{\mathrm{rop}}(\boldsymbol{U}, \boldsymbol{V}) := \tfrac{1}{m}\langle \psi^{\mathrm{rop}}(\boldsymbol{U}), \psi^{\mathrm{rop}}(\boldsymbol{V})\rangle, \tag{4.3}$$

is an unbiased estimator of the projection kernel $\kappa^{\mathrm{p}}(\boldsymbol{U}, \boldsymbol{V})$. Indeed, we observe that

$$\begin{aligned}
\tfrac{1}{m}\mathbb{E}\langle \psi^{\mathrm{rop}}(\boldsymbol{U}), \psi^{\mathrm{rop}}(\boldsymbol{V})\rangle &= \frac{1}{m}\sum_{i=1}^{m} \mathbb{E}\boldsymbol{a}_i^\top \phi^{\mathrm{p}}(\boldsymbol{U}) \boldsymbol{b}_i \boldsymbol{b}_i^\top \phi^{\mathrm{p}}(\boldsymbol{V}) \boldsymbol{a}_i \\
&= \frac{1}{m}\sum_{i=1}^{m} \mathbb{E}(\boldsymbol{a}_i^\top \phi^{\mathrm{p}}(\boldsymbol{U}) \phi^{\mathrm{p}}(\boldsymbol{V}) \boldsymbol{a}_i) \\
&= \tfrac{1}{m}\textstyle\sum_{i=1}^{m} \operatorname{tr}(\boldsymbol{U}\boldsymbol{U}^\top \boldsymbol{V}\boldsymbol{V}^\top) \\
&= \|\boldsymbol{U}^\top \boldsymbol{V}\|_F^2 = \kappa^{\mathrm{p}}(\boldsymbol{U}, \boldsymbol{V}),
\end{aligned}$$

where we used the cyclic property of the trace.

Although $\widehat{\kappa}^{\mathrm{rop}}$ estimates κ^{p} on average, its concentration behaviour is less favourable than the one obtained from bounded random features. Indeed, each term of the sum composing $m\widehat{\kappa}^{\mathrm{rop}}$ has the same distribution as the random variable

$$Z := (\boldsymbol{a}^\top \boldsymbol{U}\boldsymbol{U}^\top \boldsymbol{b})(\boldsymbol{a}^\top \boldsymbol{V}\boldsymbol{V}^\top \boldsymbol{b}) = \boldsymbol{\alpha}^\top \boldsymbol{\beta} \boldsymbol{\alpha}'^\top \boldsymbol{\beta}',$$

where $\boldsymbol{a}, \boldsymbol{b} \sim_{\text{i.i.d.}} \mathcal{N}(\boldsymbol{0}, \mathbf{I}_n)$ gives the projections $\boldsymbol{\alpha} := \boldsymbol{U}^\top \boldsymbol{a}$ and $\boldsymbol{\beta} := \boldsymbol{U}^\top \boldsymbol{b}$ onto $\boldsymbol{U}$, and similarly for the projections $\boldsymbol{\alpha}'$ and $\boldsymbol{\beta}'$ onto $\boldsymbol{V}$. This is mainly the product of four (correlated) Gaussian random variables. The random variable Z has thus distribution tails that are heavier than the sub-exponential random variables [148]. As highlighted in [24], this places our estimator in a sub-Weibull (with parameter $1/2$) regime, where deviations are still controlled, but they decay much more slowly than the exponential-type rates enjoyed by sub-exponential random variables.

This distinction is especially important when going from a fixed pair of subspaces to a uniform guarantee over the full Grassmannian. For any fixed pair $\mathcal{P}, \mathcal{Q}$ with bases $\boldsymbol{U}, \boldsymbol{V} \in \mathbb{V}(k,n)$, the empirical kernel $\widehat{\kappa}^{\mathrm{rop}}(\boldsymbol{U}, \boldsymbol{V})$ still concentrates around $\kappa^{\mathrm{p}}(\boldsymbol{U}, \boldsymbol{V})$, with a failure probability decaying as $\exp(-c\sqrt{m})$ for a fixed approximation error. For

a finite dataset of N subspaces, a union bound over all pairs therefore gives a failure probability scaling as $N^2 \exp(-c\sqrt{m})$. Hence, for a fixed approximation error and target failure probability, a Johnson-Lindenstrauss like guarantee can still be obtained with m scaling as $\mathcal{O}(\log^2 N)$, which explains why unbounded ROP features can perform well on finite datasets. However, obtaining a data-independent approximation guarantee over *all pairs* of subspaces in the continuous space $\mathcal{G}(k,n)$ is much more demanding with such heavy-tailed distributions. In fact, following the proof techniques used in this work suggests that to approximate κ^{p} with $\widehat{\kappa}^{\mathrm{rop}}$ for all subspaces in $\mathbb{V}(k,n)$ requires $m = \mathcal{O}((kn)^2)$ measurements, *i.e.*, an embedding dimension much larger than the dimension of $\mathcal{G}(k,n)$. This is also consistent with the observations of [41] and [29] who explain that ψ^{rop} cannot satisfy the RIP with a reasonable number of measurements. These limitations motivate the introduction of bounded non-linear functions in Sec. 4.3, 4.4 and 4.5.

4.2.3 Improving concentration behaviour

We can both remove the computational complexity and storage limitations of dense random projections and the heavy-tailed distribution of rank-one projections by composing ROPs with a bounded, non-linear function. Practically, given a bounded function $g : \mathbb{R} \to \mathbb{K}$ (with $\mathbb{K}$ equal to $\mathbb{R}$ or $\mathbb{C}$), with $|g(\lambda)| \leq 1$ for all $\lambda \in \mathbb{R}$ without loss of generality, we apply it componentwise to the rank-one projection operator ψ^{rop}, *i.e.*, we define

$$\psi^g : \boldsymbol{U} \in \mathbb{V}(k,n) \;\mapsto g(\psi^{\mathrm{rop}}(\boldsymbol{U})) \in \mathbb{K}^m. \tag{4.4}$$

This raises several central questions. First, which kernel κ is reached in expectation by the empirical kernel

$$\widehat{\kappa}^g(\boldsymbol{U},\boldsymbol{V}) = \tfrac{1}{m}\langle \psi^g(\boldsymbol{U}), \psi^g(\boldsymbol{V})\rangle, \quad \boldsymbol{U},\boldsymbol{V} \in \mathbb{V}(k,n).$$

Is it a valid kernel with respect to the Grassmannian geometry? And can we uniformly bound the related approximation error between $\widehat{\kappa}^g$ and κ^g?

Among the many possible choices of g, we focus on three specific examples. The first is the *sign function*, $g(\lambda) = \operatorname{sign}(\lambda)$ equals 1 if $\lambda > 0$ and to -1 otherwise. It is inspired by the one-bit compressive sensing literature [86, 67] and yields very light *binary* random features $\psi^{\pm} := \operatorname{sign} \circ\ \psi^{\mathrm{rop}}$ with codomain in $\{\pm 1\}^m$, *i.e.*, each subspace is encoded over no more than m bits. It is studied in detail in Sec. 4.3.

The second construction is motivated by the fact that no closed-form expression of $\kappa^{\pm}$ is known when $k > 1$. We therefore introduce an aggregated variant of the binary embedding. Instead of applying the sign function to a single rank-one projection, the embedding ψ^{Σ} is defined as a vector with each entry containing the sum of N_{Σ} independent rank-one projections in each coordinate. It still is defined in $\{\pm 1\}^m$ and therefore retains the same m-bit representation. This aggregation allows us to apply a Berry-Esseen theorem and compare the resulting kernel κ^{Σ} with a Gaussian limit. As $N_{\Sigma} \to \infty$, this gives access to an explicit closed-form Grassmannian kernel, with an approximation error controlled by N_{Σ}. It is studied in detail in Sec. 4.4.

The third is reminiscent of the random Fourier features [125] and applies the complex exponential map, $g(\lambda) = \exp(\mathrm{i}\omega\lambda)$ for some frequency $\omega \in \mathbb{R}$, to ψ^{rop}, *i.e.*, we define the *periodic* random features $\psi^{\circ}(\boldsymbol{U}) := \exp(\mathrm{i}\omega\psi^{\mathrm{rop}}(\boldsymbol{U}))$ for any subspace $\mathcal{P}$ with basis $\boldsymbol{U} \in \mathbb{V}(k, n)$. It is studied in detail in Sec. 4.5.

4.3 Signed random features for Grassmannian kernels

In this section we study the first of the two functions suggested above to improve the concentration properties of the rank-one operator, namely the sign function applied componentwise to the rank-one projection operator ψ^{rop}. It yields the binary random feature map $\psi^{\pm}$, whose values belong to $\{\pm 1\}^m$. This construction has the advantage of taming the heavy-tails of raw rank-one projections as well as producing extremely compact embeddings thanks to its binary nature.

We will begin in Sec. 4.3.1 by formally defining the embedding, identifying the kernel induced by $\psi^{\pm}$ in expectation and studying its basic properties. In particular, we show that this kernel is well defined on $\mathcal{G}(k, n)$.

We then turn to the approximation properties of the empirical kernel associated with $\psi^{\pm}$. After a simple pointwise concentration bound, we develop the proof of a uniform approximation result over $\mathbb{V}(k, n)$. The discontinuous nature of $\psi^{\pm}$ makes this proof challenging. We will construct it from smaller lemmas in Sec. 4.3.2.

Next, in Sec. 4.3.3 we study the particular case working on $\mathcal{G}(1, n)$. This case is particularly interesting as 1-dimensional subspaces are equivalent to vectors, which links back directly to the case studied in Chap. 3. Furthermore, although no closed form of $\kappa^{\pm}$ is known in general, we obtain an explicit formula in this particular case.

4.3.1 Definition and properties

To define the binary random features $\psi^{\pm} : \mathbb{V}(k,n) \to \{\pm 1\}^m$, we set $g(\cdot) = \operatorname{sign}(\cdot)$ in Eq. 4.4, *i.e.*, given the m random vectors $\boldsymbol{a}_i, \boldsymbol{b}_i \sim_{\text{i.i.d.}} \mathcal{N}(\mathbf{0}, \mathbf{I}_n)$, with $1 \leq i \leq m$,

$$\psi^{\pm}(\boldsymbol{U}) := \left(\operatorname{sign}(\boldsymbol{a}_i^{\top} \phi^{\mathrm{p}}(\boldsymbol{U}) \boldsymbol{b}_i)\right)_{i=1}^{m}, \quad \boldsymbol{U} \in \mathbb{V}(k,n). \tag{4.5}$$

This provides the empirical kernel

$$\widehat{\kappa}^{\pm}(\boldsymbol{U}, \boldsymbol{V}) := \tfrac{1}{m} \langle \psi^{\pm}(\boldsymbol{U}), \psi^{\pm}(\boldsymbol{V}) \rangle,$$

and, given any pair of bases $\boldsymbol{U}, \boldsymbol{V} \in \mathbb{V}(k,n)$, we are interested in the kernel

$$\kappa^{\pm}(\boldsymbol{U}, \boldsymbol{V}) := \mathbb{E}[\langle \psi^{\pm}(\boldsymbol{U}), \psi^{\pm}(\boldsymbol{V}) \rangle]. \tag{4.6}$$

The binary codomain of $\psi^{\pm}$ offers the advantage of reducing the cost of storing and possibly transmitting subspace random features: each of the m random features requires a single bit instead of floating-point numbers which are typically stored using 32 or 64 bits [104, 81]. This is especially valuable in settings where bandwidth or memory is constrained (*e.g.*, embedded devices or distributed systems). Moreover, binary vectors enable extremely fast bitwise operations, making both storage and computation far more efficient than in the full-precision case.

Unfortunately, when $k > 1$, there is no known closed form for $\kappa^{\pm}$. However, despite not knowing its analytical form, this kernel is valid as shown in the following proposition.

Proposition 4.2 (Validity of $\kappa^{\pm}$)**.** *For bases $\boldsymbol{U}, \boldsymbol{V} \in \mathbb{V}(k,n)$ whose subspaces form the principal angles $\theta_1, \ldots, \theta_k$, the kernel $\kappa^{\pm}$ is positive-definite, symmetric and only depends on these principal angles,* i.e., *$\kappa^{\pm}(\boldsymbol{U}, \boldsymbol{V}) = f(\theta_1, \ldots, \theta_k)$ for some function f. Moreover, given $\boldsymbol{g} \sim \mathcal{N}(\boldsymbol{0}, \mathbf{I}_n)$,*

$$\kappa^{\pm}(\boldsymbol{U}, \boldsymbol{V}) = 1 - \tfrac{2}{\pi} \mathbb{E} \measuredangle(\phi^{\mathrm{p}}(\boldsymbol{U})\boldsymbol{g}, \phi^{\mathrm{p}}(\boldsymbol{V})\boldsymbol{g}), \tag{4.7}$$

with $\measuredangle(\boldsymbol{u}, \boldsymbol{v})$ the angle between two vectors $\boldsymbol{u}$ and $\boldsymbol{v}$.

Proof. By design, $\kappa^{\pm}$ is obviously symmetric. Regarding its positive-semidefiniteness, we observe that, given an arbitrary number of N elements $\boldsymbol{U}_1, \ldots, \boldsymbol{U}_N$ in $\mathbb{V}(k,n)$, with $\boldsymbol{P}_i := \boldsymbol{U}_i \boldsymbol{U}_i^{\top}$, the Gram matrix $\boldsymbol{K} \in \mathbb{R}^{N \times N}$ whose entries are $K_{ij} = \kappa^{\pm}(\boldsymbol{U}_i, \boldsymbol{U}_j)$ is such that for any $\boldsymbol{c} \in \mathbb{R}^N$, $\boldsymbol{c}^{\top} \boldsymbol{K} \boldsymbol{c} = \mathbb{E}(\boldsymbol{c}^{\top} \boldsymbol{p})^2 \geq 0$, with $\boldsymbol{p} \in \{\pm 1\}^N$ such that $p_i = \operatorname{sign}(\boldsymbol{a}^{\top} \boldsymbol{P}_i \boldsymbol{b})$.

Regarding the angular dependency of the kernel, since for any $n \times n$ matrix $\boldsymbol{M}$ each component of $\psi^{\text{rop}}(\boldsymbol{M})$ is distributed as $\boldsymbol{a}^\top \boldsymbol{M}\boldsymbol{b}$, with $\boldsymbol{a}, \boldsymbol{b} \sim_{\text{i.i.d.}} \mathcal{N}(\boldsymbol{0}, \mathbf{I}_n)$, by exploiting the rotational invariance of these random vectors, we have

$$\begin{aligned}\kappa^{\pm}(\boldsymbol{U}, \boldsymbol{V}) &= \mathbb{E}\text{sign}(\boldsymbol{a}^\top \boldsymbol{P}\boldsymbol{b})\text{sign}(\boldsymbol{a}^\top \boldsymbol{Q}\boldsymbol{b}) \\ &= \mathbb{E}\text{sign}(\boldsymbol{a}^\top \boldsymbol{R}\boldsymbol{P}\boldsymbol{R}^\top \boldsymbol{b})\text{sign}(\boldsymbol{a}^\top \boldsymbol{R}\boldsymbol{Q}\boldsymbol{R}^\top \boldsymbol{b}) \\ &= \kappa^{\pm}(\boldsymbol{R}\boldsymbol{U}, \boldsymbol{R}\boldsymbol{V}),\end{aligned}$$

for any orthogonal matrix $\boldsymbol{R} \in \mathsf{SO}(n)$. Therefore, from Thm. 2.1, $\kappa^{\pm}(\boldsymbol{U}, \boldsymbol{V})$ only depends on the principal angles between $\boldsymbol{U}$ and $\boldsymbol{V}$. Finally, the expression of Eq. 4.7 is a simple consequence of the arcsin law used in [146, sec. 3, eq. 17] or Grothendieck's identity in [148, Lem. 3.6.5], *i.e.*, we show easily that, by the law of total expectation,

$$\begin{aligned}\kappa^{\pm}(\boldsymbol{U}, \boldsymbol{V}) &= \mathbb{E}[\text{sign}\big(\boldsymbol{a}^\top(\boldsymbol{P}\boldsymbol{b})\big)\,\text{sign}\big(\boldsymbol{a}^\top(\boldsymbol{Q}\boldsymbol{b})\big)] \\ &= \tfrac{2}{\pi}\mathbb{E}\arcsin\Big(\frac{\boldsymbol{b}^\top \boldsymbol{P}\boldsymbol{Q}\boldsymbol{b}}{\|\boldsymbol{P}\boldsymbol{b}\|\,\|\boldsymbol{Q}\boldsymbol{b}\|}\Big),\end{aligned}$$

which shows the result. □

While we do not know any closed form formula for $\kappa^{\pm}$, Eq. 4.7 in Prop. 4.2 provides a noteworthy interpretation: $1 - \kappa^{\pm}(\boldsymbol{U}, \boldsymbol{V})$ is proportional to the average angle made by the projections of a Gaussian random vector on $\boldsymbol{U}$ and $\boldsymbol{V}$. Interestingly, a similar concept was introduced in [84] where the angle $\theta(\boldsymbol{U}, \boldsymbol{V})$ between $\boldsymbol{U}$ and $\boldsymbol{V}$ is defined as $\theta(\boldsymbol{U}, \boldsymbol{V}) = \arccos\big(\frac{1}{k}\sum_{i=1}^{k}\cos^2\theta_i\big)$, a form of non-linear averaging of the principal angles and the angular similarity between these two subspaces as $\kappa^{\text{sim}}(\boldsymbol{U}, \boldsymbol{V}) = 1 - \frac{1}{\pi}\theta(\boldsymbol{U}, \boldsymbol{V})$.

We now address the question of bounding the approximation error of $\widehat{\kappa}^{\pm}$ relative to $\kappa^{\pm}$. We can first observe the following concentration phenomenon, *i.e.*, a pointwise approximation error bound on a *fixed* pair of subspaces.

Proposition 4.3 (Pointwise approximation error of $\kappa^{\pm}$ by $\widehat{\kappa}^{\pm}$)**.** *Given a pair of subspaces in $\mathcal{G}(k, n)$ with bases $\boldsymbol{U}, \boldsymbol{V} \in \mathbb{V}(k, n)$, the m-component binary random feature map $\psi^{\pm}$ (Eq. 4.5) related to m i.i.d. Gaussian random vectors $\boldsymbol{a}_j, \boldsymbol{b}_j \sim \mathcal{N}(\boldsymbol{0}, \mathbf{I}_n)$, and an error $\delta > 0$, we have*

$$\mathbb{P}\big(\big|\tfrac{1}{m}\langle\psi^{\pm}(\boldsymbol{U}), \psi^{\pm}(\boldsymbol{V})\rangle - \kappa^{\pm}(\boldsymbol{U}, \boldsymbol{V})\big| < \delta\big) \geq 1 - 2\exp(-\tfrac{1}{2}m\delta^2).$$

Proof. The proof is a direct application of Hoeffding's inequality (see *e.g.*, [148]) since given the random variables

$$X_j := \operatorname{sign}(\boldsymbol{a}_j^\top \boldsymbol{P}\boldsymbol{b}_j)\operatorname{sign}(\boldsymbol{a}_j^\top \boldsymbol{Q}\boldsymbol{b}_j), \quad 1 \leq j \leq m,$$

all X_j's are i.i.d. , bounded, and

$$\mathbb{E}[\widehat{\kappa}^{\pm}(\boldsymbol{U}, \boldsymbol{V})] = \mathbb{E} \sum_{j=1}^{m} X_j / m = \kappa^{\pm}(\boldsymbol{U}, \boldsymbol{V}).$$

□

Then, we can show that $\widehat{\kappa}^{\pm}$ provides a uniform approximation of $\kappa^{\pm}$ over all subspace pairs in $\mathcal{G}(k, n)$.

Proposition 4.4 (Uniform approximation error for $\widehat{\kappa}^{\pm} \approx \kappa^{\pm}$)**.** *Let $\delta > 0$ and $C, C', c > 0$ be absolute constants. If*

$$m \geq C\delta^{-2} nk \log(\tfrac{n^2}{k\delta^2}),$$

then, with probability exceeding $1 - C' \exp(-c\delta^2 m)$,

$$\sup_{\boldsymbol{U}, \boldsymbol{V} \in \mathbb{V}(k,n)} \left| \widehat{\kappa}^{\pm}(\boldsymbol{U}, \boldsymbol{V}) - \kappa^{\pm}(\boldsymbol{U}, \boldsymbol{V}) \right| \leq \delta.$$

Proof. The proof is explained in detail in Sec. 4.3.2 where we first develop its building blocks before combining them to show the desired result. □

Prop. 4.4 thus shows that having a binary feature map $\psi^{\pm}$ with $m = \mathcal{O}(kn)$ components, *i.e.*, with m proportional to the intrinsic dimension of each subspace of $\mathcal{G}(k, n)$, allows us to approximate, with high and controlled probability, the expected kernel $\kappa^{\pm}$ of k-dimensional subspaces by the empirical kernel $\widehat{\kappa}^{\pm}$ reached by mere inner product of the random features of the two subspaces.

4.3.2 Proof of Prop. 4.4

Rather than attacking the uniform bound directly, we establish a series of intermediate results clarifying how sign changes could occur under small perturbations of the projector. We begin with the stability of the simpler one-bit map $\boldsymbol{x} \mapsto \operatorname{sign}(\boldsymbol{a}^\top \boldsymbol{x})$, and then transfer this idea to the behaviour of $\psi^{\pm}$ in the neighbourhood of a projector $\boldsymbol{P}^* \in \mathbb{G}(k, n)$. This leads to bounds on the probability that one measurement flips sign in the neighbourhood, and then to a concentration result controlling the

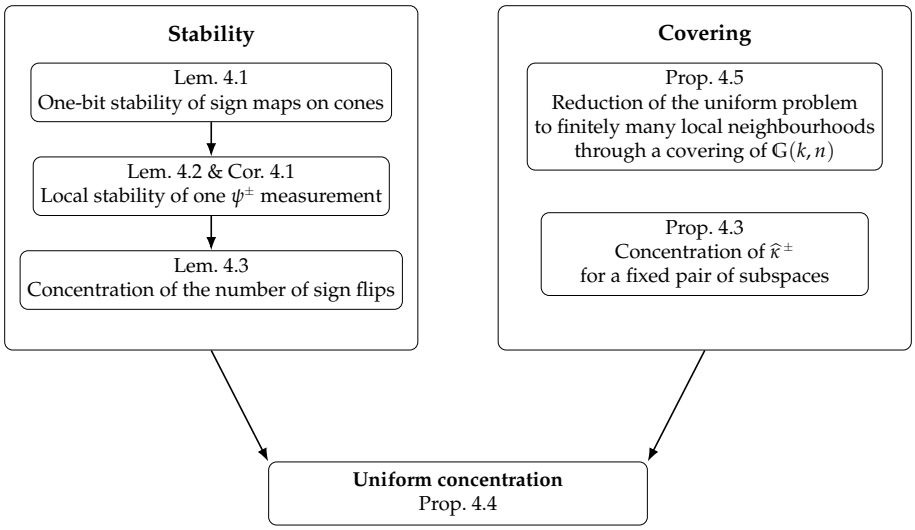


Fig. 4.2 Structure of the proof of Prop. 4.4. The argument combines a local stability analysis of $\psi^{\pm}$ with a covering argument on $\mathbb{G}(k,n)$.

total number of such flips in the full measurement vector. In parallel, we introduce a covering of $\mathbb{G}(k,n)$ allowing us to reduce the uniform statement to finitely many local estimates. The final proposition is then obtained by combining these local stability results, the covering argument, and a union bound. We present this proof step by step in the following lemmas and propositions so that the role of each ingredient remains explicit all the way to the final concentration result. A visual depiction of the proof structure is given in Fig. 4.1.

Lemma 4.1 (Stability of 1-bit map on a cone)**.** *Let* $\mathcal{C}_\alpha(\boldsymbol{u}) = \{\boldsymbol{x} \in \mathbb{R}^n : \measuredangle(\boldsymbol{x},\boldsymbol{u}) \leq \alpha\}$ *be a cone of axis* $\boldsymbol{u} \in \mathbb{S}^{n-1}$ *and half aperture* $0 \leq \alpha \leq \pi$. *Given the random map* $\bar{\psi} : \boldsymbol{x} \in \mathbb{R}^n \mapsto \operatorname{sign}(\boldsymbol{a}^\top \boldsymbol{x})$ *with* $\boldsymbol{a} \sim \mathcal{N}(\boldsymbol{0}, \mathbf{I}_n)$, *we have*

$$\mathbb{P}\left[\bar{\psi} \notin \mathsf{C}^0\big(\mathcal{C}_\alpha(\boldsymbol{u})\big)\right] = 1$$

if $\alpha > \pi/2$ *and*

$$\mathbb{P}\left[\bar{\psi} \notin \mathsf{C}^0\big(\mathcal{C}_\alpha(\boldsymbol{u})\big)\right] \leq \sqrt{\tfrac{2}{\pi}}(\tan\alpha)\sqrt{n}, \quad \text{if } 0 < \alpha \leq \pi/2,$$

where $\mathsf{C}^0(\mathcal{S})$ *denotes continuous functions on a set* $\mathcal{S}$.

Proof. The proof is adapted from [144, Corollary 12]. We first observe that $\bar{\psi}$ is discontinuous over $\mathcal{C}_\alpha(\boldsymbol{u})$ iff $|\boldsymbol{a}^\top \boldsymbol{u}| \leq \epsilon \|\boldsymbol{a}\|$ with $\epsilon := \sin\alpha$. Therefore, by the rotational invariance of the Gaussian distribution we

can choose $\boldsymbol{u} = \boldsymbol{e}_1 = (1, 0, \ldots, 0)^\top$ and the probability above amounts to computing

$$\begin{aligned} p :&= \mathbb{P}[|a_1|^2 \leq \epsilon^2 \|\boldsymbol{a}\|^2] \\ &= \mathbb{P}[a_1^2 \leq \frac{\epsilon^2}{(1-\epsilon^2)}(a_2^2 + \ldots + a_n^2)] \\ &= \mathbb{E}_\xi \mathbb{P}[a_1^2 \leq \frac{\epsilon^2}{(1-\epsilon^2)}\xi | \xi], \end{aligned}$$

where $\xi \sim \chi^2(n-1)$ is independent of a_1. This gives

$$\mathbb{P}[a_1^2 \leq \tfrac{\epsilon^2}{(1-\epsilon^2)}\xi | \xi] \leq \sqrt{\tfrac{2}{\pi}} \tfrac{\epsilon}{\sqrt{1-\epsilon^2}} \sqrt{\xi} = \sqrt{\tfrac{2}{\pi}} (\tan \alpha) \sqrt{\xi}$$

Observing that $\mathbb{E}_\xi \sqrt{\xi} \leq \sqrt{\mathbb{E}_\xi \xi} \leq \sqrt{n-1} \leq \sqrt{n}$ by Jensen's inequality gives the result. □

From this result, we can show that, when the projectors of two subspaces are close in Frobenius norm, each component of the binary random features related to $\psi^\pm$ coincides with high probability.

Lemma 4.2 (Local stability of the binary embedding)**.** *We consider the subspace $\boldsymbol{P}^* \in \mathbb{G}(k, n)$ and the random vectors $\boldsymbol{a}, \boldsymbol{b} \sim \mathcal{N}(\boldsymbol{0}, \mathbf{I}_n)$. Define the map $\bar{\psi} : \boldsymbol{M} \in \mathbb{R}^{n \times n} \mapsto \operatorname{sign}(\boldsymbol{a}^\top \boldsymbol{M} \boldsymbol{b})$, the ball*

$$\mathbb{B}_\epsilon^F(\boldsymbol{P}^*) = \{\boldsymbol{M} \in \mathbb{R}^{n \times n} : \|\boldsymbol{M} - \boldsymbol{P}^*\|_F \leq \epsilon\}$$

of radius $\epsilon > 0$ and centred on $\boldsymbol{P}^* \in \mathbb{R}^{n \times n}$, and the neighbourhood

$$\mathcal{N}_\epsilon(\boldsymbol{P}^*) := \mathbb{B}_\epsilon^F(\boldsymbol{P}^*) \cap \mathbb{G}(k, n).$$

Then,

$$\begin{aligned} \mathbb{P}\left[\exists \boldsymbol{P}, \boldsymbol{Q} \in \mathcal{N}_\epsilon(\boldsymbol{P}^*) : \bar{\psi}(\boldsymbol{P}) \neq \bar{\psi}(\boldsymbol{Q})\right] &= \mathbb{P}\left[\bar{\psi} \notin \mathsf{C}^0\big(\mathcal{N}_\epsilon(\boldsymbol{P}^*)\big)\right] \\ &\leq 4 \tfrac{n+k}{\sqrt{k}} \epsilon^{\frac{k}{k+1}}. \end{aligned}$$

Proof. First, we can assume $\boldsymbol{P}^* = \operatorname{bdiag}(\mathbf{I}_k, \mathbf{0}_{n-k \times n-k})$ from the rotational invariance of $\boldsymbol{a}$ and $\boldsymbol{b}$. Second, we note that since $\bar{\psi}$ is the sign of a linear functional, it fails to be continuous on $\mathcal{N}_\epsilon(\boldsymbol{P}^*)$ if and only if

there exist $\boldsymbol{P}, \boldsymbol{Q} \in \mathcal{N}_\epsilon(\boldsymbol{P}^*)$ such that $\bar{\psi}(\boldsymbol{P}) \neq \bar{\psi}(\boldsymbol{Q})$. Therefore,

$$\begin{aligned} p_\epsilon :&= \mathbb{P}\big[\bar{\psi} \notin \mathsf{C}^0\big(\mathcal{N}_\epsilon(\boldsymbol{P}^*)\big)\big] \\ &= \mathbb{P}\big[\exists \boldsymbol{P}, \boldsymbol{Q} \in \mathcal{N}_\epsilon(\boldsymbol{P}^*) : \bar{\psi}(\boldsymbol{P}) \neq \bar{\psi}(\boldsymbol{Q})\big] \\ &= \mathbb{P}\big[\exists \boldsymbol{P}, \boldsymbol{Q} \in \mathcal{N}_\epsilon(\boldsymbol{P}^*) : \operatorname{sign}(\boldsymbol{a}^\top(\boldsymbol{P}\boldsymbol{b})) \neq \operatorname{sign}(\boldsymbol{a}^\top(\boldsymbol{Q}\boldsymbol{b}))\big]. \end{aligned}$$

Given a parameter $\rho > 0$ to be fixed later, let us define the event

$$\mathcal{E}_\rho = \big\{ \|\boldsymbol{P}^*\boldsymbol{b}\| > \rho\|\boldsymbol{b}\| \big\},$$

which also defines $\boldsymbol{b}'$, the distribution of $\boldsymbol{b}|\mathcal{E}_\rho$. Therefore, by the laws of total probability and expectations,

$$\begin{aligned} p_\epsilon &= \mathbb{P}\big[\exists \boldsymbol{P}, \boldsymbol{Q} \in \mathcal{N}_\epsilon(\boldsymbol{P}^*) : \operatorname{sign}(\boldsymbol{a}^\top(\boldsymbol{P}\boldsymbol{b})) \neq \operatorname{sign}(\boldsymbol{a}^\top(\boldsymbol{Q}\boldsymbol{b}))|\mathcal{E}_\rho\big]\mathbb{P}[\mathcal{E}_\rho] \\ &\quad + \mathbb{P}\big[\exists \boldsymbol{P}, \boldsymbol{Q} \in \mathcal{N}_\epsilon(\boldsymbol{P}^*) : \operatorname{sign}(\boldsymbol{a}^\top(\boldsymbol{P}\boldsymbol{b})) \neq \operatorname{sign}(\boldsymbol{a}^\top(\boldsymbol{Q}\boldsymbol{b}))|\mathcal{E}_\rho^c\big]\mathbb{P}[\mathcal{E}_\rho^c] \\ &\leq \mathbb{P}\big[\exists \boldsymbol{P}, \boldsymbol{Q} \in \mathcal{N}_\epsilon(\boldsymbol{P}^*) : \operatorname{sign}(\boldsymbol{a}^\top(\boldsymbol{P}\boldsymbol{b})) \neq \operatorname{sign}(\boldsymbol{a}^\top(\boldsymbol{Q}\boldsymbol{b}))|\mathcal{E}_\rho\big] + \mathbb{P}[\mathcal{E}_\rho^c] \\ &= \mathbb{E}_{\boldsymbol{b}'}\mathbb{P}_{\boldsymbol{a}}\big[\exists \boldsymbol{P}, \boldsymbol{Q} \in \mathcal{N}_\epsilon(\boldsymbol{P}^*) : \operatorname{sign}(\boldsymbol{a}^\top(\boldsymbol{P}\boldsymbol{b}')) \neq \operatorname{sign}(\boldsymbol{a}^\top(\boldsymbol{Q}\boldsymbol{b}'))|\boldsymbol{b}'\big] \\ &\quad + \mathbb{P}[\mathcal{E}_\rho^c]. \end{aligned}$$

However, if $\boldsymbol{P} \in \mathcal{N}_\epsilon(\boldsymbol{P}^*)$, then the vectors $\boldsymbol{\beta} = \boldsymbol{P}\boldsymbol{b}'$ and $\boldsymbol{\beta}^* = \boldsymbol{P}^*\boldsymbol{b}'$ are at most $\epsilon' := \epsilon\|\boldsymbol{b}'\|$ far apart since

$$\|\boldsymbol{\beta} - \boldsymbol{\beta}^*\| = \|(\boldsymbol{P} - \boldsymbol{P}^*)\boldsymbol{b}'\| \leq \|\boldsymbol{P} - \boldsymbol{P}^*\|_F\|\boldsymbol{b}'\| \leq \epsilon\|\boldsymbol{b}'\|,$$

while, similarly, $\gamma = \boldsymbol{Q}\boldsymbol{b}'$ is at most ϵ' far apart from $\boldsymbol{\beta}^*$. Therefore,

$$\begin{aligned} p_\epsilon &\leq \mathbb{E}_{\boldsymbol{b}'}\mathbb{P}_{\boldsymbol{a}}\big[\exists \boldsymbol{\beta}, \gamma \in \mathbb{B}^2_{\epsilon'}(\boldsymbol{\beta}^*) : \operatorname{sign}(\boldsymbol{a}^\top\boldsymbol{\beta}) \neq \operatorname{sign}(\boldsymbol{a}^\top\gamma)|\boldsymbol{b}'\big] + \mathbb{P}[\mathcal{E}_\rho^c] \\ &\leq \mathbb{E}_{\boldsymbol{b}'}\mathbb{P}_{\boldsymbol{a}}\big[\bar{\psi} \notin \mathsf{C}^0(\mathbb{B}^2_{\epsilon'}(\boldsymbol{\beta}^*))|\boldsymbol{b}'\big] + \mathbb{P}[\mathcal{E}_\rho^c], \end{aligned}$$

where $\bar{\psi} : \boldsymbol{x} \in \mathbb{R}^n \mapsto \operatorname{sign}(\boldsymbol{a}^\top\boldsymbol{x})$, and $\mathbb{B}^2_{\epsilon'}(\boldsymbol{u})$ is the ℓ_2-ball of radius ϵ' around a vector $\boldsymbol{u}$.

Let us observe that $\mathbb{B}^2_{\epsilon'}(\boldsymbol{\beta}^*)$ is contained in a cone with a half aperture α respecting

$$\sin\alpha = \tfrac{\epsilon'}{\|\boldsymbol{\beta}^*\|} = \tfrac{\epsilon\|\boldsymbol{b}'\|}{\|\boldsymbol{P}^*\boldsymbol{b}'\|} \leq \tfrac{\epsilon}{\rho},$$

since the support of $\boldsymbol{b}'$ is fixed by $\mathcal{E}_\rho$. Therefore, from Lem. 4.1, provided that $\epsilon < \rho$, we have

$$\mathbb{E}_{\boldsymbol{b}'}\mathbb{P}_{\boldsymbol{a}}\big[\bar{\psi} \notin \mathsf{C}^0(\mathbb{B}^2_{\epsilon'}(\boldsymbol{\beta}^*))|\boldsymbol{b}'\big] \leq \sqrt{\tfrac{2}{\pi}}\frac{\epsilon\rho^{-1}}{(1-\epsilon^2\rho^{-2})^{1/2}}\sqrt{n}.$$

Therefore,

$$p_\epsilon \leq \sqrt{\tfrac{2}{\pi}}\tfrac{\epsilon\rho^{-1}}{(1-\epsilon^2\rho^{-2})^{1/2}}\sqrt{n} + \mathbb{P}[\mathcal{E}_\rho^c]. \tag{4.8}$$

Let us now bound $\mathbb{P}[\mathcal{E}_\rho^c]$. Using the structure of $\boldsymbol{P}^*$, we have

$$\begin{aligned}\mathbb{P}[\mathcal{E}_\rho^c] &= \mathbb{P}\big[\textstyle\sum_{i=1}^k b_i^2 \leq \rho^2 \|\boldsymbol{b}\|^2\big] \\ &= \mathbb{P}\big[\textstyle\sum_{i=1}^k b_i^2 \leq \frac{\rho^2}{1-\rho^2} \sum_{i=k+1}^n b_i^2\big] \\ &= \mathbb{P}\big[X^2 \leq \tfrac{\rho^2}{1-\rho^2} Y^2\big],\end{aligned}$$

where X^2 and Y^2 are two independent χ^2-distributions with k and $n-k$ degrees of freedom, respectively. Again by the law of total expectation,

$$\mathbb{P}[\mathcal{E}_\rho^c] = \mathbb{E}_{Y^2}\mathbb{P}_{X^2}\big[X^2 \leq \tfrac{\rho^2}{1-\rho^2} Y^2\big]. \tag{4.9}$$

For any $\lambda > 0$,

$$\mathbb{P}(X^2 \leq \lambda^2) = (2^{k/2}\Gamma(k/2))^{-1} \int_0^{\lambda^2} t^{k/2-1} e^{-t/2} \mathrm{d}t.$$

Since

$$\textstyle\int_0^{\lambda^2} t^{k/2-1} e^{-t/2} \mathrm{d}t \leq \frac{2}{k}\lambda^k,$$

this shows that

$$\mathbb{P}(X^2 \leq \lambda^2) \leq 2^{-k/2}\Gamma(k/2+1)^{-1}\lambda^k.$$

With Stirling's bound

$$\Gamma(k/2+1) \geq \left(\frac{k}{2e}\right)^{k/2},$$

we get

$$\mathbb{P}(X^2 \leq \lambda^2) \leq \left(\frac{e}{k}\right)^{k/2} \lambda^k.$$

Injecting this bound in Eq. 4.9 with

$$\lambda = \frac{\rho}{\sqrt{1-\rho^2}}\sqrt{Y^2},$$

we get

$$\mathbb{P}[\mathcal{E}_\rho^c] \leq \left(\tfrac{e}{k}\right)^{k/2} \left(\tfrac{\rho^2}{1-\rho^2}\right)^{k/2} \mathbb{E}(Y^2)^{k/2}.$$

Since, by Jensen and from the moments of a χ^2_{n-k}-distribution,

$$(\mathbb{E}(Y^2)^{k/2})^2 \leq \mathbb{E}(Y^2)^k = \prod_{j=1}^k (n+k-2j),$$

we get the crude bound

$$\mathbb{E}(Y^2)^{k/2} \leq (n+k)^{k/2}.$$

Therefore

$$\mathbb{P}[\mathcal{E}_\rho^c] \leq \left(\tfrac{e(n+k)\rho^2}{k(1-\rho^2)}\right)^{k/2}. \tag{4.10}$$

We now set

$$\rho^2 = \epsilon^{\frac{2}{k+1}} \tfrac{k}{3e(n+k)} < \tfrac{1}{3}.$$

As observed above, we must impose the condition $\epsilon < \rho$. Since

$$\epsilon < \rho \quad \iff \quad \epsilon^2 < \rho^2 \quad \iff \quad \epsilon^2 < \epsilon^{\frac{2}{k+1}} \frac{k}{3e(n+k)},$$

this is equivalent to

$$\epsilon^{2-\frac{2}{k+1}} < \frac{k}{3e(n+k)}.$$

Since

$$2 - \frac{2}{k+1} = \frac{2k}{k+1},$$

this is met whenever we impose the stronger condition

$$\epsilon < \left(\tfrac{k}{9e(n+k)}\right)^{\frac{k+1}{2k}}. \tag{4.11}$$

With this setting, Eq. 4.10 gives

$$\begin{aligned}
\mathbb{P}[\mathcal{E}_\rho^c] &\leq \left(\tfrac{e(n+k)\rho^2}{k(1-\rho^2)}\right)^{k/2} \\
&= \left(\tfrac{e(n+k)}{k(1-\rho^2)} \epsilon^{\frac{2}{k+1}} \tfrac{k}{3e(n+k)}\right)^{k/2} \\
&= \left(\tfrac{\epsilon^{2/(k+1)}}{3(1-\rho^2)}\right)^{k/2}.
\end{aligned}$$

Since $\rho^2 < 1/3$, we have $1 - \rho^2 > 2/3$, hence $\frac{1}{3(1-\rho^2)} \leq \frac{1}{2}$. Therefore,

$$\mathbb{P}[\mathcal{E}_\rho^c] \leq \epsilon^{\frac{k}{k+1}} \left(\tfrac{1}{2}\right)^{k/2} \leq \epsilon^{\frac{k}{k+1}}.$$

Moreover, from the choice of ρ, we have

$$\tfrac{\epsilon}{\rho} = \epsilon^{\frac{k}{k+1}} \sqrt{\tfrac{3e(n+k)}{k}}.$$

Now, by Eq. 4.11,

$$\epsilon^{\frac{2k}{k+1}} < \frac{k}{9e(n+k)},$$

so that

$$\epsilon^{\frac{k}{k+1}} < \sqrt{\frac{k}{9e(n+k)}}.$$

Injecting this in the previous expression gives

$$\frac{\epsilon}{\rho} = \epsilon^{\frac{k}{k+1}} \sqrt{\frac{3e(n+k)}{k}} \leq \frac{1}{\sqrt{3}}.$$

Hence

$$\frac{\epsilon^2}{\rho^2} \leq \frac{1}{3}, \qquad 1 - \frac{\epsilon^2}{\rho^2} \geq \frac{2}{3}, \qquad \frac{1}{\sqrt{1-\epsilon^2\rho^{-2}}} \leq \sqrt{\frac{3}{2}}.$$

Gathering all the bounds provides

$$\begin{aligned} p_\epsilon &\leq \sqrt{\frac{2}{\pi}} \frac{\epsilon\rho^{-1}}{(1-\epsilon^2\rho^{-2})^{1/2}} \sqrt{n} + \epsilon^{\frac{k}{k+1}} \\ &\leq \sqrt{\frac{2}{\pi}} \sqrt{\frac{3}{2}} \epsilon^{\frac{k}{k+1}} \sqrt{\frac{3e(n+k)}{k}} \sqrt{n} + \epsilon^{\frac{k}{k+1}} \\ &= \sqrt{\frac{9e}{\pi}} \epsilon^{\frac{k}{k+1}} \frac{\sqrt{n(n+k)}}{\sqrt{k}} + \epsilon^{\frac{k}{k+1}}. \end{aligned}$$

Since $\sqrt{n(n+k)} \leq n+k$, we get

$$p_\epsilon \leq \sqrt{\frac{9e}{\pi}} \epsilon^{\frac{k}{k+1}} \frac{n+k}{\sqrt{k}} + \epsilon^{\frac{k}{k+1}}.$$

Finally, observing that $\sqrt{9e/\pi} < 3$ and that $(n+k)/\sqrt{k} \geq 1$, we obtain

$$\begin{aligned} p_\epsilon &\leq 3\epsilon^{\frac{k}{k+1}} \frac{n+k}{\sqrt{k}} + \epsilon^{\frac{k}{k+1}} \\ &\leq 4\epsilon^{\frac{k}{k+1}} \frac{n+k}{\sqrt{k}}. \end{aligned}$$

Finally, if Eq. 4.11 does not hold, then the right-hand side above exceeds 1, showing the vacuity of the bound on p_ϵ; this allows us to forget the condition on ϵ. □

A simple rescaling allows us to simplify the last lemma to get the following corollary.

Corollary 4.1 (Local stability of the binary embedding (Simplified))**.** *Under the conventions and conditions of Lem. 4.2, for $\delta > 0$, we have*

$$\mathbb{P}\Big[\bar{\psi} \notin \mathsf{C}^0\big(\mathcal{N}_{\eta(\delta)}(\boldsymbol{P}^*)\big)\Big] \leq \delta, \qquad \eta(\delta) := \Big(\frac{\sqrt{k}}{4(n+k)}\delta\Big)^{\frac{k+1}{k}}.$$

Proof. By Lem. 4.2, for every $\epsilon > 0$, we have

$$\mathbb{P}\Big[\bar{\psi} \notin \mathsf{C}^0\big(\mathcal{N}_\epsilon(\boldsymbol{P}^*)\big)\Big] \leq 4\tfrac{n+k}{\sqrt{k}}\epsilon^{\frac{k}{k+1}}.$$

We now choose $\epsilon = \eta(\delta)$, with

$$\eta(\delta) := \Big(\tfrac{\sqrt{k}}{4(n+k)}\delta\Big)^{\frac{k+1}{k}}.$$

Since

$$\eta(\delta)^{\frac{k}{k+1}} = \Big(\tfrac{\sqrt{k}}{4(n+k)}\delta\Big)^{\frac{k+1}{k}\frac{k}{k+1}} = \tfrac{\sqrt{k}}{4(n+k)}\delta,$$

we obtain

$$4\tfrac{n+k}{\sqrt{k}}\eta(\delta)^{\frac{k}{k+1}} = 4\tfrac{n+k}{\sqrt{k}}\tfrac{\sqrt{k}}{4(n+k)}\delta = \delta.$$

Therefore,

$$\mathbb{P}\Big[\bar{\psi} \notin \mathsf{C}^0\big(\mathcal{N}_{\eta(\delta)}(\boldsymbol{P}^*)\big)\Big] \leq \delta,$$

which proves the claim. □

Lem. 4.2 and Cor. 4.1 give the probability of sign flip for one element in the measurement vector when $\boldsymbol{P}$ is moved around within ϵ. The next lemma will further characterise the probability of having at most a certain number of flips in the full measurements vector for a small perturbation of $\boldsymbol{P}$.

Lemma 4.3. *Under the conventions introduced in Lem. 4.2, given an integer m, we consider m random maps $\bar{\psi}_i : \boldsymbol{P} \in \mathbb{G}(k,n) \mapsto \operatorname{sign}(\boldsymbol{a}_i^\top \boldsymbol{P}\boldsymbol{b}_i)$ with $\boldsymbol{a}_i, \boldsymbol{b}_i \sim_{\text{i.i.d.}} \mathcal{N}(\boldsymbol{0}, \mathbf{I}_n)$, $1 \leq i \leq m$. Given $\delta > 0$, the function $\eta(\delta)$ defined in Cor. 4.1, and $\boldsymbol{P}^* \in \mathbb{G}(k,n)$, we have*

$$\mathbb{P}\Big[\big|\{i : \bar{\psi}_i \notin \mathsf{C}^0\big(\mathcal{N}_{\eta(\delta)}(\boldsymbol{P}^*)\big)\}\big| > 2\delta m\Big] \leq C\exp(-c\delta^2 m),$$

for absolute constants $C, c > 0$.

Proof. For each $1 \leq i \leq m$, define the binary random variable $Z_i = \mathbb{1}\big[\bar{\psi}_i \notin \mathsf{C}^0\big(\mathcal{N}_{\eta(\delta)}(\boldsymbol{P}^*)\big)\big]$, with $\mathbb{1}[\mathcal{E}]$ being equal to 1 if the event $\mathcal{E}$ is true, and to 0 otherwise. By Cor. 4.1, we have $p_\delta := \mathbb{E}Z_i \leq \delta$. The random variables Z_i are independent and bounded, hence sub-Gaussian. Therefore, from [148, Prop. 2.6.1 and Theorem 2.6.2], for any $\rho > 0$,

$$\begin{aligned}\mathbb{P}[\textstyle\sum_{i=1}^m Z_i > \delta m + \rho m] &\leq \mathbb{P}[\sum_{i=1}^m Z_i \geq m p_\delta + \rho m] \\ &\leq C\exp(-c\rho^2 m),\end{aligned}$$

for some absolute constants $C, c > 0$. Choosing $\rho = \delta$ yields

$$\mathbb{P}[\sum_{i=1}^{m} Z_i > 2\delta m] \leq C \exp(-c\delta^2 m).$$

□

In addition to the previous stability properties, our proof also uses the possibility to *cover* the space of projectors.

Proposition 4.5 (Covering $\mathbb{G}(k,n)$)**.** *Given $\epsilon > 0$, there exists a covering $\mathcal{N}_\epsilon \subset \mathbb{G}(k,n)$ of $\mathbb{G}(k,n)$,* i.e.*, such that*

$$\forall \boldsymbol{P} \in \mathbb{G}(k,n),\ \exists \boldsymbol{P}^* \in \mathcal{N}_\epsilon :\ \|\boldsymbol{P} - \boldsymbol{P}^*\|_F \leq \epsilon,$$

whose cardinality is bounded by

$$|\mathcal{N}_\epsilon| \leq (18\sqrt{k}/\epsilon)^{(2n+1)k}.$$

Proof. The set $\mathbb{G}(k,n)$ is included in the set

$$\mathcal{R}^F(k,n) := \mathcal{R}(k,n) \cap \sqrt{k}\mathbb{B}_F^{n\times n}$$

of $n \times n$ rank-k matrices with Frobenius norm bounded by $\sqrt{k}$ (since if $\boldsymbol{P} \in \mathbb{G}(k,n)$, $\mathrm{rank}(\boldsymbol{P}) = k$ and $\|\boldsymbol{P}\|_F = \sqrt{k}$), we can cover $\mathbb{G}(k,n)$ from a covering of $\mathcal{R}^F(k,n)$. Indeed, given two compact sets $\mathcal{A} \subset \mathcal{B}$ in a metric space $(\mathcal{X}, d)$ with distance d, if $\mathcal{B}_{\epsilon/2} \subset \mathcal{B}$ is an $\epsilon/2$-covering of $\mathcal{B}$, *i.e.*, such that for any $\boldsymbol{x} \in \mathcal{B}$ there is a $\boldsymbol{x}' \in \mathcal{B}_{\epsilon/2}$ such that $d(\boldsymbol{x}, \boldsymbol{x}') \leq \epsilon/2$, then we can build an ϵ-covering $\mathcal{A}_\epsilon \subset \mathcal{A}$ of $\mathcal{A}$ with $|\mathcal{A}_\epsilon| \leq |\mathcal{B}_{\epsilon/2}|$ (see *e.g.*, [148, Chap. 4]). From [35, Lem. 3.1], there exists an $\epsilon/2$-covering of $\mathcal{R}(k,n) \cap \mathbb{B}_F^{n\times n}$, with d set to the Frobenius distance, whose cardinality is bounded by $(18/\epsilon)^{(2n+1)k}$. This means that, by rescaling the set diameter by $\sqrt{k}$, there exists an $\epsilon/2$-covering of $\mathcal{R}^F(k,n)$ with cardinality bound $(18\sqrt{k}/\epsilon)^{(2n+1)k}$, and thus, from the considerations above, there exists an ϵ-covering $\mathcal{N}_\epsilon \subset \mathbb{G}(k,n)$ with

$$|\mathcal{N}_\epsilon| \leq (18\sqrt{k}/\epsilon)^{(2n+1)k}.$$

□

We can now use previous lemmas to provide the final proof for Prop. 4.4 that we restate here for simplicity.

Proposition (Prop. 4.4 (restated)). *Let $\delta > 0$ and $C, C', c > 0$ be absolute constants. If*

$$m \geq C\delta^{-2}nk\log(\tfrac{n^2}{k\delta^2}),$$

then, with probability exceeding $1 - C'\exp(-c\delta^2 m)$,

$$\sup_{\boldsymbol{U},\boldsymbol{V}\in\mathbb{V}(k,n)} \left|\widehat{\kappa}^{\pm}(\boldsymbol{U},\boldsymbol{V}) - \kappa^{\pm}(\boldsymbol{U},\boldsymbol{V})\right| \leq \delta.$$

Proof. Given a radius $\epsilon > 0$ to be fixed later, we know from Prop. 4.5 that there exists an ϵ-covering $\mathcal{N}_\epsilon \subset \mathbb{G}(k,n)$ of $\mathbb{G}(k,n)$ with cardinality $|\mathcal{N}_\epsilon| \leq (18\sqrt{k}/\epsilon)^{(2n+1)k}$. We also consider the set $\mathcal{V}_\epsilon \subset \mathbb{V}(k,n)$ such that for any $\boldsymbol{U}^* \in \mathcal{V}_\epsilon$, $\boldsymbol{P}^* := \boldsymbol{U}^*\boldsymbol{U}^{*\top} \in \mathcal{N}_\epsilon$, and where we ensure that each $\boldsymbol{P}^*$ is represented by only one basis in $\mathcal{V}_\epsilon$, *i.e.*, $|\mathcal{V}_\epsilon| = |\mathcal{N}_\epsilon|$. This does not mean, however, that $\mathcal{V}_\epsilon$ is a covering of $\mathbb{V}(k,n)$.

The general objective of this proof is thus to upper bound, with controlled probability, the approximation error $|\widehat{\kappa}^{\pm}(\boldsymbol{U},\boldsymbol{V}) - \kappa^{\pm}(\boldsymbol{U},\boldsymbol{V})|$ for all pairs of subspace bases $\boldsymbol{U},\boldsymbol{V} \in \mathbb{V}(k,n)$. Let us first observe that, given $\boldsymbol{P}^* = \boldsymbol{U}^*\boldsymbol{U}^{*\top}$ and $\boldsymbol{Q}^* = \boldsymbol{V}^*\boldsymbol{V}^{*\top}$ the closest projectors in $\mathcal{N}_\epsilon$ to $\boldsymbol{P} = \boldsymbol{U}\boldsymbol{U}^\top$ and $\boldsymbol{Q} = \boldsymbol{V}\boldsymbol{V}^\top$, we can upper bound this error with 3 terms and target the following objective bound:

$$\left|\widehat{\kappa}^{\pm}(\boldsymbol{U},\boldsymbol{V}) - \kappa^{\pm}(\boldsymbol{U},\boldsymbol{V})\right| \leq T_1 + T_2 + T_3 \leq \delta,$$

where $m\widehat{\kappa}^{\pm}(\boldsymbol{U},\boldsymbol{V}) = \langle \psi^{\pm}(\boldsymbol{U}), \psi^{\pm}(\boldsymbol{V})\rangle$, $\kappa^{\pm}(\boldsymbol{U},\boldsymbol{V}) = \mathbb{E}\widehat{\kappa}^{\pm}(\boldsymbol{U},\boldsymbol{V})$, and the three terms

$$T_1 = \left|\widehat{\kappa}^{\pm}(\boldsymbol{U},\boldsymbol{V}) - \widehat{\kappa}^{\pm}(\boldsymbol{U}^*,\boldsymbol{V}^*)\right|, \quad T_2 = \left|\widehat{\kappa}^{\pm}(\boldsymbol{U}^*,\boldsymbol{V}^*) - \kappa^{\pm}(\boldsymbol{U}^*,\boldsymbol{V}^*)\right|,$$
$$T_3 = \left|\kappa^{\pm}(\boldsymbol{U}^*,\boldsymbol{V}^*) - \kappa^{\pm}(\boldsymbol{U},\boldsymbol{V})\right|.$$

We now handle these three terms separately, enforcing each of them to be at most $\delta/3$. The two first terms will be bounded conditionally to two probabilistic events (defined below), respectively, $\mathcal{E}_1$ and $\mathcal{E}_2$, while the last term is deterministic. By a union bound, the final approximation error will thus have a probability failure summing the probabilities of failure of the two first events.

(a) Bound on T_1: Given the random vectors $\boldsymbol{a}_i, \boldsymbol{b}_i \sim_{\text{i.i.d.}} \mathcal{N}(\boldsymbol{0}, \mathbf{I}_n)$ and the mapping $\bar{\psi}_i : \boldsymbol{P} \in \mathbb{G}(k,n) \mapsto \text{sign}(\boldsymbol{a}_i^\top \boldsymbol{P}\boldsymbol{b}_i)$, $1 \leq i \leq m$, defining the

feature map $\psi^{\pm}$, the first term T_1 can be upper bounded by

$$\begin{aligned}
T_1 &\leq \tfrac{1}{m}\textstyle\sum_{i=1}^{m} \left|\operatorname{sign}(\boldsymbol{a}_i^\top \boldsymbol{P}\boldsymbol{b}_i)\operatorname{sign}(\boldsymbol{a}_i^\top \boldsymbol{Q}\boldsymbol{b}_i) - \operatorname{sign}(\boldsymbol{a}_i^\top \boldsymbol{P}^*\boldsymbol{b}_i)\operatorname{sign}(\boldsymbol{a}_i^\top \boldsymbol{Q}^*\boldsymbol{b}_i)\right| \\
&= \tfrac{2}{m}\textstyle\sum_{i=1}^{m} \mathbb{1}\left[\bar{\psi}_i(\boldsymbol{P})\bar{\psi}_i(\boldsymbol{Q}) \neq \bar{\psi}_i(\boldsymbol{P}^*)\bar{\psi}_i(\boldsymbol{Q}^*)\right] \\
&\leq \tfrac{2}{m}\textstyle\sum_{i=1}^{m} \mathbb{1}\left[\bar{\psi}_i(\boldsymbol{P}) \neq \bar{\psi}_i(\boldsymbol{P}^*) \text{ or } \bar{\psi}_i(\boldsymbol{Q}) \neq \bar{\psi}_i(\boldsymbol{Q}^*)\right] \\
&\leq \tfrac{2}{m}\textstyle\sum_{i=1}^{m} \mathbb{1}\left[\bar{\psi}_i(\boldsymbol{P}) \neq \bar{\psi}_i(\boldsymbol{P}^*)\right] + \tfrac{2}{m}\textstyle\sum_{i=1}^{m} \mathbb{1}\left[\bar{\psi}_i(\boldsymbol{Q}) \neq \bar{\psi}_i(\boldsymbol{Q}^*)\right] \\
&\leq \tfrac{2}{m}\textstyle\sum_{i=1}^{m} \mathbb{1}\left[\bar{\psi}_i \notin \mathsf{C}^0(\mathcal{N}_\epsilon(\boldsymbol{P}^*))\right] + \tfrac{2}{m}\textstyle\sum_{i=1}^{m} \mathbb{1}\left[\bar{\psi}_i \notin \mathsf{C}^0(\mathcal{N}_\epsilon(\boldsymbol{Q}^*))\right].
\end{aligned}$$

where $\mathcal{N}_\epsilon(\boldsymbol{P}^*) = \mathbb{B}_\epsilon^F(\boldsymbol{P}^*) \cap \mathbb{G}(k,n)$ is an ϵ-ball centred on $\boldsymbol{P}^*$. Defining the first event

$$\mathcal{E}_1 := \{\forall \boldsymbol{P}' \in \mathcal{N}_\epsilon : \left|\{i : \bar{\psi}_i \notin \mathsf{C}^0\big(\mathcal{N}_\epsilon(\boldsymbol{P}')\big)\}\right| \leq \tfrac{1}{12}\delta m\},$$

it is clear that if $\mathcal{E}_1$ holds, then $T_1 \leq \delta/3$, for all possible $\boldsymbol{P}$, $\boldsymbol{Q}$, $\boldsymbol{P}^*$ and $\boldsymbol{Q}^*$.

From Cor. 4.1, setting $\epsilon = \eta(\delta/12)$, we find by union bound over all elements of $\mathcal{N}_{\eta(\delta/12)}$ that, for some $C, c > 0$,

$$\begin{aligned}
\mathbb{P}[\mathcal{E}_1^c] &\leq C|\mathcal{N}_{\eta(\delta/12)}| \exp(-c\delta^2 m) \\
&\leq C \exp\left((2n+1)k \log\left(18\tfrac{\sqrt{k}}{\eta(\delta/12)}\right) - c\delta^2 m\right) \\
&\leq C \exp\left(cnk \log\left(\tfrac{\sqrt{k}}{\eta(\delta/12)}\right) - c'\delta^2 m\right).
\end{aligned}$$

(b) Bound on T_2: Regarding the second term T_2, we can ensure that $T_2 \leq \delta/3$ for all possible $\boldsymbol{P}$, $\boldsymbol{Q}$, $\boldsymbol{P}^*$ and $\boldsymbol{Q}^*$ if this event holds:

$$\mathcal{E}_2 := \left\{\forall \boldsymbol{U}^*, \boldsymbol{V}^* \in \mathcal{V}_\epsilon,\ \left|\tfrac{1}{m}\langle \psi^{\pm}(\boldsymbol{U}^*), \psi^{\pm}(\boldsymbol{V}^*)\rangle - \kappa^{\pm}(\boldsymbol{U}^*, \boldsymbol{V}^*)\right| < \tfrac{\delta}{3}\right\},$$

with $\epsilon = \eta(\delta/12)$. From Prop. 4.3 and a union bound on all possible pairs of bases picked in $\mathcal{V}_\epsilon \times \mathcal{V}_\epsilon$, we know that, for some $C, c, c' > 0$,

$$\begin{aligned}
\mathbb{P}[\mathcal{E}_2^c] &\leq 2|\mathcal{N}_{\eta(\delta/12)}|^2 \exp(-\tfrac{1}{2}m\delta^2) \\
&\leq C \exp\left(cnk \log\left(\frac{\sqrt{k}}{\eta(\delta/12)}\right) - c'\delta^2 m\right).
\end{aligned}$$

(c) Bound on T_3: Finally, regarding T_3, we can bound with several calls

to the triangular inequality, so that, for all possible $\boldsymbol{P}$, $\boldsymbol{Q}$, $\boldsymbol{P}^*$ and $\boldsymbol{Q}^*$,

$$\begin{aligned}
T_3 \leq & \frac{1}{m}\sum_{i=1}^{m}\mathbb{E}\left|\operatorname{sign}(\boldsymbol{a}_i^\top \boldsymbol{P}\boldsymbol{b}_i)\operatorname{sign}(\boldsymbol{a}_i^\top \boldsymbol{Q}\boldsymbol{b}_i) - \operatorname{sign}(\boldsymbol{a}_i^\top \boldsymbol{P}^*\boldsymbol{b}_i)\operatorname{sign}(\boldsymbol{a}_i^\top \boldsymbol{Q}^*\boldsymbol{b}_i)\right| \\
= & \tfrac{1}{m}\textstyle\sum_{i=1}^{m}\mathbb{E}\left[2\mathbb{1}\left[\bar{\psi}_i(\boldsymbol{P})\bar{\psi}_i(\boldsymbol{Q}) \neq \bar{\psi}_i(\boldsymbol{P}^*)\bar{\psi}_i(\boldsymbol{Q}^*)\right]\right] \\
\leq & 2\mathbb{P}\left[\bar{\psi}(\boldsymbol{P})\bar{\psi}(\boldsymbol{Q}) \neq \bar{\psi}(\boldsymbol{P}^*)\bar{\psi}(\boldsymbol{Q}^*)\right] \\
\leq & 2\mathbb{P}\left[\bar{\psi}(\boldsymbol{P}) \neq \bar{\psi}(\boldsymbol{P}^*)\right] + 2\mathbb{P}\left[\bar{\psi}(\boldsymbol{Q}) \neq \bar{\psi}(\boldsymbol{Q}^*)\right] \\
= & 2\mathbb{P}\left[\bar{\psi} \notin \mathsf{C}^0\left(\mathcal{N}_{\eta(\delta/12)}(\boldsymbol{P}^*)\right)\right] + 2\mathbb{P}\left[\bar{\psi} \notin \mathsf{C}^0\left(\mathcal{N}_{\eta(\delta/12)}(\boldsymbol{Q}^*)\right)\right],
\end{aligned}$$

with $\bar{\psi} : \boldsymbol{P} \in \mathbb{G}(k,n) \mapsto \operatorname{sign}(\boldsymbol{a}^\top \boldsymbol{P}\boldsymbol{b})$ for $\boldsymbol{a},\boldsymbol{b} \sim \mathcal{N}(\boldsymbol{0},\mathbf{I}_n)$. Finally, Cor. 4.1 allows us to conclude that

$$\begin{aligned}
T_3 &\leq 2\mathbb{P}\left[\bar{\psi} \notin \mathsf{C}^0\left(\mathcal{N}_{\eta(\delta/12)}(\boldsymbol{P}^*)\right)\right] + 2\mathbb{P}\left[\bar{\psi} \notin \mathsf{C}^0\left(\mathcal{N}_{\eta(\delta/12)}(\boldsymbol{Q}^*)\right)\right] \\
&\leq 4\delta/12 = \delta/3.
\end{aligned}$$

Final bound: Gathering the three terms T_1, T_2, and T_3, we can finally state that

$$\sup_{\boldsymbol{U},\boldsymbol{V}\in\mathbb{V}(k,n)} \left|\widehat{\kappa}^{\pm}(\boldsymbol{U},\boldsymbol{V}) - \kappa^{\pm}(\boldsymbol{U},\boldsymbol{V})\right| \leq \delta,$$

with a failure probability given by

$$p_{\text{fail}} = \mathbb{P}(\mathcal{E}_1^c) + \mathbb{P}(\mathcal{E}_2^c) \leq C\exp\left(cnk\log\left(\tfrac{\sqrt{k}}{\eta(\delta/12)}\right) - c'\delta^2 m\right).$$

Therefore, imposing

$$m \geq C\delta^{-2}nk\log\left(\tfrac{\sqrt{k}}{\eta(\delta/12)}\right),$$

we get

$$p_{\text{fail}} \leq C\exp(-c\delta^2 m).$$

Since, using the expression of η,

$$\begin{aligned}
\log\left(\tfrac{\sqrt{k}}{\eta(\delta/12)}\right) &= \log(\sqrt{k}) + \tfrac{k+1}{k}\log\left(\tfrac{48(n+k)}{\sqrt{k}\delta}\right) \\
&\leq C\log\left(\tfrac{n^2}{k\delta^2}\right),
\end{aligned}$$

the condition on m simplifies into the stronger condition

$$m \geq C\delta^{-2}nk\log\left(\tfrac{n^2}{k\delta^2}\right).$$

□

4.3.3 The one-dimensional subspace case

When $k = 1$, we are faced with a particularly interesting special case. Indeed, working in $\mathbb{G}(1, n)$ is akin to working with one-dimensional subspaces of $\mathbb{R}^n$, *i.e.*, with lines through the origin. Equivalently, such a subspace can be represented by a unit vector $\boldsymbol{u} \in \mathrm{S}^{n-1}$ (up to a sign), and two such subspaces are entirely characterised by their single principal angle. In this regime, the Grassmannian geometry reduces to a purely angular geometry on vectors, which makes the behaviour of the kernel $\kappa^{\pm}$ much easier to interpret.

The case $k = 1$ also connects this chapter with Chap. 3, where we studied quadratic embeddings of vectors and used them to estimate quantities of the form $\langle \boldsymbol{u}, \boldsymbol{x} \rangle^2$ from random quadratic measurements. The sign product embedding led to an asymmetric kernel interpretation, presented in Sec. 3.4, where $\boldsymbol{x}$ is represented through a quadratic embedding and $\boldsymbol{u}$ through a binarised version of the same embedding. When $k = 1$, the Grassmannian setting gives a symmetric version of the same idea. Both arguments now represent one-dimensional subspaces, and the binary embedding induces a genuine symmetric kernel depending only on the angle between them.

A second reason why this special case deserves to be isolated is that, unlike the general situation, the kernel $\kappa^{\pm}$ admits here a closed-form expression. More precisely, as shown in Prop. 4.6, it becomes a simple quadratic function of the principal angle between the two lines. This explicit formula gives a concrete interpretation of the binary Grassmannian kernel. Moreover, since the general uniform concentration result developed in Prop. 4.4 applies for all $k \geq 1$, this closed-form kernel also benefits from the same concentration guarantees, so that its empirical approximation $\widehat{\kappa}^{\pm}$ remains uniformly controlled with high probability.

We now show the closed form of $\kappa^{\pm}$ when $k = 1$.

Proposition 4.6 (Expectation of the 1-d case)**.** *For two vectors $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^n$ such that $\|\boldsymbol{u}\| = \|\boldsymbol{v}\| = 1$, with $\boldsymbol{P} = \boldsymbol{u}\boldsymbol{u}^\top, \boldsymbol{Q} = \boldsymbol{v}\boldsymbol{v}^\top$, and for $\widehat{\kappa}^{\pm}(\boldsymbol{U}, \boldsymbol{V})$ defined as earlier, we have*

$$\mathbb{E}\left[\widehat{\kappa}^{\pm}(\boldsymbol{u}, \boldsymbol{v})\right] = \left(1 - \tfrac{2\theta}{\pi}\right)^2, \tag{4.12}$$

where θ is the single principal angle between $\boldsymbol{P}$ and $\boldsymbol{Q}$ (or more simply, the angle between $\boldsymbol{u}$ and $\boldsymbol{v}$).

Proof. For $k = 1$, the projections of $\boldsymbol{g} \sim \mathcal{N}(\boldsymbol{0}, \mathbf{I}_n)$ onto the two subspaces are $\boldsymbol{P}\boldsymbol{g} = a\boldsymbol{u}$ and $\boldsymbol{Q}\boldsymbol{g} = b\boldsymbol{v}$, where $a = \boldsymbol{u}^\top \boldsymbol{g}$ and $b = \boldsymbol{v}^\top \boldsymbol{g}$ are correlated

Gaussian random variables with $\mathbb{E}[a] = \mathbb{E}[b] = 0$ and $\mathbb{E}[ab] = \cos\theta$. The angle between $\boldsymbol{Pg}$ and $\boldsymbol{Qg}$ is therefore θ if and only if $ab > 0$ and $\pi - \theta$ if $ab < 0$. Let $p = \mathbb{P}(ab > 0)$, then

$$\mathbb{E}[\measuredangle(\boldsymbol{Pg}, \boldsymbol{Qg})] = \theta p + (\pi - \theta)(1 - p).$$

On the other hand, since $\operatorname{sign}(a)\operatorname{sign}(b) = 1$ if $ab > 0$ and -1 otherwise, we have

$$\mathbb{E}[\operatorname{sign}(a)\operatorname{sign}(b)] = 2p - 1.$$

For correlated Gaussians with correlation $\cos\theta$, we have

$$\mathbb{E}[\operatorname{sign}(a)\operatorname{sign}(b)] = \tfrac{2}{\pi}\arcsin(\cos\theta) = 1 - \tfrac{2\theta}{\pi}$$

(see [130], p.230). Hence $2p - 1 = 1 - \frac{2\theta}{\pi}$ and therefore $p = 1 - \frac{\theta}{\pi}$. Substituting in the expectation above gives

$$\mathbb{E}\measuredangle(\boldsymbol{Pg}, \boldsymbol{Qg}) = \theta\big(1 - \tfrac{\theta}{\pi}\big) + (\pi - \theta)\tfrac{\theta}{\pi}.$$

Inserting this quantity into Eq. 4.7 of Prop. 4.2, we obtain

$$\kappa^{\pm}(\boldsymbol{U}, \boldsymbol{V}) = 1 - \tfrac{2}{\pi}\mathbb{E}\measuredangle(\boldsymbol{Pg}, \boldsymbol{Qg}) = \big(1 - \tfrac{2\theta}{\pi}\big)^2.$$

□

Remark 4.1 (Geometric interpretation of Prop. 4.6)**.** *Geometrically, this expression can be understood by thinking about hyperplane separation. For $k = 1$, one coordinate of the binary embedding can be written as*

$$\operatorname{sign}(\boldsymbol{a}^\top \boldsymbol{P}\boldsymbol{b}) = \operatorname{sign}(\boldsymbol{a}^\top \boldsymbol{u})\operatorname{sign}(\boldsymbol{b}^\top \boldsymbol{u}).$$

Hence, each binary ROP feature combines two independent binary comparisons with the random vectors $\boldsymbol{a}$ and $\boldsymbol{b}$. For two lines forming a single principal angle θ, a random hyperplane separates these lines with probability θ/π. A single comparison with random vectors therefore has expected product of the two binary outputs equal to $1 - 2\theta/\pi$. Since the two tests entering the ROP feature are independent, their contributions multiply, giving the squared angular similarity $(1 - 2\theta/\pi)^2$. The kernel is therefore equal to 1 for identical subspaces and decreases to 0 when the two vectors become orthogonal.

The following corollary specialises the general uniform concentration result to the rank-one case, where the kernel admits the explicit expression given in Prop. 4.6.

Corollary 4.2 (Uniform approximation in the rank-one case)**.** *Let $\delta > 0$ and $C, C', c > 0$ be absolute constants. If*

$$m \geq C\delta^{-2} n \log\left(\tfrac{n^2}{\delta^2}\right),$$

then, with probability exceeding $1 - C' \exp(-c\delta^2 m)$,

$$\sup_{\boldsymbol{u},\boldsymbol{v}\in\mathbb{R}^n\setminus\{\boldsymbol{0}\}} \left|\widehat{\kappa}^{\pm}(\boldsymbol{u},\boldsymbol{v}) - \left(1 - \tfrac{2\theta}{\pi}\right)^2\right| \leq \delta,$$

where θ denotes the angle between $\boldsymbol{u}$ and $\boldsymbol{v}$.

Proof. This is a direct consequence of Prop. 4.4 with $k = 1$ together with Prop. 4.6. Moreover, in the rank-one case, both $\widehat{\kappa}^{\pm}(\boldsymbol{u},\boldsymbol{v})$ and $\kappa^{\pm}(\boldsymbol{u},\boldsymbol{v})$ are invariant under the rescaling of $\boldsymbol{u}$ and $\boldsymbol{v}$ by nonzero scalars, so the statement may be written for arbitrary nonzero vectors rather than only for unit vectors. □

In summary, the case $k = 1$ forms a natural bridge between the vector setting of quadratic random embeddings described in Chap. 3 and the Grassmannian kernel perspective developed here. In this rank-one case, the kernel $\kappa^{\pm}$ admits an *explicit* expression as a quadratic function of the angle between two lines. This complements the asymmetric kernel viewpoint from Sec. 3.4 arising from the sign product embedding (Prop. 3.2) in the vector case by showing that, when both arguments are treated symmetrically as one-dimensional subspaces, the same philosophy leads to a genuine symmetric kernel. Finally, this explicit kernel still benefits from the general uniform concentration result, so that its empirical approximation $\widehat{\kappa}^{\pm}$ remains uniformly controlled over $\mathbb{R}^n$ with high probability.

4.4 Aggregated binary features for an explicit kernel

The binary kernel $\kappa^{\pm}$ has two attractive features: it is induced by one-bit random features and it admits uniform approximation guarantees. Its main limitation is analytical. Apart from the case $k = 1$, no closed-form expression of $\kappa^{\pm}$ is known. This makes it harder to compare it directly to known Grassmannian kernels or to interpret precisely how it depends on the principal angles.

We now introduce a modified binary construction whose purpose is to approximately recover an *explicit* Grassmannian kernel. The idea is to aggregate several independent rank-one projections before apply-

ing the sign function. Instead of applying the sign to a single rank-one projection, we apply it to a sum of N_Σ independent copies which creates a smoothing effect. Using the Berry-Esseen theorem, this averaged quantity can be compared to a Gaussian limit. This comparison makes it possible to identify an explicit limiting kernel, up to an approximation error controlled by N_Σ.

This construction should not be viewed as a more efficient replacement for the binary embedding $\psi^{\pm}$, since each feature now requires N_Σ rank-one projections instead of one. It provides an explicit formula that can be used as a reference kernel, or be used directly in settings where a closed-form expression is required, while remaining connected to a one-bit random feature construction, thus making its interest mainly analytical.

We first define the necessary objects, study the validity of the kernel induced by this new embedding, and state Prop. 4.9, which controls the error between the expected kernel and its analytical approximation, in Sec. 4.4.1. The proof of the proposition is then given in Sec. 4.4.2.

4.4.1 Definition and properties

Given an integer $N_\Sigma \geq 1$, we consider the aggregated random feature map

$$\psi^\Sigma : \boldsymbol{U} \in \mathbb{V}(k,n) \mapsto \psi^\Sigma(\boldsymbol{U}) = \operatorname{sign}\Big(\big(\textstyle\sum_{i=1}^{N_\Sigma} \boldsymbol{a}_{ij}^\top \boldsymbol{U}\boldsymbol{U}^\top \boldsymbol{b}_{ij} \big)_{j=1}^{m} \Big) \in \{-1,1\}^m,$$

where, for each $1 \leq j \leq m$ and $1 \leq i \leq N_\Sigma$, the vectors $\boldsymbol{a}_{ij}, \boldsymbol{b}_{ij} \sim_{\text{i.i.d.}} \mathcal{N}(\boldsymbol{0}, \mathbf{I}_n)$ are all independent. Compared to $\psi^{\pm}$, each component of $\psi^\Sigma(\boldsymbol{P})$ is thus obtained by summing N_Σ independent rank-one projections before applying the sign function.

As before, this binary embedding induces the empirical kernel

$$\widehat{\kappa}^\Sigma(\boldsymbol{U}, \boldsymbol{V}) := \tfrac{1}{m} \langle \psi^\Sigma(\boldsymbol{U}), \psi^\Sigma(\boldsymbol{V}) \rangle,$$

as well as the expected kernel

$$\kappa^\Sigma(\boldsymbol{U}, \boldsymbol{V}) := \mathbb{E}\Big[\tfrac{1}{m} \langle \psi^\Sigma(\boldsymbol{U}), \psi^\Sigma(\boldsymbol{V}) \rangle \Big].$$

Since the components of ψ^Σ remain binary, this construction keeps the compactness and storage advantages of one-bit embeddings, up to the additional factor N_Σ in the number of rank-one projections required to compute each feature.

Proposition 4.7 (Validity of κ^Σ)**.** *Let $N_\Sigma \geq 1$. For bases $\boldsymbol{U}, \boldsymbol{V} \in \mathbb{V}(k,n)$ whose subspaces form the principal angles $\theta_1, \dots, \theta_k$, the kernel κ^Σ is positive-definite, symmetric and only depends on these principal angles,* i.e.,

$$\kappa^\Sigma(\boldsymbol{U}, \boldsymbol{V}) = f(\theta_1, \dots, \theta_k)$$

for some function f.

Proof. By design, κ^Σ is symmetric. Regarding positive-definiteness, let $\boldsymbol{U}_1, \dots, \boldsymbol{U}_N \in \mathbb{V}(k,n)$ and let $\boldsymbol{c} = (c_1, \dots, c_N)^\top \in \mathbb{R}^N$. We denote by $\boldsymbol{K} \in \mathbb{R}^{N \times N}$ the Gram matrix associated with κ^Σ, defined component by component as $K_{rs} := \kappa^\Sigma(\boldsymbol{U}_r, \boldsymbol{U}_s)$. By definition of κ^Σ, we have

$$K_{rs} = \mathbb{E}\Big[\tfrac{1}{m}\langle \psi^\Sigma(\boldsymbol{U}_r), \psi^\Sigma(\boldsymbol{U}_s)\rangle\Big].$$

Therefore,

$$\begin{aligned}
\boldsymbol{c}^\top \boldsymbol{K} \boldsymbol{c} &= \textstyle\sum_{r,s=1}^N c_r c_s K_{rs} \\
&= \mathbb{E}\Big[\tfrac{1}{m} \textstyle\sum_{r,s=1}^N c_r c_s \langle \psi^\Sigma(\boldsymbol{U}_r), \psi^\Sigma(\boldsymbol{U}_s)\rangle\Big] \\
&= \langle \textstyle\sum_{r=1}^N c_r \psi^\Sigma(\boldsymbol{U}_r), \sum_{s=1}^N c_s \psi^\Sigma(\boldsymbol{U}_s)\rangle \\
&= \Big\| \textstyle\sum_{r=1}^N c_r \psi^\Sigma(\boldsymbol{U}_r) \Big\|_2^2,
\end{aligned}$$

by the bilinearity of the Euclidean scalar product. Hence,

$$\boldsymbol{c}^\top \boldsymbol{K} \boldsymbol{c} = \mathbb{E}\Big[\tfrac{1}{m}\Big\| \textstyle\sum_{r=1}^N c_r \psi^\Sigma(\boldsymbol{U}_r) \Big\|_2^2\Big] \geq 0.$$

This proves that κ^Σ is positive-definite. In addition, since $\psi^\Sigma(\boldsymbol{U})$ only depends on $\boldsymbol{U}$ via $\phi^{\mathrm{p}}(\boldsymbol{U}) = \boldsymbol{U}\boldsymbol{U}^\top$, the kernel κ^Σ is independent of the choice of orthonormal basis representing the subspace.

It remains to show that κ^Σ only depends on the principal angles. Let $\boldsymbol{P} = \phi^{\mathrm{p}}(\boldsymbol{U})$ and $\boldsymbol{Q} = \phi^{\mathrm{p}}(\boldsymbol{V})$. Since the components of ψ^Σ are identically distributed, we can focus on a single coordinate of ψ^Σ to find κ^Σ in expectation. By rotational invariance of the Gaussian distribution, for any orthogonal matrix $\boldsymbol{R} \in \mathsf{SO}(n)$, the vectors $\boldsymbol{R}^\top \boldsymbol{a}_\ell$ and $\boldsymbol{R}^\top \boldsymbol{b}_\ell$ have the same joint distribution as $\boldsymbol{a}_\ell$ and $\boldsymbol{b}_\ell$. Hence,

$$\begin{aligned}
\kappa^\Sigma(\boldsymbol{U}, \boldsymbol{V}) &= \mathbb{E}\Big[\operatorname{sign}\Big(\textstyle\sum_{\ell=1}^{N_\Sigma} \boldsymbol{a}_\ell^\top \boldsymbol{P} \boldsymbol{b}_\ell\Big) \operatorname{sign}\Big(\textstyle\sum_{\ell=1}^{N_\Sigma} \boldsymbol{a}_\ell^\top \boldsymbol{Q} \boldsymbol{b}_\ell\Big)\Big] \\
&= \mathbb{E}\Big[\operatorname{sign}\Big(\textstyle\sum_{\ell=1}^{N_\Sigma} (\boldsymbol{R}^\top \boldsymbol{a}_\ell)^\top \boldsymbol{P} (\boldsymbol{R}^\top \boldsymbol{b}_\ell)\Big) \operatorname{sign}\Big(\textstyle\sum_{\ell=1}^{N_\Sigma} (\boldsymbol{R}^\top \boldsymbol{a}_\ell)^\top \boldsymbol{Q} (\boldsymbol{R}^\top \boldsymbol{b}_\ell)\Big)\Big] \\
&= \mathbb{E}\Big[\operatorname{sign}\Big(\textstyle\sum_{\ell=1}^{N_\Sigma} \boldsymbol{a}_\ell^\top \boldsymbol{R}\boldsymbol{P}\boldsymbol{R}^\top \boldsymbol{b}_\ell\Big) \operatorname{sign}\Big(\textstyle\sum_{\ell=1}^{N_\Sigma} \boldsymbol{a}_\ell^\top \boldsymbol{R}\boldsymbol{Q}\boldsymbol{R}^\top \boldsymbol{b}_\ell\Big)\Big] \\
&= \kappa^\Sigma(\boldsymbol{R}\boldsymbol{U}, \boldsymbol{R}\boldsymbol{V}).
\end{aligned}$$

The kernel κ^{Σ} is thus rotationally invariant. By Thm. 2.1, it only depends on the principal angles between the two subspaces. □

We now introduce the closed-form kernel that is approximated by the aggregated construction.

Definition 4.4.1. *For $\boldsymbol{U}, \boldsymbol{V} \in \mathbb{V}(k, n)$ with principal angles $\theta_1, \dots, \theta_k$, we define*

$$\kappa_{\infty}^{\Sigma}(\boldsymbol{U}, \boldsymbol{V}) = \tfrac{2}{\pi} \arcsin(\tfrac{1}{k} \textstyle\sum_{i=1}^{k} \cos^2(\theta_i)).$$

The subscript ∞ indicates that this kernel appears as the limiting object when the aggregation parameter N_{Σ} tends to infinity.

Proposition 4.8 (Validity of κ_{∞}^{Σ})**.** *The function κ_{∞}^{Σ} is a positive semi-definite kernel on $\mathcal{G}(k, n)$. Moreover, it only depends on the principal angles between the two subspaces.*

Proof. The dependence on the principal angles follows directly from the definition. We prove positive semi-definiteness by realising κ_{∞}^{Σ} as the expectation of a binary random feature kernel.

Let $\boldsymbol{G} \in \mathbb{R}^{n \times n}$ have independent standard Gaussian entries. For $\boldsymbol{U} \in \mathbb{V}(k, n)$, define

$$z_{\boldsymbol{U}} = \tfrac{1}{\sqrt{k}} \operatorname{tr}(\boldsymbol{G}^{\top} \boldsymbol{U} \boldsymbol{U}^{\top}).$$

Since $z_{\boldsymbol{U}}$ is a linear combination of independent standard Gaussian variables, it is Gaussian. Its variance is

$$\mathbb{E}[z_{\boldsymbol{U}}^2] = \tfrac{1}{k} \|\boldsymbol{U}\boldsymbol{U}^{\top}\|_F^2 = 1,$$

because $\boldsymbol{U}\boldsymbol{U}^{\top}$ is an orthogonal projector of rank k. Thus $z_{\boldsymbol{U}}$ is a centred standard Gaussian random variable.

For two subspaces represented by $\boldsymbol{U}$ and $\boldsymbol{V}$, we can find the covariance by replacing the traces by Frobenius inner products,

$$\mathbb{E}[z_{\boldsymbol{U}} z_{\boldsymbol{V}}] = \tfrac{1}{k} \langle \boldsymbol{U}\boldsymbol{U}^{\top}, \boldsymbol{V}\boldsymbol{V}^{\top} \rangle_F = \tfrac{1}{k} \operatorname{tr}(\boldsymbol{U}\boldsymbol{U}^{\top}\boldsymbol{V}\boldsymbol{V}^{\top}).$$

The last trace is the projection kernel on the Grassmannian, thus we can call the principal angles back in to find

$$\operatorname{tr}(\boldsymbol{U}\boldsymbol{U}^{\top}\boldsymbol{V}\boldsymbol{V}^{\top}) = \textstyle\sum_{i=1}^{k} \cos^2(\theta_i).$$

Therefore the correlation between $z_{\boldsymbol{U}}$ and $z_{\boldsymbol{V}}$ is

$$\rho = \mathbb{E}[z_{\boldsymbol{U}} z_{\boldsymbol{V}}] = \tfrac{1}{k} \textstyle\sum_{i=1}^{k} \cos^2(\theta_i).$$

By Grothendieck's identity [148, Lem. 3.6.5] or the result from [130, Sec. 6.4],

$$\mathbb{E}[\text{sign}(z_{\boldsymbol{U}})\text{sign}(z_{\boldsymbol{V}})] = \tfrac{2}{\pi}\arcsin(\rho) = \kappa_\infty^\Sigma(\boldsymbol{U},\boldsymbol{V}).$$

Now let $\boldsymbol{U}_1,\dots,\boldsymbol{U}_N \in \mathbb{V}(k,n)$ and $c_1,\dots,c_N \in \mathbb{R}$. Then

$$\textstyle\sum_{i,j=1}^N c_i c_j \kappa_\infty^\Sigma(\boldsymbol{U}_i,\boldsymbol{U}_j) = \mathbb{E}[(\sum_{i=1}^N c_i \text{sign}(z_{\boldsymbol{U}_i}))^2] \geq 0.$$

Thus κ_∞^Σ is positive semi-definite. □

Prop. 4.8 shows that κ_∞^Σ is a valid Grassmannian kernel with an explicit formula. We now show that this ideal kernel can be arbitrarily close to κ^Σ at the cost of N_Σ being larger.

Proposition 4.9. *Let $\boldsymbol{U},\boldsymbol{V} \in \mathbb{R}^{n\times n}$ be two orthonormal bases with principal angles $\theta_1,\dots,\theta_k$, and let ψ^Σ be as defined earlier. Then, for some absolute constant $C > 0$, we have*

$$\left|\tfrac{1}{m}\mathbb{E}\left[\langle\psi^\Sigma(\boldsymbol{U}),\psi^\Sigma(\boldsymbol{V})\rangle\right] - \kappa_\infty^\Sigma(\boldsymbol{U},\boldsymbol{V})\right| \leq \tfrac{C}{\sqrt{N_\Sigma}}.$$

Proof. See Sec. 4.4.2. □

This proposition thus shows that an explicit formula for a kernel is achievable at the cost of modifying slightly the embedding procedure. Practically this will be close to $\kappa^\pm$ but it provides a usable formula in contexts where an explicit formula is necessary as well as an interpretable formula for a binary embedding in terms of principal angles. The proof for this proposition is presented in Sec. 4.4.2 and experiments validate the dependence of this approximation on N_Σ in Sec. 4.7.2.

Since we keep this particular approximation strictly as a way to illustrate the behaviour of κ^Σ as $N_\Sigma \to \infty$, we do not derive concentration bounds for $\widehat{\kappa}^\Sigma$. For fixed subspaces, the empirical average in ψ^Σ remains an average of independent bounded functions of ψ^{rop}, so concentration around its expectation is still expected. However, a full uniform analysis over $\mathcal{G}(k,n)$ is not needed for the role played by this construction here.

4.4.2 Proof of Prop. 4.9

We first present the following result which will be necessary to obtain a bound on the error between the expectation of κ^Σ and its explicit formula.

Lemma 4.4. *Let*

$$z := \begin{pmatrix} a^\top P b \\ a^\top Q b \end{pmatrix},$$

where $P, Q \in \mathbb{R}^{n\times n}$ are two orthogonal projection matrices of rank $k \leq \frac{n}{2}$, and $a, b \sim \mathcal{N}(0, \mathbf{I}_n)$. Let $\Sigma = \mathrm{Cov}(z)$, and assume that Σ is invertible. Then

$$\mathbb{E}\|\Sigma^{-1/2} z\|_2^3 \leq C,$$

for some absolute constant $C > 0$.

Proof. Let us first compute the covariance matrix of z.

$$z = \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} = \begin{pmatrix} a^\top P b \\ a^\top Q b \end{pmatrix}.$$

Since for two symmetric matrices $A, B \in \mathbb{R}^{n\times n}$ and two independent Gaussian vectors $a, b \sim \mathcal{N}(0, \mathbf{I}_n)$, we have

$$\begin{aligned} \mathbb{E}[(a^\top A b)(a^\top B b)] &= \mathbb{E}_a \left[\mathbb{E}_b [a^\top A b b^\top B a] \right] \\ &= \mathbb{E}_a \left[a^\top A \mathbf{I}_n B a \right] \\ &= \mathrm{tr}(AB \mathbb{E}_a [a a^\top]) \\ &= \mathrm{tr}(AB), \end{aligned} \tag{4.13}$$

and because P and Q are symmetric, we have

$$\Sigma = \begin{pmatrix} \mathbb{E}[z_1^2] & \mathbb{E}[z_1 z_2] \\ \mathbb{E}[z_1 z_2] & \mathbb{E}[z_2^2] \end{pmatrix} = \begin{pmatrix} k & \mathrm{tr}(PQ) \\ \mathrm{tr}(PQ) & k \end{pmatrix}.$$

This matrix is symmetric and can be diagonalised as $\Sigma = X \Lambda X^{-1}$, where

$$X = \tfrac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \qquad \Lambda = \begin{pmatrix} \lambda_+ & 0 \\ 0 & \lambda_- \end{pmatrix},$$

with the corresponding eigenvalues

$$\lambda_+ = k + \mathrm{tr}(PQ), \qquad \lambda_- = k - \mathrm{tr}(PQ).$$

Since the object of interest in this proof is $\mathbb{E}\|\Sigma^{-1/2} z\|_2^3$ we will now focus on $\|\Sigma^{-1/2} z\|_2$. Using the diagonalisation of Σ, we can express the squared norm as

$$\|\Sigma^{-1/2} z\|_2^2 = z^\top (\Sigma^{-1/2})^\top \Sigma^{-1/2} z = z^\top \Sigma^{-1} z.$$

Substituting $\boldsymbol{\Sigma}^{-1} = \boldsymbol{X}\boldsymbol{\Lambda}^{-1}\boldsymbol{X}^\top$ into this expression gives:

$$\|\boldsymbol{\Sigma}^{-1/2}\boldsymbol{z}\|_2^2 = \boldsymbol{z}^\top(\boldsymbol{X}\boldsymbol{\Lambda}^{-1}\boldsymbol{X}^\top)\boldsymbol{z} = (\boldsymbol{X}^\top\boldsymbol{z})^\top\boldsymbol{\Lambda}^{-1}(\boldsymbol{X}^\top\boldsymbol{z}).$$

We now define

$$\begin{pmatrix} u \\ v \end{pmatrix} := \boldsymbol{X}^\top\boldsymbol{z} = \tfrac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}\begin{pmatrix} z_1 \\ z_2 \end{pmatrix} = \begin{pmatrix} \frac{z_1+z_2}{\sqrt{2}} \\ \frac{z_1-z_2}{\sqrt{2}} \end{pmatrix},$$

and therefore,

$$\|\boldsymbol{\Sigma}^{-1/2}\boldsymbol{z}\|_2^2 = \begin{pmatrix} u & v \end{pmatrix}\begin{pmatrix} \lambda_+^{-1} & 0 \\ 0 & \lambda_-^{-1} \end{pmatrix}\begin{pmatrix} u \\ v \end{pmatrix} = \tfrac{u^2}{\lambda_+} + \tfrac{v^2}{\lambda_-}. \tag{4.14}$$

Using this result we can write $\|\boldsymbol{\Sigma}^{-1/2}\boldsymbol{z}\|_2 = \sqrt{\frac{u^2}{\lambda_+} + \frac{v^2}{\lambda_-}}$, and since $\sqrt{x+y} \leq \sqrt{x} + \sqrt{y}$ for $x, y \geq 0$, we have

$$\|\boldsymbol{\Sigma}^{-1/2}\boldsymbol{z}\|_2 \leq \sqrt{\tfrac{u^2}{\lambda_+}} + \sqrt{\tfrac{v^2}{\lambda_-}} = \tfrac{|u|}{\lambda_+^{1/2}} + \tfrac{|v|}{\lambda_-^{1/2}}.$$

This allows us to bound $\|\boldsymbol{\Sigma}^{-1/2}\boldsymbol{z}\|_2^3$. Since $f(x) = x^3$ is convex for $x \geq 0$ we can apply Jensen's inequality which gives

$$\left(\tfrac{a+b}{2}\right)^3 \leq \tfrac{a^3+b^3}{2}.$$

This leads us to

$$(a+b)^3 \leq 4(a^3+b^3),$$

and therefore we obtain:

$$\|\boldsymbol{\Sigma}^{-1/2}\boldsymbol{z}\|_2^3 \leq 4\tfrac{|u|^3}{\lambda_+^{3/2}} + 4\tfrac{|v|^3}{\lambda_-^{3/2}}.$$

We can then substitute the result from Eq. 4.14 and take expectations (since the bound holds for every value of u and v) to get

$$\mathbb{E}\|\boldsymbol{\Sigma}^{-1/2}\boldsymbol{z}\|_2^3 \leq \mathbb{E}\Big(\tfrac{|u|}{\lambda_+^{1/2}} + \tfrac{|v|}{\lambda_-^{1/2}}\Big)^3 \leq 4\tfrac{\mathbb{E}[|u|^3]}{\lambda_+^{3/2}} + 4\tfrac{\mathbb{E}[|v|^3]}{\lambda_-^{3/2}}. \tag{4.15}$$

All that remains now is to find bounds on $\mathbb{E}[|u|^3]$ and $\mathbb{E}[|v|^3]$. Following the canonical representation of Grassmannian projectors given in [47], and assuming $k \leq n/2$, we can write,

$$\boldsymbol{U}^\top = (\boldsymbol{C}, \boldsymbol{S}, \boldsymbol{0}_{k\times n-2k}), \qquad \boldsymbol{V}^\top = (\boldsymbol{C}, -\boldsymbol{S}, \boldsymbol{0}_{k\times n-2k}),$$

where $\boldsymbol{P} = \boldsymbol{U}\boldsymbol{U}^\top$, $\boldsymbol{Q} = \boldsymbol{V}\boldsymbol{V}^\top$,

$$\boldsymbol{C} = \operatorname{diag}\big(\cos(\theta_1/2), \ldots, \cos(\theta_k/2)\big)$$

and

$$\boldsymbol{S} = \operatorname{diag}\big(\sin(\theta_1/2), \dots, \sin(\theta_k/2)\big).$$

With this representation, we have

$$\boldsymbol{P} + \boldsymbol{Q} = \begin{pmatrix} 2\boldsymbol{C}^2 & \mathbf{0}_{k\times k} & \mathbf{0}_{k\times n-2k} \\ \mathbf{0}_{k\times k} & 2\boldsymbol{S}^2 & \mathbf{0}_{k\times n-2k} \\ \mathbf{0}_{n-2k\times k} & \mathbf{0}_{n-2k\times k} & \mathbf{0}_{n-2k\times n-2k} \end{pmatrix},$$

and

$$\boldsymbol{P} - \boldsymbol{Q} = \begin{pmatrix} \mathbf{0}_{k\times k} & 2\boldsymbol{C}\boldsymbol{S} & \mathbf{0}_{k\times n-2k} \\ 2\boldsymbol{S}\boldsymbol{C} & \mathbf{0}_{k\times k} & \mathbf{0}_{k\times n-2k} \\ \mathbf{0}_{n-2k\times k} & \mathbf{0}_{n-2k\times k} & \mathbf{0}_{n-2k\times n-2k} \end{pmatrix}.$$

Using these block forms, we get

$$\begin{aligned} u &= \tfrac{1}{\sqrt{2}}\boldsymbol{a}^\top(\boldsymbol{P}+\boldsymbol{Q})\boldsymbol{b} \\ &= \sqrt{2}\textstyle\sum_{j=1}^{k}\big(\cos^2(\theta_j/2)a_jb_j + \sin^2(\theta_j/2)a_{k+j}b_{k+j}\big), \end{aligned}$$

and

$$\begin{aligned} v &= \tfrac{1}{\sqrt{2}}\boldsymbol{a}^\top(\boldsymbol{P}-\boldsymbol{Q})\boldsymbol{b} \\ &= \sqrt{2}\textstyle\sum_{j=1}^{k}\cos(\theta_j/2)\sin(\theta_j/2)\big(a_jb_{k+j} + a_{k+j}b_j\big) \\ &= \tfrac{1}{\sqrt{2}}\textstyle\sum_{j=1}^{k}\sin(\theta_j)\big(a_jb_{k+j} + a_{k+j}b_j\big). \end{aligned}$$

Let us first focus on u. By Cauchy-Schwarz applied to the random variables $|u|$ and u^2 we have

$$\mathbb{E}|u|^3 = \mathbb{E}[|u|u^2] \le \big(\mathbb{E}[u^2]\big)^{1/2}\big(\mathbb{E}[u^4]\big)^{1/2}. \tag{4.16}$$

Thus it is enough to find bounds on $\mathbb{E}[u^2]$ and $\mathbb{E}[u^4]$. Let us define the independent random variables

$$\xi_j := a_jb_j, \qquad \xi_{k+j} := a_{k+j}b_{k+j}, \qquad 1 \le j \le k,$$

and the coefficients

$$\alpha_j := \sqrt{2}\cos^2(\theta_j/2), \qquad \alpha_{k+j} := \sqrt{2}\sin^2(\theta_j/2).$$

Then

$$u = \textstyle\sum_{\ell=1}^{2k}\alpha_\ell\xi_\ell.$$

The variables ξ_ℓ are independent, centred, and satisfy

$$\mathbb{E}[\xi_\ell^2] = 1, \qquad \mathbb{E}[\xi_\ell^4] = \mathbb{E}[a^4]\mathbb{E}[b^4] = 9,$$

since a_j and b_j are independent and since $\mathbb{E}[g^4] = 3$ for $g \sim \mathcal{N}(0,1)$. Since $u = \sum_{\ell=1}^{2k} \alpha_\ell \xi_\ell$, the multinomial expansion of u^4 contains terms of the form

$$\alpha_{\ell_1}\alpha_{\ell_2}\alpha_{\ell_3}\alpha_{\ell_4}\xi_{\ell_1}\xi_{\ell_2}\xi_{\ell_3}\xi_{\ell_4}.$$

Because the random variables ξ_ℓ are independent and centred, every term where at least one index has an odd power has zero expectation. Thus, the only surviving terms are those where all four indices are equal, and those where two distinct indices each appear twice. Therefore,

$$\begin{aligned}\mathbb{E}[u^4] &= \textstyle\sum_{\ell=1}^{2k} \alpha_\ell^4 \mathbb{E}[\xi_\ell^4] + 6\sum_{1\le\ell<r\le 2k} \alpha_\ell^2\alpha_r^2 \mathbb{E}[\xi_\ell^2]\mathbb{E}[\xi_r^2] \\ &= 9\textstyle\sum_{\ell=1}^{2k} \alpha_\ell^4 + 6\sum_{1\le\ell<r\le 2k} \alpha_\ell^2\alpha_r^2.\end{aligned}$$

Now,

$$\Big(\textstyle\sum_{\ell=1}^{2k} \alpha_\ell^2\Big)^2 = \sum_{\ell=1}^{2k} \alpha_\ell^4 + 2\sum_{1\le\ell<r\le 2k} \alpha_\ell^2\alpha_r^2.$$

Hence

$$9\Big(\textstyle\sum_{\ell=1}^{2k} \alpha_\ell^2\Big)^2 = 9\sum_{\ell=1}^{2k} \alpha_\ell^4 + 18\sum_{1\le\ell<r\le 2k} \alpha_\ell^2\alpha_r^2,$$

which gives

$$9\textstyle\sum_{\ell=1}^{2k} \alpha_\ell^4 + 6\sum_{1\le\ell<r\le 2k} \alpha_\ell^2\alpha_r^2 \le 9\Big(\sum_{\ell=1}^{2k} \alpha_\ell^2\Big)^2.$$

Therefore,

$$\mathbb{E}[u^4] \le 9\Big(\textstyle\sum_{\ell=1}^{2k} \alpha_\ell^2\Big)^2.$$

Moreover,

$$\mathbb{E}[u^2] = \mathbb{E}\Big[\Big(\textstyle\sum_{\ell=1}^{2k} \alpha_\ell\xi_\ell\Big)^2\Big] = \mathbb{E}\Big[\textstyle\sum_{\ell=1}^{2k} \alpha_\ell^2\xi_\ell^2 + 2\sum_{1\le\ell<r\le 2k} \alpha_\ell\alpha_r\xi_\ell\xi_r\Big].$$

Since the ξ_ℓ are centred and independent, the cross terms have zero expectation, and since $\mathbb{E}[\xi_\ell^2] = 1$ we get

$$\mathbb{E}[u^2] = \textstyle\sum_{\ell=1}^{2k} \alpha_\ell^2.$$

Using the previous bounds along with Eq. 4.16, we obtain

$$\mathbb{E}|u|^3 \le 3\Big(\textstyle\sum_{\ell=1}^{2k} \alpha_\ell^2\Big)^{3/2}.$$

Finally, writing $c := \cos(\theta_j/2)$ and $s := \sin(\theta_j/2)$)

$$\begin{aligned}\textstyle\sum_{\ell=1}^{2k} \alpha_\ell^2 &= 2\sum_{j=1}^{k}(c^4+s^4) = 2\sum_{j=1}^{k}\left[(c^2+s^2)^2 - 2c^2s^2\right] \\ &= 2\textstyle\sum_{j=1}^{k}\left[1 - \frac{1}{2}\sin^2(\theta_j)\right] = \sum_{j=1}^{k}\left(1+\cos^2(\theta_j)\right) \\ &= k + \operatorname{tr}(\boldsymbol{PQ}) = \lambda_+.\end{aligned}$$

Hence

$$\mathbb{E}|u|^3 \leq 3\lambda_+^{3/2}.$$

Let us now focus on v. As for u, our goal is to bound $\mathbb{E}|v|^3$. By Cauchy-Schwarz applied to the random variables $|v|$ and v^2, we have

$$\mathbb{E}|v|^3 = \mathbb{E}[|v|v^2] \leq \big(\mathbb{E}[v^2]\big)^{1/2}\big(\mathbb{E}[v^4]\big)^{1/2}. \tag{4.17}$$

It is therefore enough to control $\mathbb{E}[v^2]$ and $\mathbb{E}[v^4]$. Recall that

$$v = \tfrac{1}{\sqrt{2}} \textstyle\sum_{j=1}^{k} \sin(\theta_j)\big(a_j b_{k+j} + a_{k+j} b_j\big).$$

Define the independent random variables

$$\eta_j := a_j b_{k+j} + a_{k+j} b_j, \qquad 1 \leq j \leq k,$$

and the coefficients

$$\beta_j := \tfrac{1}{\sqrt{2}} \sin(\theta_j).$$

Then

$$v = \textstyle\sum_{j=1}^{k} \beta_j \eta_j.$$

The variables η_j are independent and centred. Moreover,

$$\mathbb{E}[\eta_j^2] = \mathbb{E}[(a_j b_{k+j})^2] + \mathbb{E}[(a_{k+j} b_j)^2] = 2,$$

since the cross term has zero expectation. Similarly, since $a_j b_{k+j}$ and $a_{k+j} b_j$ are independent and centred,

$$\begin{aligned}\mathbb{E}[\eta_j^4] &= \mathbb{E}[(a_j b_{k+j})^4] + \mathbb{E}[(a_{k+j} b_j)^4] + 6\mathbb{E}[(a_j b_{k+j})^2]\mathbb{E}[(a_{k+j} b_j)^2] \\ &= 9 + 9 + 6 = 24.\end{aligned}$$

We first compute the fourth moment. Similarly to what we did for u, since $v = \sum_{j=1}^{k} \beta_j \eta_j$, the multinomial expansion of v^4 contains terms of the form

$$\beta_{j_1}\beta_{j_2}\beta_{j_3}\beta_{j_4}\eta_{j_1}\eta_{j_2}\eta_{j_3}\eta_{j_4}.$$

Because the random variables η_j are independent and centred, every term where at least one index appears an odd number of times has zero expectation. Thus, the only surviving terms are those where all four indices are equal, and those where two distinct indices each appear twice. Therefore,

$$\begin{aligned}\mathbb{E}[v^4] &= \textstyle\sum_{j=1}^{k} \beta_j^4 \mathbb{E}[\eta_j^4] + 6 \sum_{1 \leq j < \ell \leq k} \beta_j^2 \beta_\ell^2 \mathbb{E}[\eta_j^2]\mathbb{E}[\eta_\ell^2] \\ &= 24 \textstyle\sum_{j=1}^{k} \beta_j^4 + 24 \sum_{1 \leq j < \ell \leq k} \beta_j^2 \beta_\ell^2.\end{aligned}$$

Now,

$$\Big(\sum_{j=1}^k \beta_j^2\Big)^2 = \sum_{j=1}^k \beta_j^4 + 2\sum_{1\le j<\ell\le k} \beta_j^2\beta_\ell^2.$$

Hence

$$24\Big(\sum_{j=1}^k \beta_j^2\Big)^2 = 24\sum_{j=1}^k \beta_j^4 + 48\sum_{1\le j<\ell\le k} \beta_j^2\beta_\ell^2,$$

which gives

$$24\sum_{j=1}^k \beta_j^4 + 24\sum_{1\le j<\ell\le k} \beta_j^2\beta_\ell^2 \le 24\Big(\sum_{j=1}^k \beta_j^2\Big)^2.$$

Therefore,

$$\mathbb{E}[v^4] \le 24\Big(\sum_{j=1}^k \beta_j^2\Big)^2.$$

We also have

$$\mathbb{E}[v^2] = \mathbb{E}\Big[\Big(\sum_{j=1}^k \beta_j\eta_j\Big)^2\Big] = \mathbb{E}\Big[\sum_{j=1}^k \beta_j^2\eta_j^2 + 2\sum_{1\le j<\ell\le k} \beta_j\beta_\ell\eta_j\eta_\ell\Big].$$

Again since the η_j are centred and independent, the cross terms have zero expectation, and since $\mathbb{E}[\eta_j^2] = 2$ we get

$$\mathbb{E}[v^2] = 2\sum_{j=1}^k \beta_j^2.$$

Using the previous bounds along with Eq. 4.17, we obtain

$$\mathbb{E}|v|^3 \le \Big(2\sum_{j=1}^k \beta_j^2\Big)^{1/2}\Big(24\Big(\sum_{j=1}^k \beta_j^2\Big)^2\Big)^{1/2} = \sqrt{6}\Big(2\sum_{j=1}^k \beta_j^2\Big)^{3/2}.$$

Finally,

$$\begin{aligned} 2\sum_{j=1}^k \beta_j^2 &= \sum_{j=1}^k \sin^2(\theta_j) \\ &= k - \operatorname{tr}(\boldsymbol{P}\boldsymbol{Q}) \\ &= \lambda_-. \end{aligned}$$

Hence

$$\mathbb{E}|v|^3 \le \sqrt{6}\lambda_-^{3/2}.$$

We can now conclude. Using the bounds on $\mathbb{E}[|u|^3]$ and $\mathbb{E}[|v|^3]$ along with Eq. 4.15, we get

$$\mathbb{E}\|\boldsymbol{\Sigma}^{-1/2}\boldsymbol{z}\|_2^3 \le \mathbb{E}\Big[4\tfrac{|u|^3}{\lambda_+^{3/2}} + 4\tfrac{|v|^3}{\lambda_-^{3/2}}\Big] \le 12 + 4\sqrt{6} \le C,$$

for some $C > 0$, which proves the result.

□

Lem. 4.4 provides a bound on the third moment of z which will allow us to apply a multivariate Berry-Esseen theorem to the aggregated variables. More precisely, it shows that the normalised third moment remains bounded by an absolute constant, which is independent of the subspace dimension k. This will allow us to compare the distribution of the aggregated pair of rank-one projections with the Gaussian distribution, thereby allowing us to compare the expected signed kernel with the corresponding Gaussian sign kernel, for which the classical arcsine formula gives an explicit expression. Let us now prove Prop. 4.9.

Proof of Prop. 4.9

Proof. Throughout this proof we will use $\boldsymbol{P} := \phi^{\mathrm{P}}(\boldsymbol{U})$ and $\boldsymbol{Q} := \phi^{\mathrm{P}}(\boldsymbol{V})$. Since the m coordinates of $\psi^{\Sigma}(\boldsymbol{U})$ and $\psi^{\Sigma}(\boldsymbol{V})$ are independent and identically distributed with respect to the index j, by linearity of expectation it is enough to study one coordinate. We can thus fix $j = 1$ in ψ^{Σ} and write $\boldsymbol{a}_{i1}$ as $\boldsymbol{a}_i$ as well as $\boldsymbol{b}_{i1}$ as $\boldsymbol{b}_i$ for clarity. We define

$$X_{N_\Sigma} := \sum_{i=1}^{N_\Sigma} \boldsymbol{a}_i^\top \boldsymbol{U}\boldsymbol{U}^\top \boldsymbol{b}_i, \qquad Y_{N_\Sigma} := \sum_{i=1}^{N_\Sigma} \boldsymbol{a}_i^\top \boldsymbol{V}\boldsymbol{V}^\top \boldsymbol{b}_i.$$

Then

$$\tfrac{1}{m}\mathbb{E}\Big[\langle \psi^{\Sigma}(\boldsymbol{U}), \psi^{\Sigma}(\boldsymbol{V})\rangle\Big] = \mathbb{E}\big[\operatorname{sign}(X_{N_\Sigma})\operatorname{sign}(Y_{N_\Sigma})\big].$$

Since the sign is invariant under multiplication by a positive scalar, this is also equal to

$$\mathbb{E}\Big[\operatorname{sign}\Big(\tfrac{X_{N_\Sigma}}{\sqrt{N_\Sigma}}\Big)\operatorname{sign}\Big(\tfrac{Y_{N_\Sigma}}{\sqrt{N_\Sigma}}\Big)\Big].$$

We now define

$$\boldsymbol{z}_i := \begin{pmatrix} \boldsymbol{a}_i^\top \boldsymbol{U}\boldsymbol{U}^\top \boldsymbol{b}_i \\ \boldsymbol{a}_i^\top \boldsymbol{V}\boldsymbol{V}^\top \boldsymbol{b}_i \end{pmatrix}, \qquad \boldsymbol{s} := \tfrac{1}{\sqrt{N_\Sigma}} \sum_{i=1}^{N_\Sigma} \boldsymbol{z}_i = \begin{pmatrix} X_{N_\Sigma}/\sqrt{N_\Sigma} \\ Y_{N_\Sigma}/\sqrt{N_\Sigma} \end{pmatrix}.$$

The vectors $\boldsymbol{z}_i$ are independent and identically distributed, centred, and belong to $\mathbb{R}^2$.

Let $\boldsymbol{\Sigma} := \operatorname{Cov}(\boldsymbol{z}_i)$, which will be the same for any value of i since all random variables are i.i.d. . If $\boldsymbol{U}\boldsymbol{U}^\top = \boldsymbol{V}\boldsymbol{V}^\top \leftrightarrow \boldsymbol{P} = \boldsymbol{Q}$, then

$$\tfrac{1}{m}\mathbb{E}\Big[\langle \psi^{\Sigma}(\boldsymbol{U}), \psi^{\Sigma}(\boldsymbol{V})\rangle\Big] = 1, \qquad \tfrac{2}{\pi}\arcsin\Big(\tfrac{1}{k}\sum_{i=1}^{k}\cos^2(\theta_i)\Big) = 1,$$

so the result is immediate. We therefore assume in the rest of the proof that $\boldsymbol{P} \neq \boldsymbol{Q}$, so that $\boldsymbol{\Sigma}$ is invertible.

Let us compute $\boldsymbol{\Sigma}$. Since $\boldsymbol{P}$ and $\boldsymbol{Q}$ are orthogonal projectors of rank k, we know that

$$\operatorname{tr}(\boldsymbol{P}\boldsymbol{Q}) = \sum_{l=1}^{k}\cos^2(\theta_l).$$

so we can compute

$$\begin{aligned}\boldsymbol{\Sigma} &= \begin{pmatrix} \mathbb{E}[(\boldsymbol{a}^\top \boldsymbol{P}\boldsymbol{b})^2] & \mathbb{E}[(\boldsymbol{a}^\top \boldsymbol{P}\boldsymbol{b})(\boldsymbol{a}^\top \boldsymbol{Q}\boldsymbol{b})] \\ \mathbb{E}[(\boldsymbol{a}^\top \boldsymbol{Q}\boldsymbol{b})(\boldsymbol{a}^\top \boldsymbol{P}\boldsymbol{b})] & \mathbb{E}[(\boldsymbol{a}^\top \boldsymbol{Q}\boldsymbol{b})^2] \end{pmatrix} \\ &= \begin{pmatrix} k & \operatorname{tr}(\boldsymbol{P}\boldsymbol{Q}) \\ \operatorname{tr}(\boldsymbol{P}\boldsymbol{Q}) & k \end{pmatrix},\end{aligned}$$

where $\boldsymbol{a}, \boldsymbol{b}$ are independent copies of the vectors $\boldsymbol{a}_i, \boldsymbol{b}_i$ and where we reused Eq. 4.13 from the proof of Lem. 4.4 to write $\mathbb{E}[(\boldsymbol{a}^\top \boldsymbol{P}\boldsymbol{b})(\boldsymbol{a}^\top \boldsymbol{Q}\boldsymbol{b})] = \operatorname{tr}(\boldsymbol{P}\boldsymbol{Q})$. Hence, if $\boldsymbol{G} = (G_1, G_2) \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma})$, the correlation coefficient of G_1 and G_2 is

$$\rho := \tfrac{\operatorname{tr}(\boldsymbol{P}\boldsymbol{Q})}{\sqrt{k}\sqrt{k}} = \tfrac{1}{k}\sum_{l=1}^{k} \cos^2(\theta_l).$$

We now compare the distribution of $\boldsymbol{s}$ to the Gaussian distribution $\mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma})$. By the multivariate Berry-Esseen theorem for convex sets in dimension 2 (see [20, Theorem 1.1] or [128, Theorem 1.1]), applied to $\frac{1}{\sqrt{N_\Sigma}}\sum_i^{N_\Sigma} \boldsymbol{z}_i$, for every convex set $A \subset \mathbb{R}^2$,

$$\left|\mathbb{P}(\boldsymbol{s} \in A) - \mathbb{P}(\boldsymbol{G} \in A)\right| \leq \tfrac{C}{\sqrt{N_\Sigma}}\beta_\Sigma,$$

with

$$\beta_\Sigma := \mathbb{E}\|\boldsymbol{\Sigma}^{-1/2}\boldsymbol{z}_i\|^3,$$

for some absolute constant $C > 0$.

In order to show that we may apply the multivariate Berry-Esseen theorem, we must observe that the value of $\mathbb{E}[\operatorname{sign}(X)\operatorname{sign}(Y)]$ depends on the probability of $\begin{pmatrix} X & Y \end{pmatrix}^\top$ being in one of the four orthants of $\mathbb{R}^2$. Let us focus on the first quadrant

$$A_+ := \{(x, y) \in \mathbb{R}^2 : x > 0,\ y > 0\},$$

which is convex. Therefore,

$$\left|\mathbb{P}(\boldsymbol{s} \in A_+) - \mathbb{P}(\boldsymbol{G} \in A_+)\right| \leq \tfrac{C}{\sqrt{N_\Sigma}}\beta_\Sigma. \tag{4.18}$$

Next, denoting the first element of $\boldsymbol{s}$ by $s_{N_\Sigma,1}$ and the second by $s_{N_\Sigma,2}$, and using the symmetry with respect to the origin, $\boldsymbol{s}$ satisfies

$$\mathbb{P}(s_{N_\Sigma,1} > 0, s_{N_\Sigma,2} > 0) = \mathbb{P}(s_{N_\Sigma,1} < 0, s_{N_\Sigma,2} < 0),$$

and similarly for $\boldsymbol{G}$. Therefore if we denote the four quadrants of $\mathbb{R}^2$ by $Q_1, \ldots, Q_4$, we have

$$\mathbb{E}[\operatorname{sign}(s_{N_\Sigma,1})\operatorname{sign}(s_{N_\Sigma,2})] = \mathbb{P}(Q_1) - \mathbb{P}(Q_2) + \mathbb{P}(Q_3) - \mathbb{P}(Q_4).$$

By symmetry, $\mathbb{P}(Q_1) = \mathbb{P}(Q_3) = \mathbb{P}(\boldsymbol{s} \in A_+)$ and $\mathbb{P}(Q_2) = \mathbb{P}(Q_4) = 1/2 - \mathbb{P}(A_+)$, which gives

$$\mathbb{E}[\operatorname{sign}(s_{N_\Sigma,1})\operatorname{sign}(s_{N_\Sigma,2})] = 4\mathbb{P}(\boldsymbol{s} \in A_+) - 1.$$

and

$$\mathbb{E}[\operatorname{sign}(G_1)\operatorname{sign}(G_2)] = 4\mathbb{P}(\boldsymbol{G} \in A_+) - 1.$$

Combining these identities with Eq. 4.18 gives

$$\left|\mathbb{E}[\operatorname{sign}(X_{N_\Sigma})\operatorname{sign}(Y_{N_\Sigma})] - \mathbb{E}[\operatorname{sign}(G_1)\operatorname{sign}(G_2)]\right| \leq \tfrac{C}{\sqrt{N_\Sigma}}\beta_\Sigma, \quad (4.19)$$

with an absolute constant $C > 0$.

Finally, since (G_1, G_2) is a centred bivariate Gaussian vector with correlation coefficient ρ, we can use Grothendieck's identity [148, Lem. 3.6.6] or the result from [130, Sec. 6.4]

$$\mathbb{E}[\operatorname{sign}(G_1)\operatorname{sign}(G_2)] = \tfrac{2}{\pi}\arcsin(\rho) = \tfrac{2}{\pi}\arcsin\left(\tfrac{1}{k}\textstyle\sum_{i=1}^{k}\cos^2(\theta_i)\right).$$

Injecting this identity into Eq. 4.19, we obtain

$$\left|\tfrac{1}{m}\mathbb{E}\left[\langle\psi^\Sigma(\boldsymbol{U}), \psi^\Sigma(\boldsymbol{V})\rangle\right] - \tfrac{2}{\pi}\arcsin\left(\tfrac{1}{k}\textstyle\sum_{i=1}^{k}\cos^2(\theta_i)\right)\right| \leq \tfrac{C}{\sqrt{N_\Sigma}}\beta_\Sigma.$$

Using the constant used for bounding β_Σ from Lem. 4.4 and absorbing this constant into C yields the desired result:

$$\left|\tfrac{1}{m}\mathbb{E}\left[\langle\psi^\Sigma(\boldsymbol{U}), \psi^\Sigma(\boldsymbol{V})\rangle\right] - \tfrac{2}{\pi}\arcsin\left(\tfrac{1}{k}\textstyle\sum_{i=1}^{k}\cos^2(\theta_i)\right)\right| \leq \tfrac{C}{\sqrt{N_\Sigma}},$$

for some $C > 0$. □

4.5 Periodic random features for Grassmannian kernels

We now turn to the second bounded non-linear function considered in this chapter, namely the complex exponential. As in the signed case we are going to compose the rank-one projection operator ψ^{rop} with a bounded function applied componentwise. However, the resulting embedding behaves quite differently here since it takes values in a complex space rather than in $\{\pm 1\}^m$.

Another important difference is that the kernel induced by this embedding admits a closed-form expression in terms of the principal angles between subspaces. This allows us to analyse its behaviour more explicitly, in particular through a *frequency parameter* ω, and to relate it to more classical Grassmannian kernels for different values of this parameter.

4.5.1 Definition

We apply to the ROP operator ψ^{rop} the bounded function $g(\lambda) = \exp(\mathrm{i}\omega\lambda)$, where $\omega \in \mathbb{R}$ is a frequency parameter. The resulting periodic random feature map $\psi^\circ = g \circ \psi^{\mathrm{rop}}$ is reminiscent of the random Fourier features of [125], where random projections are followed by a periodic non-linear function. Given m random vectors $\boldsymbol{a}_i, \boldsymbol{b}_i \sim_{\text{i.i.d.}} \mathcal{N}(\mathbf{0}, \mathbf{I}_n)$, for $1 \leq i \leq m$, we define

$$\psi^\circ(\boldsymbol{U}) := \Big(\exp(\mathrm{i}\omega\, \boldsymbol{a}_i^\top \phi^{\mathrm{p}}(\boldsymbol{U}) \boldsymbol{b}_i) \Big)_{i=1}^m, \quad \boldsymbol{U} \in \mathbb{V}(k,n). \tag{4.20}$$

We can then show that the empirical kernel $\widehat{\kappa}^\circ := \frac{1}{m}\langle \psi^\circ(\boldsymbol{U}), \psi^\circ(\boldsymbol{V}) \rangle$ between two k-dimensional subspace bases $\boldsymbol{U}, \boldsymbol{V} \in \mathbb{V}(k,n)$ is an unbiased estimator of the following kernel.

Proposition 4.10 (Validity of κ°)**.** *Given $\boldsymbol{U}, \boldsymbol{V} \in \mathbb{V}(k,n)$, with principal angles $\theta_1, \ldots, \theta_k$ between them, and ψ° defined as above, the kernel κ° is positive-definite, symmetric and only depends on the principal angles between the two subspaces $\boldsymbol{U}$ and $\boldsymbol{V}$. Moreover,*

$$\kappa^\circ(\boldsymbol{U}, \boldsymbol{V}) := \mathbb{E}\widehat{\kappa}^\circ(\boldsymbol{U}, \boldsymbol{V}) = \textstyle\prod_{j=1}^k (1 + \omega^2 \sin^2\theta_j)^{-1}. \tag{4.21}$$

Proof. The symmetry and positive-semidefiniteness of κ° are proved similarly to those of $\kappa^\pm$. Regarding the dependence of κ° on the subspace principal angles, we first note that, given $\boldsymbol{U}, \boldsymbol{V} \in \mathbb{V}(k,n)$ and the random vectors $\boldsymbol{a}, \boldsymbol{b} \sim_{\text{i.i.d.}} \mathcal{N}(\mathbf{0}, \mathbf{I}_n)$,

$$\kappa^\circ(\boldsymbol{U}, \boldsymbol{V}) = \mathbb{E}\widehat{\kappa}^\circ(\boldsymbol{U}, \boldsymbol{V}) = \mathbb{E}\exp(\mathrm{i}\omega \boldsymbol{a}^\top (\boldsymbol{P} - \boldsymbol{Q}) \boldsymbol{b}),$$

with the projectors $\boldsymbol{P} = \boldsymbol{U}\boldsymbol{U}^\top$ and $\boldsymbol{Q} = \boldsymbol{V}\boldsymbol{V}^\top$. The rotational invariance of the random vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ thus shows that κ° is rotationally invariant, *i.e.*, $\kappa^\circ(\boldsymbol{U}, \boldsymbol{V}) = \kappa^\circ(\boldsymbol{R}\boldsymbol{U}, \boldsymbol{R}\boldsymbol{V})$, for any rotation $\boldsymbol{R} \in \mathsf{SO}(n)$. From Thm. 2.1, $\kappa^\circ(\boldsymbol{U}, \boldsymbol{V})$ only depends on the principal angles between $\boldsymbol{U}$ and $\boldsymbol{V}$. Moreover, following [47], we can always apply a rotation $\boldsymbol{R}$ such that $\boldsymbol{U}$ and $\boldsymbol{V}$ are "rotated" to make angles $\pm\theta_j/2$ around the identity, *i.e.*,

$$\boldsymbol{U}^\top = (\boldsymbol{C}, \boldsymbol{S}, \mathbf{0}_{k \times n-2k}) \in \mathbb{R}^{k \times n}, \quad \boldsymbol{V}^\top = (\boldsymbol{C}, -\boldsymbol{S}, \mathbf{0}_{k \times n-2k}) \in \mathbb{R}^{k \times n},$$

with

$$\boldsymbol{C} = \operatorname{diag}\big(\cos(\theta_1/2), \ldots, \cos(\theta_k/2)\big)$$

and

$$\boldsymbol{S} = \operatorname{diag}\big(\sin(\theta_1/2), \ldots, \sin(\theta_k/2)\big).$$

Using this representation, a few computations show that, up to a permutation matrix $\mathbf{\Pi} \in \{0,1\}^{n\times n}$, $\boldsymbol{P} - \boldsymbol{Q}$ is block-diagonal with k independent 2×2 blocks, *i.e.*, $\boldsymbol{P} - \boldsymbol{Q} = \mathbf{\Pi}^\top \mathbf{\Gamma} \mathbf{\Pi}$, with

$$\mathbf{\Gamma} := \operatorname{bdiag}\left(\sin\theta_1 \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \dots, \sin\theta_k \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, 0, \dots, 0 \right) \in \mathbb{R}^{n\times n},$$

where we append as many zeros as necessary to get an $n \times n$ matrix. Since the distribution of the vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ is permutation invariant, $\boldsymbol{a}^\top(\boldsymbol{P} - \boldsymbol{Q})\boldsymbol{b}$ is distributed as $\boldsymbol{a}^\top \mathbf{\Gamma} \boldsymbol{b}$, so that $\kappa^\circ(\boldsymbol{U}, \boldsymbol{V}) = \mathbb{E}\exp(\mathrm{i}\boldsymbol{a}^\top \mathbf{\Gamma} \boldsymbol{b})$. However, since $\boldsymbol{a}^\top \mathbf{\Gamma} \boldsymbol{b} = \sum_{j=1}^k \sin\theta_{2j}(a_{2j}b_{2j+1} + a_{2j+1}b_{2j})$, we get from the independence of the components of $\boldsymbol{a}$ and $\boldsymbol{b}$,

$$\begin{aligned} \kappa^\circ(\boldsymbol{U}, \boldsymbol{V}) &= \textstyle\prod_{j=1}^k \mathbb{E}\exp(\mathrm{i}\, a_{2j}b_{2j+1}\, \sin\theta_j)\mathbb{E}\exp(\mathrm{i}\, b_{2j}a_{2j+1}\, \sin\theta_j) \\ &= \textstyle\prod_{j=1}^k \left(\mathbb{E}\exp(\mathrm{i}\sin\theta_j gg')\right)^2, \end{aligned}$$

f with $g, g' \sim_{\text{i.i.d.}} \mathcal{N}(0,1)$. Defining $u = (g+g')/\sqrt{2}$, $v = (g-g')/\sqrt{2}$, we observe that $gg' = \frac{1}{2}(u^2 - v^2)$ with independent $u, v \sim \mathcal{N}(0,1)$, *i.e.*, $2gg'$ is the difference of two independent χ^2-distributions with one degree of freedom. Since for any $t \in \mathbb{R}$ $\mathbb{E}e^{\mathrm{i}tX} = (1-2\mathrm{i}t)^{-1/2}$ if $X \sim \chi^2_1$, we obtain for any $\alpha \in \mathbb{R}$,

$$\mathbb{E}\exp(\mathrm{i}\alpha gg') = (1-i\alpha)^{-1/2}(1+i\alpha)^{-1/2} = (1+\alpha^2)^{-1/2}.$$

This means that,

$$\begin{aligned} \kappa^\circ(\boldsymbol{U}, \boldsymbol{V}) &= \prod_{j=1}^k \left(\mathbb{E}\exp(\mathrm{i}\omega\, gg'\, \sin\theta_j)\right)^2 \\ &= \prod_{j=1}^k (1+\omega^2\sin^2\theta_j)^{-1}, \end{aligned}$$

which completes the proof. □

Remark 4.2. *While the periodic random feature ψ° defined in Eq. 4.20 is complex, the resulting kernel κ° is real,* i.e., *the imaginary part of $\widehat{\kappa}^\circ$ vanishes in expectation. Moreover, we show easily that the $2m$-component periodic random feature map*

$$\psi^{\circ\prime}(\boldsymbol{U}) := \begin{pmatrix} \Re(\psi^\circ(\boldsymbol{U})) \\ \Im(\psi^\circ(\boldsymbol{U})) \end{pmatrix}, \tag{4.22}$$

provides also an unbiased estimator of $\kappa^\circ(\boldsymbol{U}, \boldsymbol{V})$ via $\frac{1}{m}\langle \psi^{\circ\prime}(\boldsymbol{U}), \psi^{\circ\prime}(\boldsymbol{V})\rangle$ since

$$\langle \psi^{\circ\prime}(\boldsymbol{U}), \psi^{\circ\prime}(\boldsymbol{V})\rangle = \Re\big(\langle \psi^\circ(\boldsymbol{U}), \psi^\circ(\boldsymbol{V})\rangle\big).$$

Just like we did for $\kappa^{\pm}$, we now consider the approximation error we make by replacing κ° by $\widehat{\kappa}^{\circ}$. We first study a bound on the pointwise approximation error, *i.e.*, the absolute error made on a fixed pair of subspaces.

Proposition 4.11 (Pointwise approximation error of κ° by $\widehat{\kappa}^{\circ}$)**.** *Given two subspace bases $\boldsymbol{U}, \boldsymbol{V} \in \mathbb{V}(k, n)$, the m-component periodic random feature map ψ° in Eq. 4.20 and related to m i.i.d. Gaussian random vectors $\boldsymbol{a}_j, \boldsymbol{b}_j \sim \mathcal{N}(\boldsymbol{0}, \mathbf{I}_n)$, and an error $\delta > 0$, we have*

$$\mathbb{P}\Big(\big|\tfrac{1}{m}\langle \psi^{\circ}(\boldsymbol{U}), \psi^{\circ}(\boldsymbol{V})\rangle - \kappa^{\circ}(\boldsymbol{U}, \boldsymbol{V})\big| < \delta\Big) \geq 1 - 4\exp(-\tfrac{1}{4}m\delta^2).$$

Proof. Defining the m i.i.d. complex random variables

$$X_j := {\psi^{\circ}}_j(\boldsymbol{U}){\psi^{\circ}}_j^*(\boldsymbol{V}), \quad \text{for} \quad 1 \leq j \leq m,$$

we have

$$m\widehat{\kappa}^{\circ}(\boldsymbol{U}, \boldsymbol{V}) = \textstyle\sum_{j=1}^m X_j, \quad m\kappa^{\circ}(\boldsymbol{U}, \boldsymbol{V}) = \sum_{j=1}^m \mathbb{E}X_j$$

and

$$\begin{aligned}\mathbb{P}\big[|\widehat{\kappa}^{\circ}(\boldsymbol{U}, \boldsymbol{V}) - \kappa^{\circ}(\boldsymbol{U}, \boldsymbol{V})| \geq \delta\big] \\ \leq \mathbb{P}\big[|\tfrac{1}{m}\textstyle\sum_{j=1}^m \Re(X_j) - \Re\big(\kappa^{\circ}(\boldsymbol{U}, \boldsymbol{V})\big)| \geq \delta/\sqrt{2}\big] \\ + \mathbb{P}\big[|\tfrac{1}{m}\textstyle\sum_{j=1}^m \Im(X_j) - \Im\big(\kappa^{\circ}(\boldsymbol{U}, \boldsymbol{V})\big)| \geq \delta/\sqrt{2}\big].\end{aligned}$$

Since $\max(|\Re(X_j)|, |\Im(X_j)|) \leq 1$, Hoeffding's inequality provides

$$\mathbb{P}[|\tfrac{1}{m}\textstyle\sum_{j=1}^m \Re(X_j) - \Re\kappa^{\circ}(\boldsymbol{U}, \boldsymbol{V})| \geq \delta/\sqrt{2}] \leq 2\exp(-\tfrac{1}{4}m\delta^2),$$

and similarly for the imaginary part. Gathering the bounds provides the result. □

Armed with these results we can now attack the uniform bound in the same way as we did for $\kappa^{\pm}$. The statement and the proof follow a very similar pattern.

Proposition 4.12 (Uniform approximation error for $\widehat{\kappa}^\circ \approx \kappa^\circ$)**.** *Given a distortion $\delta > 0$, and some absolute constants $C, C', c > 0$, if*

$$m \geq C\delta^{-2} nk \log(\omega\sqrt{k}n/\delta),$$

then

$$\sup_{\boldsymbol{U},\boldsymbol{V}\in\mathbb{V}(k,n)} \left|\widehat{\kappa}^\circ(\boldsymbol{U},\boldsymbol{V}) - \kappa^\circ(\boldsymbol{U},\boldsymbol{V})\right| \leq \delta$$

with probability at least $1 - C'\exp(-c\delta^2 m)$, for some absolute constant $c, c_0 > 0$.

Proof. The proof for this proposition is delayed to Sec. 4.5.2 for more clarity. □

Prop. 4.12 shows that, as in the signed case, a number of random features proportional to the intrinsic dimension nk of the problem (up to log factors) is sufficient to obtain a uniform approximation of the target kernel with high probability.

Remark 4.3. *The uniform guarantees of Prop. 4.12 and Prop. 4.4 are worst-case results over all pairs of k-dimensional subspaces in $\mathbb{R}^n$. Their dependences on nk are therefore coherent with the intrinsic dimension of $\mathcal{G}(k,n)$. It may be possible to improve requirements on m when restricting the approximation to structured subsets of $\mathcal{G}(k,n)$, for instance subspaces with sparse bases, or subspaces satisfying additional constraints on their principal angles. In this case, the proof would require replacing the covering number of the full Grassmannian by a covering number adapted to the restricted class. We leave this direction for future work.*

4.5.2 Proof of Prop. 4.12

In this section we prove the uniform concentration result stated in Prop. 4.12.

Proof. Given a radius $\epsilon > 0$ to be fixed later, we know from Prop. 4.5 that there exists an ϵ-covering $\mathcal{N}_\epsilon \subset \mathbb{G}(k,n)$ of $\mathbb{G}(k,n)$ with cardinality $|\mathcal{N}_\epsilon| \leq (18\sqrt{k}/\epsilon)^{(2n+1)k}$. As in the proof of Prop. 4.4, we consider a set $\mathcal{V}_\epsilon \subset \mathbb{V}(k,n)$ such that for any $\boldsymbol{U}^* \in \mathcal{V}_\epsilon$, $\boldsymbol{P}^* := \boldsymbol{U}^*\boldsymbol{U}^{*\top} \in \mathcal{N}_\epsilon$, and such that each $\boldsymbol{P}^* \in \mathcal{N}_\epsilon$ is represented by a unique basis in $\mathcal{V}_\epsilon$, *i.e.*, $|\mathcal{V}_\epsilon| = |\mathcal{N}_\epsilon|$.

The proof proceeds in a similar way to that of Prop. 4.4, by decomposing the approximation error into three terms and then controlling

each term separately. For arbitrary $\boldsymbol{U}, \boldsymbol{V} \in \mathbb{V}(k,n)$ and associated net points $\boldsymbol{U}^*, \boldsymbol{V}^* \in \mathcal{V}_\epsilon$, with projectors $\boldsymbol{P}^* = \boldsymbol{U}^*\boldsymbol{U}^{*\top}$ and $\boldsymbol{Q}^* = \boldsymbol{V}^*\boldsymbol{V}^{*\top}$, we write

$$\left|\widehat{\kappa}^\circ(\boldsymbol{U},\boldsymbol{V}) - \kappa^\circ(\boldsymbol{U},\boldsymbol{V})\right| \leq T_1 + T_2 + T_3,$$

with $T_1 := \left|\widehat{\kappa}^\circ(\boldsymbol{U},\boldsymbol{V}) - \widehat{\kappa}^\circ(\boldsymbol{U}^*,\boldsymbol{V}^*)\right|$, $T_2 := \left|\widehat{\kappa}^\circ(\boldsymbol{U}^*,\boldsymbol{V}^*) - \kappa^\circ(\boldsymbol{U}^*,\boldsymbol{V}^*)\right|$, and $T_3 := \left|\kappa^\circ(\boldsymbol{U}^*,\boldsymbol{V}^*) - \kappa^\circ(\boldsymbol{U},\boldsymbol{V})\right|$.

We now bound each of the three terms by $\delta/3$ in order to ensure a total approximation error bounded by δ.

(a) Bound on T_1: Let us first consider the event $\mathcal{E}_1 := \{\frac{1}{mn}\sum_{j=1}^m(\|\boldsymbol{a}_j\|_2^2 + \|\boldsymbol{b}_j\|_2^2) \leq 3\}$. Since $\boldsymbol{a}_j, \boldsymbol{b}_j \sim_{\text{i.i.d.}} \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}_n)$, the random variable $Z := \sum_{j=1}^m(\|\boldsymbol{a}_j\|_2^2 + \|\boldsymbol{b}_j\|_2^2)$ is distributed as a χ^2_{2mn}-distribution with $2mn$ degrees of freedom, and $\mathbb{E}Z = 2mn$. Moreover, by the Laurent–Massart inequality [95, Lemma 1], there exist absolute constants $c > 0$ such that

$$\mathbb{P}(\mathcal{E}_1) \geq 1 - \exp(-cmn). \tag{4.23}$$

Let us assume that $\mathcal{E}_1$ holds. For any $\boldsymbol{P}, \boldsymbol{Q} \in \mathbb{G}(k,n)$, let $f(\boldsymbol{P},\boldsymbol{Q}) := \frac{1}{m}\sum_{j=1}^m \exp\left(i\omega \boldsymbol{a}_j^\top(\boldsymbol{P}-\boldsymbol{Q})\boldsymbol{b}_j\right)$, and $\boldsymbol{P}^*, \boldsymbol{Q}^* \in \mathbb{G}(k,n)$ be associated net points with $\|\boldsymbol{P}-\boldsymbol{P}^*\|_F \leq \epsilon$ and $\|\boldsymbol{Q}-\boldsymbol{Q}^*\|_F \leq \epsilon$. Then

$$\begin{aligned} mT_1 = m|f(\boldsymbol{P},\boldsymbol{Q}) - f(\boldsymbol{P}^*,\boldsymbol{Q}^*)| \leq \sum_{j=1}^m \big|\exp\left(i\omega \boldsymbol{a}_j^\top(\boldsymbol{P}-\boldsymbol{Q})\boldsymbol{b}_j\right) \\ - \exp\left(i\omega \boldsymbol{a}_j^\top(\boldsymbol{P}^*-\boldsymbol{Q}^*)\boldsymbol{b}_j\right)\big|. \end{aligned}$$

Using $|\exp(ix) - \exp(iy)| \leq |x-y|$ for any $x, y \in \mathbb{R}$, we obtain

$$m|f(\boldsymbol{P},\boldsymbol{Q}) - f(\boldsymbol{P}^*,\boldsymbol{Q}^*)| \leq \omega \sum_{j=1}^m \left|\boldsymbol{a}_j^\top\left((\boldsymbol{P}-\boldsymbol{P}^*) - (\boldsymbol{Q}-\boldsymbol{Q}^*)\right)\boldsymbol{b}_j\right|.$$

Hence, by the triangle inequality,

$$|f(\boldsymbol{P},\boldsymbol{Q}) - f(\boldsymbol{P}^*,\boldsymbol{Q}^*)| \leq \frac{\omega}{m}\sum_{j=1}^m \Big(|\boldsymbol{a}_j^\top(\boldsymbol{P}-\boldsymbol{P}^*)\boldsymbol{b}_j| + |\boldsymbol{a}_j^\top(\boldsymbol{Q}-\boldsymbol{Q}^*)\boldsymbol{b}_j|\Big).$$

Using Cauchy–Schwarz gives $|\boldsymbol{a}_j^\top(\boldsymbol{P}-\boldsymbol{P}^*)\boldsymbol{b}_j| = |\langle \boldsymbol{P}-\boldsymbol{P}^*, \boldsymbol{a}_j\boldsymbol{b}_j^\top\rangle| \leq \|\boldsymbol{P}-\boldsymbol{P}^*\|_F\|\boldsymbol{a}_j\|_2\|\boldsymbol{b}_j\|_2$, and similarly for $\boldsymbol{Q}$. Using $\|\boldsymbol{P}-\boldsymbol{P}^*\|_F \leq \epsilon$, $\|\boldsymbol{Q}-\boldsymbol{Q}^*\|_F \leq \epsilon$ and $\|x\|\|y\| \leq (\|x\|^2 + \|y\|^2)/2$ yields

$$\begin{aligned} T_1 = |f(\boldsymbol{P},\boldsymbol{Q}) - f(\boldsymbol{P}^*,\boldsymbol{Q}^*)| &\leq \frac{2\omega\epsilon}{m}\sum_{j=1}^m \|\boldsymbol{a}_j\|_2\|\boldsymbol{b}_j\|_2 \\ &\leq \frac{\omega\epsilon}{m}\sum_{j=1}^m \left(\|\boldsymbol{a}_j\|_2^2 + \|\boldsymbol{b}_j\|_2^2\right). \end{aligned}$$

Therefore, under the event $\mathcal{E}_1$, $T_1 \leq 3\omega n\epsilon$. In particular, choosing $\epsilon = \frac{\delta}{9\omega n}$ ensures $T_1 \leq \frac{\delta}{3}$.

(b) Bound on T_2: We now control the second term T_2. For any fixed $\boldsymbol{U}^*, \boldsymbol{V}^* \in \mathcal{V}_\epsilon$, with associated projectors $\boldsymbol{P}^* = \phi^{\text{P}}(\boldsymbol{U}^*)$ and $\boldsymbol{Q}^* =$

$\phi^{\mathrm{P}}(\boldsymbol{V}^*)$, Prop. 4.11 ensures the existence of absolute constants $C, c > 0$ such that

$$\mathbb{P}\Big(\big|\widehat{\kappa}^{\circ}(\boldsymbol{U}^*, \boldsymbol{V}^*) - \kappa^{\circ}(\boldsymbol{U}^*, \boldsymbol{V}^*)\big| \geq \tfrac{\delta}{3}\Big) \leq C \exp(-cm\delta^2).$$

Let us define the second event

$$\mathcal{E}_2 = \Big\{\forall \boldsymbol{U}^*, \boldsymbol{V}^* \in \mathcal{V}_\epsilon, \big|\widehat{\kappa}^{\circ}(\boldsymbol{U}^*, \boldsymbol{V}^*) - \kappa^{\circ}(\boldsymbol{U}^*, \boldsymbol{V}^*)\big| < \tfrac{\delta}{3}\Big\}.$$

Since $|\mathcal{V}_\epsilon| = |\mathcal{N}_\epsilon|$, a union bound over all pairs $\boldsymbol{U}^*, \boldsymbol{V}^* \in \mathcal{V}_\epsilon$ yields

$$\mathbb{P}(\mathcal{E}_2^c) \leq C|\mathcal{N}_\epsilon|^2 \exp(-cm\delta^2).$$

Using the covering bound from Prop. 4.5 with the value of $\epsilon = \delta/9\omega n$ set above, we have $|\mathcal{N}_\epsilon| \leq (18\sqrt{k}/\epsilon)^{(2n+1)k} \leq (162\omega\sqrt{k}n/\delta)^{(2n+1)k}$, and hence, for some absolute constants $C, C', c > 0$,

$$\mathbb{P}(\mathcal{E}_2^c) \leq C \exp\Big(C' nk \log(\omega\sqrt{k}n/\delta) - cm\delta^2\Big).$$

In particular, updating the constants, if

$$m \geq C\delta^{-2} nk \log(\omega\sqrt{k}n/\delta),$$

then $\mathbb{P}(\mathcal{E}_2^c) \leq C' \exp(-cm\delta^2)$. Under this condition, $T_2 \leq \frac{\delta}{3}$ simultaneously for all pairs of $\boldsymbol{U}^*, \boldsymbol{V}^* \in \mathcal{V}_\epsilon$ with probability at least $1 - C' \exp(-cm\delta^2)$.
(c) Bound on T_3: We now control the deterministic term

$$T_3 = \big|\kappa^{\circ}(\boldsymbol{U}, \boldsymbol{V}) - \kappa^{\circ}(\boldsymbol{U}^*, \boldsymbol{V}^*)\big|.$$

Let us define for $\boldsymbol{P}, \boldsymbol{Q} \in \mathbb{G}(k, n)$, $h(\boldsymbol{P}, \boldsymbol{Q}) = \exp\big(i\omega \boldsymbol{a}^\top (\boldsymbol{P} - \boldsymbol{Q})\boldsymbol{b}\big)$, so that $\kappa^{\circ}(\boldsymbol{U}, \boldsymbol{V}) = \mathbb{E}[h(\boldsymbol{P}, \boldsymbol{Q})]$. By linearity of expectation and the triangle inequality,

$$T_3 \leq \mathbb{E}\big[|h(\boldsymbol{P}, \boldsymbol{Q}) - h(\boldsymbol{P}^*, \boldsymbol{Q}^*)|\big].$$

Using again $|\exp(ix) - \exp(iy)| \leq |x - y|$, we obtain by Cauchy–Schwarz

$$\begin{aligned} |h(\boldsymbol{P}, \boldsymbol{Q}) - h(\boldsymbol{P}^*, \boldsymbol{Q}^*)| &\leq \omega\big(|\boldsymbol{a}^\top (\boldsymbol{P} - \boldsymbol{P}^*)\boldsymbol{b}| + |\boldsymbol{a}^\top (\boldsymbol{Q} - \boldsymbol{Q}^*)\boldsymbol{b}|\big) \\ &\leq \omega(\|\boldsymbol{P} - \boldsymbol{P}^*\|_F + \|\boldsymbol{Q} - \boldsymbol{Q}^*\|_F)\|\boldsymbol{a}\|_2\|\boldsymbol{b}\|_2, \end{aligned}$$

so that, taking expectations, using $\mathbb{E}[\|\boldsymbol{a}\|_2\|\boldsymbol{b}\|_2] \leq n$ and the value of ϵ set above,

$$T_3 \leq \omega\big(\|\boldsymbol{P} - \boldsymbol{P}^*\|_F + \|\boldsymbol{Q} - \boldsymbol{Q}^*\|_F\big)\mathbb{E}[\|\boldsymbol{a}\|_2\|\boldsymbol{b}\|_2] \leq 2\omega n\epsilon \leq \tfrac{2}{9}\delta < \tfrac{1}{3}\delta,$$

which holds uniformly over all $\boldsymbol{U}, \boldsymbol{V} \in \mathbb{V}(k, n)$.

Final bound: We now combine the bounds $T_1 \leq \delta/3$, $T_2 \leq \delta/3$, and $T_3 \leq \delta/3$ obtained above, as well as the probabilities (by union bound) and conditions under which they hold. We proved that $\mathbb{P}[T_1 \leq \delta/3|\mathcal{E}_1] = 1$ and $\mathbb{P}[T_2 \leq \delta/3|\mathcal{E}_2] = 1$, with also $\mathbb{P}[T_3 \leq \delta/3] = 1$. Therefore, $\mathbb{P}[T_1 + T_2 + T_3 > \delta] \leq \mathbb{P}[T_1 > \delta/3] + \mathbb{P}[T_2 > \delta/3]$. However,

$$\begin{aligned}\mathbb{P}[T_1 > \delta/3] &= 1 - \mathbb{P}[T_1 \leq \delta/3|\mathcal{E}_1]\mathbb{P}[\mathcal{E}_1] - \mathbb{P}[T_1 \leq \delta/3|\mathcal{E}_1^c]\mathbb{P}[\mathcal{E}_1^c] \\ &= 1 - \mathbb{P}[\mathcal{E}_1] - \mathbb{P}[T_1 \leq \delta/3|\mathcal{E}_1^c]\mathbb{P}[\mathcal{E}_1^c] \leq \mathbb{P}[\mathcal{E}_1^c],\end{aligned}$$

and, similarly, $\mathbb{P}[T_2 > \delta/3] \leq \mathbb{P}[\mathcal{E}_2^c]$. This shows that, for some constants $C, C', c > 0$, provided that

$$m \geq C\delta^{-2} nk \log(\omega\sqrt{k}n/\delta),$$

with probability exceeding $1 - \mathbb{P}(\mathcal{E}_1^c) - \mathbb{P}(\mathcal{E}_2^c) \geq 1 - C'\exp(-c\delta^2 m)$, we have

$$\sup_{\boldsymbol{U},\boldsymbol{V}\in\mathbb{V}(k,n)} \left|\widehat{\kappa}^\circ(\boldsymbol{U}, \boldsymbol{V}) - \kappa^\circ(\boldsymbol{U}, \boldsymbol{V})\right| = T_1 + T_2 + T_3 \leq \delta,$$

which concludes the proof. □

4.5.3 Properties

We conclude this section by studying the shape and properties of the kernel κ° and its dependence on the frequency parameter ω; in particular, we show below that this parameter determines two specific regimes where κ° is either close to the projection kernel or to a kernel that is reminiscent of the Binet-Cauchy kernel. The first regime is valid at large frequency.

Proposition 4.13 (Large-frequency regime of κ°)**.** *Let us consider two subspace bases $\boldsymbol{U}, \boldsymbol{V} \in \mathbb{V}(k, n)$ with projectors $\boldsymbol{P}$ and $\boldsymbol{Q}$ and principal angles $0 < \theta_1 \leq \cdots \leq \theta_k \leq \frac{\pi}{2}$. We have*

$$\lim_{\omega\to+\infty} \omega^{2k}\kappa^\circ(\boldsymbol{U}, \boldsymbol{V}) = \prod_{j=1}^k (\sin^2\theta_j)^{-1} = \operatorname{pdet}(\boldsymbol{P}(\mathbf{I}_n - \boldsymbol{Q})\boldsymbol{P})^{-1}.$$

Proof. This is a direct consequence of Eq. 4.21. □

Interestingly, we can relate the shape of κ° in this large-frequency limit to an extension of the Binet-Cauchy kernel. Generalising this kernel to subspaces $\boldsymbol{U}_1$ and $\boldsymbol{U}_2$ of different dimensions (k_1 and k_2, respectively) and principal angles $\{\theta'_j\}_{j=1}^{\min(k_1,k_2)}$, through the definition

$$\lim_{\omega\to+\infty} \omega^{2k}\kappa^\circ(\boldsymbol{U}, \boldsymbol{V}) := \prod_{j=1}^{\min(k_1,k_2)} \cos^2\theta'_j,$$

we get, under the conditions of Prop. 4.13 and if $k < n/2$,

$$\kappa^{\circ}(\boldsymbol{U}, \boldsymbol{V}) = \kappa^{\mathrm{BC}}(\boldsymbol{U}, \boldsymbol{V}^{\perp})^{-1},$$

since $\boldsymbol{U}$ and $\boldsymbol{V}^{\perp}$ then make k principal angles $\{\frac{\pi}{2} - \theta_j\}_{j=1}^{k}$ [113, 94]. Following a previous interpretation, this shows also that, in this regime, the kernel $\kappa^{\circ}(\boldsymbol{U}, \boldsymbol{V})$ is inversely proportional to the volume made by the basis $\boldsymbol{U}$ when projected onto $\boldsymbol{V}$.

The second regime is reached at small frequency. We show below that κ° then mimics a radial basis function (RBF) kernel (see Eq. 2.15).

Proposition 4.14 (Small-frequency regime of κ°)**.** *Let us consider two subspace bases $\boldsymbol{U}, \boldsymbol{V} \in \mathbb{V}(k, n)$ with projectors $\boldsymbol{P}$ and $\boldsymbol{Q}$. We have*

$$\left|\kappa^{\circ}(\boldsymbol{U}, \boldsymbol{V}) - \exp\left(-\tfrac{1}{2}\omega^2 \|\boldsymbol{P} - \boldsymbol{Q}\|_F^2\right)\right| \leq \tfrac{1}{4}\omega^4 \|\boldsymbol{P} - \boldsymbol{Q}\|_F^2 \leq \tfrac{k}{2}\omega^4.$$

Proof. Given the two subspace principal angles $\{\theta_j\}_{j=1}^{k}$, since

$$\log \kappa^{\circ}(\boldsymbol{U}, \boldsymbol{V}) = -\sum_{j=1}^{k} \log(1 + \omega^2 \sin^2 \theta_j) \text{ and } |\log(1+t) - t| \leq t^2/2$$

for any $t > 0$, we get

$$|\log \kappa^{\circ}(\boldsymbol{U}, \boldsymbol{V}) + \omega^2 \sum_{j=1}^{k} \sin^2 \theta_j| \leq \frac{1}{2}\omega^4 \sum_{j=1}^{k} \sin^4 \theta_j \leq \frac{1}{2}\omega^4 \sum_{j=1}^{k} \sin^2 \theta_j.$$

Therefore, since $|e^u - e^v| \leq \max(e^u, e^v)|u - v|$ for real u and v, we get

$$\left|\kappa^{\circ}(\boldsymbol{U}, \boldsymbol{V}) - \exp(-\omega^2 \textstyle\sum_{j=1}^{k} \sin^2 \theta_j)\right| \leq \tfrac{1}{2}\omega^4 \textstyle\sum_{j=1}^{k} \sin^2 \theta_j,$$

where we used the fact that, by design,

$$\kappa^{\circ}(\boldsymbol{U}, \boldsymbol{V}) \leq 1, \quad \text{and} \quad \exp(-\omega^2 \sum_{j=1}^{k} \sin^2 \theta_j) \leq 1.$$

The result is obtained by observing that $2\sum_{j=1}^{k} \sin^2 \theta_j = \|\boldsymbol{P} - \boldsymbol{Q}\|_F^2$. □

In summary, Props. 4.13 and 4.14 show that κ° can adopt two distinct behaviours: at large frequency ($\omega \gg 1$) $\kappa^{\circ}(\boldsymbol{U}, \boldsymbol{V})$ is related to a variant of the Binet-Cauchy kernel between $\boldsymbol{U}$ and the orthogonal subspace $\boldsymbol{V}^{\perp}$; it is also more sensitive to small changes in the subspace principal angles. By contrast, at low frequency ω, κ° is well approximated

by a RBF kernel in the projection distance $\|\boldsymbol{P} - \boldsymbol{Q}\|_F$. This gives a function more tolerant to small variations of angles. These characteristics are numerically confirmed in Sec. 4.7 when comparing 2-dimensional subspaces. In practice, choosing ω amounts to trading off sensitivity to fine geometric variations against stability.

4.6 Computational and memory costs

The rapidity and efficiency with which we compute the three random feature maps $\psi^{\pm}$, ψ^{Σ} and ψ° (introduced in Sec. 4.3 and 4.5) on subspaces of $\mathcal{G}(k, n)$ depend on those of the ROP projection operator ψ^{rop}.

A priori, when we compute the m-component ROP $\psi^{\text{rop}}(\boldsymbol{U})$ of a subspace basis $\boldsymbol{U} \in \mathbb{V}(k, n)$, each component $1 \leq j \leq m$ of ψ^{rop} requires storing $\mathcal{O}(n)$ values from the random generation of the probing vectors $\boldsymbol{a}_j, \boldsymbol{b}_j \sim_{\text{i.i.d.}} \mathcal{N}(\boldsymbol{0}, \mathbf{I}_n)$. It also needs $\mathcal{O}(kn)$ operations from the cost of computing $\boldsymbol{a}_j^\top \boldsymbol{U}\boldsymbol{U}^\top \boldsymbol{b}_j$ from the inner product of $\bar{\boldsymbol{a}}_j := \boldsymbol{U}^\top \boldsymbol{a}_j$ and $\bar{\boldsymbol{b}}_j := \boldsymbol{U}^\top \boldsymbol{b}_j$. This way of factoring the computation also requires storing $\mathcal{O}(km)$ intermediate values, those from the k-component vectors $\bar{\boldsymbol{a}}_j$ and $\bar{\boldsymbol{b}}_j$, and an additional computational complexity $\mathcal{O}(km)$ for the intermediate inner products evaluation. Consequently, the projection of one subspace into $\mathbb{R}^m$ through ψ^{rop} needs a total storage of $\mathcal{O}(mn + km)$ values and a computational complexity of $\mathcal{O}(kmn + km) = \mathcal{O}(kmn)$.

Regarding the complexities of $\psi^{\pm}$ and ψ°, considering that evaluating their respective non-linear functions brings negligible extra computations, we established through Props. 4.4 and 4.12 that we must take $m = \mathcal{O}(\delta^{-2} kn)$ random features, up to log factors, to ensure that the approximation error of the target kernels is uniformly bounded by $\delta > 0$. The overall memory complexity of $\psi^{\pm}$ and ψ° is thus $\mathcal{O}(\delta^{-2}(kn^2 + k^2 n))$ with a computational complexity of $\mathcal{O}(\delta^{-2} k^2 n^2)$.

Overall, for large values of k and n, these complexities are lower than those of the dense random projections Ψ introduced in Sec. 4.2. Since dense random projections require us to pick $m = \mathcal{O}(\delta^{-2} k^3 n)$ projections to approximate the projection kernel with uniform error δ, they have a complexity $\mathcal{O}(\delta^{-2} k^3 n^3)$ for both storing the m probing matrices of size n^2 and computing the m dense projections. Nevertheless, the complexities discussed so far describe the cost of embedding a single subspace and do not explicitly take into account the number N of instances in a dataset. This dependence is important when comparing random features with the direct use of Grassmannian kernels since random-feature methods require embedding each of the N instances, whereas exact kernel methods require pairwise kernel evaluations and

Random feat.	# of feat. m	Memory	Computation	Bound	Target kernel
Dense Ψ	$\mathcal{O}(\delta^{-2}k^3n)$	$\mathcal{O}(\delta^{-2}k^3n^3)$	$\mathcal{O}(\delta^{-2}k^3n^3)$	Prop. 4.1	Proj. kernel κ^{p}
Binary $\psi^{\pm}$	$\mathcal{O}(\delta^{-2}kn)$	$\mathcal{O}(\delta^{-2}kn^2)$	$\mathcal{O}(\delta^{-2}k^2n^2)$	Prop. 4.4	New kernel: $\kappa^{\pm}$
Aggregated ψ^{Σ}	$\mathcal{O}(\delta^{-2}kn)$	$\mathcal{O}(\delta^{-2}kn^2N_\Sigma)$	$\mathcal{O}(\delta^{-2}k^2n^2N_\Sigma)$	Assumed	Close to κ^{Σ}_{∞}
Periodic ψ°	$\mathcal{O}(\delta^{-2}kn)$	$\mathcal{O}(\delta^{-2}kn^2)$	$\mathcal{O}(\delta^{-2}k^2n^2)$	Prop. 4.12	New kernel: κ°

Table 4.1 Computational and memory complexities per instance for k-dimensional subspaces of $\mathbb{R}^n$ to get a uniform approximation error $\delta > 0$ between the empirical kernel and the related target kernel.

Method	Computation	Memory
Exact kernels $\kappa^{\mathrm{p}}, \kappa^{\mathrm{BC}}, \kappa^{\circ}, \kappa^{\Sigma}_{\infty}$	$\mathcal{O}(N^2nk^2)$	$\mathcal{O}(N^2)$
Binary/periodic ROP $\psi^{\pm}, \psi^{\circ}$	$\mathcal{O}(Nk^2n^2)$	$\mathcal{O}(kn^2 + Nkn)$
Dense embedding ψ^{d}	$\mathcal{O}(Nk^3n^3)$	$\mathcal{O}(k^3n^3 + Nk^3n)$

Table 4.2 Computational and memory complexities for N subspaces of dimension k in $\mathbb{R}^n$. For the random-feature methods, we use the number of features required to obtain a fixed uniform approximation error δ, *i.e.*, , $m = \mathcal{O}(kn)$ for the binary and periodic ROP embeddings and $m = \mathcal{O}(k^3n)$ for the dense embedding, up to log factors. For the direct kernel methods, the computation includes constructing the $N \times N$ Gram matrix. For the random-feature methods, it is equivalent to embedding all N subspaces, and the memory includes both the random vectors and the resulting feature representations. Methods with the same complexities are grouped together.

the storage of an $N \times N$ Gram matrix. Consequently, exact kernels may remain computationally more efficient for small datasets, while random-feature methods become increasingly interesting as N grows, since their computation and storage scale linearly rather than quadratically with N. We provide this comparison in Tab. 4.2.

The computational and memory requirements for the embedding procedure will be further alleviated in Chap. 5 where we will introduce structured embeddings and Optical Processing Units, capable of replicating the effects of ψ^{rop}. The per-instance complexity results are summarised in Tab. 4.1, while the equivalent table taking the dataset size into account can be found in Tab. 4.2.

Remark 4.4. *The computational complexities shown in Tables 4.1 and 4.2 count the total number of operations and should thus not be interpreted as execution times on a particular hardware architecture. It should be noted that random projections are naturally well suited to parallel computation, since they can be formulated with matrix operations that are efficiently parallelised on GPUs [143, 76]. Dense random projections can especially benefit from highly optimised parallel matrix operations, while ROP and structured projections can also be parallelised. GPU computation could therefore substan-*

tially reduce practical execution times compared with a CPU implementation, without changing the asymptotic number of operations required by the different methods. The complexity comparisons above should thus be understood as comparisons of the total computational work involved, rather than as direct predictions of running time.

Similarly, the memory costs assume that the random operators are explicitly stored. When the random matrices or vectors are generated pseudo-randomly, we may instead store only the random seed and regenerate their coefficients when required. This can strongly reduce their storage requirements. It does not, however, remove the need to generate and apply these coefficients, and therefore does not eliminate their computational cost. Depending on the implementation, regenerated coefficients may also need to be stored temporarily or generated progressively during the computation.

4.7 Experiments

In this section we demonstrate experimentally the results developed in theory in the previous section. We start with some experiments to confirm the results in expectation of our kernels? We then analyse the geometric properties of those. Finally, we demonstrate on a real dataset how our kernel approximations can be used in a real setting.

4.7.1 Dense projections versus rank-one projections

As a first analysis, we want to stress the computational advantage of using (unbounded) rank-one projections instead of dense Gaussian projections with the objective of approximating the projection kernel κ^{p}. Indeed, the related empirical kernels both approximate κ^{p}, but their computational and memory costs differ substantially, since dense projections require storing m full $n \times n$ matrices while ROP features only require m pairs of vectors in $\mathbb{R}^n$. We remind that, as stated in Sec. 4.2.2, while unbounded ROPs allow for an approximation of the κ^{p} on finite datasets, they are theoretically not optimal when we target a uniform characterisation of this kernel over the whole space $\mathcal{G}(k, n)$.

To this end, we generate 100 pairs of subspaces in $\mathcal{G}(5, 100)$ with pre-selected principal angles. For each value of m, we compare the approximated $\widehat{\kappa}^{\mathrm{d}}$ and $\widehat{\kappa}^{\mathrm{rop}}$ with the exact projection kernel κ^{p}. The experiment is repeated five times, and Fig. 4.3 shows the average mean and maximum absolute errors of these approximations.

Both empirical kernels converge to κ^{p} as m increases. Dense projections are slightly more accurate, however this comes at a much larger computational and memory cost. In this setting, storing dense matri-

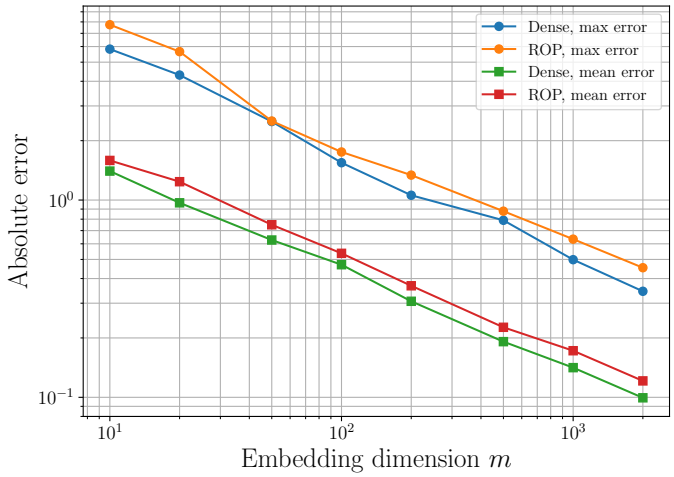


Fig. 4.3 Approximation of the projection kernel with $\widehat{\kappa}^{\mathrm{d}}$ and $\widehat{\kappa}^{\mathrm{rop}}$ on $\mathcal{G}(5,100)$. Both empirical kernels target κ^{p}. Both approximations converge, although dense projections are slightly more accurate, this comes at the cost of a much higher computational and memory requirements.

ces requires mn^2 real values, whereas storing ROP vectors requires only $2mn$. For $m = 2000$, this makes $\widehat{\kappa}^{\mathrm{d}}$ about $50\times$ less efficient than $\widehat{\kappa}^{\mathrm{rop}}$ in terms of memory and about $125\times$ slower in terms of execution time. This illustrates the trade-off between a slight accuracy gain and computational efficiency, thus motivating the use of rank-one projections.

4.7.2 Comparing the kernels

We then illustrate the geometry of the periodic Grassmann kernel in the simple case $k = 2$. Using the closed form expression, we evaluate κ° on a uniform grid of principal angles $(\theta_1, \theta_2) \in [0, \pi/2]^2$ and display the resulting 200×200 heatmaps for three values of $\omega \in \{0.5, 1, 2\}$.

We also execute the same experiment using the exact projection kernel κ^{p}, the Binet-Cauchy kernel κ^{BC} and the empirical sign kernel $\kappa^{\pm}$, evaluated with the formula of Prop. 4.2. The corresponding heat maps are illustrated in the lower row of Fig. 4.4. We see that all four kernels share the property of being large when the principal angles are small but their behaviour varies slightly as shown by the level curves. κ^{BC} and κ° with large ω show more sensitivity for small angles, κ^{p}, $\kappa^{\pm}$ and κ° with reasonable ω are well spread, and κ° with small ω shows a lot less sensitivity to large angles. This confirms that κ° gives a trade-off between a locally and a globally sensitive similarity metric on $\mathcal{G}(2, n)$.

Next, in order to illustrate Prop. 4.9, we numerically compare the expected aggregated kernel κ^{Σ} with its explicit approximation κ^{Σ}_{∞}. We

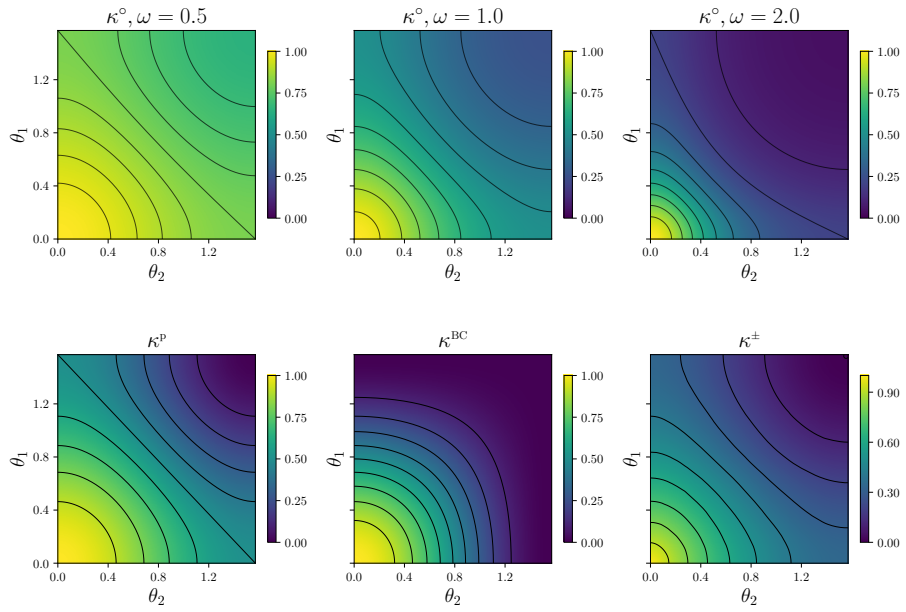


Fig. 4.4 Top: geometry of the periodic Grassmannian kernel κ° for $k = 2$ for various ω. Bottom: comparison between the projection, Binet-Cauchy and sign kernels (all normalised to $[0, 1]$) with $k = 2$.

estimate κ^Σ directly by sampling independent aggregated coordinates for various values of N_Σ.

We test several configurations of principal angles. We take $n = 512$ and consider the cases $k = 1, 2, 3$. For $k = 1$, we use 25 angles uniformly spaced between 0 and $\pi/2$. For $k = 2$ and $k = 3$, we use 25 principal-angle vectors sampled uniformly at random from $[0, \pi/2]^k$.

For each value of N_Σ and each principal-angle configuration, we generate $m = 5 \times 10^3$ independent aggregated coordinates and compute

$$\widehat{\kappa}^\Sigma(\boldsymbol{U}, \boldsymbol{V}) = \tfrac{1}{m} \langle \psi^\Sigma(\boldsymbol{U}), \psi^\Sigma(\boldsymbol{V}) \rangle.$$

Keeping m deliberately large allows us to focus our attention on the effects of N_Σ. We then compare this estimate with the explicit limiting approximation

$$\kappa^\Sigma_\infty(\boldsymbol{U}, \boldsymbol{V}) = \tfrac{2}{\pi} \arcsin\left(\tfrac{1}{k} \textstyle\sum_{i=1}^k \cos^2 \theta_i\right).$$

We consider several values of N_Σ. For each k and each value of N_Σ, we compute the absolute error $|\widehat{\kappa}^\Sigma - \kappa^\Sigma_\infty|$ for all tested principal-angle configurations, and show the average of these errors. The resulting curves are plotted as functions of N_Σ, along with a reference (dotted) line proportional to $N_\Sigma^{-1/2}$.

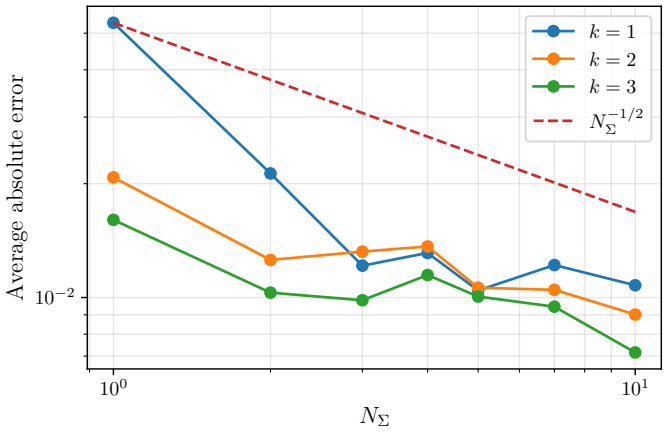


Fig. 4.5 Numerical verification of the convergence of κ^Σ to κ^Σ_∞ for $n = 512$ and $k = 1, 2, 3$. For each value of N_Σ, we test 25 configurations of principal angles and estimate the expectation defining κ^Σ using 5×10^3 samples per configuration. The curves show the average absolute error $|\widehat{\kappa}^\Sigma - \kappa^\Sigma_\infty|$ over the tested angle configurations, together with a reference $N_\Sigma^{-1/2}$ decay.

The results are shown in Fig. 4.5. For all three values of k, the empirical error decreases as N_Σ increases. This is consistent with Prop. 4.9, which theoretically shows convergence of the aggregated kernel towards the explicit expression κ^Σ_∞ as the number of aggregated rank-one terms increases. The three curves also show a similar dependence on N_Σ for the different values of k. This is consistent with the bound of Prop. 4.9, whose convergence rate $O(N_\Sigma^{-1/2})$ does not depend on the subspace dimension k. Testing $k = 1, 2, 3$ therefore allows us to verify empirically that increasing the subspace dimension does not substantially alter the observed convergence towards κ^Σ_∞ in these experiments.

4.7.3 Error analysis

In this experiment, we study the empirical decay of the approximation error for a fixed pair of *orthogonal* subspaces (similarly to the experiment of Sec. 3.5.1). We take $n = 512$ and $k = 9$. We take $\boldsymbol{U} = (\boldsymbol{e}_1, \dots, \boldsymbol{e}_k)$ and $\boldsymbol{V} = (\boldsymbol{e}_{k+1}, \dots, \boldsymbol{e}_{2k})$, with $\boldsymbol{e}_i$ the i^{th} vector of the canonical basis, such that all principal angles between these two subspaces are $\pi/2$.

This choice gives us an explicit target value. For the signed kernel, the random variables defining the two signs are independent and symmetric, hence

$$\kappa^\pm(\boldsymbol{U}, \boldsymbol{V}) = 0.$$

For the periodic kernel, the closed form of κ° gives

$$\kappa^\circ(\boldsymbol{U}, \boldsymbol{V}) = \prod_{j=1}^{k} (1 + \omega^2 \sin^2 \theta_j)^{-1} = (1 + \omega^2)^{-k}.$$

We set $\omega = 1$, so that the target value for κ° is 2^{-k}.

For each value of m, we compute the empirical kernels $\widehat{\kappa}^{\pm}(\boldsymbol{U}, \boldsymbol{V})$ and $\widehat{\kappa}^{\circ}(\boldsymbol{U}, \boldsymbol{V})$ over independent trials. We then measure the absolute errors

$$|\widehat{\kappa}^{\pm}(\boldsymbol{U}, \boldsymbol{V}) - \kappa^{\pm}(\boldsymbol{U}, \boldsymbol{V})| = |\widehat{\kappa}^{\pm}(\boldsymbol{U}, \boldsymbol{V})|$$

and

$$|\widehat{\kappa}^{\circ}(\boldsymbol{U}, \boldsymbol{V}) - \kappa^{\circ}(\boldsymbol{U}, \boldsymbol{V})|.$$

The results are shown in Fig. 4.6 as functions of the oversampling ratio $\rho = m/(nk)$. We use $n = 512$, $k = 9$, and $\omega = 1$. The number of random features m takes 14 logarithmically spaced values between 10 and 10^4, and each value is repeated over 50 independent trials. The solid curves show the mean absolute error over the trials, and the shaded regions correspond to one standard deviation. Both empirical errors follow the reference $\rho^{-1/2}$ decay very closely as expected. We also tested the same thing for $\widehat{\kappa}^{\Sigma}$ with $N_\Sigma = 3$ and $N_\Sigma = 5$. Its error behaves in the same way as the signed kernel $\widehat{\kappa}^{\pm}$, so we do not include it in the figure to keep the plot readable.

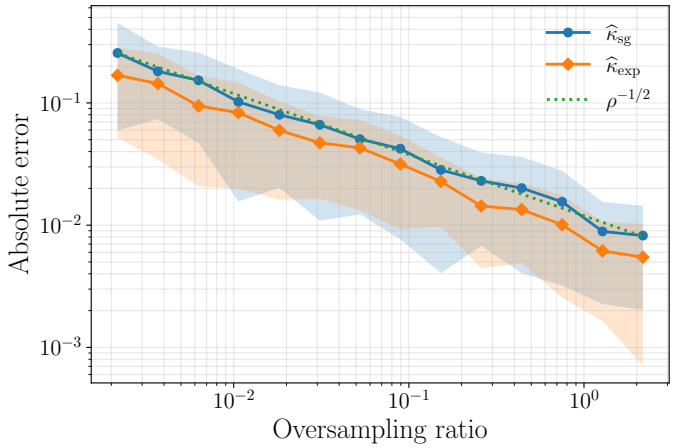


Fig. 4.6 Empirical decay of the approximation error for two orthogonal subspaces in $\mathcal{G}(k, n)$, with $n = 512$ and $k = 9$. The horizontal axis shows the oversampling ratio $\rho = m/(nk)$. The curves show the absolute errors for $\kappa^{\pm}$ and κ°, averaged over independent trials. The dotted line shows the reference $\rho^{-1/2}$ decay.

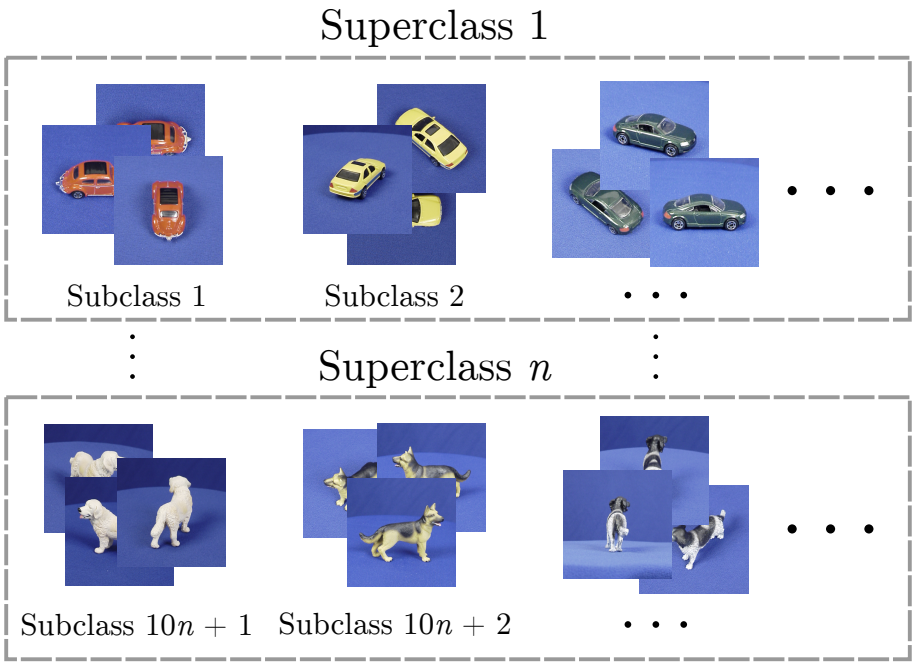


Fig. 4.7 Structure of the ETH-80 dataset

4.7.4 ETH-80 Classification

We evaluate our methods on the ETH-80 dataset [103, 39], which consists of 80 *objects* grouped into 8 *superclasses* (*e.g.*, apple, car, cow), each object being captured from 41 viewpoints under moderate illumination variations. Fig. 4.7 illustrates the hierarchical structure of the dataset, with multiple images per object and multiple objects per superclass. All images are resized to 32×32 pixels, converted to greyscale, so that $n = 32 \times 32 = 1024$.

This experiment is an image-set classification problem, in line with other subspace classification experiments [74, 84, 155]. Each instance to be classified is a *group* of images of the same object acquired under different camera poses or illumination conditions. This group is then represented by the subspace spanned by the corresponding observations.

More precisely, given a collection of N_i images associated with one object, we collect them into a matrix $\boldsymbol{X} \in \mathbb{R}^{n \times N_i}$ and extract an orthonormal basis $\boldsymbol{U} \in \mathbb{R}^{n \times k}$ by retaining the k leading left singular vectors of $\boldsymbol{X}$. Unless otherwise stated, we fix $k = 9$, in line with the empirical observations reported in [84] for a similar dataset. The parameter k is therefore not tuned specifically to optimise the performance of the methods considered here. The resulting task is therefore a classifica-

tion problem between subspace-valued instances. It does not address the separate problem of assigning a single isolated image to a subspace.

We consider two classification settings of increasing difficulty. In the first one, *superclass (8-way) classification*, each object is represented by a single subspace, but labels are given at the superclass level. This setting results in a small number of subspaces and is useful to assess the approximation quality of the proposed approximated kernels. In the second one, *object (80-way) classification*, each object constitutes its own class. To increase the number of available subspaces and stress the computational aspects of kernel evaluation, multiple subspaces are generated per object by subsampling images.

For both settings, we compare the exact projection kernel κ^{p} to its random approximation $\widehat{\kappa}^{\mathrm{rop}}$, evaluate the binary kernel $\widehat{\kappa}^{\pm}$ (for which we do not have a closed-form $\kappa^{\pm}$ available), and compare the periodic kernel $\widehat{\kappa}^{\circ}$ to its exact version κ°.

Recall that $\widehat{\kappa}^{\mathrm{rop}}$ targets the exact projection kernel κ^{p}, whereas the binary and periodic constructions induce the kernels $\kappa^{\pm}$ and κ°, respectively. Therefore, differences in classification accuracy may reflect both the quality of the random approximation and the suitability of the corresponding Grassmannian kernel for the considered dataset. Our goal is not to show that bounded ROP features always outperform unbounded ROP features in finite-data classification, but instead to show that they provide light random features associated with kernels for which stronger uniform approximation guarantees can be computed.

Superclass (8-way) classification

In this first setting, we perform superclass classification on ETH-80, following a similar experiment as [155]. Each object is represented by a single k-dimensional subspace, while labels are assigned at the superclass level. For each superclass, we randomly select 7 objects for training and 3 for testing. This makes a total of 56 training subspaces and 24 test subspaces. Subspaces are constructed as described above, keeping $k = 9$ dimensions. Test subspaces are built independently from the corresponding test objects.

We begin by evaluating the exact projection kernel κ^{p} and the exact periodic kernel κ°, both of which reach perfect classification accuracy on this task. This confirms that the problem is well suited to subspace-based representations and kernel methods.

To better visualise the geometry of this classification problem, Fig. 4.8 shows a two-dimensional t-SNE representation of the 80 subspaces. We first compute the pairwise projection distances as defined

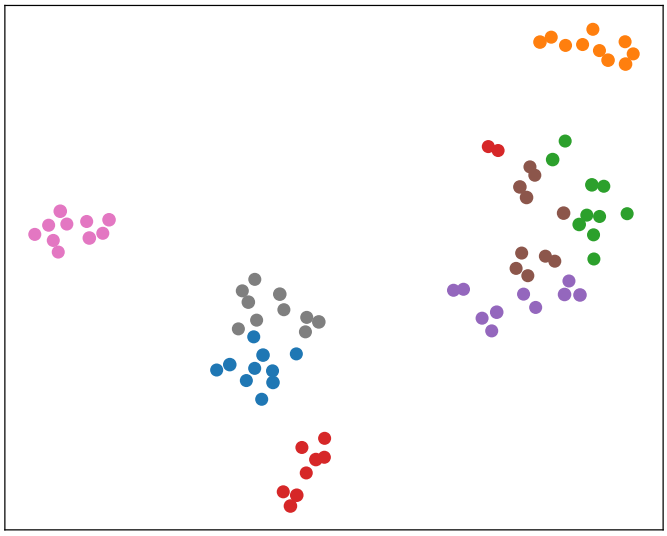


Fig. 4.8 ETH-80 superclass dataset visualisation with t-SNE

in Eq. 2.10:

$$d(\boldsymbol{U},\boldsymbol{V}) = \tfrac{1}{\sqrt{2}}\|\boldsymbol{P}-\boldsymbol{Q}\|_F = \tfrac{1}{\sqrt{2}}\|\boldsymbol{U}\boldsymbol{U}^\top - \boldsymbol{V}\boldsymbol{V}^\top\|_F = \sqrt{k - \|\boldsymbol{U}^\top\boldsymbol{V}\|_F^2}.$$

The resulting representation shows a clear structure, with several classes forming well separated clusters[1]. This provides a qualitative explanation for the good classification results obtained with the exact kernels, while noting that the two-dimensional representation does not preserve all the information available to the classifiers.

We then replace these exact kernels with their approximated versions. To compare embedding sizes across methods, we show the results as a function of the *oversampling ratio* defined as $\rho = \frac{m}{nk}$, where m denotes the embedding dimension. Small values of ρ correspond to compressed representations, $\rho \approx 1$ to embeddings of comparable size to the original basis, and $\rho \gg 1$ to overcomplete embeddings. In the following experiments, we focus on $\rho = 0.05$ and $\rho = 0.20$, which already provide good accuracy results.

Fig. 4.9 shows the classification accuracy as a function of ρ, while Tab. 4.3 summarises runtimes and accuracies for the two selected values of ρ. As expected, increasing ρ improves the accuracy of all random

[1]Interestingly, some neighbouring clusters also correspond to visually similar objects, such as cows and horses or tomatoes and apples. The geometry even seems to have developed opinions about crockery, since the two isolated points of the red coffee-cup class correspond to brown cups, whereas the remaining cups are white. Apparently, the Grassmannian notices these things.

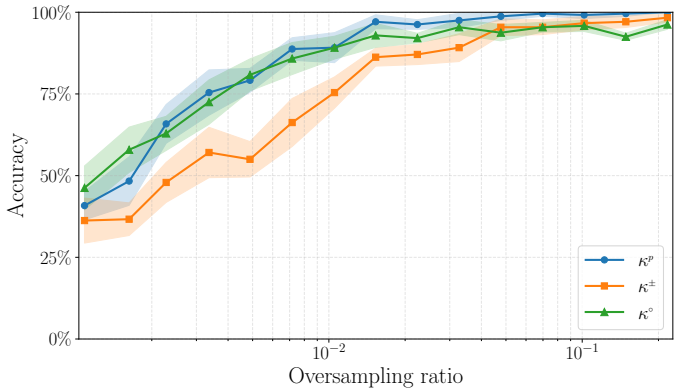


Fig. 4.9 ETH-80 superclass classification results as a function of the oversampling ratio.

Method	ρ	**Total time (s)**	**Accuracy (%)**
Exact kernels (no embedding)			
Projection kernel κ^{P}	—	0.2633	100.00
Periodic kernel κ°	—	0.2381	100.00
Random embeddings at two oversampling ratios			
ROP	0.05 / 0.20	1.3211 / 5.3102	98.12 / 99.79
Binary ROP	0.05 / 0.20	1.3211 / 5.3111	93.12 / 96.04
Periodic ROP	0.05 / 0.20	1.3240 / 5.3145	93.54 / 93.75

Table 4.3 Superclass classification. Total time includes embedding computation and SVM training/evaluation. Results averaged over 20 runs with different random embeddings.

embedding methods. They indeed converge quickly towards the performance of the exact kernels.

From a computational perspective, this setting is too small for random embeddings to fully show their strength. With only 56 training subspaces, exact kernel computation is already fast, and ROP embeddings do not yet accelerate the procedure. This experiment is therefore primarily a validation of the approximation behaviour of the proposed kernels. The computational benefits of random embeddings will become apparent when the number of subspaces increases, as shown in the next experiment.

Object (80-way) classification

We now turn to the object classification task, in which each individual object constitutes its own class. This setting is significantly more complex, both in terms of classification difficulty and computational cost, and is well-suited to highlight the advantages of the approximated kernels.

For each object, the 41 images are randomly split into 28 training images and 13 testing images. From the training images, we randomly select 15 images to construct a k-dimensional subspace, keeping $k = 9$ dimensions. This procedure is repeated 10 times per object, yielding 10 training subspaces in $\mathcal{G}(9, 1024)$ for each of the 80 objects. At test time, a single subspace per object is built from the remaining 13 images, still with $k = 9$. This ensures that no test image contributes in any way to the construction of the training subspaces.

As in the previous setting, we first evaluate the exact projection kernel κ^{P} and the exact periodic kernel κ°. While both kernels achieve strong classification performance, the cost of computing the corresponding kernel matrices now becomes substantial due to the large number of subspaces involved, with computation times close to one minute under our experimental conditions.

We again evaluate the approximated versions of these kernels. Classification accuracy as a function of the oversampling ratio ρ is shown in Fig. 4.10. Runtimes and accuracies at the same oversampling ratios $\rho = 0.05$ and $\rho = 0.20$ as in the superclass setting are summarised in Tab. 4.4.

The accuracy trends are consistent with those observed previously. Increasing ρ improves performance for all random methods, although larger embedding sizes are required in this more complex setting to approach the accuracy of the exact kernels. The embeddings eventually get to the performance of the exact projection kernel, but at the cost of increased computation time, which can exceed that of the deterministic kernels.

4.8 Conclusion

In this chapter we studied random feature maps for data represented by linear subspaces. Starting from the projector representation of the Grassmannian, we first showed that dense random projections can approximate the projection kernel, but at a cost that scales poorly with the ambient dimension. Rank-one projections offer a lighter alternative, however the raw features have poor concentration guarantees, which

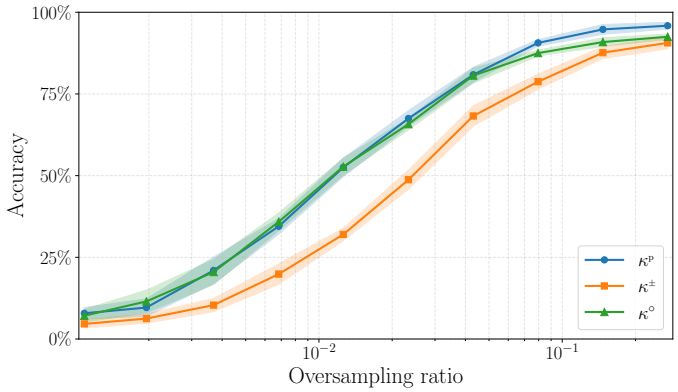


Fig. 4.10 Accuracy vs sub-sampling ratio for object classification with ROP embedding

Method	ρ	Total time (s)	Accuracy (%)
Exact kernels (no embedding)			
Projection kernel κ^{P}	—	52.8856	98.75
Periodic kernel κ°	—	58.7874	90.00
Embedding methods at two oversampling ratios			
ROP	0.05 / 0.20	24.1868 / 85.2087	84.19 / 94.06
Binary ROP	0.05 / 0.20	24.2049 / 85.2449	72.25 / 91.75
Periodic ROP	0.05 / 0.20	24.3763 / 86.4142	81.38 / 92.31

Table 4.4 Object classification. Total time includes embedding computation and SVM training/evaluation. Results averaged over 20 runs with different random embeddings.

motivates the use of bounded non-linear transformations.

The first construction used the sign function to obtain binary random features. This gives compact embeddings and leads to an empirical kernel $\widehat{\kappa}^{\pm}$ that uniformly approximates a deterministic Grassmannian kernel $\kappa^{\pm}$. Although this kernel does not admit a simple closed form in general, we showed that it is well defined, positive semi-definite, invariant under changes of basis, and determined by the principal angles between the subspaces. The case $k = 1$ also connects this construction to the sign product ideas introduced in Chap. 3, now in a symmetric Grassmannian setting.

We then introduced an aggregated binary construction. This change keeps the binary nature of the embedding while producing an expected kernel κ^{Σ} that becomes close to an explicit kernel κ^{Σ}_{∞} as the aggregation

parameter N_Σ increases. This gives a more interpretable binary kernel, with a closed formula depending on the average squared cosine of the principal angles.

Finally, we studied periodic random features, obtained by applying a complex exponential to the rank-one projections. In that case, the expected kernel admits the closed form κ°, and the parameter ω controls how strongly the kernel reacts to changes in the principal angles.

We illustrated all these results with both synthetic and real datasets, confirming the expected results. Although successful, those computation times can still be improved, motivating the next chapter, where we keep the same rank-one projection principle but replace the Gaussian measurements by faster mechanisms.

5 Fast computation of random embeddings

5.1 Introduction

The previous chapters introduced two uses of quadratic random embeddings. In Chapter 3, the debiased embedding ψ^- was used to perform signal estimation directly in the embedding domain, without reconstructing the original signal. In Chapter 4, rank-one projections of subspace projectors were used to build random features for Grassmannian kernel approximation.

This chapter studies how these embeddings can be implemented more efficiently. We consider two complementary approaches based on structured random transformations and optical processing units.

5.1.1 Motivation

In both cases, the mathematical construction relies on applying a certain number of random measurements m. This number is in both cases proportional, up to log factors, to the intrinsic dimension of the objects considered. These embeddings are statistically useful, but are also costly as described in Tab. 4.1.

We can indeed see that all the versions of the embeddings considered have, both for their memory and computational complexity, a dependence on at least n^2, with n the ambient dimension of the vectors considered to build the subspaces. It is therefore desirable to find ways of alleviating these costs, by finding ways to make the creation of those random features lighter both in terms of memory and computational footprint. To this end, we will consider two different approaches.

The first is based on structured embeddings and will allow us to generate close approximations to Gaussian vectors generated by mixing a deterministic Hadamard transform with a random diagonal matrix. This reduces both the computational load, since Hadamard transforms can be performed very efficiently, and the memory footprint, since only the diagonal of random matrices needs to be stored. This will substantially accelerate the embedding procedure while retaining good performance metrics as will be demonstrated experimentally.

The second approach will rely on an optical processing unit (OPU) which is capable of performing symmetric ROPs (ψ^{s}) physically by encoding the vectors to project on a light beam, sending it through a scattering layer and recording the intensity of the output. This makes the computational cost constant regardless of the dimension of the input vector, provided it does not exceed a certain physical limit, since the scattering of light will always happen at the same speed. It also reduces the memory footprint to nothing for the transform since it is stored physically in the scattering layer. It also gives access to a large number of random features instantly.

In both cases, we will explain the theory behind them and develop the necessary tools where appropriate. As we will see, the OPU requires particular attention because its physical constraints require us to quantise and encode the input data before the desired transformations can be computed.

5.1.2 Related work

This chapter continues two existing lines of work. The structured constructions adapt ideas developed for fast random projections and random features, where Gaussian matrices are replaced by compositions of Hadamard or Fourier transforms and random diagonal matrices [4, 5, 96, 163, 65, 23, 43, 147]. Here, these ideas are adapted to the rank-one projections of low-rank matrices used in the previous chapters.

The optical part continues earlier work on quadratic random projections performed by scattering light [133, 28, 56, 132, 152]. Our contribution is to adapt this mechanism to the subspace embeddings considered here by introducing the quantisation, binary encoding and decoding procedures required by the physical constraints of the OPU.

5.1.3 Related publications

This chapter is related to the following publications, with contributions from Vincent Schellekens, Laurent Daudet, Laurent Jacques, and the author of the present document.

- R. Delogne, V. Schellekens, L. Daudet, and L. Jacques. "Signal processing with optical quadratic random sketches". In: *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2023, pp. 1–5, [56]
- R. Delogne and L. Jacques. "Random features for Grassmannian kernel approximation with bounded rank-one projections". In: *Under review for Transactions on Machine Learning Research (TMLR)* (2026). URL: `openreview.net/forum?id=wq18dZJ2pA`, [52]

5.2 Structured transforms for fast embeddings

A specific line of work has shown that random projections of vectors performed by dense random matrices can be replaced by compositions of *fast orthogonal transforms* and *random diagonal sign matrices*, creating vectors that behave similarly to i.i.d. Gaussian vectors at a lower computational cost. Examples include the Fast Johnson–Lindenstrauss Transform [4], the Fastfood construction for random features [96], the BCH–code based JL embeddings [5], Orthogonal Random Features [163], signed circulant matrix random mapping [65], or Hadamard blocks for compressive covariance sketching [147]. While differing in detail, these methods share the common leitmotif of using a small number of random diagonal matrices mixed with fast transforms (Hadamard or Fourier), sometimes organised in blocks to generate large numbers of structured random vectors. In particular, the use of a small number of successive Hadamard–diagonal compositions has been studied explicitly in the context of structured random projections, with theoretical and empirical evidence that as few as three such blocks already yield distributions close to Gaussian [23, 43].

We decide to adapt this philosophy to rank-one projections of low-rank matrices (obtained from low-rank subspace projectors) by adopting the method proposed in [43, 163, 147]. Mathematically, given a subspace basis $\boldsymbol{U} \in \mathbb{V}(k,n)$ (with projector $\boldsymbol{P} = \boldsymbol{U}\boldsymbol{U}^\top$) made of k orthonormal vectors $\boldsymbol{U} = (\boldsymbol{u}_1, \dots, \boldsymbol{u}_k)$, defining $T = \lceil m/n \rceil$, we pick the m probing vectors $\{\boldsymbol{a}_j\}_{j=1}^m$ and $\{\boldsymbol{b}_j\}_{j=1}^m$ involved in $\psi^{\text{rop}}(\boldsymbol{U})$ as the m first columns of two $n \times Tn$ matrices $\overline{\boldsymbol{G}} := [\boldsymbol{G}_1, \dots, \boldsymbol{G}_T]$ and $\overline{\boldsymbol{G}}' := [\boldsymbol{G}'_1, \dots, \boldsymbol{G}'_T]$, respectively. Each $n \times n$ matrix $\boldsymbol{G}_t, \boldsymbol{G}'_t$, $1 \le t \le T$, is i.i.d. as the random matrix $\boldsymbol{G}$ whose *distribution* is defined by the following S-factor generative model:

$$\boldsymbol{G} = \sqrt{n} \prod_{j=1}^{S} \big(\operatorname{diag}(\boldsymbol{\varepsilon}_j) \boldsymbol{H} \big), \quad \text{with } \boldsymbol{\varepsilon}_1, \dots, \boldsymbol{\varepsilon}_S \sim_{\text{i.i.d.}} \boldsymbol{\varepsilon}, \tag{5.1}$$

with the normalised Walsh-Hadamard matrix $\boldsymbol{H} \in \{\pm 1/\sqrt{n}\}^{n \times n} \cap$

$\mathsf{O}(n)$, and $\boldsymbol{\varepsilon}$ an n-length Rademacher random vector with entries i.i.d. as $\mathcal{U}(\pm 1)$. The normalisation $\sqrt{n}$ in Eq. 5.1 ensures that the columns of $\boldsymbol{G}$ have squared norm n, which is also the expected squared norm of a Gaussian random vector in $\mathbb{R}^n$.

For small values of S (typically $S = 3$), picking m columns of $\overline{\boldsymbol{G}}$ (or $\overline{\boldsymbol{G}}'$) is known to generate an $n \times m$ matrix $\boldsymbol{W}^\top$ with similar properties to an $n \times m$ Gaussian random matrix, *e.g.*, $\boldsymbol{W}$ satisfies the Johnson-Lindenstrauss lemma *w.h.p.* [5].

Thanks to this specific design of the ROP probing vectors, to project the subspace $\boldsymbol{U}$, we first compute, for $1 \leq i \leq k$,

$$(\boldsymbol{a}_j^\top \boldsymbol{u}_i)_{j=1}^m = \boldsymbol{S}(\overline{\boldsymbol{G}}^\top \boldsymbol{u}_i), \quad (\boldsymbol{b}_j^\top \boldsymbol{u}_i)_{j=1}^m = \boldsymbol{S}(\overline{\boldsymbol{G}}'^\top \boldsymbol{u}_i). \tag{5.2}$$

with $\boldsymbol{S} \in \{0,1\}^{m \times Tn}$ the selection matrix extracting the first m components of any vector in $\mathbb{R}^{Tn}$. From the structure of $\overline{\boldsymbol{G}}$ and $\overline{\boldsymbol{G}}'$ and the use of the fast Walsh-Hadamard transform[1], computing $\overline{\boldsymbol{G}}^\top \boldsymbol{u}_i$ and $\overline{\boldsymbol{G}}'^\top \boldsymbol{u}_i$ for all $1 \leq i \leq k$ in Eq. 5.2 requires $\mathcal{O}(SkTn \log n) = \mathcal{O}(Skm \log n)$ operations, and storing $\mathcal{O}(SkTn) = \mathcal{O}(Skm)$ Rademacher components (from Eq. 5.1) as well as the storage of the $\mathcal{O}(km)$ intermediate values computed in Eq. 5.2. The final multiplication by $\boldsymbol{S}$ has negligible complexity.

Next, we form the fast, structured ROP operator

$$\psi_{\mathrm{st}}(\boldsymbol{U}) = \Big(\sum_{i=1}^k \boldsymbol{a}_j^\top \boldsymbol{u}_i \boldsymbol{u}_i^\top \boldsymbol{b}_j \Big)_{j=1}^m = \Big(\boldsymbol{a}_j^\top \boldsymbol{U}\boldsymbol{U}^\top \boldsymbol{b}_j \Big)_{j=1}^m,$$

which needs an extra $\mathcal{O}(km)$ operations.

The whole computation of $\psi_{\mathrm{st}}(\boldsymbol{U})$ requires a total storage of $\mathcal{O}(Skm)$ values, and a total computational complexity of $\mathcal{O}(Skm \log n)$ operations. In practice, we find that using more than $S = 3$ provides no significant gain. This empirical observation is consistent with the findings in [163, Equation 5] or [147, Section 3.1], where triple Hadamard blocks are shown to be sufficient for embedding covariance matrices.

Therefore, considering that $S = \mathcal{O}(1)$, and assuming that the results of Props. 4.4 and 4.12 still hold when $\psi^\pm$ and ψ° rely on ψ^{rop}, if we take $m = \mathcal{O}(\delta^{-2} kn)$ random features (up to log factors) to ensure that the approximation error of the target kernels is uniformly bounded by $\delta > 0$, we can state that the computation of both types of random features, binary or periodic, requires a total storage of $\mathcal{O}(\delta^{-2} k^2 n)$ values, and a total computational complexity of $\mathcal{O}(\delta^{-2} k^2 n \log n)$ operations.

[1] We used the implementation developed for [7] and available at `https://github.com/FALCONN-LIB/FFHT`.

Random feature	# of features m	Error bound	Target kernel
Dense Ψ	$\mathcal{O}(\delta^{-2}k^3n)$	Prop. 4.1	Proj. kernel κ^{P}
Binary $\psi^{\pm}$	$\mathcal{O}(\delta^{-2}kn)$	Prop. 4.4	New kernel $\kappa^{\pm}$
Periodic ψ°	$\mathcal{O}(\delta^{-2}kn)$	Prop. 4.12	New kernel κ°
Structured ($\supset \psi_{\mathrm{st}}$)	$\mathcal{O}(\delta^{-2}kn)$ assumed	assumed	Various $\kappa^{\pm}$, κ°

Table 5.1 Number of random features required for k-dimensional subspaces of $\mathbb{R}^n$ to obtain a uniform approximation error $\delta > 0$ between the empirical kernel and the corresponding target kernel.

Random feature	Memory	Computation
Dense Ψ	$\mathcal{O}(\delta^{-2}k^3n^3)$	$\mathcal{O}(\delta^{-2}k^3n^3)$
Binary $\psi^{\pm}$	$\mathcal{O}(\delta^{-2}(kn^2))$	$\mathcal{O}(\delta^{-2}k^2n^2)$
Periodic ψ°	$\mathcal{O}(\delta^{-2}(kn^2))$	$\mathcal{O}(\delta^{-2}k^2n^2)$
Structured ($\supset \psi_{\mathrm{st}}$)	$\mathcal{O}(\delta^{-2}k^2n)$	$\mathcal{O}(\delta^{-2}k^2n\log n)$

Table 5.2 Memory and computational complexities per instance for k-dimensional subspaces of $\mathbb{R}^n$ after choosing the number of features needed to obtain a uniform approximation error $\delta > 0$.

We summarise the computational and memory costs of the various random feature maps introduced so far in Tab. 5.1 and Tab. 5.2. Practical applications and statistical analysis of this particular type of embedding can be found in Sec. 5.5.

5.3 Optical processing units for random embeddings

The second route we consider in this chapter relies on *optical computing*. While the structured embeddings of Section 5.2 reduce the computational cost of generating and applying random embeddings, Optical Processing Units offer a different approach by completely stripping off the computational aspect of the embedding. Instead of explicitly storing or applying a random matrix, the embedding is implemented *physically* through the propagation of light in a scattering medium. In this section, we explain how this optical mechanism can be interpreted as a quadratic random embedding, how it relates to the embeddings used in chapters 3 and 4, and how we can apply this process to data after a special encoding procedure to satisfy the OPU requirements.

5.3.1 Optical random projections

Using the OPU in order to apply an embedding will offer two main advantages. First of all, the vectors required to apply the embeddings no longer need to be stored since they are *physically* contained inside the OPU and remain stable over time as the OPU is used and reused (up to some noise). Furthermore, as we will see shortly, since the OPU process uses light to encode and process data, the processing step is done *at the speed of light*, and independently of the ambient dimension of the object to embed [132, 56, 28, 152].

A similar advantage exists with respect to the number m of random projections returned for each input. Unlike a digital implementation, where computing more projections requires more operations, the OPU generates a large number of output coordinates in parallel and can return up to millions of random projections for a single input [28]. Hence, within the hardware limits of the device, increasing m does not require an increase in processing time, and reducing m does not directly allow different inputs to be processed at a higher rate. In our setting, the rate at which distinct inputs can be processed is instead more directly affected by the number of separate OPU calls required to encode one input as we will describe later.

Practically, an OPU processes a *binary* input vector $\boldsymbol{x} \in \{0,1\}^n$, with $n \leq \mathcal{O}(10^6)$ by using it to program a digital micro-mirror device (DMD). In this DMD, only correctly oriented "on" (1) mirrors (as opposed to mis-aligned "off" (0) mirrors) can reflect part of an incident laser beam to a stable scattering medium. After scattering, the intensity of a complex pattern, or speckle, arising from constructive and destructive interferences of the scattered light, is recorded by a camera. Since the camera records only intensities, the resulting digital output is quadratic in the input. In an idealised model, this produces measurements of the form

$$\psi_{\text{opu}}(\boldsymbol{x}) = \left(|\boldsymbol{r}_i^* \boldsymbol{x}|^2\right)_{i=1}^m,$$

where $\boldsymbol{r}_i = \boldsymbol{r}_{i,R} + \mathrm{i}\boldsymbol{r}_{i,I} \in \mathbb{C}^n$, with $\boldsymbol{r}_{i,R}, \boldsymbol{r}_{i,I} \sim_{\text{i.i.d.}} \mathcal{N}(\boldsymbol{0}, \frac{1}{2}\mathbf{I}_n)$ independent. Equivalently, we write $\boldsymbol{r}_i \sim_{\text{i.i.d.}} \mathcal{CN}(\boldsymbol{0}, \mathbf{I}_n)$ with this convention. Thus the OPU naturally implements a complex equivalent process of the symmetric quadratic embedding ψ^{s}. This process is illustrated in Fig. 5.1.

This physical implementation has two important consequences. First of all, in terms of efficiency, the random vectors for the embedding no longer need to be generated, stored, or applied digitally. They are fixed by the scattering medium and accessed through the optical propagation itself, at the speed of light. This is precisely why OPUs are attrac-

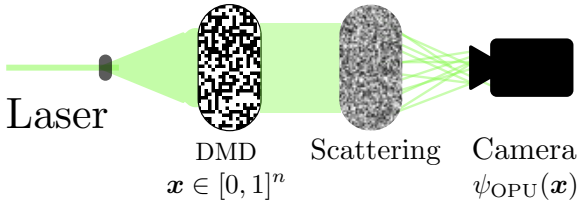


Fig. 5.1 Encoding, scattering and recording process of a binary vector $\boldsymbol{x}$ in the OPU.

tive for random projection methods. The second consequence is more of a constraint. The OPU does not accept arbitrary real-valued vectors as inputs but only binary ones. Since the input is encoded through the DMD as we just described, OPU inputs have to be vectors in $\{0,1\}^n$. This binary input constraint is not a small implementation detail since all objects appearing in the previous chapters are real-valued. Therefore, the OPU cannot be applied directly to the objects appearing in earlier sections of this work. To this end we will develop, later in this section, a method to efficiently decompose *quantised vectors* into binary vectors such that those binary vectors may be sent through the OPU and their outputs recombined in a form equivalent to what would have been the output if the OPU had been capable of processing quantised vectors. The number of vectors to be sent to the OPU will depend directly on the number of quantisation levels L.

In this section we will thus first explain how the natural OPU transformation can be processed such that it replicates ψ^{-} and ψ^{rop} studied previously. We will then describe how a vector quantised to L levels can be decomposed into binary vectors, with those binary vectors sent through the OPU, and their outputs recombined such that the final result is the exact embedding of the quantised vector.

5.3.2 Quadratic embeddings with an OPU

As we just mentioned, we model the OPU as follows. Given a binary input vector $\boldsymbol{x} \in \{0,1\}^n$, the device returns an m-dimensional vector

$$\psi_{\mathrm{opu}}(\boldsymbol{x}) = \left(|\boldsymbol{r}_i^* \boldsymbol{x}|^2\right)_{i=1}^m,$$

where $\boldsymbol{r}_i \in \mathbb{C}^n$ are independent complex Gaussian vectors and where $\boldsymbol{r}_i^*$ denotes the transpose complex conjugate of $\boldsymbol{r}_i$. Equivalently, each coordinate can be written as

$$|\boldsymbol{r}_i^* \boldsymbol{x}|^2 = \boldsymbol{x}^\top \boldsymbol{r}_i \boldsymbol{r}_i^* \boldsymbol{x} = \langle \boldsymbol{x}\boldsymbol{x}^\top, \boldsymbol{r}_i \boldsymbol{r}_i^* \rangle_F.$$

Thus the OPU performs complex quadratic measurements of the lifted matrix $\boldsymbol{x}\boldsymbol{x}^\top$. This is the same lifted viewpoint developed originally in

Chap. 3, meaning that the embedding is non-linear in $\boldsymbol{x}$, but linear in the rank-one matrix $\boldsymbol{x}\boldsymbol{x}^\top$.

Let us now see how this can be used to mimic ψ^-. In Chap. 3, the basic symmetric embedding was

$$\psi^{\mathrm{s}}(\boldsymbol{x}) = \big((\boldsymbol{a}_i^\top \boldsymbol{x})^2\big)_{i=1}^m,$$

with $(\boldsymbol{a}_i)_{i=1}^m$ random i.i.d. vectors in $\mathbb{R}^n$ and $\boldsymbol{a}_i \sim_{\text{i.i.d.}} \mathcal{N}(\boldsymbol{0}, \mathbf{I}_n)$. The debiased embedding was defined as

$$\psi^-(\boldsymbol{x}) = \big((\boldsymbol{a}_i^\top \boldsymbol{x})^2 - (\boldsymbol{b}_i^\top \boldsymbol{x})^2\big)_{i=1}^m,$$

with $\boldsymbol{a}_i, \boldsymbol{b}_i$ random i.i.d. vectors in $\mathbb{R}^n$ with $\boldsymbol{a}_i, \boldsymbol{b}_i \sim_{\text{i.i.d.}} \mathcal{N}(\boldsymbol{0}, \mathbf{I}_n)$.

Since the OPU gives the complex equivalent of ψ^{s}, by taking twice as many measurements ($2m$), splitting the output vector in two and differencing the two half vectors, we can obtain

$$\psi^-_{\text{opu}}(\boldsymbol{x}) = \big(|\boldsymbol{r}_{i,1}^* \boldsymbol{x}|^2 - |\boldsymbol{r}_{i,2}^* \boldsymbol{x}|^2\big)_{i=1}^m,$$

where we set $\boldsymbol{r}_{i,1} := \boldsymbol{r}_i$ and $\boldsymbol{r}_{i,2} := \boldsymbol{r}_{m+i}$ for $i = 1, \dots, m$, with $(\boldsymbol{r}_\ell)_{\ell=1}^{2m}$ i.i.d. complex Gaussian vectors in $\mathbb{C}^n$, with $\boldsymbol{r}_\ell \sim_{\text{i.i.d.}} \mathcal{CN}(\boldsymbol{0}, \mathbf{I}_n)$. Up to the use of complex Gaussian vectors, it is the same debiasing mechanism used for ψ^- in Chapter 3.

Writing $\boldsymbol{r}_{i,\ell} = \boldsymbol{g}_{i,\ell} + \mathrm{i}\boldsymbol{h}_{i,\ell}$, with $\boldsymbol{g}_{i,\ell}, \boldsymbol{h}_{i,\ell} \sim_{\text{i.i.d.}} \mathcal{N}(\boldsymbol{0}, \frac{1}{2}\mathbf{I}_n)$, we can set

$$\boldsymbol{a}_i = \sqrt{2}\boldsymbol{g}_{i,1}, \qquad \boldsymbol{b}_i = \sqrt{2}\boldsymbol{g}_{i,2}, \qquad \boldsymbol{c}_i = \sqrt{2}\boldsymbol{h}_{i,1}, \qquad \boldsymbol{d}_i = \sqrt{2}\boldsymbol{h}_{i,2}.$$

Then $\boldsymbol{a}_i, \boldsymbol{b}_i, \boldsymbol{c}_i, \boldsymbol{d}_i \sim_{\text{i.i.d.}} \mathcal{N}(\boldsymbol{0}, \mathbf{I}_n)$ and

$$\psi^-_{\text{opu}}(\boldsymbol{x})_i = \tfrac{1}{2}\big((\boldsymbol{a}_i^\top \boldsymbol{x})^2 - (\boldsymbol{b}_i^\top \boldsymbol{x})^2\big) + \tfrac{1}{2}\big((\boldsymbol{c}_i^\top \boldsymbol{x})^2 - (\boldsymbol{d}_i^\top \boldsymbol{x})^2\big).$$

Thus each coordinate of $\psi^-_{\text{opu}}(\boldsymbol{x})$ is in fact the average of two independent $\psi^-(\boldsymbol{x})$ coordinates. It therefore does not exactly have the same distribution as $\psi^-(\boldsymbol{x})$, but it is an averaged version of it, with the same expectation and a smaller variance, similar to the idea of the aggregated embedding from Sec. 4.4.

Next, if we look at how the OPU relates to ψ^{rop} from chapter 4, the goal is now to reproduce the asymmetric measurements used for subspace embeddings[2]. Recall that given $\boldsymbol{U} \in \mathbb{V}(k, n)$, we write

$$\boldsymbol{P} = \boldsymbol{U}\boldsymbol{U}^\top = \textstyle\sum_{j=1}^k \boldsymbol{u}_j \boldsymbol{u}_j^\top.$$

[2]This possibility was already hinted at in Rem. 3.2 where we explained that ψ^- was equivalent to ψ^{rop} up to a rescaling

The i-th coordinate of $\psi^{\mathrm{rop}}(\boldsymbol{U})$ is therefore

$$\psi^{\mathrm{rop}}(\boldsymbol{U})_i = \boldsymbol{a}_i^\top \boldsymbol{P}\boldsymbol{b}_i = \sum_{j=1}^k \boldsymbol{a}_i^\top \boldsymbol{u}_j \boldsymbol{u}_j^\top \boldsymbol{b}_i,$$

with $\boldsymbol{a}_i, \boldsymbol{b}_i$ random i.i.d. vectors in $\mathbb{R}^n$ with $\boldsymbol{a}_i, \boldsymbol{b}_i \sim_{\text{i.i.d.}} \mathcal{N}(\boldsymbol{0}, \mathbf{I}_n)$.

The previous discussion shows that a differenced OPU measurement applied to a real vector $\boldsymbol{x}$ gives, up to a factor $1/2$, the average of two independent asymmetric rank-one projections of $\boldsymbol{x}\boldsymbol{x}^\top$. Applying this to each column $\boldsymbol{u}_j$ and summing over j gives

$$\tfrac{1}{2}\sum_{j=1}^k \psi_{\mathrm{opu}}^-(\boldsymbol{u}_j)_i = \tfrac{1}{2}\boldsymbol{a}_i^\top \boldsymbol{P}\boldsymbol{b}_i + \tfrac{1}{2}\boldsymbol{c}_i^\top \boldsymbol{P}\boldsymbol{d}_i,$$

where $\boldsymbol{a}_i, \boldsymbol{b}_i, \boldsymbol{c}_i, \boldsymbol{d}_i \sim_{\text{i.i.d.}} \mathcal{N}(\boldsymbol{0}, \mathbf{I}_n)$. Thus the OPU construction does not recover one specific coordinate of $\psi^{\mathrm{rop}}(\boldsymbol{U})$, but the average of two independent coordinates with the same distribution. This remains a valid random feature construction since, after a fixed rescaling, its expected scalar product is the same as for a single rank-one projection, while the averaging reduces its variance. After applying the sign function, it also matches the aggregated embedding of Sec. 4.4 with $N_\Sigma = 2$, up to the fixed scaling factor.

The resulting quantities can then be composed with the same bounded non-linear functions as in Chapter 4, for instance the sign function or the complex exponential. This gives an OPU-compatible way to construct the Grassmannian random features introduced earlier. The remaining obstruction is not the form of the embedding anymore, but the binary input constraint: the vectors $\boldsymbol{x}$, $\boldsymbol{u}$, and $\boldsymbol{u}_j$ are generally real-valued and therefore cannot be sent directly to the OPU. The next subsection addresses this issue through quantisation and binary encoding.

5.3.3 Encoding quantised vectors for the OPU

In this section we focus on the mechanism that allows us to evaluate quadratic embeddings of quantised vectors using the OPU. Since the OPU accepts only binary inputs $\boldsymbol{x} \in \{0,1\}^n$, in order to increase precision beyond the 1-bit regime, we will consider a uniform quantiser with L levels applied to a real vector $\boldsymbol{x} \in \mathbb{R}^n$ and a procedure to split this into binary vectors. The procedure is exact in the ideal noiseless OPU model, provided that all calls use the same fixed optical transformation. Long-term measurements of speckle patterns indicate that this transformation can remain stable over extended periods, as reported for instance in the supplementary material of [152]. In practice, however, the measurements are affected by experimental noise and hardware constraints, which must be accounted for when implementing and evaluating the procedure.

In what follows we call a vector OPU-compatible if it belongs to $\{0,1\}^n$, since it can be directly fed into the OPU to produce quadratic measurements. We first recall Def. 2.5.1 of the uniform quantiser.

Definition 5.3.1 (Uniform quantisation)**.** *Let $\delta > 0$ denote the quantisation step and let $L \in \mathbb{N}$ be an even number of levels. We define the uniform quantiser as*

$$q(t) = \delta(\lfloor t/\delta \rfloor + 1/2),$$

followed by clipping to the interval

$$[-(L/2 - 1/2)\delta, (L/2 - 1/2)\delta].$$

Applied componentwise, this creates a quantised vector $\boldsymbol{q} = q(\boldsymbol{x}) \in \mathbb{R}^n$ whose entries take L values equally spaced around zero. The possible values are gathered into the alphabet

$$\{-\delta(L-1)/2, \ldots, -\delta/2, \delta/2, \ldots, \delta(L-1)/2\}. \tag{5.3}$$

Since the OPU only accepts binary vectors, the goal is to evaluate quadratic measurements of the form $|\boldsymbol{r}^*\boldsymbol{q}|^2$ using only OPU-compatible inputs, while keeping the same hidden vector $\boldsymbol{r}$ across all calls involved in the reconstruction.

We now explain how $\boldsymbol{q}$ can be decomposed into signed bitplanes. Let

$$B = \lceil \log_2 L \rceil$$

denote the number of bitplanes. Since the quantiser is uniform with step δ and has an even number L of levels, each rescaled magnitude $2|q_\ell|/\delta$ belongs to $\{1, 3, \ldots, L-1\}$. This rescaling removes the quantisation step and turns each magnitude into an odd integer. We encode this integer in binary and attach the sign of q_ℓ to each active bit. This gives the decomposition

$$\boldsymbol{q} = \textstyle\sum_{i=1}^{B} G_i \boldsymbol{q}_i,$$

where $\boldsymbol{q}_i \in \{-1, 0, 1\}^n$ and

$$G_i = \delta 2^{i-2}.$$

The bitplanes are technically not arbitrary ternary vectors since for each coordinate, all non-zero bitplane entries have the sign of the corresponding entry of $\boldsymbol{q}$. Two bitplanes can therefore never contain opposite signs at the same coordinate. Consequently, for all i, j,

$$\boldsymbol{q}_i - \boldsymbol{q}_j \in \{-1, 0, 1\}^n.$$

The following proposition shows that this structure is enough to reconstruct the OPU embedding of $\boldsymbol{q}$ exactly from binary measurements.

Proposition 5.1 (Binary encoding of quantised OPU measurements)**.** *A vector $\boldsymbol{q} \in \mathbb{R}^n$ whose entries are quantised using Def. 5.3.1 with $L = 2^B$ levels can be written as a weighted sum of $\mathcal{O}(\log L)$ vectors in $\{-1, 0, 1\}^n$. Moreover, the embedding $\psi_{\text{opu}}(\boldsymbol{q})$ can be reconstructed from quadratic measurements of binary vectors using*

$$N_{\text{OPU}}(L) = 3B + 3\binom{B}{2} = \tfrac{3B(B+1)}{2},$$

where $B = \lceil \log_2 L \rceil$. In particular, the number of required OPU calls scales as $\mathcal{O}((\log_2 L)^2)$.

Proof. Let $B = \lceil \log_2 L \rceil$. As described above, the quantised vector can be written as

$$\boldsymbol{q} = \textstyle\sum_{i=1}^{B} G_i \boldsymbol{q}_i, \tag{5.4}$$

where $\boldsymbol{q}_i \in \{-1, 0, 1\}^n$ are signed bitplanes and $G_i = \delta 2^{i-2}$. Expanding the outer product gives

$$\boldsymbol{q}\boldsymbol{q}^\top = \underbrace{\sum_{i=1}^{B} G_i^2 \boldsymbol{q}_i \boldsymbol{q}_i^\top}_{\text{Symmetric terms}} + \underbrace{\sum_{1 \le i < j \le B} G_i G_j (\boldsymbol{q}_i \boldsymbol{q}_j^\top + \boldsymbol{q}_j \boldsymbol{q}_i^\top)}_{\text{Cross-terms}}. \tag{5.5}$$

It is therefore enough to show how each symmetric term $\boldsymbol{q}_i \boldsymbol{q}_i^\top$ and each cross-term $\boldsymbol{q}_i \boldsymbol{q}_j^\top + \boldsymbol{q}_j \boldsymbol{q}_i^\top$ can be reconstructed from OPU-compatible inputs.

Symmetric terms. Since the signed bitplanes are ternary vectors (taking values in $\{-1, 0, 1\}$), we first describe how to evaluate them by splitting them into binary vectors. Let $\boldsymbol{s} \in \{-1, 0, 1\}^n$. We write

$$\boldsymbol{s} = \boldsymbol{s}^+ - \boldsymbol{s}^-, \qquad \boldsymbol{s}^+ = \mathbb{1}\{\boldsymbol{s} = 1\}, \qquad \boldsymbol{s}^- = \mathbb{1}\{\boldsymbol{s} = -1\},$$

where $\mathbb{1}$ is applied componentwise, and define

$$\boldsymbol{r}_s = \boldsymbol{s}^+ + \boldsymbol{s}^-.$$

The vectors $\boldsymbol{s}^+$, $\boldsymbol{s}^-$ and $\boldsymbol{r}_s$ are binary, hence OPU-compatible. Since $\boldsymbol{s}^+$ and $\boldsymbol{s}^-$ have disjoint supports, expanding the outer products gives

$$\boldsymbol{s}\boldsymbol{s}^\top = \boldsymbol{s}^+\boldsymbol{s}^{+\top} + \boldsymbol{s}^-\boldsymbol{s}^{-\top} - \boldsymbol{s}^+\boldsymbol{s}^{-\top} - \boldsymbol{s}^-\boldsymbol{s}^{+\top}$$

and

$$\boldsymbol{r}_s\boldsymbol{r}_s^\top = \boldsymbol{s}^+\boldsymbol{s}^{+\top} + \boldsymbol{s}^-\boldsymbol{s}^{-\top} + \boldsymbol{s}^+\boldsymbol{s}^{-\top} + \boldsymbol{s}^-\boldsymbol{s}^{+\top}.$$

Combining these two identities yields

$$\boldsymbol{s}\boldsymbol{s}^\top = 2\boldsymbol{s}^+\boldsymbol{s}^{+\top} + 2\boldsymbol{s}^-\boldsymbol{s}^{-\top} - \boldsymbol{r}_s\boldsymbol{r}_s^\top. \tag{5.6}$$

Thus the quadratic measurements of a ternary vector $\boldsymbol{s}$ can be reconstructed from three OPU calls applied to $\boldsymbol{s}^+$, $\boldsymbol{s}^-$ and $\boldsymbol{r}_s$.

Applying this identity to every $\boldsymbol{q}_i$, we obtain the diagonal terms

$$\psi_{\text{opu}}(\boldsymbol{q}_i) = 2\psi_{\text{opu}}(\boldsymbol{q}_i^+) + 2\psi_{\text{opu}}(\boldsymbol{q}_i^-) - \psi_{\text{opu}}(\boldsymbol{r}_i), \tag{5.7}$$

where

$$\boldsymbol{q}_i^+ = \mathbb{1}\{\boldsymbol{q}_i = 1\}, \qquad \boldsymbol{q}_i^- = \mathbb{1}\{\boldsymbol{q}_i = -1\}, \qquad \boldsymbol{r}_i = \boldsymbol{q}_i^+ + \boldsymbol{q}_i^-.$$

This reconstructs the OPU measurements associated with the $\boldsymbol{q}_i\boldsymbol{q}_i^\top$ term in Eq. 5.5 using three binary inputs.

Cross-terms. It remains to handle the cross-terms. For $1 \leq i < j \leq B$, define

$$\boldsymbol{d}_{ij} := \boldsymbol{q}_i - \boldsymbol{q}_j.$$

By the decomposition of $\boldsymbol{q}$ in ternary bitplanes in Eq. 5.4 we know that the same entry of all vectors $\boldsymbol{q}_i$ will have the same sign as the corresponding entry in $\boldsymbol{q}$, hence we know that $\boldsymbol{d}_{ij} \in \{-1, 0, 1\}^n$. Therefore we can apply the same ternary decomposition as Eq. 5.7 by defining

$$\boldsymbol{d}_{ij}^+ = \mathbb{1}\{\boldsymbol{d}_{ij} = 1\}, \qquad \boldsymbol{d}_{ij}^- = \mathbb{1}\{\boldsymbol{d}_{ij} = -1\}, \qquad \boldsymbol{r}_{ij} = \boldsymbol{d}_{ij}^+ + \boldsymbol{d}_{ij}^-,$$

to obtain

$$\psi_{\text{opu}}(\boldsymbol{q}_i - \boldsymbol{q}_j) = \psi_{\text{opu}}(\boldsymbol{d}_{ij}) = 2\psi_{\text{opu}}(\boldsymbol{d}_{ij}^+) + 2\psi_{\text{opu}}(\boldsymbol{d}_{ij}^-) - \psi_{\text{opu}}(\boldsymbol{r}_{ij}).$$

Notice that in general,

$$\boldsymbol{q}_i\boldsymbol{q}_j^\top + \boldsymbol{q}_j\boldsymbol{q}_i^\top = \boldsymbol{q}_i\boldsymbol{q}_i^\top + \boldsymbol{q}_j\boldsymbol{q}_j^\top - (\boldsymbol{q}_i - \boldsymbol{q}_j)(\boldsymbol{q}_i - \boldsymbol{q}_j)^\top.$$

Therefore we can have access to the transformation of the cross-terms with

$$\boldsymbol{c}_{ij} := \psi_{\text{opu}}(\boldsymbol{q}_i) + \psi_{\text{opu}}(\boldsymbol{q}_j) - \psi_{\text{opu}}(\boldsymbol{d}_{ij}). \tag{5.8}$$

Recombining. Combining the diagonal and cross-term contributions gives

$$\psi_{\text{opu}}(\boldsymbol{q}) = \textstyle\sum_{i=1}^B G_i^2 \psi_{\text{opu}}(\boldsymbol{q}_i) + \sum_{1 \leq i < j \leq B} G_i G_j \boldsymbol{c}_{ij}. \tag{5.9}$$

The full decomposition procedure is described in Algo. 1 and the corresponding recombination procedure is given in Algo. 2.

These procedures are exact and under the assumption that ψ_{opu} remains perfectly constant from one use to another, the recombined vector given in Eq. 5.9 gives exactly what $\psi_{\text{opu}}(\boldsymbol{q})$ would give if the OPU tolerated non-binarised vectors.

Finally, each of the B diagonal terms requires three OPU calls, and each of the $\binom{B}{2}$ cross-terms requires three additional OPU calls through the ternary difference $\boldsymbol{q}_i - \boldsymbol{q}_j$. Therefore

$$N_{\text{OPU}}(L) = 3B + 3\binom{B}{2} = \tfrac{3B(B+1)}{2}.$$

This proves the claim. □

Algorithm 1: OPU Encoder

Input: $\boldsymbol{q} \in \mathbb{R}^n$ quantised with L levels, step δ
Output: Ordered set $\mathcal{S}$ of binary vectors in $\{0,1\}^n$
$B \leftarrow \lceil \log_2 L \rceil$;
$\mathcal{S} \leftarrow \emptyset$;
for $i = 1$ **to** B **do**
  Compute bitplane $\boldsymbol{q}_i \in \{-1, 0, 1\}^n$;
  $\boldsymbol{q}_i^+ \leftarrow \mathbb{1}\{\boldsymbol{q}_i = 1\}$;
  $\boldsymbol{q}_i^- \leftarrow \mathbb{1}\{\boldsymbol{q}_i = -1\}$;
  $\boldsymbol{r}_i \leftarrow \boldsymbol{q}_i^+ + \boldsymbol{q}_i^-$;
  $\mathcal{S} \leftarrow \mathcal{S} \cup \{\boldsymbol{q}_i^+, \boldsymbol{q}_i^-, \boldsymbol{r}_i\}$;
for $1 \le i < j \le B$ **do**
  $\boldsymbol{d}_{ij} \leftarrow \boldsymbol{q}_i - \boldsymbol{q}_j$;
  $\boldsymbol{d}_{ij}^+ \leftarrow \mathbb{1}\{\boldsymbol{d}_{ij} = 1\}$;
  $\boldsymbol{d}_{ij}^- \leftarrow \mathbb{1}\{\boldsymbol{d}_{ij} = -1\}$;
  $\boldsymbol{r}_{ij} \leftarrow \boldsymbol{d}_{ij}^+ + \boldsymbol{d}_{ij}^-$;
  $\mathcal{S} \leftarrow \mathcal{S} \cup \{\boldsymbol{d}_{ij}^+, \boldsymbol{d}_{ij}^-, \boldsymbol{r}_{ij}\}$;
return $\mathcal{S}$;

Algorithm 2: OPU Decoder

Input: Ordered OPU outputs $\psi_{\text{opu}}(\boldsymbol{b}) = \left(|\boldsymbol{r}_i^\top \boldsymbol{b}|^2\right)_{i=1}^m$ for all $\boldsymbol{b} \in \mathcal{S}$, weights $(G_i)_{i=1}^B$
Output: $\psi_{\text{opu}}(\boldsymbol{q}) = \left(|\boldsymbol{r}_i^\top \boldsymbol{q}|^2\right)_{i=1}^m$
for $i = 1$ **to** B **do**
  $\psi_{\text{opu}}(\boldsymbol{q}_i) \leftarrow 2\psi_{\text{opu}}(\boldsymbol{q}_i^+) + 2\psi_{\text{opu}}(\boldsymbol{q}_i^-) - \psi_{\text{opu}}(\boldsymbol{r}_i)$;
for $1 \leq i < j \leq B$ **do**
  $\psi_{\text{opu}}(\boldsymbol{d}_{ij}) \leftarrow 2\psi_{\text{opu}}(\boldsymbol{d}_{ij}^+) + 2\psi_{\text{opu}}(\boldsymbol{d}_{ij}^-) - \psi_{\text{opu}}(\boldsymbol{r}_{ij})$;
  $\boldsymbol{c}_{ij} \leftarrow \psi_{\text{opu}}(\boldsymbol{q}_i) + \psi_{\text{opu}}(\boldsymbol{q}_j) - \psi_{\text{opu}}(\boldsymbol{d}_{ij})$;
for $i = 1$ **to** B **do**
  $\psi_{\text{opu}}(\boldsymbol{q}) \leftarrow \psi_{\text{opu}}(\boldsymbol{q}) + G_i^2 \psi_{\text{opu}}(\boldsymbol{q}_i)$;
for $1 \leq i < j \leq B$ **do**
  $\psi_{\text{opu}}(\boldsymbol{q}) \leftarrow \psi_{\text{opu}}(\boldsymbol{q}) + G_i G_j \boldsymbol{c}_{ij}$;
return $\psi_{\text{opu}}(\boldsymbol{q})$;

Remark 5.1 (Subspace-valued data and OPU)**.** *The symmetric quantisation and signed bit-plane decomposition introduced above are chosen to be useful for general subspace-valued data, whose basis vectors may contain both positive and negative entries. This is also the case for the image-based application we showed in the experimental section of Chap. 4. Although the original image intensities are non-negative, the subspaces are represented by orthonormal bases obtained from an SVD, whose entries are not in general positive. Hence, the vectors actually encoded for the OPU contain both positive and negative values, making the symmetric quantisation appropriate in this setting.*

5.4 Noise in optical embeddings

Since OPUs are physical objects, every time they are called upon, some noise is going to appear. Since embedding a single quantised vector requires a number of calls to the OPU that grows with the number of quantisation levels L we are going to study how the noise levels increase with L. We will first assume that the noise is centred on zero and study how the OPU encoding procedure described above amplifies its variance. We will then study the impact of a non-zero mean noise effect on the final kernel approximation with random features from the OPU.

5.4.1 Noise variance

For a binary vector $\boldsymbol{x}$, we will model a noisy debiased OPU measurement as

$$\psi^{-}_{\text{opu}}(\boldsymbol{x}) = \widetilde{\psi}^{-}_{\text{opu}}(\boldsymbol{x}) + \epsilon(\boldsymbol{x}), \tag{5.10}$$

where $\widetilde{\psi}^{-}_{\text{opu}}(\boldsymbol{x})$ denotes the ideal noiseless debiased measurement and $\epsilon(\boldsymbol{x})$ the random noise term induced by the OPU for one particular processing of $\boldsymbol{x}$.

Remark 5.2 (Noise model)**.** *The raw output of the OPU is an optical intensity and is therefore non-negative. The centred additive model we describe in Eq. 5.10 thus describes the effective noise after the debiasing operation in ψ^{-}_{opu}, which subtracts pairs of OPU measurements. If the noise affecting the two raw measurements has the same mean, this common mean is cancelled by the subtraction. We therefore assume a centred effective noise with variance σ^2 for what follows. This is a simplified model, since physical effects such as input-dependent noise, quantisation and clipping might not satisfy these assumptions.*

Since each output coordinate of $\psi^{-}_{\text{opu}}(\boldsymbol{x})$ is treated independently in the same manner, we work with one arbitrary coordinate and omit the coordinate index throughout this section.

We will also assume that the noise behaves as a random variable independent of the OPU entry and of the output coordinate considered and satisfying

$$\mathbb{E}[\epsilon(\boldsymbol{x})] = 0, \qquad \text{Var}(\epsilon(\boldsymbol{x})) = \sigma^2.$$

Under those conditions, we can formulate the following proposition about the variance of the noise when using the OPU for a vector quantised to L levels.

Proposition 5.2 (Noise amplification through the OPU decoder)**.** *Let $\boldsymbol{q}$ be a vector quantised with L levels and decomposed as*

$$\boldsymbol{q} = \sum_{i=1}^{B} G_i \boldsymbol{q}_i, \qquad B = \lceil \log_2 L \rceil.$$

Under the independent centred noise model above, the decoded measurement satisfies

$$\psi_{\text{opu}}^{-}(\boldsymbol{q}) = \widetilde{\psi}_{\text{opu}}^{-}(\boldsymbol{q}) + \epsilon(\boldsymbol{q}),$$

where

$$\operatorname{Var}(\epsilon(\boldsymbol{q})) = 9\sigma^2 \left(S^2 Q + \tfrac{1}{2}(Q^2 - R) \right), \tag{5.11}$$

with

$$S = \sum_{i=1}^{B} G_i, \qquad Q = \sum_{i=1}^{B} G_i^2, \qquad R = \sum_{i=1}^{B} G_i^4.$$

Proof. We will proceed by following the decomposition described in the proof of Prop. 5.1. Recall that for a quantised vector $\boldsymbol{q}$ we can decompose it following Eq. 5.7 and Eq. 5.8.

Let us first consider a ternary vector $\boldsymbol{s} \in \{-1, 0, 1\}^n$. By Eq. 5.7, its quadratic measurement is reconstructed from three OPU calls as

$$\psi_{\text{opu}}^{-}(\boldsymbol{s}) = 2\psi_{\text{opu}}^{-}(\boldsymbol{s}^+) + 2\psi_{\text{opu}}^{-}(\boldsymbol{s}^-) - \psi_{\text{opu}}^{-}(\boldsymbol{r}_s).$$

Using the noise model in Eq. 5.10, the noise in this reconstruction is

$$\eta_s := 2\epsilon(\boldsymbol{s}^+) + 2\epsilon(\boldsymbol{s}^-) - \epsilon(\boldsymbol{r}_s).$$

Since the three noise variables are independent and have variance σ^2, we obtain

$$\operatorname{Var}(\eta_s) = 4\sigma^2 + 4\sigma^2 + \sigma^2 = 9\sigma^2. \tag{5.12}$$

Let η_i denote the noise in the reconstructed measurement of $\boldsymbol{q}_i$, and let η_{ij}^d denote the noise in the reconstructed measurement of $\boldsymbol{d}_{ij} = \boldsymbol{q}_i - \boldsymbol{q}_j$. Applying the recombination formula in Eq. 5.9, the total decoder noise is

$$\epsilon(\boldsymbol{q}) = \sum_{i=1}^{B} G_i^2 \eta_i + \sum_{i<j} G_i G_j \left(\eta_i + \eta_j - \eta_{ij}^d \right).$$

The noise η_i appears in the diagonal term and in every cross-term involving bitplane i. Its total coefficient is therefore

$$G_i^2 + \sum_{j \neq i} G_i G_j = G_i \sum_{j=1}^{B} G_j = G_i S,$$

with $S := \sum_{i=1}^{B} G_i$. It follows that

$$\epsilon(\boldsymbol{q}) = \sum_{i=1}^{B} G_i S \eta_i - \sum_{i<j} G_i G_j \eta_{ij}^d. \tag{5.13}$$

All the noise variables in Eq. 5.13 are independent and have variance $9\sigma^2$ since they are induced by ternary vectors. Hence

$$\mathrm{Var}(\epsilon(\boldsymbol{q})) = 9\sigma^2\left(S^2\sum_{i=1}^{B} G_i^2 + \sum_{i<j} G_i^2 G_j^2\right).$$

Let

$$Q := \sum_{i=1}^{B} G_i^2 \qquad \text{and} \qquad R := \sum_{i=1}^{B} G_i^4,$$

we have

$$\begin{aligned} Q^2 &= \left(\sum_{i=1}^{B} G_i^2\right)^2 \\ &= \sum_{i=1}^{B} G_i^4 + 2\sum_{i<j} G_i^2 G_j^2, \end{aligned}$$

meaning that

$$\sum_{i<j} G_i^2 G_j^2 = \tfrac{1}{2}(Q^2 - R),$$

and therefore,

$$\mathrm{Var}(\epsilon(\boldsymbol{q})) = 9\sigma^2\left(S^2 Q + \tfrac{1}{2}(Q^2 - R)\right).$$

□

This proposition shows that the variance of an OPU output directly depends on the weights G_i induced by the quantisation. Without carefully selecting those, the risk is to create a noise whose variance grows without bound as L (and thus the number of calls to the OPU) increases. Luckily the next corollary shows that if we design the quantisation step with this in mind we can keep the noise variance bounded even for arbitrarily large values of L.

Lemma 5.1 (Limit of the decoder noise variance)**.** *Let $L = 2^B$ and $G_i = \delta 2^{i-2}$. If*

$$\delta = \tfrac{\alpha}{L},$$

where $\alpha > 0$ is a real constant, then

$$\lim_{B\to\infty} \mathrm{Var}(\epsilon(\boldsymbol{q})) = \tfrac{\sigma^2\alpha^4}{5}.$$

Proof. Using $G_i = \delta 2^{i-2}$ and the geometric-series identity

$$\sum_{k=0}^{B-1} r^k = \tfrac{r^B - 1}{r - 1},$$

we obtain

$$\begin{aligned} S &= \sum_{i=1}^{B} G_i = \tfrac{\delta}{2} \sum_{k=0}^{B-1} 2^k = \tfrac{\delta}{2}(2^B - 1), \\ Q &= \sum_{i=1}^{B} G_i^2 = \tfrac{\delta^2}{4} \sum_{k=0}^{B-1} 4^k = \tfrac{\delta^2}{12}(4^B - 1), \\ R &= \sum_{i=1}^{B} G_i^4 = \tfrac{\delta^4}{16} \sum_{k=0}^{B-1} 16^k = \tfrac{\delta^4}{240}(16^B - 1). \end{aligned}$$

Substituting $L = 2^B$ and $\delta = \alpha / L$ gives

$$S = \tfrac{\alpha}{2}\left(1 - \tfrac{1}{2^B}\right), \qquad Q = \tfrac{\alpha^2}{12}\left(1 - \tfrac{1}{4^B}\right), \qquad R = \tfrac{\alpha^4}{240}\left(1 - \tfrac{1}{16^B}\right).$$

Therefore,

$$\lim_{B\to\infty} S = \tfrac{\alpha}{2}, \qquad \lim_{B\to\infty} Q = \tfrac{\alpha^2}{12}, \qquad \lim_{B\to\infty} R = \tfrac{\alpha^4}{240}.$$

Substituting these limits into Eq. 5.11 gives

$$\lim_{B\to\infty} \operatorname{Var}(\epsilon(\boldsymbol{q})) = 9\sigma^2 \left(\tfrac{\alpha^4}{48} + \tfrac{1}{2}\left(\tfrac{\alpha^4}{144} - \tfrac{\alpha^4}{240}\right)\right) = \tfrac{\sigma^2\alpha^4}{5}.$$

□

If we take $\sigma = 1$ and $\alpha = 1$, we can numerically find the relative increase in standard deviation compared to when $L = 2$:

L	increase from $L = 2$
2	1.00
4	1.75
8	2.09
16	2.24

This result is valid for any value of σ and α since they just show the increase relative to the variance when $L = 2$.

Corollary 5.1 (Noise variance for subspace embeddings)**.** *Let $\boldsymbol{U} = (\boldsymbol{q}_1, \ldots, \boldsymbol{q}_k) \in \mathbb{V}(k, n)$ be the basis of a subspace $\mathcal{U} \in \mathcal{G}(k, n)$, with each $\boldsymbol{q}_\ell$ quantised with the same δ. Let $\epsilon(\boldsymbol{q}_\ell)$ denote the noise associated with $\psi_{\text{opu}}(\boldsymbol{q}_\ell)$ with $\operatorname{Var}(\epsilon(\boldsymbol{q}_\ell))$ given by Lemma 5.1 for all l. Then*

$$\operatorname{Var}(\epsilon(\boldsymbol{U})) = k \operatorname{Var}(\epsilon(\boldsymbol{q})).$$

Proof. The decoded embedding of $\boldsymbol{U}$ is obtained by summing the decoded embeddings of its k quantised columns. Therefore the total noise is

$$\epsilon(\boldsymbol{U}) = \sum_{\ell=1}^{k} \epsilon(\boldsymbol{q}_\ell).$$

Assuming that the noises associated with the different columns are independent, and since every column is quantised using the same step δ, Lemma 5.1 gives the same variance for every $\boldsymbol{q}_k$. Hence

$$\mathrm{Var}(\epsilon(\boldsymbol{U})) = k\,\mathrm{Var}(\epsilon(\boldsymbol{q})).$$

□

Since the factor k does not depend on L, the relative increases in standard deviation given in the previous table remain unchanged.

In practice, for a subspace $\boldsymbol{U} \in \mathbb{V}(k, n)$, we set

$$\delta = \tfrac{2M}{L}, \qquad M = \max_{i,j} |U_{ij}|.$$

This choice divides the range $[-M, M]$ into L intervals of width δ. Moreover, since $\delta = 2M/L$, the condition of Lemma 5.1 holds with $\alpha = 2M$, and the variance of the decoded noise remains bounded as L increases.

5.4.2 Effect of a non-zero mean

The previous result assumes that the noise in every OPU call has zero mean. The final decoded noise may, however, have a non-zero mean because the physical noise can depend on the input, on the output coordinate, or on the part of the OPU output being used. Let $\widetilde{\psi}^{-}_{\mathrm{opu}}(\boldsymbol{U}) \in \mathbb{R}^m$ denote the ideal decoded and debiased embedding of a subspace $\boldsymbol{U}$, and let

$$\psi^{-}_{\mathrm{opu}}(\boldsymbol{U}) = \widetilde{\psi}^{-}_{\mathrm{opu}}(\boldsymbol{U}) + Z(\boldsymbol{U})$$

denote the observed embedding, where $Z(\boldsymbol{U}) \in \mathbb{R}^m$ is the additive noise introduced by the physical OPU and the decoding procedure. We define

$$\mu(\boldsymbol{U}) = \mathbb{E}[Z(\boldsymbol{U})],$$

which is the mean noise vector. For two subspaces $\boldsymbol{U}$ and $\boldsymbol{V}$, their embeddings can be used as random feature vectors to approximate a kernel between them. Taking the noise into account gives

$$\begin{aligned}\psi^{-}_{\mathrm{opu}}(\boldsymbol{U})^\top \psi^{-}_{\mathrm{opu}}(\boldsymbol{V}) =& \widetilde{\psi}^{-}_{\mathrm{opu}}(\boldsymbol{U})^\top \widetilde{\psi}^{-}_{\mathrm{opu}}(\boldsymbol{V}) + \widetilde{\psi}^{-}_{\mathrm{opu}}(\boldsymbol{U})^\top Z(\boldsymbol{V}) \\ &+ \widetilde{\psi}^{-}_{\mathrm{opu}}(\boldsymbol{V})^\top Z(\boldsymbol{U}) + Z(\boldsymbol{U})^\top Z(\boldsymbol{V}).\end{aligned}$$

Assume that the ideal embeddings are centred and independent of the noise. Taking the expectation gives

$$\mathbb{E}[\psi^{-}_{\mathrm{opu}}(\boldsymbol{U})^\top \psi^{-}_{\mathrm{opu}}(\boldsymbol{V})] = \kappa(\boldsymbol{U}, \boldsymbol{V}) + \mu(\boldsymbol{U})^\top \mu(\boldsymbol{V}) + \mathrm{tr}\left(\mathrm{Cov}(Z(\boldsymbol{U}), Z(\boldsymbol{V}))\right) \tag{5.14}$$

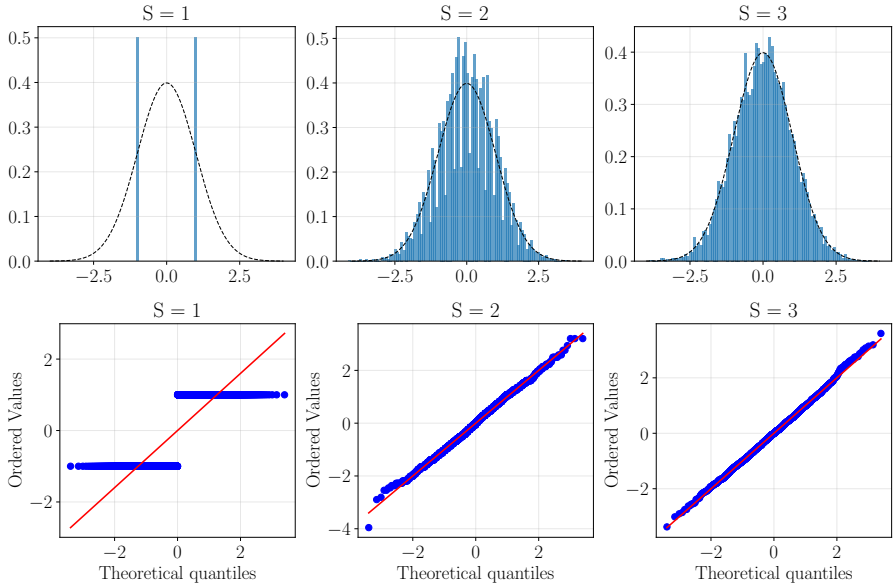


Fig. 5.2 **Top:** Empirical distribution of a row of the structured Hadamard-diagonal embedding for increasing numbers of Hadamard blocks S. **Bottom:** Q-Q plots comparing the row-wise distribution to a standard Gaussian.

where $\kappa(\boldsymbol{U}, \boldsymbol{V})$ denotes the kernel induced by the random mapping ψ_{opu}^{-} and where

$$\text{Cov}(Z(\boldsymbol{U}), Z(\boldsymbol{V})) = \mathbb{E}\left[(Z(\boldsymbol{U}) - \mu(\boldsymbol{U}))(Z(\boldsymbol{V}) - \mu(\boldsymbol{V}))^\top\right].$$

A non-zero mean therefore adds the term $\mu(\boldsymbol{U})^\top \mu(\boldsymbol{V})$ to the expected kernel. If the noises associated with $\boldsymbol{U}$ and $\boldsymbol{V}$ are independent, the cross-covariance term vanishes.

5.5 Experiments

In this section we aim to repeat the experiments on subspace classification from Chap. 4. We first analyse the structured embedding and how to use it in practice. We then turn our attention to the more delicate OPU case with an analysis of the theoretical results found earlier and application to the same classification problem.

5.5.1 Analysis of structured embeddings

In this experiment, we investigate how the depth S of the structured Hadamard-diagonal embedding influences the behaviour of individual

rows of the resulting random projection matrix. For a fixed dimension n and a fixed row index j, we consider the random vector $\boldsymbol{u} = \sqrt{n}\, \boldsymbol{e}_j^\top \boldsymbol{G}$, where $\boldsymbol{G}$ is generated according to the model in Eq. 5.1, and $\boldsymbol{e}_j$ denotes the j-th canonical basis vector of $\mathbb{R}^n$. For increasing values of the number of Hadamard blocks S, we examine the empirical distribution of the entries of $\boldsymbol{u}$, which corresponds to the distribution of the coefficients of a single row of the structured embedding matrix.

Figure 5.2 shows both the empirical histograms of the row entries (top) and the corresponding Q-Q plots against a standard Gaussian (bottom). For $S = 1$, the distribution is clearly non-Gaussian. As S increases, the empirical distributions progressively smooth out and become increasingly symmetric, while the Q-Q plots approach the identity line. Already at $S = 3$, the row-wise distribution closely matches a Gaussian.

These observations empirically support the use of $S = 3$ Hadamard-diagonal blocks to generate structured random projections with an approximately Gaussian distribution. This corroborates earlier findings on structured matrices [43, 23]. In all subsequent experiments, we will therefore fix $S = 3$ unless otherwise stated.

5.5.2 Experiments on subspaces with structured embeddings

We now return to the ETH-80 subspace classification experiments of Sec. 4.7.4. The goal here is not to introduce a new classification protocol, but to study whether the embeddings of Chap. 4 can be replaced by the structured embeddings introduced in the present chapter.

We keep the same dataset, preprocessing, train-test splits and classifiers as in Sec. 4.7.4. Each image is resized to 32×32, so that $n = 1024$, and each subspace is of dimension $k = 9$. We also keep the same two tasks. The first one is superclass classification, where each object is represented by a single subspace and the label is the superclass. The second one is object classification, where each individual object defines its own class and several subspaces are generated for each object in the exact same way as in Sec. 4.7.4.

In Chap. 4, we compared exact Grassmannian kernels with Gaussian rank-one random features and their bounded variants. We now add the structured embeddings introduced in this chapter. For a basis $\boldsymbol{U} \in \mathbb{V}(k, n)$, we consider the real-valued, binary and periodic structured embeddings

$$\psi_{\mathrm{st}}(\boldsymbol{U}), \qquad \operatorname{sign}(\psi_{\mathrm{st}}(\boldsymbol{U})), \qquad \exp(\mathrm{i}\omega\psi_{\mathrm{st}}(\boldsymbol{U})).$$

with ω a real parameter. As before, the embedding size is described

through the oversampling ratio

$$\rho = \frac{m}{nk}.$$

The ψ_{st} embeddings use $S = 3$ Hadamard–diagonal blocks, in line with the empirical analysis of Sec. 5.5.1. We report the same two values $\rho = 0.05$ and $\rho = 0.20$ as in Chapter 4, which makes the comparison with the Gaussian embeddings direct.

Superclass classification

We first repeat the superclass classification experiment. Since this setting contains only 56 training subspaces and 24 test subspaces, it is not the regime where random features are expected to bring the largest computational advantage. It is nevertheless useful as a first check that replacing Gaussian vectors by structured ones does not destroy the approximation behaviour observed in Chap. 4.

Fig. 5.3 shows the classification accuracy obtained with the structured embeddings as a function of ρ. The corresponding runtimes and accuracies at $\rho = 0.05$ and $\rho = 0.20$ are reported in Tab. 5.3, together with the exact kernels and Gaussian embeddings from Sec. 4.7.4.

We see that the structured embedding strategy gives very similar trends compared to the unstructured one showing an increase in accuracy as ρ grows. Furthermore, in spite of their much lower time requirements (from 10 to 20$\times$ faster than the unstructured version), the structured embeddings still give very good accuracy. At $\rho = 0.20$, the structured binary embedding reaches 96.46% accuracy, while the real-valued and periodic structured embeddings reach 93.54%.

In this setting, the main conclusion is therefore not that structured embeddings outperform exact kernels, but that they recover most of the classification accuracy while greatly reducing the cost of the random feature computation.

Object classification

We now repeat the object classification experiment, where each of the 80 objects defines its own class. This setting is more demanding than superclass classification as it contains 10$\times$ more subspaces. It is therefore better suited to show the computational benefit of replacing exact kernel computations or Gaussian random features by structured embeddings.

Fig. 5.4 shows the classification accuracy obtained with structured embeddings as a function of ρ. The timing and accuracy values at $\rho = 0.05$ and $\rho = 0.20$ are reported in Tab. 5.4.

The Gaussian embeddings from Chap. 4 approach the exact projection kernel as ρ increases, but this comes with a larger runtime cost than the non-embedded classification. At $\rho = 0.20$, the Gaussian ROP

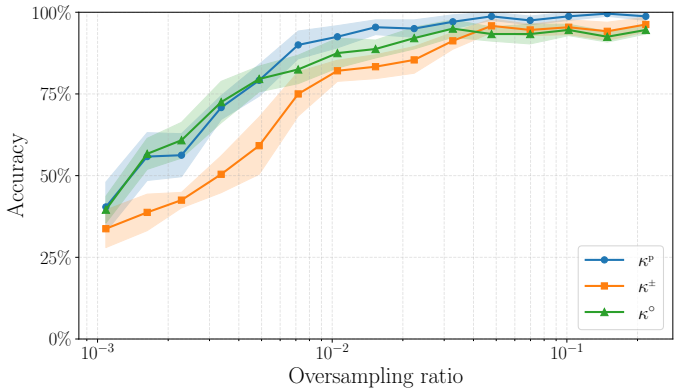


Fig. 5.3 ETH-80 superclass classification with structured embeddings using $S = 3$ Hadamard-diagonal blocks.

Method	ρ	Total time (s)	Accuracy (%)
Exact kernels			
Projection kernel κ^{P}	—	0.2633	100.00
Periodic kernel κ°	—	0.2381	100.00
Gaussian embeddings from Chapter 4			
Gaussian ROP	0.05 / 0.20	1.3211 / 5.3102	98.12 / 99.79
Gaussian binary ROP	0.05 / 0.20	1.3211 / 5.3111	93.12 / 96.04
Gaussian periodic ROP	0.05 / 0.20	1.3240 / 5.3145	93.54 / 93.75
Structured embeddings			
Structured ROP	0.05 / 0.20	0.1046 / 0.1963	94.79 / 93.54
Structured binary ROP	0.05 / 0.20	0.1048 / 0.1972	92.29 / 96.46
Structured periodic ROP	0.05 / 0.20	0.1055 / 0.2025	94.79 / 93.54

Table 5.3 Superclass classification with exact kernels, Gaussian embeddings and structured embeddings. Total time includes embedding computation and SVM training and evaluation. Random embedding results are averaged over 20 trials.

embedding reaches 94.06% accuracy but requires 85 seconds in total compared to 53 seconds.

The structured embeddings keep essentially the same accuracy as their Gaussian counterparts, while being much faster to compute. At $\rho = 0.20$, the structured variants are about $20\times$ faster than the corresponding Gaussian embeddings. In all three cases, the accuracy remains within less than 2% of the Gaussian version, with the binary and periodic variants almost unchanged. This confirms that the structured construction mostly preserves the behaviour of the embeddings from Chap. 4, while substantially reducing their computational cost.

This experiment is the clearest illustration of the role of structured embeddings. They do not change the kernels studied in Chap. 4. In-

stead, they provide faster approximations of the same random feature constructions. Compared with Gaussian embeddings, the loss in accuracy is moderate at $\rho = 0.20$, while the runtime reduction is large. Compared with exact kernels, the structured methods avoid direct pairwise kernel computations and provide explicit features that can be used with linear learning methods.

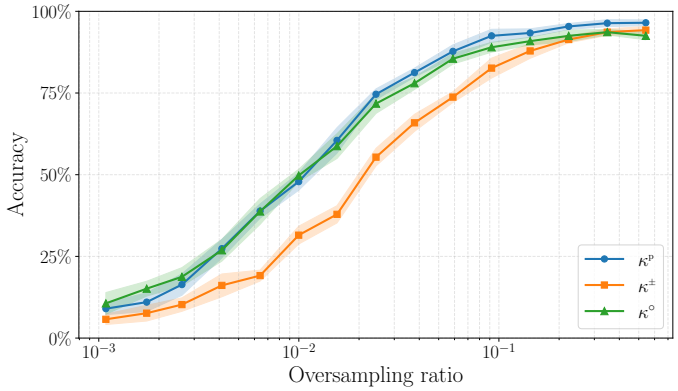


Fig. 5.4 ETH-80 object classification with structured embeddings using $S = 3$ Hadamard–diagonal blocks.

Method	ρ	Total time (s)	Accuracy (%)
Exact kernels			
Projection kernel κ^{p}	—	52.8856	98.75
Periodic kernel κ°	—	58.7874	90.00
Gaussian embeddings from Chapter 4			
Gaussian ROP	0.05 / 0.20	24.1868 / 85.2087	84.19 / 94.06
Gaussian binary ROP	0.05 / 0.20	24.2049 / 85.2449	72.25 / 91.75
Gaussian periodic ROP	0.05 / 0.20	24.3763 / 86.4142	81.38 / 92.31
Structured embeddings			
Structured ROP	0.05 / 0.20	1.8729 / 4.0293	83.00 / 92.38
Structured binary ROP	0.05 / 0.20	1.8925 / 4.0837	72.13 / 91.25
Structured periodic ROP	0.05 / 0.20	1.9967 / 5.2383	83.00 / 92.38

Table 5.4 Object classification with exact kernels, Gaussian embeddings and structured embeddings. Total time includes embedding computation and SVM training and evaluation. Random embedding results are averaged over 20 runs.

5.5.3 Analysis of the OPU

Before applying the encoded optical embeddings to classification, we examine two effects that cause the physical OPU to differ from the ideal model. We first study how its output depends on the density of a bi-

nary input. We then measure the noise remaining after the complete encoding, decoding and debiasing procedure.

Effect of input density. The ideal Gaussian model does not fully describe the measurements produced by the physical OPU. The camera records each light intensity using 8 bits, giving 256 possible values between 0 and 255 and there is noise present in every measurement. Other physical effects may also affect the output, but we do not consider them here. The physical measurement is therefore better described by

$$\psi_{\text{opu}}(\boldsymbol{x})_j = \mathcal{Q}_8\big(\widetilde{\psi}_{\text{opu}}(\boldsymbol{x})_j + \epsilon(\boldsymbol{x})_j\big), \tag{5.15}$$

where $\widetilde{\psi}_{\text{opu}}$ denotes the ideal natural OPU embedding, $\epsilon(\boldsymbol{x})_j$ denotes the noise affecting coordinate j, and $\mathcal{Q}_8$ is the 8-bit quantiser used by the OPU camera. Denoting its clipping level by $C > 0$, its quantisation step is $\frac{C}{2^8}$. Values close to zero are therefore difficult to distinguish from the noise and from the first quantisation levels. Values above the available camera range are clipped to the largest output value.

Both effects depend on the density of the binary input denoted by d. Let $\boldsymbol{b} \in \{0,1\}^n$ and define

$$s = \|\boldsymbol{b}\|^2 = |\operatorname{supp}(\boldsymbol{b})|, \qquad d = \tfrac{s}{n}.$$

The density d is the proportion of non-zero entries in $\boldsymbol{b}$. Since the entries of $\boldsymbol{b}$ are binary, its squared norm is exactly equal to its number of non-zero entries. Under the ideal Gaussian model, $\|\psi^{-}_{\text{opu}}(\boldsymbol{b})\|$ should be proportional to $s = \|\boldsymbol{b}\|^2$.

We tested this numerically by taking vectors $\boldsymbol{b}_i \in \{0,1\}^{1024}$ with varying levels of sparsity and measuring $\|\psi^{-}_{\text{opu}}(\boldsymbol{b}_i)\|$ against $\|\boldsymbol{b}_i\|^2$. Fig. 5.5 shows that for densities below approximately 20%, the optical signal is weak and the output seems strongly affected by noise and for densities above approximately 80%, an increasing number of camera pixels reach their maximal value, which breaks the linear dependence on the norm. The OPU therefore behaves closest to the ideal model when the binary input density lies between these two values. This observation will be important for the encoded subspace embeddings, since their binary decomposition can produce both very sparse and very dense inputs.

Noise after decoding. We next measure the noise affecting the final decoded embeddings. We fix a subspace basis $\boldsymbol{U}_0 \in \mathbb{V}(9, 1024)$ and apply the complete quantisation, encoding, optical measurement, decoding and debiasing procedure with $R = 20$ repetitions. We repeat the experiment for quantisation levels $L \in \{2, 4, 8, 16\}$. The OPU produces 2^{17} random features before debiasing, resulting in a final embedding

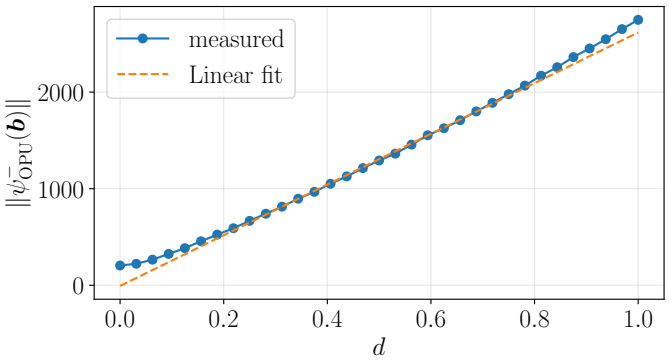
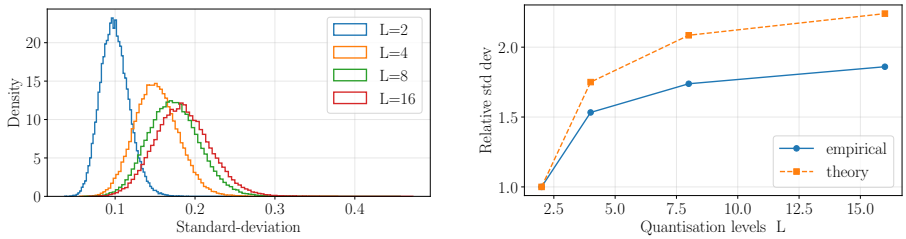


Fig. 5.5 Norm of the debiased OPU embedding as a function of the density s/n of the binary input. Each input is measured 20 times. The approximately linear behaviour is mainly observed when the proportion of active entries lies between 20% and 80%.

dimension $m = 65\,536$. We collect the resulting embeddings for every quantisation level L in a large matrix $\boldsymbol{\Phi}_L \in \mathbb{R}^{65\,536\times 20}$ and compute the variance of each row of $\boldsymbol{\Phi}_L$. This tells us how much an entry of the output vector (supposed to remain constant in the noiseless case) will vary over 20 trials. This is shown on the left side of Fig. 5.6. We can see that the empirical distributions of those variances shift towards higher values as L grows, and the shift is strongest going from $L = 2$ to $L = 4$. It then becomes smaller for subsequent values.

We also average the standard deviations of the output to obtain a single empirical noise level for each value of L. We compare this quantity with the standard deviation predicted by the noise model derived from Lem. 5.1, after estimating the base noise variance from the experiment with $L = 2$. The result is shown on the right side of Fig. 5.6. The empirical and theoretical curves follow the same trend, with a large increase from $L = 2$ to $L = 4$ followed by progressively smaller increases. However, the theoretical model predicts a larger standard deviation increase. This discrepancy may result from effects that are not captured by the simplified noise model. For example, Lem. 5.1 assumes that every binary OPU call has the same independent noise variance. However, the previous experiment has shown that the assumption of having a common noise model for every binary vector input is likely inaccurate.

Finally, we examine whether the final debiased embeddings remain centred around zero. For each repetition, we average the entries of $\bar{\psi}_{\mathrm{opu}}(\boldsymbol{U}_0)$, then average the resulting values over the 20 trials. Since the ideal debiased embedding is centred on zero in expectation, this quantity provides an estimate of an additional scalar bias introduced



(a) Distribution of the empirical variances per coordinate.

(b) Mean standard deviation.

Fig. 5.6 Noise measured after the complete OPU encoding and decoding procedure over 20 repetitions of the same subspace.

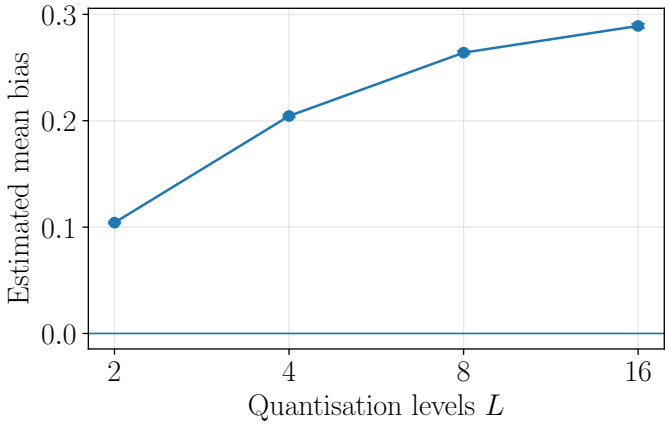


Fig. 5.7 Bias measured after the complete OPU encoding and decoding procedure over 20 repetitions of the same subspace.

by the physical OPU. Figure 5.7 shows that this estimated bias increases with the number of quantisation levels. The increase is again strongest between $L = 2$ and $L = 4$ and becomes smaller for larger values of L, following a trend similar to the standard deviation measured in Fig. 5.6.

5.5.4 Experiments on subspaces with the OPU

We now apply the complete OPU procedure to the two ETH-80 classification tasks considered in Chap. 4 and earlier in this chapter. We keep the same data, train-test splits, subspace dimensions and classification protocol. The only change is the embedding used to represent each subspace.

Each basis $\boldsymbol{U} \in \mathbb{V}(9, 1024)$ is first quantised using the procedure introduced in Algo. 1. Its columns are then decomposed into binary vectors, which are transmitted to the OPU. The optical measurements

are combined by the decoder to reconstruct the quadratic embedding of the quantised basis, after which two halves of the OPU output are subtracted to form the debiased embedding $\psi^{-}_{\text{opu}}(\boldsymbol{U})$.

We repeat the experiment for $L \in \{2, 4, 8, 16\}$. The physical OPU returns 2^{17} coordinates before debiasing. Subtracting the two halves therefore gives a final embedding dimension of $m = 65\,536$.

To study the influence of m, we randomly permute the coordinates of each complete embedding and retain only the first m coordinates. This truncation is performed after the optical measurements have been acquired so that the OPU is only used once for the whole dataset.

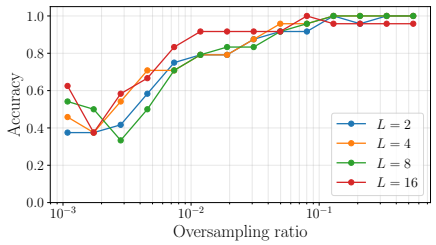
For each value of L, we consider the three kernel approximations used in Chap. 4 (raw, sign, and periodic) and apply the same classification method as in Sec. 4.7. Again, we show the classification accuracy as a function of the oversampling ratio $\rho = m/(nk)$.

Classification with an ideal numerical OPU. Before considering the measurements from the physical machine, we repeat the experiment with a noiseless numerical implementation of the ideal OPU model. The numerical model uses a fixed Gaussian matrix and computes the corresponding quadratic measurements. It therefore preserves the random matrix between different inputs and different values of L, as does the physical machine, while removing camera quantisation, measurement noise and clipping. Of course since this procedure is computed with traditional code, we cannot push the number of random features as high as the OPU without incurring massive computational costs. We therefore keep $m \leq 5000$, which will be sufficient to observe the effects of the number of quantisation levels.

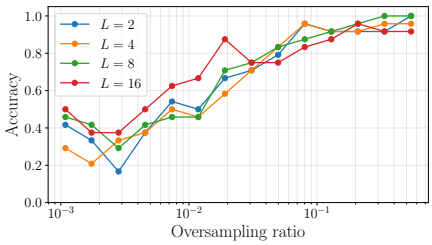
We first tested the complete quantisation, binary encoding, numerical OPU measurement and decoding procedure. The decoded measurements are the same as the direct quadratic measurements of the quantised subspaces up to numerical precision. This confirms that any difference between the numerical and physical experiments does not come from an algebraic error in the encoder or decoder.

The results are shown in Fig. 5.8 for the superclass setting and in Fig. 5.9 for the object setting. In the superclass setting, the influence of L is limited. Increasing L only slightly improves the rate at which the accuracy grows with m.

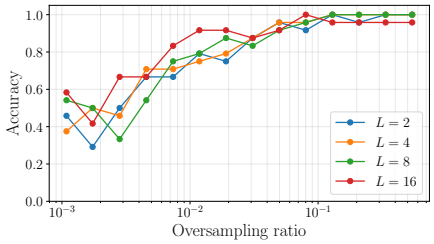
The influence of quantisation is much clearer in the object classification setting. At $m = 5000$, the raw feature accuracy is about 55% for $L = 2$, 54% for $L = 4$, 65% for $L = 8$ and 94% for $L = 16$, showing an increase of nearly 40% accuracy from $L = 2$ to $L = 16$. Similar trends can be observed in the other two feature types. Finer quantisation therefore gives a substantial improvement on this task.



(a) Raw features.



(b) Binary features.



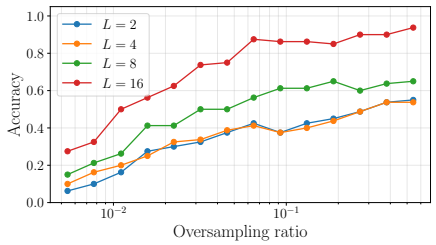
(c) Periodic features.

Fig. 5.8 Superclass classification using noiseless numerical OPU measurements. Each curve corresponds to a number of quantisation levels L.

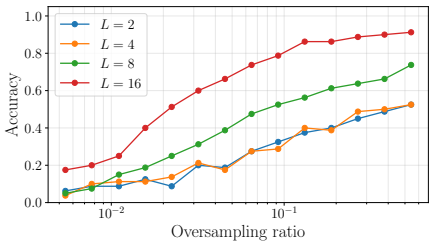
It must be noted that the limited impact of a larger L for the superclass setting can be explained in part by the fact that the problem is by nature easier. We can indeed recall that in Chap. 4 the random feature constructions all reached close to 100% even with an oversampling ratio of $\rho = 0.2$. By contrast, the object classification problem required more random features and did not reach such a high accuracy score. This suggests that it is much more subtle as it requires smaller details of objects to be taken into account which is why a finer quantisation improves the classification result significantly.

Classification with the physical OPU. We now replace the ideal numerical measurements with the measurements returned by the physical OPU. The same quantisation, encoding, decoding and classification procedures are used. The superclass and object classification results are shown in Figs. 5.10 and 5.11 respectively.

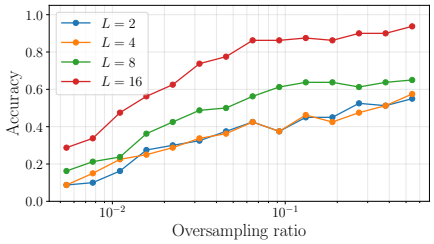
The physical measurements show the complete *opposite* trend compared to the numerical experiment. For superclass classification, $L = 2$ and $L = 4$ both reach perfect accuracy with m large enough for all three kernel approximations. However, the performance then decreases in all three kernels for $L = 8$ and decreases further for $L = 16$. At the largest embedding dimension, the raw features reach about 88% accuracy for $L = 8$ and 67% for $L = 16$. The periodic features reach 88% and 75%, while the binary features reach 58% and 50% for the same values of L.



(a) Raw features.



(b) Binary features.



(c) Periodic features.

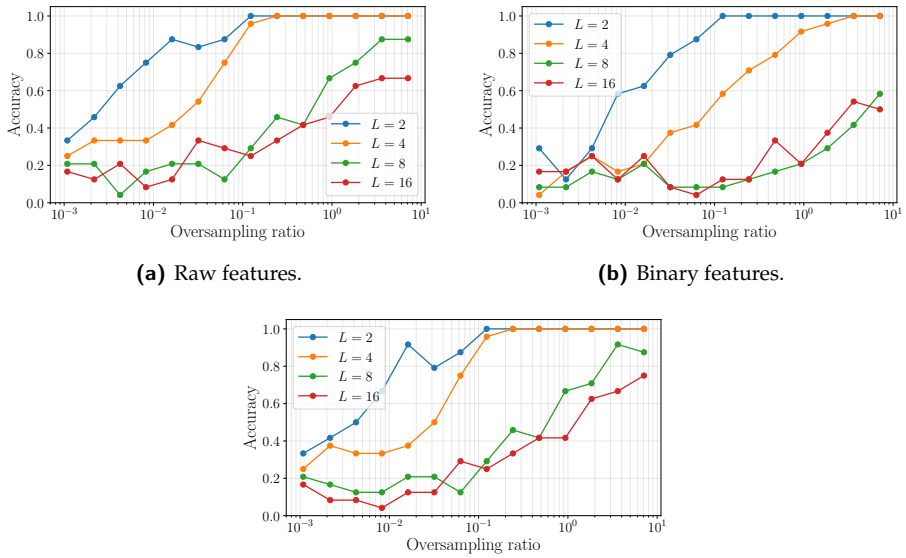
Fig. 5.9 Object classification using noiseless numerical OPU measurements. Each curve corresponds to a number of quantisation levels L.

This trend is even stronger for the object classification problem. The best results are obtained with $L = 2$ and performance then decreases for all higher values of L. Furthermore, the maximum accuracy reached using the physical OPU does not exceed 65% across all methods and quantisation levels for this object classification problem, by contrast with the numerical OPU which reaches over 85% accuracy in some cases.

The object classification curves also show that increasing m does not seem to increase accuracy beyond a certain point. The accuracy initially grows as more features are considered, but then reaches a plateau. The degradation in accuracy associated with a larger L therefore cannot be explained only by an embedding dimension too low.

This conclusion is unchanged by removing the bias considered in Sec. 5.5.3. For each embedded subspace, we subtract the average of the coordinates of its debiased embedding. This removes an estimated bias induced by the physical OPU. This shows that the observed degradation cannot be removed by subtracting a single bias from every entry of the random feature vectors.

Density of the encoded binary inputs. In order to investigate why the performances of the real OPU do not match their expected behaviour, we will verify whether the conditions for optimal OPU performance laid out in Sec. 5.5.3 are satisfied. Our analysis showed that



(a) Raw features.

(b) Binary features.

(c) Periodic features.

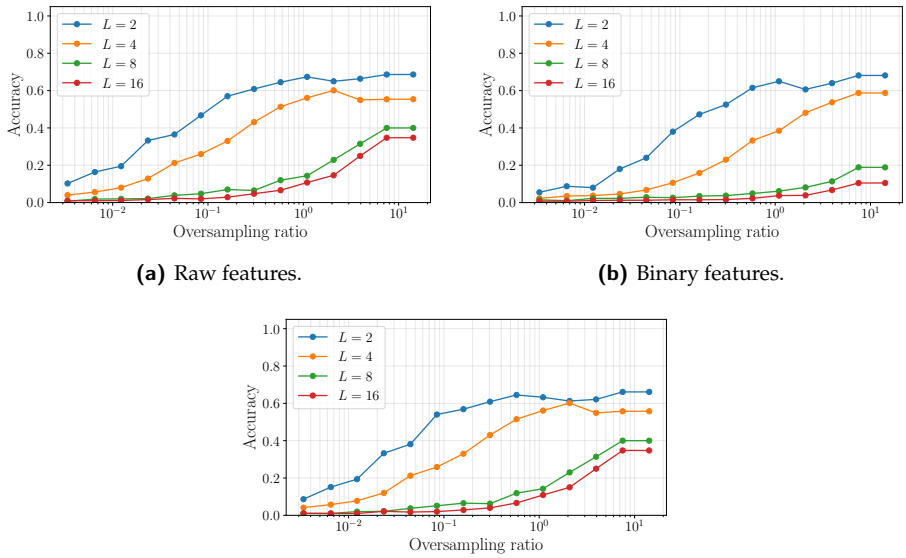
Fig. 5.10 Superclass classification using the physical OPU. Each curve corresponds to a number of quantisation levels L. The OPU produces 2^{17} coordinates before debiasing and 65536 coordinates after debiasing.

the physical OPU behaves closest to its ideal model when the density d of a binary input lies approximately between 0.2 and 0.8. We now examine whether the binary vectors produced by the encoder satisfy this condition.

For every training and test subspace from the object classification problem, we quantise its basis and collect all binary vectors generated by the encoder from Algo. 1. We then compute their density and repeat this procedure for every value of L. The resulting empirical distributions are shown in Fig. 5.12. The two vertical lines indicate the approximate optimal operating range of 0.2 to 0.8 identified in Fig. 5.5.

A large proportion of the encoded vectors lies outside the interval between 0.2 and 0.8. In the object classification setting, this proportion is approximately 41% for $L = 2$, 61% for $L = 4$, 70% for $L = 8$ and 74% for $L = 16$.

This already paints a bleak picture. However if we take the weights added in the decoder from Algo. 2 into account, we realise that some values with larger weights have greater influence than others with small weights. Some measurements are indeed multiplied by larger coefficients than others during decoding. We therefore repeat the calculation by weighting every binary OPU measurement with the weights G_i. Under this weighting, the proportion associated with inputs out-



(a) Raw features.

(b) Binary features.

(c) Periodic features.

Fig. 5.11 Object classification using the physical OPU. Each curve corresponds to a number of quantisation levels L. The OPU produces 2^{17} coordinates before debiasing and 65536 coordinates after debiasing.

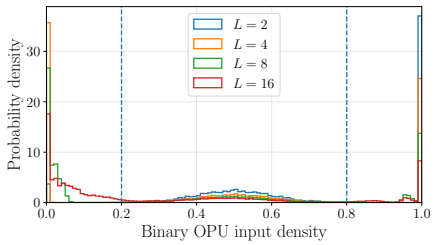
side the operating range reaches approximately 79% for $L = 4$, 95% for $L = 8$, and 99% for $L = 16$.

Note that if we consider the subspaces in the superclass dataset, the results are very similar. We choose not to report them here since they do not add much information.

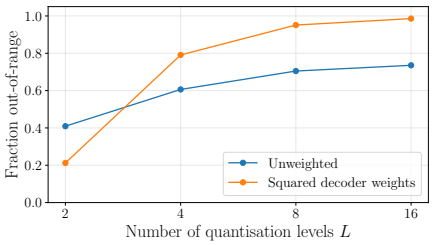
Interpretation of the results. The numerical and physical experiments showed two completely opposite effects: on the one hand, increasing the number of quantisation levels had a positive impact on results, on the other hand, the other setting showed a decrease in performance when L increases.

This first shows that the encoding and decoding procedures of Algos. 1 and 2 work as intended on a perfect numerical model but seem unfit for the real OPU. We can indeed see that the binarised vectors created by the encoding procedure lie increasingly outside the efficient operating range of the OPU (Fig. 5.6). Furthermore, the increased number of OPU calls as L grows also increases the physical noise calculated in theory in Sec. 5.4 and empirically in Sec. 5.5.3.

These observations may explain the classification trends in both settings. Increasing L reduces quantisation error, but also increases the decoded noise and the contribution of binary inputs outside the preferred density range. The first effect can initially compensate for the second,



(a) Empirical density distributions of the encoded binary OPU inputs for object classification. The vertical dashed lines mark the interval $[0.2, 0.8]$.



(b) Proportion of encoded binary inputs with density outside $[0.2, 0.8]$ for object classification. We compare the unweighted proportion with the proportion weighted by the squared decoder coefficients.

Fig. 5.12 Density analysis of the binary vectors generated by the encoder in the object-classification setting. The superclass setting shows the same qualitative behaviour and is omitted for brevity.

while for larger L the gain from finer quantisation becomes smaller but the decrease of performance because of physical constraints dominates.

This explanation may be sufficient to justify the results, however it does not allow us to draw further conclusions as to whether one effect is more or less influential on the final result. Furthermore, other effects may be present inside the OPU and have thus not been taken into account.

5.6 Conclusion

This chapter studied two ways of reducing the cost of the random embeddings introduced in the previous chapters. The first replaces independent Gaussian vectors with structured Hadamard-diagonal transformations. The second applies the computation of quadratic random measurements with an optical processing unit.

The structured construction preserves the form of the rank-one embeddings from Chap. 4, while reducing their memory and computational requirements. Using $S = 3$ Hadamard-diagonal blocks produces vectors whose empirical distributions are already close to Gaussian, as illustrated in Fig. 5.2. The ETH-80 experiments confirm that this change allows us to reach similar levels of accuracy in the same experimental setting. At oversampling level $\rho = 0.20$, the three structured variants remain within less than 2% of their Gaussian equivalents in the object classification experiment, while reducing the computation time by approximately a factor of 20, as shown in Tab. 5.4. The structured embeddings therefore provide a practical way to keep the behaviour of the random features from Chap. 4 at a substantially lower cost.

The OPU follows a different strategy by implementing a fixed quadratic random transformation through the propagation of light in a scattering medium instead of constructing and applying the random vectors digitally. Its binary input constraint does not allow the direct processing of real-valued vectors as considered in the previous chapters. We therefore introduced an encoder that decomposes a vector quantised with L levels into binary vectors, together with a decoder that reconstructs its quadratic embedding. Proposition 5.1 shows that this reconstruction is exact in the ideal noiseless model and requires $\mathcal{O}((\log_2 L)^2)$ OPU calls.

The repeated optical calls also introduce physical noise. Proposition 5.2 describes how this noise is amplified by the decoder, while Lemma 5.1 shows that its variance remains bounded when the quantisation step decreases proportionally to $1/L$. The experiments in Fig. 5.6 empirically showed that this behaviour was slightly less noisy than predicted by theory but followed the same trend with the largest increase in noise occurring between $L = 2$ and $L = 4$, followed by progressively smaller increases.

The numerical OPU experiments confirm that the encoding and decoding procedure behaves as intended in the ideal model. In particular, finer quantisation substantially improves the more difficult object classification task, as shown in Fig. 5.9. The physical OPU produces the opposite trend. Increasing L lowers the classification accuracy in both settings, and the object classification performance remains far below what was obtained with the structured embeddings.

The analysis of the encoded binary inputs provides a possible explanation for this discrepancy. The physical OPU behaves closest to its ideal model when the binary input density lies approximately between 0.2 and 0.8, as shown in Fig. 5.5. However, the encoder increasingly produces heavily weighted binary vectors outside this interval as L grows. Figure 5.12 shows that most of the contribution predicted by the weighted decoder coefficients is associated with such inputs for the largest values of L. The gain obtained from finer quantisation is therefore opposed by increased noise and by measurements taken outside the preferred operating range of the device.

The optical experiments, however, demonstrate the mathematical construction of the encoder and decoder. Quantised quadratic subspace embeddings can be reconstructed exactly from binary optical measurements in the ideal model. The limitation is in the interaction between the encoder and the physical properties of the available OPU. In its current form, the method is therefore a demonstration of feasibility rather than a competitive implementation of the considered classi-

fication tasks. A more effective optical construction would require an encoder designed jointly with the physical device, so that the binary inputs remain within a suitable density range while limiting both the decoder coefficients and the number of optical calls.

6
Conclusion

We first thank the honourable reader for making it this far, assuming of course that the reader did not jump directly to the conclusion.

Having presented the main results obtained over the past few years, we now step back and consider the manuscript as a whole. This final chapter summarises its principal contributions, discusses their limitations, and outlines several directions for future work.

6.1 Summary and future work

Chap. 2 began with the two main difficulties associated with large datasets. Individual objects may have a high dimension n, while the dataset itself may contain a large number N of objects. Both quantities can make storage and computation difficult. Kernel methods are particularly attractive because they allow complex comparisons through an implicit feature representation, but they also illustrate the difficulty created by a large N, since they rely on pairwise evaluations and the construction of Gram matrices. Random features provide a way to retain the advantages of kernel methods while replacing the implicit feature map with an explicit finite-dimensional embedding whose scalar products approximate the kernel.

The discussion then went from kernel approximations to embeddings as changes of representation. After introducing deterministic and linear random embeddings, we turned to quadratic transformations, which are at the core of the contributions developed later. We then introduced subspace-valued data, their representation through orthonormal bases and projectors, and their geometry through principal angles and Grassmannian distances. This made it possible to return to kernels and random features in the specific setting of the Grassmannian.

Finally, quantisation and binarisation were presented as further ways of reducing the cost of storing and processing embedded data. Concentration inequalities and covering arguments then provided the tools needed to move from identities in expectation to fixed-point and uniform approximation guarantees. Together, these ideas establish the progression followed in the later chapters, from quadratic estimation in the embedding domain to Grassmannian random features and their efficient implementation.

Signal processing in the quadratic embedding domain

Summary of contributions

Chap. 3 considered whether useful signal-processing operations can be performed directly from random quadratic embeddings, without reconstructing the original signal. The chapter focused on the debiased embedding ψ^-, obtained by taking differences of two independent random quadratic measurements. This debiasing step removes the nonzero mean of the original symmetric quadratic embedding and makes it possible to compare an embedded signal with the signed embedding of a fixed united vector.

The main contribution is the sign product embedding property established in Prop. 3.2. For a fixed unit vector $\boldsymbol{u}$, the quantity

$$\tfrac{\pi}{4m}\langle \operatorname{sign}(\psi^-(\boldsymbol{u})), \psi^-(\boldsymbol{x})\rangle$$

approximates $\langle \boldsymbol{u}, \boldsymbol{x}\rangle^2$ uniformly over all k-sparse signals $\boldsymbol{x}$, provided that

$$m \geq C\delta^{-2}k \log\left(\tfrac{n}{k\delta}\right).$$

The required embedding dimension therefore depends linearly on the intrinsic dimension k, up to logarithmic factors, rather than directly on the ambient dimension n.

The result was then extended from a single reference vector to a finite family of vectors. Cor. 3.2 shows that the same embedded signal can be compared simultaneously with a predetermined collection of unit vectors, with only a logarithmic dependence on the number of those vectors. This also gives a random-feature interpretation of the estimator. The signal $\boldsymbol{x}$ is represented by $\psi^-(\boldsymbol{x})$, while $\boldsymbol{u}$ is represented by $\operatorname{sign}(\psi^-(\boldsymbol{u}))$, and their scalar product approximates the quadratic kernel

$$\kappa^{\mathrm{S}}(\boldsymbol{x}, \boldsymbol{u}) = \langle \boldsymbol{x}, \boldsymbol{u}\rangle^2.$$

This differs from a classical symmetric random-feature construction, but it is sufficient when incoming signals are compared with a fixed collection of unit vectors.

Experimentally, we showed that the empirical distortion decreases with the embedding dimension as predicted by the theoretical analysis. The MNIST experiments showed that centroid classification can be performed in the embedding domain. The synthetic image-sequence experiment in Sec. 3.5.3 further demonstrated that localised information can be tracked from quadratic embeddings alone by comparing each embedded frame with a small collection of embedded vectors for each region of the image considered.

Perspectives

Several questions remain open. The first concerns the range of signal-processing procedures that can be performed in the embedding domain. The experiments considered relatively simple estimators and classifiers, but a natural continuation would be to develop statistically backed detection and estimation rules directly from the embedded measurements. Hypothesis tests, likelihood-ratio methods and maximum-likelihood estimators could provide better control of false-alarm or missed-detection probabilities.

A second direction concerns the class of quantities that can be estimated. The sign product construction is naturally adapted to the squared scalar product $\langle \boldsymbol{u}, \boldsymbol{x} \rangle^2$, which follows from the quadratic structure of the measurements. It remains to determine whether other transformations of the same embedding can approximate different functions of $\langle \boldsymbol{u}, \boldsymbol{x} \rangle$, or whether related random embeddings can preserve other quantities suited to tasks other than those presented.

Finally, the theoretical result is uniform over sparse signals $\boldsymbol{x}$, but it is proved for a fixed reference vector $\boldsymbol{u}$. An interesting extension would be to obtain a guarantee that is also uniform over a continuous set of vectors rather than a finite collection.

Random features for Grassmannian kernels

Summary of contributions

Chap. 4 moved from vector-valued signals to data represented by low-dimensional subspaces. Its goal is to replace exact Grassmannian kernel evaluations with scalar products between explicit random feature vectors, avoiding the construction and storage of a complete Gram matrix.

The chapter first considers dense linear random projections of the subspace projectors. These create a direct approximation of the projection kernel κ^{p}, but require storing and applying full random matrices

in $\mathbb{R}^{n\times n}$. We therefore introduced the rank-one embedding

$$\psi^{\mathrm{rop}}(\boldsymbol{U}) = \left(\boldsymbol{a}_i^\top \boldsymbol{U}\boldsymbol{U}^\top \boldsymbol{b}_i\right)_{i=1}^m.$$

Its empirical scalar products also approximate κ^{p}, but the resulting random variables are heavy-tailed and thus have poor concentration properties. This motivated the use of bounded non-linear transformations of the rank-one embedding.

The first construction applies the sign function and produces the binary embedding $\psi^{\pm}$. The corresponding empirical kernel

$$\widehat{\kappa}^{\pm}(\boldsymbol{U}, \boldsymbol{V}) = \tfrac{1}{m}\langle \psi^{\pm}(\boldsymbol{U}), \psi^{\pm}(\boldsymbol{V})\rangle$$

approximates a deterministic kernel $\kappa^{\pm}$. We showed that $\kappa^{\pm}$ is a well-defined positive semidefinite Grassmannian kernel, invariant to the choice of bases and to rotations of the subspaces. It therefore depends only on their principal angles despite having no simple closed form in general. In the special case $k = 1$, it reduces to an explicit function of the single principal angle.

Prop. 4.4 establishes a uniform approximation guarantee over all pairs of subspaces. More precisely, an embedding dimension satisfying

$$m \geq C\delta^{-2} nk \log\left(\tfrac{n^2}{k\delta^2}\right)$$

is sufficient to obtain a uniform error bounded by δ with high probability. The number of features therefore scales with the intrinsic dimension of $\mathcal{G}(k, n)$ up to logarithmic factors.

We then introduced the aggregated binary embedding ψ^{Σ}. This approximation sums N_Σ independent rank-one measurements before applying the sign function. Its expected kernel κ^{Σ} approaches the explicit kernel

$$\kappa_\infty^{\Sigma}(\boldsymbol{U}, \boldsymbol{V}) = \tfrac{2}{\pi} \arcsin\left(\tfrac{1}{k}\textstyle\sum_{i=1}^k \cos^2\theta_i\right),$$

where $\theta_1, \ldots, \theta_k$ are the principal angles between $\boldsymbol{U}$ and $\boldsymbol{V}$. Prop. 4.9 bounds the difference between κ^{Σ} and κ_∞^{Σ} by $C/\sqrt{N_\Sigma}$. The construction therefore gives an interpretable binary kernel, although it requires more rank-one measurements for every feature.

The final construction applies the complex exponential to the rank-one embedding and produces the features ψ°. Their expected kernel has the explicit expression

$$\kappa^{\circ}(\boldsymbol{U}, \boldsymbol{V}) = \textstyle\prod_{i=1}^k \left(1 + \omega^2 \sin^2\theta_i\right)^{-1}.$$

The parameter ω controls the geometry of the kernel. For small ω, κ° is close to a radial basis function kernel in the projection distance. Larger

values make it more sensitive to small changes in the principal angles. Prop. 4.12 gives a uniform approximation guarantee with a number of features proportional to nk up to logarithmic factors.

The experiments first confirmed the geometry of the proposed kernels and showed that their empirical approximation errors decrease approximately as $\rho^{-1/2}$, where $\rho = m/(nk)$ represents the oversampling ratio. This shows that the embedding dimension is directly linked to the intrinsic dimension of the subspaces nk. The comparison with dense projections also showed that rank-one projections approximate the same projection kernel with substantially lower memory and computational costs.

The ETH-80 experiments demonstrated that the raw, binary and periodic features can all be used for subspace classification, with useful accuracy already obtained for embedding dimensions well below the size nk of the original basis representation. In the object classification setting, small embeddings provide faster but less accurate alternatives to the exact kernels, while larger embeddings approach their accuracy. The experiments therefore show that Grassmannian kernels can be approximated effectively with explicit rank-one random features, while Chap. 5 investigates how these features can be computed more efficiently.

Perspectives

The aggregated binary construction is only partially analysed. Prop. 4.9 controls the difference between its expected kernel κ^{Σ} and the limiting kernel κ^{Σ}_{∞}, but no uniform concentration result was developed for the empirical kernel $\widehat{\kappa}^{\Sigma}$. A complete analysis should both control the error caused by the finite aggregation parameter N_{Σ} and the error caused by the finite embedding dimension m. This would clarify the trade-off between approximation quality and the total number mN_{Σ} of rank-one measurements.

A second direction concerns subspaces of different dimensions. The results in Chap. 4 compare elements of a fixed Grassmannian $\mathcal{G}(k, n)$. Some applications may instead require comparisons between subspaces whose dimensions differ, including the comparison of a vector (a one-dimensional subspace) with a higher-dimensional subspace.

The sign, aggregated sign and periodic transformations represent only a small number of the bounded functions that could be applied to rank-one measurements. Other transformations would potentially induce other Grassmannian kernels with different dependence on the principal angles. It is not yet clear whether other bounded transformations of the rank-one embedding would lead to useful Grassmannian

kernels. A broader direction would be to establish a general result describing which bounded functions produce valid kernels and under which conditions their empirical approximations concentrate uniformly over the Grassmannian.

Another possible extension would be to consider kernels defined on covariance matrices instead of projectors. As we saw, projection matrices representing subspaces form a class of positive semidefinite matrices, just like covariance matrices. However, the latter are not constrained to be idempotent or to have fixed eigenvalues. Rank-one measurements remain natural in this setting, since for a covariance matrix $\boldsymbol{\Sigma}$ they take the form $\langle \boldsymbol{a}\boldsymbol{a}^\top, \boldsymbol{\Sigma} \rangle_{\mathrm{F}} = \boldsymbol{a}^\top \boldsymbol{\Sigma} \boldsymbol{a}$. This suggests that random features similar to those developed here could be constructed for covariance matrices. The main difficulty is that kernels on covariance matrices may depend on geometric properties that are more complex than those of the Grassmannian considered in this work. Extending the present approach would therefore require determining which covariance matrix kernels can be recovered or approximated from rank-one measurements and how the corresponding geometric quantities are preserved.

Fast computation of random embeddings

Summary of contributions

Chap. 5 demonstrated two ways of reducing the cost of the random embeddings introduced in the previous chapters. The first replaces independent Gaussian vectors with structured Hadamard-diagonal transformations. The second uses an Optical Processing Unit to compute quadratic random measurements physically.

The structured embedding uses products of random diagonal sign matrices and Walsh-Hadamard transforms to generate the vectors of ψ_{st}. This reduces the cost of the rank-one embeddings both in terms of computation and in terms of memory. The empirical study in Fig. 5.2 shows that three blocks already produce coefficients whose distribution is close to Gaussian.

The ETH-80 experiments confirm that the structured construction preserves the behaviour of the Gaussian embeddings from Chap. 4. In the object classification setting, the raw, binary and periodic structured embeddings all reach accuracies above 91% at $\rho = 0.20$, remaining within 2% of their Gaussian counterparts. Their computation is approximately 20 times faster than that of the embeddings of Chap. 4. The structured embeddings therefore provide a practical way to compute the Grassmannian random features at a substantially lower cost.

The second part of the chapter considers Optical Processing Units. Their physical scattering mechanism produces complex quadratic random measurements, which can be debiased by subtracting two halves of the output. The result corresponds to an average of independent versions of the quadratic embeddings introduced in the previous chapters.

Since the OPU only accepts binary inputs, we introduced an encoding procedure in Algo. 1 for vectors quantised with L levels. The quantised vector is decomposed into ternary bitplanes, which are then represented through binary vectors compatible with the device. Prop. 5.1 shows that the quadratic embedding of the quantised vector can be reconstructed exactly in the ideal noiseless model using

$$N_{\mathrm{OPU}}(L) = \frac{3B(B+1)}{2}$$

OPU calls, where $B = \lceil \log_2 L \rceil$. The number of calls thus scales as $\mathcal{O}((\log_2 L)^2)$.

We also studied how physical noise is amplified by the decoder of Algo. 2. Prop. 5.2 gives the variance of the final decoded noise as a function of the bitplane weights. Lem. 5.1 then shows that this variance remains bounded when the quantisation step decreases proportionally to $1/L$. The empirical measurements in Fig. 5.6 follow the same trend, with the largest increase occurring between $L = 2$ and $L = 4$.

The numerical OPU experiments confirm that the encoder and decoder behave exactly as intended in the ideal model. They also show that finer quantisation can substantially improve the more difficult object classification problem. The physical OPU produces a different result. Its accuracy decreases for larger values of L in both classification settings.

The analysis of the density of the vectors sent to the OPU gives a possible explanation for this unexpected result. The physical OPU behaves closest to its ideal model when the binary input density lies approximately between 0.2 and 0.8. However, the encoder produces an increasing proportion of influential vectors outside this range as L grows, as shown in Fig. 5.12. This effect is combined with the increase in decoded noise caused by the larger number of optical measurements.

The optical results therefore confirm the mathematical encoding and decoding construction, but do not yet show a competitive implementation on the physical device considered here. The main limitation does not come from an algebraic error in the method, but from the interaction between the encoder and the OPU.

Perspectives

The structured embeddings are currently supported mainly by complexity arguments and experimental results. The concentration guarantees established in Chap. 4 were proved for independent Gaussian probing vectors and were assumed to remain valid for the structured construction. A first theoretical direction would be to establish direct approximation guarantees for the Hadamard-diagonal embeddings and determine how many blocks are needed.

Finally, the optical encoding should also be redesigned with the physical device in mind. The current procedure produces many binary vectors that are either too sparse or too dense, and it requires several OPU calls for every quantised basis vector. A better encoder should keep the transmitted vectors within a suitable density range while trying to reduce the number of calls.

6.2 Final remarks

The common thread throughout this manuscript is that random measurements can be useful even when they do not allow us to reconstruct the object that produced them. In many signal processing and learning problems, the final goal is not to recover a vector, an image, or a subspace in full detail but to compare, detect, classify, or estimate a specific quantity. When the embedding preserves that quantity with good accuracy, the embedded representation can be enough.

This is the role played here by quadratic and rank-one measurements. Their interest does not come from preserving everything. It comes from the fact that they preserve certain interactions. In the vector setting, debiased quadratic measurements allow us to estimate squared scalar products with fixed patterns. In the Grassmannian setting, rank-one measurements allow us to approximate kernels depending on the geometry of subspaces. In both cases, bounded transformations such as signs or complex exponentials make the embeddings more stable and more suitable for random feature constructions.

This also gives a useful warning. Random quadratic embeddings should not be treated as black-box compression mechanisms that automatically retain all useful information. They are only meaningful once the task to be performed has been identified. The right question is therefore not whether a compressed representation resembles an original object, but whether it preserves the comparisons and estimators required by a task. This task-driven reading is what makes the constructions studied in this manuscript useful, and also what defines their limits.

The results presented here do not close the subject. They open several paths, from sharper guarantees to faster implementations and optical experiments. More importantly, they show that a change of representation can change the way we approach a problem. We do not always need to reconstruct the original object in order to use it. Sometimes it is enough to preserve the right quantity to reach the right decisions. In that sense, quadratic and rank-one random embeddings offer a way of turning difficult high-dimensional objects into lighter representations that can still carry the information needed to act.

List of acronyms

AI ..... Artificial intelligence.

CLAFIC ..... Class-featuring information compression.

CS ..... Compressed sensing.

DCT ..... Discrete cosine transform.

DMD ..... Digital micromirror device.

FFHT ..... Fast Fast Hadamard Transform.

HDHD ..... Hadamard-diagonal transform.

IT ..... Information technology.

JL ..... Johnson-Lindenstrauss.

LDA ..... Linear discriminant analysis.

LLM ..... Large language model.

LSH ..... Locality-sensitive hashing.

MNIST ..... Modified National Institute of Standards and Technology database.

OPU ..... Optical processing unit.

PCA ..... Principal component analysis.

PSD ..... Positive semidefinite.

Q-Q ..... Quantile-quantile.

QR ..... Orthogonal-triangular factorisation.

RAM ..... Random-access memory.

RBF ..... Radial basis function.

RIP ..... Restricted isometry property.

ROM ..... Read-only memory.

ROP ..... Rank-one projection.
SP ..... Signal processing.
SPD ..... Symmetric positive-definite.
SPE ..... Sign product embedding.
SVD ..... Singular value decomposition.
SVM ..... Support vector machine.
t-SNE ..... t-distributed stochastic neighbour embedding.
UMAP .. Uniform Manifold Approximation and Projection.

Bibliography

[1] D. Achlioptas. "Database-friendly random projections: Johnson-Lindenstrauss with binary coins". In: *J. Comput. Syst. Sci.* 66.4 (2003), pp. 671–687. DOI: 10.1016/S0022-0000(03)00025-4.

[2] R. Agrawal, T. Campbell, J. H. Huggins, and T. Broderick. "Data-dependent compression of random features for large-scale kernel approximation". In: *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*. Vol. 89. Proceedings of Machine Learning Research. PMLR, 2019, pp. 1822–1831.

[3] A. Ai, A. Lapanowski, Y. Plan, and R. Vershynin. "One-bit compressed sensing with non-Gaussian measurements". In: *Linear Algebra Appl.* 441 (2014), pp. 222–239. DOI: 10.1016/j.laa.2013.04.002.

[4] N. Ailon and B. Chazelle. "The Fast Johnson–Lindenstrauss Transform and Approximate Nearest Neighbors". In: *SIAM Journal on Computing* 39.1 (2009), pp. 302–322. DOI: 10.1137/060673096.

[5] N. Ailon and E. Liberty. "Fast Dimension Reduction Using Rademacher Series on Dual BCH Codes". In: *Discrete and Computational Geometry* 42.4 (2009), pp. 615–630. DOI: 10.1007/s00454âĂŠ008âĂŠ9110âĂŠx.

[6] M. A. Aizerman, E. M. Braverman, and L. I. Rozonoer. "Theoretical foundations of the potential function method in pattern recognition learning". In: *Autom. Remote Control* 25.6 (1964). English translation from Avtomatika i Telemekhanika, vol. 25, no. 6, pp. 917–936, pp. 821–837.

[7] A. Andoni, P. Indyk, T. Laarhoven, I. Razenshteyn, and L. Schmidt. "Practical and Optimal LSH for Angular Distance". In: *Advances in Neural Information Processing Systems*. Vol. 28. Full version available at arXiv:1509.02897. 2015.

[8] N. Aronszajn. "Theory of reproducing kernels". In: *Trans. Amer. Math. Soc.* 68.3 (1950), pp. 337–404. DOI: 10.1090/S0002-9947-1950-0051437-7.

[9] F. Bach. *Learning theory from first principles*. MIT press, 2024.

[10] F. Bach and M. Jordan. "Predictive low-rank decomposition for kernel methods". In: *Proceedings of the 22nd International Conference on Machine Learning*. ICML '05. Bonn, Germany: Association for Computing Machinery, 2005, pp. 33–40.

[11] R. Balan, B. Bodmann, P. Casazza, and D. Edidin. "Fast algorithms for signal reconstruction without phase". In: *Wavelets XII*. Vol. 6701. Proc. SPIE. 2007, pp. 670111920–670111932.

[12] R. Balan, B. Bodmann, P. Casazza, and D. Edidin. "Painless reconstruction from magnitudes of frame coefficients". In: *J. Fourier Anal. Appl.* 15 (2009), pp. 488–501.

[13] R. Balan, P. Casazza, and D. Edidin. "On signal reconstruction without noisy phase". In: *Appl. Comput. Harmon. Anal.* 20 (2006), pp. 345–356.

[14] A. S. Bandeira, M. Fickus, D. G. Mixon, and P. Wong. "The road to deterministic matrices with the restricted isometry property". In: *J. Fourier Anal. Appl.* 19.6 (2013), pp. 1123–1149. DOI: 10.1007/s00041-013-9293-2.

[15] R. Baraniuk, M. Davenport, R. DeVore, and M. Wakin. "A Simple Proof of the Restricted Isometry Property for Random Matrices". In: *Constr. Approx.* 28.3 (2008), pp. 253–263.

[16] R. G. Baraniuk. "Compressive sensing". In: *IEEE Signal Process. Mag.* 24.4 (2007), pp. 118–121. DOI: 10.1109/MSP.2007.4286571.

[17] R. Basri, T. Hassner, and L. Zelnik-Manor. "Approximate Nearest Subspace Search". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33.2 (2011), pp. 266–278. DOI: 10.1109/TPAMI.2010.110.

[18] R. Basri and D. W. Jacobs. "Lambertian Reflectance and Linear Subspaces". In: *IEEE Trans. Pattern Anal. Mach. Intell.* 25.2 (2003), pp. 218–233. DOI: 10.1109/TPAMI.2003.1177153.

[19] Y. Bengio, A. Courville, and P. Vincent. "Representation Learning: A Review and New Perspectives". In: *IEEE Trans. Pattern Anal. Mach. Intell.* 35.8 (2013), pp. 1798–1828. DOI: 10.1109/TPAMI.2013.50.

[20] V. Bentkus. "A Lyapunov-type Bound in Rd". In: *Theory of Probability and Its Applications* 49 (2005), pp. 311–322.

[21] E. Bingham and H. Mannila. "Random projection in dimensionality reduction: Applications to image and text data". In: *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. San Francisco, CA, USA: Association for Computing Machinery, 2001, pp. 245–250. ISBN: 978-1-58113-391-2. DOI: 10.1145/502512.502546.

[22] Å. Björck and G. H. Golub. "Numerical methods for computing angles between linear subspaces". In: *Math. Comp.* 27.123 (1973), pp. 579–594. DOI: 10.1090/S0025-5718-1973-0348991-3.

[23] M. Bojarski, A. Choromanska, K. Choromanski, F. Fagan, C. Gouy-Pailler, A. Morvan, N. Sakr, T. Sarlos, and J. Atif. "Structured Adaptive and Random Spinners for Fast Machine Learning Computations". In: *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*. AISTATS 2017. Fort Lauderdale, Florida, USA: PMLR, 2017, pp. 1020–1029.

[24] H. Bong and A. K. Kuchibhotla. "Tight Concentration Inequality for Sub-Weibull Random Variables with Generalized Bernstein Orlicz Norms". In: *arXiv preprint* (2023). arXiv: 2302.03850.

[25] Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik. "A Training Algorithm for Optimal Margin Classifiers". In: *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*. 1992, pp. 144–152.

[26] P. T. Boufounos, L. Jacques, F. Krahmer, and R. Saab. "Quantization and compressive sensing". In: *Compressed Sensing and its Applications: MATHEON Workshop 2013*. Ed. by H. Boche, R. Calderbank, G. Kutyniok, and J. Vybíral. Applied and Numerical Harmonic Analysis. Cham: Birkhäuser, 2015, pp. 193–237. ISBN: 978-3-319-16042-9. DOI: 10.1007/978-3-319-16042-9_7.

[27] N. Bourbaki. *Éléments d'histoire des mathématiques*. Berlin: Springer, 2007. ISBN: 978-3-540-33938-0.

[28] C. Brossollet, A. Cappelli, I. Carron, C. Chaintoutis, A. Chatelain, L. Daudet, S. Gigan, D. Hesslow, F. Krzakala, J. Launay, S. Mokaadi, F. Moreau, K. Muller, R. Ohana, G. Pariente, I. Poli, and E. Tommasone. "LightOn Optical Processing Unit: Scaling-up AI and HPC with a Non von Neumann co-processor". In: *2021 IEEE Hot Chips 33 Symposium (HCS)*. Also available as arXiv:2107.11814 [cs.AR]. 2021, pp. 1–11. DOI: 10.1109/HCS52781.2021.9567166.

[29] T. Cai and A. Zhang. "ROP: Matrix recovery via rank-one projections". In: *The annals of statistics* 43.1 (2015). ISSN: 0090-5364.

[30] J. Calbert, R. Delogne, and C. Monnoyer. *LEPL1101 - Algèbre linéaire*. Open Educational Resource, Université catholique de Louvain. Course material, version 0.0.42, academic year 2023–2024; professor: Verleysen, M. and Wertz, V. 2023.

[31] R. Calderbank, S. Jafarpour, and R. Schapire. "Compressed Learning: Universal Sparse Dimensionality Reduction and Learning in the Measurement Domain". Unpublished manuscript. 2009.

[32] E. Candès, J. Romberg, and T. Tao. "Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information". In: *IEEE Trans. Inf. Theory* 52.2 (2006), pp. 489–509.

[33] E. J. Candès. "Compressive sampling". In: *Proceedings of the International Congress of Mathematicians, Madrid, August 22–30, 2006, Volume III*. EMS Press lists online publication date as 15 May 2007. Z"urich: European Mathematical Society, 2006, pp. 1433–1452. DOI: 10.4171/022-3/69.

[34] E. J. Candès. "The restricted isometry property and its implications for compressed sensing". In: *C. R. Math.* 346.9-10 (2008), pp. 589–592. DOI: 10.1016/j.crma.2008.03.014.

[35] E. J. Candès and Y. Plan. "Tight Oracle Bounds for Low-Rank Matrix Recovery from a Minimal Number of Random Measurements". In: *IEEE Trans. Inf. Theory* 57.4 (2011), pp. 2342–2359. DOI: 10.1109/TIT.2011.2111771.

[36] E. J. Candès, T. Strohmer, and V. Voroninski. "PhaseLift: Exact and Stable Signal Recovery from Magnitude Measurements via Convex Programming". In: *Commun. Pure Appl. Math.* 66.8 (2013). doi:10.1002/cpa.21432, pp. 1241–1274.

[37] E. J. Candès and T. Tao. "Decoding by linear programming". In: *IEEE Trans. Inf. Theory* 51.12 (2005), pp. 4203–4215. DOI: 10.1109/TIT.2005.858979.

[38] M. S. Charikar. "Similarity estimation techniques from rounding algorithms". In: *Proceedings of the 34th Annual ACM Symposium on Theory of Computing*. doi: 10.1145/509907.509965. 2002, pp. 380–388.

[39] Kai-Xuan Chen, Jie-Yi Ren, Xiao-Jun Wu, and Josef Kittler. "Covariance Descriptors on a Gaussian Manifold and their Application to Image Set Classification". In: *Pattern Recognition* (2020), p. 107463.

[40] Y. Chen, Y. Chi, and A. J. Goldsmith. "Estimation of simultaneously structured covariance matrices from quadratic measurements". In: *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Florence, Italy: IEEE, 2014, pp. 7669–7673. ISBN: 978-1-4799-2893-4. DOI: 10.1109/ICASSP.2014.6855092.

[41] Y. Chen, Y. Chi, and A. J. Goldsmith. "Exact and Stable Covariance Estimation from Quadratic Sampling via Convex Programming". In: *IEEE Trans. Inf. Theory* 61.7 (2015), pp. 4034–4059. DOI: 10.1109/TIT.2015.2429594.

[42] Y. Chi and H. Fu. "Subspace Learning From Bits". In: *IEEE Trans. Signal Process.* 65.8 (2017), pp. 2009–2022.

[43] K. Choromanski, F. Fagan, C. Gouy-Pailler, A. Morvan, T. Sarlos, and J. Atif. "TripleSpin: A Generic Compact Paradigm for Fast Machine Learning Computations". In: *arXiv:1605.09046* (2016). Preprint. DOI: 10.48550/arXiv.1605.09046.

[44] K. M. Choromanski, H. Chen, H. Lin, Y. Ma, A. Sehanobish, D. Jain, M. S. Ryoo, J. Varley, A. Zeng, V. Likhosherstov, D. Kalashnikov, V. Sindhwani, and A. Weller. "Hybrid random features". In: *International Conference on Learning Representations*. 2022.

[45] K. M. Choromanski, V. Likhosherstov, D. M. Dohan, X. Song, A. Gane, T. Sarlos, P. Hawkins, J. Q. Davis, A. Mohiuddin, L. Kaiser, D. Belanger, L. Colwell, and A. Weller. "Rethinking attention with Performers". In: *International Conference on Learning Representations*. 2021.

[46] L. Clissa, M. Lassnig, and L. Rinaldi. "How big is Big Data? A comprehensive survey of data production, storage, and streaming in science and industry". In: *Front. Big Data* 6 (2023). DOI: 10.3389/fdata.2023.1271639.

[47] J. H. Conway, R. H. Hardin, and N. J. A. Sloane. "Packing Lines, Planes, etc.: Packings in Grassmannian Spaces". In: *Experimental Mathematics* 5.2 (1996), pp. 139–159. DOI: 10.1080/10586458.1996.10504585.

[48] I. Daubechies. *Ten Lectures on Wavelets*. Vol. 61. CBMS-NSF Regional Conference Series in Applied Mathematics. Philadelphia, PA: SIAM, 1992. ISBN: 9780898712742. DOI: 10.1137/1.9781611970104.

[49] M. A. Davenport, P. T. Boufounos, M. B. Wakin, and R. G. Baraniuk. “Signal Processing with Compressive Measurements”. In: *IEEE J. Sel. Top. Signal Process.* 4.2 (2010), pp. 445–460.

[50] M. A. Davenport, Y. Plan, E. van den Berg, and M. Wootters. “1-Bit matrix completion”. In: *Information and Inference: A Journal of the IMA* 3.3 (2014), pp. 189–223. DOI: 10.1093/imaiai/iau006.

[51] R. Delogne and L. Jacques. “Quadratic polynomial kernel approximation with asymmetric embeddings”. In: *International Workshop on Deep Learning and Kernel Machines (DEEPK)*. 2024.

[52] R. Delogne and L. Jacques. “Random features for Grassmannian kernel approximation with bounded rank-one projections”. In: *Under review for Transactions on Machine Learning Research (TMLR)* (2026).

[53] R. Delogne and L. Jacques. “Random Features for Grassmannian Kernels”. In: *2025 IEEE Statistical Signal Processing Workshop (SSP)*. 2025, pp. 221–224.

[54] R. Delogne, V. Schellekens, L. Daudet, and L. Jacques. “ROP inception: Signal estimation with quadratic random sketching”. In: *Proceedings of the 30th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)*. 2022.

[55] R. Delogne, V. Schellekens, L. Daudet, and L. Jacques. “Signal processing after quadratic random sketching with optical units”. In: *International Symposium on Computational Sensing (ISCS)*. 2023.

[56] R. Delogne, V. Schellekens, L. Daudet, and L. Jacques. “Signal processing with optical quadratic random sketches”. In: *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2023, pp. 1–5.

[57] L. Deng. “The MNIST Database of Handwritten Digit Images for Machine Learning Research [Best of the Web]”. In: *IEEE Signal Process. Mag.* 29.6 (2012). doi:10.1109/MSP.2012.2211477, pp. 141–142.

[58] J. Dieudonné. “The Tragedy of Grassmann”. In: *Séminaire de Philosophie et Mathématiques* 2 (1979), pp. 1–14.

[59] I. Dokmanić, R. Parhizkar, J. Ranieri, and M. Vetterli. “Euclidean distance matrices: Essential theory, algorithms, and applications”. In: *IEEE Signal Process. Mag.* 32.6 (2015), pp. 12–30. DOI: 10.1109/MSP.2015.2398954. arXiv: 1502.07541 [cs.OH].

[60] D. Donoho. "Compressed sensing". In: *IEEE Trans. Inf. Theory* 52.4 (2006), pp. 1289–1306.

[61] D. L. Donoho. "High-dimensional data analysis: The curses and blessings of dimensionality". In: *Mathematical Challenges of the 21st Century*. AMS National Meeting, Los Angeles, CA, USA, August 6–12, 2000. American Mathematical Society, 2000, pp. 1–32.

[62] P. Drineas and M. W. Mahoney. "Approximating a Gram Matrix for Improved Kernel-Based Learning". In: *Learning Theory: 18th Annual Conference on Computational Learning Theory (COLT 2005), Lecture Notes in Computer Science, Volume 3559*. Ed. by P. Auer and R. Meir. Springer, Berlin Heidelberg, 2005, pp. 323–337.

[63] A. Edelman, T. A. Arias, and S. T. Smith. "The geometry of algorithms with orthogonality constraints". In: *SIAM J. Matrix Anal. Appl.* 20.2 (1998), pp. 303–353. DOI: 10.1137/S0895479895290954.

[64] J. Fan and R. Li. "Statistical challenges with high dimensionality: Feature selection in knowledge discovery". In: *Proceedings of the International Congress of Mathematicians, Madrid, August 22–30, 2006, Volume III*. Zürich: European Mathematical Society, 2006, pp. 595–622. DOI: 10.4171/022-3/31.

[65] C. Feng, Q. Hu, and S. Liao. "Random feature mapping with signed circulant matrix projection". In: *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence*. AAAI Press, 2015, pp. 3490–3496.

[66] I. Florescu. *Probability and stochastic processes*. John Wiley & Sons, 2014.

[67] S. Foucart. "Flavors of Compressive Sensing". In: *Approximation Theory XV: San Antonio 2016*. Ed. by G. Fasshauer and L. Schumaker. Vol. 201. Springer Proceedings in Mathematics and Statistics. Springer, 2017, pp. 61–104.

[68] S. Foucart and H. Rauhut. *A Mathematical Introduction to Compressive Sensing*. 1st ed. Applied and Numerical Harmonic Analysis. XVIII, 625 pp. New York, NY: Birkhäuser, 2013. ISBN: 978-0-8176-4948-7. DOI: 10.1007/978-0-8176-4948-7.

[69] F. L. Gewers, G. R. Ferreira, H. F. de Arruda, F. N. Silva, C. H. Comin, D. R. Amancio, and L. da F. Costa. "Principal component analysis: A natural approach to data exploration". In: *ACM Com-*

put. Surv. 54.4 (2021), 70:1–70:34. DOI: 10.1145/3447755. arXiv: 1804.02502 [cs.CE].

[70] G. H. Golub and C. F. Van Loan. *Matrix computations*. 3rd ed. Johns Hopkins Studies in the Mathematical Sciences. Baltimore, MD: Johns Hopkins University Press, 1996. ISBN: 978-0-8018-5414-9.

[71] H. Grassmann. *Die lineale Ausdehnungslehre, ein neuer Zweig der Mathematik: Dargestellt und durch Anwendungen auf die übrigen Zweige der Mathematik, wie auch auf die Statik, Mechanik, die Lehre vom Magnetismus und die Krystallonomie erläutert*. Leipzig, Germany: O. Wigand, 1844.

[72] R. M. Gray and D. L. Neuhoff. "Quantization". In: *IEEE Trans. Inf. Theory* 44.6 (1998), pp. 2325–2383. DOI: 10.1109/18.720541.

[73] D. Gross, Y. Liu, S. T. Flammia, S. Becker, and J. Eisert. "Quantum-state tomography via compressed sensing". In: *Phys. Rev. Lett.* 105.15 (2010), p. 150401.

[74] J. Hamm and D. D. Lee. "Grassmann discriminant analysis: a unifying view on subspace-based learning". In: *Proceedings of the 25th international conference on Machine learning*. 2008, pp. 376–383.

[75] M. T. Harandi, M. Salzmann, S. Jayasumana, R. Hartley, and H. Li. "Expanding the Family of Grassmannian Kernels: An Embedding Perspective". In: (2014). Available on arXiv. DOI: 10.48550/arXiv.1407.1123. eprint: arXiv:1407.1123.

[76] N. Heavner, C. Chen, A. Gopal, and P.-G. Martinsson. "Efficient algorithms for computing rank-revealing factorizations on a GPU". In: *Numer. Linear Algebra Appl.* 30.6 (2023). doi: 10.1002/nla.2515, e2515.

[77] D. Hesslow, A. Cappelli, I. Carron, L. Daudet, R. Lafargue, K. Müller, R. Ohana, G. Pariente, and I. Poli. "Photonic co-processors in HPC: Using LightOn OPUs for randomized numerical linear algebra". In: *2021 IEEE Hot Chips 33 Symposium (HCS)*. IEEE, 2021, pp. 1–9. DOI: 10.1109/HCS52781.2021.9566948. arXiv: 2104.14429 [stat.ML].

[78] M. Hilbert and P. López. "The world's technological capacity to store, communicate, and compute information". In: *Science* 332.6025 (2011), pp. 60–65. DOI: 10.1126/science.1200970.

[79] D. Hill and M. Slawski. *Linear regression from 1-bit quantized data*. arXiv preprint. 2026. DOI: 10.48550/arXiv.2603.28989. arXiv: 2603.28989 [math.ST].

[80] N. Hurt. *Phase Retrieval and Zero Crossings*. Norwell, MA: Kluwer Academic Publishers, 1989.

[81] IEEE. *IEEE Standard for Floating-Point Arithmetic*. IEEE Std 754-2019 (Revision of IEEE Std 754-2008). 2019. DOI: 10.1109/IEEESTD.2019.8766229.

[82] P. Indyk and R. Motwani. "Approximate Nearest Neighbors: Towards Removing the Curse of Dimensionality". In: *Proc. 30th Annu. ACM Symp. Theory Comput. (STOC)*. New York, NY, USA: ACM, 1998, pp. 604–613.

[83] International Energy Agency. *Energy and AI*. Report. IEA report on electricity demand from artificial intelligence and data centres. Paris: International Energy Agency, 2025.

[84] Ji J., Li J., Tian Q., Yan S., and Zhang B. "Angular-Similarity-Preserving Binary Signatures for Linear Subspaces". In: *IEEE Transactions on Image Processing* 24.11 (2015), pp. 4372–4380. DOI: 10.1109/TIP.2015.2451173.

[85] L. Jacques, K. Degraux, and C. De Vleeschouwer. "Quantized Iterative Hard Thresholding: Bridging 1-bit and High-Resolution Quantized Compressed Sensing". In: *Proceedings of the International Conference on Sampling Theory and Applications (SampTA)*. IEEE. 2013, pp. 105–108.

[86] L. Jacques, J. N. Laska, P. T. Boufounos, and R. G. Baraniuk. "Robust 1-Bit Compressive Sensing via Binary Stable Embeddings of Sparse Vectors". In: *IEEE Transactions on Information Theory* 59.4 (2013), pp. 2082–2102.

[87] A. Janjeva, J. Harris, S. Mercer, B. Sum, J. Lee, and J. Park. *Semiconductor supply chains, AI and economic statecraft*. Research report. Centre for Emerging Technology and Security, The Alan Turing Institute, 2024.

[88] S. Jayasumana, R. Hartley, M. Salzmann, H. Li, and M. Harandi. "Kernel Methods on Riemannian Manifolds with Gaussian RBF Kernels". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37.12 (2015), pp. 2464–2477. DOI: 10.1109/TPAMI.2015.2414422.

[89] Y. Jiang and Y. Chi. "Covariance tracking from sketches of rapid data streams". In: *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. South Brisbane, QLD, Australia: IEEE, 2015, pp. 5470–5474. ISBN: 978-1-4673-6997-8. DOI: 10.1109/ICASSP.2015.7179017.

[90] M. S. Johnson and D. R. Giller. *MIT's role in Project Apollo, Volume 5: Software.* Technical report NASA-CR-140339. Cambridge, MA: Charles Stark Draper Laboratory, Massachusetts Institute of Technology, Mar. 1971.

[91] W. B. Johnson and J. Lindenstrauss. "Extensions of Lipschitz mappings into a Hilbert space". In: *Conference in Modern Analysis and Probability (New Haven, Conn., 1982)*. Vol. 26. Contemporary Mathematics. American Mathematical Society, 1984, pp. 189–206.

[92] I. M. Johnstone and D. M. Titterington. "Statistical challenges of high-dimensional data". In: *Philos. Trans. R. Soc. A* 367.1906 (2009), pp. 4237–4253. DOI: 10.1098/rsta.2009.0159.

[93] M. Klibanov, P. Sacks, and A. Tikhonravov. "The phase retrieval problem". In: *Inverse Problems* 11 (1995), pp. 1–28.

[94] A. V. Knyazev and P. Zhu. "Principal angles between subspaces and their tangents". In: *arXiv preprint arXiv:1209.0523* (2012).

[95] B. Laurent and P. Massart. "Adaptive estimation of a quadratic functional by model selection". In: *The Annals of Statistics* 28.5 (2000), pp. 1302–1338. DOI: 10.1214/aos/1015957395.

[96] Q. Le, T. Sarlos, and A. Smola. "Fastfood: Approximating Kernel Expansions in Log-linear Time". In: *Proc. 30th Int. Conf. Mach. Learn. (ICML)*. 2013, pp. 244–252.

[97] O. Leblanc, C. S. Chu, L. Jacques, and Y. Wiaux. "MROP: Modulated rank-one projections for compressive radio interferometric imaging". In: *Mon. Not. R. Astron. Soc.* 543.2 (2025), pp. 1727–1747. DOI: 10.1093/mnras/staf1510.

[98] O. Leblanc, M. Hofer, S. Sivankutty, H. Rigneault, and L. Jacques. "An interferometric view of speckle imaging". In: *Workshop on Low-Rank Models and Applications (LRMA 22)*. Poster. Mons, Belgium, 2022.

[99] O. Leblanc, M. Hofer, S. Sivankutty, H. Rigneault, and L. Jacques. "Interferometric lensless imaging: Rank-one projections of image frequencies with speckle illuminations". In: *IEEE Trans. Comput. Imaging* 10 (2024), pp. 208–222. DOI: 10.1109/TCI.2024.3359178.

[100] O. Leblanc, Y. Wiaux, and L. Jacques. "Compressive radio-interferometric sensing with random beamforming as rank-one signal covariance projections". In: *IEEE Trans. Comput. Imaging* 11 (2025), pp. 1229–1242. DOI: 10.1109/TCI.2025.3587449.

[101] Y. LeCun, Y. Bengio, and G. Hinton. "Deep Learning". In: *Nature* 521.7553 (2015), pp. 436–444. DOI: 10.1038/nature14539.

[102] J. A. Lee and M. Verleysen, eds. *Nonlinear dimensionality reduction*. 1st ed. Information Science and Statistics. XVII, 309 pp. New York, NY: Springer, 2007. ISBN: 978-0-387-39351-3. DOI: 10.1007/978-0-387-39351-3.

[103] Bastian Leibe and Bernt Schiele. "Analyzing appearance and contour based methods for object categorization". In: *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.* Vol. 2. IEEE. 2003, pp. II–409.

[104] D. Lemire. "Number parsing at a gigabyte per second". In: *Softw. Pract. Exp.* 51.8 (2021), pp. 1700–1727. DOI: 10.1002/spe.2984.

[105] G. Li and Y. Gu. "Restricted Isometry Property of Gaussian Random Projection for Finite Set of Subspaces". In: *IEEE Trans. Signal Process.* 66.7 (2018), pp. 1705–1720.

[106] V. Likhosherstov, K. M. Choromanski, K. A. Dubey, F. Liu, T. Sarlos, and A. Weller. "Dense-exponential random features: sharp positive estimators of the Gaussian kernel". In: *Advances in Neural Information Processing Systems*. Vol. 36. doi: 10.52202/075280-0046. 2023, pp. 957–985.

[107] F. Liu, X. Huang, Y. Chen, and J. A. K. Suykens. "Random Features for Kernel Approximation: A Survey on Algorithms, Theory, and Beyond". In: *IEEE Trans. Pattern Anal. Mach. Intell.* 44.10 (2022), pp. 6674–6695.

[108] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma. "Robust Recovery of Subspace Structures by Low-Rank Representation". In: *IEEE Trans. Pattern Anal. Mach. Intell.* 35.1 (2013), pp. 171–184.

[109] Y. Liu, J. Cao, C. Liu, K. Ding, and L. Jin. "Datasets for large language models: A comprehensive survey". In: *Artif. Intell. Rev.* 58.12 (2025), p. 403. DOI: 10.1007/s10462-025-11403-7. arXiv: 2402.18041 [cs.CL].

[110] A. M. Maiden and J. M. Rodenburg. "An improved ptychographical phase retrieval algorithm for diffractive imaging". In: *Ultramicroscopy* 109.10 (2009). doi: 10.1016/j.ultramic.2009.05.012, pp. 1256–1262.

[111] A. M. Maiden, J. M. Rodenburg, and M. J. Humphry. "Optical ptychography: a practical implementation with useful resolution". In: *Opt. Lett.* 35.15 (2010). doi: 10.1364/OL.35.002585, pp. 2585–2587.

[112] S. Mallat. *A Wavelet Tour of Signal Processing: The Sparse Way*. 3rd ed. Burlington, MA: Academic Press, 2009. ISBN: 9780123743701. DOI: 10.1016/B978-0-12-374370-1.X0001-8.

[113] A. L. G. Mandolesi. "Grassmann angles between real or complex subspaces". In: *arXiv preprint* (2021). arXiv:1910.00147. arXiv: 1910.00147 [math.MG].

[114] J. Mercer. "Functions of positive and negative type, and their connection with the theory of integral equations". In: *Philos. Trans. R. Soc. Lond. A* 209.441–458 (1909), pp. 415–446. DOI: 10.1098/rsta.1909.0016.

[115] T. Mikolov, K. Chen, G. Corrado, and J. Dean. "Efficient estimation of word representations in vector space". In: *ICLR 2013, Scottsdale, Arizona, USA*. Ed. by Y. Bengio and Y. LeCun. 2013. arXiv: 1301.3781 [cs.CL].

[116] A. Moshtaghpour, A. Velazco-Torrejon, A. W. Robinson, N. D. Browning, and A. I. Kirkland. "LoRePIE: ℓ_0 regularized extended ptychographical iterative engine for low-dose and fast electron ptychography". In: *Opt. Express* 33.5 (2025). doi: 10.1364/OE.551986, pp. 9357–9368.

[117] J. Nash. "The imbedding problem for Riemannian manifolds". In: *Ann. of Math. (2)* 63.1 (1956), pp. 20–63. DOI: 10.2307/1969989.

[118] J. J. O'Connor and E. F. Robertson. *Hermann Grassmann*. maths history.st-andrews.ac.uk/Biographies/Grassmann/. Accessed: 2025-11-06. 2005.

[119] M. Ozuysal, M. Calonder, V. Lepetit, and P. Fua. "Fast keypoint recognition using random ferns". In: *IEEE Trans. Pattern Anal. Mach. Intell.* 32.3 (2010). doi: 10.1109/TPAMI.2009.23, pp. 448–461.

[120] E. Parzen. "Extraction and detection problems and reproducing kernel Hilbert spaces". In: *J. Soc. Ind. Appl. Math. Ser. A Control* 1.1 (1962), pp. 35–62. DOI: 10.1137/0301004.

[121] Y. Plan and R. Vershynin. "One-bit compressed sensing by linear programming". In: *Commun. Pure Appl. Math.* 66.8 (2013), pp. 1275–1297. DOI: 10.1002/cpa.21442. arXiv: 1109.4299 [cs.IT].

[122] Y. Plan and R. Vershynin. "Robust 1-bit Compressed Sensing and Sparse Logistic Regression: A Convex Programming Approach". In: *IEEE Transactions on Information Theory* 59.1 (2013), pp. 482–494. DOI: 10.1109/TIT.2012.2207945.

[123] K. Purushottam and K. Harish. "Random Feature Maps for Dot Product Kernels". In: *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics*. Ed. by Neil D. Lawrence and Mark Girolami. Vol. 22. Proceedings of Machine Learning Research. La Palma, Canary Islands: PMLR, 2012, pp. 583–591.

[124] L. Qiu, Y. Zhang, and C.-K. Li. "Unitarily Invariant Metrics on the Grassmann Space". In: *SIAM Journal on Matrix Analysis and Applications* 27.2 (2005), pp. 507–531. DOI: 10.1137/040607605.

[125] A. Rahimi and B. Recht. "Random Features for Large-Scale Kernel Machines". In: *Advances in Neural Information Processing Systems*. Vol. 20. 2007, pp. 1177–1184.

[126] A. Rahimi and B. Recht. "Weighted Sums of Random Kitchen Sinks: Replacing Minimization with Randomization in Learning". In: *Advances in Neural Information Processing Systems*. Vol. 21. 2008, pp. 1313–1320.

[127] J. Rahnenführer, F. Ambrogi, F. Azuaje, A.-L. Boulesteix, H. Binder, S. Michiels, W. Sauerbrei, and L. McShane. "Statistical analysis of high-dimensional biomedical data: A gentle introduction to analytical goals, common approaches and challenges". In: *BMC Med.* 21 (2023), p. 182. DOI: 10.1186/s12916-023-02858-y.

[128] M. Raič. "A Multivariate Berry–Esseen Theorem with Explicit Constants". In: *Bernoulli* 25.4A (2019), pp. 2824–2853. DOI: 10.3150/18-BEJ1072.

[129] S. R. Rao, R. Tron, R. Vidal, and Y. Ma. "Motion Segmentation in the Presence of Outlying, Incomplete, or Corrupted Trajectories". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32.10 (2010), pp. 1832–1845.

[130] C. Rose and M.D. Smith. *Mathematical Statistics with Mathematica*. New York: Springer-Verlag, 2002.

[131] R. Rubinstein, A. M. Bruckstein, and M. Elad. "Dictionaries for sparse representation modeling". In: *Proc. IEEE* 98.6 (2010), pp. 1045–1057. DOI: 10.1109/JPROC.2010.2040551.

[132] A. Saad-Falcon, B. Ancelin, and J. Romberg. "Subspace Tracking with Dynamical Models on the Grassmannian". In: *2024 IEEE 13rd Sensor Array and Multichannel Signal Processing Workshop (SAM)*. IEEE. 2024, pp. 1–5.

[133] 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: *Proc. 2016 IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*. 2016, pp. 6215–6219. DOI: 10.1109/ICASSP.2016.7472872.

[134] B. Sch"olkopf, R. Herbrich, and A. J. Smola. "A generalized representer theorem". In: *Computational Learning Theory*. Ed. by D. Helmbold and B. Williamson. Vol. 2111. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2001, pp. 416–426. DOI: 10.1007/3-540-44581-1_27.

[135] V. Schellekens and L. Jacques. "Breaking the Waves: Asymmetric Random Periodic Features for Low-Bitrate Kernel Machines". In: *Information and Inference: A Journal of the IMA* 11.1 (2022), pp. 385–421.

[136] B. Schölkopf and A. J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Adaptive Computation and Machine Learning. Cambridge, MA: MIT Press, 2002, p. 626. ISBN: 978-0-262-19475-4.

[137] S. Schwarz and T. Tsiftsis. "Codebook training for trellis-based hierarchical Grassmannian classification". In: *IEEE Wireless Communications Letters* 11.3 (2021), pp. 636–640.

[138] J. Sevilla, L. Heim, A. Ho, T. Besiroglu, M. Hobbhahn, and P. Villalobos. "Compute trends across three eras of machine learning". In: *2022 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2022, pp. 1–8. DOI: 10.1109/IJCNN55064.2022.9891914. arXiv: 2202.05924 [cs.LG].

[139] A. Shehabi, S. J. Smith, A. Hubbard, A. Newkirk, N. Lei, M. A. B. Siddik, B. Holecek, J. Koomey, E. Masanet, and D. Sartor. *2024 United States data center energy usage report*. Technical report LBNL-2001637. Berkeley, CA: Lawrence Berkeley National Laboratory, 2024.

[140] A. Srivastava and E. Klassen. "Bayesian and geometric subspace tracking". In: *Advances in Applied Probability* 36.1 (2004), pp. 43–56.

[141] Stanford Institute for Human-Centered Artificial Intelligence. *The AI index 2026 annual report*. Report. Stanford University, 2026.

[142] E. Strubell, A. Ganesh, and A. McCallum. "Energy and policy considerations for deep learning in NLP". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2019, pp. 3645–3650. DOI: 10.18653/v1/P19-1355.

[143] Ł. Struski, P. Morkisz, P. Spurek, S. Rodriguez Bernabeu, and T. Trzci'nski. "Efficient GPU implementation of randomized SVD and its applications". In: *Expert Syst. Appl.* 248 (2024). doi: 10.1016/j.eswa.2024.123462, p. 123462.

[144] J Tachella and L. Jacques. "Learning to reconstruct signals from binary measurements alone". In: *Transactions on Machine Learning Research* (2023). Featured Certification. ISSN: 2835-8856.

[145] A. Tasissa and R. Lai. "Exact reconstruction of Euclidean distance geometry problem using low-rank matrix completion". In: *IEEE Trans. Inf. Theory* 65.5 (2019), pp. 3124–3144. DOI: 10.1109/TIT.2018.2881749. arXiv: 1804.04310 [cs.IT].

[146] J. H. Van Vleck and D. Middleton. "The Spectrum of Clipped Noise". In: *Proc. IEEE* 54.1 (1966), pp. 2–19.

[147] T. Vayer, E. Lasalle, R. Gribonval, and P. Gonçalves. "Compressive Recovery of Sparse Precision Matrices". In: *arXiv preprint arXiv:2311.04673* (2023).

[148] R. Vershynin. *High-Dimensional Probability: An Introduction with Applications in Data Science*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge: Cambridge University Press, 2018. ISBN: 978-1-108-41519-4. DOI: 10.1017/9781108231596.

[149] R. Vershynin. "Introduction to the Non-Asymptotic Analysis of Random Matrices". In: *Compressed Sensing: Theory and Applications*. Ed. by Y. C. Eldar and G. Kutyniok. Cambridge University Press, 2012, pp. 210–268.

[150] B. Wang, X. Liu, K. Xia, K. Ramamohanarao, and D. Tao. "Random Angular Projection for Fast Nearest Subspace Search". In: *Pacific Rim Conference on Multimedia*. Springer. 2018, pp. 15–26.

[151] X. Wang, S. Atev, J. Wright, and G. Lerman. "Fast Subspace Search via Grassmannian Based Hashing". In: *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*. 2013, pp. 2776–2783.

[152] Z. Wang, K. Müller, M. Filipovich, J. Launay, R. Ohana, G. Pariente, S. Mokaadi, C. Brossollet, F. Moreau, A. Cappelli, I. Poli, I. Carron, L. Daudet, F. Krzakala, and S. Gigan. *Streamlined optical training of large-scale modern deep learning architectures with direct feedback alignment*. arXiv:2409.12965v2 [cs.ET]. 2024. DOI: 10.48550/arXiv.2409.12965. arXiv: 2409.12965 [cs.ET].

[153] S. Watanabe, P. F. Lambert, C. A. Kulikowski, J. L. Buxton, and R. Walker. "Evaluation and Selection of Variables in Pattern Recognition". In: *Computer and Information Sciences*. Ed. by J. Tou. Vol. 2. New York: Academic Press, 1967, pp. 91–122.

[154] S. Watanabe and N. Pakvasa. "Subspace Method of Pattern Recognition". In: *Proc. 1st Int. Conf. Pattern Recognit. (ICPR)*. 1973, pp. 25–32.

[155] D. Wei, X. Shen, Q. Sun, X. Gao, and W. Yan. "Prototype Learning and Collaborative Representation Using Grassmann Manifolds for Image Set Classification". In: *Pattern Recognit.* 100 (2020), p. 107123.

[156] H. Whitney. "Differentiable manifolds". In: *Ann. of Math. (2)* 37.3 (1936), pp. 645–680. DOI: 10.2307/1968482.

[157] L. Wolf and A. Shashua. "Learning Distance Functions Using Equivalence Relations". In: *J. Mach. Learn. Res.* 4 (2003), pp. 913–931.

[158] Y.-C. Wong. "Differential Geometry of Grassmann Manifolds". In: *Proceedings of the National Academy of Sciences* 57.3 (1967), pp. 589–594.

[159] D. P. Woodruff. "Sketching as a tool for numerical linear algebra". In: *Found. Trends Theor. Comput. Sci.* 10.1–2 (2014). doi: 10.1561/0400000060, pp. 1–157.

[160] C.-J. Wu, R. Raghavendra, U. Gupta, B. Acun, N. Ardalani, K. Maeng, G. Chang, F. Aga, J. Huang, C. Bai, M. Gschwind, A. Gupta, M. Ott, A. Melnikov, S. Candido, D. Brooks, G. Chauhan, B. Lee, H.-H. S. Lee, B. Akyildiz, M. Balandat, J. Spisak, R. Jain, M. Rabbat, and K. Hazelwood. "Sustainable AI: Environmental implications, challenges and opportunities". In: *Proceedings of Machine Learning and Systems*. Vol. 4. 2022, pp. 795–813.

[161] C. Xu and L. Jacques. "Quantized compressive sensing with RIP matrices: The benefit of dithering". In: *Inf. Inference* 9.3 (2020), pp. 543–586. DOI: 10.1093/imaiai/iaz021. arXiv: 1801.05870 [cs.IT].

[162] K. Ye and L.-H. Lim. "Schubert varieties and distances between subspaces of different dimensions". In: *SIAM J. Matrix Anal. Appl.* 37.3 (2016), pp. 1176–1197. DOI: 10.1137/15M1054201.

[163] F. 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 29*. 2016, pp. 1975–1983.

[164] J. Zhang, A. May, T. Dao, and C. Re. "Low-precision random Fourier features for memory-constrained kernel approximation". In: *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics*. Vol. 89. Proceedings of Machine Learning Research. PMLR, 2019, pp. 1264–1274.

[165] A. Zymnis, S. Boyd, and E. Candès. "Compressed sensing with quantized measurements". In: *IEEE Signal Process. Lett.* 17.2 (2010), pp. 149–152. DOI: 10.1109/LSP.2009.2035667.